

---

# **python-gitlab Documentation**

***Release 4.4.0***

**Gauvain Pocentek, Mika Mäenpää**

**Apr 25, 2024**



## TABLE OF CONTENTS

<b>1</b>	<b>Features</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Using the docker images</b>	<b>7</b>
3.1	Usage inside GitLab CI . . . . .	7
3.2	Building the image . . . . .	8
<b>4</b>	<b>Bug reports</b>	<b>9</b>
<b>5</b>	<b>Gitter Community Chat</b>	<b>11</b>
<b>6</b>	<b>Documentation</b>	<b>13</b>
6.1	Build the docs . . . . .	13
<b>7</b>	<b>Contributing</b>	<b>15</b>
7.1	Getting started with the CLI . . . . .	15
7.2	Getting started with the API . . . . .	20
7.3	Advanced usage . . . . .	27
7.4	CLI examples . . . . .	31
7.5	API examples . . . . .	36
7.6	API reference (gitlab package) . . . . .	160
7.7	CLI reference (gitlab command) . . . . .	476
7.8	Lower-level APIs . . . . .	818
7.9	CHANGELOG . . . . .	820
7.10	Release notes . . . . .	1008
7.11	FAQ . . . . .	1011
<b>8</b>	<b>Indices and tables</b>	<b>1013</b>
	<b>Python Module Index</b>	<b>1015</b>
	<b>Index</b>	<b>1017</b>



`python-gitlab` is a Python package providing access to the GitLab server API. It supports the v4 API of GitLab, and provides a CLI tool (`gitlab`).



## **FEATURES**

`python-gitlab` enables you to:

- write Pythonic code to manage your GitLab resources.
- pass arbitrary parameters to the GitLab API. Simply follow GitLab's docs on what parameters are available.
- access arbitrary endpoints as soon as they are available on GitLab, by using lower-level API methods.
- use persistent requests sessions for authentication, proxy and certificate handling.
- handle smart retries on network and server errors, with rate-limit handling.
- flexible handling of paginated responses, including lazy iterators.
- automatically URL-encode paths and parameters where needed.
- automatically convert some complex data structures to API attribute types
- merge configuration from config files, environment variables and arguments.





## INSTALLATION

As of 4.0.0, `python-gitlab` is compatible with Python 3.8+.

Use `pip` to install the latest stable version of `python-gitlab`:

```
$ pip install --upgrade python-gitlab
```

The current development version is available on both [GitHub.com](https://github.com/python-gitlab/python-gitlab) and [GitLab.com](https://gitlab.com/python-gitlab/python-gitlab), and can be installed directly from the git repository:

```
$ pip install git+https://github.com/python-gitlab/python-gitlab.git
```

From GitLab:

```
$ pip install git+https://gitlab.com/python-gitlab/python-gitlab.git
```



## USING THE DOCKER IMAGES

`python-gitlab` provides Docker images in two flavors, based on the Alpine and Debian slim python [base images](#). The default tag is `alpine`, but you can explicitly use the alias (see below).

The alpine image is smaller, but you may want to use the Debian-based slim tag (currently based on `-slim-bullseye`) if you are running into issues or need a more complete environment with a bash shell, such as in CI jobs.

The images are published on the GitLab registry, for example:

- `registry.gitlab.com/python-gitlab/python-gitlab:latest` (latest, alpine alias)
- `registry.gitlab.com/python-gitlab/python-gitlab:alpine` (latest alpine)
- `registry.gitlab.com/python-gitlab/python-gitlab:slim-bullseye` (latest slim-bullseye)
- `registry.gitlab.com/python-gitlab/python-gitlab:v3.2.0` (alpine alias)
- `registry.gitlab.com/python-gitlab/python-gitlab:v3.2.0-alpine`
- `registry.gitlab.com/python-gitlab/python-gitlab:v3.2.0-slim-bullseye`

You can run the Docker image directly from the GitLab registry:

```
$ docker run -it --rm registry.gitlab.com/python-gitlab/python-gitlab:latest <command> ..  
↪ .
```

For example, to get a project on GitLab.com (without authentication):

```
$ docker run -it --rm registry.gitlab.com/python-gitlab/python-gitlab:latest project get  
↪ --id gitlab-org/gitlab
```

You can also mount your own config file:

```
$ docker run -it --rm -v /path/to/python-gitlab.cfg:/etc/python-gitlab.cfg registry.  
↪ gitlab.com/python-gitlab/python-gitlab:latest <command> ...
```

### 3.1 Usage inside GitLab CI

If you want to use the Docker image directly inside your GitLab CI as an image, you will need to override the `entrypoint`, as noted in the [official GitLab documentation](#):

```
Job Name:  
  image:  
    name: registry.gitlab.com/python-gitlab/python-gitlab:latest
```

(continues on next page)

(continued from previous page)

```
entrypoint: [""]
before_script:
  gitlab --version
script:
  gitlab <command>
```

## 3.2 Building the image

To build your own image from this repository, run:

```
$ docker build -t python-gitlab:latest .
```

Run your own image:

```
$ docker run -it --rm python-gitlab:latest <command> ...
```

Build a Debian slim-based image:

```
$ docker build -t python-gitlab:latest --build-arg PYTHON_FLAVOR=slim-bullseye .
```

## **BUG REPORTS**

Please report bugs and feature requests at <https://github.com/python-gitlab/python-gitlab/issues>.



## **GITTER COMMUNITY CHAT**

We have a [gitter](https://gitter.im/python-gitlab/Lobby) community chat available at <https://gitter.im/python-gitlab/Lobby>, which you can also directly access via the Open Chat button below.

If you have a simple question, the community might be able to help already, without you opening an issue. If you regularly use python-gitlab, we also encourage you to join and participate. You might discover new ideas and use cases yourself!





## DOCUMENTATION

The full documentation for CLI and API is available on [readthedocs](#).

### 6.1 Build the docs

We use `tox` to manage our environment and build the documentation:

```
pip install tox
tox -e docs
```



## CONTRIBUTING

For guidelines for contributing to `python-gitlab`, refer to [CONTRIBUTING.rst](#).

### 7.1 Getting started with the CLI

`python-gitlab` provides a **gitlab** command-line tool to interact with GitLab servers.

This is especially convenient for running quick ad-hoc commands locally, easily interacting with the API inside GitLab CI, or with more advanced shell scripting when integrating with other tooling.

#### 7.1.1 Configuration

`gitlab` allows setting configuration options via command-line arguments, environment variables, and configuration files.

For a complete list of global CLI options and their environment variable equivalents, see *CLI reference (gitlab command)*.

With no configuration provided, `gitlab` will default to unauthenticated requests against [GitLab.com](#).

With no configuration but running inside a GitLab CI job, it will default to authenticated requests using the current job token against the current instance (via `CI_SERVER_URL` and `CI_JOB_TOKEN` environment variables).

**Warning:** Please note the job token has very limited permissions and can only be used with certain endpoints. You may need to provide a personal access token instead.

When you provide configuration, values are evaluated with the following precedence:

1. Explicitly provided CLI arguments,
2. Environment variables,
3. Configuration files:
  - a. explicitly defined config files:
    - i. via the `--config-file` CLI argument,
    - ii. via the `PYTHON_GITLAB_CFG` environment variable,
  - b. user-specific config file,
  - c. system-level config file,
4. Environment variables always present in CI (`CI_SERVER_URL`, `CI_JOB_TOKEN`).

Additionally, authentication will take the following precedence when multiple options or environment variables are present:

1. Private token,
2. OAuth token,
3. CI job token.

## Configuration files

gitlab looks up 3 configuration files by default:

### The `PYTHON_GITLAB_CFG` environment variable

An environment variable that contains the path to a configuration file.

### `/etc/python-gitlab.cfg`

System-wide configuration file

### `~/.python-gitlab.cfg`

User configuration file

You can use a different configuration file with the `--config-file` option.

**Warning:** If the `PYTHON_GITLAB_CFG` environment variable is defined and the target file exists, it will be the only configuration file parsed by gitlab.

If the environment variable is defined and the target file cannot be accessed, gitlab will fail explicitly.

## Configuration file format

The configuration file uses the INI format. It contains at least a `[global]` section, and a specific section for each GitLab server. For example:

```
[global]
default = somewhere
ssl_verify = true
timeout = 5

[somewhere]
url = https://some.whe.re
private_token = vTbFeqJYCY3sibBP7BZM
api_version = 4

[elsewhere]
url = http://else.whe.re:8080
private_token = helper: path/to/helper.sh
timeout = 1
```

The `default` option of the `[global]` section defines the GitLab server to use if no server is explicitly specified with the `--gitlab` CLI option.

The `[global]` section also defines the values for the default connection parameters. You can override the values in each GitLab server section.

Table 1: Global options

Option	Possible values	Description
ssl_verify	True, False, or a str	Verify the SSL certificate. Set to False to disable verification, though this will create warnings. Any other value is interpreted as path to a CA_BUNDLE file or directory with certificates of trusted CAs.
timeout	Integer	Number of seconds to wait for an answer before failing.
api_version	4	The API version to use to make queries. Only 4 is available since 1.5.0.
per_page	Integer between 1 and 100	The number of items to return in listing queries. GitLab limits the value at 100.
user_agent	str	A string defining a custom user agent to use when gitlab makes requests.

You must define the `url` in each GitLab server section.

**Warning:** Note that a url that results in 301/302 redirects will raise an error, so it is highly recommended to use the final destination in the `url` field. For example, if the GitLab server you are using redirects requests from http to https, make sure to use the `https://` protocol in the URL definition.

A URL that redirects using 301/302 (rather than 307/308) will most likely [cause malformed POST and PUT requests](#). python-gitlab will therefore raise a `RedirectionError` when it encounters a redirect which it believes will cause such an error, to avoid confusion between successful GET and failing POST/PUT requests on the same instance.

Only one of `private_token`, `oauth_token` or `job_token` should be defined. If neither are defined an anonymous request will be sent to the Gitlab server, with very limited permissions.

We recommend that you use [Credential helpers](#) to securely store your tokens.

Table 2: GitLab server options

Option	Description
url	URL for the GitLab server. Do <b>NOT</b> use a URL which redirects.
private_token	Your user token. Login/password is not supported. Refer to <a href="#">the official documentation</a> to learn how to obtain a token.
oauth_token	An OAuth token for authentication. The Gitlab server must be configured to support this authentication method.
job_token	Your job token. See <a href="#">the official documentation</a> to learn how to obtain a token.
api_version	GitLab API version to use. Only 4 is available since 1.5.0.

## Credential helpers

For all configuration options that contain secrets (for example, `personal_token`, `oauth_token`, `job_token`), you can specify a helper program to retrieve the secret indicated by a `helper:` prefix. This allows you to fetch values from a local keyring store or cloud-hosted vaults such as Bitwarden. Environment variables are expanded if they exist and `~` expands to your home directory.

It is expected that the helper program prints the secret to standard output. To use shell features such as piping to retrieve the value, you will need to use a wrapper script; see below.

Example for a [keyring](#) helper:

```
[global]
default = somewhere
ssl_verify = true
timeout = 5

[somewhere]
url = http://somewhe.re
private_token = helper: keyring get Service Username
timeout = 1
```

Example for a `pass` helper with a wrapper script:

```
[global]
default = somewhere
ssl_verify = true
timeout = 5

[somewhere]
url = http://somewhe.re
private_token = helper: /path/to/helper.sh
timeout = 1
```

In `/path/to/helper.sh`:

```
#!/bin/bash
pass show path/to/credentials | head -n 1
```

## 7.1.2 CLI

### Objects and actions

The `gitlab` command expects two mandatory arguments. The first one is the type of object that you want to manipulate. The second is the action that you want to perform. For example:

```
$ gitlab project list
```

Use the `--help` option to list the available object types and actions:

```
$ gitlab --help
$ gitlab project --help
```

Some actions require additional parameters. Use the `--help` option to list mandatory and optional arguments for an action:

```
$ gitlab project create --help
```

## Optional arguments

Use the following optional arguments to change the behavior of `gitlab`. These options must be defined before the mandatory arguments.

- verbose, -v**  
Outputs detail about retrieved objects. Available for legacy (default) output only.
- config-file, -c**  
Path to a configuration file.
- gitlab, -g**  
ID of a GitLab server defined in the configuration file.
- output, -o**  
Output format. Defaults to a custom format. Can also be `yaml` or `json`.

---

**Important:** The `PyYAML` package is required to use the `yaml` output option. You need to install it explicitly using `pip install python-gitlab[yaml]`

---

- fields, -f**  
Comma-separated list of fields to display (`yaml` and `json` output formats only). If not used, all the object fields are displayed.

Example:

```
$ gitlab -o yaml -f id,permissions -g elsewhere -c /tmp/gl.cfg project list
```

## Reading values from files

You can make `gitlab` read values from files instead of providing them on the command line. This is handy for values containing new lines for instance:

```
$ cat > /tmp/description << EOF
This is the description of my project.

It is obviously the best project around
EOF
$ gitlab project create --name SuperProject --description @/tmp/description
```

If you want to explicitly pass an argument starting with `@`, you can escape it using `@@`:

```
$ gitlab project-tag list --project-id somenamespace/myproject
...
name: @at-started-tag
...
$ gitlab project-tag delete --project-id somenamespace/myproject --name '@@at-started-tag'
```

### 7.1.3 Enabling shell autocompletion

To get autocompletion, you'll need to install the package with the extra "autocompletion":

```
pip install python_gitlab[autocompletion]
```

Add the appropriate command below to your shell's config file so that it is run on startup. You will likely have to restart or re-login for the autocompletion to start working.

#### Bash

```
eval "$(register-python-argcomplete gitlab)"
```

#### tcsh

```
eval ``register-python-argcomplete --shell tcsh gitlab``
```

#### fish

```
register-python-argcomplete --shell fish gitlab | .
```

#### Zsh

**Warning:** Zsh autocompletion support is broken right now in the argcomplete python package. Perhaps it will be fixed in a future release of argcomplete at which point the following instructions will enable autocompletion in zsh.

To activate completions for zsh you need to have bashcompinit enabled in zsh:

```
autoload -U bashcompinit
bashcompinit
```

Afterwards you can enable completion for gitlab:

```
eval "$(register-python-argcomplete gitlab)"
```

## 7.2 Getting started with the API

python-gitlab only supports GitLab API v4.



### 7.2.1 gitlab.Gitlab class

To connect to GitLab.com or another GitLab instance, create a `gitlab.Gitlab` object:

**Hint:** You can use different types of tokens for authenticated requests against the GitLab API. You will most likely want to use a resource (project/group) access token or a personal access token.

For the full list of available options and how to obtain these tokens, please see <https://docs.gitlab.com/ee/api/index.html#authentication>.

```
import gitlab

# anonymous read-only access for public resources (GitLab.com)
gl = gitlab.Gitlab()

# anonymous read-only access for public resources (self-hosted GitLab instance)
gl = gitlab.Gitlab('https://gitlab.example.com')

# private token or personal token authentication (GitLab.com)
gl = gitlab.Gitlab(private_token='JVNSEs8EwWRx5yDxM5q')

# private token or personal token authentication (self-hosted GitLab instance)
gl = gitlab.Gitlab(url='https://gitlab.example.com', private_token='JVNSEs8EwWRx5yDxM5q
↪')

# oauth token authentication
gl = gitlab.Gitlab('https://gitlab.example.com', oauth_token='my_long_token_here')

# job token authentication (to be used in CI)
# bear in mind the limitations of the API endpoints it supports:
# https://docs.gitlab.com/ee/ci/jobs/ci_job_token.html
import os
gl = gitlab.Gitlab('https://gitlab.example.com', job_token=os.environ['CI_JOB_TOKEN'])

# Define your own custom user agent for requests
gl = gitlab.Gitlab('https://gitlab.example.com', user_agent='my-package/1.0.0')

# make an API request to create the gl.user object. This is not required but may be
↪useful
# to validate your token authentication. Note that this will not work with job tokens.
gl.auth()

# Enable "debug" mode. This can be useful when trying to determine what
# information is being sent back and forth to the GitLab server.
# Note: this will cause credentials and other potentially sensitive
# information to be printed to the terminal.
gl.enable_debug()
```

You can also use configuration files to create `gitlab.Gitlab` objects:

```
gl = gitlab.Gitlab.from_config('somewhere', ['/tmp/gl.cfg'])
```

See the [Configuration](#) section for more information about configuration files.

**Warning:** Note that a url that results in 301/302 redirects will raise an error, so it is highly recommended to use the final destination in the `url` field. For example, if the GitLab server you are using redirects requests from http to https, make sure to use the `https://` protocol in the URL definition.

A URL that redirects using 301/302 (rather than 307/308) will most likely cause malformed POST and PUT requests. python-gitlab will therefore raise a `RedirectionError` when it encounters a redirect which it believes will cause such an error, to avoid confusion between successful GET and failing POST/PUT requests on the same instance.

### Note on password authentication

GitLab has long removed password-based basic authentication. You can currently still use the [resource owner password credentials](#) flow to obtain an OAuth token.

However, we do not recommend this as it will not work with 2FA enabled, and GitLab is removing ROPC-based flows without client IDs in a future release. We recommend you obtain tokens for automated workflows as linked above or obtain a session cookie from your browser.

For a python example of password authentication using the ROPC-based OAuth2 flow, see [this Ansible snippet](#).

## 7.2.2 Managers

The `gitlab.Gitlab` class provides managers to access the GitLab resources. Each manager provides a set of methods to act on the resources. The available methods depend on the resource type.

Examples:

```
# list all the projects
projects = gl.projects.list(iterator=True)
for project in projects:
    print(project)

# get the group with id == 2
group = gl.groups.get(2)
for project in group.projects.list(iterator=True):
    print(project)
```

**Warning:** Calling `list()` without any arguments will by default not return the complete list of items. Use either the `get_all=True` or `iterator=True` parameters to get all the items when using listing methods. See the [Pagination](#) section for more information.

```
# create a new user
user_data = {'email': 'jen@foo.com', 'username': 'jen', 'name': 'Jen'}
user = gl.users.create(user_data)
print(user)
```

**Note:** python-gitlab attempts to sync the required, optional, and mutually exclusive attributes for resource creation and update with the upstream API.

You are encouraged to follow upstream API documentation for each resource to find these - each resource documented here links to the corresponding upstream resource documentation at the top of the page.

The attributes of objects are defined upon object creation, and depend on the GitLab API itself. To list the available information associated with an object use the `attributes` attribute:

```
project = gl.projects.get(1)
print(project.attributes)
```

Some objects also provide managers to access related GitLab resources:

```
# list the issues for a project
project = gl.projects.get(1)
issues = project.issues.list(get_all=True)
```

python-gitlab allows to send any data to the GitLab server when making queries. In case of invalid or missing arguments python-gitlab will raise an exception with the GitLab server error message:

```
>>> gl.projects.list(sort='invalid value')
...
GitlabListError: 400: sort does not have a valid value
```

You can use the `query_parameters` argument to send arguments that would conflict with python or python-gitlab when using them as kwargs:

```
gl.user_activities.list(from_='2019-01-01', iterator=True) ## invalid

gl.user_activities.list(query_parameters={'from': '2019-01-01'}, iterator=True) # OK
```

### 7.2.3 Gitlab Objects

You can update or delete a remote object when it exists locally:

```
# update the attributes of a resource
project = gl.projects.get(1)
project.wall_enabled = False
# don't forget to apply your changes on the server:
project.save()

# delete the resource
project.delete()
```

Some classes provide additional methods, allowing more actions on the GitLab resources. For example:

```
# star a git repository
project = gl.projects.get(1)
project.star()
```

You can print a Gitlab Object. For example:

```
project = gl.projects.get(1)
print(project)
```

(continues on next page)

(continued from previous page)

```
# Or in a prettier format.
project.pprint()

# Or explicitly via `pformat()`. This is equivalent to the above.
print(project.pformat())
```

You can also extend the object if the parameter isn't explicitly listed. For example, if you want to update a field that has been newly introduced to the Gitlab API, setting the value on the object is accepted:

```
issues = project.issues.list(state='opened')
for issue in issues:
    issue.my_super_awesome_feature_flag = "random_value"
    issue.save()
```

### As a dictionary

You can get a dictionary representation copy of the Gitlab Object. Modifications made to the dictionary will have no impact on the GitLab Object.

- `asdict()` method. Returns a dictionary representation of the Gitlab object.
- **attributes property.** Returns a dictionary representation of the Gitlab object. Also returns any relevant parent object attributes.

```
project = gl.projects.get(1)
project_dict = project.asdict()

# Or a dictionary representation also containing some of the parent attributes
issue = project.issues.get(1)
attribute_dict = issue.attributes

# The following will return the same value
title = issue.title
title = issue.attributes["title"]
```

---

**Hint:** This can be used to access attributes that clash with python-gitlab's own methods or managers. Note that:

`attributes` returns the parent object attributes that are defined in `object._from_parent_attrs`. For example, a `ProjectIssue` object will have a `project_id` key in the dictionary returned from `attributes` but `asdict()` will not.

---

### As JSON

You can get a JSON string representation of the Gitlab Object. For example:

```
project = gl.projects.get(1)
print(project.to_json())
# Use arguments supported by `json.dump()`
print(project.to_json(sort_keys=True, indent=4))
```

## 7.2.4 Base types

The gitlab package provides some base types.

- `gitlab.Gitlab` is the primary class, handling the HTTP requests. It holds the GitLab URL and authentication information.
- `gitlab.base.RESTObject` is the base class for all the GitLab v4 objects. These objects provide an abstraction for GitLab resources (projects, groups, and so on).
- `gitlab.base.RESTManager` is the base class for v4 objects managers, providing the API to manipulate the resources and their attributes.

## 7.2.5 Lazy objects

To avoid useless API calls to the server you can create lazy objects. These objects are created locally using a known ID, and give access to other managers and methods.

The following example will only make one API call to the GitLab server to star a project (the previous example used 2 API calls):

```
# star a git repository
project = gl.projects.get(1, lazy=True) # no API call
project.star() # API call
```

## 7.2.6 head() methods

All endpoints that support `get()` and `list()` also support a `head()` method. In this case, the server responds only with headers and not the response JSON or body. This allows more efficient API calls, such as checking repository file size without fetching its content.

---

**Note:** In some cases, GitLab may omit specific headers. See more in the [Pagination](#) section.

---

```
# See total number of personal access tokens for current user
gl.personal_access_tokens.head()
print(headers["X-Total"])

# See returned content-type for project GET endpoint
headers = gl.projects.head("gitlab-org/gitlab")
print(headers["Content-Type"])
```

## 7.2.7 Pagination

You can use pagination to iterate over long lists. All the Gitlab objects listing methods support the `page` and `per_page` parameters:

```
ten_first_groups = gl.groups.list(page=1, per_page=10)
```

**Warning:** The first page is page 1, not page 0.

By default GitLab does not return the complete list of items. Use the `get_all` parameter to get all the items when using listing methods:

```
all_groups = gl.groups.list(get_all=True)

all_owned_projects = gl.projects.list(owned=True, get_all=True)
```

You can define the `per_page` value globally to avoid passing it to every `list()` method call:

```
gl = gitlab.Gitlab(url, token, per_page=50)
```

Gitlab allows to also use keyset pagination. You can supply it to your project listing, but you can also do so globally. Be aware that GitLab then also requires you to only use supported order options. At the time of writing, only `order_by="id"` works.

```
gl = gitlab.Gitlab(url, token, pagination="keyset", order_by="id", per_page=100)
gl.projects.list()
```

Reference: <https://docs.gitlab.com/ce/api/README.html#keyset-based-pagination>

`list()` methods can also return a generator object, by passing the argument `iterator=True`, which will handle the next calls to the API when required. This is the recommended way to iterate through a large number of items:

```
items = gl.groups.list(iterator=True)
for item in items:
    print(item.attributes)
```

The generator exposes extra listing information as received from the server:

- `current_page`: current page number (first page is 1)
- `prev_page`: if `None` the current page is the first one
- `next_page`: if `None` the current page is the last one
- `per_page`: number of items per page
- `total_pages`: total number of pages available. This may be a `None` value.
- `total`: total number of items in the list. This may be a `None` value.

---

**Note:** For performance reasons, if a query returns more than 10,000 records, GitLab does not return the `total_pages` or `total` headers. In this case, `total_pages` and `total` will have a value of `None`.

For more information see: [https://docs.gitlab.com/ee/user/gitlab\\_com/index.html#pagination-response-headers](https://docs.gitlab.com/ee/user/gitlab_com/index.html#pagination-response-headers)

---

---

**Note:** Prior to python-gitlab 3.6.0 the argument `as_list` was used instead of `iterator`. `as_list=False` is the equivalent of `iterator=True`.

---

---

**Note:** If `page` and `iterator=True` are used together, the latter is ignored.

---

### 7.2.8 Sudo

If you have the administrator status, you can use `sudo` to act as another user. For example:

```
p = gl.projects.create({'name': 'awesome_project'}, sudo='user1')
```

### 7.2.9 Logging

To enable debug logging from the underlying `requests` and `http.client` calls, you can use `enable_debug()` on your Gitlab instance. For example:

```
import os
import gitlab

gl = gitlab.Gitlab(private_token=os.getenv("GITLAB_TOKEN"))
gl.enable_debug()
```

By default, python-gitlab will mask the token used for authentication in logging output. If you'd like to debug credentials sent to the API, you can disable masking explicitly:

```
gl.enable_debug(mask_credentials=False)
```

### 7.2.10 Attributes in updated objects

When methods manipulate an existing object, such as with `refresh()` and `save()`, the object will only have attributes that were returned by the server. In some cases, such as when the initial request fetches attributes that are needed later for additional processing, this may not be desired:

```
project = gl.projects.get(1, statistics=True)
project.statistics

project.refresh()
project.statistics # AttributeError
```

To avoid this, either copy the object/attributes before calling `refresh()/save()` or subsequently perform another `get()` call as needed, to fetch the attributes you want.

## 7.3 Advanced usage

### 7.3.1 Using a custom session

python-gitlab relies on `requests.Session` objects to perform all the HTTP requests to the GitLab servers.

You can provide a custom session to create `gitlab.Gitlab` objects:

```
import gitlab
import requests

session = requests.Session()
gl = gitlab.Gitlab(session=session)
```

(continues on next page)

(continued from previous page)

```
# or when instantiating from configuration files
gl = gitlab.Gitlab.from_config('somewhere', ['/tmp/gl.cfg'], session=session)
```

Reference: <https://requests.readthedocs.io/en/latest/user/advanced/#session-objects>

### 7.3.2 Context manager

You can use Gitlab objects as context managers. This makes sure that the `requests.Session` object associated with a Gitlab instance is always properly closed when you exit a `with` block:

```
with gitlab.Gitlab(host, token) as gl:
    gl.projects.list()
```

**Warning:** The context manager will also close the custom `Session` object you might have used to build the Gitlab instance.

### 7.3.3 netrc authentication

python-gitlab reads credentials from `.netrc` files via the `requests` backend only if you do not provide any other type of authentication yourself.

If you'd like to disable reading netrc files altogether, you can follow *Using a custom session* and explicitly set `trust_env=False` as described in the `requests` documentation.

```
import gitlab
import requests

session = requests.Session(trust_env=False)
gl = gitlab.Gitlab(session=session)
```

Reference: <https://requests.readthedocs.io/en/latest/user/authentication/#netrc-authentication>

### 7.3.4 Proxy configuration

python-gitlab accepts the standard `http_proxy`, `https_proxy` and `no_proxy` environment variables via the `requests` backend. Uppercase variables are also supported.

For more granular control, you can also explicitly set proxies by *Using a custom session* as described in the `requests` documentation.

Reference: <https://requests.readthedocs.io/en/latest/user/advanced/#proxies>



### 7.3.5 SSL certificate verification

python-gitlab relies on the CA certificate bundle in the `certifi` package that comes with the requests library.

If you need python-gitlab to use your system CA store instead, you can provide the path to the CA bundle in the `REQUESTS_CA_BUNDLE` environment variable.

Reference: <https://requests.readthedocs.io/en/latest/user/advanced/#ssl-cert-verification>

### 7.3.6 Client side certificate

The following sample illustrates how to use a client-side certificate:

```
import gitlab
import requests

session = requests.Session()
session.cert = ('/path/to/client.cert', '/path/to/client.key')
gl = gitlab.Gitlab(url, token, api_version=4, session=session)
```

Reference: <https://requests.readthedocs.io/en/latest/user/advanced/#client-side-certificates>

### 7.3.7 Rate limits

python-gitlab obeys the rate limit of the GitLab server by default. On receiving a 429 response (Too Many Requests), python-gitlab sleeps for the amount of time in the Retry-After header that GitLab sends back. If GitLab does not return a response with the Retry-After header, python-gitlab will perform an exponential backoff.

If you don't want to wait, you can disable the rate-limiting feature, by supplying the `obey_rate_limit` argument.

```
import gitlab
import requests

gl = gitlab.Gitlab(url, token, api_version=4)
gl.projects.list(get_all=True, obey_rate_limit=False)
```

If you do not disable the rate-limiting feature, you can supply a custom value for `max_retries`; by default, this is set to 10. To retry without bound when throttled, you can set this parameter to -1. This parameter is ignored if `obey_rate_limit` is set to False.

```
import gitlab
import requests

gl = gitlab.Gitlab(url, token, api_version=4)
gl.projects.list(get_all=True, max_retries=12)
```

**Warning:** You will get an Exception, if you then go over the rate limit of your GitLab instance.

### 7.3.8 Transient errors

GitLab server can sometimes return a transient HTTP error. python-gitlab can automatically retry in such case, when `retry_transient_errors` argument is set to `True`. When enabled, HTTP error codes 500 (Internal Server Error), 502 (502 Bad Gateway), 503 (Service Unavailable), 504 (Gateway Timeout), and Cloudflare errors (520-530) are retried.

Additionally, HTTP error code 409 (Conflict) is retried if the reason is a [Resource lock](#).

It will retry until reaching the `max_retries` value. By default, `retry_transient_errors` is set to `False` and an exception is raised for these errors.

```
import gitlab
import requests

gl = gitlab.Gitlab(url, token, api_version=4)
gl.projects.list(get_all=True, retry_transient_errors=True)
```

The default `retry_transient_errors` can also be set on the Gitlab object and overridden by individual API calls.

```
import gitlab
import requests
gl = gitlab.Gitlab(url, token, api_version=4, retry_transient_errors=True)
gl.projects.list(get_all=True) # retries due to default
↪ value
gl.projects.list(get_all=True, retry_transient_errors=False) # does not retry
```

### 7.3.9 Timeout

python-gitlab will by default use the `timeout` option from its configuration for all requests. This is passed downwards to the `requests` module at the time of making the HTTP request. However if you would like to override the global timeout parameter for a particular call, you can provide the `timeout` parameter to that API invocation:

```
import gitlab

gl = gitlab.Gitlab(url, token, api_version=4)
gl.projects.import_github(ACCESS_TOKEN, 123456, "root", timeout=120.0)
```

### 7.3.10 Typing

Generally, python-gitlab is a fully typed package. However, currently you may still need to do some [type narrowing](#) on your own, such as for nested API responses and Union return types. For example:

```
from typing import TYPE_CHECKING

import gitlab

gl = gitlab.Gitlab(url, token, api_version=4)
license = gl.get_license()

if TYPE_CHECKING:
    assert isinstance(license["plan"], str)
```

## 7.4 CLI examples

See also:

For a complete list of objects and actions available, see *CLI reference (gitlab command)*.

### 7.4.1 CI Lint

**ci-lint has been Removed in Gitlab 16, use project-ci-lint instead**

Lint a CI YAML configuration from a string:

---

**Note:** To see output, you will need to use the `-v/--verbose` flag.

To exit with non-zero on YAML lint failures instead, use the `validate` subcommand shown below.

---

```
$ gitlab --verbose ci-lint create --content \
"---
test:
  script:
    - echo hello
"
```

Lint a CI YAML configuration from a file (see *Reading values from files*):

```
$ gitlab --verbose ci-lint create --content @.gitlab-ci.yml
```

Validate a CI YAML configuration from a file (lints and exits with non-zero on failure):

```
$ gitlab ci-lint validate --content @.gitlab-ci.yml
```

### 7.4.2 Project CI Lint

Lint a project's CI YAML configuration:

```
$ gitlab --verbose project-ci-lint create --project-id group/my-project --content @.
↪gitlab-ci.yml
```

Validate a project's CI YAML configuration (lints and exits with non-zero on failure):

```
$ gitlab project-ci-lint validate --project-id group/my-project --content @.gitlab-ci.yml
```

Lint a project's current CI YAML configuration:

```
$ gitlab --verbose project-ci-lint get --project-id group/my-project
```

Lint a project's current CI YAML configuration on a specific branch:

```
$ gitlab --verbose project-ci-lint get --project-id group/my-project --ref my-branch
```

### 7.4.3 Projects

List the projects (paginated):

```
$ gitlab project list
```

List all the projects:

```
$ gitlab project list --get-all
```

List all projects of a group:

```
$ gitlab group-project list --get-all --group-id 1
```

List all projects of a group and its subgroups:

```
$ gitlab group-project list --get-all --include-subgroups true --group-id 1
```

Limit to 5 items per request, display the 1st page only

```
$ gitlab project list --page 1 --per-page 5
```

Get a specific project (id 2):

```
$ gitlab project get --id 2
```

### 7.4.4 Users

Get a specific user by id:

```
$ gitlab user get --id 3
```

Create a user impersonation token (admin-only):

```
gitlab user-impersonation-token create --user-id 2 --name test-token --scopes api,read_
↪user
```

### 7.4.5 Deploy tokens

Create a deploy token for a project:

```
$ gitlab -v project-deploy-token create --project-id 2 \
  --name bar --username root --expires-at "2021-09-09" --scopes "api,read_repository"
```

List deploy tokens for a group:

```
$ gitlab -v group-deploy-token list --group-id 3
```

## 7.4.6 Personal access tokens

List the current user's personal access tokens (or all users' tokens, if admin):

```
$ gitlab -v personal-access-token list
```

Revoke a personal access token by id:

```
$ gitlab personal-access-token delete --id 1
```

Revoke the personal access token currently used:

```
$ gitlab personal-access-token delete --id self
```

Create a personal access token for a user (admin only):

```
$ gitlab -v user-personal-access-token create --user-id 2 \
  --name personal-access-token --expires-at "2023-01-01" --scopes "api,read_repository"
↩
```

## 7.4.7 Resource access tokens

Create a project access token:

```
$ gitlab -v project-access-token create --project-id 2 \
  --name project-token --expires-at "2023-01-01" --scopes "api,read_repository"
```

List project access tokens:

```
$ gitlab -v project-access-token list --project-id 3
```

Revoke a project access token:

```
$ gitlab project-access-token delete --project-id 3 --id 1
```

Create a group access token:

```
$ gitlab -v group-access-token create --group-id 2 \
  --name group-token --expires-at "2022-01-01" --scopes "api,read_repository"
```

List group access tokens:

```
$ gitlab -v group-access-token list --group-id 3
```

Revoke a group access token:

```
$ gitlab group-access-token delete --group-id 3 --id 1
```

## 7.4.8 Packages

List packages for a project:

```
$ gitlab -v project-package list --project-id 3
```

List packages for a group:

```
$ gitlab -v group-package list --group-id 3
```

Get a specific project package by id:

```
$ gitlab -v project-package get --id 1 --project-id 3
```

Delete a specific project package by id:

```
$ gitlab -v project-package delete --id 1 --project-id 3
```

Upload a generic package to a project:

```
$ gitlab generic-package upload --project-id 1 --package-name hello-world \  
  --package-version v1.0.0 --file-name hello.tar.gz --path /path/to/hello.tar.gz
```

Download a project's generic package:

```
$ gitlab generic-package download --project-id 1 --package-name hello-world \  
  --package-version v1.0.0 --file-name hello.tar.gz > /path/to/hello.tar.gz
```

## 7.4.9 Issues

Get a list of issues for this project:

```
$ gitlab project-issue list --project-id 2
```

## 7.4.10 Snippets

Delete a snippet (id 3):

```
$ gitlab project-snippet delete --id 3 --project-id 2
```

Update a snippet:

```
$ gitlab project-snippet update --id 4 --project-id 2 \  
  --code "My New Code"
```

Create a snippet:

```
$ gitlab project-snippet create --project-id 2  
Impossible to create object (Missing attribute(s): title, file-name, code)  
$ # oops, let's add the attributes:  
$ gitlab project-snippet create --project-id 2 --title "the title" \  
  --file-name "the name" --code "the code"
```

### 7.4.11 Commits

Get a specific project commit by its SHA id:

```
$ gitlab project-commit get --project-id 2 --id a43290c
```

Get the signature (e.g. GPG or x509) of a signed commit:

```
$ gitlab project-commit signature --project-id 2 --id a43290c
```

Define the status of a commit (as would be done from a CI tool for example):

```
$ gitlab project-commit-status create --project-id 2 \
  --commit-id a43290c --state success --name ci/jenkins \
  --target-url http://server/build/123 \
  --description "Jenkins build succeeded"
```

Get the merge base for two or more branches, tags or commits:

```
gitlab project repository-merge-base --id 1 --refs bd1324e2f,main,v1.0.0
```

### 7.4.12 Artifacts

Download the artifacts zip archive of a job:

```
$ gitlab project-job artifacts --id 10 --project-id 1 > artifacts.zip
```

### 7.4.13 Runners

List owned runners:

```
$ gitlab runner list
```

List owned runners with a filter:

```
$ gitlab runner list --scope active
```

List all runners in the GitLab instance (specific and shared):

```
$ gitlab runner-all list
```

Get a runner's details:

```
$ gitlab -v runner get --id 123
```

### 7.4.14 Other

Use sudo to act as another user (admin only):

```
$ gitlab project create --name user_project1 --sudo username
```

List values are comma-separated:

```
$ gitlab issue list --labels foo,bar
```

## 7.5 API examples

### 7.5.1 Access requests

Users can request access to groups and projects.

When access is granted the user should be given a numerical access level. The following constants are provided to represent the access levels:

- `gitlab.const.AccessLevel.GUEST: 10`
- `gitlab.const.AccessLevel.REPORTER: 20`
- `gitlab.const.AccessLevel.DEVELOPER: 30`
- `gitlab.const.AccessLevel.MAINTAINER: 40`
- `gitlab.const.AccessLevel.OWNER: 50`

### References

- v4 API:
  - `gitlab.v4.objects.ProjectAccessRequest`
  - `gitlab.v4.objects.ProjectAccessRequestManager`
  - `gitlab.v4.objects.Project.accessrequests`
  - `gitlab.v4.objects.GroupAccessRequest`
  - `gitlab.v4.objects.GroupAccessRequestManager`
  - `gitlab.v4.objects.Group.accessrequests`
- GitLab API: [https://docs.gitlab.com/ce/api/access\\_requests.html](https://docs.gitlab.com/ce/api/access_requests.html)

### Examples

List access requests from projects and groups:

```
p_ars = project.accessrequests.list()
g_ars = group.accessrequests.list()
```

Create an access request:



```
p_ar = project.accessrequests.create()
g_ar = group.accessrequests.create()
```

Approve an access request:

```
ar.approve() # defaults to DEVELOPER level
ar.approve(access_level=gitlab.const.AccessLevel.MAINTAINER) # explicitly set access_
↪ level
```

Deny (delete) an access request:

```
project.accessrequests.delete(user_id)
group.accessrequests.delete(user_id)
# or
ar.delete()
```

## 7.5.2 Appearance

### Reference

- v4 API:
  - `gitlab.v4.objects.ApplicationAppearance`
  - `gitlab.v4.objects.ApplicationAppearanceManager`
  - `gitlab.Gitlab.appearance`
- GitLab API: <https://docs.gitlab.com/ce/api/appearance.html>

### Examples

Get the appearance:

```
appearance = gl.appearance.get()
```

Update the appearance:

```
appearance.title = "Test"
appearance.save()
```

## 7.5.3 Applications

### Reference

- v4 API:
  - `gitlab.v4.objects.Applications`
  - `gitlab.v4.objects.ApplicationManager`
  - `gitlab.Gitlab.applications`
- GitLab API: <https://docs.gitlab.com/ce/api/applications.html>

## Examples

List all OAuth applications:

```
applications = gl.applications.list()
```

Create an application:

```
gl.applications.create({'name': 'your_app', 'redirect_uri': 'http://application.url',  
↳ 'scopes': 'read_user openid profile email'})
```

Delete an applications:

```
gl.applications.delete(app_id)  
# or  
application.delete()
```

## 7.5.4 Award Emojis

### Reference

- v4 API:
  - *gitlab.v4.objects.ProjectIssueAwardEmoji*
  - *gitlab.v4.objects.ProjectIssueNoteAwardEmoji*
  - *gitlab.v4.objects.ProjectMergeRequestAwardEmoji*
  - *gitlab.v4.objects.ProjectMergeRequestNoteAwardEmoji*
  - *gitlab.v4.objects.ProjectSnippetAwardEmoji*
  - *gitlab.v4.objects.ProjectSnippetNoteAwardEmoji*
  - *gitlab.v4.objects.ProjectIssueAwardEmojiManager*
  - *gitlab.v4.objects.ProjectIssueNoteAwardEmojiManager*
  - *gitlab.v4.objects.ProjectMergeRequestAwardEmojiManager*
  - *gitlab.v4.objects.ProjectMergeRequestNoteAwardEmojiManager*
  - *gitlab.v4.objects.ProjectSnippetAwardEmojiManager*
  - *gitlab.v4.objects.ProjectSnippetNoteAwardEmojiManager*
- GitLab API: [https://docs.gitlab.com/ce/api/award\\_emoji.html](https://docs.gitlab.com/ce/api/award_emoji.html)

## Examples

List emojis for a resource:

```
emojis = obj.awardemojis.list()
```

Get a single emoji:

```
emoji = obj.awardemojis.get(emoji_id)
```

Add (create) an emoji:

```
emoji = obj.awardemojis.create({'name': 'tractor'})
```

Delete an emoji:

```
emoji.delete
# or
obj.awardemojis.delete(emoji_id)
```

## 7.5.5 Badges

Badges can be associated with groups and projects.

### Reference

- v4 API:
  - *gitlab.v4.objects.GroupBadge*
  - *gitlab.v4.objects.GroupBadgeManager*
  - *gitlab.v4.objects.Group.badges*
  - *gitlab.v4.objects.ProjectBadge*
  - *gitlab.v4.objects.ProjectBadgeManager*
  - *gitlab.v4.objects.Project.badges*
- GitLab API:
  - [https://docs.gitlab.com/ce/api/group\\_badges.html](https://docs.gitlab.com/ce/api/group_badges.html)
  - [https://docs.gitlab.com/ce/api/project\\_badges.html](https://docs.gitlab.com/ce/api/project_badges.html)

### Examples

List badges:

```
badges = group_or_project.badges.list()
```

Get a badge:

```
badge = group_or_project.badges.get(badge_id)
```

Create a badge:

```
badge = group_or_project.badges.create({'link_url': link, 'image_url': image_link})
```

Update a badge:

```
badge.image_url = new_image_url
badge.link_url = new_link_url
badge.save()
```

Delete a badge:

```
badge.delete()
```

Render a badge (preview the generate URLs):

```
output = group_or_project.badges.render(link, image_link)
print(output['rendered_link_url'])
print(output['rendered_image_url'])
```

## 7.5.6 Branches

### References

- v4 API:
  - *gitlab.v4.objects.ProjectBranch*
  - *gitlab.v4.objects.ProjectBranchManager*
  - *gitlab.v4.objects.Project.branches*
- GitLab API: <https://docs.gitlab.com/ce/api/branches.html>

### Examples

Get the list of branches for a repository:

```
branches = project.branches.list()
```

Get a single repository branch:

```
branch = project.branches.get('main')
```

Create a repository branch:

```
branch = project.branches.create({'branch': 'feature1',
                                  'ref': 'main'})
```

Delete a repository branch:

```
project.branches.delete('feature1')
# or
branch.delete()
```

Delete the merged branches for a project:

```
project.delete_merged_branches()
```

To manage protected branches, see *Protected branches*.

## 7.5.7 Migrations (Bulk Imports)

### References

- v4 API:
  - `gitlab.v4.objects.BulkImport`
  - `gitlab.v4.objects.BulkImportManager`
  - `gitlab.Gitlab.bulk_imports`
  - `gitlab.v4.objects.BulkImportAllEntity`
  - `gitlab.v4.objects.BulkImportAllEntityManager`
  - `gitlab.Gitlab.bulk_import_entities`
  - `gitlab.v4.objects.BulkImportEntity`
  - `gitlab.v4.objects.BulkImportEntityManager`
  - `gitlab.v4.objects.BulkImport.entities`
- GitLab API: [https://docs.gitlab.com/ee/api/bulk\\_imports.html](https://docs.gitlab.com/ee/api/bulk_imports.html)

### Examples

**Note:** Like the project/group imports and exports, this is an asynchronous operation and you will need to refresh the state from the server to get an accurate migration status. See *Examples* in the import/export section for more details and examples.

Start a bulk import/migration of a group and wait for completion:

```
# Create the migration
configuration = {
    "url": "https://gitlab.example.com",
    "access_token": private_token,
}
entity = {
    "source_full_path": "source_group",
    "source_type": "group_entity",
    "destination_slug": "imported-group",
    "destination_namespace": "imported-namespace",
}
migration = gl.bulk_imports.create(
    {
        "configuration": configuration,
        "entities": [entity],
    }
)

# Wait for the 'finished' status
while migration.status != "finished":
    time.sleep(1)
    migration.refresh()
```

List all migrations:

```
gl.bulk_imports.list()
```

List the entities of all migrations:

```
gl.bulk_import_entities.list()
```

Get a single migration by ID:

```
migration = gl.bulk_imports.get(123)
```

List the entities of a single migration:

```
entities = migration.entities.list()
```

Get a single entity of a migration by ID:

```
entity = migration.entities.get(123)
```

Refresh the state of a migration or entity from the server:

```
migration.refresh()
entity.refresh()

print(migration.status)
print(entity.status)
```

## 7.5.8 Broadcast messages

You can use broadcast messages to display information on all pages of the gitlab web UI. You must have administration permissions to manipulate broadcast messages.

### References

- v4 API:
  - *gitlab.v4.objects.BroadcastMessage*
  - *gitlab.v4.objects.BroadcastMessageManager*
  - *gitlab.Gitlab.broadcastmessages*
- GitLab API: [https://docs.gitlab.com/ce/api/broadcast\\_messages.html](https://docs.gitlab.com/ce/api/broadcast_messages.html)

## Examples

List the messages:

```
msgs = gl.broadcastmessages.list()
```

Get a single message:

```
msg = gl.broadcastmessages.get(msg_id)
```

Create a message:

```
msg = gl.broadcastmessages.create({'message': 'Important information'})
```

The date format for the `starts_at` and `ends_at` parameters is `YYYY-MM-ddThh:mm:ssZ`.

Update a message:

```
msg.font = '#444444'
msg.color = '#999999'
msg.save()
```

Delete a message:

```
gl.broadcastmessages.delete(msg_id)
# or
msg.delete()
```

## 7.5.9 CI Lint

### Reference

- v4 API:
  - `gitlab.v4.objects.CiLint`
  - `gitlab.v4.objects.CiLintManager`
  - `gitlab.Gitlab.ci_lint`
  - `gitlab.v4.objects.ProjectCiLint`
  - `gitlab.v4.objects.ProjectCiLintManager`
  - `gitlab.v4.objects.Project.ci_lint`
- GitLab API: <https://docs.gitlab.com/ee/api/lint.html>

## Examples

Lint a CI YAML configuration:

```
gitlab_ci_yaml = """.api_test:
  rules:
    - if: $CI_PIPELINE_SOURCE=="merge_request_event"
      changes:
        - src/api/*
deploy:
  extends:
    - .api_test
  rules:
    - when: manual
      allow_failure: true
  script:
    - echo "hello world"
"""
lint_result = gl.ci_lint.create({"content": gitlab_ci_yaml})

print(lint_result.status) # Print the status of the CI YAML
print(lint_result.merged_yaml) # Print the merged YAML file
```

Lint a project's CI configuration:

```
lint_result = project.ci_lint.get()
assert lint_result.valid is True # Test that the .gitlab-ci.yml is valid
print(lint_result.merged_yaml) # Print the merged YAML file
```

Lint a CI YAML configuration with a namespace:

```
lint_result = project.ci_lint.create({"content": gitlab_ci_yaml})
assert lint_result.valid is True # Test that the .gitlab-ci.yml is valid
print(lint_result.merged_yaml) # Print the merged YAML file
```

Validate a CI YAML configuration (raises `GitlabCiLintError` on failures):

```
# returns None
gl.ci_lint.validate({"content": gitlab_ci_yaml})

# raises GitlabCiLintError
gl.ci_lint.validate({"content": "invalid"})
```

Validate a CI YAML configuration with a namespace:

```
# returns None
project.ci_lint.validate({"content": gitlab_ci_yaml})

# raises GitlabCiLintError
project.ci_lint.validate({"content": "invalid"})
```



## 7.5.10 Commits

### Commits

#### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectCommit`
  - `gitlab.v4.objects.ProjectCommitManager`
  - `gitlab.v4.objects.Project.commits`

#### Examples

List the commits for a project:

```
commits = project.commits.list()
```

You can use the `ref_name`, `since` and `until` filters to limit the results:

```
commits = project.commits.list(ref_name='my_branch')
commits = project.commits.list(since='2016-01-01T00:00:00Z')
```

List all commits for a project (see [Pagination](#)) on all branches:

```
commits = project.commits.list(get_all=True, all=True)
```

Create a commit:

```
# See https://docs.gitlab.com/ce/api/commits.html#create-a-commit-with-multiple-files-
→and-actions
# for actions detail
data = {
    'branch': 'main',
    'commit_message': 'blah blah blah',
    'actions': [
        {
            'action': 'create',
            'file_path': 'README.rst',
            'content': open('path/to/file.rst').read(),
        },
        {
            # Binary files need to be base64 encoded
            'action': 'create',
            'file_path': 'logo.png',
            'content': base64.b64encode(open('logo.png', mode='r+b').read()).decode(),
            'encoding': 'base64',
        }
    ]
}

commit = project.commits.create(data)
```

Get a commit detail:

```
commit = project.commits.get('e3d5a71b')
```

Get the diff for a commit:

```
diff = commit.diff()
```

Cherry-pick a commit into another branch:

```
commit.cherry_pick(branch='target_branch')
```

Revert a commit on a given branch:

```
commit.revert(branch='target_branch')
```

Get the references the commit has been pushed to (branches and tags):

```
commit.refs() # all references
commit.refs('tag') # only tags
commit.refs('branch') # only branches
```

Get the signature of the commit (if the commit was signed, e.g. with GPG or x509):

```
commit.signature()
```

List the merge requests related to a commit:

```
commit.merge_requests()
```

## Commit comments

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectCommitComment`
  - `gitlab.v4.objects.ProjectCommitCommentManager`
  - `gitlab.v4.objects.ProjectCommit.comments`
- GitLab API: <https://docs.gitlab.com/ce/api/commits.html>

### Examples

Get the comments for a commit:

```
comments = commit.comments.list()
```

Add a comment on a commit:

```
# Global comment
commit = commit.comments.create({'note': 'This is a nice comment'})
# Comment on a line in a file (on the new version of the file)
commit = commit.comments.create({'note': 'This is another comment',
```

(continues on next page)

(continued from previous page)

```
'line': 12,  
'line_type': 'new',  
'path': 'README.rst'})
```

## Commit status

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectCommitStatus`
  - `gitlab.v4.objects.ProjectCommitStatusManager`
  - `gitlab.v4.objects.ProjectCommit.statuses`
- GitLab API: <https://docs.gitlab.com/ce/api/commits.html>

### Examples

List the statuses for a commit:

```
statuses = commit.statuses.list()
```

Change the status of a commit:

```
commit.statuses.create({'state': 'success'})
```

## 7.5.11 Deploy keys

### Deploy keys

#### Reference

- v4 API:
  - `gitlab.v4.objects.DeployKey`
  - `gitlab.v4.objects.DeployKeyManager`
  - `gitlab.Gitlab.deploykeys`
- GitLab API: [https://docs.gitlab.com/ce/api/deploy\\_keys.html](https://docs.gitlab.com/ce/api/deploy_keys.html)

## Examples

List the deploy keys:

```
keys = gl.deploykeys.list()
```

## Deploy keys for projects

Deploy keys can be managed on a per-project basis.

## Reference

- v4 API:
  - *gitlab.v4.objects.ProjectKey*
  - *gitlab.v4.objects.ProjectKeyManager*
  - *gitlab.v4.objects.Project.keys*
- GitLab API: [https://docs.gitlab.com/ce/api/deploy\\_keys.html](https://docs.gitlab.com/ce/api/deploy_keys.html)

## Examples

List keys for a project:

```
keys = project.keys.list()
```

Get a single deploy key:

```
key = project.keys.get(key_id)
```

Create a deploy key for a project:

```
key = project.keys.create({'title': 'jenkins key',  
                           'key': open('/home/me/.ssh/id_rsa.pub').read()})
```

Delete a deploy key for a project:

```
key = project.keys.list(key_id)  
# or  
key.delete()
```

Enable a deploy key for a project:

```
project.keys.enable(key_id)
```

Disable a deploy key for a project:

```
project.keys.delete(key_id)
```

## 7.5.12 Deploy tokens

Deploy tokens allow read-only access to your repository and registry images without having a user and a password.

### Deploy tokens

This endpoint requires admin access.

### Reference

- v4 API:
  - *gitlab.v4.objects.DeployToken*
  - *gitlab.v4.objects.DeployTokenManager*
  - *gitlab.Gitlab.deploytokens*
- GitLab API: [https://docs.gitlab.com/ce/api/deploy\\_tokens.html](https://docs.gitlab.com/ce/api/deploy_tokens.html)

### Examples

Use the `list()` method to list all deploy tokens across the GitLab instance.

```
# List deploy tokens
deploy_tokens = gl.deploytokens.list()
```

### Project deploy tokens

This endpoint requires project maintainer access or higher.

### Reference

- v4 API:
  - *gitlab.v4.objects.ProjectDeployToken*
  - *gitlab.v4.objects.ProjectDeployTokenManager*
  - *gitlab.v4.objects.Project.deploytokens*
- GitLab API: [https://docs.gitlab.com/ce/api/deploy\\_tokens.html#project-deploy-tokens](https://docs.gitlab.com/ce/api/deploy_tokens.html#project-deploy-tokens)

## Examples

List the deploy tokens for a project:

```
deploy_tokens = project.deploytokens.list()
```

Get a deploy token for a project by id:

```
deploy_token = project.deploytokens.get(deploy_token_id)
```

Create a new deploy token to access registry images of a project:

In addition to required parameters `name` and `scopes`, this method accepts the following parameters:

- `expires_at` Expiration date of the deploy token. Does not expire if no value is provided.
- `username` Username for deploy token. Default is `gitlab+deploy-token-{n}`

```
deploy_token = project.deploytokens.create({'name': 'token1', 'scopes': ['read_registry',  
→], 'username': '', 'expires_at': ''})  
# show its id  
print(deploy_token.id)  
# show the token value. Make sure you save it, you won't be able to access it again.  
print(deploy_token.token)
```

**Warning:** With GitLab 12.9, even though `username` and `expires_at` are not required, they always have to be passed to the API. You can set them to empty strings, see: <https://gitlab.com/gitlab-org/gitlab/-/issues/211878>. Also, the `username`'s value is ignored by the API and will be overridden with `gitlab+deploy-token-{n}`, see: <https://gitlab.com/gitlab-org/gitlab/-/issues/211963> These issues were fixed in GitLab 12.10.

Remove a deploy token from the project:

```
deploy_token.delete()  
# or  
project.deploytokens.delete(deploy_token.id)
```

## Group deploy tokens

### Reference

- v4 API:
  - `gitlab.v4.objects.GroupDeployToken`
  - `gitlab.v4.objects.GroupDeployTokenManager`
  - `gitlab.v4.objects.Group.deploytokens`
- GitLab API: [https://docs.gitlab.com/ce/api/deploy\\_tokens.html#group-deploy-tokens](https://docs.gitlab.com/ce/api/deploy_tokens.html#group-deploy-tokens)

## Examples

List the deploy tokens for a group:

```
deploy_tokens = group.deploytokens.list()
```

Get a deploy token for a group by id:

```
deploy_token = group.deploytokens.get(deploy_token_id)
```

Create a new deploy token to access all repositories of all projects in a group:

In addition to required parameters `name` and `scopes`, this method accepts the following parameters:

- `expires_at` Expiration date of the deploy token. Does not expire if no value is provided.
- `username` Username for deploy token. Default is `gitlab+deploy-token-{n}`

```
deploy_token = group.deploytokens.create({'name': 'token1', 'scopes': ['read_repository',
↪], 'username': '', 'expires_at': ''})
# show its id
print(deploy_token.id)
```

**Warning:** With GitLab 12.9, even though `username` and `expires_at` are not required, they always have to be passed to the API. You can set them to empty strings, see: <https://gitlab.com/gitlab-org/gitlab/-/issues/211878>. Also, the `username`'s value is ignored by the API and will be overridden with `gitlab+deploy-token-{n}`, see: <https://gitlab.com/gitlab-org/gitlab/-/issues/211963> These issues were fixed in GitLab 12.10.

Remove a deploy token from the group:

```
deploy_token.delete()
# or
group.deploytokens.delete(deploy_token.id)
```

## 7.5.13 Deployments

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectDeployment`
  - `gitlab.v4.objects.ProjectDeploymentManager`
  - `gitlab.v4.objects.Project.deployments`
- GitLab API: <https://docs.gitlab.com/ce/api/deployments.html>

## Examples

List deployments for a project:

```
deployments = project.deployments.list()
```

Get a single deployment:

```
deployment = project.deployments.get(deployment_id)
```

Create a new deployment:

```
deployment = project.deployments.create({
    "environment": "Test",
    "sha": "1agf4gs",
    "ref": "main",
    "tag": False,
    "status": "created",
})
```

Update a deployment:

```
deployment = project.deployments.get(42)
deployment.status = "failed"
deployment.save()
```

Approve a deployment:

```
deployment = project.deployments.get(42)
# `status` must be either "approved" or "rejected".
deployment.approval(status="approved")
```

Reject a deployment:

```
deployment = project.deployments.get(42)
# Using the optional `comment` and `represented_as` arguments
deployment.approval(status="rejected", comment="Fails CI", represented_as="security")
```

## Merge requests associated with a deployment

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectDeploymentMergeRequest`
  - `gitlab.v4.objects.ProjectDeploymentMergeRequestManager`
  - `gitlab.v4.objects.ProjectDeployment.mergerequests`
- GitLab API: <https://docs.gitlab.com/ee/api/deployments.html#list-of-merge-requests-associated-with-a-deployment>



## Examples

List the merge requests associated with a deployment:

```
deployment = project.deployments.get(42, lazy=True)
mrs = deployment.mergerequests.list()
```

## 7.5.14 Discussions

Discussions organize the notes in threads. See the *Notes* chapter for more information about notes.

Discussions are available for project issues, merge requests, snippets and commits.

## Reference

- v4 API:

Issues:

- *gitlab.v4.objects.ProjectIssueDiscussion*
- *gitlab.v4.objects.ProjectIssueDiscussionManager*
- *gitlab.v4.objects.ProjectIssueDiscussionNote*
- *gitlab.v4.objects.ProjectIssueDiscussionNoteManager*
- *gitlab.v4.objects.ProjectIssue.notes*

MergeRequests:

- *gitlab.v4.objects.ProjectMergeRequestDiscussion*
- *gitlab.v4.objects.ProjectMergeRequestDiscussionManager*
- *gitlab.v4.objects.ProjectMergeRequestDiscussionNote*
- *gitlab.v4.objects.ProjectMergeRequestDiscussionNoteManager*
- *gitlab.v4.objects.ProjectMergeRequest.notes*

Snippets:

- *gitlab.v4.objects.ProjectSnippetDiscussion*
- *gitlab.v4.objects.ProjectSnippetDiscussionManager*
- *gitlab.v4.objects.ProjectSnippetDiscussionNote*
- *gitlab.v4.objects.ProjectSnippetDiscussionNoteManager*
- *gitlab.v4.objects.ProjectSnippet.notes*

- GitLab API: <https://docs.gitlab.com/ce/api/discussions.html>

## Examples

List the discussions for a resource (issue, merge request, snippet or commit):

```
discussions = resource.discussions.list()
```

Get a single discussion:

```
discussion = resource.discussions.get(discussion_id)
```

You can access the individual notes in the discussion through the `notes` attribute. It holds a list of notes in chronological order:

```
# ``resource.notes`` is a DiscussionNoteManager, so we need to get the
# object notes using ``attributes``
for note in discussion.attributes['notes']:
    print(note['body'])
```

---

**Note:** The notes are dicts, not objects.

---

You can add notes to existing discussions:

```
new_note = discussion.notes.create({'body': 'Episode IV: A new note'})
```

You can get and update a single note using the `*DiscussionNote` resources:

```
discussion = resource.discussions.get(discussion_id)
# Get the latest note's id
note_id = discussion.attributes['notes'][-1]['id']
last_note = discussion.notes.get(note_id)
last_note.body = 'Updated comment'
last_note.save()
```

Create a new discussion:

```
discussion = resource.discussions.create({'body': 'First comment of discussion'})
```

You can comment on merge requests and commit diffs. Provide the `position` dict to define where the comment should appear in the diff:

```
mr_diff = mr.diffs.get(diff_id)
mr.dussions.create({'body': 'Note content',
                    'position': {
                        'base_sha': mr_diff.base_commit_sha,
                        'start_sha': mr_diff.start_commit_sha,
                        'head_sha': mr_diff.head_commit_sha,
                        'position_type': 'text',
                        'new_line': 1,
                        'old_path': 'README.rst',
                        'new_path': 'README.rst'
                    }})
```

Resolve / unresolve a merge request discussion:

```
mr_d = mr.discussions.get(d_id)
mr_d.resolved = True # True to resolve, False to unresolve
mr_d.save()
```

Delete a comment:

```
discussions.notes.delete(note_id)
# or
note.delete()
```

### 7.5.15 Draft Notes

Draft notes are pending, unpublished comments on merge requests. They can be either start a discussion, or be associated with an existing discussion as a reply. They are viewable only by the author until they are published.

#### Reference

- v4 API:
  - *gitlab.v4.objects.ProjectMergeRequestDraftNote*
  - *gitlab.v4.objects.ProjectMergeRequestDraftNoteManager*
  - *gitlab.v4.objects.ProjectMergeRequest.draft\_notes*
- GitLab API: [https://docs.gitlab.com/ee/api/draft\\_notes.html](https://docs.gitlab.com/ee/api/draft_notes.html)

#### Examples

List all draft notes for a merge request:

```
draft_notes = merge_request.draft_notes.list()
```

Get a draft note for a merge request by ID:

```
draft_note = merge_request.draft_notes.get(note_id)
```

**Warning:** When creating or updating draft notes, you can provide a complex nested `position` argument as a dictionary. Please consult the upstream API documentation linked above for the exact up-to-date attributes.

Create a draft note for a merge request:

```
draft_note = merge_request.draft_notes.create({'note': 'note content'})
```

Update an existing draft note:

```
draft_note.note = 'updated note content'
draft_note.save()
```

Delete an existing draft note:

```
draft_note.delete()
```

Publish an existing draft note:

```
draft_note.publish()
```

Publish all existing draft notes for a merge request in bulk:

```
merge_request.draft_notes.bulk_publish()
```

## 7.5.16 Environments

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectEnvironment`
  - `gitlab.v4.objects.ProjectEnvironmentManager`
  - `gitlab.v4.objects.Project.environments`
- GitLab API: <https://docs.gitlab.com/ce/api/environments.html>

### Examples

List environments for a project:

```
environments = project.environments.list()
```

Create an environment for a project:

```
environment = project.environments.create({'name': 'production'})
```

Retrieve a specific environment for a project:

```
environment = project.environments.get(112)
```

Update an environment for a project:

```
environment.external_url = 'http://foo.bar.com'  
environment.save()
```

Delete an environment for a project:

```
environment = project.environments.delete(environment_id)  
# or  
environment.delete()
```

Stop an environment:

```
environment.stop()
```

To manage protected environments, see *Protected environments*.

## 7.5.17 Events

### Events

#### Reference

- v4 API:
  - `gitlab.v4.objects.Event`
  - `gitlab.v4.objects.EventManager`
  - `gitlab.Gitlab.events`
  - `gitlab.v4.objects.ProjectEvent`
  - `gitlab.v4.objects.ProjectEventManager`
  - `gitlab.v4.objects.Project.events`
  - `gitlab.v4.objects.UserEvent`
  - `gitlab.v4.objects.UserEventManager`
  - `gitlab.v4.objects.User.events`
- GitLab API: <https://docs.gitlab.com/ce/api/events.html>

### Examples

You can list events for an entire Gitlab instance (admin), users and projects. You can filter you events you want to retrieve using the `action` and `target_type` attributes. The possible values for these attributes are available on [the gitlab documentation](#).

List all the events (paginated):

```
events = gl.events.list()
```

List the issue events on a project:

```
events = project.events.list(target_type='issue')
```

List the user events:

```
events = project.events.list()
```

### Resource state events

#### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectIssueResourceStateEvent`
  - `gitlab.v4.objects.ProjectIssueResourceStateEventManager`
  - `gitlab.v4.objects.ProjectIssue.resourcestateevents`
  - `gitlab.v4.objects.ProjectMergeRequestResourceStateEvent`

- `gitlab.v4.objects.ProjectMergeRequestResourceStateEventManager`
- `gitlab.v4.objects.ProjectMergeRequest.resourcestateevents`
- GitLab API: [https://docs.gitlab.com/ee/api/resource\\_state\\_events.html](https://docs.gitlab.com/ee/api/resource_state_events.html)

## Examples

You can list and get specific resource state events (via their id) for project issues and project merge requests.

List the state events of a project issue (paginated):

```
state_events = issue.resourcestateevents.list()
```

Get a specific state event of a project issue by its id:

```
state_event = issue.resourcestateevents.get(1)
```

List the state events of a project merge request (paginated):

```
state_events = mr.resourcestateevents.list()
```

Get a specific state event of a project merge request by its id:

```
state_event = mr.resourcestateevents.get(1)
```

## 7.5.18 Epics

### Epics

#### Reference

- v4 API:
  - `gitlab.v4.objects.GroupEpic`
  - `gitlab.v4.objects.GroupEpicManager`
  - `gitlab.Gitlab.Group.epics`
- GitLab API: <https://docs.gitlab.com/ee/api/epics.html> (EE feature)

## Examples

List the epics for a group:

```
epics = groups.epics.list()
```

Get a single epic for a group:

```
epic = group.epics.get(epic_iid)
```

Create an epic for a group:

```
epic = group.epics.create({'title': 'My Epic'})
```

Edit an epic:

```
epic.title = 'New title'
epic.labels = ['label1', 'label2']
epic.save()
```

Delete an epic:

```
epic.delete()
```

## Epics issues

### Reference

- v4 API:
  - `gitlab.v4.objects.GroupEpicIssue`
  - `gitlab.v4.objects.GroupEpicIssueManager`
  - `gitlab.Gitlab.GroupEpic.issues`
- GitLab API: [https://docs.gitlab.com/ee/api/epic\\_issues.html](https://docs.gitlab.com/ee/api/epic_issues.html) (EE feature)

### Examples

List the issues associated with an issue:

```
ei = epic.issues.list()
```

Associate an issue with an epic:

```
# use the issue id, not its iid
ei = epic.issues.create({'issue_id': 4})
```

Move an issue in the list:

```
ei.move_before_id = epic_issue_id_1
# or
ei.move_after_id = epic_issue_id_2
ei.save()
```

Delete an issue association:

```
ei.delete()
```

## 7.5.19 Features flags

### Reference

- v4 API:
  - `gitlab.v4.objects.Feature`
  - `gitlab.v4.objects.FeatureManager`
  - `gitlab.Gitlab.features`
- GitLab API: <https://docs.gitlab.com/ce/api/features.html>

### Examples

List features:

```
features = gl.features.list()
```

Create or set a feature:

```
feature = gl.features.set(feature_name, True)
feature = gl.features.set(feature_name, 30)
feature = gl.features.set(feature_name, True, user=filipowm)
feature = gl.features.set(feature_name, 40, group=mygroup)
```

Delete a feature:

```
feature.delete()
```

## 7.5.20 Geo nodes

### Reference

- v4 API:
  - `gitlab.v4.objects.GeoNode`
  - `gitlab.v4.objects.GeoNodeManager`
  - `gitlab.Gitlab.geonodes`
- GitLab API: [https://docs.gitlab.com/ee/api/geo\\_nodes.html](https://docs.gitlab.com/ee/api/geo_nodes.html) (EE feature)

### Examples

List the geo nodes:

```
nodes = gl.geonodes.list()
```

Get the status of all the nodes:

```
status = gl.geonodes.status()
```

Get a specific node and its status:



```
node = gl.geonodes.get(node_id)
node.status()
```

Edit a node configuration:

```
node.url = 'https://secondary.mygitlab.domain'
node.save()
```

Delete a node:

```
node.delete()
```

List the sync failure on the current node:

```
failures = gl.geonodes.current_failures()
```

## 7.5.21 Groups

### Groups

#### Reference

- v4 API:
  - *gitlab.v4.objects.Group*
  - *gitlab.v4.objects.GroupManager*
  - *gitlab.Gitlab.groups*
- GitLab API: <https://docs.gitlab.com/ce/api/groups.html>

#### Examples

List the groups:

```
groups = gl.groups.list()
```

Get a group's detail:

```
group = gl.groups.get(group_id)
```

List a group's projects:

```
projects = group.projects.list()
```

List a group's shared projects:

```
projects = group.shared_projects.list()
```

---

**Note:** GroupProject and SharedProject objects returned by these two API calls are very limited, and do not provide all the features of Project objects. If you need to manipulate projects, create a new Project object:

```
first_group_project = group.projects.list()[0]
manageable_project = gl.projects.get(first_group_project.id, lazy=True)
```

You can filter and sort the result using the following parameters:

- `archived`: limit by archived status
- `visibility`: limit by visibility. Allowed values are `public`, `internal` and `private`
- `search`: limit to groups matching the given value
- `order_by`: sort by criteria. Allowed values are `id`, `name`, `path`, `created_at`, `updated_at` and `last_activity_at`
- `sort`: sort order: `asc` or `desc`
- `ci_enabled_first`: return CI enabled groups first
- `include_subgroups`: include projects in subgroups

Create a group:

```
group = gl.groups.create({'name': 'group1', 'path': 'group1'})
```

**Warning:** On GitLab.com, creating top-level groups is currently [not permitted using the API](#). You can only use the API to create subgroups.

Create a subgroup under an existing group:

```
subgroup = gl.groups.create({'name': 'subgroup1', 'path': 'subgroup1', 'parent_id':  
↪parent_group_id})
```

Update a group:

```
group.description = 'My awesome group'  
group.save()
```

Set the avatar image for a group:

```
# the avatar image can be passed as data (content of the file) or as a file  
# object opened in binary mode  
group.avatar = open('path/to/file.png', 'rb')  
group.save()
```

Remove a group:

```
gl.groups.delete(group_id)  
# or  
group.delete()
```

Restore a Group marked for deletion (Premium only)::

```
group.restore()
```

Share/unshare the group with a group:

```
group.share(group2.id, gitlab.const.AccessLevel.DEVELOPER)
group.unshare(group2.id)
```

## Import / Export

You can export groups from gitlab, and re-import them to create new groups.

## Reference

- v4 API:
  - `gitlab.v4.objects.GroupExport`
  - `gitlab.v4.objects.GroupExportManager`
  - `gitlab.v4.objects.Group.exports`
  - `gitlab.v4.objects.GroupImport`
  - `gitlab.v4.objects.GroupImportManager`
  - `gitlab.v4.objects.Group.imports`
  - `gitlab.v4.objects.GroupManager.import_group`
- GitLab API: [https://docs.gitlab.com/ce/api/group\\_import\\_export.html](https://docs.gitlab.com/ce/api/group_import_export.html)

## Examples

A group export is an asynchronous operation. To retrieve the archive generated by GitLab you need to:

1. Create an export using the API
2. Wait for the export to be done
3. Download the result

**Warning:** Unlike the Project Export API, GitLab does not provide an `export_status` for Group Exports. It is up to the user to ensure the export is finished.

However, Group Exports only contain metadata, so they are much faster than Project Exports.

```
# Create the export
group = gl.groups.get(my_group)
export = group.exports.create()

# Wait for the export to finish
time.sleep(3)

# Download the result
with open('/tmp/export.tgz', 'wb') as f:
    export.download(streamed=True, action=f.write)
```

Import the group:

```
with open('/tmp/export.tgz', 'rb') as f:
    gl.groups.import_group(f, path='imported-group', name="Imported Group")
```

## Subgroups

### Reference

- v4 API:
  - `gitlab.v4.objects.GroupSubgroup`
  - `gitlab.v4.objects.GroupSubgroupManager`
  - `gitlab.v4.objects.Group.subgroups`

### Examples

List the subgroups for a group:

```
subgroups = group.subgroups.list()
```

---

**Note:** The `GroupSubgroup` objects don't expose the same API as the `Group` objects. If you need to manipulate a subgroup as a group, create a new `Group` object:

```
real_group = gl.groups.get(subgroup_id, lazy=True)
real_group.issues.list()
```

## Descendant Groups

### Reference

- v4 API:
  - `gitlab.v4.objects.GroupDescendantGroup`
  - `gitlab.v4.objects.GroupDescendantGroupManager`
  - `gitlab.v4.objects.Group.descendant_groups`

### Examples

List the descendant groups of a group:

```
descendant_groups = group.descendant_groups.list()
```

---

**Note:** Like the `GroupSubgroup` objects described above, `GroupDescendantGroup` objects do not expose the same API as the `Group` objects. Create a new `Group` object instead if needed, as shown in the subgroup example.

---

## Group custom attributes

### Reference

- v4 API:
  - `gitlab.v4.objects.GroupCustomAttribute`
  - `gitlab.v4.objects.GroupCustomAttributeManager`
  - `gitlab.v4.objects.Group.customattributes`
- GitLab API: [https://docs.gitlab.com/ce/api/custom\\_attributes.html](https://docs.gitlab.com/ce/api/custom_attributes.html)

### Examples

List custom attributes for a group:

```
attrs = group.customattributes.list()
```

Get a custom attribute for a group:

```
attr = group.customattributes.get(attr_key)
```

Set (create or update) a custom attribute for a group:

```
attr = group.customattributes.set(attr_key, attr_value)
```

Delete a custom attribute for a group:

```
attr.delete()
# or
group.customattributes.delete(attr_key)
```

Search groups by custom attribute:

```
group.customattributes.set('role': 'admin')
gl.groups.list(custom_attributes={'role': 'admin'})
```

## Group members

The following constants define the supported access levels:

- `gitlab.const.AccessLevel.GUEST = 10`
- `gitlab.const.AccessLevel.REPORTER = 20`
- `gitlab.const.AccessLevel.DEVELOPER = 30`
- `gitlab.const.AccessLevel.MAINTAINER = 40`
- `gitlab.const.AccessLevel.OWNER = 50`

## Reference

- v4 API:
  - `gitlab.v4.objects.GroupMember`
  - `gitlab.v4.objects.GroupMemberManager`
  - `gitlab.v4.objects.GroupMemberAllManager`
  - `gitlab.v4.objects.GroupBillableMember`
  - `gitlab.v4.objects.GroupBillableMemberManager`
  - `gitlab.v4.objects.Group.members`
  - `gitlab.v4.objects.Group.members_all`
  - `gitlab.v4.objects.Group.billable_members`
- GitLab API: <https://docs.gitlab.com/ce/api/members.html>

Billable group members are only available in GitLab EE.

## Examples

List only direct group members:

```
members = group.members.list()
```

List the group members recursively (including inherited members through ancestor groups):

```
members = group.members_all.list(get_all=True)
```

Get only direct group member:

```
members = group.members.get(member_id)
```

Get a member of a group, including members inherited through ancestor groups:

```
members = group.members_all.get(member_id)
```

Add a member to the group:

```
member = group.members.create({'user_id': user_id,  
                               'access_level': gitlab.const.AccessLevel.GUEST})
```

Update a member (change the access level):

```
member.access_level = gitlab.const.AccessLevel.DEVELOPER  
member.save()
```

Remove a member from the group:

```
group.members.delete(member_id)  
# or  
member.delete()
```

List billable members of a group (top-level groups only):

```
billable_members = group.billable_members.list()
```

Remove a billable member from the group:

```
group.billable_members.delete(member_id)
# or
billable_member.delete()
```

List memberships of a billable member:

```
billable_member.memberships.list()
```

## LDAP group links

Add an LDAP group link to an existing GitLab group:

```
ldap_link = group.ldap_group_links.create({
    'provider': 'ldapmain',
    'group_access': gitlab.const.AccessLevel.DEVELOPER,
    'cn': 'ldap_group_cn'
})
```

List a group's LDAP group links:

```
group.ldap_group_links.list()
```

Remove a link:

```
ldap_link.delete()
# or by explicitly providing the CN or filter
group.ldap_group_links.delete(provider='ldapmain', cn='ldap_group_cn')
group.ldap_group_links.delete(provider='ldapmain', filter='(cn=Common Name)')
```

Sync the LDAP groups:

```
group.ldap_sync()
```

You can use the `ldapgroups` manager to list available LDAP groups:

```
# listing (supports pagination)
ldap_groups = gl.ldapgroups.list()

# filter using a group name
ldap_groups = gl.ldapgroups.list(search='foo')

# list the groups for a specific LDAP provider
ldap_groups = gl.ldapgroups.list(search='foo', provider='ldapmain')
```

## SAML group links

Add a SAML group link to an existing GitLab group:

```
saml_link = group.saml_group_links.create({
    "saml_group_name": "<your_saml_group_name>",
    "access_level": <chosen_access_level>
})
```

List a group's SAML group links:

```
group.saml_group_links.list()
```

Get a SAML group link:

```
group.saml_group_links.get("<your_saml_group_name>")
```

Remove a link:

```
saml_link.delete()
```

## Groups hooks

### Reference

- v4 API:
  - *gitlab.v4.objects.GroupHook*
  - *gitlab.v4.objects.GroupHookManager*
  - *gitlab.v4.objects.Group.hooks*
- GitLab API: <https://docs.gitlab.com/ce/api/groups.html#hooks>

### Examples

List the group hooks:

```
hooks = group.hooks.list()
```

Get a group hook:

```
hook = group.hooks.get(hook_id)
```

Create a group hook:

```
hook = group.hooks.create({'url': 'http://my/action/url', 'push_events': 1})
```

Update a group hook:

```
hook.push_events = 0
hook.save()
```

Delete a group hook:



```
group.hooks.delete(hook_id)
# or
hook.delete()
```

## Group push rules

### Reference

- v4 API:
  - `gitlab.v4.objects.GroupPushRules`
  - `gitlab.v4.objects.GroupPushRulesManager`
  - `gitlab.v4.objects.Group.pushrules`
- GitLab API: <https://docs.gitlab.com/ee/api/groups.html#push-rules>

### Examples

Create group push rules (at least one rule is necessary):

```
group.pushrules.create({'deny_delete_tag': True})
```

Get group push rules:

```
pr = group.pushrules.get()
```

Edit group push rules:

```
pr.branch_name_regex = '^((master|develop|support-\d+|release-\d+\.|hotfix-\d+|feature-\d+)+)$'
pr.save()
```

Delete group push rules:

```
pr.delete()
```

## 7.5.22 Group Access Tokens

Get a list of group access tokens

### References

- v4 API:
  - `gitlab.v4.objects.GroupAccessToken`
  - `gitlab.v4.objects.GroupAccessTokenManager`
  - `gitlab.Gitlab.group_access_tokens`
- GitLab API: [https://docs.gitlab.com/ee/api/group\\_access\\_tokens.html](https://docs.gitlab.com/ee/api/group_access_tokens.html)

## Examples

List group access tokens:

```
access_tokens = gl.groups.get(1, lazy=True).access_tokens.list()
print(access_tokens[0].name)
```

Get a group access token by id:

```
token = group.access_tokens.get(123)
print(token.name)
```

Create group access token:

```
access_token = gl.groups.get(1).access_tokens.create({"name": "test", "scopes": ["api"],
↪ "expires_at": "2023-06-06"})
```

Revoke a group access token:

```
gl.groups.get(1).access_tokens.delete(42)
# or
access_token.delete()
```

Rotate a group access token and retrieve its new value:

```
token = group.access_tokens.get(42, lazy=True)
token.rotate()
print(token.token)
# or directly using a token ID
new_token = group.access_tokens.rotate(42)
print(new_token.token)
```

## 7.5.23 Invitations

Invitations let you invite or add users to a group or project.

### Reference

- v4 API:
  - `gitlab.v4.objects.GroupInvitation`
  - `gitlab.v4.objects.GroupInvitationManager`
  - `gitlab.v4.objects.Group.invitations`
  - `gitlab.v4.objects.ProjectInvitation`
  - `gitlab.v4.objects.ProjectInvitationManager`
  - `gitlab.v4.objects.Project.invitations`
- GitLab API: <https://docs.gitlab.com/ce/api/invitations.html>

## Examples

**Danger:** Creating an invitation with `create()` returns a status response, rather than invitation details, because it allows sending multiple invitations at the same time.

Thus when using several emails, you do not create a real invitation object you can manipulate, because python-gitlab cannot know which email to track as the ID.

In that case, use a **lazy** `get()` method shown below using a specific email address to create an invitation object you can manipulate.

Create an invitation:

```
invitation = group_or_project.invitations.create(
    {
        "email": "email@example.com",
        "access_level": gitlab.const.AccessLevel.DEVELOPER,
    }
)
```

List invitations for a group or project:

```
invitations = group_or_project.invitations.list()
```

**Warning:** As mentioned above, GitLab does not provide a real GET endpoint for a single invitation. We can create a lazy object to later manipulate it.

Update an invitation:

```
invitation = group_or_project.invitations.get("email@example.com", lazy=True)
invitation.access_level = gitlab.const.AccessLevel.DEVELOPER
invitation.save()

# or
group_or_project.invitations.update(
    "email@example.com",
    {"access_level": gitlab.const.AccessLevel.DEVELOPER}
)
```

Delete an invitation:

```
invitation = group_or_project.invitations.get("email@example.com", lazy=True)
invitation.delete()

# or
group_or_project.invitations.delete("email@example.com")
```

## 7.5.24 Issues

### Reported issues

#### Reference

- v4 API:
  - `gitlab.v4.objects.Issue`
  - `gitlab.v4.objects.IssueManager`
  - `gitlab.Gitlab.issues`
- GitLab API: <https://docs.gitlab.com/ce/api/issues.html>

#### Examples

List the issues:

```
issues = gl.issues.list()
```

Use the `state` and `label` parameters to filter the results. Use the `order_by` and `sort` attributes to sort the results:

```
open_issues = gl.issues.list(state='opened')
closed_issues = gl.issues.list(state='closed')
tagged_issues = gl.issues.list(labels=['foo', 'bar'])
```

---

**Note:** It is not possible to edit or delete Issue objects. You need to create a `ProjectIssue` object to perform changes:

```
issue = gl.issues.list()[0]
project = gl.projects.get(issue.project_id, lazy=True)
editable_issue = project.issues.get(issue.iid, lazy=True)
editable_issue.title = updated_title
editable_issue.save()
```

### Group issues

#### Reference

- v4 API:
  - `gitlab.v4.objects.GroupIssue`
  - `gitlab.v4.objects.GroupIssueManager`
  - `gitlab.v4.objects.Group.issues`
- GitLab API: <https://docs.gitlab.com/ce/api/issues.html>

## Examples

List the group issues:

```
issues = group.issues.list()
# Filter using the state, labels and milestone parameters
issues = group.issues.list(milestone='1.0', state='opened')
# Order using the order_by and sort parameters
issues = group.issues.list(order_by='created_at', sort='desc')
```

**Note:** It is not possible to edit or delete GroupIssue objects. You need to create a ProjectIssue object to perform changes:

```
issue = group.issues.list()[0]
project = gl.projects.get(issue.project_id, lazy=True)
editable_issue = project.issues.get(issue.iid, lazy=True)
editable_issue.title = updated_title
editable_issue.save()
```

## Project issues

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectIssue`
  - `gitlab.v4.objects.ProjectIssueManager`
  - `gitlab.v4.objects.Project.issues`
- GitLab API: <https://docs.gitlab.com/ce/api/issues.html>

## Examples

List the project issues:

```
issues = project.issues.list()
# Filter using the state, labels and milestone parameters
issues = project.issues.list(milestone='1.0', state='opened')
# Order using the order_by and sort parameters
issues = project.issues.list(order_by='created_at', sort='desc')
```

Get a project issue:

```
issue = project.issues.get(issue_iid)
```

Create a new issue:

```
issue = project.issues.create({'title': 'I have a bug',
                              'description': 'Something useful here.'})
```

Update an issue:

```
issue.labels = ['foo', 'bar']
issue.save()
```

Close / reopen an issue:

```
# close an issue
issue.state_event = 'close'
issue.save()
# reopen it
issue.state_event = 'reopen'
issue.save()
```

Delete an issue (admin or project owner only):

```
project.issues.delete(issue_id)
# or
issue.delete()
```

Assign the issues:

```
issue = gl.issues.list()[0]
issue.assignee_ids = [25, 10, 31, 12]
issue.save()
```

---

**Note:** The Gitlab API explicitly references that the *assignee\_id* field is deprecated, so using a list of user IDs for *assignee\_ids* is how to assign an issue to a user(s).

---

Subscribe / unsubscribe from an issue:

```
issue.subscribe()
issue.unsubscribe()
```

Move an issue to another project:

```
issue.move(other_project_id)
```

Reorder an issue on a board:

```
issue.reorder(move_after_id=2, move_before_id=3)
```

Make an issue as todo:

```
issue.todo()
```

Get time tracking stats:

```
issue.time_stats()
```

On recent versions of Gitlab the time stats are also returned as an issue object attribute:

```
issue = project.issue.get(iid)
print(issue.attributes['time_stats'])
```

Set a time estimate for an issue:

```
issue.time_estimate('3h30m')
```

Reset a time estimate for an issue:

```
issue.reset_time_estimate()
```

Add spent time for an issue:

```
issue.add_spent_time('3h30m')
```

Reset spent time for an issue:

```
issue.reset_spent_time()
```

Get user agent detail for the issue (admin only):

```
detail = issue.user_agent_detail()
```

Get the list of merge requests that will close an issue when merged:

```
mrs = issue.closed_by()
```

Get the merge requests related to an issue:

```
mrs = issue.related_merge_requests()
```

Get the list of participants:

```
users = issue.participants()
```

Get the list of iteration events:

```
iteration_events = issue.resource_iteration_events.list()
```

Get the list of weight events:

```
weight_events = issue.resource_weight_events.list()
```

## Issue links

### Reference

- v4 API:
  - *gitlab.v4.objects.ProjectIssueLink*
  - *gitlab.v4.objects.ProjectIssueLinkManager*
  - *gitlab.v4.objects.ProjectIssue.links*
- GitLab API: [https://docs.gitlab.com/ee/api/issue\\_links.html](https://docs.gitlab.com/ee/api/issue_links.html)

## Examples

List the issues linked to `i1`:

```
links = i1.links.list()
```

Link issue `i1` to issue `i2`:

```
data = {
    'target_project_id': i2.project_id,
    'target_issue_iid': i2.iid
}
src_issue, dest_issue = i1.links.create(data)
```

---

**Note:** The `create()` method returns the source and destination `ProjectIssue` objects, not a `ProjectIssueLink` object.

---

Delete a link:

```
i1.links.delete(issue_link_id)
```

## Issues statistics

### Reference

- v4 API:
  - `gitlab.v4.objects.IssuesStatistics`
  - `gitlab.v4.objects.IssuesStatisticsManager`
  - `gitlab.issues_statistics`
  - `gitlab.v4.objects.GroupIssuesStatistics`
  - `gitlab.v4.objects.GroupIssuesStatisticsManager`
  - `gitlab.v4.objects.Group.issues_statistics`
  - `gitlab.v4.objects.ProjectIssuesStatistics`
  - `gitlab.v4.objects.ProjectIssuesStatisticsManager`
  - `gitlab.v4.objects.Project.issues_statistics`
- GitLab API: [https://docs.gitlab.com/ce/api/issues\\_statistics.htm](https://docs.gitlab.com/ce/api/issues_statistics.htm)



## Examples

Get statistics of all issues created by the current user:

```
statistics = gl.issues_statistics.get()
```

Get statistics of all issues the user has access to:

```
statistics = gl.issues_statistics.get(scope='all')
```

Get statistics of issues for the user with foobar in the title or the description:

```
statistics = gl.issues_statistics.get(search='foobar')
```

Get statistics of all issues in a group:

```
statistics = group.issues_statistics.get()
```

Get statistics of issues in a group with foobar in the title or the description:

```
statistics = group.issues_statistics.get(search='foobar')
```

Get statistics of all issues in a project:

```
statistics = project.issues_statistics.get()
```

Get statistics of issues in a project with foobar in the title or the description:

```
statistics = project.issues_statistics.get(search='foobar')
```

## 7.5.25 Iterations

### Reference

- v4 API:
  - *gitlab.v4.objects.GroupIteration*
  - *gitlab.v4.objects.GroupIterationManager*
  - *gitlab.v4.objects.Group.iterations*
  - *gitlab.v4.objects.ProjectIterationManager*
  - *gitlab.v4.objects.Project.iterations*
- GitLab API: <https://docs.gitlab.com/ee/api/iterations.html>

## Examples

---

**Note:** GitLab no longer has project iterations. Using a project endpoint returns the ancestor groups' iterations.

---

List iterations for a project's ancestor groups:

```
iterations = project.iterations.list()
```

List iterations for a group:

```
iterations = group.iterations.list()
```

## 7.5.26 CI/CD job token scope

### Reference

- v4 API:
  - *gitlab.v4.objects.ProjectJobTokenScope*
  - *gitlab.v4.objects.ProjectJobTokenScopeManager*
  - *gitlab.v4.objects.Project.job\_token\_scope*
- GitLab API: [https://docs.gitlab.com/ee/api/project\\_job\\_token\\_scopes.html](https://docs.gitlab.com/ee/api/project_job_token_scopes.html)

## Examples

**Warning:** The GitLab API does **not** return any data when saving or updating the job token scope settings. You need to call `refresh()` (or `get()` a new object) as shown below to get the latest state.

Get a project's CI/CD job token access settings:

```
scope = project.job_token_scope.get()
print(scope.inbound_enabled)
# True
```

Update the job token scope settings:

```
scope.enabled = False
scope.save()
```

**Warning:** As you can see above, the attributes you receive from and send to the GitLab API are not consistent. GitLab returns `inbound_enabled` and `outbound_enabled`, but expects `enabled`, which only refers to the inbound scope. This is important when accessing and updating these attributes.

Or update the job token scope settings directly:

```
project.job_token_scope.update(new_data={"enabled": True})
```

Refresh the current state of job token scope:

```
scope.refresh()
print(scope.inbound_enabled)
# False
```

## 7.5.27 Keys

### Keys

#### Reference

- v4 API
  - *gitlab.v4.objects.Key*
  - *gitlab.v4.objects.KeyManager*
  - *gitlab.Gitlab.keys*
- GitLab API: <https://docs.gitlab.com/ce/api/keys.html>

### Examples

Get an ssh key by its id (requires admin access):

```
key = gl.keys.get(key_id)
```

Get an ssh key (requires admin access) or a deploy key by its fingerprint:

```
key = gl.keys.get(fingerprint="SHA256:ERJJ/OweAM6jA80jJ/gXs4N5fqUaREEJnz/EyfywFX")
```

## 7.5.28 Issue boards

### Boards

Boards are a visual representation of existing issues for a project or a group. Issues can be moved from one list to the other to track progress and help with priorities.

#### Reference

- v4 API:
  - *gitlab.v4.objects.ProjectBoard*
  - *gitlab.v4.objects.ProjectBoardManager*
  - *gitlab.v4.objects.Project.boards*
  - *gitlab.v4.objects.GroupBoard*

- `gitlab.v4.objects.GroupBoardManager`
- `gitlab.v4.objects.Group.boards`
- GitLab API:
  - <https://docs.gitlab.com/ce/api/boards.html>
  - [https://docs.gitlab.com/ce/api/group\\_boards.html](https://docs.gitlab.com/ce/api/group_boards.html)

## Examples

Get the list of existing boards for a project or a group:

```
# item is a Project or a Group
boards = project_or_group.boards.list()
```

Get a single board for a project or a group:

```
board = project_or_group.boards.get(board_id)
```

Create a board:

```
board = project_or_group.boards.create({'name': 'new-board'})
```

---

**Note:** Board creation is not supported in the GitLab CE edition.

---

Delete a board:

```
board.delete()
# or
project_or_group.boards.delete(board_id)
```

---

**Note:** Board deletion is not supported in the GitLab CE edition.

---

## Board lists

Boards are made of lists of issues. Each list is associated to a label, and issues tagged with this label automatically belong to the list.

## Reference

- v4 API:
  - `gitlab.v4.objects.ProjectBoardList`
  - `gitlab.v4.objects.ProjectBoardListManager`
  - `gitlab.v4.objects.ProjectBoard.lists`
  - `gitlab.v4.objects.GroupBoardList`
  - `gitlab.v4.objects.GroupBoardListManager`

- `gitlab.v4.objects.GroupBoard.lists`
- GitLab API:
  - <https://docs.gitlab.com/ce/api/boards.html>
  - [https://docs.gitlab.com/ce/api/group\\_boards.html](https://docs.gitlab.com/ce/api/group_boards.html)

## Examples

List the issue lists for a board:

```
b_lists = board.lists.list()
```

Get a single list:

```
b_list = board.lists.get(list_id)
```

Create a new list:

```
# First get a ProjectLabel
label = get_or_create_label()
# Then use its ID to create the new board list
b_list = board.lists.create({'label_id': label.id})
```

Change a list position. The first list is at position 0. Moving a list will set it at the given position and move the following lists up a position:

```
b_list.position = 2
b_list.save()
```

Delete a list:

```
b_list.delete()
```

## 7.5.29 Labels

### Project labels

#### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectLabel`
  - `gitlab.v4.objects.ProjectLabelManager`
  - `gitlab.v4.objects.Project.labels`
- GitLab API: <https://docs.gitlab.com/ce/api/labels.html>

## Examples

List labels for a project:

```
labels = project.labels.list()
```

Create a label for a project:

```
label = project.labels.create({'name': 'foo', 'color': '#8899aa'})
```

Update a label for a project:

```
# change the name of the label:
label.new_name = 'bar'
label.save()
# change its color:
label.color = '#112233'
label.save()
```

Promote a project label to a group label:

```
label.promote()
```

Delete a label for a project:

```
project.labels.delete(label_id)
# or
label.delete()
```

Manage labels in issues and merge requests:

```
# Labels are defined as lists in issues and merge requests. The labels must
# exist.
issue = p.issues.create({'title': 'issue title',
                        'description': 'issue description',
                        'labels': ['foo']})
issue.labels.append('bar')
issue.save()
```

## Label events

Resource label events keep track about who, when, and which label was added or removed to an issuable.

Group epic label events are only available in the EE edition.

## Reference

- v4 API:
  - `gitlab.v4.objects.ProjectIssueResourceLabelEvent`
  - `gitlab.v4.objects.ProjectIssueResourceLabelEventManager`
  - `gitlab.v4.objects.ProjectIssue.resourcelabelevents`
  - `gitlab.v4.objects.ProjectMergeRequestResourceLabelEvent`
  - `gitlab.v4.objects.ProjectMergeRequestResourceLabelEventManager`
  - `gitlab.v4.objects.ProjectMergeRequest.resourcelabelevents`
  - `gitlab.v4.objects.GroupEpicResourceLabelEvent`
  - `gitlab.v4.objects.GroupEpicResourceLabelEventManager`
  - `gitlab.v4.objects.GroupEpic.resourcelabelevents`
- GitLab API: [https://docs.gitlab.com/ee/api/resource\\_label\\_events.html](https://docs.gitlab.com/ee/api/resource_label_events.html)

## Examples

Get the events for a resource (issue, merge request or epic):

```
events = resource.resourcelabelevents.list()
```

Get a specific event for a resource:

```
event = resource.resourcelabelevents.get(event_id)
```

### 7.5.30 Notification settings

You can define notification settings globally, for groups and for projects. Valid levels are defined as constants:

- `gitlab.const.NotificationLevel.DISABLED`
- `gitlab.const.NotificationLevel.PARTICIPATING`
- `gitlab.const.NotificationLevel.WATCH`
- `gitlab.const.NotificationLevel.GLOBAL`
- `gitlab.const.NotificationLevel.MENTION`
- `gitlab.const.NotificationLevel.CUSTOM`

You get access to fine-grained settings if you use the `NOTIFICATION_LEVEL_CUSTOM` level.

## Reference

- v4 API:
  - `gitlab.v4.objects.NotificationSettings`
  - `gitlab.v4.objects.NotificationSettingsManager`
  - `gitlab.Gitlab.notificationsettings`
  - `gitlab.v4.objects.GroupNotificationSettings`
  - `gitlab.v4.objects.GroupNotificationSettingsManager`
  - `gitlab.v4.objects.Group.notificationsettings`
  - `gitlab.v4.objects.ProjectNotificationSettings`
  - `gitlab.v4.objects.ProjectNotificationSettingsManager`
  - `gitlab.v4.objects.Project.notificationsettings`
- GitLab API: [https://docs.gitlab.com/ce/api/notification\\_settings.html](https://docs.gitlab.com/ce/api/notification_settings.html)

## Examples

Get the notifications settings:

```
# global settings
settings = gl.notificationsettings.get()
# for a group
settings = gl.groups.get(group_id).notificationsettings.get()
# for a project
settings = gl.projects.get(project_id).notificationsettings.get()
```

Update the notifications settings:

```
# use a predefined level
settings.level = gitlab.const.NotificationLevel.WATCH

# create a custom setup
settings.level = gitlab.const.NotificationLevel.CUSTOM
settings.save() # will create additional attributes, but not mandatory

settings.new_merge_request = True
settings.new_issue = True
settings.new_note = True
settings.save()
```



### 7.5.31 Merge Trains

#### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectMergeTrain`
  - `gitlab.v4.objects.ProjectMergeTrainManager`
  - `gitlab.v4.objects.Project.merge_trains`
- GitLab API: [https://docs.gitlab.com/ee/api/merge\\_trains.html](https://docs.gitlab.com/ee/api/merge_trains.html)

#### Examples

List merge trains for a project:

```
merge_trains = project.merge_trains.list()
```

List active merge trains for a project:

```
merge_trains = project.merge_trains.list(scope="active")
```

List completed (have been merged) merge trains for a project:

```
merge_trains = project.merge_trains.list(scope="complete")
```

### 7.5.32 Merge requests

You can use merge requests to notify a project that a branch is ready for merging. The owner of the target project can accept the merge request.

Merge requests are linked to projects, but they can be listed globally or for groups.

#### Group and global listing

#### Reference

- v4 API:
  - `gitlab.v4.objects.GroupMergeRequest`
  - `gitlab.v4.objects.GroupMergeRequestManager`
  - `gitlab.v4.objects.Group.mergerequests`
  - `gitlab.v4.objects.MergeRequest`
  - `gitlab.v4.objects.MergeRequestManager`
  - `gitlab.Gitlab.mergerequests`
- GitLab API: [https://docs.gitlab.com/ce/api/merge\\_requests.html](https://docs.gitlab.com/ce/api/merge_requests.html)

## Examples

List the merge requests created by the user of the token on the GitLab server:

```
mrs = gl.mergerequests.list()
```

List the merge requests available on the GitLab server:

```
mrs = gl.mergerequests.list(scope="all")
```

List the merge requests for a group:

```
group = gl.groups.get('mygroup')
mrs = group.mergerequests.list()
```

---

**Note:** It is not possible to edit or delete `MergeRequest` and `GroupMergeRequest` objects. You need to create a `ProjectMergeRequest` object to apply changes:

```
mr = group.mergerequests.list()[0]
project = gl.projects.get(mr.project_id, lazy=True)
editable_mr = project.mergerequests.get(mr.iid, lazy=True)
editable_mr.title = updated_title
editable_mr.save()
```

---

## Project merge requests

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectMergeRequest`
  - `gitlab.v4.objects.ProjectMergeRequestManager`
  - `gitlab.v4.objects.Project.mergerequests`
- GitLab API: [https://docs.gitlab.com/ce/api/merge\\_requests.html](https://docs.gitlab.com/ce/api/merge_requests.html)

## Examples

List MRs for a project:

```
mrs = project.mergerequests.list()
```

You can filter and sort the returned list with the following parameters:

- **state:** state of the MR. It can be one of `all`, `merged`, `opened`, `closed` or `locked`
- **order\_by:** sort by `created_at` or `updated_at`
- **sort:** sort order (`asc` or `desc`)

You can find a full updated list of parameters here: [https://docs.gitlab.com/ee/api/merge\\_requests.html#list-merge-requests](https://docs.gitlab.com/ee/api/merge_requests.html#list-merge-requests)

For example:

```
mrs = project.mergerequests.list(state='merged', order_by='updated_at')
```

Get a single MR:

```
mr = project.mergerequests.get(mr_iid)
```

**Get MR reviewer details::**

```
mr = project.mergerequests.get(mr_iid) reviewers = mr.viewer_details.list()
```

Create a MR:

```
mr = project.mergerequests.create({'source_branch': 'cool_feature',
                                   'target_branch': 'main',
                                   'title': 'merge cool feature',
                                   'labels': ['label1', 'label2']})
```

Update a MR:

```
mr.description = 'New description'
mr.labels = ['foo', 'bar']
mr.save()
```

Change the state of a MR (close or reopen):

```
mr.state_event = 'close' # or 'reopen'
mr.save()
```

Delete a MR:

```
project.mergerequests.delete(mr_iid)
# or
mr.delete()
```

Accept a MR:

```
mr.merge()
```

Schedule a MR to merge after the pipeline(s) succeed:

```
mr.merge(merge_when_pipeline_succeeds=True)
```

Cancel a MR from merging when the pipeline succeeds:

```
# Cancel a MR from being merged that had been previously set to
# 'merge_when_pipeline_succeeds=True'
mr.cancel_merge_when_pipeline_succeeds()
```

List commits of a MR:

```
commits = mr.commits()
```

List the changes of a MR:

```
changes = mr.changes()
```

List issues that will close on merge:

```
mr.closes_issues()
```

Subscribe to / unsubscribe from a MR:

```
mr.subscribe()  
mr.unsubscribe()
```

Mark a MR as todo:

```
mr.todo()
```

List the diffs for a merge request:

```
diffs = mr.diffs.list()
```

Get a diff for a merge request:

```
diff = mr.diffs.get(diff_id)
```

Get time tracking stats:

```
time_stats = mr.time_stats()
```

On recent versions of Gitlab the time stats are also returned as a merge request object attribute:

```
mr = project.mergerequests.get(id)  
print(mr.attributes['time_stats'])
```

Set a time estimate for a merge request:

```
mr.time_estimate('3h30m')
```

Reset a time estimate for a merge request:

```
mr.reset_time_estimate()
```

Add spent time for a merge request:

```
mr.add_spent_time('3h30m')
```

Reset spent time for a merge request:

```
mr.reset_spent_time()
```

Get user agent detail for the issue (admin only):

```
detail = issue.user_agent_detail()
```

Attempt to rebase an MR:

```
mr.rebase()
```

Clear all approvals of a merge request (possible with project or group access tokens only):

```
mr.reset_approvals()
```

Get status of a rebase for an MR:

```
mr = project.mergerequests.get(mr_id, include_rebase_in_progress=True)
print(mr.rebase_in_progress, mr.merge_error)
```

For more info see: [https://docs.gitlab.com/ee/api/merge\\_requests.html#rebase-a-merge-request](https://docs.gitlab.com/ee/api/merge_requests.html#rebase-a-merge-request)

Attempt to merge changes between source and target branch:

```
response = mr.merge_ref()
print(response['commit_id'])
```

## Merge Request Pipelines

### Reference

- v4 API:
  - *gitlab.v4.objects.ProjectMergeRequestPipeline*
  - *gitlab.v4.objects.ProjectMergeRequestPipelineManager*
  - *gitlab.v4.objects.ProjectMergeRequest.pipelines*
- GitLab API: [https://docs.gitlab.com/ee/api/merge\\_requests.html#list-mr-pipelines](https://docs.gitlab.com/ee/api/merge_requests.html#list-mr-pipelines)

### Examples

List pipelines for a merge request:

```
pipelines = mr.pipelines.list()
```

Create a pipeline for a merge request:

```
pipeline = mr.pipelines.create()
```

## 7.5.33 Merge request approvals settings

Merge request approvals can be defined at the project level or at the merge request level.

## References

- v4 API:
  - `gitlab.v4.objects.ProjectApproval`
  - `gitlab.v4.objects.ProjectApprovalManager`
  - `gitlab.v4.objects.ProjectApprovalRule`
  - `gitlab.v4.objects.ProjectApprovalRuleManager`
  - `gitlab.v4.objects.Project.approvals`
  - `gitlab.v4.objects.ProjectMergeRequestApproval`
  - `gitlab.v4.objects.ProjectMergeRequestApprovalManager`
  - `gitlab.v4.objects.ProjectMergeRequest.approvals`
  - `gitlab.v4.objects.ProjectMergeRequestApprovalRule`
  - `gitlab.v4.objects.ProjectMergeRequestApprovalRuleManager`
  - `gitlab.v4.objects.ProjectMergeRequest.approval_rules`
  - `gitlab.v4.objects.ProjectMergeRequestApprovalState`
  - `gitlab.v4.objects.ProjectMergeRequestApprovalStateManager`
  - `gitlab.v4.objects.ProjectMergeRequest.approval_state`
- GitLab API: [https://docs.gitlab.com/ee/api/merge\\_request\\_approvals.html](https://docs.gitlab.com/ee/api/merge_request_approvals.html)

## Examples

List project-level MR approval rules:

```
p_mras = project.approvalrules.list()
```

Change project-level MR approval rule:

```
p_approvalrule.user_ids = [234]
p_approvalrule.save()
```

Delete project-level MR approval rule:

```
p_approvalrule.delete()
```

Get project-level or MR-level MR approvals settings:

```
p_mras = project.approvals.get()
mr_mras = mr.approvals.get()
```

Get MR-level approval state:

```
mr_approval_state = mr.approval_state.get()
```

Change project-level or MR-level MR approvals settings:

```
p_mras.approvals_before_merge = 2
p_mras.save()

mr.approvals.set_approvers(approvals_required=1)
# or
mr_mras.approvals_required = 1
mr_mras.save()
```

Create a new MR-level approval rule or change an existing MR-level approval rule:

```
mr.approvals.set_approvers(approvals_required = 1, approver_ids=[105],
                           approver_group_ids=[653, 654],
                           approval_rule_name="my MR custom approval rule")
```

List MR-level MR approval rules:

```
mr.approval_rules.list()
```

Get a single MR approval rule:

```
approval_rule_id = 123
mr_approvalrule = mr.approval_rules.get(approval_rule_id)
```

Delete MR-level MR approval rule:

```
rules = mr.approval_rules.list()
rules[0].delete()

# or
mr.approval_rules.delete(approval_id)
```

Change MR-level MR approval rule:

```
mr_approvalrule.user_ids = [105]
mr_approvalrule.approvals_required = 2
mr_approvalrule.group_ids = [653, 654]
mr_approvalrule.save()
```

Create a MR-level MR approval rule:

```
mr.approval_rules.create({
    "name": "my MR custom approval rule",
    "approvals_required": 2,
    "rule_type": "regular",
    "user_ids": [105],
    "group_ids": [653, 654],
})
```

## 7.5.34 Milestones

### Project milestones

#### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectMilestone`
  - `gitlab.v4.objects.ProjectMilestoneManager`
  - `gitlab.v4.objects.Project.milestones`
  - `gitlab.v4.objects.GroupMilestone`
  - `gitlab.v4.objects.GroupMilestoneManager`
  - `gitlab.v4.objects.Group.milestones`
- GitLab API:
  - <https://docs.gitlab.com/ce/api/milestones.html>
  - [https://docs.gitlab.com/ce/api/group\\_milestones.html](https://docs.gitlab.com/ce/api/group_milestones.html)

#### Examples

List the milestones for a project or a group:

```
p_milestones = project.milestones.list()
g_milestones = group.milestones.list()
```

You can filter the list using the following parameters:

- `iids`: unique IDs of milestones for the project
- `state`: either `active` or `closed`
- `search`: to search using a string

```
p_milestones = project.milestones.list(state='closed')
g_milestones = group.milestones.list(state='active')
```

Get a single milestone:

```
p_milestone = project.milestones.get(milestone_id)
g_milestone = group.milestones.get(milestone_id)
```

Create a milestone:

```
milestone = project.milestones.create({'title': '1.0'})
```

Edit a milestone:

```
milestone.description = 'v 1.0 release'
milestone.save()
```

Change the state of a milestone (activate / close):



```
# close a milestone
milestone.state_event = 'close'
milestone.save()

# activate a milestone
milestone.state_event = 'activate'
milestone.save()
```

Promote a project milestone:

```
milestone.promote()
```

List the issues related to a milestone:

```
issues = milestone.issues()
```

List the merge requests related to a milestone:

```
merge_requests = milestone.merge_requests()
```

## Milestone events

Resource milestone events keep track of what happens to GitLab issues and merge requests.

## Reference

- v4 API:
  - *gitlab.v4.objects.ProjectIssueResourceMilestoneEvent*
  - *gitlab.v4.objects.ProjectIssueResourceMilestoneEventManager*
  - *gitlab.v4.objects.ProjectIssue.resourcemilestoneevents*
  - *gitlab.v4.objects.ProjectMergeRequestResourceMilestoneEvent*
  - *gitlab.v4.objects.ProjectMergeRequestResourceMilestoneEventManager*
  - *gitlab.v4.objects.ProjectMergeRequest.resourcemilestoneevents*
- GitLab API: [https://docs.gitlab.com/ee/api/resource\\_milestone\\_events.html](https://docs.gitlab.com/ee/api/resource_milestone_events.html)

## Examples

Get milestones for a resource (issue, merge request):

```
milestones = resource.resourcemilestoneevents.list()
```

Get a specific milestone for a resource:

```
milestone = resource.resourcemilestoneevents.get(milestone_id)
```

### 7.5.35 Namespaces

#### Reference

- v4 API:
  - `gitlab.v4.objects.Namespace`
  - `gitlab.v4.objects.NamespaceManager`
  - `gitlab.Gitlab.namespaces`
- GitLab API: <https://docs.gitlab.com/ce/api/namespaces.html>

#### Examples

List namespaces:

```
namespaces = gl.namespaces.list()
```

Search namespaces:

```
namespaces = gl.namespaces.list(search='foo')
```

Get a namespace by ID or path:

```
namespace = gl.namespaces.get("my-namespace")
```

Get existence of a namespace by path:

```
namespace = gl.namespaces.exists("new-namespace")

if namespace.exists:
    # get suggestions of namespaces that don't already exist
    print(namespace.suggests)
```

### 7.5.36 Notes

You can manipulate notes (comments) on group epics, project issues, merge requests and snippets.

#### Reference

- v4 API:
  - Epics:
    - `gitlab.v4.objects.GroupEpicNote`
    - `gitlab.v4.objects.GroupEpicNoteManager`
    - `gitlab.v4.objects.GroupEpic.notes`
  - Issues:
    - `gitlab.v4.objects.ProjectIssueNote`
    - `gitlab.v4.objects.ProjectIssueNoteManager`

- `gitlab.v4.objects.ProjectIssue.notes`

MergeRequests:

- `gitlab.v4.objects.ProjectMergeRequestNote`
- `gitlab.v4.objects.ProjectMergeRequestNoteManager`
- `gitlab.v4.objects.ProjectMergeRequest.notes`

Snippets:

- `gitlab.v4.objects.ProjectSnippetNote`
- `gitlab.v4.objects.ProjectSnippetNoteManager`
- `gitlab.v4.objects.ProjectSnippet.notes`

- GitLab API: <https://docs.gitlab.com/ce/api/notes.html>

## Examples

List the notes for a resource:

```
e_notes = epic.notes.list()
i_notes = issue.notes.list()
mr_notes = mr.notes.list()
s_notes = snippet.notes.list()
```

Get a note for a resource:

```
e_note = epic.notes.get(note_id)
i_note = issue.notes.get(note_id)
mr_note = mr.notes.get(note_id)
s_note = snippet.notes.get(note_id)
```

Create a note for a resource:

```
e_note = epic.notes.create({'body': 'note content'})
i_note = issue.notes.create({'body': 'note content'})
mr_note = mr.notes.create({'body': 'note content'})
s_note = snippet.notes.create({'body': 'note content'})
```

Update a note for a resource:

```
note.body = 'updated note content'
note.save()
```

Delete a note for a resource:

```
note.delete()
```

## 7.5.37 Packages

Packages allow you to utilize GitLab as a private repository for a variety of common package managers, as well as GitLab's generic package registry.

### Project Packages

#### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectPackage`
  - `gitlab.v4.objects.ProjectPackageManager`
  - `gitlab.v4.objects.Project.packages`
- GitLab API: <https://docs.gitlab.com/ee/api/packages.html#within-a-project>

#### Examples

List the packages in a project:

```
packages = project.packages.list()
```

Filter the results by `package_type` or `package_name`

```
packages = project.packages.list(package_type='pypi')
```

Get a specific package of a project by id:

```
package = project.packages.get(1)
```

Delete a package from a project:

```
package.delete()  
# or  
project.packages.delete(package.id)
```

### Group Packages

#### Reference

- v4 API:
  - `gitlab.v4.objects.GroupPackage`
  - `gitlab.v4.objects.GroupPackageManager`
  - `gitlab.v4.objects.Group.packages`
- GitLab API: <https://docs.gitlab.com/ee/api/packages.html#within-a-group>

## Examples

List the packages in a group:

```
packages = group.packages.list()
```

Filter the results by `package_type` or `package_name`

```
packages = group.packages.list(package_type='pypi')
```

## Project Package Files

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectPackageFile`
  - `gitlab.v4.objects.ProjectPackageFileManager`
  - `gitlab.v4.objects.ProjectPackage.package_files`
- GitLab API: <https://docs.gitlab.com/ee/api/packages.html#list-package-files>

## Examples

List package files for package in project:

```
package = project.packages.get(1)
package_files = package.package_files.list()
```

Delete a package file in a project:

```
package = project.packages.get(1)
file = package.package_files.list()[0]
file.delete()
```

## Project Package Pipelines

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectPackagePipeline`
  - `gitlab.v4.objects.ProjectPackagePipelineManager`
  - `gitlab.v4.objects.ProjectPackage.pipelines`
- GitLab API: <https://docs.gitlab.com/ee/api/packages.html#list-package-pipelines>

## Examples

List package pipelines for package in project:

```
package = project.packages.get(1)
package_pipelines = package.pipelines.list()
```

## Generic Packages

You can use python-gitlab to upload and download generic packages.

## Reference

- v4 API:
  - *gitlab.v4.objects.GenericPackage*
  - *gitlab.v4.objects.GenericPackageManager*
  - *gitlab.v4.objects.Project.generic\_packages*
- GitLab API: [https://docs.gitlab.com/ee/user/packages/generic\\_packages](https://docs.gitlab.com/ee/user/packages/generic_packages)

## Examples

Upload a generic package to a project:

```
project = gl.projects.get(1, lazy=True)
package = project.generic_packages.upload(
    package_name="hello-world",
    package_version="v1.0.0",
    file_name="hello.tar.gz",
    path="/path/to/local/hello.tar.gz"
)
```

Download a project's generic package:

```
project = gl.projects.get(1, lazy=True)
package = project.generic_packages.download(
    package_name="hello-world",
    package_version="v1.0.0",
    file_name="hello.tar.gz",
)
```

---

**Hint:** You can use the Packages API described above to find packages and retrieve the metadata you need download them.

---

## 7.5.38 Pages domains

### Admin

#### References

- v4 API:
  - `gitlab.v4.objects.PagesDomain`
  - `gitlab.v4.objects.PagesDomainManager`
  - `gitlab.Gitlab.pagesdomains`
- GitLab API: [https://docs.gitlab.com/ce/api/pages\\_domains.html#list-all-pages-domains](https://docs.gitlab.com/ce/api/pages_domains.html#list-all-pages-domains)

#### Examples

List all the existing domains (admin only):

```
domains = gl.pagesdomains.list()
```

### Project pages domain

#### References

- v4 API:
  - `gitlab.v4.objects.ProjectPagesDomain`
  - `gitlab.v4.objects.ProjectPagesDomainManager`
  - `gitlab.v4.objects.Project.pagesdomains`
- GitLab API: [https://docs.gitlab.com/ce/api/pages\\_domains.html#list-pages-domains](https://docs.gitlab.com/ce/api/pages_domains.html#list-pages-domains)

#### Examples

List domains for a project:

```
domains = project.pagesdomains.list()
```

Get a single domain:

```
domain = project.pagesdomains.get('d1.example.com')
```

Create a new domain:

```
domain = project.pagesdomains.create({'domain': 'd2.example.com'})
```

Update an existing domain:

```
domain.certificate = open('d2.crt').read()
domain.key = open('d2.key').read()
domain.save()
```

Delete an existing domain:

```
domain.delete
# or
project.pagesdomains.delete('d2.example.com')
```

## 7.5.39 Personal Access Tokens

### References

- v4 API:
  - `gitlab.v4.objects.PersonalAccessToken`
  - `gitlab.v4.objects.PersonalAccessTokenManager`
  - `gitlab.Gitlab.personal_access_tokens`
  - `gitlab.v4.objects.UserPersonalAccessToken`
  - `gitlab.v4.objects.UserPersonalAccessTokenManager`
  - `gitlab.Gitlab.User.personal_access_tokens`
- GitLab API:
  - [https://docs.gitlab.com/ee/api/personal\\_access\\_tokens.html](https://docs.gitlab.com/ee/api/personal_access_tokens.html)
  - <https://docs.gitlab.com/ee/api/users.html#create-a-personal-access-token>

### Examples

List personal access tokens:

```
access_tokens = gl.personal_access_tokens.list()
print(access_tokens[0].name)
```

List personal access tokens from other user\_id (admin only):

```
access_tokens = gl.personal_access_tokens.list(user_id=25)
```

Get a personal access token by id:

```
gl.personal_access_tokens.get(123)
```

Get the personal access token currently used:

```
gl.personal_access_tokens.get("self")
```

Revoke a personal access token fetched via list:

```
access_token = access_tokens[0]
access_token.delete()
```

Revoke a personal access token by id:



```
gl.personal_access_tokens.delete(123)
```

Revoke the personal access token currently used:

```
gl.personal_access_tokens.delete("self")
```

Rotate a personal access token and retrieve its new value:

```
token = gl.personal_access_tokens.get(42, lazy=True)
token.rotate()
print(token.token)
# or directly using a token ID
new_token_dict = gl.personal_access_tokens.rotate(42)
print(new_token_dict)
```

Create a personal access token for a user (admin only):

```
user = gl.users.get(25, lazy=True)
access_token = user.personal_access_tokens.create({"name": "test", "scopes": "api"})
```

---

**Note:** As you can see above, you can only create personal access tokens via the Users API, but you cannot revoke these objects directly. This is because the create API uses a different endpoint than the list and revoke APIs. You need to fetch the token via the list or get API first to revoke it.

---

## 7.5.40 Pipelines and Jobs

### Project pipelines

A pipeline is a group of jobs executed by GitLab CI.

#### Reference

- v4 API:
  - *gitlab.v4.objects.ProjectPipeline*
  - *gitlab.v4.objects.ProjectPipelineManager*
  - *gitlab.v4.objects.Project.pipelines*
- GitLab API: <https://docs.gitlab.com/ce/api/pipelines.html>

## Examples

List pipelines for a project:

```
pipelines = project.pipelines.list()
```

Get a pipeline for a project:

```
pipeline = project.pipelines.get(pipeline_id)
```

Get variables of a pipeline:

```
variables = pipeline.variables.list()
```

Create a pipeline for a particular reference with custom variables:

```
pipeline = project.pipelines.create({'ref': 'main', 'variables': [{ 'key': 'MY_VARIABLE',  
↪ 'value': 'hello' }]} )
```

Retry the failed builds for a pipeline:

```
pipeline.retry()
```

Cancel builds in a pipeline:

```
pipeline.cancel()
```

Delete a pipeline:

```
pipeline.delete()
```

## Triggers

Triggers provide a way to interact with the GitLab CI. Using a trigger a user or an application can run a new build/job for a specific commit.

## Reference

- v4 API:
  - `gitlab.v4.objects.ProjectTrigger`
  - `gitlab.v4.objects.ProjectTriggerManager`
  - `gitlab.v4.objects.Project.triggers`
- GitLab API: [https://docs.gitlab.com/ce/api/pipeline\\_triggers.html](https://docs.gitlab.com/ce/api/pipeline_triggers.html)

## Examples

List triggers:

```
triggers = project.triggers.list()
```

Get a trigger:

```
trigger = project.triggers.get(trigger_token)
```

Create a trigger:

```
trigger = project.triggers.create({'description': 'mytrigger'})
```

Remove a trigger:

```
project.triggers.delete(trigger_token)
# or
trigger.delete()
```

Full example with wait for finish:

```
def get_or_create_trigger(project):
    trigger_description = 'my_trigger_id'
    for t in project.triggers.list():
        if t.description == trigger_description:
            return t
    return project.triggers.create({'description': trigger_description})

trigger = get_or_create_trigger(project)
pipeline = project.trigger_pipeline('main', trigger.token, variables={"DEPLOY_ZONE": "us-
↪west1"})
while pipeline.finished_at is None:
    pipeline.refresh()
    time.sleep(1)
```

You can trigger a pipeline using token authentication instead of user authentication. To do so create an anonymous Gitlab instance and use lazy objects to get the associated project:

```
gl = gitlab.Gitlab(URL) # no authentication
project = gl.projects.get(project_id, lazy=True) # no API call
project.trigger_pipeline('main', trigger_token)
```

Reference: <https://docs.gitlab.com/ee/ci/triggers/#trigger-token>

## Pipeline schedules

You can schedule pipeline runs using a cron-like syntax. Variables can be associated with the scheduled pipelines.

### Reference

- v4 API
  - `gitlab.v4.objects.ProjectPipelineSchedule`
  - `gitlab.v4.objects.ProjectPipelineScheduleManager`
  - `gitlab.v4.objects.Project.pipelineschedules`
  - `gitlab.v4.objects.ProjectPipelineScheduleVariable`
  - `gitlab.v4.objects.ProjectPipelineScheduleVariableManager`
  - `gitlab.v4.objects.ProjectPipelineSchedule.variables`
  - `gitlab.v4.objects.ProjectPipelineSchedulePipeline`
  - `gitlab.v4.objects.ProjectPipelineSchedulePipelineManager`
  - `gitlab.v4.objects.ProjectPipelineSchedule.pipelines`
- GitLab API: [https://docs.gitlab.com/ce/api/pipeline\\_schedules.html](https://docs.gitlab.com/ce/api/pipeline_schedules.html)

### Examples

List pipeline schedules:

```
scheds = project.pipelineschedules.list()
```

Get a single schedule:

```
sched = project.pipelineschedules.get(schedule_id)
```

Create a new schedule:

```
sched = project.pipelineschedules.create({  
    'ref': 'main',  
    'description': 'Daily test',  
    'cron': '0 1 * * *'})
```

Update a schedule:

```
sched.cron = '1 2 * * *'  
sched.save()
```

Take ownership of a schedule:

```
sched.take_ownership()
```

Trigger a pipeline schedule immediately:

```
sched = projects.pipelineschedules.get(schedule_id)  
sched.play()
```

Delete a schedule:

```
sched.delete()
```

List schedule variables:

```
# note: you need to use get() to retrieve the schedule variables. The
# attribute is not present in the response of a list() call
sched = projects.pipelineschedules.get(schedule_id)
vars = sched.attributes['variables']
```

Create a schedule variable:

```
var = sched.variables.create({'key': 'foo', 'value': 'bar'})
```

Edit a schedule variable:

```
var.value = 'new_value'
var.save()
```

Delete a schedule variable:

```
var.delete()
```

List all pipelines triggered by a pipeline schedule:

```
pipelines = sched.pipelines.list()
```

## Jobs

Jobs are associated to projects, pipelines and commits. They provide information on the jobs that have been run, and methods to manipulate them.

## Reference

- v4 API
  - `gitlab.v4.objects.ProjectJob`
  - `gitlab.v4.objects.ProjectJobManager`
  - `gitlab.v4.objects.Project.jobs`
- GitLab API: <https://docs.gitlab.com/ce/api/jobs.html>

## Examples

Jobs are usually automatically triggered, but you can explicitly trigger a new job:

```
project.trigger_build('main', trigger_token,
                      {'extra_var1': 'foo', 'extra_var2': 'bar'})
```

List jobs for the project:

```
jobs = project.jobs.list()
```

Get a single job:

```
project.jobs.get(job_id)
```

List the jobs of a pipeline:

```
project = gl.projects.get(project_id)
pipeline = project.pipelines.get(pipeline_id)
jobs = pipeline.jobs.list()
```

---

**Note:** Job methods (play, cancel, and so on) are not available on ProjectPipelineJob objects. To use these methods create a ProjectJob object:

```
pipeline_job = pipeline.jobs.list()[0]
job = project.jobs.get(pipeline_job.id, lazy=True)
job.retry()
```

---

Get the artifacts of a job:

```
build_or_job.artifacts()
```

Get the artifacts of a job by its name from the latest successful pipeline of a branch or tag:

```
project.artifacts.download(ref_name='main', job='build')
```

**Warning:** Artifacts are entirely stored in memory in this example.

You can download artifacts as a stream. Provide a callable to handle the stream:

```
with open("archive.zip", "wb") as f:
    build_or_job.artifacts(streamed=True, action=f.write)
```

You can also directly stream the output into a file, and unzip it afterwards:

```
zipfn = "___artifacts.zip"
with open(zipfn, "wb") as f:
    build_or_job.artifacts(streamed=True, action=f.write)
subprocess.run(["unzip", "-bo", zipfn])
os.unlink(zipfn)
```

Or, you can also use the underlying response iterator directly:

```
artifact_bytes_iterator = build_or_job.artifacts(iterator=True)
```

This can be used with frameworks that expect an iterator (such as FastAPI/Starlette's `StreamingResponse`) to forward a download from GitLab without having to download the entire content server-side first:

```
@app.get("/download_artifact")
def download_artifact():
    artifact_bytes_iterator = build_or_job.artifacts(iterator=True)
    return StreamingResponse(artifact_bytes_iterator, media_type="application/zip")
```

Delete all artifacts of a project that can be deleted:

```
project.artifacts.delete()
```

Get a single artifact file:

```
build_or_job.artifact('path/to/file')
```

Get a single artifact file by branch and job:

```
project.artifacts.raw('branch', 'path/to/file', 'job')
```

Mark a job artifact as kept when expiration is set:

```
build_or_job.keep_artifacts()
```

Delete the artifacts of a job:

```
build_or_job.delete_artifacts()
```

Get a job log file / trace:

```
build_or_job.trace()
```

**Warning:** Traces are entirely stored in memory unless you use the streaming feature. See *the artifacts example*.

Cancel/retry a job:

```
build_or_job.cancel()
build_or_job.retry()
```

Play (trigger) a job:

```
build_or_job.play()
```

Erase a job (artifacts and trace):

```
build_or_job.erase()
```

## Pipeline bridges

Get a list of bridge jobs (including child pipelines) for a pipeline.

### Reference

- v4 API
  - *gitlab.v4.objects.ProjectPipelineBridge*
  - *gitlab.v4.objects.ProjectPipelineBridgeManager*
  - *gitlab.v4.objects.ProjectPipeline.bridges*
- GitLab API: <https://docs.gitlab.com/ee/api/jobs.html#list-pipeline-bridges>

### Examples

List bridges for the pipeline:

```
bridges = pipeline.bridges.list()
```

## Pipeline test report

Get a pipeline's complete test report.

### Reference

- v4 API
  - *gitlab.v4.objects.ProjectPipelineTestReport*
  - *gitlab.v4.objects.ProjectPipelineTestReportManager*
  - *gitlab.v4.objects.ProjectPipeline.test\_report*
- GitLab API: <https://docs.gitlab.com/ee/api/pipelines.html#get-a-pipelines-test-report>

### Examples

Get the test report for a pipeline:

```
test_report = pipeline.test_report.get()
```



## Pipeline test report summary

Get a pipeline's test report summary.

### Reference

- v4 API
  - `gitlab.v4.objects.ProjectPipelineTestReportSummary`
  - `gitlab.v4.objects.ProjectPipelineTestReportSummaryManager`
  - `gitlab.v4.objects.ProjectPipeline.test_report_summary`
- GitLab API: <https://docs.gitlab.com/ee/api/pipelines.html#get-a-pipelines-test-report-summary>

### Examples

Get the test report summary for a pipeline:

```
test_report_summary = pipeline.test_report_summary.get()
```

## 7.5.41 Projects

### Projects

#### Reference

- v4 API:
  - `gitlab.v4.objects.Project`
  - `gitlab.v4.objects.ProjectManager`
  - `gitlab.Gitlab.projects`
- GitLab API: <https://docs.gitlab.com/ce/api/projects.html>

### Examples

List projects:

```
projects = gl.projects.list()
```

The API provides several filtering parameters for the listing methods:

- **archived**: if `True` only archived projects will be returned
- **visibility**: returns only projects with the specified visibility (can be `public`, `internal` or `private`)
- **search**: returns project matching the given pattern

Results can also be sorted using the following parameters:

- **order\_by**: sort using the given argument. Valid values are `id`, `name`, `path`, `created_at`, `updated_at` and `last_activity_at`. The default is to sort by `created_at`

- sort: sort order (asc or desc)

```
# List all projects (default 20)
projects = gl.projects.list(get_all=True)
# Archived projects
projects = gl.projects.list(archived=1)
# Limit to projects with a defined visibility
projects = gl.projects.list(visibility='public')

# List owned projects
projects = gl.projects.list(owned=True)

# List starred projects
projects = gl.projects.list(starred=True)

# Search projects
projects = gl.projects.list(search='keyword')
```

---

**Note:** To list the starred projects of another user, see the *Users API docs*.

---

---

**Note:** Fetching a list of projects, doesn't include all attributes of all projects. To retrieve all attributes, you'll need to fetch a single project

---

Get a single project:

```
# Get a project by ID
project_id = 851
project = gl.projects.get(project_id)

# Get a project by name with namespace
project_name_with_namespace = "namespace/project_name"
project = gl.projects.get(project_name_with_namespace)
```

Create a project:

```
project = gl.projects.create({'name': 'project1'})
```

Create a project for a user (admin only):

```
alice = gl.users.list(username='alice')[0]
user_project = alice.projects.create({'name': 'project'})
user_projects = alice.projects.list()
```

Create a project in a group:

```
# You need to get the id of the group, then use the namespace_id attribute
# to create the group
group_id = gl.groups.list(search='my-group')[0].id
project = gl.projects.create({'name': 'myrepo', 'namespace_id': group_id})
```

List a project's groups:

```
# Get a list of ancestor/parent groups for a project.
groups = project.groups.list()
```

Update a project:

```
project.snippets_enabled = 1
project.save()
```

Set the avatar image for a project:

```
# the avatar image can be passed as data (content of the file) or as a file
# object opened in binary mode
project.avatar = open('path/to/file.png', 'rb')
project.save()
```

Delete a project:

```
gl.projects.delete(project_id)
# or
project.delete()
```

Restore a project marked for deletion (Premium only):

```
project.restore()
```

Fork a project:

```
fork = project.forks.create({})

# fork to a specific namespace
fork = project.forks.create({'namespace': 'myteam'})
```

Get a list of forks for the project:

```
forks = project.forks.list()
```

Create/delete a fork relation between projects (requires admin permissions):

```
project.create_fork_relation(source_project.id)
project.delete_fork_relation()
```

Get languages used in the project with percentage value:

```
languages = project.languages()
```

Star/unstar a project:

```
project.star()
project.unstar()
```

Archive/unarchive a project:

```
project.archive()
project.unarchive()
```

Start the housekeeping job:

```
project.housekeeping()
```

List the repository tree:

```
# list the content of the root directory for the default branch
items = project.repository_tree()

# list the content of a subdirectory on a specific branch
items = project.repository_tree(path='docs', ref='branch1')
```

Get the content and metadata of a file for a commit, using a blob sha:

```
items = project.repository_tree(path='docs', ref='branch1')
file_info = p.repository_blob(items[0]['id'])
content = base64.b64decode(file_info['content'])
size = file_info['size']
```

Update a project submodule:

```
items = project.update_submodule(
    submodule="foo/bar",
    branch="main",
    commit_sha="4c3674f66071e30b3311dac9b9ccc90502a72664",
    commit_message="Message", # optional
)
```

Get the repository archive:

```
tgz = project.repository_archive()

# get the archive for a branch/tag/commit
tgz = project.repository_archive(sha='4567abc')

# get the archive in a different format
zip = project.repository_archive(format='zip')
```

---

**Note:** For the formats available, refer to <https://docs.gitlab.com/ce/api/repositories.html#get-file-archive>

---

**Warning:** Archives are entirely stored in memory unless you use the streaming feature. See *the artifacts example*.

Get the content of a file using the blob id:

```
# find the id for the blob (simple search)
id = [d['id'] for d in p.repository_tree() if d['name'] == 'README.rst'][0]

# get the content
file_content = p.repository_raw_blob(id)
```

**Warning:** Blobs are entirely stored in memory unless you use the streaming feature. See *the artifacts example*.

Get a snapshot of the repository:

```
tar_file = project.snapshot()
```

**Warning:** Snapshots are entirely stored in memory unless you use the streaming feature. See [the artifacts example](#).

Compare two branches, tags or commits:

```
result = project.repository_compare('main', 'branch1')

# get the commits
for commit in result['commits']:
    print(commit)

# get the diffs
for file_diff in result['diffs']:
    print(file_diff)
```

Get the merge base for two or more branches, tags or commits:

```
commit = project.repository_merge_base(['main', 'v1.2.3', 'bd1324e2f'])
```

Get a list of contributors for the repository:

```
contributors = project.repository_contributors()
```

Get a list of users for the repository:

```
users = p.users.list()

# search for users
users = p.users.list(search='pattern')
```

Start the pull mirroring process (EE edition):

```
project.mirror_pull()
```

Get a project's pull mirror details (EE edition):

```
mirror_pull_details = project.mirror_pull_details()
```

## Import / Export

You can export projects from gitlab, and re-import them to create new projects or overwrite existing ones.

## Reference

- v4 API:
  - `gitlab.v4.objects.ProjectExport`
  - `gitlab.v4.objects.ProjectExportManager`
  - `gitlab.v4.objects.Project.exports`
  - `gitlab.v4.objects.ProjectImport`
  - `gitlab.v4.objects.ProjectImportManager`
  - `gitlab.v4.objects.Project.imports`
  - `gitlab.v4.objects.ProjectManager.import_project`
- GitLab API: [https://docs.gitlab.com/ce/api/project\\_import\\_export.html](https://docs.gitlab.com/ce/api/project_import_export.html)

## Examples

A project export is an asynchronous operation. To retrieve the archive generated by GitLab you need to:

1. Create an export using the API
2. Wait for the export to be done
3. Download the result

```
# Create the export
p = gl.projects.get(my_project)
export = p.exports.create()

# Wait for the 'finished' status
export.refresh()
while export.export_status != 'finished':
    time.sleep(1)
    export.refresh()

# Download the result
with open('/tmp/export.tgz', 'wb') as f:
    export.download(streamed=True, action=f.write)
```

You can export and upload a project to an external URL (see upstream documentation for more details):

```
project.exports.create(
    {
        "upload":
            {
                "url": "http://localhost:8080",
                "method": "POST"
            }
    }
)
```

You can also get the status of an existing export, regardless of whether it was created via the API or the Web UI:

```
project = gl.projects.get(my_project)
```

```
# Gets the current export status
export = project.exports.get()
```

Import the project into the current user's namespace:

```
with open('/tmp/export.tgz', 'rb') as f:
    output = gl.projects.import_project(
        f, path='my_new_project', name='My New Project'
    )

# Get a ProjectImport object to track the import status
project_import = gl.projects.get(output['id'], lazy=True).imports.get()
while project_import.import_status != 'finished':
    time.sleep(1)
    project_import.refresh()
```

Import the project into a namespace and override parameters:

```
with open('/tmp/export.tgz', 'rb') as f:
    output = gl.projects.import_project(
        f,
        path='my_new_project',
        name='My New Project',
        namespace='my-group',
        override_params={'visibility': 'private'},
    )
```

Import the project using file stored on a remote URL:

```
output = gl.projects.remote_import(
    url="https://whatever.com/url/file.tar.gz",
    path="my_new_remote_project",
    name="My New Remote Project",
    namespace="my-group",
    override_params={'visibility': 'private'},
)
```

Import the project using file stored on AWS S3:

```
output = gl.projects.remote_import_s3(
    path="my_new_remote_project",
    region="aws-region",
    bucket_name="aws-bucket-name",
    file_key="aws-file-key",
    access_key_id="aws-access-key-id",
    secret_access_key="secret-access-key",
    name="My New Remote Project",
    namespace="my-group",
    override_params={'visibility': 'private'},
)
```

## Project custom attributes

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectCustomAttribute`
  - `gitlab.v4.objects.ProjectCustomAttributeManager`
  - `gitlab.v4.objects.Project.customattributes`
- GitLab API: [https://docs.gitlab.com/ce/api/custom\\_attributes.html](https://docs.gitlab.com/ce/api/custom_attributes.html)

### Examples

List custom attributes for a project:

```
attrs = project.customattributes.list()
```

Get a custom attribute for a project:

```
attr = project.customattributes.get(attr_key)
```

Set (create or update) a custom attribute for a project:

```
attr = project.customattributes.set(attr_key, attr_value)
```

Delete a custom attribute for a project:

```
attr.delete()  
# or  
project.customattributes.delete(attr_key)
```

Search projects by custom attribute:

```
project.customattributes.set('type', 'internal')  
gl.projects.list(custom_attributes={'type': 'internal'})
```

## Project files

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectFile`
  - `gitlab.v4.objects.ProjectFileManager`
  - `gitlab.v4.objects.Project.files`
- GitLab API: [https://docs.gitlab.com/ce/api/repository\\_files.html](https://docs.gitlab.com/ce/api/repository_files.html)



## Examples

Get a file:

```
f = project.files.get(file_path='README.rst', ref='main')

# get the base64 encoded content
print(f.content)

# get the decoded content
print(f.decode())
```

Get file details from headers, without fetching its entire content:

```
headers = project.files.head('README.rst', ref='main')

# Get the file size:
# For a full list of headers returned, see upstream documentation.
# https://docs.gitlab.com/ee/api/repository_files.html#get-file-from-repository
print(headers["X-Gitlab-Size"])
```

Get a raw file:

```
raw_content = project.files.raw(file_path='README.rst', ref='main')
print(raw_content)
with open('/tmp/raw-download.txt', 'wb') as f:
    project.files.raw(file_path='README.rst', ref='main', streamed=True, action=f.write)
```

Create a new file:

```
f = project.files.create({'file_path': 'testfile.txt',
                          'branch': 'main',
                          'content': file_content,
                          'author_email': 'test@example.com',
                          'author_name': 'yourname',
                          'commit_message': 'Create testfile'})
```

Update a file. The entire content must be uploaded, as plain text or as base64 encoded text:

```
f.content = 'new content'
f.save(branch='main', commit_message='Update testfile')

# or for binary data
# Note: decode() is required with python 3 for data serialization. You can omit
# it with python 2
f.content = base64.b64encode(open('image.png').read()).decode()
f.save(branch='main', commit_message='Update testfile', encoding='base64')
```

Delete a file:

```
f.delete(commit_message='Delete testfile', branch='main')
# or
project.files.delete(file_path='testfile.txt', commit_message='Delete testfile', branch=
    ↪ 'main')
```

Get file blame:

```
b = project.files.blame(file_path='README.rst', ref='main')
```

## Project tags

### Reference

- v4 API:
  - *gitlab.v4.objects.ProjectTag*
  - *gitlab.v4.objects.ProjectTagManager*
  - *gitlab.v4.objects.Project.tags*
- GitLab API: <https://docs.gitlab.com/ce/api/tags.html>

### Examples

List the project tags:

```
tags = project.tags.list()
```

Get a tag:

```
tag = project.tags.get('1.0')
```

Create a tag:

```
tag = project.tags.create({'tag_name': '1.0', 'ref': 'main'})
```

Delete a tag:

```
project.tags.delete('1.0')  
# or  
tag.delete()
```

### Project snippets

The snippet visibility can be defined using the following constants:

- `gitlab.const.Visibility.PRIVATE`
- `gitlab.const.Visibility.INTERNAL`
- `gitlab.const.Visibility.PUBLIC`

## Reference

- v4 API:
  - `gitlab.v4.objects.ProjectSnippet`
  - `gitlab.v4.objects.ProjectSnippetManager`
  - `gitlab.v4.objects.Project.files`
- GitLab API: [https://docs.gitlab.com/ce/api/project\\_snippets.html](https://docs.gitlab.com/ce/api/project_snippets.html)

## Examples

List the project snippets:

```
snippets = project.snippets.list()
```

Get a snippet:

```
snippet = project.snippets.get(snippet_id)
```

Get the content of a snippet:

```
print(snippet.content())
```

**Warning:** The snippet content is entirely stored in memory unless you use the streaming feature. See *the artifacts example*.

Create a snippet:

```
snippet = project.snippets.create({'title': 'sample 1',
                                   'files': [{
                                       'file_path': 'foo.py',
                                       'content': 'import gitlab'
                                   }],
                                   'visibility_level':
                                   gitlab.const.Visibility.PRIVATE})
```

Update a snippet:

```
snippet.code = 'import gitlab\nimport whatever'
snippet.save
```

Delete a snippet:

```
project.snippets.delete(snippet_id)
# or
snippet.delete()
```

Get user agent detail (admin only):

```
detail = snippet.user_agent_detail()
```

## Notes

See *Notes*.

## Project members

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectMember`
  - `gitlab.v4.objects.ProjectMemberManager`
  - `gitlab.v4.objects.ProjectMemberAllManager`
  - `gitlab.v4.objects.Project.members`
  - `gitlab.v4.objects.Project.members_all`
- GitLab API: <https://docs.gitlab.com/ce/api/members.html>

### Examples

List only direct project members:

```
members = project.members.list()
```

List the project members recursively (including inherited members through ancestor groups):

```
members = project.members_all.list(get_all=True)
```

Search project members matching a query string:

```
members = project.members.list(query='bar')
```

Get only direct project member:

```
member = project.members.get(user_id)
```

Get a member of a project, including members inherited through ancestor groups:

```
members = project.members_all.get(member_id)
```

Add a project member:

```
member = project.members.create({'user_id': user.id, 'access_level':  
                                gitlab.const.AccessLevel.DEVELOPER})
```

Modify a project member (change the access level):

```
member.access_level = gitlab.const.AccessLevel.MAINTAINER  
member.save()
```

Remove a member from the project team:

```
project.members.delete(user.id)
# or
member.delete()
```

Share/unshare the project with a group:

```
project.share(group.id, gitlab.const.AccessLevel.DEVELOPER)
project.unshare(group.id)
```

## Project hooks

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectHook`
  - `gitlab.v4.objects.ProjectHookManager`
  - `gitlab.v4.objects.Project.hooks`
- GitLab API: <https://docs.gitlab.com/ce/api/projects.html#hooks>

### Examples

List the project hooks:

```
hooks = project.hooks.list()
```

Get a project hook:

```
hook = project.hooks.get(hook_id)
```

Create a project hook:

```
hook = project.hooks.create({'url': 'http://my/action/url', 'push_events': 1})
```

Update a project hook:

```
hook.push_events = 0
hook.save()
```

Delete a project hook:

```
project.hooks.delete(hook_id)
# or
hook.delete()
```

## Project Integrations

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectIntegration`
  - `gitlab.v4.objects.ProjectIntegrationManager`
  - `gitlab.v4.objects.Project.integrations`
- GitLab API: <https://docs.gitlab.com/ce/api/integrations.html>

### Examples

**Danger:** Since GitLab 13.12, `get()` calls to project integrations return a 404 Not Found response until they have been activated the first time.

To avoid this, we recommend using `lazy=True` to prevent making the initial call when activating new integrations unless they have previously already been activated.

Configure and enable an integration for the first time:

```
integration = project.integrations.get('asana', lazy=True)

integration.api_key = 'randomkey'
integration.save()
```

Get an existing integration:

```
integration = project.integrations.get('asana')
# display its status (enabled/disabled)
print(integration.active)
```

List active project integrations:

```
integration = project.integrations.list()
```

List the code names of available integrations (doesn't return objects):

```
integrations = project.integrations.available()
```

Disable an integration:

```
integration.delete()
```

## File uploads

### Reference

- v4 API:
  - `gitlab.v4.objects.Project.upload`
- Gitlab API: <https://docs.gitlab.com/ce/api/projects.html#upload-a-file>

### Examples

Upload a file into a project using a filesystem path:

```
project.upload("filename.txt", filepath="/some/path/filename.txt")
```

Upload a file into a project without a filesystem path:

```
project.upload("filename.txt", filedata="Raw data")
```

Upload a file and comment on an issue using the uploaded file's markdown:

```
uploaded_file = project.upload("filename.txt", filedata="data")
issue = project.issues.get(issue_id)
issue.notes.create({
    "body": "See the attached file: {}".format(uploaded_file["markdown"])
})
```

Upload a file and comment on an issue while using custom markdown to reference the uploaded file:

```
uploaded_file = project.upload("filename.txt", filedata="data")
issue = project.issues.get(issue_id)
issue.notes.create({
    "body": "See the [attached file]({})".format(uploaded_file["url"])
})
```

## Project push rules

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectPushRules`
  - `gitlab.v4.objects.ProjectPushRulesManager`
  - `gitlab.v4.objects.Project.pushrules`
- GitLab API: <https://docs.gitlab.com/ee/api/projects.html#push-rules>

## Examples

Create project push rules (at least one rule is necessary):

```
project.pushrules.create({'deny_delete_tag': True})
```

Get project push rules:

```
pr = project.pushrules.get()
```

Edit project push rules:

```
pr.branch_name_regex = '^((main|develop|support-\d+|release-\d+\.|hotfix-\d+|feature-\d+))$'
pr.save()
```

Delete project push rules:

```
pr.delete()
```

## Project protected tags

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectProtectedTag`
  - `gitlab.v4.objects.ProjectProtectedTagManager`
  - `gitlab.v4.objects.Project.protectedtags`
- GitLab API: [https://docs.gitlab.com/ce/api/protected\\_tags.html](https://docs.gitlab.com/ce/api/protected_tags.html)

## Examples

Get a list of protected tags from a project:

```
protected_tags = project.protectedtags.list()
```

Get a single protected tag or wildcard protected tag:

```
protected_tag = project.protectedtags.get('v*')
```

Protect a single repository tag or several project repository tags using a wildcard protected tag:

```
project.protectedtags.create({'name': 'v*', 'create_access_level': '40'})
```

Unprotect the given protected tag or wildcard protected tag.:

```
protected_tag.delete()
```



## Additional project statistics

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectAdditionalStatistics`
  - `gitlab.v4.objects.ProjectAdditionalStatisticsManager`
  - `gitlab.v4.objects.Project.additionalstatistics`
- GitLab API: [https://docs.gitlab.com/ce/api/project\\_statistics.html](https://docs.gitlab.com/ce/api/project_statistics.html)

### Examples

Get all additional statistics of a project:

```
statistics = project.additionalstatistics.get()
```

Get total fetches in last 30 days of a project:

```
total_fetches = project.additionalstatistics.get().fetches['total']
```

## Project storage

This endpoint requires admin access.

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectStorage`
  - `gitlab.v4.objects.ProjectStorageManager`
  - `gitlab.v4.objects.Project.storage`
- GitLab API: <https://docs.gitlab.com/ee/api/projects.html#get-the-path-to-repository-storage>

### Examples

Get the repository storage details for a project:

```
storage = project.storage.get()
```

Get the repository storage disk path:

```
disk_path = project.storage.get().disk_path
```

## 7.5.42 Project Access Tokens

Get a list of project access tokens

### References

- v4 API:
  - `gitlab.v4.objects.ProjectAccessToken`
  - `gitlab.v4.objects.ProjectAccessTokenManager`
  - `gitlab.Gitlab.project_access_tokens`
- GitLab API: [https://docs.gitlab.com/ee/api/project\\_access\\_tokens.html](https://docs.gitlab.com/ee/api/project_access_tokens.html)

### Examples

List project access tokens:

```
access_tokens = gl.projects.get(1, lazy=True).access_tokens.list()
print(access_tokens[0].name)
```

Get a project access token by id:

```
token = project.access_tokens.get(123)
print(token.name)
```

Create project access token:

```
access_token = gl.projects.get(1).access_tokens.create({"name": "test", "scopes": ["api↵"], "expires_at": "2023-06-06"})
```

Revoke a project access token:

```
gl.projects.get(1).access_tokens.delete(42)
# or
access_token.delete()
```

Rotate a project access token and retrieve its new value:

```
token = project.access_tokens.get(42, lazy=True)
token.rotate()
print(token.token)
# or directly using a token ID
new_token = project.access_tokens.rotate(42)
print(new_token.token)
```

### 7.5.43 Protected branches

You can define a list of protected branch names on a repository. Names can use wildcards (\*).

#### References

- v4 API:
  - `gitlab.v4.objects.ProjectProtectedBranch`
  - `gitlab.v4.objects.ProjectProtectedBranchManager`
  - `gitlab.v4.objects.Project.protectedbranches`
- GitLab API: [https://docs.gitlab.com/ce/api/protected\\_branches.html#protected-branches-api](https://docs.gitlab.com/ce/api/protected_branches.html#protected-branches-api)

#### Examples

Get the list of protected branches for a project:

```
p_branches = project.protectedbranches.list()
```

Get a single protected branch:

```
p_branch = project.protectedbranches.get('main')
```

Update a protected branch:

```
p_branch.allow_force_push = True
p_branch.save()
```

Create a protected branch:

```
p_branch = project.protectedbranches.create({
    'name': '*-stable',
    'merge_access_level': gitlab.const.AccessLevel.DEVELOPER,
    'push_access_level': gitlab.const.AccessLevel.MAINTAINER
})
```

Create a protected branch with more granular access control:

```
p_branch = project.protectedbranches.create({
    'name': '*-stable',
    'allowed_to_push': [{"user_id": 99}, {"user_id": 98}],
    'allowed_to_merge': [{"group_id": 653}],
    'allowed_to_unprotect': [{"access_level": gitlab.const.AccessLevel.MAINTAINER}]
})
```

Delete a protected branch:

```
project.protectedbranches.delete('*-stable')
# or
p_branch.delete()
```

## 7.5.44 Protected environments

You can list and manage protected environments in a project.

### References

- v4 API:
  - *gitlab.v4.objects.ProjectProtectedEnvironment*
  - *gitlab.v4.objects.ProjectProtectedEnvironmentManager*
  - *gitlab.v4.objects.Project.protected\_environment*
- GitLab API: [https://docs.gitlab.com/ee/api/protected\\_environments.html](https://docs.gitlab.com/ee/api/protected_environments.html)

### Examples

Get the list of protected environments for a project:

```
p_environments = project.protected_environments.list()
```

Get a single protected environment:

```
p_environments = project.protected_environments.get('production')
```

Protect an existing environment:

```
p_environment = project.protected_environments.create(  
    {  
        'name': 'production',  
        'deploy_access_levels': [  
            {'access_level': 40}  
        ],  
    }  
)
```

Unprotect a protected environment:

```
p_environment = project.protected_environments.delete('production')  
# or  
p_environment.delete()
```

## 7.5.45 Releases

### Project releases

#### Reference

- v4 API:
  - *gitlab.v4.objects.ProjectRelease*
  - *gitlab.v4.objects.ProjectReleaseManager*

- `gitlab.v4.objects.Project.releases`
- Gitlab API: <https://docs.gitlab.com/ee/api/releases/index.html>

## Examples

Get a list of releases from a project:

```
project = gl.projects.get(project_id, lazy=True)
release = project.releases.list()
```

Get a single release:

```
release = project.releases.get('v1.2.3')
```

Edit a release:

```
release.name = "Demo Release"
release.description = "release notes go here"
release.save()
```

Create a release for a project tag:

```
release = project.releases.create({'name': 'Demo Release', 'tag_name': 'v1.2.3',
↪ 'description': 'release notes go here'})
```

Delete a release:

```
# via its tag name from project attributes
release = project.releases.delete('v1.2.3')

# delete object directly
release.delete()
```

**Note:** The Releases API is one of the few working with `CI_JOB_TOKEN`, but the project can't be fetched with the token. Thus use *lazy* for the project as in the above example.

Also be aware that most of the capabilities of the endpoint were not accessible with `CI_JOB_TOKEN` until Gitlab version 14.5.

## Project release links

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectReleaseLink`
  - `gitlab.v4.objects.ProjectReleaseLinkManager`
  - `gitlab.v4.objects.ProjectRelease.links`
- Gitlab API: <https://docs.gitlab.com/ee/api/releases/links.html>

## Examples

Get a list of releases from a project:

```
links = release.links.list()
```

Get a single release link:

```
link = release.links.get(1)
```

Create a release link for a release:

```
link = release.links.create({"url": "https://example.com/asset", "name": "asset"})
```

Delete a release link:

```
# via its ID from release attributes
release.links.delete(1)

# delete object directly
link.delete()
```

## 7.5.46 Runners

Runners are external processes used to run CI jobs. They are deployed by the administrator and registered to the GitLab instance.

Shared runners are available for all projects. Specific runners are enabled for a list of projects.

### Global runners (admin)

## Reference

- v4 API:
  - *gitlab.v4.objects.Runner*
  - *gitlab.v4.objects.RunnerManager*
  - *gitlab.Gitlab.runners*
  - *gitlab.v4.objects.RunnerAll*
  - *gitlab.v4.objects.RunnerAllManager*
  - *gitlab.Gitlab.runners\_all*
- GitLab API: <https://docs.gitlab.com/ce/api/runners.html>

## Examples

Use the `runners.list()` and `runners_all.list()` methods to list runners. `runners.list()` - Get a list of specific runners available to the user `runners_all.list()` - Get a list of all runners in the GitLab instance (specific and shared). Access is restricted to users with administrator access.

Both methods accept a `scope` parameter to filter the list. Allowed values for this parameter are:

- `active`
- `paused`
- `online`
- `specific` (`runners_all.list()` only)
- `shared` (`runners_all.list()` only)

**Note:** The returned objects hold minimal information about the runners. Use the `get()` method to retrieve detail about a runner.

Runners returned via `runners_all.list()` also cannot be manipulated directly. You will need to use the `get()` method to create an editable object.

```
# List owned runners
runners = gl.runners.list()

# List owned runners with a filter
runners = gl.runners.list(scope='active')

# List all runners in the GitLab instance (specific and shared), using a filter
runners = gl.runners_all.list(scope='paused')
```

Get a runner's detail:

```
runner = gl.runners.get(runner_id)
```

Register a new runner:

```
runner = gl.runners.create({'token': secret_token})
```

**Note:** A new runner registration workflow has been introduced since GitLab 16.0. This new workflow comes with a new API endpoint to create runners, which does not use registration tokens.

The new endpoint can be called using `gl.user.runners.create()` after authenticating with `gl.auth()`.

Update a runner:

```
runner = gl.runners.get(runner_id)
runner.tag_list.append('new_tag')
runner.save()
```

Remove a runner:

```
gl.runners.delete(runner_id)
# or
runner.delete()
```

Remove a runner by its authentication token:

```
gl.runners.delete(token="runner-auth-token")
```

Verify a registered runner token:

```
try:
    gl.runners.verify(runner_token)
    print("Valid token")
except GitlabVerifyError:
    print("Invalid token")
```

## Project/Group runners

### Reference

- v4 API:
  - *gitlab.v4.objects.ProjectRunner*
  - *gitlab.v4.objects.ProjectRunnerManager*
  - *gitlab.v4.objects.Project.runners*
  - *gitlab.v4.objects.GroupRunner*
  - *gitlab.v4.objects.GroupRunnerManager*
  - *gitlab.v4.objects.Group.runners*
- GitLab API: <https://docs.gitlab.com/ce/api/runners.html>

### Examples

List the runners for a project:

```
runners = project.runners.list()
```

Enable a specific runner for a project:

```
p_runner = project.runners.create({'runner_id': runner.id})
```

Disable a specific runner for a project:

```
project.runners.delete(runner.id)
```



## Runner jobs

### Reference

- v4 API:
  - `gitlab.v4.objects.RunnerJob`
  - `gitlab.v4.objects.RunnerJobManager`
  - `gitlab.v4.objects.Runner.jobs`
- GitLab API: <https://docs.gitlab.com/ce/api/runners.html>

### Examples

List for jobs for a runner:

```
jobs = runner.jobs.list()
```

Filter the list using the jobs status:

```
# status can be 'running', 'success', 'failed' or 'canceled'
active_jobs = runner.jobs.list(status='running')
```

## 7.5.47 Project Remote Mirrors

Remote Mirrors allow you to set up push mirroring for a project.

### References

- v4 API:
  - `gitlab.v4.objects.ProjectRemoteMirror`
  - `gitlab.v4.objects.ProjectRemoteMirrorManager`
  - `gitlab.v4.objects.Project.remote_mirrors`
- GitLab API: [https://docs.gitlab.com/ce/api/remote\\_mirrors.html](https://docs.gitlab.com/ce/api/remote_mirrors.html)

### Examples

Get the list of a project's remote mirrors:

```
mirrors = project.remote_mirrors.list()
```

Create (and enable) a remote mirror for a project:

```
mirror = project.remote_mirrors.create({'url': 'https://gitlab.com/example.git',
                                         'enabled': True})
```

Update an existing remote mirror's attributes:

```
mirror.enabled = False
mirror.only_protected_branches = True
mirror.save()
```

Delete an existing remote mirror:

```
mirror.delete()
```

## 7.5.48 Registry Repositories

### References

- v4 API:
  - *gitlab.v4.objects.ProjectRegistryRepository*
  - *gitlab.v4.objects.ProjectRegistryRepositoryManager*
  - *gitlab.v4.objects.Project.repositories*
- Gitlab API: [https://docs.gitlab.com/ce/api/container\\_registry.html](https://docs.gitlab.com/ce/api/container_registry.html)

### Examples

Get the list of container registry repositories associated with the project:

```
repositories = project.repositories.list()
```

Get the list of all project container registry repositories in a group:

```
repositories = group.registry_repositories.list()
```

Delete repository:

```
project.repositories.delete(id=x)
# or
repository = repositories.pop()
repository.delete()
```

## 7.5.49 Registry Repository Tags

### References

- v4 API:
  - *gitlab.v4.objects.ProjectRegistryTag*
  - *gitlab.v4.objects.ProjectRegistryTagManager*
  - *gitlab.v4.objects.Repository.tags*
- Gitlab API: [https://docs.gitlab.com/ce/api/container\\_registry.html](https://docs.gitlab.com/ce/api/container_registry.html)

## Examples

Get the list of repository tags in given registry:

```
repositories = project.repositories.list()
repository = repositories.pop()
tags = repository.tags.list()
```

Get specific tag:

```
repository.tags.get(id=tag_name)
```

Delete tag:

```
repository.tags.delete(id=tag_name)
# or
tag = repository.tags.get(id=tag_name)
tag.delete()
```

Delete tag in bulk:

```
repository.tags.delete_in_bulk(keep_n=1)
# or
repository.tags.delete_in_bulk(older_than="1m")
# or
repository.tags.delete_in_bulk(name_regex="v.+ ", keep_n=2)
```

---

**Note:** Delete in bulk is asynchronous operation and may take a while. Refer to: [https://docs.gitlab.com/ee/api/container\\_registry.html#delete-repository-tags-in-bulk](https://docs.gitlab.com/ee/api/container_registry.html#delete-repository-tags-in-bulk)

---

## 7.5.50 Resource Groups

### Reference

- v4 API:
  - *gitlab.v4.objects.ProjectResourceGroup*
  - *gitlab.v4.objects.ProjectResourceGroupManager*
  - *gitlab.v4.objects.Project.resource\_groups*
  - *gitlab.v4.objects.ProjectResourceGroupUpcomingJob*
  - *gitlab.v4.objects.ProjectResourceGroupUpcomingJobManager*
  - *gitlab.v4.objects.ProjectResourceGroup.upcoming\_jobs*
- Gitlab API: [https://docs.gitlab.com/ee/api/resource\\_groups.html](https://docs.gitlab.com/ee/api/resource_groups.html)

## Examples

List resource groups for a project:

```
project = gl.projects.get(project_id, lazy=True)
resource_group = project.resource_groups.list()
```

Get a single resource group:

```
resource_group = project.resource_groups.get("production")
```

Edit a resource group:

```
resource_group.process_mode = "oldest_first"
resource_group.save()
```

List upcoming jobs for a resource group:

```
upcoming_jobs = resource_group.upcoming_jobs.list()
```

## 7.5.51 Search API

You can search for resources at the top level, in a project or in a group. Searches are based on a scope (issues, merge requests, and so on) and a search string. The following constants are provided to represent the possible scopes:

- Shared scopes (global, group and project):
  - `gitlab.const.SearchScope.PROJECTS`: projects
  - `gitlab.const.SearchScope.ISSUES`: issues
  - `gitlab.const.SearchScope.MERGE_REQUESTS`: merge\_requests
  - `gitlab.const.SearchScope.MILESTONES`: milestones
  - `gitlab.const.SearchScope.WIKI_BLOBS`: wiki\_blobs
  - `gitlab.const.SearchScope.COMMITS`: commits
  - `gitlab.const.SearchScope.BLOBS`: blobs
  - `gitlab.const.SearchScope.USERS`: users
- specific global scope:
  - `gitlab.const.SearchScope.GLOBAL_SNIPPET_TITLES`: snippet\_titles
- specific project scope:
  - `gitlab.const.SearchScope.PROJECT_NOTES`: notes

## Reference

- v4 API:
  - `gitlab.Gitlab.search`
  - `gitlab.v4.objects.Group.search`
  - `gitlab.v4.objects.Project.search`
- GitLab API: <https://docs.gitlab.com/ce/api/search.html>

## Examples

Search for issues matching a specific string:

```
# global search
gl.search(gitlab.const.SearchScope.ISSUES, 'regression')

# group search
group = gl.groups.get('mygroup')
group.search(gitlab.const.SearchScope.ISSUES, 'regression')

# project search
project = gl.projects.get('myproject')
project.search(gitlab.const.SearchScope.ISSUES, 'regression')
```

The `search()` methods implement the pagination support:

```
# get lists of 10 items, and start at page 2
gl.search(gitlab.const.SearchScope.ISSUES, search_str, page=2, per_page=10)

# get a generator that will automatically make required API calls for
# pagination
for item in gl.search(gitlab.const.SearchScope.ISSUES, search_str, iterator=True):
    do_something(item)
```

The search API doesn't return objects, but dicts. If you need to act on objects, you need to create them explicitly:

```
for item in gl.search(gitlab.const.SearchScope.ISSUES, search_str, iterator=True):
    issue_project = gl.projects.get(item['project_id'], lazy=True)
    issue = issue_project.issues.get(item['iid'])
    issue.state = 'closed'
    issue.save()
```

## 7.5.52 Secure Files

### secure files

### References

- v4 API:
  - `gitlab.v4.objects.ProjectSecureFile`

- `gitlab.v4.objects.ProjectSecureFileManager`
- `gitlab.v4.objects.Project.secure_files`
- GitLab API: [https://docs.gitlab.com/ee/api/secure\\_files.html](https://docs.gitlab.com/ee/api/secure_files.html)

## Examples

Get a project secure file:

```
secure_files = gl.projects.get(1, lazy=True).secure_files.get(1)
print(secure_files.name)
```

List project secure files:

```
secure_files = gl.projects.get(1, lazy=True).secure_files.list()
print(secure_files[0].name)
```

Create project secure file:

```
secure_file = gl.projects.get(1).secure_files.create({"name": "test", "file": "secure.txt"
↪ })
```

Download a project secure file:

```
content = secure_file.download()
print(content)
with open("/tmp/secure.txt", "wb") as f:
    secure_file.download(streamed=True, action=f.write)
```

Remove a project secure file:

```
gl.projects.get(1).secure_files.delete(1)
# or
secure_file.delete()
```

## 7.5.53 Settings

### Reference

- v4 API:
  - `gitlab.v4.objects.ApplicationSettings`
  - `gitlab.v4.objects.ApplicationSettingsManager`
  - `gitlab.Gitlab.settings`
- GitLab API: <https://docs.gitlab.com/ce/api/settings.html>

## Examples

Get the settings:

```
settings = gl.settings.get()
```

Update the settings:

```
settings.signin_enabled = False
settings.save()
```

## 7.5.54 Snippets

### Reference

- v4 API:
  - `gitlab.v4.objects.Snippet`
  - `gitlab.v4.objects.SnippetManager`
  - `gitlab.Gitlab.snippets`
- GitLab API: <https://docs.gitlab.com/ce/api/snippets.html>

## Examples

List snippets owned by the current user:

```
snippets = gl.snippets.list()
```

List the public snippets:

```
public_snippets = gl.snippets.public()
```

Get a snippet:

```
snippet = gl.snippets.get(snippet_id)
# get the content
content = snippet.content()
```

**Warning:** Blobs are entirely stored in memory unless you use the streaming feature. See *the artifacts example*.

Create a snippet:

```
snippet = gl.snippets.create({'title': 'snippet1',
                              'files': [{
                                  'file_path': 'foo.py',
                                  'content': 'import gitlab'
                              }]},
                              {})
```

Update the snippet attributes:

```
snippet.visibility_level = gitlab.const.Visibility.PUBLIC
snippet.save()
```

To update a snippet code you need to create a `ProjectSnippet` object:

```
snippet = gl.snippets.get(snippet_id)
project = gl.projects.get(snippet.project_id, lazy=True)
editable_snippet = project.snippets.get(snippet.id)
editable_snippet.code = new_snippet_content
editable_snippet.save()
```

Delete a snippet:

```
gl.snippets.delete(snippet_id)
# or
snippet.delete()
```

Get user agent detail (admin only):

```
detail = snippet.user_agent_detail()
```

## 7.5.55 Statistics

### Reference

- v4 API:
  - `gitlab.v4.objects.ApplicationStatistics`
  - `gitlab.v4.objects.ApplicationStatisticsManager`
  - `gitlab.Gitlab.statistics`
- GitLab API: <https://docs.gitlab.com/ee/api/statistics.html>

### Examples

Get the statistics:

```
statistics = gl.statistics.get()
```

## 7.5.56 System hooks

### Reference

- v4 API:
  - `gitlab.v4.objects.Hook`
  - `gitlab.v4.objects.HookManager`
  - `gitlab.Gitlab.hooks`
- GitLab API: [https://docs.gitlab.com/ce/api/system\\_hooks.html](https://docs.gitlab.com/ce/api/system_hooks.html)



## Examples

List the system hooks:

```
hooks = gl.hooks.list()
```

Create a system hook:

```
gl.hooks.get(hook_id)
```

Test a system hook. The returned object is not usable (it misses the hook ID):

```
hook = gl.hooks.create({'url': 'http://your.target.url'})
```

Delete a system hook:

```
gl.hooks.delete(hook_id)
# or
hook.delete()
```

## 7.5.57 Templates

You can request templates for different type of files:

- License files
- .gitignore files
- GitLab CI configuration files
- Dockerfiles

### License templates

#### Reference

- v4 API:
  - `gitlab.v4.objects.License`
  - `gitlab.v4.objects.LicenseManager`
  - `gitlab.Gitlab.licenses`
- GitLab API: <https://docs.gitlab.com/ce/api/templates/licenses.html>

## Examples

List known license templates:

```
licenses = gl.licenses.list()
```

Generate a license content for a project:

```
license = gl.licenses.get('apache-2.0', project='foobar', fullname='John Doe')
print(license.content)
```

## .gitignore templates

### Reference

- v4 API:
  - `gitlab.v4.objects.Gitignore`
  - `gitlab.v4.objects.GitignoreManager`
  - `gitlab.Gitlab.gitignores`
- GitLab API: <https://docs.gitlab.com/ce/api/templates/gitignores.html>

### Examples

List known gitignore templates:

```
gitignores = gl.gitignores.list()
```

Get a gitignore template:

```
gitignore = gl.gitignores.get('Python')
print(gitignore.content)
```

## GitLab CI templates

### Reference

- v4 API:
  - `gitlab.v4.objects.Gitlabciyaml`
  - `gitlab.v4.objects.GitlabciyamlManager`
  - `gitlab.Gitlab.gitlabciyaml`
- GitLab API: [https://docs.gitlab.com/ce/api/templates/gitlab\\_ci\\_ymls.html](https://docs.gitlab.com/ce/api/templates/gitlab_ci_ymls.html)

### Examples

List known GitLab CI templates:

```
gitlabciyaml = gl.gitlabciyaml.list()
```

Get a GitLab CI template:

```
gitlabciyaml = gl.gitlabciyaml.get('Pelican')
print(gitlabciyaml.content)
```

## Dockerfile templates

### Reference

- v4 API:
  - `gitlab.v4.objects.Dockerfile`
  - `gitlab.v4.objects.DockerfileManager`
  - `gitlab.Gitlab.gitlabciymls`
- GitLab API: Not documented.

### Examples

List known Dockerfile templates:

```
dockerfiles = gl.dockerfiles.list()
```

Get a Dockerfile template:

```
dockerfile = gl.dockerfiles.get('Python')
print(dockerfile.content)
```

## 7.5.58 Todos

### Reference

- v4 API:
  - `Todo`
  - `TodoManager`
  - `gitlab.Gitlab.todos`
- GitLab API: <https://docs.gitlab.com/ce/api/todos.html>

### Examples

List active todos:

```
todos = gl.todos.list()
```

You can filter the list using the following parameters:

- `action`: can be assigned, mentioned, build\_failed, marked, or approval\_required
- `author_id`
- `project_id`
- `state`: can be pending or done
- `type`: can be Issue or MergeRequest

For example:

```
todos = gl.todos.list(project_id=1)
todos = gl.todos.list(state='done', type='Issue')
```

Mark a todo as done:

```
todos = gl.todos.list(project_id=1)
todos[0].mark_as_done()
```

Mark all the todos as done:

```
gl.todos.mark_all_as_done()
```

## 7.5.59 Topics

Topics can be used to categorize projects and find similar new projects.

### Reference

- v4 API:
  - *gitlab.v4.objects.Topic*
  - *gitlab.v4.objects.TopicManager*
  - *gitlab.Gitlab.topics*
- GitLab API: <https://docs.gitlab.com/ce/api/topics.html>

This endpoint requires admin access for creating, updating and deleting objects.

### Examples

List project topics on the GitLab instance:

```
topics = gl.topics.list()
```

Get a specific topic by its ID:

```
topic = gl.topics.get(topic_id)
```

Create a new topic:

```
topic = gl.topics.create({"name": "my-topic", "title": "my title"})
```

Update a topic:

```
topic.description = "My new topic"
topic.save()

# or
gl.topics.update(topic_id, {"description": "My new topic"})
```

Delete a topic:

```
topic.delete()

# or
gl.topics.delete(topic_id)
```

Merge a source topic into a target topic:

```
gl.topics.merge(topic_id, target_topic_id)
```

## 7.5.60 Users and current user

The Gitlab API exposes user-related method that can be manipulated by admins only.

The currently logged-in user is also exposed.

### Users

#### References

- v4 API:
  - `gitlab.v4.objects.User`
  - `gitlab.v4.objects.UserManager`
  - `gitlab.Gitlab.users`
- GitLab API:
  - <https://docs.gitlab.com/ce/api/users.html>
  - <https://docs.gitlab.com/ee/api/projects.html#list-projects-starred-by-a-user>

### Examples

Get the list of users:

```
users = gl.users.list()
```

Search users whose username match a given string:

```
users = gl.users.list(search='foo')
```

Get a single user:

```
# by ID
user = gl.users.get(user_id)
# by username
user = gl.users.list(username='root')[0]
```

Create a user:

```
user = gl.users.create({'email': 'john@doe.com',
                        'password': 's3cur3s3cr3T',
                        'username': 'jdoe',
                        'name': 'John Doe'})
```

Update a user:

```
user.name = 'Real Name'
user.save()
```

Delete a user:

```
gl.users.delete(user_id)
# or
user.delete()
```

Block/Unblock a user:

```
user.block()
user.unblock()
```

Activate/Deactivate a user:

```
user.activate()
user.deactivate()
```

Ban/Unban a user:

```
user.ban()
user.unban()
```

Follow/Unfollow a user:

```
user.follow()
user.unfollow()
```

Set the avatar image for a user:

```
# the avatar image can be passed as data (content of the file) or as a file
# object opened in binary mode
user.avatar = open('path/to/file.png', 'rb')
user.save()
```

Set an external identity for a user:

```
user.provider = 'oauth2_generic'
user.extern_uid = '3'
user.save()
```

Delete an external identity by provider name:

```
user.identityproviders.delete('oauth2_generic')
```

Get the followers of a user

```
user.followers_users.list()
```

Get the followings of a user

```
user.following_users.list()
```

List a user's starred projects

```
user.starred_projects.list()
```

If the GitLab instance has new user account approval enabled some users may have `user.state == 'blocked_pending_approval'`. Administrators can approve and reject such users:

```
user.approve()
user.reject()
```

## User custom attributes

### References

- v4 API:
  - `gitlab.v4.objects.UserCustomAttribute`
  - `gitlab.v4.objects.UserCustomAttributeManager`
  - `gitlab.v4.objects.User.customattributes`
- GitLab API: [https://docs.gitlab.com/ce/api/custom\\_attributes.html](https://docs.gitlab.com/ce/api/custom_attributes.html)

### Examples

List custom attributes for a user:

```
attrs = user.customattributes.list()
```

Get a custom attribute for a user:

```
attr = user.customattributes.get(attr_key)
```

Set (create or update) a custom attribute for a user:

```
attr = user.customattributes.set(attr_key, attr_value)
```

Delete a custom attribute for a user:

```
attr.delete()
# or
user.customattributes.delete(attr_key)
```

Search users by custom attribute:

```
user.customattributes.set('role', 'QA')
gl.users.list(custom_attributes={'role': 'QA'})
```

## User impersonation tokens

### References

- v4 API:
  - `gitlab.v4.objects.UserImpersonationToken`
  - `gitlab.v4.objects.UserImpersonationTokenManager`
  - `gitlab.v4.objects.User.impersonationtokens`
- GitLab API: <https://docs.gitlab.com/ce/api/users.html#get-all-impersonation-tokens-of-a-user>

List impersonation tokens for a user:

```
i_t = user.impersonationtokens.list(state='active')
i_t = user.impersonationtokens.list(state='inactive')
```

Get an impersonation token for a user:

```
i_t = user.impersonationtokens.get(i_t_id)
```

Create and use an impersonation token for a user:

```
i_t = user.impersonationtokens.create({'name': 'token1', 'scopes': ['api']})
# use the token to create a new gitlab connection
user_gl = gitlab.Gitlab(gitlab_url, private_token=i_t.token)
```

Revoke (delete) an impersonation token for a user:

```
i_t.delete()
```

## User projects

### References

- v4 API:
  - `gitlab.v4.objects.UserProject`
  - `gitlab.v4.objects.UserProjectManager`
  - `gitlab.v4.objects.User.projects`
- GitLab API: <https://docs.gitlab.com/ee/api/projects.html#list-user-projects>

List visible projects in the user's namespace:

```
projects = user.projects.list()
```

---

**Note:** Only the projects in the user's namespace are returned. Projects owned by the user in any group or subgroups are not returned. An empty list is returned if a profile is set to private.

---



## User memberships

### References

- v4 API:
  - *gitlab.v4.objects.UserMembership*
  - *gitlab.v4.objects.UserMembershipManager*
  - *gitlab.v4.objects.User.memberships*
- GitLab API: <https://docs.gitlab.com/ee/api/users.html#user-memberships>

List direct memberships for a user:

```
memberships = user.memberships.list()
```

List only direct project memberships:

```
memberships = user.memberships.list(type='Project')
```

List only direct group memberships:

```
memberships = user.memberships.list(type='Namespace')
```

---

**Note:** This endpoint requires admin access.

---

## Current User

### References

- v4 API:
  - *gitlab.v4.objects.CurrentUser*
  - *gitlab.v4.objects.CurrentUserManager*
  - *gitlab.Gitlab.user*
- GitLab API: <https://docs.gitlab.com/ee/api/users.html>

### Examples

Get the current user:

```
gl.auth()  
current_user = gl.user
```

## GPG keys

### References

You can manipulate GPG keys for the current user and for the other users if you are admin.

- v4 API:
  - `gitlab.v4.objects.CurrentUserGPGKey`
  - `gitlab.v4.objects.CurrentUserGPGKeyManager`
  - `gitlab.v4.objects.CurrentUser.gpgkeys`
  - `gitlab.v4.objects.UserGPGKey`
  - `gitlab.v4.objects.UserGPGKeyManager`
  - `gitlab.v4.objects.User.gpgkeys`
- GitLab API: <https://docs.gitlab.com/ce/api/users.html#list-all-gpg-keys>

### Examples

List GPG keys for a user:

```
gpgkeys = user.gpgkeys.list()
```

Get a GPG gpgkey for a user:

```
gpgkey = user.gpgkeys.get(key_id)
```

Create a GPG gpgkey for a user:

```
# get the key with `gpg --export -a GPG_KEY_ID`  
k = user.gpgkeys.create({'key': public_key_content})
```

Delete a GPG gpgkey for a user:

```
user.gpgkeys.delete(key_id)  
# or  
gpgkey.delete()
```

## SSH keys

### References

You can manipulate SSH keys for the current user and for the other users if you are admin.

- v4 API:
  - `gitlab.v4.objects.CurrentUserKey`
  - `gitlab.v4.objects.CurrentUserKeyManager`
  - `gitlab.v4.objects.CurrentUser.keys`
  - `gitlab.v4.objects.UserKey`

- `gitlab.v4.objects.UserKeyManager`
- `gitlab.v4.objects.User.keys`
- GitLab API: <https://docs.gitlab.com/ce/api/users.html#list-ssh-keys>

## Examples

List SSH keys for a user:

```
keys = user.keys.list()
```

Create an SSH key for a user:

```
key = user.keys.create({'title': 'my_key',
                        'key': open('/home/me/.ssh/id_rsa.pub').read()})
```

Get an SSH key for a user by id:

```
key = user.keys.get(key_id)
```

Delete an SSH key for a user:

```
user.keys.delete(key_id)
# or
key.delete()
```

## Status

## References

You can manipulate the status for the current user and you can read the status of other users.

- v4 API:
  - `gitlab.v4.objects.CurrentUserStatus`
  - `gitlab.v4.objects.CurrentUserStatusManager`
  - `gitlab.v4.objects.CurrentUser.status`
  - `gitlab.v4.objects.UserStatus`
  - `gitlab.v4.objects.UserStatusManager`
  - `gitlab.v4.objects.User.status`
- GitLab API: <https://docs.gitlab.com/ce/api/users.html#user-status>

## Examples

Get current user status:

```
status = user.status.get()
```

Update the status for the current user:

```
status = user.status.get()
status.message = "message"
status.emoji = "thumbsup"
status.save()
```

Get the status of other users:

```
gl.users.get(1).status.get()
```

## Emails

### References

You can manipulate emails for the current user and for the other users if you are admin.

- v4 API:
  - `gitlab.v4.objects.CurrentUserEmail`
  - `gitlab.v4.objects.CurrentUserEmailManager`
  - `gitlab.v4.objects.CurrentUser.emails`
  - `gitlab.v4.objects.UserEmail`
  - `gitlab.v4.objects.UserEmailManager`
  - `gitlab.v4.objects.User.emails`
- GitLab API: <https://docs.gitlab.com/ce/api/users.html#list-emails>

## Examples

List emails for a user:

```
emails = user.emails.list()
```

Get an email for a user:

```
email = user.emails.get(email_id)
```

Create an email for a user:

```
k = user.emails.create({'email': 'foo@bar.com'})
```

Delete an email for a user:

```
user.emails.delete(email_id)
# or
email.delete()
```

## Users activities

### References

- admin only
- v4 API:
  - *gitlab.v4.objects.UserActivities*
  - *gitlab.v4.objects.UserActivitiesManager*
  - *gitlab.Gitlab.user\_activities*
- GitLab API: <https://docs.gitlab.com/ce/api/users.html#get-user-activities-admin-only>

### Examples

Get the users activities:

```
activities = gl.user_activities.list(
    query_parameters={'from': '2018-07-01'},
    get_all=True,
)
```

## Create new runner

### References

- New runner registration API endpoint (see [Migrating to the new runner registration workflow](#))
- v4 API:
  - *gitlab.v4.objects.CurrentUserRunner*
  - *gitlab.v4.objects.CurrentUserRunnerManager*
  - *gitlab.Gitlab.user.runners*
- GitLab API : <https://docs.gitlab.com/ee/api/users.html#create-a-runner>

## Examples

Create an instance-wide runner:

```
runner = gl.user.runners.create({
    "runner_type": "instance_type",
    "description": "My brand new runner",
    "paused": True,
    "locked": False,
    "run_untagged": True,
    "tag_list": ["linux", "docker", "testing"],
    "access_level": "not_protected"
})
```

Create a group runner:

```
runner = gl.user.runners.create({
    "runner_type": "group_type",
    "group_id": 12345678,
    "description": "My brand new runner",
    "paused": True,
    "locked": False,
    "run_untagged": True,
    "tag_list": ["linux", "docker", "testing"],
    "access_level": "not_protected"
})
```

Create a project runner:

```
runner = gl.user.runners.create({
    "runner_type": "project_type",
    "project_id": 987564321,
    "description": "My brand new runner",
    "paused": True,
    "locked": False,
    "run_untagged": True,
    "tag_list": ["linux", "docker", "testing"],
    "access_level": "not_protected"
})
```

### 7.5.61 CI/CD Variables

You can configure variables at the instance-level (admin only), or associate variables to projects and groups, to modify pipeline/job scripts behavior.

**Warning:** Please always follow GitLab's [rules for CI/CD variables](#), especially for values in masked variables. If you do not, your variables may silently fail to save.

## Instance-level variables

This endpoint requires admin access.

### Reference

- v4 API
  - `gitlab.v4.objects.Variable`
  - `gitlab.v4.objects.VariableManager`
  - `gitlab.Gitlab.variables`
- GitLab API
  - [https://docs.gitlab.com/ce/api/instance\\_level\\_ci\\_variables.html](https://docs.gitlab.com/ce/api/instance_level_ci_variables.html)

### Examples

List all instance variables:

```
variables = gl.variables.list()
```

Get an instance variable by key:

```
variable = gl.variables.get('key_name')
```

Create an instance variable:

```
variable = gl.variables.create({'key': 'key1', 'value': 'value1'})
```

Update a variable value:

```
variable.value = 'new_value'
variable.save()
```

Remove a variable:

```
gl.variables.delete('key_name')
# or
variable.delete()
```

## Projects and groups variables

### Reference

- v4 API
  - `gitlab.v4.objects.ProjectVariable`
  - `gitlab.v4.objects.ProjectVariableManager`
  - `gitlab.v4.objects.Project.variables`
  - `gitlab.v4.objects.GroupVariable`

- `gitlab.v4.objects.GroupVariableManager`
- `gitlab.v4.objects.Group.variables`
- GitLab API
  - [https://docs.gitlab.com/ce/api/instance\\_level\\_ci\\_variables.html](https://docs.gitlab.com/ce/api/instance_level_ci_variables.html)
  - [https://docs.gitlab.com/ce/api/project\\_level\\_variables.html](https://docs.gitlab.com/ce/api/project_level_variables.html)
  - [https://docs.gitlab.com/ce/api/group\\_level\\_variables.html](https://docs.gitlab.com/ce/api/group_level_variables.html)

## Examples

List variables:

```
p_variables = project.variables.list()
g_variables = group.variables.list()
```

Get a variable:

```
p_var = project.variables.get('key_name')
g_var = group.variables.get('key_name')
```

Create a variable:

```
var = project.variables.create({'key': 'key1', 'value': 'value1'})
var = group.variables.create({'key': 'key1', 'value': 'value1'})
```

Update a variable value:

```
var.value = 'new_value'
var.save()
# or
project.variables.update("key1", {"value": "new_value"})
```

Remove a variable:

```
project.variables.delete('key_name')
group.variables.delete('key_name')
# or
var.delete()
```

## 7.5.62 Sidekiq metrics

### Reference

- v4 API:
  - `gitlab.v4.objects.SidekiqManager`
  - `gitlab.Gitlab.sidekiq`
- GitLab API: [https://docs.gitlab.com/ce/api/sidekiq\\_metrics.html](https://docs.gitlab.com/ce/api/sidekiq_metrics.html)



## Examples

```
gl.sidekiq.queue_metrics()
gl.sidekiq.process_metrics()
gl.sidekiq.job_stats()
gl.sidekiq.compound_metrics()
```

## 7.5.63 Wiki pages

### References

- v4 API:
  - `gitlab.v4.objects.ProjectWiki`
  - `gitlab.v4.objects.ProjectWikiManager`
  - `gitlab.v4.objects.Project.wikis`
  - `gitlab.v4.objects.GroupWiki`
  - `gitlab.v4.objects.GroupWikiManager`
  - `gitlab.v4.objects.Group.wikis`
- GitLab API for Projects: <https://docs.gitlab.com/ce/api/wikis.html>
- GitLab API for Groups: [https://docs.gitlab.com/ee/api/group\\_wikis.html](https://docs.gitlab.com/ee/api/group_wikis.html)

### Examples

Get the list of wiki pages for a project. These do not contain the contents of the wiki page. You will need to call `get(slug)` to retrieve the content by accessing the content attribute:

```
pages = project.wikis.list()
```

Get the list of wiki pages for a group. These do not contain the contents of the wiki page. You will need to call `get(slug)` to retrieve the content by accessing the content attribute:

```
pages = group.wikis.list()
```

Get a single wiki page for a project:

```
page = project.wikis.get(page_slug)
```

Get a single wiki page for a group:

```
page = group.wikis.get(page_slug)
```

Get the contents of a wiki page:

```
print(page.content)
```

Create a wiki page on a project level:

```
page = project.wikis.create({'title': 'Wiki Page 1',
                             'content': open(a_file).read()})
```

Update a wiki page:

```
page.content = 'My new content'
page.save()
```

Delete a wiki page:

```
page.delete()
```

## File uploads

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectWiki.upload`
  - `gitlab.v4.objects.GroupWiki.upload`
- Gitlab API for Projects: <https://docs.gitlab.com/ee/api/wikis.html#upload-an-attachment-to-the-wiki-repository>
- Gitlab API for Groups: [https://docs.gitlab.com/ee/api/group\\_wikis.html#upload-an-attachment-to-the-wiki-repository](https://docs.gitlab.com/ee/api/group_wikis.html#upload-an-attachment-to-the-wiki-repository)

### Examples

Upload a file into a project wiki using a filesystem path:

```
page = project.wikis.get(page_slug)
page.upload("filename.txt", filepath="/some/path/filename.txt")
```

Upload a file into a project wiki with raw data:

```
page.upload("filename.txt", filedata="Raw data")
```

Upload a file into a group wiki using a filesystem path:

```
page = group.wikis.get(page_slug)
page.upload("filename.txt", filepath="/some/path/filename.txt")
```

Upload a file into a group wiki using raw data:

```
page.upload("filename.txt", filedata="Raw data")
```

## 7.5.64 Clusters (DEPRECATED)

**Warning:** Cluster support was deprecated in GitLab 14.5 and disabled by default as of GitLab 15.0

### Reference

- v4 API:
  - `gitlab.v4.objects.ProjectCluster`
  - `gitlab.v4.objects.ProjectClusterManager`
  - `gitlab.v4.objects.Project.clusters`
  - `gitlab.v4.objects.GroupCluster`
  - `gitlab.v4.objects.GroupClusterManager`
  - `gitlab.v4.objects.Group.clusters`
- GitLab API: [https://docs.gitlab.com/ee/api/project\\_clusters.html](https://docs.gitlab.com/ee/api/project_clusters.html)
- GitLab API: [https://docs.gitlab.com/ee/api/group\\_clusters.html](https://docs.gitlab.com/ee/api/group_clusters.html)

### Examples

List clusters for a project:

```
clusters = project.clusters.list()
```

Create an cluster for a project:

```
cluster = project.clusters.create(
{
    "name": "cluster1",
    "platform_kubernetes_attributes": {
        "api_url": "http://url",
        "token": "tokenval",
    },
})
```

Retrieve a specific cluster for a project:

```
cluster = project.clusters.get(cluster_id)
```

Update an cluster for a project:

```
cluster.platform_kubernetes_attributes = {"api_url": "http://newurl"}
cluster.save()
```

Delete an cluster for a project:

```
cluster = project.clusters.delete(cluster_id)
# or
cluster.delete()
```

List clusters for a group:

```
clusters = group.clusters.list()
```

Create an cluster for a group:

```
cluster = group.clusters.create(
{
    "name": "cluster1",
    "platform_kubernetes_attributes": {
        "api_url": "http://url",
        "token": "tokenval",
    },
})
```

Retrieve a specific cluster for a group:

```
cluster = group.clusters.get(cluster_id)
```

Update an cluster for a group:

```
cluster.platform_kubernetes_attributes = {"api_url": "http://newurl"}
cluster.save()
```

Delete an cluster for a group:

```
cluster = group.clusters.delete(cluster_id)
# or
cluster.delete()
```

## 7.6 API reference (gitlab package)

### 7.6.1 Module contents

Wrapper for the GitLab API.

```
class gitlab.Gitlab(url: str | None = None, private_token: str | None = None, oauth_token: str | None = None,
                    job_token: str | None = None, ssl_verify: bool | str = True, http_username: str | None =
                    None, http_password: str | None = None, timeout: float | None = None, api_version: str =
                    '4', per_page: int | None = None, pagination: str | None = None, order_by: str | None =
                    None, user_agent: str = 'python-gitlab/4.4.0', retry_transient_errors: bool = False,
                    keep_base_url: bool = False, **kwargs: Any)
```

Bases: object

Represents a GitLab server connection.

#### Parameters

- **url** (str | None) – The URL of the GitLab server (defaults to <https://gitlab.com>).
- **private\_token** (str | None) – The user private token
- **oauth\_token** (str | None) – An oauth token
- **job\_token** (str | None) – A CI job token

- **ssl\_verify** (*bool* / *str*) – Whether SSL certificates should be validated. If the value is a string, it is the path to a CA file used for certificate validation.
- **timeout** (*float* / *None*) – Timeout to use for requests to the GitLab server.
- **http\_username** (*str* / *None*) – Username for HTTP authentication
- **http\_password** (*str* / *None*) – Password for HTTP authentication
- **api\_version** (*str*) – Gitlab API version to use (support for 4 only)
- **pagination** (*str* / *None*) – Can be set to ‘keyset’ to use keyset pagination
- **order\_by** (*str* / *None*) – Set order\_by globally
- **user\_agent** (*str*) – A custom user agent to use for making HTTP requests.
- **retry\_transient\_errors** (*bool*) – Whether to retry after 500, 502, 503, 504 or 52x responses. Defaults to False.
- **keep\_base\_url** (*bool*) – keep user-provided base URL for pagination if it differs from response headers
- **session** (*requests.Session*) – HTTP Requests Session
- **backend** (*RequestsBackend*) – Backend that will be used to make http requests
- **per\_page** (*int* / *None*)
- **kwargs** (*Any*)

**property api\_url:** *str*

The computed API base URL.

**property api\_version:** *str*

The API version used (4 only).

**appearance**

See [ApplicationAppearanceManager](#)

**applications**

See [ApplicationManager](#)

**audit\_events**

See [AuditEventManager](#)

**auth()** → *None*

Performs an authentication using private token. Warns the user if a potentially misconfigured URL is detected on the client or server side.

The *user* attribute will hold a *gitlab.objects.CurrentUser* object on success.

**Return type**

*None*

**broadcastmessages**

See [BroadcastMessageManager](#)

**bulk\_import\_entities**

See [BulkImportAllEntityManager](#)

**bulk\_imports**

See [BulkImportManager](#)

**ci\_lint**

See [CiLintManager](#)

**deploykeys**

See [DeployKeyManager](#)

**deploytokens**

See [DeployTokenManager](#)

**dockerfiles**

See [DockerfileManager](#)

**enable\_debug**(*mask\_credentials: bool = True*) → None

**Parameters**

**mask\_credentials** (*bool*)

**Return type**

None

**events**

See [EventManager](#)

**features**

See [FeatureManager](#)

**classmethod from\_config**(*gitlab\_id: str | None = None, config\_files: List[str] | None = None, \*\*kwargs: Any*) → [Gitlab](#)

Create a Gitlab connection from configuration files.

**Parameters**

- **gitlab\_id** (*str | None*) – ID of the configuration section.
- **list[str]** (*config\_files*) – List of paths to configuration files.
- **config\_files** (*List[str] | None*)
- **kwargs** (*Any*)

**Return type**

[Gitlab](#)

**kwargs:**

session.requests.Session: Custom requests Session

**Returns**

A Gitlab connection.

**Raises**

[gitlab.config.GitlabDataError](#) – If the configuration is not correct.

**Parameters**

- **gitlab\_id** (*str | None*)
- **config\_files** (*List[str] | None*)
- **kwargs** (*Any*)

**Return type**

[Gitlab](#)

**geonodes**

See [GeoNodeManager](#)

**get\_license**(\*\*kwargs: Any) → Dict[str, str | Dict[str, str]]

Retrieve information about the current license.

**Parameters**

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Returns**

The current license information

**Return type**

Dict[str, str | Dict[str, str]]

**gitignores**

See [GitignoreManager](#)

**gitlabciymls**

See [GitlabciyamlManager](#)

**groups**

See [GroupManager](#)

**headers**

Headers that will be used in request to GitLab

**hooks**

See [HookManager](#)

**http\_delete**(path: str, \*\*kwargs: Any) → Response

Make a DELETE request to the Gitlab server.

**Parameters**

- **path** (str) – Path or full URL to query (‘/projects’ or ‘http://whatever/v4/api/projects’)
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Returns**

The requests object.

**Raises**

**GitlabHttpError** – When the return code is not 2xx

**Return type**

Response

**http\_get**(path: str, query\_data: Dict[str, Any] | None = None, streamed: bool = False, raw: bool = False, \*\*kwargs: Any) → Dict[str, Any] | Response

Make a GET request to the Gitlab server.

**Parameters**

- **path** (str) – Path or full URL to query (‘/projects’ or ‘http://whatever/v4/api/projects’)
- **query\_data** (Dict[str, Any] | None) – Data to send as query parameters

- **streamed** (*bool*) – Whether the data should be streamed
- **raw** (*bool*) – If True do not try to parse the output as json
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

A requests result object is streamed is True or the content type is not json. The parsed json data otherwise.

**Raises**

- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

**Return type**

*Dict[str, Any] | Response*

**http\_head**(*path: str, query\_data: Dict[str, Any] | None = None, \*\*kwargs: Any*) → *CaseInsensitiveDict[Any]*

Make a HEAD request to the Gitlab server.

**Parameters**

- **path** (*str*) – Path or full URL to query (‘/projects’ or ‘http://whatever/v4/api/projects’)
- **query\_data** (*Dict[str, Any] | None*) – Data to send as query parameters
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo, page, per\_page)

**Returns**

A requests.header object

**Raises**

**GitlabHttpError** – When the return code is not 2xx

**Return type**

*CaseInsensitiveDict[Any]*

**http\_list**(*path: str, query\_data: Dict[str, Any] | None = None, \*, iterator: bool | None = None, \*\*kwargs: Any*) → *GitlabList | List[Dict[str, Any]]*

Make a GET request to the Gitlab server for list-oriented queries.

**Parameters**

- **path** (*str*) – Path or full URL to query (‘/projects’ or ‘http://whatever/v4/api/projects’)
- **query\_data** (*Dict[str, Any] | None*) – Data to send as query parameters
- **iterator** (*bool | None*) – Indicate if should return a generator (True)
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo, page, per\_page)

**Returns**

A list of the objects returned by the server. If *iterator* is True and no pagination-related arguments (*page*, *per\_page*, *get\_all*) are defined then a GitlabList object (generator) is returned instead. This object will make API calls when needed to fetch the next items from the server.

**Raises**

- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed



**Return type**`GitlabList | List[Dict[str, Any]]`

**http\_patch**(*path*: str, \*, *query\_data*: Dict[str, Any] | None = None, *post\_data*: Dict[str, Any] | bytes | None = None, *raw*: bool = False, \*\*kwargs: Any) → Dict[str, Any] | Response

Make a PATCH request to the Gitlab server.

**Parameters**

- **path** (str) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projects')
- **query\_data** (Dict[str, Any] | None) – Data to send as query parameters
- **post\_data** (Dict[str, Any] | bytes | None) – Data to send in the body (will be converted to json by default)
- **raw** (bool) – If True, do not convert post\_data to json
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Returns**

The parsed json returned by the server.

**Raises**

- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

**Return type**`Dict[str, Any] | Response`

**http\_post**(*path*: str, *query\_data*: Dict[str, Any] | None = None, *post\_data*: Dict[str, Any] | None = None, *raw*: bool = False, *files*: Dict[str, Any] | None = None, \*\*kwargs: Any) → Dict[str, Any] | Response

Make a POST request to the Gitlab server.

**Parameters**

- **path** (str) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projects')
- **query\_data** (Dict[str, Any] | None) – Data to send as query parameters
- **post\_data** (Dict[str, Any] | None) – Data to send in the body (will be converted to json by default)
- **raw** (bool) – If True, do not convert post\_data to json
- **files** (Dict[str, Any] | None) – The files to send to the server
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Returns**

The parsed json returned by the server if json is return, else the raw content

**Raises**

- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

**Return type**`Dict[str, Any] | Response`

**http\_put**(*path*: str, *query\_data*: Dict[str, Any] | None = None, *post\_data*: Dict[str, Any] | bytes | BinaryIO | None = None, *raw*: bool = False, *files*: Dict[str, Any] | None = None, *\*\*kwargs*: Any) → Dict[str, Any] | Response

Make a PUT request to the Gitlab server.

#### Parameters

- **path** (str) – Path or full URL to query (‘/projects’ or ‘http://whatever/v4/api/projects’)
- **query\_data** (Dict[str, Any] | None) – Data to send as query parameters
- **post\_data** (Dict[str, Any] | bytes | BinaryIO | None) – Data to send in the body (will be converted to json by default)
- **raw** (bool) – If True, do not convert post\_data to json
- **files** (Dict[str, Any] | None) – The files to send to the server
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

#### Returns

The parsed json returned by the server.

#### Raises

- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

#### Return type

Dict[str, Any] | Response

**http\_request**(*verb*: str, *path*: str, *query\_data*: Dict[str, Any] | None = None, *post\_data*: Dict[str, Any] | bytes | BinaryIO | None = None, *raw*: bool = False, *streamed*: bool = False, *files*: Dict[str, Any] | None = None, *timeout*: float | None = None, *obey\_rate\_limit*: bool = True, *retry\_transient\_errors*: bool | None = None, *max\_retries*: int = 10, *\*\*kwargs*: Any) → Response

Make an HTTP request to the Gitlab server.

#### Parameters

- **verb** (str) – The HTTP method to call (‘get’, ‘post’, ‘put’, ‘delete’)
- **path** (str) – Path or full URL to query (‘/projects’ or ‘http://whatever/v4/api/projects’)
- **query\_data** (Dict[str, Any] | None) – Data to send as query parameters
- **post\_data** (Dict[str, Any] | bytes | BinaryIO | None) – Data to send in the body (will be converted to json by default)
- **raw** (bool) – If True, do not convert post\_data to json
- **streamed** (bool) – Whether the data should be streamed
- **files** (Dict[str, Any] | None) – The files to send to the server
- **timeout** (float | None) – The timeout, in seconds, for the request
- **obey\_rate\_limit** (bool) – Whether to obey 429 Too Many Request responses. Defaults to True.
- **retry\_transient\_errors** (bool | None) – Whether to retry after 500, 502, 503, 504 or 52x responses. Defaults to False.
- **max\_retries** (int) – Max retries after 429 or transient errors, set to -1 to retry forever. Defaults to 10.

- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

A requests result object.

**Raises**

**GitlabHttpError** – When the return code is not 2xx

**Return type**

*Response*

**issues**

See *IssueManager*

**issues\_statistics**

See *IssuesStatisticsManager*

**keys**

See *KeyManager*

**ldapgroups**

See *LDAPGroupManager*

**licenses**

See *LicenseManager*

**markdown**(*text: str, gfm: bool = False, project: str | None = None, \*\*kwargs: Any*) → *str*

Render an arbitrary Markdown document.

**Parameters**

- **text** (*str*) – The markdown text to render
- **gfm** (*bool*) – Render text using GitLab Flavored Markdown. Default is False
- **project** (*str | None*) – Full path of a project used a context when *gfm* is True
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabMarkdownError** – If the server cannot perform the request

**Returns**

The HTML rendering of the markdown text.

**Return type**

*str*

**classmethod merge\_config**(*options: Dict[str, Any], gitlab\_id: str | None = None, config\_files: List[str] | None = None*) → *Gitlab*

Create a Gitlab connection by merging configuration with the following precedence:

1. Explicitly provided CLI arguments,
2. Environment variables,
3. **Configuration files:**
  - a. **explicitly defined config files:**
    - i. via the *-config-file* CLI argument,

- ii. via the `PYTHON_GITLAB_CFG` environment variable,
  - b. user-specific config file,
  - c. system-level config file,
4. Environment variables always present in CI (`CI_SERVER_URL`, `CI_JOB_TOKEN`).

**Parameters**

- **options** (`Dict[str, Any]`) – A dictionary of explicitly provided key-value options.
- **gitlab\_id** (`str | None`) – ID of the configuration section.
- **config\_files** (`List[str] | None`) – List of paths to configuration files.

**Returns**

A Gitlab connection.

**Return type**

(*gitlab.Gitlab*)

**Raises**

*gitlab.config.GitlabDataError* – If the configuration is not correct.

**mergerequests**

See *MergeRequestManager*

**namespaces**

See *NamespaceManager*

**notificationsettings**

See *NotificationSettingsManager*

**pagesdomains**

See *PagesDomainManager*

**personal\_access\_tokens**

See *PersonalAccessTokenManager*

**projects**

See *ProjectManager*

**registry\_repositories**

See *RegistryRepositoryManager*

**runners**

See *RunnerManager*

**runners\_all**

See *RunnerManager*

**search**(*scope: str, search: str, \*\*kwargs: Any*) → *GitlabList* | `List[Dict[str, Any]]`

Search GitLab resources matching the provided string.

**Parameters**

- **scope** (*str*) – Scope of the search
- **search** (*str*) – Search string
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo`)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabSearchError** – If the server failed to perform the request

**Returns**

A list of dicts describing the resources found.

**Return type**

[GitlabList](#) | *List[Dict[str, Any]]*

**set\_license**(*license*: str, *\*\*kwargs*: Any) → Dict[str, Any]

Add a new license.

**Parameters**

- **license** (str) – The license string
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabPostError** – If the server cannot perform the request

**Returns**

The new license information

**Return type**

*Dict[str, Any]*

**settings**

See [ApplicationSettingsManager](#)

**sidekiq**

See [SidekiqManager](#)

**snippets**

See [SnippetManager](#)

**ssl\_verify**

Whether SSL certificates should be validated

**statistics**

See [ApplicationStatisticsManager](#)

**timeout**

Timeout to use for requests to gitlab server

**todos**

See [TodoManager](#)

**topics**

See [TopicManager](#)

**property url: str**

The user-provided server URL.

**user\_activities**

See [UserActivitiesManager](#)

**users**

See [\*UserManager\*](#)

**variables**

See [\*VariableManager\*](#)

**version()** → Tuple[str, str]

Returns the version and revision of the gitlab server.

Note that self.version and self.revision will be set on the gitlab object.

**Returns**

**The server version and server revision.**

(‘unknown’, ‘unknown’) if the server doesn’t perform as expected.

**Return type**

*Tuple*[str, str]

**exception** gitlab.GitlabActivateError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.GitlabAttachFileError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.GitlabAuthenticationError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [\*GitlabError\*](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabBanError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabBlockError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabBuildCancelError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabCancelError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabBuildEraseError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabRetryError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabBuildPlayError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabRetryError`](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabBuildRetryError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [\*GitlabRetryError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabCancelError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabCherryPickError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabCiLintError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [\*GitlabError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)



**Return type**

None

**exception** `gitlab.GitlabConnectionError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [GitlabError](#)**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** `gitlab.GitlabCreateError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [GitlabOperationError](#)**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** `gitlab.GitlabDeactivateError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [GitlabOperationError](#)**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** `gitlab.GitlabDeleteError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [GitlabOperationError](#)**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** `gitlab.GitlabDeploymentApprovalError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: `Exception`

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabFollowError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabGetError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabGroupTransferError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** gitlab.GitlabHeadError(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** gitlab.GitlabHousekeepingError(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** gitlab.GitlabHttpError(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [GitlabError](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** gitlab.GitlabImportError(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** `gitlab.GitlabInvitationError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [`GitlabOperationError`](#)**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** `gitlab.GitlabJobCancelError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [`GitlabCancelError`](#)**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** `gitlab.GitlabJobEraseError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [`GitlabRetryError`](#)**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** `gitlab.GitlabJobPlayError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [`GitlabRetryError`](#)**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** gitlab.GitlabJobRetryError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabRetryError](#)

#### Parameters

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

#### Return type

None

**exception** gitlab.GitlabLicenseError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

#### Return type

None

**class** gitlab.GitlabList(*gl: [Gitlab](#), url: str, query\_data: Dict[str, [Any](#)], get\_next: bool = True, \*\*kwargs: [Any](#)*)

Bases: `object`

Generator representing a list of remote objects.

The object handles the links returned by a query to the API, and will call the API again when needed.

#### Parameters

- **gl** ([Gitlab](#))
- **url** (*str*)
- **query\_data** (*Dict[str, [Any](#)]*)
- **get\_next** (*bool*)
- **kwargs** ([Any](#))

**property** current\_page: `int`

The current page number.

**next()** → `Dict[str, Any]`

#### Return type

`Dict[str, Any]`

**property** next\_page: `int | None`

The next page number.

If None, the current page is the last.

**property per\_page:** `int | None`

The number of items per page.

**property prev\_page:** `int | None`

The previous page number.

If None, the current page is the first.

**property total:** `int | None`

The total number of items.

**property total\_pages:** `int | None`

The total number of pages.

**exception** `gitlab.GitlabListError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

#### Parameters

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

#### Return type

None

**exception** `gitlab.GitlabMRApprovalError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

#### Parameters

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

#### Return type

None

**exception** `gitlab.GitlabMRClosedError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

#### Parameters

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

#### Return type

None

**exception** `gitlab.GitlabMRForbiddenError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

#### Parameters

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabMROnBuildSuccessError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabMRRebaseError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabMRResetApprovalError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabMarkdownError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** `gitlab.GitlabOperationError`(*error\_message: str | bytes = "*, *response\_code: int | None = None*,  
*response\_body: bytes | None = None*)

Bases: [`GitlabError`](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabOwnershipError`(*error\_message: str | bytes = "*, *response\_code: int | None = None*,  
*response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabParsingError`(*error\_message: str | bytes = "*, *response\_code: int | None = None*,  
*response\_body: bytes | None = None*)

Bases: [`GitlabError`](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabPipelineCancelError`(*error\_message: str | bytes = "*, *response\_code: int | None = None*, *response\_body: bytes | None = None*)

Bases: [`GitlabCancelError`](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None



**exception** `gitlab.GitlabPipelinePlayError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabRetryError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabPipelineRetryError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabRetryError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabProjectDeployKeyError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabPromoteError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabProtectError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabRenderError`(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabRepairError`(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabRestoreError`(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabRetryError`(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** `gitlab.GitlabRevertError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabRotateError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabSearchError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabSetError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabStopError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabSubscribeError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabTimeTrackingError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabTodoError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabTopicMergeError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabTransferProjectError`(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabUnbanError`(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabUnblockError`(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.GitlabUnfollowError`(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** `gitlab.GitlabUnsubscribeError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabUpdateError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabUploadError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabUserApproveError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.GitlabUserRejectError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [`GitlabOperationError`](#)

#### Parameters

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

#### Return type

*None*

**exception** `gitlab.GitlabVerifyError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [`GitlabOperationError`](#)

#### Parameters

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

#### Return type

*None*

**exception** `gitlab.RedirectError`(*error\_message*: *str* | *bytes* = "", *response\_code*: *int* | *None* = *None*, *response\_body*: *bytes* | *None* = *None*)

Bases: [`GitlabError`](#)

#### Parameters

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

#### Return type

*None*

**class** `gitlab.Gitlab`(*url*: *str* | *None* = *None*, *private\_token*: *str* | *None* = *None*, *oauth\_token*: *str* | *None* = *None*, *job\_token*: *str* | *None* = *None*, *ssl\_verify*: *bool* | *str* = *True*, *http\_username*: *str* | *None* = *None*, *http\_password*: *str* | *None* = *None*, *timeout*: *float* | *None* = *None*, *api\_version*: *str* = '4', *per\_page*: *int* | *None* = *None*, *pagination*: *str* | *None* = *None*, *order\_by*: *str* | *None* = *None*, *user\_agent*: *str* = 'python-gitlab/4.4.0', *retry\_transient\_errors*: *bool* = *False*, *keep\_base\_url*: *bool* = *False*, *\*\*kwargs*: [`Any`](#))

Bases: `object`

Represents a GitLab server connection.

#### Parameters

- **url** (*str* | *None*) – The URL of the GitLab server (defaults to <https://gitlab.com>).
- **private\_token** (*str* | *None*) – The user private token
- **oauth\_token** (*str* | *None*) – An oauth token
- **job\_token** (*str* | *None*) – A CI job token

- **ssl\_verify** (*bool* / *str*) – Whether SSL certificates should be validated. If the value is a string, it is the path to a CA file used for certificate validation.
- **timeout** (*float* / *None*) – Timeout to use for requests to the GitLab server.
- **http\_username** (*str* / *None*) – Username for HTTP authentication
- **http\_password** (*str* / *None*) – Password for HTTP authentication
- **api\_version** (*str*) – Gitlab API version to use (support for 4 only)
- **pagination** (*str* / *None*) – Can be set to ‘keyset’ to use keyset pagination
- **order\_by** (*str* / *None*) – Set order\_by globally
- **user\_agent** (*str*) – A custom user agent to use for making HTTP requests.
- **retry\_transient\_errors** (*bool*) – Whether to retry after 500, 502, 503, 504 or 52x responses. Defaults to False.
- **keep\_base\_url** (*bool*) – keep user-provided base URL for pagination if it differs from response headers
- **session** (*requests.Session*) – HTTP Requests Session
- **backend** (*RequestsBackend*) – Backend that will be used to make http requests
- **per\_page** (*int* / *None*)
- **kwargs** (*Any*)

**property api\_url:** *str*

The computed API base URL.

**property api\_version:** *str*

The API version used (4 only).

**appearance**

See [\*ApplicationAppearanceManager\*](#)

**applications**

See [\*ApplicationManager\*](#)

**audit\_events**

See [\*AuditEventManager\*](#)

**auth()** → *None*

Performs an authentication using private token. Warns the user if a potentially misconfigured URL is detected on the client or server side.

The *user* attribute will hold a *gitlab.objects.CurrentUser* object on success.

**Return type**

*None*

**broadcastmessages**

See [\*BroadcastMessageManager\*](#)

**bulk\_import\_entities**

See [\*BulkImportAllEntityManager\*](#)

**bulk\_imports**

See [\*BulkImportManager\*](#)



**ci\_lint**See [CiLintManager](#)**deploykeys**See [DeployKeyManager](#)**deploytokens**See [DeployTokenManager](#)**dockerfiles**See [DockerfileManager](#)**enable\_debug**(*mask\_credentials: bool = True*) → None**Parameters****mask\_credentials** (*bool*)**Return type**

None

**events**See [EventManager](#)**features**See [FeatureManager](#)**classmethod from\_config**(*gitlab\_id: str | None = None, config\_files: List[str] | None = None, \*\*kwargs: Any*) → [Gitlab](#)

Create a Gitlab connection from configuration files.

**Parameters**

- **gitlab\_id** (*str | None*) – ID of the configuration section.
- **list[str]** (*config\_files*) – List of paths to configuration files.
- **config\_files** (*List[str] | None*)
- **kwargs** (*Any*)

**Return type**[Gitlab](#)**kwargs:**

session requests.Session: Custom requests Session

**Returns**

A Gitlab connection.

**Raises**[gitlab.config.GitlabDataError](#) – If the configuration is not correct.**Parameters**

- **gitlab\_id** (*str | None*)
- **config\_files** (*List[str] | None*)
- **kwargs** (*Any*)

**Return type**[Gitlab](#)

**geonodes**

See [\*GeoNodeManager\*](#)

**get\_license**(\*\*kwargs: [\*Any\*](#)) → Dict[str, str | Dict[str, str]]

Retrieve information about the current license.

**Parameters**

**\*\*kwargs** ([\*Any\*](#)) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Returns**

The current license information

**Return type**

Dict[str, str | Dict[str, str]]

**gitignores**

See [\*GitignoreManager\*](#)

**gitlabciymls**

See [\*GitlabciyamlManager\*](#)

**groups**

See [\*GroupManager\*](#)

**headers**

Headers that will be used in request to GitLab

**hooks**

See [\*HookManager\*](#)

**http\_delete**(path: str, \*\*kwargs: [\*Any\*](#)) → Response

Make a DELETE request to the Gitlab server.

**Parameters**

- **path** (str) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projects')
- **\*\*kwargs** ([\*Any\*](#)) – Extra options to send to the server (e.g. sudo)

**Returns**

The requests object.

**Raises**

**GitlabHttpError** – When the return code is not 2xx

**Return type**

*Response*

**http\_get**(path: str, query\_data: Dict[str, [\*Any\*](#)] | None = None, streamed: bool = False, raw: bool = False, \*\*kwargs: [\*Any\*](#)) → Dict[str, [\*Any\*](#)] | Response

Make a GET request to the Gitlab server.

**Parameters**

- **path** (str) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projects')

- **query\_data** (*Dict[str, Any] | None*) – Data to send as query parameters
- **streamed** (*bool*) – Whether the data should be streamed
- **raw** (*bool*) – If True do not try to parse the output as json
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

A requests result object is streamed is True or the content type is not json. The parsed json data otherwise.

**Raises**

- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

**Return type**

*Dict[str, Any] | Response*

**http\_head**(*path: str, query\_data: Dict[str, Any] | None = None, \*\*kwargs: Any*) → *CaseInsensitiveDict[Any]*

Make a HEAD request to the Gitlab server.

**Parameters**

- **path** (*str*) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projects')
- **query\_data** (*Dict[str, Any] | None*) – Data to send as query parameters
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo, page, per\_page)

**Returns**

*A requests.header object*

**Raises**

**GitlabHttpError** – When the return code is not 2xx

**Return type**

*CaseInsensitiveDict[Any]*

**http\_list**(*path: str, query\_data: Dict[str, Any] | None = None, \*, iterator: bool | None = None, \*\*kwargs: Any*) → *GitlabList | List[Dict[str, Any]]*

Make a GET request to the Gitlab server for list-oriented queries.

**Parameters**

- **path** (*str*) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projects')
- **query\_data** (*Dict[str, Any] | None*) – Data to send as query parameters
- **iterator** (*bool | None*) – Indicate if should return a generator (True)
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo, page, per\_page)

**Returns**

A list of the objects returned by the server. If *iterator* is True and no pagination-related arguments (*page*, *per\_page*, *get\_all*) are defined then a *GitlabList* object (generator) is returned instead. This object will make API calls when needed to fetch the next items from the server.

**Raises**

- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

**Return type**

`GitlabList` | `List[Dict[str, Any]]`

**http\_patch**(*path*: `str`, \*, *query\_data*: `Dict[str, Any]` | `None` = `None`, *post\_data*: `Dict[str, Any]` | `bytes` | `None` = `None`, *raw*: `bool` = `False`, *\*\*kwargs*: `Any`) → `Dict[str, Any]` | `Response`

Make a PATCH request to the Gitlab server.

**Parameters**

- **path** (`str`) – Path or full URL to query (‘/projects’ or ‘http://whatever/v4/api/projects’)
- **query\_data** (`Dict[str, Any]` | `None`) – Data to send as query parameters
- **post\_data** (`Dict[str, Any]` | `bytes` | `None`) – Data to send in the body (will be converted to json by default)
- **raw** (`bool`) – If True, do not convert post\_data to json
- **\*\*kwargs** (`Any`) – Extra options to send to the server (e.g. sudo)

**Returns**

The parsed json returned by the server.

**Raises**

- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

**Return type**

`Dict[str, Any]` | `Response`

**http\_post**(*path*: `str`, *query\_data*: `Dict[str, Any]` | `None` = `None`, *post\_data*: `Dict[str, Any]` | `None` = `None`, *raw*: `bool` = `False`, *files*: `Dict[str, Any]` | `None` = `None`, *\*\*kwargs*: `Any`) → `Dict[str, Any]` | `Response`

Make a POST request to the Gitlab server.

**Parameters**

- **path** (`str`) – Path or full URL to query (‘/projects’ or ‘http://whatever/v4/api/projects’)
- **query\_data** (`Dict[str, Any]` | `None`) – Data to send as query parameters
- **post\_data** (`Dict[str, Any]` | `None`) – Data to send in the body (will be converted to json by default)
- **raw** (`bool`) – If True, do not convert post\_data to json
- **files** (`Dict[str, Any]` | `None`) – The files to send to the server
- **\*\*kwargs** (`Any`) – Extra options to send to the server (e.g. sudo)

**Returns**

The parsed json returned by the server if json is return, else the raw content

**Raises**

- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

**Return type***Dict[str, Any] | Response*

**http\_put**(*path: str, query\_data: Dict[str, Any] | None = None, post\_data: Dict[str, Any] | bytes | BinaryIO | None = None, raw: bool = False, files: Dict[str, Any] | None = None, \*\*kwargs: Any*) → *Dict[str, Any] | Response*

Make a PUT request to the Gitlab server.

**Parameters**

- **path** (*str*) – Path or full URL to query (‘/projects’ or ‘http://whatever/v4/api/projects’)
- **query\_data** (*Dict[str, Any] | None*) – Data to send as query parameters
- **post\_data** (*Dict[str, Any] | bytes | BinaryIO | None*) – Data to send in the body (will be converted to json by default)
- **raw** (*bool*) – If True, do not convert post\_data to json
- **files** (*Dict[str, Any] | None*) – The files to send to the server
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The parsed json returned by the server.

**Raises**

- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

**Return type***Dict[str, Any] | Response*

**http\_request**(*verb: str, path: str, query\_data: Dict[str, Any] | None = None, post\_data: Dict[str, Any] | bytes | BinaryIO | None = None, raw: bool = False, streamed: bool = False, files: Dict[str, Any] | None = None, timeout: float | None = None, obey\_rate\_limit: bool = True, retry\_transient\_errors: bool | None = None, max\_retries: int = 10, \*\*kwargs: Any*) → *Response*

Make an HTTP request to the Gitlab server.

**Parameters**

- **verb** (*str*) – The HTTP method to call (‘get’, ‘post’, ‘put’, ‘delete’)
- **path** (*str*) – Path or full URL to query (‘/projects’ or ‘http://whatever/v4/api/projects’)
- **query\_data** (*Dict[str, Any] | None*) – Data to send as query parameters
- **post\_data** (*Dict[str, Any] | bytes | BinaryIO | None*) – Data to send in the body (will be converted to json by default)
- **raw** (*bool*) – If True, do not convert post\_data to json
- **streamed** (*bool*) – Whether the data should be streamed
- **files** (*Dict[str, Any] | None*) – The files to send to the server
- **timeout** (*float | None*) – The timeout, in seconds, for the request
- **obey\_rate\_limit** (*bool*) – Whether to obey 429 Too Many Request responses. Defaults to True.

- **retry\_transient\_errors** (*bool* / *None*) – Whether to retry after 500, 502, 503, 504 or 52x responses. Defaults to False.
- **max\_retries** (*int*) – Max retries after 429 or transient errors, set to -1 to retry forever. Defaults to 10.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

A requests result object.

**Raises**

**GitlabHttpError** – When the return code is not 2xx

**Return type**

*Response*

**issues**

See *IssueManager*

**issues\_statistics**

See *IssuesStatisticsManager*

**keys**

See *KeyManager*

**ldapgroups**

See *LDAPGroupManager*

**licenses**

See *LicenseManager*

**markdown**(*text*: *str*, *gfm*: *bool* = *False*, *project*: *str* | *None* = *None*, **\*\*kwargs**: *Any*) → *str*

Render an arbitrary Markdown document.

**Parameters**

- **text** (*str*) – The markdown text to render
- **gfm** (*bool*) – Render text using GitLab Flavored Markdown. Default is False
- **project** (*str* / *None*) – Full path of a project used a context when *gfm* is True
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabMarkdownError** – If the server cannot perform the request

**Returns**

The HTML rendering of the markdown text.

**Return type**

*str*

**classmethod merge\_config**(*options*: *Dict*[*str*, *Any*], *gitlab\_id*: *str* | *None* = *None*, *config\_files*: *List*[*str*] | *None* = *None*) → *Gitlab*

Create a Gitlab connection by merging configuration with the following precedence:

1. Explicitly provided CLI arguments,
2. Environment variables,

### 3. Configuration files:

#### a. explicitly defined config files:

- i. via the `--config-file` CLI argument,
- ii. via the `PYTHON_GITLAB_CFG` environment variable,

#### b. user-specific config file,

#### c. system-level config file,

### 4. Environment variables always present in CI (`CI_SERVER_URL`, `CI_JOB_TOKEN`).

#### Parameters

- **options** (`Dict[str, Any]`) – A dictionary of explicitly provided key-value options.
- **gitlab\_id** (`str | None`) – ID of the configuration section.
- **config\_files** (`List[str] | None`) – List of paths to configuration files.

#### Returns

A Gitlab connection.

#### Return type

(*gitlab.Gitlab*)

#### Raises

*gitlab.config.GitlabDataError* – If the configuration is not correct.

#### mergerequests

See *MergeRequestManager*

#### namespaces

See *NamespaceManager*

#### notificationsettings

See *NotificationSettingsManager*

#### pagesdomains

See *PagesDomainManager*

#### personal\_access\_tokens

See *PersonalAccessTokenManager*

#### projects

See *ProjectManager*

#### registry\_repositories

See *RegistryRepositoryManager*

#### runners

See *RunnerManager*

#### runners\_all

See *RunnerManager*

**search**(*scope: str*, *search: str*, *\*\*kwargs: Any*) → *GitlabList* | List[Dict[str, Any]]

Search GitLab resources matching the provided string.

#### Parameters

- **scope** (*str*) – Scope of the search
- **search** (*str*) – Search string
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabSearchError** – If the server failed to perform the request

**Returns**

A list of dicts describing the resources found.

**Return type**

[GitlabList](#) | *List[Dict[str, Any]]*

**set\_license**(*license: str*, *\*\*kwargs: Any*) → *Dict[str, Any]*

Add a new license.

**Parameters**

- **license** (*str*) – The license string
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabPostError** – If the server cannot perform the request

**Returns**

The new license information

**Return type**

*Dict[str, Any]*

**settings**

See [ApplicationSettingsManager](#)

**sidekiq**

See [SidekiqManager](#)

**snippets**

See [SnippetManager](#)

**ssl\_verify**

Whether SSL certificates should be validated

**statistics**

See [ApplicationStatisticsManager](#)

**timeout**

Timeout to use for requests to gitlab server

**todos**

See [TodoManager](#)

**topics**

See [TopicManager](#)



**property url:** `str`

The user-provided server URL.

**user:** `objects.CurrentUser | None`

**user\_activities**

See [\*UserActivitiesManager\*](#)

**users**

See [\*UserManager\*](#)

**variables**

See [\*VariableManager\*](#)

**version()** → `Tuple[str, str]`

Returns the version and revision of the gitlab server.

Note that `self.version` and `self.revision` will be set on the gitlab object.

#### Returns

**The server version and server revision.**

(‘unknown’, ‘unknown’) if the server doesn’t perform as expected.

#### Return type

`Tuple[str, str]`

**class** `gitlab.GitlabList`(*gl*: [`Gitlab`](#), *url*: `str`, *query\_data*: `Dict[str, Any]`, *get\_next*: `bool = True`, *\*\*kwargs*: `Any`)

Bases: `object`

Generator representing a list of remote objects.

The object handles the links returned by a query to the API, and will call the API again when needed.

#### Parameters

- **gl** ([`Gitlab`](#))
- **url** (`str`)
- **query\_data** (`Dict[str, Any]`)
- **get\_next** (`bool`)
- **kwargs** (`Any`)

**property current\_page:** `int`

The current page number.

**next()** → `Dict[str, Any]`

#### Return type

`Dict[str, Any]`

**property next\_page:** `int | None`

The next page number.

If `None`, the current page is the last.

**property per\_page:** `int | None`

The number of items per page.

**property prev\_page:** `int | None`

The previous page number.

If None, the current page is the first.

**property total:** `int | None`

The total number of items.

**property total\_pages:** `int | None`

The total number of pages.

## 7.6.2 Subpackages

### gitlab.v4 package

#### Submodules

#### gitlab.v4.objects module

**class** `gitlab.v4.objects.Any(*args, **kwargs)`

Bases: `object`

Special type indicating an unconstrained type.

- Any is compatible with every type.
- Any assumed to have all methods.
- All values assumed to be instances of Any.

Note that all the above statements are true from the point of view of static type checkers. At runtime, Any should not be used with instance checks.

**class** `gitlab.v4.objects.Application(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)`

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** `gitlab.v4.objects.ApplicationAppearance(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)`

Bases: [SaveMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**ApplicationAppearanceManager**(*gl*: Gitlab, *parent*: RESTObject | None = None)

Bases: [GetWithoutIdMixin](#), [UpdateMixin](#), [RESTManager](#)

### Object Creation

### Object update

Optional attributes for object update:

- title
- description
- logo
- header\_logo
- favicon
- new\_project\_guidelines
- header\_message
- footer\_message
- message\_background\_color
- message\_font\_color
- email\_header\_and\_footer\_enabled

### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

**get**(\*\*kwargs: Any) → [ApplicationAppearance](#)

Retrieve a single object.

### Parameters

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

### Returns

The generated RESTObject

### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

### Return type

[ApplicationAppearance](#)

**update**(*id*: int | str | None = None, *new\_data*: Dict[str, Any] | None = None, \*\*kwargs: Any) → Dict[str, Any]

Update an object on the server.

### Parameters

- **id** (int | str | None) – ID of the object to update (can be None if not required)
- **new\_data** (Dict[str, Any] | None) – the update data for the object
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Returns**

The new object data (*not* a `RESTObject`)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUpdateError** – If the server cannot perform the request

**Return type**

`Dict[str, Any]`

**class** gitlab.v4.objects.**ApplicationManager**(gl: `Gitlab`, parent: `RESTObject` | `None` = `None`)

Bases: `ListMixin`, `CreateMixin`, `DeleteMixin`, `RESTManager`

**Object Creation**

Required attributes for object create:

- `name`
- `redirect_uri`
- `scopes`

Optional attributes for object create:

- `confidential`

**Object update****Parameters**

- **gl** (`Gitlab`)
- **parent** (`RESTObject` | `None`)

**class** gitlab.v4.objects.**ApplicationSettings**(manager: `RESTManager`, attrs: `Dict[str, Any]`, \*,  
created\_from\_list: `bool` = `False`, lazy: `bool` = `False`)

Bases: `SaveMixin`, `RESTObject`

**Parameters**

- **manager** (`RESTManager`)
- **attrs** (`Dict[str, Any]`)
- **created\_from\_list** (`bool`)
- **lazy** (`bool`)

**class** gitlab.v4.objects.**ApplicationSettingsManager**(gl: `Gitlab`, parent: `RESTObject` | `None` = `None`)

Bases: `GetWithoutIdMixin`, `UpdateMixin`, `RESTManager`

**Object Creation****Object update**

Optional attributes for object update:

- `id`
- `default_projects_limit`
- `signup_enabled`
- `password_authentication_enabled_for_web`
- `gravatar_enabled`

- `sign_in_text`
- `created_at`
- `updated_at`
- `home_page_url`
- `default_branch_protection`
- `restricted_visibility_levels`
- `max_attachment_size`
- `session_expire_delay`
- `default_project_visibility`
- `default_snippet_visibility`
- `default_group_visibility`
- `outbound_local_requests_whitelist`
- `disabled_oauth_sign_in_sources`
- `domain_whitelist`
- `domain_blacklist_enabled`
- `domain_blacklist`
- `domain_allowlist`
- `domain_denylist_enabled`
- `domain_denylist`
- `external_authorization_service_enabled`
- `external_authorization_service_url`
- `external_authorization_service_default_label`
- `external_authorization_service_timeout`
- `import_sources`
- `user_oauth_applications`
- `after_sign_out_path`
- `container_registry_token_expire_delay`
- `repository_storages`
- `plantuml_enabled`
- `plantuml_url`
- `terminal_max_session_time`
- `polling_interval_multiplier`
- `rsa_key_restriction`
- `dsa_key_restriction`
- `ecdsa_key_restriction`
- `ed25519_key_restriction`

- `first_day_of_week`
- `enforce_terms`
- `terms`
- `performance_bar_allowed_group_id`
- `instance_statistics_visibility_private`
- `user_show_add_ssh_key_message`
- `file_template_project_id`
- `local_markdown_version`
- `asset_proxy_enabled`
- `asset_proxy_url`
- `asset_proxy_whitelist`
- `asset_proxy_allowlist`
- `geo_node_allowed_ips`
- `allow_local_requests_from_hooks_and_services`
- `allow_local_requests_from_web_hooks_and_services`
- `allow_local_requests_from_system_hooks`

#### Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) | `None`)

`get(**kwargs: Any) → ApplicationSettings`

Retrieve a single object.

#### Parameters

**`**kwargs`** ([Any](#)) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated [RESTObject](#)

#### Raises

- **`GitlabAuthenticationError`** – If authentication is not correct
- **`GitlabGetError`** – If the server cannot perform the request

#### Return type

[ApplicationSettings](#)

`update(id: int | str | None = None, new_data: Dict[str, Any] | None = None, **kwargs: Any) → Dict[str, Any]`

Update an object on the server.

#### Parameters

- **`id`** (`int` | `str` | `None`) – ID of the object to update (can be `None` if not required)
- **`new_data`** (`Dict[str, Any]` | `None`) – the update data for the object
- **`**kwargs`** ([Any](#)) – Extra options to send to the server (e.g. `sudo`)

**Returns**

The new object data (*not* a `RESTObject`)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUpdateError** – If the server cannot perform the request

**Return type**

`Dict[str, Any]`

```
class gitlab.v4.objects.ApplicationStatistics(manager: RESTManager, attrs: Dict[str, Any], *,
                                             created_from_list: bool = False, lazy: bool = False)
```

Bases: `RESTObject`

**Parameters**

- **manager** (`RESTManager`)
- **attrs** (`Dict[str, Any]`)
- **created\_from\_list** (`bool`)
- **lazy** (`bool`)

```
class gitlab.v4.objects.ApplicationStatisticsManager(gl: Gitlab, parent: RESTObject | None =
                                                    None)
```

Bases: `GetWithoutIdMixin`, `RESTManager`

**Object Creation****Object update****Parameters**

- **gl** (`Gitlab`)
- **parent** (`RESTObject` | `None`)

```
get(**kwargs: Any) → ApplicationStatistics
```

Retrieve a single object.

**Parameters**

**\*\*kwargs** (`Any`) – Extra options to send to the server (e.g. `sudo`)

**Returns**

The generated `RESTObject`

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

`ApplicationStatistics`

```
class gitlab.v4.objects.AuditEvent(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
                                   bool = False, lazy: bool = False)
```

Bases: `RESTObject`

**Parameters**

- **manager** (`RESTManager`)

- `attrs` (*Dict[str, Any]*)
- `created_from_list` (*bool*)
- `lazy` (*bool*)

**class** gitlab.v4.objects.**AuditEventManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *RetrieveMixin, RESTManager*

#### Object listing filters

- `created_after`
- `created_before`
- `entity_type`
- `entity_id`

#### Object Creation

#### Object update

##### Parameters

- `gl` (*Gitlab*)
- `parent` (*RESTObject | None*)

**get**(*id: str | int, lazy: bool = False, \*\*kwargs: Any*) → *AuditEvent*

Retrieve a single object.

##### Parameters

- `id` (*str | int*) – ID of the object to retrieve
- `lazy` (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- `**kwargs` (*Any*) – Extra options to send to the server (e.g. `sudo`)

##### Returns

The generated *RESTObject*.

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

##### Return type

*AuditEvent*

**class** gitlab.v4.objects.**BroadcastMessage**(*manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False*)

Bases: *SaveMixin, ObjectDeleteMixin, RESTObject*

##### Parameters

- `manager` (*RESTManager*)
- `attrs` (*Dict[str, Any]*)
- `created_from_list` (*bool*)
- `lazy` (*bool*)



**class** gitlab.v4.objects.**BroadcastMessageManager**(gl: [Gitlab](#), parent: [RESTObject](#) | None = None)

Bases: [CRUDMixin](#), [RESTManager](#)

### Object Creation

Required attributes for object create:

- message

Optional attributes for object create:

- starts\_at
- ends\_at
- color
- font
- target\_access\_levels

### Object update

Optional attributes for object update:

- message
- starts\_at
- ends\_at
- color
- font
- target\_access\_levels

### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

**get**(id: str | int, lazy: bool = False, \*\*kwargs: Any) → [BroadcastMessage](#)

Retrieve a single object.

### Parameters

- **id** (str | int) – ID of the object to retrieve
- **lazy** (bool) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

### Returns

The generated [RESTObject](#).

### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

### Return type

[BroadcastMessage](#)

```
class gitlab.v4.objects.BulkImport(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: [RefreshMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

**entities:** [BulkImportEntityManager](#)

```
class gitlab.v4.objects.BulkImportAllEntity(manager: RESTManager, attrs: Dict[str, Any], *,
    created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

```
class gitlab.v4.objects.BulkImportAllEntityManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [RESTManager](#)

#### Object listing filters

- **sort**
- **status**

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | [None](#))

```
class gitlab.v4.objects.BulkImportEntity(manager: RESTManager, attrs: Dict[str, Any], *,
    created_from_list: bool = False, lazy: bool = False)
```

Bases: [RefreshMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

**class** gitlab.v4.objects.**BulkImportEntityManager**(gl: [Gitlab](#), parent: [RESTObject](#) | None = None)

Bases: [RetrieveMixin](#), [RESTManager](#)

#### Object listing filters

- sort
- status

#### Object Creation

#### Object update

##### Parameters

- gl ([Gitlab](#))
- parent ([RESTObject](#) | None)

**get**(id: str | int, lazy: bool = False, \*\*kwargs: Any) → [BulkImportEntity](#)

Retrieve a single object.

##### Parameters

- id (str | int) – ID of the object to retrieve
- lazy (bool) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- \*\*kwargs (Any) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated [RESTObject](#).

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

##### Return type

[BulkImportEntity](#)

**class** gitlab.v4.objects.**BulkImportManager**(gl: [Gitlab](#), parent: [RESTObject](#) | None = None)

Bases: [CreateMixin](#), [RetrieveMixin](#), [RESTManager](#)

#### Object listing filters

- sort
- status

#### Object Creation

Required attributes for object create:

- configuration
- entities

#### Object update

##### Parameters

- gl ([Gitlab](#))
- parent ([RESTObject](#) | None)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *BulkImport*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*BulkImport*

**class** gitlab.v4.objects.**CiLint**(*manager*: *RESTManager*, *attrs*: *Dict*[*str*, *Any*], \*, *created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**CiLintManager**(*gl*: *Gitlab*, *parent*: *RESTObject* | *None* = *None*)

Bases: *CreateMixin*, *RESTManager*

#### Object Creation

Required attributes for object create:

- **content**

Optional attributes for object create:

- **include\_merged\_yaml**
- **include\_jobs**

#### Object update

##### Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)

**validate**(\**args*: *Any*, *\*\*kwargs*: *Any*) → *None*

Raise an error if the CI Lint results are not valid.

This is a custom python-gitlab method to wrap lint endpoints.

#### Parameters

- **args** (*Any*)
- **kwargs** (*Any*)

**Return type**  
None

**class** gitlab.v4.objects.**CurrentUser**(*manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False*)

Bases: *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**emails:** *CurrentUserEmailManager*

**gpgkeys:** *CurrentUserGPGKeyManager*

**keys:** *CurrentUserKeyManager*

**runners:** *CurrentUserRunnerManager*

**status:** *CurrentUserStatusManager*

**class** gitlab.v4.objects.**CurrentUserEmail**(*manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False*)

Bases: *ObjectDeleteMixin, RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**CurrentUserEmailManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *RetrieveMixin, CreateMixin, DeleteMixin, RESTManager*

#### Object Creation

Required attributes for object create:

- **email**

#### Object update

##### Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *CurrentUserEmail*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*CurrentUserEmail*

```
class gitlab.v4.objects.CurrentUserGPGKey(manager: RESTManager, attrs: Dict[str, Any], *,
                                           created_from_list: bool = False, lazy: bool = False)
```

Bases: *ObjectDeleteMixin*, *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.CurrentUserGPGKeyManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *RetrieveMixin*, *CreateMixin*, *DeleteMixin*, *RESTManager*

#### Object Creation

Required attributes for object create:

- **key**

#### Object update

##### Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *CurrentUserGPGKey*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[CurrentUserGPGKey](#)

```
class gitlab.v4.objects.CurrentUserKey(manager: RESTManager, attrs: Dict[str, Any], *,
                                       created_from_list: bool = False, lazy: bool = False)
```

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

```
class gitlab.v4.objects.CurrentUserKeyManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [RetrieveMixin](#), [CreateMixin](#), [DeleteMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- **title**
- **key**

**Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | [None](#))

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → CurrentUserKey
```

Retrieve a single object.

**Parameters**

- **id** ([str](#) | [int](#)) – ID of the object to retrieve
- **lazy** ([bool](#)) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[CurrentUserKey](#)

**class** gitlab.v4.objects.**CurrentUserManager**(gl: [Gitlab](#), parent: [RESTObject](#) | None = None)

Bases: [GetWithoutIdMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

**get**(\*\*kwargs: Any) → [CurrentUser](#)

Retrieve a single object.

##### Parameters

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated [RESTObject](#)

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

##### Return type

[CurrentUser](#)

**class** gitlab.v4.objects.**CurrentUserRunner**(manager: [RESTManager](#), attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False)

Bases: [RESTObject](#)

##### Parameters

- **manager** ([RESTManager](#))
- **attrs** (Dict[str, Any])
- **created\_from\_list** (bool)
- **lazy** (bool)

**class** gitlab.v4.objects.**CurrentUserRunnerManager**(gl: [Gitlab](#), parent: [RESTObject](#) | None = None)

Bases: [CreateMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **runner\_type**

Optional attributes for object create:

- **group\_id**
- **project\_id**
- **description**
- **paused**
- **locked**
- **run\_untagged**



- tag\_list
- access\_level
- maximum\_timeout
- maintenance\_note

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**class** gitlab.v4.objects.**CurrentUserStatus**(*manager*: [RESTManager](#), *attrs*: *Dict[str, Any]*, \*, *created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: [SaveMixin](#), [RESTObject](#)

##### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**CurrentUserStatusManager**(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | *None* = *None*)

Bases: [GetWithoutIdMixin](#), [UpdateMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

Optional attributes for object update:

- emoji
- message

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(\*\**kwargs*: *Any*) → [CurrentUserStatus](#)

Retrieve a single object.

##### Parameters

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated [RESTObject](#)

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

##### Return type

[CurrentUserStatus](#)

```
class gitlab.v4.objects.DeployKey(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

```
class gitlab.v4.objects.DeployKeyManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | [None](#))

```
class gitlab.v4.objects.DeployToken(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

```
class gitlab.v4.objects.DeployTokenManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | [None](#))

```
class gitlab.v4.objects.Dockerfile(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

**class** gitlab.v4.objects.**DockerfileManager**(gl: [Gitlab](#), parent: [RESTObject](#) | None = None)

Bases: [RetrieveMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

**get**(id: str | int, lazy: bool = False, \*\*kwargs: Any) → [Dockerfile](#)

Retrieve a single object.

##### Parameters

- **id** (str | int) – ID of the object to retrieve
- **lazy** (bool) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated [RESTObject](#).

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

##### Return type

[Dockerfile](#)

**class** gitlab.v4.objects.**Event**(manager: [RESTManager](#), attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False)

Bases: [RESTObject](#)

##### Parameters

- **manager** ([RESTManager](#))
- **attrs** (Dict[str, Any])
- **created\_from\_list** (bool)
- **lazy** (bool)

**class** gitlab.v4.objects.**EventManager**(gl: [Gitlab](#), parent: [RESTObject](#) | None = None)

Bases: [ListMixin](#), [RESTManager](#)

#### Object listing filters

- action
- target\_type
- before
- after
- sort
- scope

## Object Creation

### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
class gitlab.v4.objects.Feature(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.FeatureManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [DeleteMixin](#), [RESTManager](#)

## Object Creation

### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
set(name: str, value: bool | int, feature_group: str | None = None, user: str | None = None, group: str | None = None, project: str | None = None, **kwargs: Any) → Feature
```

Create or update the object.

#### Parameters

- **name** (*str*) – The value to set for the object
- **value** (*bool* | *int*) – The value to set for the object
- **feature\_group** (*str* | *None*) – A feature group name
- **user** (*str* | *None*) – A GitLab username
- **group** (*str* | *None*) – A GitLab group
- **project** (*str* | *None*) – A GitLab project in form group/project
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabSetError** – If an error occurred

#### Returns

The created/updated attribute

#### Return type

[Feature](#)

```
class gitlab.v4.objects.GenericPackage(manager: RESTManager, attrs: Dict[str, Any], *,
                                       created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.GenericPackageManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
download(package_name: str, package_version: str, file_name: str, streamed: bool = False, action:
Callable[[bytes], None] | None = None, chunk_size: int = 1024, *, iterator: bool = False,
**kwargs: Any) → bytes | Iterator[Any] | None
```

Download a generic package.

#### Parameters

- **package\_name** (*str*) – The package name.
- **package\_version** (*str*) – The package version.
- **file\_name** (*str*) – The name of the file in the registry
- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk\_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable[[bytes], None] | None*) – Callable responsible of dealing with chunk of data
- **chunk\_size** (*int*) – Size of each chunk
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server failed to perform the request

#### Returns

The package content if streamed is False, None otherwise

#### Return type

*bytes | Iterator[Any] | None*

**upload**(*package\_name: str, package\_version: str, file\_name: str, path: str | Path | None = None, select: str | None = None, data: bytes | BinaryIO | None = None, \*\*kwargs: Any*) → *GenericPackage*

Upload a file as a generic package.

#### Parameters

- **package\_name** (*str*) – The package name. Must follow generic package name regex rules
- **package\_version** (*str*) – The package version. Must follow semantic version regex rules
- **file\_name** (*str*) – The name of the file as uploaded in the registry
- **path** (*str | Path | None*) – The path to a local file to upload
- **select** (*str | None*) – GitLab API accepts a value of ‘package\_file’
- **data** (*bytes | BinaryIO | None*)
- **kwargs** (*Any*)

#### Raises

- **GitlabConnectionError** – If the server cannot be reached
- **GitlabUploadError** – If the file upload fails
- **GitlabUploadError** – If path cannot be read
- **GitlabUploadError** – If both path and data are passed

#### Returns

An object storing the metadata of the uploaded package.

#### Return type

*GenericPackage*

[https://docs.gitlab.com/ee/user/packages/generic\\_packages/](https://docs.gitlab.com/ee/user/packages/generic_packages/)

**class** gitlab.v4.objects.**GeoNode**(*manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False*)

Bases: *SaveMixin, ObjectDeleteMixin, RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**repair**(*\*\*kwargs: Any*) → *None*

Repair the OAuth authentication of the geo node.

#### Parameters

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabRepairError** – If the server failed to perform the request

**Return type**

None

**status**(\*\*kwargs: [Any](#)) → Dict[str, [Any](#)]

Get the status of the geo node.

**Parameters****\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server failed to perform the request

**Returns**

The status of the geo node

**Return type**Dict[str, [Any](#)]**class** gitlab.v4.objects.**GeoNodeManager**(gl: [Gitlab](#), parent: [RESTObject](#) | None = None)Bases: [RetrieveMixin](#), [UpdateMixin](#), [DeleteMixin](#), [RESTManager](#)**Object Creation****Object update**

Optional attributes for object update:

- enabled
- url
- files\_max\_capacity
- repos\_max\_capacity

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) / None)

**current\_failures**(\*\*kwargs: [Any](#)) → List[Dict[str, [Any](#)]]

Get the list of failures on the current geo node.

**Parameters****\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server failed to perform the request

**Returns**

The list of failures

**Return type**List[Dict[str, [Any](#)]]

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *GeoNode*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*GeoNode*

**status**(*\*\*kwargs*: *Any*) → List[Dict[*str*, *Any*]]

Get the status of all the geo nodes.

#### Parameters

- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server failed to perform the request

#### Returns

The status of all the geo nodes

#### Return type

List[Dict[*str*, *Any*]]

**class** gitlab.v4.objects.**Gitignore**(*manager*: *RESTManager*, *attrs*: Dict[*str*, *Any*], \*, *created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (Dict[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**GitignoreManager**(*gl*: *Gitlab*, *parent*: *RESTObject* | *None* = *None*)

Bases: *RetrieveMixin*, *RESTManager*

#### Object Creation

#### Object update

#### Parameters

- **gl** (*Gitlab*)



- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *Gitignore*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated *RESTObject*.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*Gitignore*

```
class gitlab.v4.objects.Gitlabciyaml(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.GitlabciyamlManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *Gitlabciyaml*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated *RESTObject*.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type***Gitlabciyaml*

```
class gitlab.v4.objects.Group(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: *SaveMixin, ObjectDeleteMixin, RESTObject*

**Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**access\_tokens:** *GroupAccessTokenManager*

**accessrequests:** *GroupAccessRequestManager*

**audit\_events:** *GroupAuditEventManager*

**badges:** *GroupBadgeManager*

**billable\_members:** *GroupBillableMemberManager*

**boards:** *GroupBoardManager*

**clusters:** *GroupClusterManager*

**customattributes:** *GroupCustomAttributeManager*

**deploytokens:** *GroupDeployTokenManager*

**descendant\_groups:** *GroupDescendantGroupManager*

**epics:** *GroupEpicManager*

**exports:** *GroupExportManager*

**hooks:** *GroupHookManager*

**imports:** *GroupImportManager*

**invitations:** *GroupInvitationManager*

**issues:** *GroupIssueManager*

**issues\_statistics:** *GroupIssuesStatisticsManager*

**iterations:** *GroupIterationManager*

**labels:** *GroupLabelManager*

**ldap\_group\_links:** *GroupLDAPGroupLinkManager*

**ldap\_sync**(\*\*kwargs: Any) → None

Sync LDAP groups.

**Parameters**

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

**Return type**

None

**members:** *GroupMemberManager*

**members\_all:** *GroupMemberAllManager*

**mergerequests:** *GroupMergeRequestManager*

**milestones:** *GroupMilestoneManager*

**notificationsettings:** *GroupNotificationSettingsManager*

**packages:** *GroupPackageManager*

**projects:** *GroupProjectManager*

**pushrules:** *GroupPushRulesManager*

**registry\_repositories:** *GroupRegistryRepositoryManager*

**restore**(\*\*kwargs: Any) → None

Restore a group marked for deletion..

**Parameters**

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabRestoreError** – If the server failed to perform the request

**Return type**

None

**runners:** *GroupRunnerManager*

**saml\_group\_links:** *GroupSAMLGroupLinkManager*

**search**(scope: str, search: str, \*\*kwargs: Any) → *GitlabList* | List[Dict[str, Any]]

Search the group resources matching the provided string.

**Parameters**

- **scope** (str) – Scope of the search
- **search** (str) – Search string
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabSearchError** – If the server failed to perform the request

**Returns**

A list of dicts describing the resources found.

**Return type**

[GitlabList](#) | *List[Dict[str, Any]]*

**share**(*group\_id: int, group\_access: int, expires\_at: str | None = None, \*\*kwargs: Any*) → None

Share the group with a group.

**Parameters**

- **group\_id** (*int*) – ID of the group.
- **group\_access** (*int*) – Access level for the group.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)
- **expires\_at** (*str | None*)
- **\*\*kwargs**

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server failed to perform the request

**Returns**

Group

**Return type**

None

**shared\_projects:** [SharedProjectManager](#)

**subgroups:** [GroupSubgroupManager](#)

**transfer**(*group\_id: int | None = None, \*\*kwargs: Any*) → None

Transfer the group to a new parent group or make it a top-level group.

Requires GitLab 14.6.

**Parameters**

- **group\_id** (*int | None*) – ID of the new parent group. When not specified, the group to transfer is instead turned into a top-level group.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGroupTransferError** – If the group could not be transferred

**Return type**

None

**transfer\_project**(*project\_id: int, \*\*kwargs: Any*) → None

Transfer a project to this group.

**Parameters**

- **to\_project\_id** – ID of the project to transfer
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)
- **project\_id** (*int*)
- **\*\*kwargs**

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabTransferProjectError** – If the project could not be transferred

**Return type**

None

**unshare**(*group\_id: int, \*\*kwargs: Any*) → None

Delete a shared group link within a group.

**Parameters**

- **group\_id** (*int*) – ID of the group.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the server failed to perform the request

**Return type**

None

**variables:** *GroupVariableManager*

**wikis:** *GroupWikiManager*

```
class gitlab.v4.objects.GroupAccessRequest(manager: RESTManager, attrs: Dict[str, Any], *,
                                           created_from_list: bool = False, lazy: bool = False)
```

Bases: *AccessRequestMixin, ObjectDeleteMixin, RESTObject*

**Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.GroupAccessRequestManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *ListMixin, CreateMixin, DeleteMixin, RESTManager*

**Object Creation****Object update****Parameters**

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

```
class gitlab.v4.objects.GroupAccessToken(manager: RESTManager, attrs: Dict[str, Any], *,
                                         created_from_list: bool = False, lazy: bool = False)
```

Bases: [ObjectDeleteMixin](#), [ObjectRotateMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

```
class gitlab.v4.objects.GroupAccessTokenManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [CreateMixin](#), [DeleteMixin](#), [RetrieveMixin](#), [RotateMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **name**
- **scopes**

Optional attributes for object create:

- **access\_level**
- **expires\_at**

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | [None](#))

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → GroupAccessToken
```

Retrieve a single object.

##### Parameters

- **id** ([str](#) | [int](#)) – ID of the object to retrieve
- **lazy** ([bool](#)) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated [RESTObject](#).

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

##### Return type

[GroupAccessToken](#)

```
class gitlab.v4.objects.GroupAuditEvent(manager: RESTManager, attrs: Dict[str, Any], *,
                                         created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**GroupAuditEventManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: [RetrieveMixin](#), [RESTManager](#)

**Object listing filters**

- **created\_after**
- **created\_before**

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id: str | int, lazy: bool = False, \*\*kwargs: Any*) → [GroupAuditEvent](#)

Retrieve a single object.

**Parameters**

- **id** (*str | int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated [RESTObject](#).

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[GroupAuditEvent](#)

**class** gitlab.v4.objects.**GroupBadge**(*manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False*)

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**GroupBadgeManager**(*gl*: Gitlab, *parent*: RESTObject | None = None)

Bases: *BadgeRenderMixin*, *CRUDMixin*, *RESTManager*

#### Object Creation

Required attributes for object create:

- *link\_url*
- *image\_url*

#### Object update

Optional attributes for object update:

- *link\_url*
- *image\_url*

#### Parameters

- *gl* (Gitlab)
- *parent* (RESTObject | None)

**get**(*id*: str | int, *lazy*: bool = False, *\*\*kwargs*: Any) → *GroupBadge*

Retrieve a single object.

#### Parameters

- *id* (str | int) – ID of the object to retrieve
- *lazy* (bool) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- *\*\*kwargs* (Any) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*GroupBadge*

**class** gitlab.v4.objects.**GroupBillableMember**(*manager*: RESTManager, *attrs*: Dict[str, Any], \*, *created\_from\_list*: bool = False, *lazy*: bool = False)

Bases: *ObjectDeleteMixin*, *RESTObject*

#### Parameters

- *manager* (RESTManager)
- *attrs* (Dict[str, Any])
- *created\_from\_list* (bool)
- *lazy* (bool)

**memberships**: *GroupBillableMemberMembershipManager*



**class** gitlab.v4.objects.**GroupBillableMemberManager**(gl: [Gitlab](#), parent: [RESTObject](#) | None = None)

Bases: [ListMixin](#), [DeleteMixin](#), [RESTManager](#)

Object listing filters

- search
- sort

Object Creation

Object update

Parameters

- gl ([Gitlab](#))
- parent ([RESTObject](#) | None)

**class** gitlab.v4.objects.**GroupBillableMemberMembership**(manager: [RESTManager](#), attrs: [Dict](#)[[str](#), [Any](#)], \*, created\_from\_list: bool = False, lazy: bool = False)

Bases: [RESTObject](#)

Parameters

- manager ([RESTManager](#))
- attrs ([Dict](#)[[str](#), [Any](#)])
- created\_from\_list (bool)
- lazy (bool)

**class** gitlab.v4.objects.**GroupBillableMemberMembershipManager**(gl: [Gitlab](#), parent: [RESTObject](#) | None = None)

Bases: [ListMixin](#), [RESTManager](#)

Object Creation

Object update

Parameters

- gl ([Gitlab](#))
- parent ([RESTObject](#) | None)

**class** gitlab.v4.objects.**GroupBoard**(manager: [RESTManager](#), attrs: [Dict](#)[[str](#), [Any](#)], \*, created\_from\_list: bool = False, lazy: bool = False)

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

Parameters

- manager ([RESTManager](#))
- attrs ([Dict](#)[[str](#), [Any](#)])
- created\_from\_list (bool)
- lazy (bool)

lists: [GroupBoardListManager](#)

```
class gitlab.v4.objects.GroupBoardList(manager: RESTManager, attrs: Dict[str, Any], *,
                                       created_from_list: bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

```
class gitlab.v4.objects.GroupBoardListManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [CRUDMixin](#), [RESTManager](#)

#### Object Creation

Mutually exclusive attributes for object create:

- **label\_id**
- **assignee\_id**
- **milestone\_id**

#### Object update

Required attributes for object update:

- **position**

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | [None](#))

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → GroupBoardList
```

Retrieve a single object.

#### Parameters

- **id** ([str](#) | [int](#)) – ID of the object to retrieve
- **lazy** ([bool](#)) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[GroupBoardList](#)

**class** gitlab.v4.objects.**GroupBoardManager**(gl: [Gitlab](#), parent: [RESTObject](#) | None = None)

Bases: [CRUDMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- name

#### Object update

##### Parameters

- gl ([Gitlab](#))
- parent ([RESTObject](#) | None)

**get**(id: str | int, lazy: bool = False, \*\*kwargs: Any) → [GroupBoard](#)

Retrieve a single object.

##### Parameters

- id (str | int) – ID of the object to retrieve
- lazy (bool) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- \*\*kwargs (Any) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated [RESTObject](#).

##### Raises

- [GitlabAuthenticationError](#) – If authentication is not correct
- [GitlabGetError](#) – If the server cannot perform the request

##### Return type

[GroupBoard](#)

**class** gitlab.v4.objects.**GroupCluster**(manager: [RESTManager](#), attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False)

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

##### Parameters

- manager ([RESTManager](#))
- attrs (Dict[str, Any])
- created\_from\_list (bool)
- lazy (bool)

**class** gitlab.v4.objects.**GroupClusterManager**(gl: [Gitlab](#), parent: [RESTObject](#) | None = None)

Bases: [CRUDMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- name
- platform\_kubernetes\_attributes

Optional attributes for object create:

- domain
- enabled
- managed
- environment\_scope

### Object update

Optional attributes for object update:

- name
- domain
- management\_project\_id
- platform\_kubernetes\_attributes
- environment\_scope

### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**create**(*data*: *Dict*[*str*, *Any*] | *None* = *None*, *\*\*kwargs*: *Any*) → *GroupCluster*

Create a new object.

### Parameters

- **data** (*Dict*[*str*, *Any*] | *None*) – Parameters to send to the server to create the resource
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo` or `'ref_name'`, `'stage'`, `'name'`, `'all'`)

### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

### Returns

A new instance of the `manage object class` build with the data sent by the server

### Return type

[GroupCluster](#)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *GroupCluster*

Retrieve a single object.

### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If *True*, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo`)

### Returns

The generated `RESTObject`.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type***GroupCluster*

```
class gitlab.v4.objects.GroupCustomAttribute(manager: RESTManager, attrs: Dict[str, Any], *,
                                             created_from_list: bool = False, lazy: bool = False)
```

Bases: *ObjectDeleteMixin*, *RESTObject*

**Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.GroupCustomAttributeManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *RetrieveMixin*, *SetMixin*, *DeleteMixin*, *RESTManager*

**Object Creation****Object update****Parameters**

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → GroupCustomAttribute
```

Retrieve a single object.

**Parameters**

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type***GroupCustomAttribute*

```
class gitlab.v4.objects.GroupDeployToken(manager: RESTManager, attrs: Dict[str, Any], *,
                                          created_from_list: bool = False, lazy: bool = False)
```

Bases: *ObjectDeleteMixin*, *RESTObject*

**Parameters**

- **manager** (*RESTManager*)

- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**GroupDeployTokenManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *RetrieveMixin, CreateMixin, DeleteMixin, RESTManager*

#### Object listing filters

- **scopes**

#### Object Creation

Required attributes for object create:

- **name**
- **scopes**

Optional attributes for object create:

- **expires\_at**
- **username**

#### Object update

##### Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

**get**(*id: str | int, lazy: bool = False, \*\*kwargs: Any*) → *GroupDeployToken*

Retrieve a single object.

##### Parameters

- **id** (*str | int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated RESTObject.

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

##### Return type

*GroupDeployToken*

**class** gitlab.v4.objects.**GroupDescendantGroup**(*manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False*)

Bases: *RESTObject*

##### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)

- `created_from_list` (*bool*)
- `lazy` (*bool*)

**class** gitlab.v4.objects.**GroupDescendantGroupManager**(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | *None* = *None*)

Bases: [GroupSubgroupManager](#)

This manager inherits from [GroupSubgroupManager](#) as descendant groups share all attributes with subgroups, except the path and object class.

#### Object listing filters

- `skip_groups`
- `all_available`
- `search`
- `order_by`
- `sort`
- `statistics`
- `owned`
- `with_custom_attributes`
- `min_access_level`

#### Object Creation

#### Object update

##### Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) | *None*)

**class** gitlab.v4.objects.**GroupEpic**(*manager*: [RESTManager](#), *attrs*: *Dict*[*str*, *Any*], \*, *created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: [ObjectDeleteMixin](#), [SaveMixin](#), [RESTObject](#)

##### Parameters

- `manager` ([RESTManager](#))
- `attrs` (*Dict*[*str*, *Any*])
- `created_from_list` (*bool*)
- `lazy` (*bool*)

**issues**: [GroupEpicIssueManager](#)

**notes**: [GroupEpicNoteManager](#)

**resourcelabelevents**: [GroupEpicResourceLabelEventManager](#)

**class** gitlab.v4.objects.**GroupEpicAwardEmoji**(*manager*: [RESTManager](#), *attrs*: *Dict*[*str*, *Any*], \*, *created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

##### Parameters

- `manager` ([RESTManager](#))

- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**GroupEpicAwardEmojiManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *NoUpdateMixin, RESTManager*

#### Object Creation

Required attributes for object create:

- **name**

#### Object update

##### Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

**get**(*id: str | int, lazy: bool = False, \*\*kwargs: Any*) → *GroupEpicAwardEmoji*

Retrieve a single object.

##### Parameters

- **id** (*str | int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated RESTObject.

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

##### Return type

*GroupEpicAwardEmoji*

**class** gitlab.v4.objects.**GroupEpicDiscussionNote**(*manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False*)

Bases: *SaveMixin, ObjectDeleteMixin, RESTObject*

##### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**GroupEpicDiscussionNoteManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *GetMixin, CreateMixin, UpdateMixin, DeleteMixin, RESTManager*

#### Object Creation



Required attributes for object create:

- `body`

Optional attributes for object create:

- `created_at`

### Object update

Required attributes for object update:

- `body`

#### Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) / `None`)

`get(id: str | int, lazy: bool = False, **kwargs: Any) → GroupEpicDiscussionNote`

Retrieve a single object.

#### Parameters

- `id` (`str` / `int`) – ID of the object to retrieve
- `lazy` (`bool`) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- `**kwargs` (`Any`) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated [RESTObject](#).

#### Raises

- `GitlabAuthenticationError` – If authentication is not correct
- `GitlabGetError` – If the server cannot perform the request

#### Return type

[GroupEpicDiscussionNote](#)

```
class gitlab.v4.objects.GroupEpicIssue(manager: RESTManager, attrs: Dict[str, Any], *,
                                       created_from_list: bool = False, lazy: bool = False)
```

Bases: [ObjectDeleteMixin](#), [SaveMixin](#), [RESTObject](#)

#### Parameters

- `manager` ([GroupEpicIssueManager](#))
- `attrs` (`Dict[str, Any]`)
- `created_from_list` (`bool`)
- `lazy` (`bool`)

`manager:` [GroupEpicIssueManager](#)

`save(**kwargs: Any) → None`

Save the changes made to the object to the server.

The object is updated to match what the server returns.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUpdateError** – If the server cannot perform the request

**Return type**

None

```
class gitlab.v4.objects.GroupEpicIssueManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *ListMixin*, *CreateMixin*, *UpdateMixin*, *DeleteMixin*, *RESTManager*

**Object Creation**

Required attributes for object create:

- `issue_id`

**Object update**

Optional attributes for object update:

- `move_before_id`
- `move_after_id`

**Parameters**

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)

```
create(data: Dict[str, Any] | None = None, **kwargs: Any) → GroupEpicIssue
```

Create a new object.

**Parameters**

- **data** (*Dict[str, Any]* | *None*) – Parameters to send to the server to create the resource
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

**Returns**

A new instance of the manage object class build with  
the data sent by the server

**Return type**

*GroupEpicIssue*

```
class gitlab.v4.objects.GroupEpicManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *CRUDMixin*, *RESTManager*

**Object listing filters**

- `author_id`
- `labels`

- `order_by`
- `sort`
- `search`

### Object Creation

Required attributes for object create:

- `title`

Optional attributes for object create:

- `labels`
- `description`
- `start_date`
- `end_date`

### Object update

Optional attributes for object update:

- `title`
- `labels`
- `description`
- `start_date`
- `end_date`

### Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) / `None`)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *GroupEpic*

Retrieve a single object.

### Parameters

- **id** (*str* / *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo`)

### Returns

The generated `RESTObject`.

### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

### Return type

[GroupEpic](#)

```
class gitlab.v4.objects.GroupEpicNote(manager: RESTManager, attrs: Dict[str, Any], *,
                                     created_from_list: bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

awardemojis: [GroupEpicNoteAwardEmojiManager](#)

```
class gitlab.v4.objects.GroupEpicNoteAwardEmoji(manager: RESTManager, attrs: Dict[str, Any], *,
                                                created_from_list: bool = False, lazy: bool = False)
```

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

```
class gitlab.v4.objects.GroupEpicNoteAwardEmojiManager(gl: Gitlab, parent: RESTObject | None =
                                                         None)
```

Bases: [NoUpdateMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **name**

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | [None](#))

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → GroupEpicNoteAwardEmoji
```

Retrieve a single object.

##### Parameters

- **id** ([str](#) | [int](#)) – ID of the object to retrieve
- **lazy** ([bool](#)) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated [RESTObject](#).

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct

- **GitlabGetError** – If the server cannot perform the request

**Return type**[GroupEpicNoteAwardEmoji](#)

**class** gitlab.v4.objects.**GroupEpicNoteManager**(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | *None* = *None*)

Bases: [CRUDMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- *body*

Optional attributes for object create:

- *created\_at*

**Object update**

Required attributes for object update:

- *body*

**Parameters**

- *gl* ([Gitlab](#))
- *parent* ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [GroupEpicNote](#)

Retrieve a single object.

**Parameters**

- *id* (*str* | *int*) – ID of the object to retrieve
- *lazy* (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- *\*\*kwargs* (*Any*) – Extra options to send to the server (e.g. *sudo*)

**Returns**

The generated [RESTObject](#).

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**[GroupEpicNote](#)

**class** gitlab.v4.objects.**GroupEpicResourceLabelEvent**(*manager*: [RESTManager](#), *attrs*: *Dict*[*str*, *Any*], *\*, created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: [RESTObject](#)

**Parameters**

- *manager* ([RESTManager](#))
- *attrs* (*Dict*[*str*, *Any*])
- *created\_from\_list* (*bool*)

- **lazy** (*bool*)

```
class gitlab.v4.objects.GroupEpicResourceLabelEventManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → GroupEpicResourceLabelEvent
```

Retrieve a single object.

##### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated [RESTObject](#).

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

##### Return type

[GroupEpicResourceLabelEvent](#)

```
class gitlab.v4.objects.GroupExport(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [DownloadMixin](#), [RESTObject](#)

##### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.GroupExportManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [GetWithoutIdMixin](#), [CreateMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*\*\*kwargs*: *Any*) → *GroupExport*

Retrieve a single object.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

*GroupExport*

**class** gitlab.v4.objects.**GroupHook**(*manager*: *RESTManager*, *attrs*: *Dict[str, Any]*, \*, *created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

**Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**GroupHookManager**(*gl*: *Gitlab*, *parent*: *RESTObject* | *None* = *None*)

Bases: *CRUDMixin*, *RESTManager*

**Object Creation**

Required attributes for object create:

- url

Optional attributes for object create:

- push\_events
- issues\_events
- confidential\_issues\_events
- merge\_requests\_events
- tag\_push\_events
- note\_events
- confidential\_note\_events
- job\_events
- pipeline\_events
- wiki\_page\_events
- deployment\_events
- releases\_events
- subgroup\_events

- `enable_ssl_verification`
- `token`

### Object update

Required attributes for object update:

- `url`

Optional attributes for object update:

- `push_events`
- `issues_events`
- `confidential_issues_events`
- `merge_requests_events`
- `tag_push_events`
- `note_events`
- `confidential_note_events`
- `job_events`
- `pipeline_events`
- `wiki_page_events`
- `deployment_events`
- `releases_events`
- `subgroup_events`
- `enable_ssl_verification`
- `token`

### Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) / `None`)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *GroupHook*

Retrieve a single object.

### Parameters

- `id` (*str* / *int*) – ID of the object to retrieve
- `lazy` (*bool*) – If `True`, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- `**kwargs` (*Any*) – Extra options to send to the server (e.g. `sudo`)

### Returns

The generated `RESTObject`.

### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request



**Return type**[GroupHook](#)

```
class gitlab.v4.objects.GroupImport(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[str, Any])
- **created\_from\_list** (bool)
- **lazy** (bool)

```
class gitlab.v4.objects.GroupImportManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [GetWithoutIdMixin](#), [RESTManager](#)

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
get(**kwargs: Any) → GroupImport
```

Retrieve a single object.

**Parameters**

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated [RESTObject](#)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**[GroupImport](#)

```
class gitlab.v4.objects.GroupInvitation(manager: RESTManager, attrs: Dict[str, Any], *,
    created_from_list: bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[str, Any])
- **created\_from\_list** (bool)
- **lazy** (bool)

```
class gitlab.v4.objects.GroupInvitationManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [InvitationMixin](#), [RESTManager](#)

**Object listing filters**

- `query`

### Object Creation

Required attributes for object create:

- `access_level`

Optional attributes for object create:

- `expires_at`
- `invite_source`
- `tasks_to_be_done`
- `tasks_project_id`

Mutually exclusive attributes for object create:

- `email`
- `user_id`

### Object update

Optional attributes for object update:

- `access_level`
- `expires_at`

#### Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) | `None`)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *GroupInvitation*

Retrieve a single object.

#### Parameters

- `id` (*str* | *int*) – ID of the object to retrieve
- `lazy` (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- `**kwargs` (*Any*) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[GroupInvitation](#)

```
class gitlab.v4.objects.GroupIssue(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**GroupIssueManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: [ListMixin](#), [RESTManager](#)

#### Object listing filters

- state
- labels
- milestone
- order\_by
- sort
- iids
- author\_id
- iteration\_id
- assignee\_id
- my\_reaction\_emoji
- search
- created\_after
- created\_before
- updated\_after
- updated\_before

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**class** gitlab.v4.objects.**GroupIssuesStatistics**(*manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False*)

Bases: [RefreshMixin](#), [RESTObject](#)

##### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.GroupIssuesStatisticsManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [GetWithoutIdMixin](#), [RESTManager](#)

Object listing filters

- `iids`

Object Creation

Object update

Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) | `None`)

```
get(**kwargs: Any) → GroupIssuesStatistics
```

Retrieve a single object.

Parameters

**`**kwargs`** ([Any](#)) – Extra options to send to the server (e.g. `sudo`)

Returns

The generated [RESTObject](#)

Raises

- **`GitlabAuthenticationError`** – If authentication is not correct
- **`GitlabGetError`** – If the server cannot perform the request

Return type

[GroupIssuesStatistics](#)

```
class gitlab.v4.objects.GroupIteration(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

Parameters

- `manager` ([RESTManager](#))
- `attrs` ([Dict](#)[`str`, [Any](#)])
- `created_from_list` (`bool`)
- `lazy` (`bool`)

```
class gitlab.v4.objects.GroupIterationManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [RESTManager](#)

Object listing filters

- `state`
- `search`
- `include_ancestors`

Object Creation

Object update

Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**class** gitlab.v4.objects.[GroupLDAPGroupLink](#)(*manager*: [RESTManager](#), *attrs*: *Dict[str, Any]*, \*, *created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**delete**(\*\**kwargs*: *Any*) → *None*

Delete the LDAP group link from the server.

#### Parameters

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the server cannot perform the request

#### Return type

*None*

**class** gitlab.v4.objects.[GroupLDAPGroupLinkManager](#)(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | *None* = *None*)

Bases: [ListMixin](#), [CreateMixin](#), [DeleteMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **provider**
- **group\_access**

Mutually exclusive attributes for object create:

- **cn**
- **filter**

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**class** gitlab.v4.objects.[GroupLabel](#)(*manager*: [RESTManager](#), *attrs*: *Dict[str, Any]*, \*, *created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: [SubscribableMixin](#), [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([GroupLabelManager](#))
- **attrs** (*Dict[str, Any]*)

- `created_from_list` (*bool*)
- `lazy` (*bool*)

**manager:** *GroupLabelManager*

**save**(\*\**kwargs*: *Any*) → *None*

Saves the changes made to the object to the server.

The object is updated to match what the server returns.

#### Parameters

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo`)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct.
- **GitlabUpdateError** – If the server cannot perform the request.

#### Return type

*None*

**class** `gitlab.v4.objects.GroupLabelManager`(*gl*: *Gitlab*, *parent*: *RESTObject* | *None* = *None*)

Bases: *RetrieveMixin*, *CreateMixin*, *UpdateMixin*, *DeleteMixin*, *RESTManager*

#### Object Creation

Required attributes for object create:

- `name`
- `color`

Optional attributes for object create:

- `description`
- `priority`

#### Object update

Required attributes for object update:

- `name`

Optional attributes for object update:

- `new_name`
- `color`
- `description`
- `priority`

#### Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, \*\**kwargs*: *Any*) → *GroupLabel*

Retrieve a single object.

#### Parameters

- **id** (*str* / *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

*GroupLabel*

**update**(*name: str | None, new\_data: Dict[str, Any] | None = None, \*\*kwargs: Any*) → *Dict[str, Any]*

Update a Label on the server.

**Parameters**

- **name** (*str* / *None*) – The name of the label
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)
- **new\_data** (*Dict[str, Any] | None*)
- **\*\*kwargs**

**Return type**

*Dict[str, Any]*

**class** gitlab.v4.objects.**GroupManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *CRUDMixin, RESTManager*

**Object listing filters**

- skip\_groups
- all\_available
- search
- order\_by
- sort
- statistics
- owned
- with\_custom\_attributes
- min\_access\_level
- top\_level\_only

**Object Creation**

Required attributes for object create:

- name
- path

Optional attributes for object create:

- description
- membership\_lock
- visibility
- share\_with\_group\_lock
- require\_two\_factor\_authentication
- two\_factor\_grace\_period
- project\_creation\_level
- auto\_devops\_enabled
- subgroup\_creation\_level
- emails\_disabled
- avatar
- mentions\_disabled
- lfs\_enabled
- request\_access\_enabled
- parent\_id
- default\_branch\_protection
- shared\_runners\_minutes\_limit
- extra\_shared\_runners\_minutes\_limit

#### Object update

Optional attributes for object update:

- name
- path
- description
- membership\_lock
- share\_with\_group\_lock
- visibility
- require\_two\_factor\_authentication
- two\_factor\_grace\_period
- project\_creation\_level
- auto\_devops\_enabled
- subgroup\_creation\_level
- emails\_disabled
- avatar
- mentions\_disabled
- lfs\_enabled
- request\_access\_enabled



- `default_branch_protection`
- `file_template_project_id`
- `shared_runners_minutes_limit`
- `extra_shared_runners_minutes_limit`
- `prevent_forking_outside_group`
- `shared_runners_setting`

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: [Any](#)) → [Group](#)

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[Group](#)

**import\_group**(*file*: [BinaryIO](#), *path*: *str*, *name*: *str*, *parent\_id*: *int* | *str* | *None* = *None*, *\*\*kwargs*: [Any](#)) → [Dict](#)[*str*, [Any](#)] | [Response](#)

Import a group from an archive file.

#### Parameters

- **file** ([BinaryIO](#)) – Data or file object containing the group
- **path** (*str*) – The path for the new group to be imported.
- **name** (*str*) – The name for the new group.
- **parent\_id** (*int* | *str* | *None*) – ID of a parent group that the group will be imported into.
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. `sudo`)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabImportError** – If the server failed to perform the request

#### Returns

A representation of the import status.

**Return type***Dict[str, Any] | Response*

```
class gitlab.v4.objects.GroupMember(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: *SaveMixin, ObjectDeleteMixin, RESTObject*

**Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.GroupMemberAll(manager: RESTManager, attrs: Dict[str, Any], *,
    created_from_list: bool = False, lazy: bool = False)
```

Bases: *RESTObject*

**Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.GroupMemberAllManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *RetrieveMixin, RESTManager*

**Object Creation****Object update****Parameters**

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → GroupMemberAll
```

Retrieve a single object.

**Parameters**

- **id** (*str | int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type***GroupMemberAll*

**class** gitlab.v4.objects.**GroupMemberManager**(gl: [Gitlab](#), parent: [RESTObject](#) | None = None)

Bases: [CRUDMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- `access_level`
- `user_id`

Optional attributes for object create:

- `expires_at`
- `tasks_to_be_done`

#### Object update

Required attributes for object update:

- `access_level`

Optional attributes for object update:

- `expires_at`

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

**get**(id: str | int, lazy: bool = False, \*\*kwargs: Any) → [GroupMember](#)

Retrieve a single object.

#### Parameters

- **id** (str | int) – ID of the object to retrieve
- **lazy** (bool) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[GroupMember](#)

**class** gitlab.v4.objects.**GroupMergeRequest**(manager: [RESTManager](#), attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False)

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (Dict[str, Any])

- `created_from_list` (*bool*)
- `lazy` (*bool*)

**class** gitlab.v4.objects.**GroupMergeRequestManager**(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | *None* = *None*)

Bases: [ListMixin](#), [RESTManager](#)

#### Object listing filters

- `state`
- `order_by`
- `sort`
- `milestone`
- `view`
- `labels`
- `created_after`
- `created_before`
- `updated_after`
- `updated_before`
- `scope`
- `author_id`
- `assignee_id`
- `approver_ids`
- `approved_by_ids`
- `my_reaction_emoji`
- `source_branch`
- `target_branch`
- `search`
- `wip`

#### Object Creation

#### Object update

##### Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) | *None*)

**class** gitlab.v4.objects.**GroupMilestone**(*manager*: [RESTManager](#), *attrs*: *Dict*[*str*, *Any*], \*,  
*created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

##### Parameters

- `manager` ([RESTManager](#))
- `attrs` (*Dict*[*str*, *Any*])
- `created_from_list` (*bool*)

- **lazy** (*bool*)

**issues**(\*\*kwargs: *Any*) → *RESTObjectList*

List issues related to this milestone.

#### Parameters

- **all** – If True, return all the items, without pagination
- **per\_page** – Number of items to retrieve per request
- **page** – ID of the page to return (starts with page 1)
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the list could not be retrieved

#### Returns

The list of issues

#### Return type

*RESTObjectList*

**merge\_requests**(\*\*kwargs: *Any*) → *RESTObjectList*

List the merge requests related to this milestone.

#### Parameters

- **all** – If True, return all the items, without pagination
- **per\_page** – Number of items to retrieve per request
- **page** – ID of the page to return (starts with page 1)
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the list could not be retrieved

#### Returns

The list of merge requests

#### Return type

*RESTObjectList*

**class** gitlab.v4.objects.**GroupMilestoneManager**(gl: *Gitlab*, parent: *RESTObject* | *None* = *None*)

Bases: *CRUDMixin*, *RESTManager*

#### Object listing filters

- **iids**
- **state**
- **search**

#### Object Creation

Required attributes for object create:

- **title**

Optional attributes for object create:

- `description`
- `due_date`
- `start_date`

#### Object update

Optional attributes for object update:

- `title`
- `description`
- `due_date`
- `start_date`
- `state_event`

#### Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) / `None`)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *GroupMilestone*

Retrieve a single object.

#### Parameters

- `id` (*str* / *int*) – ID of the object to retrieve
- `lazy` (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- `**kwargs` (*Any*) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[GroupMilestone](#)

```
class gitlab.v4.objects.GroupNotificationSettings(manager: RESTManager, attrs: Dict[str, Any], *,
                                                  created_from_list: bool = False, lazy: bool =
                                                  False)
```

Bases: [NotificationSettings](#)

#### Parameters

- `manager` ([RESTManager](#))
- `attrs` (*Dict*[*str*, *Any*])
- `created_from_list` (*bool*)
- `lazy` (*bool*)

```
class gitlab.v4.objects.GroupNotificationSettingsManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *NotificationSettingsManager*

#### Object Creation

#### Object update

Optional attributes for object update:

- level
- notification\_email
- new\_note
- new\_issue
- reopen\_issue
- close\_issue
- reassign\_issue
- new\_merge\_request
- reopen\_merge\_request
- close\_merge\_request
- reassign\_merge\_request
- merge\_merge\_request

#### Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)

```
get(**kwargs: Any) → GroupNotificationSettings
```

Retrieve a single object.

#### Parameters

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*GroupNotificationSettings*

```
class gitlab.v4.objects.GroupPackage(manager: RESTManager, attrs: Dict[str, Any], *,
                                     created_from_list: bool = False, lazy: bool = False)
```

Bases: *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)

- `created_from_list` (*bool*)
- `lazy` (*bool*)

**class** gitlab.v4.objects.**GroupPackageManager**(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | *None* = *None*)

Bases: [ListMixin](#), [RESTManager](#)

#### Object listing filters

- `exclude_subgroups`
- `order_by`
- `sort`
- `package_type`
- `package_name`

#### Object Creation

#### Object update

##### Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) | *None*)

**class** gitlab.v4.objects.**GroupProject**(*manager*: [RESTManager](#), *attrs*: *Dict*[*str*, [Any](#)], \*,  
*created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: [RESTObject](#)

##### Parameters

- `manager` ([RESTManager](#))
- `attrs` (*Dict*[*str*, [Any](#)])
- `created_from_list` (*bool*)
- `lazy` (*bool*)

**class** gitlab.v4.objects.**GroupProjectManager**(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | *None* = *None*)

Bases: [ListMixin](#), [RESTManager](#)

#### Object listing filters

- `archived`
- `visibility`
- `order_by`
- `sort`
- `search`
- `simple`
- `owned`
- `starred`
- `with_custom_attributes`
- `include_subgroups`
- `with_issues_enabled`



- `with_merge_requests_enabled`
- `with_shared`
- `min_access_level`
- `with_security_reports`

### Object Creation

### Object update

#### Parameters

- `gl` (`Gitlab`)
- `parent` (`RESTObject` | `None`)

```
class gitlab.v4.objects.GroupPushRules(manager: RESTManager, attrs: Dict[str, Any], *,
                                       created_from_list: bool = False, lazy: bool = False)
```

Bases: `SaveMixin`, `ObjectDeleteMixin`, `RESTObject`

#### Parameters

- `manager` (`RESTManager`)
- `attrs` (`Dict[str, Any]`)
- `created_from_list` (`bool`)
- `lazy` (`bool`)

```
class gitlab.v4.objects.GroupPushRulesManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: `GetWithoutIdMixin`, `CreateMixin`, `UpdateMixin`, `DeleteMixin`, `RESTManager`

### Object Creation

Optional attributes for object create:

- `deny_delete_tag`
- `member_check`
- `prevent_secrets`
- `commit_message_regex`
- `commit_message_negative_regex`
- `branch_name_regex`
- `author_email_regex`
- `file_name_regex`
- `max_file_size`
- `commit_committer_check`
- `reject_unsigned_commits`

### Object update

Optional attributes for object update:

- `deny_delete_tag`
- `member_check`
- `prevent_secrets`

- `commit_message_regex`
- `commit_message_negative_regex`
- `branch_name_regex`
- `author_email_regex`
- `file_name_regex`
- `max_file_size`
- `commit_committer_check`
- `reject_unsigned_commits`

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(\*\*kwargs: *Any*) → *GroupPushRules*

Retrieve a single object.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo`)

**Returns**

The generated [RESTObject](#)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

*GroupPushRules*

**class** gitlab.v4.objects.**GroupRegistryRepositoryManager**(gl: [Gitlab](#), parent: [RESTObject](#) | *None* = *None*)

Bases: [ListMixin](#), [RESTManager](#)

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**class** gitlab.v4.objects.**GroupRunner**(manager: [RESTManager](#), attrs: *Dict*[*str*, *Any*], \*, created\_from\_list: *bool* = *False*, lazy: *bool* = *False*)

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**GroupRunnerManager**(gl: Gitlab, parent: RESTObject | None = None)

Bases: *ListMixin*, *RESTManager*

#### Object listing filters

- scope
- tag\_list

#### Object Creation

Required attributes for object create:

- runner\_id

#### Object update

##### Parameters

- gl (Gitlab)
- parent (RESTObject | None)

**class** gitlab.v4.objects.**GroupSAMLGroupLink**(manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False)

Bases: *ObjectDeleteMixin*, *RESTObject*

##### Parameters

- manager (RESTManager)
- attrs (Dict[str, Any])
- created\_from\_list (bool)
- lazy (bool)

**class** gitlab.v4.objects.**GroupSAMLGroupLinkManager**(gl: Gitlab, parent: RESTObject | None = None)

Bases: *NoUpdateMixin*, *RESTManager*

#### Object Creation

Required attributes for object create:

- saml\_group\_name
- access\_level

#### Object update

##### Parameters

- gl (Gitlab)
- parent (RESTObject | None)

**get**(id: str | int, lazy: bool = False, \*\*kwargs: Any) → *GroupSAMLGroupLink*

Retrieve a single object.

##### Parameters

- id (str | int) – ID of the object to retrieve
- lazy (bool) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- \*\*kwargs (Any) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[GroupSAMLGroupLink](#)

```
class gitlab.v4.objects.GroupSubgroup(manager: RESTManager, attrs: Dict[str, Any], *,
                                       created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.GroupSubgroupManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [RESTManager](#)

**Object listing filters**

- skip\_groups
- all\_available
- search
- order\_by
- sort
- statistics
- owned
- with\_custom\_attributes
- min\_access\_level

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
class gitlab.v4.objects.GroupVariable(manager: RESTManager, attrs: Dict[str, Any], *,
                                       created_from_list: bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)

- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**GroupVariableManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: [CRUDMixin](#), [RESTManager](#)

### Object Creation

Required attributes for object create:

- **key**
- **value**

Optional attributes for object create:

- **protected**
- **variable\_type**
- **masked**

### Object update

Required attributes for object update:

- **key**
- **value**

Optional attributes for object update:

- **protected**
- **variable\_type**
- **masked**

### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id: str | int, lazy: bool = False, \*\*kwargs: Any*) → [GroupVariable](#)

Retrieve a single object.

### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

### Returns

The generated [RESTObject](#).

### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

### Return type

[GroupVariable](#)

```
class gitlab.v4.objects.GroupWiki(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [UploadMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

```
class gitlab.v4.objects.GroupWikiManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [CRUDMixin](#), [RESTManager](#)

#### Object listing filters

- **with\_content**

#### Object Creation

Required attributes for object create:

- **title**
- **content**

Optional attributes for object create:

- **format**

#### Object update

Optional attributes for object update:

- **title**
- **content**
- **format**

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | [None](#))

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → GroupWiki
```

Retrieve a single object.

#### Parameters

- **id** ([str](#) | [int](#)) – ID of the object to retrieve
- **lazy** ([bool](#)) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct

- **GitlabGetError** – If the server cannot perform the request

**Return type**

GroupWiki

```
class gitlab.v4.objects.Hook(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

```
class gitlab.v4.objects.HookManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [NoUpdateMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- **url**

**Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | [None](#))

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → Hook
```

Retrieve a single object.

**Parameters**

- **id** ([str](#) | [int](#)) – ID of the object to retrieve
- **lazy** ([bool](#)) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated [RESTObject](#).

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**[Hook](#)

```
class gitlab.v4.objects.Issue(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))

- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**IssueManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *RetrieveMixin, RESTManager*

#### Object listing filters

- state
- labels
- milestone
- scope
- author\_id
- iteration\_id
- assignee\_id
- my\_reaction\_emoji
- iids
- order\_by
- sort
- search
- created\_after
- created\_before
- updated\_after
- updated\_before

#### Object Creation

#### Object update

##### Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

**get**(*id: str | int, lazy: bool = False, \*\*kwargs: Any*) → *Issue*

Retrieve a single object.

##### Parameters

- **id** (*str | int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated RESTObject.

##### Raises



- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**[Issue](#)

```
class gitlab.v4.objects.IssuesStatistics(manager: RESTManager, attrs: Dict[str, Any], *,
                                         created_from_list: bool = False, lazy: bool = False)
```

Bases: [RefreshMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[str, Any])
- **created\_from\_list** (bool)
- **lazy** (bool)

```
class gitlab.v4.objects.IssuesStatisticsManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [GetWithoutIdMixin](#), [RESTManager](#)

**Object listing filters**

- **iids**

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
get(**kwargs: Any) → IssuesStatistics
```

Retrieve a single object.

**Parameters**

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated [RESTObject](#)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**[IssuesStatistics](#)

```
class gitlab.v4.objects.Key(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool =
                             False, lazy: bool = False)
```

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[str, Any])
- **created\_from\_list** (bool)

- **lazy** (*bool*)

**class** gitlab.v4.objects.**KeyManager**(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | *None* = *None*)

Bases: [GetMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *int* | *str* | *None* = *None*, *lazy*: *bool* = *False*, *\*\*kwargs*: [Any](#)) → [Key](#)

Retrieve a single object.

##### Parameters

- **id** (*int* | *str* | *None*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. `sudo`)

##### Returns

The generated [RESTObject](#).

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

##### Return type

[Key](#)

**class** gitlab.v4.objects.**LDAPGroup**(*manager*: [RESTManager](#), *attrs*: *Dict*[*str*, [Any](#)], \*, *created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: [RESTObject](#)

##### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, [Any](#)])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**LDAPGroupManager**(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | *None* = *None*)

Bases: [RESTManager](#)

#### Object listing filters

- `search`
- `provider`

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**list**(\*\*kwargs: [Any](#)) → List[[LDAPGroup](#)] | [RESTObjectList](#)

Retrieve a list of objects.

#### Parameters

- **all** – If True, return all the items, without pagination
- **per\_page** – Number of items to retrieve per request
- **page** – ID of the page to return (starts with page 1)
- **iterator** – If set to True and no pagination option is defined, return a generator instead of a list
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

#### Returns

The list of objects, or a generator if *iterator* is True

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the server cannot perform the request

#### Return type

List[[LDAPGroup](#)] | [RESTObjectList](#)

```
class gitlab.v4.objects.License(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool
                                = False, lazy: bool = False)
```

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (Dict[str, [Any](#)])
- **created\_from\_list** (bool)
- **lazy** (bool)

```
class gitlab.v4.objects.LicenseManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

#### Object listing filters

- popular

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(id: str | int, lazy: bool = False, \*\*kwargs: [Any](#)) → [License](#)

Retrieve a single object.

#### Parameters

- **id** (*str* / *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[License](#)

```
class gitlab.v4.objects.MergeRequest(manager: RESTManager, attrs: Dict[str, Any], *,
                                     created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.MergeRequestManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [RESTManager](#)

**Object listing filters**

- state
- order\_by
- sort
- milestone
- view
- labels
- with\_labels\_details
- with\_merge\_status\_recheck
- created\_after
- created\_before
- updated\_after
- updated\_before
- scope
- author\_id
- author\_username
- assignee\_id

- `approver_ids`
- `approved_by_ids`
- `reviewer_id`
- `reviewer_username`
- `my_reaction_emoji`
- `source_branch`
- `target_branch`
- `search`
- `in`
- `wip`
- `not`
- `environment`
- `deployed_before`
- `deployed_after`

**Object Creation****Object update****Parameters**

- `gl` (`Gitlab`)
- `parent` (`RESTObject` | `None`)

```
class gitlab.v4.objects.Namespace(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: `RESTObject`

**Parameters**

- `manager` (`RESTManager`)
- `attrs` (`Dict[str, Any]`)
- `created_from_list` (`bool`)
- `lazy` (`bool`)

```
class gitlab.v4.objects.NamespaceManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: `RetrieveMixin`, `RESTManager`

**Object listing filters**

- `search`

**Object Creation****Object update****Parameters**

- `gl` (`Gitlab`)
- `parent` (`RESTObject` | `None`)

**exists**(*namespace: str*, *\*\*kwargs: Any*) → *Namespace*

Get existence of a namespace by path.

**Parameters**

- **namespace** (*str*) – The path to the namespace.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server failed to perform the request

**Returns**

Data on namespace existence returned from the server.

**Return type**

*Namespace*

**get**(*id: str | int*, *lazy: bool = False*, *\*\*kwargs: Any*) → *Namespace*

Retrieve a single object.

**Parameters**

- **id** (*str | int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

*Namespace*

**class** gitlab.v4.objects.**NotificationSettings**(*manager: RESTManager*, *attrs: Dict[str, Any]*, \*,  
*created\_from\_list: bool = False*, *lazy: bool = False*)

Bases: *SaveMixin*, *RESTObject*

**Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**NotificationSettingsManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*, *UpdateMixin*, *RESTManager*

**Object Creation**

**Object update**

Optional attributes for object update:

- `level`
- `notification_email`
- `new_note`
- `new_issue`
- `reopen_issue`
- `close_issue`
- `reassign_issue`
- `new_merge_request`
- `reopen_merge_request`
- `close_merge_request`
- `reassign_merge_request`
- `merge_merge_request`

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) / *None*)

**get**(*\*\*kwargs*: *Any*) → *NotificationSettings*

Retrieve a single object.

#### Parameters

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated [RESTObject](#)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[NotificationSettings](#)

**class** `gitlab.v4.objects.PagesDomain`(*manager*: [RESTManager](#), *attrs*: *Dict*[*str*, *Any*], \*, *created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**PagesDomainManager**(gl: Gitlab, parent: RESTObject | None = None)

Bases: [ListMixin](#), [RESTManager](#)

Object Creation

Object update

Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

**class** gitlab.v4.objects.**PersonalAccessToken**(manager: [RESTManager](#), attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False)

Bases: [ObjectDeleteMixin](#), [ObjectRotateMixin](#), [RESTObject](#)

Parameters

- **manager** ([RESTManager](#))
- **attrs** (Dict[str, Any])
- **created\_from\_list** (bool)
- **lazy** (bool)

**class** gitlab.v4.objects.**PersonalAccessTokenManager**(gl: Gitlab, parent: RESTObject | None = None)

Bases: [DeleteMixin](#), [RetrieveMixin](#), [RotateMixin](#), [RESTManager](#)

Object listing filters

- **user\_id**

Object Creation

Object update

Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

**get**(id: str | int, lazy: bool = False, \*\*kwargs: Any) → [PersonalAccessToken](#)

Retrieve a single object.

Parameters

- **id** (str | int) – ID of the object to retrieve
- **lazy** (bool) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns

The generated RESTObject.

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

Return type

[PersonalAccessToken](#)



```
class gitlab.v4.objects.Project(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool
                                = False, lazy: bool = False)
```

Bases: [RefreshMixin](#), [SaveMixin](#), [ObjectDeleteMixin](#), [RepositoryMixin](#), [UploadMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

**access\_tokens:** [ProjectAccessTokenManager](#)

**accessrequests:** [ProjectAccessRequestManager](#)

**additionalstatistics:** [ProjectAdditionalStatisticsManager](#)

**approvalrules:** [ProjectApprovalRuleManager](#)

**approvals:** [ProjectApprovalManager](#)

**archive**(\*\*kwargs: [Any](#)) → None

Archive a project.

#### Parameters

- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

#### Raises

- [GitlabAuthenticationError](#) – If authentication is not correct
- [GitlabCreateError](#) – If the server failed to perform the request

#### Return type

None

**artifacts:** [ProjectArtifactManager](#)

**audit\_events:** [ProjectAuditEventManager](#)

**badges:** [ProjectBadgeManager](#)

**boards:** [ProjectBoardManager](#)

**branches:** [ProjectBranchManager](#)

**ci\_lint:** [ProjectCiLintManager](#)

**clusters:** [ProjectClusterManager](#)

**commits:** [ProjectCommitManager](#)

**create\_fork\_relation**(forked\_from\_id: int, \*\*kwargs: [Any](#)) → None

Create a forked from/to relation between existing projects.

#### Parameters

- **forked\_from\_id** (int) – The ID of the project that was forked from
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the relation could not be created

**Return type**

None

**customattributes:** *ProjectCustomAttributeManager***delete\_fork\_relation**(\*\*kwargs: Any) → None

Delete a forked relation between existing projects.

**Parameters****\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the server failed to perform the request

**Return type**

None

**deployments:** *ProjectDeploymentManager***deploytokens:** *ProjectDeployTokenManager***environments:** *ProjectEnvironmentManager***events:** *ProjectEventManager***exports:** *ProjectExportManager***files:** *ProjectFileManager***forks:** *ProjectForkManager***generic\_packages:** *GenericPackageManager***groups:** *ProjectGroupManager***hooks:** *ProjectHookManager***housekeeping**(\*\*kwargs: Any) → None

Start the housekeeping task.

**Parameters****\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabHousekeepingError** – If the server failed to perform the request

**Return type**

None

**imports:** *ProjectImportManager***integrations:** *ProjectIntegrationManager*

**invitations:** [\*ProjectInvitationManager\*](#)

**issues:** [\*ProjectIssueManager\*](#)

**issues\_statistics:** [\*ProjectIssuesStatisticsManager\*](#)

**iterations:** [\*ProjectIterationManager\*](#)

**job\_token\_scope:** [\*ProjectJobTokenScopeManager\*](#)

**jobs:** [\*ProjectJobManager\*](#)

**keys:** [\*ProjectKeyManager\*](#)

**labels:** [\*ProjectLabelManager\*](#)

**languages**(\*\*kwargs: Any) → Dict[str, Any] | Response

Get languages used in the project with percentage value.

#### Parameters

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server failed to perform the request

#### Return type

Dict[str, Any] | Response

**members:** [\*ProjectMemberManager\*](#)

**members\_all:** [\*ProjectMemberAllManager\*](#)

**merge\_trains:** [\*ProjectMergeTrainManager\*](#)

**mergerequests:** [\*ProjectMergeRequestManager\*](#)

**milestones:** [\*ProjectMilestoneManager\*](#)

**mirror\_pull**(\*\*kwargs: Any) → None

Start the pull mirroring process for the project.

#### Parameters

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server failed to perform the request

#### Return type

None

**mirror\_pull\_details**(\*\*kwargs: Any) → Dict[str, Any]

Get a project's pull mirror details.

Introduced in GitLab 15.5.

#### Parameters

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server failed to perform the request

**Returns**

dict of the parsed json returned by the server

**Return type**

*Dict[str, Any]*

**notes:** *ProjectNoteManager*

**notificationsettings:** *ProjectNotificationSettingsManager*

**packages:** *ProjectPackageManager*

**pagesdomains:** *ProjectPagesDomainManager*

**pipelines:** *ProjectPipelineManager*

**pipelineschedules:** *ProjectPipelineScheduleManager*

**protected\_environments:** *ProjectProtectedEnvironmentManager*

**protectedbranches:** *ProjectProtectedBranchManager*

**protectedtags:** *ProjectProtectedTagManager*

**pushrules:** *ProjectPushRulesManager*

**releases:** *ProjectReleaseManager*

**remote\_mirrors:** *ProjectRemoteMirrorManager*

**repositories:** *ProjectRegistryRepositoryManager*

**resource\_groups:** *ProjectResourceGroupManager*

**restore**(*\*\*kwargs: Any*) → None

Restore a project marked for deletion.

**Parameters**

***\*\*kwargs*** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabRestoreError** – If the server failed to perform the request

**Return type**

None

**runners:** *ProjectRunnerManager*

**search**(*scope: str, search: str, \*\*kwargs: Any*) → *GitlabList* | List[Dict[str, Any]]

Search the project resources matching the provided string.'

**Parameters**

- **scope** (*str*) – Scope of the search

- **search** (*str*) – Search string
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabSearchError** – If the server failed to perform the request

**Returns**

A list of dicts describing the resources found.

**Return type**

`GitlabList` | `List[Dict[str, Any]]`

**secure\_files:** `ProjectSecureFileManager`

**services:** `ProjectServiceManager`

**share**(*group\_id: int, group\_access: int, expires\_at: str | None = None, \*\*kwargs: Any*) → `None`

Share the project with a group.

**Parameters**

- **group\_id** (*int*) – ID of the group.
- **group\_access** (*int*) – Access level for the group.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)
- **expires\_at** (*str | None*)
- **\*\*kwargs**

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server failed to perform the request

**Return type**

`None`

**snapshot**(*wiki: bool = False, streamed: bool = False, action: Callable[[bytes], None] | None = None, chunk\_size: int = 1024, \*, iterator: bool = False, \*\*kwargs: Any*) → `bytes` | `Iterator[Any]` | `None`

Return a snapshot of the repository.

**Parameters**

- **wiki** (*bool*) – If True return the wiki repository
- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk\_size* and each chunk is passed to *action* for treatment.
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (`Callable[[bytes], None] | None`) – Callable responsible of dealing with chunk of data
- **chunk\_size** (*int*) – Size of each chunk
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct

- **GitlabGetError** – If the content could not be retrieved

**Returns**

The uncompressed tar archive of the repository

**Return type**

bytes | *Iterator*[*Any*] | None

**snippets:** *ProjectSnippetManager*

**star**(\*\*kwargs: *Any*) → None

Star a project.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server failed to perform the request

**Return type**

None

**storage:** *ProjectStorageManager*

**tags:** *ProjectTagManager*

**transfer**(to\_namespace: int | str, \*\*kwargs: *Any*) → None

Transfer a project to the given namespace ID

**Parameters**

- **to\_namespace** (int | str) – ID or path of the namespace to transfer the
- **to** (project)
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabTransferProjectError** – If the project could not be transferred

**Return type**

None

**trigger\_pipeline**(ref: str, token: str, variables: Dict[str, *Any*] | None = None, \*\*kwargs: *Any*) → *ProjectPipeline*

Trigger a CI build.

See <https://gitlab.com/help/ci/triggers/README.md#trigger-a-build>

**Parameters**

- **ref** (str) – Commit to build; can be a branch name or a tag
- **token** (str) – The trigger token
- **variables** (Dict[str, *Any*] | None) – Variables passed to the build script
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server failed to perform the request

**Return type***ProjectPipeline***triggers:** *ProjectTriggerManager***unarchive**(\*\*kwargs: *Any*) → None

Unarchive a project.

**Parameters****\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the server failed to perform the request

**Return type**

None

**unshare**(group\_id: int, \*\*kwargs: *Any*) → None

Delete a shared project link within a group.

**Parameters**

- **group\_id** (*int*) – ID of the group.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the server failed to perform the request

**Return type**

None

**unstar**(\*\*kwargs: *Any*) → None

Unstar a project.

**Parameters****\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the server failed to perform the request

**Return type**

None

**users:** *ProjectUserManager***variables:** *ProjectVariableManager***wikis:** *ProjectWikiManager*

```
class gitlab.v4.objects.ProjectAccessRequest(manager: RESTManager, attrs: Dict[str, Any], *,
                                             created_from_list: bool = False, lazy: bool = False)
```

Bases: [AccessRequestMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

```
class gitlab.v4.objects.ProjectAccessRequestManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [CreateMixin](#), [DeleteMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | [None](#))

```
class gitlab.v4.objects.ProjectAccessToken(manager: RESTManager, attrs: Dict[str, Any], *,
                                           created_from_list: bool = False, lazy: bool = False)
```

Bases: [ObjectDeleteMixin](#), [ObjectRotateMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

```
class gitlab.v4.objects.ProjectAccessTokenManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [CreateMixin](#), [DeleteMixin](#), [RetrieveMixin](#), [RotateMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **name**
- **scopes**

Optional attributes for object create:

- **access\_level**
- **expires\_at**

#### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | [None](#))



**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectAccessToken*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*ProjectAccessToken*

```
class gitlab.v4.objects.ProjectAdditionalStatistics(manager: RESTManager, attrs: Dict[str, Any],
*, created_from_list: bool = False, lazy: bool = False)
```

Bases: *RefreshMixin*, *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectAdditionalStatisticsManager(gl: Gitlab, parent: RESTObject | None
= None)
```

Bases: *GetWithoutIdMixin*, *RESTManager*

#### Object Creation

#### Object update

#### Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)

**get**(*\*\*kwargs*: *Any*) → *ProjectAdditionalStatistics*

Retrieve a single object.

#### Parameters

- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct

- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectAdditionalStatistics](#)

```
class gitlab.v4.objects.ProjectApproval(manager: RESTManager, attrs: Dict[str, Any], *,
                                         created_from_list: bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[str, Any])
- **created\_from\_list** (bool)
- **lazy** (bool)

```
class gitlab.v4.objects.ProjectApprovalManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [GetWithoutIdMixin](#), [UpdateMixin](#), [RESTManager](#)

**Object Creation****Object update**

Optional attributes for object update:

- **approvals\_before\_merge**
- **reset\_approvals\_on\_push**
- **disable\_overriding\_approvers\_per\_merge\_request**
- **merge\_requests\_author\_approval**
- **merge\_requests\_disable\_committers\_approval**

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
get(**kwargs: Any) → ProjectApproval
```

Retrieve a single object.

**Parameters**

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated [RESTObject](#)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectApproval](#)

```
class gitlab.v4.objects.ProjectApprovalRule(manager: RESTManager, attrs: Dict[str, Any], *,
                                             created_from_list: bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (bool)
- **lazy** (bool)

```
class gitlab.v4.objects.ProjectApprovalRuleManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [CreateMixin](#), [UpdateMixin](#), [DeleteMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- name
- approvals\_required

Optional attributes for object create:

- user\_ids
- group\_ids
- protected\_branch\_ids
- usernames

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
class gitlab.v4.objects.ProjectArtifact(manager: RESTManager, attrs: Dict[str, Any], *,
                                          created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

Dummy object to manage custom actions on artifacts

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (bool)
- **lazy** (bool)

```
class gitlab.v4.objects.ProjectArtifactManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [RESTManager](#)

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**delete**(\*\*kwargs: [Any](#)) → *None*

Delete the project's artifacts on the server.

**Parameters**

**\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the server cannot perform the request

**Return type**

*None*

**download**(ref\_name: *str*, job: *str*, streamed: *bool* = *False*, action: *Callable*[[*bytes*], *None*] | *None* = *None*, chunk\_size: *int* = 1024, \*, iterator: *bool* = *False*, \*\*kwargs: [Any](#)) → *bytes* | *Iterator*[[Any](#)] | *None*

Get the job artifacts archive from a specific tag or branch.

**Parameters**

- **ref\_name** (*str*) – Branch or tag name in repository. HEAD or SHA references are not supported.
- **job** (*str*) – The name of the job.
- **job\_token** – Job token for multi-project pipeline triggers.
- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk\_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable*[[*bytes*], *None*] | *None*) – Callable responsible of dealing with chunk of data
- **chunk\_size** (*int*) – Size of each chunk
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the artifacts could not be retrieved

**Returns**

The artifacts if *streamed* is *False*, *None* otherwise.

**Return type**

*bytes* | *Iterator*[[Any](#)] | *None*

**raw**(ref\_name: *str*, artifact\_path: *str*, job: *str*, streamed: *bool* = *False*, action: *Callable*[[*bytes*], *None*] | *None* = *None*, chunk\_size: *int* = 1024, \*, iterator: *bool* = *False*, \*\*kwargs: [Any](#)) → *bytes* | *Iterator*[[Any](#)] | *None*

Download a single artifact file from a specific tag or branch from within the job's artifacts archive.

**Parameters**

- **ref\_name** (*str*) – Branch or tag name in repository. HEAD or SHA references are not supported.
- **artifact\_path** (*str*) – Path to a file inside the artifacts archive.

- **job** (*str*) – The name of the job.
- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk\_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable[[bytes], None] | None*) – Callable responsible of dealing with chunk of data
- **chunk\_size** (*int*) – Size of each chunk
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the artifacts could not be retrieved

**Returns**

The artifact if *streamed* is False, None otherwise.

**Return type**

bytes | *Iterator*[*Any*] | None

```
class gitlab.v4.objects.ProjectAudit(manager: RESTManager, attrs: Dict[str, Any], *,
                                     created_from_list: bool = False, lazy: bool = False)
```

Bases: [ProjectAuditEvent](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectAuditEvent(manager: RESTManager, attrs: Dict[str, Any], *,
                                           created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectAuditEventManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

**Object listing filters**

- **created\_after**
- **created\_before**

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id: str | int, lazy: bool = False, \*\*kwargs: Any*) → *ProjectAuditEvent*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[ProjectAuditEvent](#)

**class** gitlab.v4.objects.**ProjectAuditManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: [ProjectAuditEventManager](#)

#### Object listing filters

- **created\_after**
- **created\_before**

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**class** gitlab.v4.objects.**ProjectBadge**(*manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False*)

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

##### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**ProjectBadgeManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: [BadgeRenderMixin](#), [CRUDMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- `link_url`
- `image_url`

### Object update

Optional attributes for object update:

- `link_url`
- `image_url`

### Parameters

- **gl** (`Gitlab`)
- **parent** (`RESTObject` / `None`)

**get**(*id*: `str` | `int`, *lazy*: `bool` = `False`, *\*\*kwargs*: `Any`) → `ProjectBadge`

Retrieve a single object.

### Parameters

- **id** (`str` / `int`) – ID of the object to retrieve
- **lazy** (`bool`) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (`Any`) – Extra options to send to the server (e.g. `sudo`)

### Returns

The generated `RESTObject`.

### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

### Return type

`ProjectBadge`

```
class gitlab.v4.objects.ProjectBoard(manager: RESTManager, attrs: Dict[str, Any], *,
                                     created_from_list: bool = False, lazy: bool = False)
```

Bases: `SaveMixin`, `ObjectDeleteMixin`, `RESTObject`

### Parameters

- **manager** (`RESTManager`)
- **attrs** (`Dict[str, Any]`)
- **created\_from\_list** (`bool`)
- **lazy** (`bool`)

**lists:** `ProjectBoardListManager`

```
class gitlab.v4.objects.ProjectBoardList(manager: RESTManager, attrs: Dict[str, Any], *,
                                          created_from_list: bool = False, lazy: bool = False)
```

Bases: `SaveMixin`, `ObjectDeleteMixin`, `RESTObject`

### Parameters

- **manager** (`RESTManager`)

- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**ProjectBoardListManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: [CRUDMixin](#), [RESTManager](#)

#### Object Creation

Mutually exclusive attributes for object create:

- **label\_id**
- **assignee\_id**
- **milestone\_id**

#### Object update

Required attributes for object update:

- **position**

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id: str | int, lazy: bool = False, \*\*kwargs: Any*) → [ProjectBoardList](#)

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[ProjectBoardList](#)

**class** gitlab.v4.objects.**ProjectBoardManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: [CRUDMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **name**

#### Object update

#### Parameters



- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectBoard*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated *RESTObject*.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*ProjectBoard*

```
class gitlab.v4.objects.ProjectBranch(manager: RESTManager, attrs: Dict[str, Any], *,
                                     created_from_list: bool = False, lazy: bool = False)
```

Bases: *ObjectDeleteMixin*, *RESTObject*

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectBranchManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *NoUpdateMixin*, *RESTManager*

#### Object Creation

Required attributes for object create:

- **branch**
- **ref**

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectBranch*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve

- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectBranch](#)

```
class gitlab.v4.objects.ProjectCiLint(manager: RESTManager, attrs: Dict[str, Any], *,
                                       created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectCiLintManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [GetWithoutIdMixin](#), [CreateMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- content

Optional attributes for object create:

- dry\_run
- include\_jobs
- ref

**Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
get(**kwargs: Any) → ProjectCiLint
```

Retrieve a single object.

**Parameters**

- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct

- **GitlabGetError** – If the server cannot perform the request

### Return type

ProjectCiLint

```
validate(*args: Any, **kwargs: Any) → None
```

Raise an error if the Project CI Lint results are not valid.

This is a custom python-gitlab method to wrap lint endpoints.

## Parameters

- **args** ([Any](#))
- **kwargs** ([Any](#))

### Return type

None

```
class gitlab.v4.objects.ProjectCluster(manager: RESTManager, attrs: Dict[str, Any], *,
                                         created_from_list: bool = False, lazy: bool = False)
```

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

## Parameters

- **manager** (`RESTManager`)
- **attrs** (`Dict[str, Any]`)
- **created\_from\_list** (`bool`)
- **lazy** (`bool`)

```
class gitlab.v4.objects.ProjectClusterManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *CRUDMixin*, *RESTManager*

## Object Creation

Required attributes for object create:

- name
- platform\_kubernetes\_attributes

Optional attributes for object create:

- domain
- enabled
- managed
- environment\_scope

## Object update

Optional attributes for object update:

- name
- domain
- management\_project\_id
- platform\_kubernetes\_attributes
- environment\_scope

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**create**(*data*: *Dict[str, Any]* | *None* = *None*, *\*\*kwargs*: *Any*) → *ProjectCluster*

Create a new object.

**Parameters**

- **data** (*Dict[str, Any]* | *None*) – Parameters to send to the server to create the resource
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo` or `'ref_name'`, `'stage'`, `'name'`, `'all'`)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

**Returns**

A new instance of the `manage` object class build with the data sent by the server

**Return type**

[ProjectCluster](#)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectCluster*

Retrieve a single object.

**Parameters**

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If *True*, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo`)

**Returns**

The generated `RESTObject`.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectCluster](#)

```
class gitlab.v4.objects.ProjectCommit(manager: RESTManager, attrs: Dict[str, Any], *,  
                                       created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)

- **lazy** (*bool*)

**cherry\_pick**(*branch: str, \*\*kwargs: Any*) → *None*

Cherry-pick a commit into a branch.

#### Parameters

- **branch** (*str*) – Name of target branch
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCherryPickError** – If the cherry-pick could not be performed

#### Return type

*None*

**comments:** *ProjectCommitCommentManager*

**diff**(*\*\*kwargs: Any*) → *GitlabList* | *List[Dict[str, Any]]*

Generate the commit diff.

#### Parameters

- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the diff could not be retrieved

#### Returns

The changes done in this commit

#### Return type

*GitlabList* | *List[Dict[str, Any]]*

**discussions:** *ProjectCommitDiscussionManager*

**merge\_requests**(*\*\*kwargs: Any*) → *GitlabList* | *List[Dict[str, Any]]*

List the merge requests related to the commit.

#### Parameters

- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the references could not be retrieved

#### Returns

The merge requests related to the commit.

#### Return type

*GitlabList* | *List[Dict[str, Any]]*

**refs**(*type: str = 'all', \*\*kwargs: Any*) → *GitlabList* | *List[Dict[str, Any]]*

List the references the commit is pushed to.

#### Parameters

- **type** (*str*) – The scope of references ('branch', 'tag' or 'all')

- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the references could not be retrieved

**Returns**

The references the commit is pushed to.

**Return type**

[GitlabList](#) | *List[Dict[str, Any]]*

**revert**(*branch: str, \*\*kwargs: Any*) → Dict[str, [Any](#)] | Response

Revert a commit on a given branch.

**Parameters**

- **branch** (*str*) – Name of target branch
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabRevertError** – If the revert could not be performed

**Returns**

The new commit data (*not* a RESTObject)

**Return type**

*Dict[str, Any] | Response*

**signature**(*\*\*kwargs: Any*) → Dict[str, [Any](#)] | Response

Get the signature of the commit.

**Parameters**

- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the signature could not be retrieved

**Returns**

The commit's signature data

**Return type**

*Dict[str, Any] | Response*

**statuses:** [ProjectCommitStatusManager](#)

**class** gitlab.v4.objects.**ProjectCommitComment**(*manager: [RESTManager](#), attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False*)

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)

- **lazy** (*bool*)

**class** gitlab.v4.objects.**ProjectCommitCommentManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: [ListMixin](#), [CreateMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **note**

Optional attributes for object create:

- **path**
- **line**
- **line\_type**

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**class** gitlab.v4.objects.**ProjectCommitDiscussion**(*manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False*)

Bases: [RESTObject](#)

##### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**notes:** [ProjectCommitDiscussionNoteManager](#)

**class** gitlab.v4.objects.**ProjectCommitDiscussionManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: [RetrieveMixin](#), [CreateMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **body**

Optional attributes for object create:

- **created\_at**

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectCommitDiscussion*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*ProjectCommitDiscussion*

```
class gitlab.v4.objects.ProjectCommitDiscussionNote(manager: RESTManager, attrs: Dict[str, Any],
*, created_from_list: bool = False, lazy: bool = False)
```

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectCommitDiscussionNoteManager(gl: Gitlab, parent: RESTObject | None
= None)
```

Bases: *GetMixin*, *CreateMixin*, *UpdateMixin*, *DeleteMixin*, *RESTManager*

#### Object Creation

Required attributes for object create:

- **body**

Optional attributes for object create:

- **created\_at**
- **position**

#### Object update

Required attributes for object update:

- **body**

#### Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)



**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectCommitDiscussionNote*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*ProjectCommitDiscussionNote*

**class** gitlab.v4.objects.**ProjectCommitManager**(*gl*: *Gitlab*, *parent*: *RESTObject* | *None* = *None*)

Bases: *RetrieveMixin*, *CreateMixin*, *RESTManager*

#### Object listing filters

- all
- ref\_name
- since
- until
- path
- with\_stats
- first\_parent
- order
- trailers

#### Object Creation

Required attributes for object create:

- branch
- commit\_message
- actions

Optional attributes for object create:

- author\_email
- author\_name

#### Object update

##### Parameters

- **gl** (*Gitlab*)

- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectCommit*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. *sudo*)

#### Returns

The generated *RESTObject*.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*ProjectCommit*

```
class gitlab.v4.objects.ProjectCommitStatus(manager: RESTManager, attrs: Dict[str, Any], *,  
                                             created_from_list: bool = False, lazy: bool = False)
```

Bases: [RefreshMixin](#), *RESTObject*

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectCommitStatusManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [CreateMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **state**

Optional attributes for object create:

- **description**
- **name**
- **context**
- **ref**
- **target\_url**
- **coverage**

#### Object update

##### Parameters

- **gl** ([Gitlab](#))

- **parent** (`RESTObject` | `None`)

**create**(*data*: `Dict[str, Any]` | `None = None`, *\*\*kwargs*: `Any`) → `ProjectCommitStatus`

Create a new object.

#### Parameters

- **data** (`Dict[str, Any]` | `None`) – Parameters to send to the server to create the resource
- **\*\*kwargs** (`Any`) – Extra options to send to the server (e.g. `sudo` or `'ref_name'`, `'stage'`, `'name'`, `'all'`)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

#### Returns

A new instance of the manage object class build with the data sent by the server

#### Return type

`ProjectCommitStatus`

```
class gitlab.v4.objects.ProjectCustomAttribute(manager: RESTManager, attrs: Dict[str, Any], *,
                                              created_from_list: bool = False, lazy: bool = False)
```

Bases: `ObjectDeleteMixin`, `RESTObject`

#### Parameters

- **manager** (`RESTManager`)
- **attrs** (`Dict[str, Any]`)
- **created\_from\_list** (`bool`)
- **lazy** (`bool`)

```
class gitlab.v4.objects.ProjectCustomAttributeManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: `RetrieveMixin`, `SetMixin`, `DeleteMixin`, `RESTManager`

#### Object Creation

#### Object update

#### Parameters

- **gl** (`Gitlab`)
- **parent** (`RESTObject` | `None`)

**get**(*id*: `str` | `int`, *lazy*: `bool = False`, *\*\*kwargs*: `Any`) → `ProjectCustomAttribute`

Retrieve a single object.

#### Parameters

- **id** (`str` | `int`) – ID of the object to retrieve
- **lazy** (`bool`) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (`Any`) – Extra options to send to the server (e.g. `sudo`)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectCustomAttribute](#)

```
class gitlab.v4.objects.ProjectDeployToken(manager: RESTManager, attrs: Dict[str, Any], *,
                                             created_from_list: bool = False, lazy: bool = False)
```

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectDeployTokenManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [RetrieveMixin](#), [CreateMixin](#), [DeleteMixin](#), [RESTManager](#)

**Object listing filters**

- **scopes**

**Object Creation**

Required attributes for object create:

- **name**
- **scopes**

Optional attributes for object create:

- **expires\_at**
- **username**

**Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectDeployToken
```

Retrieve a single object.

**Parameters**

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

`ProjectDeployToken`

```
class gitlab.v4.objects.ProjectDeployment(manager: RESTManager, attrs: Dict[str, Any], *,
                                          created_from_list: bool = False, lazy: bool = False)
```

Bases: `SaveMixin`, `RESTObject`

**Parameters**

- **manager** (`RESTManager`)
- **attrs** (`Dict[str, Any]`)
- **created\_from\_list** (`bool`)
- **lazy** (`bool`)

```
approval(status: str, comment: str | None = None, represented_as: str | None = None, **kwargs: Any) →
Dict[str, Any]
```

Approve or reject a blocked deployment.

**Parameters**

- **status** (`str`) – Either “approved” or “rejected”
- **comment** (`str | None`) – A comment to go with the approval
- **represented\_as** (`str | None`) – The name of the User/Group/Role to use for the approval, when the user belongs to multiple approval rules.
- **\*\*kwargs** (`Any`) – Extra options to send to the server (e.g. `sudo`)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabMRApprovalError** – If the approval failed

**Returns**

A dict containing the result.

**Return type**

`Dict[str, Any]`

<https://docs.gitlab.com/ee/api/deployments.html#approve-or-reject-a-blocked-deployment>

**mergerequests:** `ProjectDeploymentMergeRequestManager`

```
class gitlab.v4.objects.ProjectDeploymentManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: `RetrieveMixin`, `CreateMixin`, `UpdateMixin`, `RESTManager`

**Object listing filters**

- `order_by`
- `sort`
- `updated_after`

- `updated_before`
- `environment`
- `status`

### Object Creation

Required attributes for object create:

- `sha`
- `ref`
- `tag`
- `status`
- `environment`

### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | `None`)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectDeployment*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[ProjectDeployment](#)

```
class gitlab.v4.objects.ProjectDeploymentMergeRequest(manager: RESTManager, attrs: Dict[str,  
                                                    Any], *, created_from_list: bool = False,  
                                                    lazy: bool = False)
```

Bases: [MergeRequest](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**ProjectDeploymentMergeRequestManager**(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | *None* = *None*)

Bases: [MergeRequestManager](#)

#### Object listing filters

- state
- order\_by
- sort
- milestone
- view
- labels
- with\_labels\_details
- with\_merge\_status\_recheck
- created\_after
- created\_before
- updated\_after
- updated\_before
- scope
- author\_id
- author\_username
- assignee\_id
- approver\_ids
- approved\_by\_ids
- reviewer\_id
- reviewer\_username
- my\_reaction\_emoji
- source\_branch
- target\_branch
- search
- in
- wip
- not
- environment
- deployed\_before
- deployed\_after

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**class** gitlab.v4.objects.**ProjectEnvironment**(*manager*: [RESTManager](#), *attrs*: *Dict[str, Any]*, \*,  
*created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**stop**(\*\**kwargs*: *Any*) → *Dict[str, Any]* | *Response*

Stop the environment.

#### Parameters

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo`)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabStopError** – If the operation failed

#### Returns

A dict of the result.

#### Return type

*Dict[str, Any]* | *Response*

**class** gitlab.v4.objects.**ProjectEnvironmentManager**(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | *None* = *None*)

Bases: [RetrieveMixin](#), [CreateMixin](#), [UpdateMixin](#), [DeleteMixin](#), [RESTManager](#)

#### Object listing filters

- **name**
- **search**
- **states**

#### Object Creation

Required attributes for object create:

- **name**

Optional attributes for object create:

- **external\_url**

#### Object update

Optional attributes for object update:

- **name**
- **external\_url**

#### Parameters

- **gl** ([Gitlab](#))



- **parent** ([RESTObject](#) | *None*)

**get**(*id: str | int, lazy: bool = False, \*\*kwargs: Any*) → *ProjectEnvironment*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[ProjectEnvironment](#)

```
class gitlab.v4.objects.ProjectEvent(manager: RESTManager, attrs: Dict[str, Any], *,
                                     created_from_list: bool = False, lazy: bool = False)
```

Bases: [Event](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectEventManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [EventManager](#)

#### Object listing filters

- action
- target\_type
- before
- after
- sort
- scope

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
class gitlab.v4.objects.ProjectExport(manager: RESTManager, attrs: Dict[str, Any], *,
                                     created_from_list: bool = False, lazy: bool = False)
```

Bases: [DownloadMixin](#), [RefreshMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

```
class gitlab.v4.objects.ProjectExportManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [GetWithoutIdMixin](#), [CreateMixin](#), [RESTManager](#)

#### Object Creation

Optional attributes for object create:

- **description**

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | [None](#))

```
get(**kwargs: Any) → ProjectExport
```

Retrieve a single object.

##### Parameters

**\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated [RESTObject](#)

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

##### Return type

[ProjectExport](#)

```
class gitlab.v4.objects.ProjectFile(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
                                   bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([ProjectFileManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

**branch:** [str](#)

**commit\_message:** [str](#)

**decode()** → bytes

Returns the decoded content of the file.

**Returns**

The decoded content.

**Return type**

bytes

**delete**(*branch: str, commit\_message: str, \*\*kwargs: Any*) → None

Delete the file from the server.

**Parameters**

- **branch** (*str*) – Branch from which the file will be removed
- **commit\_message** (*str*) – Commit message for the deletion
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the server cannot perform the request

**Return type**

None

**file\_path:** *str*

**manager:** *ProjectFileManager*

**save**(*branch: str, commit\_message: str, \*\*kwargs: Any*) → None

Save the changes made to the file to the server.

The object is updated to match what the server returns.

**Parameters**

- **branch** (*str*) – Branch in which the file will be updated
- **commit\_message** (*str*) – Message to send with the commit
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUpdateError** – If the server cannot perform the request

**Return type**

None

**class** gitlab.v4.objects.**ProjectFileManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *GetMixin, CreateMixin, UpdateMixin, DeleteMixin, RESTManager*

**Object Creation**

Required attributes for object create:

- *file\_path*
- *branch*
- *content*

- `commit_message`

Optional attributes for object create:

- `encoding`
- `author_email`
- `author_name`

### Object update

Required attributes for object update:

- `file_path`
- `branch`
- `content`
- `commit_message`

Optional attributes for object update:

- `encoding`
- `author_email`
- `author_name`

### Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) | `None`)

**blame**(*file\_path*: *str*, *ref*: *str*, *\*\*kwargs*: [Any](#)) → `List[Dict[str, Any]]`

Return the content of a file for a commit.

### Parameters

- **file\_path** (*str*) – Path of the file to retrieve
- **ref** (*str*) – Name of the branch, tag or commit
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. `sudo`)

### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the server failed to perform the request

### Returns

A list of commits/lines matching the file

### Return type

`List[Dict[str, Any]]`

**create**(*data*: `Dict[str, Any] | None = None`, *\*\*kwargs*: [Any](#)) → [ProjectFile](#)

Create a new object.

### Parameters

- **data** (`Dict[str, Any] | None`) – parameters to send to the server to create the resource
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. `sudo`)

**Returns**

a new instance of the managed object class built with the data sent by the server

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

**Return type**

[ProjectFile](#)

**delete**(*file\_path*: str, *branch*: str, *commit\_message*: str, *\*\*kwargs*: Any) → None

Delete a file on the server.

**Parameters**

- **file\_path** (str) – Path of the file to remove
- **branch** (str) – Branch from which the file will be removed
- **commit\_message** (str) – Commit message for the deletion
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the server cannot perform the request

**Return type**

None

**get**(*file\_path*: str, *ref*: str, *\*\*kwargs*: Any) → [ProjectFile](#)

Retrieve a single file.

**Parameters**

- **file\_path** (str) – Path of the file to retrieve
- **ref** (str) – Name of the branch, tag or commit
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the file could not be retrieved

**Returns**

The generated RESTObject

**Return type**

[ProjectFile](#)

**raw**(*file\_path*: str, *ref*: str, *streamed*: bool = False, *action*: Callable[[...], Any] | None = None, *chunk\_size*: int = 1024, *\*, iterator*: bool = False, *\*\*kwargs*: Any) → bytes | Iterator[Any] | None

Return the content of a file for a commit.

**Parameters**

- **ref** (str) – ID of the commit
- **file\_path** (str) – Path of the file to return

- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk\_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable[[...], Any] | None*) – Callable responsible of dealing with chunk of data
- **chunk\_size** (*int*) – Size of each chunk
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the file could not be retrieved

**Returns**

The file content

**Return type**

bytes | *Iterator*[*Any*] | None

**update**(*file\_path: str, new\_data: Dict[str, Any] | None = None, \*\*kwargs: Any*) → Dict[str, Any]

Update an object on the server.

**Parameters**

- **id** – ID of the object to update (can be None if not required)
- **new\_data** (*Dict[str, Any] | None*) – the update data for the object
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)
- **file\_path** (*str*)

**Returns**

The new object data (*not* a RESTObject)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUpdateError** – If the server cannot perform the request

**Return type**

Dict[str, Any]

**class** gitlab.v4.objects.**ProjectFork**(*manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False*)

Bases: *RESTObject*

**Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**ProjectForkManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *CreateMixin, ListMixin, RESTManager*

Object listing filters

- archived
- visibility
- order\_by
- sort
- search
- simple
- owned
- membership
- starred
- statistics
- with\_custom\_attributes
- with\_issues\_enabled
- with\_merge\_requests\_enabled

### Object Creation

Optional attributes for object create:

- namespace

### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**create**(*data*: *Dict[str, Any]* | *None* = *None*, *\*\*kwargs*: *Any*) → *ProjectFork*

Creates a new object.

#### Parameters

- **data** (*Dict[str, Any]* | *None*) – Parameters to send to the server to create the resource
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

#### Returns

A new instance of the managed object class build with the data sent by the server

#### Return type

[ProjectFork](#)

```
class gitlab.v4.objects.ProjectHook(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**ProjectHookManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: [CRUDMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **url**

Optional attributes for object create:

- **push\_events**
- **issues\_events**
- **confidential\_issues\_events**
- **merge\_requests\_events**
- **tag\_push\_events**
- **note\_events**
- **job\_events**
- **pipeline\_events**
- **wiki\_page\_events**
- **enable\_ssl\_verification**
- **token**

#### Object update

Required attributes for object update:

- **url**

Optional attributes for object update:

- **push\_events**
- **issues\_events**
- **confidential\_issues\_events**
- **merge\_requests\_events**
- **tag\_push\_events**
- **note\_events**
- **job\_events**
- **pipeline\_events**
- **wiki\_events**
- **enable\_ssl\_verification**
- **token**



**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id: str | int, lazy: bool = False, \*\*kwargs: Any*) → [ProjectHook](#)

Retrieve a single object.

**Parameters**

- **id** (*str | int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated [RESTObject](#).

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectHook](#)

```
class gitlab.v4.objects.ProjectImport(manager: RESTManager, attrs: Dict[str, Any], *,
                                     created_from_list: bool = False, lazy: bool = False)
```

Bases: [RefreshMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectImportManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [GetWithoutIdMixin](#), [RESTManager](#)

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*\*\*kwargs: Any*) → [ProjectImport](#)

Retrieve a single object.

**Parameters**

- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated [RESTObject](#)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

*ProjectImport*

```
class gitlab.v4.objects.ProjectIntegration(manager: RESTManager, attrs: Dict[str, Any], *,
                                           created_from_list: bool = False, lazy: bool = False)
```

Bases: *SaveMixin, ObjectDeleteMixin, RESTObject*

**Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectIntegrationManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *GetMixin, UpdateMixin, DeleteMixin, ListMixin, RESTManager*

**Object Creation****Object update****Parameters**

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)

```
available() → List[str]
```

List the services known by python-gitlab.

**Returns**

The list of service code names.

**Return type**

*List[str]*

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectIntegration
```

Retrieve a single object.

**Parameters**

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**[ProjectIntegration](#)

```
class gitlab.v4.objects.ProjectInvitation(manager: RESTManager, attrs: Dict[str, Any], *,
                                          created_from_list: bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[str, Any])
- **created\_from\_list** (bool)
- **lazy** (bool)

```
class gitlab.v4.objects.ProjectInvitationManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [InvitationMixin](#), [RESTManager](#)

**Object listing filters**

- query

**Object Creation**

Required attributes for object create:

- access\_level

Optional attributes for object create:

- expires\_at
- invite\_source
- tasks\_to\_be\_done
- tasks\_project\_id

Mutually exclusive attributes for object create:

- email
- user\_id

**Object update**

Optional attributes for object update:

- access\_level
- expires\_at

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectInvitation
```

Retrieve a single object.

**Parameters**

- **id** (str | int) – ID of the object to retrieve

- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectInvitation](#)

```
class gitlab.v4.objects.ProjectIssue(manager: RESTManager, attrs: Dict[str, Any], *,
                                     created_from_list: bool = False, lazy: bool = False)
```

Bases: [UserAgentDetailMixin](#), [SubscribableMixin](#), [TodoMixin](#), [TimeTrackingMixin](#), [ParticipantsMixin](#), [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**awardemojis:** [ProjectIssueAwardEmojiManager](#)

**closed\_by** (**\*\*kwargs**: *Any*) → *Dict[str, Any]*

List merge requests that will close the issue when merged.

**Parameters**

- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the merge requests could not be retrieved

**Returns**

The list of merge requests.

**Return type**

*Dict[str, Any]*

**discussions:** [ProjectIssueDiscussionManager](#)

**links:** [ProjectIssueLinkManager](#)

**move** (*to\_project\_id*: *int*, **\*\*kwargs**: *Any*) → *None*

Move the issue to another project.

**Parameters**

- **to\_project\_id** (*int*) – ID of the target project
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUpdateError** – If the issue could not be moved

**Return type**

None

**notes:** *ProjectIssueNoteManager***related\_merge\_requests**(\*\*kwargs: Any) → Dict[str, Any]

List merge requests related to the issue.

**Parameters****\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the merge requests could not be retrieved

**Returns**

The list of merge requests.

**Return type**

Dict[str, Any]

**reorder**(move\_after\_id: int | None = None, move\_before\_id: int | None = None, \*\*kwargs: Any) → None

Reorder an issue on a board.

**Parameters**

- **move\_after\_id** (int | None) – ID of an issue that should be placed after this issue
- **move\_before\_id** (int | None) – ID of an issue that should be placed before this issue
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUpdateError** – If the issue could not be reordered

**Return type**

None

**resource\_iteration\_events:** *ProjectIssueResourceIterationEventManager***resource\_weight\_events:** *ProjectIssueResourceWeightEventManager***resourcelabelevents:** *ProjectIssueResourceLabelEventManager***resourcemilestoneevents:** *ProjectIssueResourceMilestoneEventManager***resourcestateevents:** *ProjectIssueResourceStateEventManager*

```
class gitlab.v4.objects.ProjectIssueAwardEmoji(manager: RESTManager, attrs: Dict[str, Any], *,
                                              created_from_list: bool = False, lazy: bool = False)
```

Bases: *ObjectDeleteMixin, RESTObject***Parameters**

- **manager** (RESTManager)

- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectIssueAwardEmojiManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *NoUpdateMixin*, *RESTManager*

### Object Creation

Required attributes for object create:

- **name**

### Object update

#### Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectIssueAwardEmoji
```

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*ProjectIssueAwardEmoji*

```
class gitlab.v4.objects.ProjectIssueDiscussion(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**notes:** *ProjectIssueDiscussionNoteManager*

```
class gitlab.v4.objects.ProjectIssueDiscussionManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [RetrieveMixin](#), [CreateMixin](#), [RESTManager](#)

### Object Creation

Required attributes for object create:

- **body**

Optional attributes for object create:

- **created\_at**

### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectIssueDiscussion
```

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[ProjectIssueDiscussion](#)

```
class gitlab.v4.objects.ProjectIssueDiscussionNote(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectIssueDiscussionNoteManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [GetMixin](#), [CreateMixin](#), [UpdateMixin](#), [DeleteMixin](#), [RESTManager](#)

### Object Creation

Required attributes for object create:

- `body`

Optional attributes for object create:

- `created_at`

### Object update

Required attributes for object update:

- `body`

#### Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) | `None`)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectIssueDiscussionNote*

Retrieve a single object.

#### Parameters

- `id` (*str* | *int*) – ID of the object to retrieve
- `lazy` (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- *\*\*kwargs* (*Any*) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[ProjectIssueDiscussionNote](#)

```
class gitlab.v4.objects.ProjectIssueLink(manager: RESTManager, attrs: Dict[str, Any], *,
                                          created_from_list: bool = False, lazy: bool = False)
```

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- `manager` ([RESTManager](#))
- `attrs` (*Dict*[*str*, *Any*])
- `created_from_list` (*bool*)
- `lazy` (*bool*)

```
class gitlab.v4.objects.ProjectIssueLinkManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [CreateMixin](#), [DeleteMixin](#), [RESTManager](#)

### Object Creation

Required attributes for object create:

- `target_project_id`



- `target_issue_iid`

### Object update

#### Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) | `None`)

**create**(*data*: *Dict*[*str*, *Any*], *\*\*kwargs*: *Any*) → *Tuple*[*RESTObject*, *RESTObject*]

Create a new object.

#### Parameters

- `data` (*Dict*[*str*, *Any*]) – parameters to send to the server to create the resource
- *\*\*kwargs* (*Any*) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The source and target issues

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

#### Return type

*Tuple*[*RESTObject*, *RESTObject*]

**class** `gitlab.v4.objects.ProjectIssueManager`(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | `None` = `None`)

Bases: [CRUDMixin](#), [RESTManager](#)

### Object listing filters

- `iids`
- `state`
- `labels`
- `milestone`
- `scope`
- `author_id`
- `iteration_id`
- `assignee_id`
- `my_reaction_emoji`
- `order_by`
- `sort`
- `search`
- `created_after`
- `created_before`
- `updated_after`
- `updated_before`

### Object Creation

Required attributes for object create:

- `title`

Optional attributes for object create:

- `description`
- `confidential`
- `assignee_ids`
- `assignee_id`
- `milestone_id`
- `labels`
- `created_at`
- `due_date`
- `merge_request_to_resolve_discussions_of`
- `discussion_to_resolve`

### Object update

Optional attributes for object update:

- `title`
- `description`
- `confidential`
- `assignee_ids`
- `assignee_id`
- `milestone_id`
- `labels`
- `state_event`
- `updated_at`
- `due_date`
- `discussion_locked`

#### Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) / `None`)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectIssue*

Retrieve a single object.

#### Parameters

- **id** (*str* / *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If `True`, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.

- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

*ProjectIssue*

```
class gitlab.v4.objects.ProjectIssueNote(manager: RESTManager, attrs: Dict[str, Any], *,
                                          created_from_list: bool = False, lazy: bool = False)
```

Bases: *SaveMixin, ObjectDeleteMixin, RESTObject*

**Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**awardemojis:** *ProjectIssueNoteAwardEmojiManager*

```
class gitlab.v4.objects.ProjectIssueNoteAwardEmoji(manager: RESTManager, attrs: Dict[str, Any], *,
                                                    created_from_list: bool = False, lazy: bool =
                                                    False)
```

Bases: *ObjectDeleteMixin, RESTObject*

**Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectIssueNoteAwardEmojiManager(gl: Gitlab, parent: RESTObject | None =
                                                         None)
```

Bases: *NoUpdateMixin, RESTManager*

**Object Creation**

Required attributes for object create:

- **name**

**Object update****Parameters**

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectIssueNoteAwardEmoji*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*ProjectIssueNoteAwardEmoji*

**class** gitlab.v4.objects.**ProjectIssueNoteManager**(*gl*: *Gitlab*, *parent*: *RESTObject* | *None* = *None*)

Bases: *CRUDMixin*, *RESTManager*

#### Object Creation

Required attributes for object create:

- **body**

Optional attributes for object create:

- **created\_at**

#### Object update

Required attributes for object update:

- **body**

#### Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectIssueNote*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct

- **GitlabGetError** – If the server cannot perform the request

**Return type**`ProjectIssueNote`

```
class gitlab.v4.objects.ProjectIssueResourceIterationEvent(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: `RESTObject`

**Parameters**

- **manager** (`RESTManager`)
- **attrs** (`Dict[str, Any]`)
- **created\_from\_list** (`bool`)
- **lazy** (`bool`)

```
class gitlab.v4.objects.ProjectIssueResourceIterationEventManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: `RetrieveMixin`, `RESTManager`

**Object Creation****Object update****Parameters**

- **gl** (`Gitlab`)
- **parent** (`RESTObject` | `None`)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectIssueResourceIterationEvent
```

Retrieve a single object.

**Parameters**

- **id** (`str` | `int`) – ID of the object to retrieve
- **lazy** (`bool`) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (`Any`) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated `RESTObject`.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**`ProjectIssueResourceIterationEvent`

```
class gitlab.v4.objects.ProjectIssueResourceLabelEvent(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: `RESTObject`

**Parameters**

- **manager** (`RESTManager`)

- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**ProjectIssueResourceLabelEventManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *RetrieveMixin, RESTManager*

**Object Creation**

**Object update**

**Parameters**

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

**get**(*id: str | int, lazy: bool = False, \*\*kwargs: Any*) → *ProjectIssueResourceLabelEvent*

Retrieve a single object.

**Parameters**

- **id** (*str | int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

*ProjectIssueResourceLabelEvent*

**class** gitlab.v4.objects.**ProjectIssueResourceMilestoneEvent**(*manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False*)

Bases: *RESTObject*

**Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**ProjectIssueResourceMilestoneEventManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *RetrieveMixin, RESTManager*

**Object Creation**

**Object update**

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectIssueResourceMilestoneEvent*

Retrieve a single object.

**Parameters**

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. *sudo*)

**Returns**

The generated *RESTObject*.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

*ProjectIssueResourceMilestoneEvent*

```
class gitlab.v4.objects.ProjectIssueResourceStateEvent(manager: RESTManager, attrs: Dict[str,  
                                                    Any], *, created_from_list: bool = False,  
                                                    lazy: bool = False)
```

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectIssueResourceStateEventManager(gl: Gitlab, parent: RESTObject |  
                                                             None = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectIssueResourceStateEvent*

Retrieve a single object.

**Parameters**

- **id** (*str* | *int*) – ID of the object to retrieve

- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectIssueResourceStateEvent](#)

```
class gitlab.v4.objects.ProjectIssueResourceWeightEvent(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectIssueResourceWeightEventManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectIssueResourceWeightEvent
```

Retrieve a single object.

**Parameters**

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request



**Return type***ProjectIssueResourceWeightEvent*

```
class gitlab.v4.objects.ProjectIssuesStatistics(manager: RESTManager, attrs: Dict[str, Any], *,
                                              created_from_list: bool = False, lazy: bool = False)
```

Bases: *RefreshMixin*, *RESTObject*

**Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectIssuesStatisticsManager(gl: Gitlab, parent: RESTObject | None =
                                                    None)
```

Bases: *GetWithoutIdMixin*, *RESTManager*

**Object listing filters**

- **iids**

**Object Creation****Object update****Parameters**

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)

```
get(**kwargs: Any) → ProjectIssuesStatistics
```

Retrieve a single object.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated *RESTObject*

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type***ProjectIssuesStatistics*

```
class gitlab.v4.objects.ProjectIterationManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *ListMixin*, *RESTManager*

**Object listing filters**

- **state**
- **search**
- **include\_ancestors**

**Object Creation****Object update**

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
class gitlab.v4.objects.ProjectJob(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [RefreshMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[str, Any])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
artifact(path: str, streamed: bool = False, action: Callable[[...], Any] | None = None, chunk_size: int = 1024, *, iterator: bool = False, **kwargs: Any) → bytes | Iterator[Any] | None
```

Get a single artifact file from within the job’s artifacts archive.

**Parameters**

- **path** (*str*) – Path of the artifact
- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk\_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable*[[...], Any] | *None*) – Callable responsible of dealing with chunk of data
- **chunk\_size** (*int*) – Size of each chunk
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the artifacts could not be retrieved

**Returns**

The artifacts if *streamed* is False, None otherwise.

**Return type**

bytes | *Iterator*[Any] | None

```
artifacts(streamed: bool = False, action: Callable[[...], Any] | None = None, chunk_size: int = 1024, *, iterator: bool = False, **kwargs: Any) → bytes | Iterator[Any] | None
```

Get the job artifacts.

**Parameters**

- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk\_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable*[[...], Any] | *None*) – Callable responsible of dealing with chunk of data

- **chunk\_size** (*int*) – Size of each chunk
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the artifacts could not be retrieved

**Returns**

The artifacts if *streamed* is False, None otherwise.

**Return type**

bytes | *Iterator*[*Any*] | None

**cancel**(\*\*kwargs: *Any*) → Dict[str, *Any*]

Cancel the job.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabJobCancelError** – If the job could not be canceled

**Return type**

Dict[str, *Any*]

**delete\_artifacts**(\*\*kwargs: *Any*) → None

Delete artifacts of a job.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the request could not be performed

**Return type**

None

**erase**(\*\*kwargs: *Any*) → None

Erase the job (remove job artifacts and trace).

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabJobEraseError** – If the job could not be erased

**Return type**

None

**keep\_artifacts**(\*\*kwargs: *Any*) → None

Prevent artifacts from being deleted when expiration is set.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the request could not be performed

**Return type**

None

**play**(\*\*kwargs: Any) → None

Trigger a job explicitly.

**Parameters**

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabJobPlayError** – If the job could not be triggered

**Return type**

None

**retry**(\*\*kwargs: Any) → Dict[str, Any]

Retry the job.

**Parameters**

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabJobRetryError** – If the job could not be retried

**Return type**

Dict[str, Any]

**trace**(streamed: bool = False, action: Callable[[...], Any] | None = None, chunk\_size: int = 1024, \*,  
iterator: bool = False, \*\*kwargs: Any) → Dict[str, Any]

Get the job trace.

**Parameters**

- **streamed** (bool) – If True the data will be processed by chunks of *chunk\_size* and each chunk is passed to *action* for treatment
- **iterator** (bool) – If True directly return the underlying response iterator
- **action** (Callable[[...], Any] | None) – Callable responsible of dealing with chunk of data
- **chunk\_size** (int) – Size of each chunk
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the artifacts could not be retrieved

**Returns**

The trace

**Return type***Dict[str, Any]*

```
class gitlab.v4.objects.ProjectJobManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *RetrieveMixin*, *RESTManager*

**Object listing filters**

- `scope`

**Object Creation****Object update****Parameters**

- `gl` (*Gitlab*)
- `parent` (*RESTObject* | *None*)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectJob
```

Retrieve a single object.

**Parameters**

- `id` (*str* | *int*) – ID of the object to retrieve
- `lazy` (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- `**kwargs` (*Any*) – Extra options to send to the server (e.g. `sudo`)

**Returns**

The generated *RESTObject*.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type***ProjectJob*

```
class gitlab.v4.objects.ProjectJobTokenScope(manager: RESTManager, attrs: Dict[str, Any], *,
                                              created_from_list: bool = False, lazy: bool = False)
```

Bases: *RefreshMixin*, *SaveMixin*, *RESTObject*

**Parameters**

- `manager` (*RESTManager*)
- `attrs` (*Dict[str, Any]*)
- `created_from_list` (*bool*)
- `lazy` (*bool*)

```
class gitlab.v4.objects.ProjectJobTokenScopeManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *GetWithoutIdMixin*, *UpdateMixin*, *RESTManager*

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(\*\*kwargs: [Any](#)) → [ProjectJobTokenScope](#)

Retrieve a single object.

**Parameters**

**\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated [RESTObject](#)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectJobTokenScope](#)

**class** gitlab.v4.objects.**ProjectKey**(manager: [RESTManager](#), attrs: *Dict[str, Any]*, \*, created\_from\_list: *bool = False*, lazy: *bool = False*)

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**ProjectKeyManager**(gl: [Gitlab](#), parent: [RESTObject](#) | *None = None*)

Bases: [CRUDMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- **title**
- **key**

Optional attributes for object create:

- **can\_push**

**Object update**

Optional attributes for object update:

- **title**
- **can\_push**

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**enable**(*key\_id*: int, *\*\*kwargs*: Any) → Dict[str, Any] | Response

Enable a deploy key for a project.

#### Parameters

- **key\_id** (int) – The ID of the key to enable
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabProjectDeployKeyError** – If the key could not be enabled

#### Returns

A dict of the result.

#### Return type

Dict[str, Any] | Response

**get**(*id*: str | int, *lazy*: bool = False, *\*\*kwargs*: Any) → ProjectKey

Retrieve a single object.

#### Parameters

- **id** (str | int) – ID of the object to retrieve
- **lazy** (bool) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

ProjectKey

```
class gitlab.v4.objects.ProjectLabel(manager: RESTManager, attrs: Dict[str, Any], *,
                                     created_from_list: bool = False, lazy: bool = False)
```

Bases: PromoteMixin, SubscribableMixin, SaveMixin, ObjectDeleteMixin, RESTObject

#### Parameters

- **manager** (ProjectLabelManager)
- **attrs** (Dict[str, Any])
- **created\_from\_list** (bool)
- **lazy** (bool)

**manager:** ProjectLabelManager

**save**(*\*\*kwargs*: Any) → None

Saves the changes made to the object to the server.

The object is updated to match what the server returns.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct.
- **GitlabUpdateError** – If the server cannot perform the request.

**Return type**

None

**class** gitlab.v4.objects.**ProjectLabelManager**(gl: *Gitlab*, parent: *RESTObject* | *None* = *None*)

Bases: *RetrieveMixin*, *CreateMixin*, *UpdateMixin*, *DeleteMixin*, *RESTManager*

**Object Creation**

Required attributes for object create:

- name
- color

Optional attributes for object create:

- description
- priority

**Object update**

Required attributes for object update:

- name

Optional attributes for object update:

- new\_name
- color
- description
- priority

**Parameters**

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)

**get**(id: *str* | *int*, lazy: *bool* = *False*, \*\*kwargs: *Any*) → *ProjectLabel*

Retrieve a single object.

**Parameters**

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**



- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectLabel](#)

**update**(*name*: *str* | *None*, *new\_data*: *Dict*[*str*, *Any*] | *None* = *None*, *\*\*kwargs*: *Any*) → *Dict*[*str*, *Any*]

Update a Label on the server.

**Parameters**

- **name** (*str* | *None*) – The name of the label
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)
- **new\_data** (*Dict*[*str*, *Any*] | *None*)
- **\*\*kwargs**

**Return type**

*Dict*[*str*, *Any*]

**class** gitlab.v4.objects.**ProjectManager**(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | *None* = *None*)

Bases: [CRUDMixin](#), [RESTManager](#)

**Object listing filters**

- archived
- id\_after
- id\_before
- last\_activity\_after
- last\_activity\_before
- membership
- min\_access\_level
- order\_by
- owned
- repository\_checksum\_failed
- repository\_storage
- search\_namespaces
- search
- simple
- sort
- starred
- statistics
- topic
- visibility
- wiki\_checksum\_failed
- with\_custom\_attributes

- `with_issues_enabled`
- `with_merge_requests_enabled`
- `with_programming_language`

### Object Creation

Optional attributes for object create:

- `name`
- `path`
- `allow_merge_on_skipped_pipeline`
- `only_allow_merge_if_all_status_checks_passed`
- `analytics_access_level`
- `approvals_before_merge`
- `auto_cancel_pending_pipelines`
- `auto_devops_deploy_strategy`
- `auto_devops_enabled`
- `autoclose_referenced_issues`
- `avatar`
- `build_coverage_regex`
- `build_git_strategy`
- `build_timeout`
- `builds_access_level`
- `ci_config_path`
- `container_expiration_policy_attributes`
- `container_registry_access_level`
- `container_registry_enabled`
- `default_branch`
- `description`
- `emails_disabled`
- `external_authorization_classification_label`
- `forking_access_level`
- `group_with_project_templates_id`
- `import_url`
- `initialize_with_readme`
- `issues_access_level`
- `issues_enabled`
- `jobs_enabled`
- `lfs_enabled`

- `merge_method`
- `merge_pipelines_enabled`
- `merge_requests_access_level`
- `merge_requests_enabled`
- `mirror_trigger_builds`
- `mirror`
- `namespace_id`
- `operations_access_level`
- `only_allow_merge_if_all_discussions_are_resolved`
- `only_allow_merge_if_pipeline_succeeds`
- `packages_enabled`
- `pages_access_level`
- `requirements_access_level`
- `printing_merge_request_link_enabled`
- `public_builds`
- `releases_access_level`
- `environments_access_level`
- `feature_flags_access_level`
- `infrastructure_access_level`
- `monitor_access_level`
- `remove_source_branch_after_merge`
- `repository_access_level`
- `repository_storage`
- `request_access_enabled`
- `resolve_outdated_diff_discussions`
- `security_and_compliance_access_level`
- `shared_runners_enabled`
- `show_default_award_emojis`
- `snippets_access_level`
- `snippets_enabled`
- `squash_option`
- `tag_list`
- `topics`
- `template_name`
- `template_project_id`
- `use_custom_template`

- visibility
- wiki\_access\_level
- wiki\_enabled

### Object update

Optional attributes for object update:

- allow\_merge\_on\_skipped\_pipeline
- only\_allow\_merge\_if\_all\_status\_checks\_passed
- analytics\_access\_level
- approvals\_before\_merge
- auto\_cancel\_pending\_pipelines
- auto\_devops\_deploy\_strategy
- auto\_devops\_enabled
- autoclose\_referenced\_issues
- avatar
- build\_coverage\_regex
- build\_git\_strategy
- build\_timeout
- builds\_access\_level
- ci\_config\_path
- ci\_default\_git\_depth
- ci\_forward\_deployment\_enabled
- ci\_allow\_fork\_pipelines\_to\_run\_in\_parent\_project
- ci\_separated\_caches
- container\_expiration\_policy\_attributes
- container\_registry\_access\_level
- container\_registry\_enabled
- default\_branch
- description
- emails\_disabled
- enforce\_auth\_checks\_on\_uploads
- external\_authorization\_classification\_label
- forking\_access\_level
- import\_url
- issues\_access\_level
- issues\_enabled
- issues\_template

- `jobs_enabled`
- `keep_latest_artifact`
- `lfs_enabled`
- `merge_commit_template`
- `merge_method`
- `merge_pipelines_enabled`
- `merge_requests_access_level`
- `merge_requests_enabled`
- `merge_requests_template`
- `merge_trains_enabled`
- `mirror_overwrites_diverged_branches`
- `mirror_trigger_builds`
- `mirror_user_id`
- `mirror`
- `mr_default_target_self`
- `name`
- `operations_access_level`
- `only_allow_merge_if_all_discussions_are_resolved`
- `only_allow_merge_if_pipeline_succeeds`
- `only_mirror_protected_branches`
- `packages_enabled`
- `pages_access_level`
- `requirements_access_level`
- `restrict_user_defined_variables`
- `path`
- `public_builds`
- `releases_access_level`
- `environments_access_level`
- `feature_flags_access_level`
- `infrastructure_access_level`
- `monitor_access_level`
- `remove_source_branch_after_merge`
- `repository_access_level`
- `repository_storage`
- `request_access_enabled`
- `resolve_outdated_diff_discussions`

- `security_and_compliance_access_level`
- `service_desk_enabled`
- `shared_runners_enabled`
- `show_default_award_emojis`
- `snippets_access_level`
- `snippets_enabled`
- `issue_branch_template`
- `squash_commit_template`
- `squash_option`
- `suggestion_commit_message`
- `tag_list`
- `topics`
- `visibility`
- `wiki_access_level`
- `wiki_enabled`

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *Project*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[Project](#)

```
import_bitbucket_server(bitbucket_server_url: str, bitbucket_server_username: str,  
                        personal_access_token: str, bitbucket_server_project: str,  
                        bitbucket_server_repo: str, new_name: str | None = None, target_namespace:  
                        str | None = None, **kwargs: Any) → Dict[str, Any] | Response
```

Import a project from BitBucket Server to Gitlab (schedule the import)

This method will return when an import operation has been safely queued, or an error has occurred. After triggering an import, check the `import_status` of the newly created project to detect when the import operation has completed.

---

**Note:** This request may take longer than most other API requests. So this method will specify a 60 second default timeout if none is specified. A timeout can be specified via kwargs to override this functionality.

---

#### Parameters

- **bitbucket\_server\_url** (*str*) – Bitbucket Server URL
- **bitbucket\_server\_username** (*str*) – Bitbucket Server Username
- **personal\_access\_token** (*str*) – Bitbucket Server personal access token/password
- **bitbucket\_server\_project** (*str*) – Bitbucket Project Key
- **bitbucket\_server\_repo** (*str*) – Bitbucket Repository Name
- **new\_name** (*str* / *None*) – New repository name (Optional)
- **target\_namespace** (*str* / *None*) – Namespace to import repository into. Supports subgroups like /namespace/subgroup (Optional)
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the server failed to perform the request

#### Returns

A representation of the import status.

#### Return type

*Dict[str, Any] | Response*

Example:

```
gl = gitlab.Gitlab_from_config()
print("Triggering import")
result = gl.projects.import_bitbucket_server(
    bitbucket_server_url="https://some.server.url",
    bitbucket_server_username="some_bitbucket_user",
    personal_access_token="my_password_or_access_token",
    bitbucket_server_project="my_project",
    bitbucket_server_repo="my_repo",
    new_name="gl_project_name",
    target_namespace="gl_project_path"
)
project = gl.projects.get(ret['id'])
print("Waiting for import to complete")
while project.import_status == u'started':
    time.sleep(1.0)
```

(continues on next page)

(continued from previous page)

```
project = gl.projects.get(project.id)
print("BitBucket import complete")
```

**import\_github**(*personal\_access\_token*: str, *repo\_id*: int, *target\_namespace*: str, *new\_name*: str | None = None, *github\_hostname*: str | None = None, *optional\_stages*: Dict[str, bool] | None = None, *\*\*kwargs*: Any) → Dict[str, Any] | Response

Import a project from Github to Gitlab (schedule the import)

This method will return when an import operation has been safely queued, or an error has occurred. After triggering an import, check the `import_status` of the newly created project to detect when the import operation has completed.

---

**Note:** This request may take longer than most other API requests. So this method will specify a 60 second default timeout if none is specified. A timeout can be specified via `kwargs` to override this functionality.

---

### Parameters

- **personal\_access\_token** (str) – GitHub personal access token
- **repo\_id** (int) – Github repository ID
- **target\_namespace** (str) – Namespace to import repo into
- **new\_name** (str | None) – New repo name (Optional)
- **github\_hostname** (str | None) – Custom GitHub Enterprise hostname. Do not set for GitHub.com. (Optional)
- **optional\_stages** (Dict[str, bool] | None) – Additional items to import. (Optional)
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the server failed to perform the request

### Returns

A representation of the import status.

### Return type

Dict[str, Any] | Response

Example:

```
gl = gitlab.Gitlab_from_config()
print("Triggering import")
result = gl.projects.import_github(ACCESS_TOKEN,
                                   123456,
                                   "my-group/my-subgroup")
project = gl.projects.get(ret['id'])
print("Waiting for import to complete")
while project.import_status == u'started':
    time.sleep(1.0)
```

(continues on next page)



(continued from previous page)

```
project = gl.projects.get(project.id)
print("Github import complete")
```

**import\_project**(*file: str, path: str, name: str | None = None, namespace: str | None = None, overwrite: bool = False, override\_params: Dict[str, Any] | None = None, \*\*kwargs: Any*) → Dict[str, Any] | Response

Import a project from an archive file.

#### Parameters

- **file** (*str*) – Data or file object containing the project
- **path** (*str*) – Name and path for the new project
- **name** (*str* | *None*) – The name of the project to import. If not provided, defaults to the path of the project.
- **namespace** (*str* | *None*) – The ID or path of the namespace that the project will be imported to
- **overwrite** (*bool*) – If True overwrite an existing project with the same path
- **override\_params** (*Dict[str, Any]* | *None*) – Set the specific settings for the project
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabImportError** – If the server failed to perform the request

#### Returns

A representation of the import status.

#### Return type

Dict[str, Any] | Response

**remote\_import**(*url: str, path: str, name: str | None = None, namespace: str | None = None, overwrite: bool = False, override\_params: Dict[str, Any] | None = None, \*\*kwargs: Any*) → Dict[str, Any] | Response

Import a project from an archive file stored on a remote URL.

#### Parameters

- **url** (*str*) – URL for the file containing the project data to import
- **path** (*str*) – Name and path for the new project
- **name** (*str* | *None*) – The name of the project to import. If not provided, defaults to the path of the project.
- **namespace** (*str* | *None*) – The ID or path of the namespace that the project will be imported to
- **overwrite** (*bool*) – If True overwrite an existing project with the same path
- **override\_params** (*Dict[str, Any]* | *None*) – Set the specific settings for the project
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabImportError** – If the server failed to perform the request

**Returns**

A representation of the import status.

**Return type**

*Dict[str, Any] | Response*

```
remote_import_s3(path: str, region: str, bucket_name: str, file_key: str, access_key_id: str,
                  secret_access_key: str, name: str | None = None, namespace: str | None = None,
                  overwrite: bool = False, override_params: Dict[str, Any] | None = None, **kwargs:
                  Any) → Dict[str, Any] | Response
```

Import a project from an archive file stored on AWS S3.

**Parameters**

- **region** (*str*) – AWS S3 region name where the file is stored
- **bucket\_name** (*str*) – AWS S3 bucket name where the file is stored
- **file\_key** (*str*) – AWS S3 file key to identify the file.
- **access\_key\_id** (*str*) – AWS S3 access key ID.
- **secret\_access\_key** (*str*) – AWS S3 secret access key.
- **path** (*str*) – Name and path for the new project
- **name** (*str* | *None*) – The name of the project to import. If not provided, defaults to the path of the project.
- **namespace** (*str* | *None*) – The ID or path of the namespace that the project will be imported to
- **overwrite** (*bool*) – If True overwrite an existing project with the same path
- **override\_params** (*Dict[str, Any]* | *None*) – Set the specific settings for the project
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabImportError** – If the server failed to perform the request

**Returns**

A representation of the import status.

**Return type**

*Dict[str, Any] | Response*

```
class gitlab.v4.objects.ProjectMember(manager: RESTManager, attrs: Dict[str, Any], *,
                                       created_from_list: bool = False, lazy: bool = False)
```

Bases: *SaveMixin, ObjectDeleteMixin, RESTObject*

**Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)

- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectMemberAll(manager: RESTManager, attrs: Dict[str, Any], *,
                                          created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectMemberAllManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

#### Object Creation

##### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectMemberAll
```

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[ProjectMemberAll](#)

```
class gitlab.v4.objects.ProjectMemberManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [CRUDMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **access\_level**
- **user\_id**

Optional attributes for object create:

- **expires\_at**

- `tasks_to_be_done`

#### Object update

Required attributes for object update:

- `access_level`

Optional attributes for object update:

- `expires_at`

#### Parameters

- `gl` (`Gitlab`)
- `parent` (`RESTObject` | `None`)

`get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectMember`

Retrieve a single object.

#### Parameters

- `id` (`str` | `int`) – ID of the object to retrieve
- `lazy` (`bool`) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- `**kwargs` (`Any`) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated `RESTObject`.

#### Raises

- `GitlabAuthenticationError` – If authentication is not correct
- `GitlabGetError` – If the server cannot perform the request

#### Return type

`ProjectMember`

```
class gitlab.v4.objects.ProjectMergeRequest(manager: RESTManager, attrs: Dict[str, Any], *,
                                             created_from_list: bool = False, lazy: bool = False)
```

Bases: `SubscribableMixin`, `TodoMixin`, `TimeTrackingMixin`, `ParticipantsMixin`, `SaveMixin`, `ObjectDeleteMixin`, `RESTObject`

#### Parameters

- `manager` (`RESTManager`)
- `attrs` (`Dict[str, Any]`)
- `created_from_list` (`bool`)
- `lazy` (`bool`)

`approval_rules:` `ProjectMergeRequestApprovalRuleManager`

`approval_state:` `ProjectMergeRequestApprovalStateManager`

`approvals:` `ProjectMergeRequestApprovalManager`

**approve**(*sha*: *str* | *None* = *None*, **\*\*kwargs**: *Any*) → Dict[*str*, *Any*]

Approve the merge request.

#### Parameters

- **sha** (*str* | *None*) – Head SHA of MR
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabMRApprovalError** – If the approval failed

#### Returns

A dict containing the result.

#### Return type

Dict[*str*, *Any*]

[https://docs.gitlab.com/ee/api/merge\\_request\\_approvals.html#approve-merge-request](https://docs.gitlab.com/ee/api/merge_request_approvals.html#approve-merge-request)

**awardemojis**: *ProjectMergeRequestAwardEmojiManager*

**cancel\_merge\_when\_pipeline\_succeeds**(**\*\*kwargs**: *Any*) → Dict[*str*, *str*]

Cancel merge when the pipeline succeeds.

#### Parameters

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabMROnBuildSuccessError** – If the server could not handle the request

#### Returns

dict of the parsed json returned by the server

#### Return type

Dict[*str*, *str*]

**changes**(**\*\*kwargs**: *Any*) → Dict[*str*, *Any*] | Response

List the merge request changes.

#### Parameters

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the list could not be retrieved

#### Returns

List of changes

#### Return type

Dict[*str*, *Any*] | Response

**closes\_issues**(**\*\*kwargs**: *Any*) → *RESTObjectList*

List issues that will close on merge.”

#### Parameters

- **all** – If True, return all the items, without pagination
- **per\_page** – Number of items to retrieve per request
- **page** – ID of the page to return (starts with page 1)
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the list could not be retrieved

**Returns**

List of issues

**Return type**

*RESTObjectList*

**commits**(*\*\*kwargs: Any*) → *RESTObjectList*

List the merge request commits.

**Parameters**

- **all** – If True, return all the items, without pagination
- **per\_page** – Number of items to retrieve per request
- **page** – ID of the page to return (starts with page 1)
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the list could not be retrieved

**Returns**

The list of commits

**Return type**

*RESTObjectList*

**diffs:** *ProjectMergeRequestDiffManager*

**discussions:** *ProjectMergeRequestDiscussionManager*

**draft\_notes:** *ProjectMergeRequestDraftNoteManager*

**merge**(*merge\_commit\_message: str | None = None, should\_remove\_source\_branch: bool | None = None, merge\_when\_pipeline\_succeeds: bool | None = None, \*\*kwargs: Any*) → *Dict[str, Any]*

Accept the merge request.

**Parameters**

- **merge\_commit\_message** (*str | None*) – Commit message
- **should\_remove\_source\_branch** (*bool | None*) – If True, removes the source branch
- **merge\_when\_pipeline\_succeeds** (*bool | None*) – Wait for the build to succeed, then merge
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabMRClosedError** – If the merge failed

**Return type***Dict[str, Any]***merge\_ref**(\*\*kwargs: Any) → Dict[str, Any] | Response

Attempt to merge changes between source and target branches into *refs/merge-requests/:iid/merge*.

**Parameters****\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)**Raises****GitlabGetError** – If cannot be merged**Return type***Dict[str, Any] | Response***notes:** *ProjectMergeRequestNoteManager***pipelines:** *ProjectMergeRequestPipelineManager***rebase**(\*\*kwargs: Any) → Dict[str, Any] | Response

Attempt to rebase the source branch onto the target branch

**Parameters****\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabMRRebaseError** – If rebasing failed

**Return type***Dict[str, Any] | Response***reset\_approvals**(\*\*kwargs: Any) → Dict[str, Any] | Response

Clear all approvals of the merge request.

**Parameters****\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabMRResetApprovalError** – If reset approval failed

**Return type***Dict[str, Any] | Response***resourcelabelevents:** *ProjectMergeRequestResourceLabelEventManager***resourcemilestoneevents:** *ProjectMergeRequestResourceMilestoneEventManager***resourcestateevents:** *ProjectMergeRequestResourceStateEventManager*

**reviewer\_details:** *ProjectMergeRequestReviewerDetailManager*

**unapprove**(\*\*kwargs: Any) → None

Unapprove the merge request.

**Parameters**

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabMRApprovalError** – If the unapproval failed

**Return type**

None

[https://docs.gitlab.com/ee/api/merge\\_request\\_approvals.html#unapprove-merge-request](https://docs.gitlab.com/ee/api/merge_request_approvals.html#unapprove-merge-request)

```
class gitlab.v4.objects.ProjectMergeRequestApproval(manager: RESTManager, attrs: Dict[str, Any],
*, created_from_list: bool = False, lazy: bool = False)
```

Bases: *SaveMixin, RESTObject*

**Parameters**

- **manager** (RESTManager)
- **attrs** (Dict[str, Any])
- **created\_from\_list** (bool)
- **lazy** (bool)

```
class gitlab.v4.objects.ProjectMergeRequestApprovalManager(gl: Gitlab, parent: RESTObject | None
= None)
```

Bases: *GetWithoutIdMixin, UpdateMixin, RESTManager*

**Object Creation**

**Object update**

Required attributes for object update:

- **approvals\_required**

**Parameters**

- **gl** (Gitlab)
- **parent** (RESTObject | None)

**get**(\*\*kwargs: Any) → *ProjectMergeRequestApproval*

Retrieve a single object.

**Parameters**

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct



- **GitlabGetError** – If the server cannot perform the request

**Return type**`ProjectMergeRequestApproval`

**set\_approvers**(*approvals\_required: int*, *approver\_ids: List[int] | None = None*, *approver\_group\_ids: List[int] | None = None*, *approval\_rule\_name: str = 'name'*, *\*\*kwargs: Any*) → `RESTObject`

Change MR-level allowed approvers and approver groups.

**Parameters**

- **approvals\_required** (*int*) – The number of required approvals for this rule
- **approver\_ids** (*List[int] | None*) – User IDs that can approve MRs
- **approver\_group\_ids** (*List[int] | None*) – Group IDs whose members can approve MRs
- **approval\_rule\_name** (*str*)
- **kwargs** (*Any*)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUpdateError** – If the server failed to perform the request

**Return type**`RESTObject`

**class** gitlab.v4.objects.**ProjectMergeRequestApprovalRule**(*manager: RESTManager*, *attrs: Dict[str, Any]*, *\*, created\_from\_list: bool = False*, *lazy: bool = False*)

Bases: `SaveMixin`, `ObjectDeleteMixin`, `RESTObject`

**Parameters**

- **manager** (`RESTManager`)
- **attrs** (`Dict[str, Any]`)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**approval\_rule\_id:** `int`

**id:** `int`

**merge\_request\_iid:** `int`

**save**(*\*\*kwargs: Any*) → `None`

Save the changes made to the object to the server.

The object is updated to match what the server returns.

**Parameters**

- **kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUpdateError** – If the server cannot perform the request

**Return type**

None

```
class gitlab.v4.objects.ProjectMergeRequestApprovalRuleManager(gl: Gitlab, parent: RESTObject |  
                                                                None = None)
```

Bases: [CRUDMixin](#), [RESTManager](#)**Object Creation**

Required attributes for object create:

- id
- merge\_request\_iid
- name
- approvals\_required

Optional attributes for object create:

- approval\_project\_rule\_id
- user\_ids
- group\_ids

**Object update**

Required attributes for object update:

- id
- merge\_request\_iid
- approval\_rule\_id
- name
- approvals\_required

Optional attributes for object update:

- user\_ids
- group\_ids

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
create(data: Dict[str, Any] | None = None, **kwargs: Any) → RESTObject
```

Create a new object.

**Parameters**

- **data** (*Dict[str, Any]* | *None*) – Parameters to send to the server to create the resource
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo` or `'ref_name'`, `'stage'`, `'name'`, `'all'`)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct

- **GitlabCreateError** – If the server cannot perform the request

#### Returns

A new instance of the `manage` object class build with the data sent by the server

#### Return type

`RESTObject`

`get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectMergeRequestApprovalRule`

Retrieve a single object.

#### Parameters

- **id** (`str` / `int`) – ID of the object to retrieve
- **lazy** (`bool`) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (`Any`) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated `RESTObject`.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

`ProjectMergeRequestApprovalRule`

```
class gitlab.v4.objects.ProjectMergeRequestApprovalState(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: `RESTObject`

#### Parameters

- **manager** (`RESTManager`)
- **attrs** (`Dict[str, Any]`)
- **created\_from\_list** (`bool`)
- **lazy** (`bool`)

```
class gitlab.v4.objects.ProjectMergeRequestApprovalStateManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: `GetWithoutIdMixin`, `RESTManager`

#### Object Creation

#### Object update

#### Parameters

- **gl** (`Gitlab`)
- **parent** (`RESTObject` / `None`)

**get**(\*\*kwargs: Any) → *ProjectMergeRequestApprovalState*

Retrieve a single object.

**Parameters**

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

*ProjectMergeRequestApprovalState*

```
class gitlab.v4.objects.ProjectMergeRequestAwardEmoji(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: *ObjectDeleteMixin*, *RESTObject*

**Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectMergeRequestAwardEmojiManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *NoUpdateMixin*, *RESTManager*

**Object Creation**

Required attributes for object create:

- **name**

**Object update**

**Parameters**

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)

**get**(id: str | int, lazy: bool = False, \*\*kwargs: Any) → *ProjectMergeRequestAwardEmoji*

Retrieve a single object.

**Parameters**

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**`ProjectMergeRequestAwardEmoji`

```
class gitlab.v4.objects.ProjectMergeRequestDiff(manager: RESTManager, attrs: Dict[str, Any], *,
                                                created_from_list: bool = False, lazy: bool = False)
```

Bases: `RESTObject`**Parameters**

- **manager** (`RESTManager`)
- **attrs** (`Dict[str, Any]`)
- **created\_from\_list** (`bool`)
- **lazy** (`bool`)

```
class gitlab.v4.objects.ProjectMergeRequestDiffManager(gl: Gitlab, parent: RESTObject | None =
                                                         None)
```

Bases: `RetrieveMixin`, `RESTManager`**Object Creation****Object update****Parameters**

- **gl** (`Gitlab`)
- **parent** (`RESTObject` | `None`)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectMergeRequestDiff
```

Retrieve a single object.

**Parameters**

- **id** (`str` | `int`) – ID of the object to retrieve
- **lazy** (`bool`) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (`Any`) – Extra options to send to the server (e.g. sudo)

**Returns**The generated `RESTObject`.**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**`ProjectMergeRequestDiff`

```
class gitlab.v4.objects.ProjectMergeRequestDiscussion(manager: RESTManager, attrs: Dict[str,
                                                                                       Any], *, created_from_list: bool = False,
                                                                                       lazy: bool = False)
```

Bases: `SaveMixin`, `RESTObject`

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**notes:** [ProjectMergeRequestDiscussionNoteManager](#)

```
class gitlab.v4.objects.ProjectMergeRequestDiscussionManager(gl: Gitlab, parent: RESTObject |  
None = None)
```

Bases: [RetrieveMixin](#), [CreateMixin](#), [UpdateMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- **body**

Optional attributes for object create:

- **created\_at**
- **position**

**Object update**

Required attributes for object update:

- **resolved**

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectMergeRequestDiscussion
```

Retrieve a single object.

**Parameters**

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated [RESTObject](#).

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectMergeRequestDiscussion](#)

```
class gitlab.v4.objects.ProjectMergeRequestDiscussionNote(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

```
class gitlab.v4.objects.ProjectMergeRequestDiscussionNoteManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [GetMixin](#), [CreateMixin](#), [UpdateMixin](#), [DeleteMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **body**

Optional attributes for object create:

- **created\_at**

#### Object update

Required attributes for object update:

- **body**

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | [None](#))

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectMergeRequestDiscussionNote
```

Retrieve a single object.

#### Parameters

- **id** ([str](#) | [int](#)) – ID of the object to retrieve
- **lazy** ([bool](#)) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[ProjectMergeRequestDiscussionNote](#)

```
class gitlab.v4.objects.ProjectMergeRequestDraftNote(manager: RESTManager, attrs: Dict[str, Any],
*, created_from_list: bool = False, lazy: bool = False)
```

Bases: *ObjectDeleteMixin*, *SaveMixin*, *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**publish**(\*\*kwargs: *Any*) → None

#### Parameters

**kwargs** (*Any*)

#### Return type

None

```
class gitlab.v4.objects.ProjectMergeRequestDraftNoteManager(gl: Gitlab, parent: RESTObject |
None = None)
```

Bases: *CRUDMixin*, *RESTManager*

#### Object Creation

Required attributes for object create:

- **note**

Optional attributes for object create:

- **commit\_id**
- **in\_reply\_to\_discussion\_id**
- **position**
- **resolve\_discussion**

#### Object update

Optional attributes for object update:

- **position**

#### Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject* / *None*)

**bulk\_publish**(\*\*kwargs: *Any*) → None

#### Parameters

**kwargs** (*Any*)

#### Return type

None



**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectMergeRequestDraftNote*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*ProjectMergeRequestDraftNote*

**class** gitlab.v4.objects.**ProjectMergeRequestManager**(*gl*: *Gitlab*, *parent*: *RESTObject* | *None* = *None*)

Bases: *CRUDMixin*, *RESTManager*

#### Object listing filters

- state
- order\_by
- sort
- milestone
- view
- labels
- created\_after
- created\_before
- updated\_after
- updated\_before
- scope
- iids
- author\_id
- assignee\_id
- approver\_ids
- approved\_by\_ids
- my\_reaction\_emoji
- source\_branch
- target\_branch
- search

- wip

### Object Creation

Required attributes for object create:

- source\_branch
- target\_branch
- title

Optional attributes for object create:

- allow\_collaboration
- allow\_maintainer\_to\_push
- approvals\_before\_merge
- assignee\_id
- assignee\_ids
- description
- labels
- milestone\_id
- remove\_source\_branch
- reviewer\_ids
- squash
- target\_project\_id

### Object update

Optional attributes for object update:

- target\_branch
- assignee\_id
- title
- description
- state\_event
- labels
- milestone\_id
- remove\_source\_branch
- discussion\_locked
- allow\_maintainer\_to\_push
- squash
- reviewer\_ids

### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) / *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectMergeRequest*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*ProjectMergeRequest*

```
class gitlab.v4.objects.ProjectMergeRequestNote(manager: RESTManager, attrs: Dict[str, Any], *,
                                                created_from_list: bool = False, lazy: bool = False)
```

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**awardemojis**: *ProjectMergeRequestNoteAwardEmojiManager*

```
class gitlab.v4.objects.ProjectMergeRequestNoteAwardEmoji(manager: RESTManager, attrs: Dict[str,
                                                                                          Any], *, created_from_list: bool =
                                                                                          False, lazy: bool = False)
```

Bases: *ObjectDeleteMixin*, *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectMergeRequestNoteAwardEmojiManager(gl: Gitlab, parent: RESTObject
                                                                | None = None)
```

Bases: *NoUpdateMixin*, *RESTManager*

#### Object Creation

Required attributes for object create:

- **name**

#### Object update

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id: str | int, lazy: bool = False, \*\*kwargs: Any*) → [ProjectMergeRequestNoteAwardEmoji](#)

Retrieve a single object.

**Parameters**

- **id** (*str | int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated [RESTObject](#).

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectMergeRequestNoteAwardEmoji](#)

```
class gitlab.v4.objects.ProjectMergeRequestNoteManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [CRUDMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- **body**

**Object update**

Required attributes for object update:

- **body**

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id: str | int, lazy: bool = False, \*\*kwargs: Any*) → [ProjectMergeRequestNote](#)

Retrieve a single object.

**Parameters**

- **id** (*str | int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated [RESTObject](#).

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type***ProjectMergeRequestNote*

```
class gitlab.v4.objects.ProjectMergeRequestPipeline(manager: RESTManager, attrs: Dict[str, Any],
*, created_from_list: bool = False, lazy: bool =
False)
```

Bases: *RESTObject***Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectMergeRequestPipelineManager(gl: Gitlab, parent: RESTObject | None
= None)
```

Bases: *CreateMixin*, *ListMixin*, *RESTManager***Object Creation****Object update****Parameters**

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)

```
class gitlab.v4.objects.ProjectMergeRequestResourceLabelEvent(manager: RESTManager, attrs:
Dict[str, Any], *,
created_from_list: bool = False,
lazy: bool = False)
```

Bases: *RESTObject***Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectMergeRequestResourceLabelEventManager(gl: Gitlab, parent:
RESTObject | None =
None)
```

Bases: *RetrieveMixin*, *RESTManager***Object Creation****Object update****Parameters**

- **gl** (*Gitlab*)

- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectMergeRequestResourceLabelEvent*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. *sudo*)

#### Returns

The generated *RESTObject*.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*ProjectMergeRequestResourceLabelEvent*

```
class gitlab.v4.objects.ProjectMergeRequestResourceMilestoneEvent(manager: RESTManager,  
                                                                    attrs: Dict[str, Any], *,  
                                                                    created_from_list: bool =  
                                                                    False, lazy: bool = False)
```

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectMergeRequestResourceMilestoneEventManager(gl: Gitlab, parent:  
                                                                           RESTObject | None  
                                                                           = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectMergeRequestResourceMilestoneEvent*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.

- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectMergeRequestResourceMilestoneEvent](#)

```
class gitlab.v4.objects.ProjectMergeRequestResourceStateEvent(manager: RESTManager, attrs:
    Dict[str, Any], *,
    created_from_list: bool = False,
    lazy: bool = False)
```

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

```
class gitlab.v4.objects.ProjectMergeRequestResourceStateEventManager(gl: Gitlab, parent:
    RESTObject | None =
    None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectMergeRequestResourceStateEvent
```

Retrieve a single object.

**Parameters**

- **id** ([str](#) | [int](#)) – ID of the object to retrieve
- **lazy** ([bool](#)) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectMergeRequestResourceStateEvent](#)

```
class gitlab.v4.objects.ProjectMergeRequestReviewerDetail(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectMergeRequestReviewerDetailManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [RESTManager](#)

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
class gitlab.v4.objects.ProjectMergeTrain(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectMergeTrainManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [RESTManager](#)

**Object listing filters**

- **scope**

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)



```
class gitlab.v4.objects.ProjectMilestone(manager: RESTManager, attrs: Dict[str, Any], *,
                                         created_from_list: bool = False, lazy: bool = False)
```

Bases: [PromoteMixin](#), [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (bool)
- **lazy** (bool)

```
issues(**kwargs: Any) → RESTObjectList
```

List issues related to this milestone.

#### Parameters

- **all** – If True, return all the items, without pagination
- **per\_page** – Number of items to retrieve per request
- **page** – ID of the page to return (starts with page 1)
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the list could not be retrieved

#### Returns

The list of issues

#### Return type

[RESTObjectList](#)

```
merge_requests(**kwargs: Any) → RESTObjectList
```

List the merge requests related to this milestone.

#### Parameters

- **all** – If True, return all the items, without pagination
- **per\_page** – Number of items to retrieve per request
- **page** – ID of the page to return (starts with page 1)
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the list could not be retrieved

#### Returns

The list of merge requests

#### Return type

[RESTObjectList](#)

**class** gitlab.v4.objects.**ProjectMilestoneManager**(gl: [Gitlab](#), parent: [RESTObject](#) | None = None)

Bases: [CRUDMixin](#), [RESTManager](#)

#### Object listing filters

- `iids`
- `state`
- `search`

#### Object Creation

Required attributes for object create:

- `title`

Optional attributes for object create:

- `description`
- `due_date`
- `start_date`
- `state_event`

#### Object update

Optional attributes for object update:

- `title`
- `description`
- `due_date`
- `start_date`
- `state_event`

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

**get**(id: str | int, lazy: bool = False, \*\*kwargs: Any) → [ProjectMilestone](#)

Retrieve a single object.

#### Parameters

- **id** (str | int) – ID of the object to retrieve
- **lazy** (bool) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**[ProjectMilestone](#)

```
class gitlab.v4.objects.ProjectNote(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[str, Any])
- **created\_from\_list** (bool)
- **lazy** (bool)

```
class gitlab.v4.objects.ProjectNoteManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)**Object Creation**

Required attributes for object create:

- **body**

**Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectNote
```

Retrieve a single object.

**Parameters**

- **id** (str | int) – ID of the object to retrieve
- **lazy** (bool) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Returns**The generated [RESTObject](#).**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**[ProjectNote](#)

```
class gitlab.v4.objects.ProjectNotificationSettings(manager: RESTManager, attrs: Dict[str, Any],
    *, created_from_list: bool = False, lazy: bool =
    False)
```

Bases: [NotificationSettings](#)**Parameters**

- **manager** ([RESTManager](#))

- `attrs` (`Dict[str, Any]`)
- `created_from_list` (`bool`)
- `lazy` (`bool`)

**class** gitlab.v4.objects.**ProjectNotificationSettingsManager**(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | *None* = *None*)

Bases: [NotificationSettingsManager](#)

#### Object Creation

#### Object update

Optional attributes for object update:

- `level`
- `notification_email`
- `new_note`
- `new_issue`
- `reopen_issue`
- `close_issue`
- `reassign_issue`
- `new_merge_request`
- `reopen_merge_request`
- `close_merge_request`
- `reassign_merge_request`
- `merge_merge_request`

#### Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) | *None*)

**get**(\*\**kwargs*: *Any*) → [ProjectNotificationSettings](#)

Retrieve a single object.

#### Parameters

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated [RESTObject](#)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[ProjectNotificationSettings](#)

```
class gitlab.v4.objects.ProjectPackage(manager: RESTManager, attrs: Dict[str, Any], *,
                                       created_from_list: bool = False, lazy: bool = False)
```

Bases: *ObjectDeleteMixin*, *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**package\_files:** *ProjectPackageFileManager*

**pipelines:** *ProjectPackagePipelineManager*

```
class gitlab.v4.objects.ProjectPackageFile(manager: RESTManager, attrs: Dict[str, Any], *,
                                           created_from_list: bool = False, lazy: bool = False)
```

Bases: *ObjectDeleteMixin*, *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectPackageFileManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *DeleteMixin*, *ListMixin*, *RESTManager*

#### Object Creation

#### Object update

#### Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)

```
class gitlab.v4.objects.ProjectPackageManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *ListMixin*, *GetMixin*, *DeleteMixin*, *RESTManager*

#### Object listing filters

- **order\_by**
- **sort**
- **package\_type**
- **package\_name**

#### Object Creation

#### Object update

#### Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectPackage*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*ProjectPackage*

```
class gitlab.v4.objects.ProjectPackagePipeline(manager: RESTManager, attrs: Dict[str, Any], *,
                                              created_from_list: bool = False, lazy: bool = False)
```

Bases: *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectPackagePipelineManager(gl: Gitlab, parent: RESTObject | None =
                                                    None)
```

Bases: *ListMixin*, *RESTManager*

#### Object Creation

#### Object update

#### Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | *None*)

```
class gitlab.v4.objects.ProjectPagesDomain(manager: RESTManager, attrs: Dict[str, Any], *,
                                           created_from_list: bool = False, lazy: bool = False)
```

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**ProjectPagesDomainManager**(gl: [Gitlab](#), parent: [RESTObject](#) | None = None)

Bases: [CRUDMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- domain

Optional attributes for object create:

- certificate
- key

#### Object update

Optional attributes for object update:

- certificate
- key

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

**get**(id: str | int, lazy: bool = False, \*\*kwargs: Any) → [ProjectPagesDomain](#)

Retrieve a single object.

#### Parameters

- **id** (str | int) – ID of the object to retrieve
- **lazy** (bool) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[ProjectPagesDomain](#)

**class** gitlab.v4.objects.**ProjectPipeline**(manager: [RESTManager](#), attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False)

Bases: [RefreshMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (Dict[str, Any])
- **created\_from\_list** (bool)
- **lazy** (bool)

**bridges:** [\*ProjectPipelineBridgeManager\*](#)

**cancel**(\*\*kwargs: [\*Any\*](#)) → Dict[str, [\*Any\*](#)] | Response

Cancel the job.

**Parameters**

**\*\*kwargs** ([\*Any\*](#)) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabPipelineCancelError** – If the request failed

**Return type**

Dict[str, [\*Any\*](#)] | Response

**jobs:** [\*ProjectPipelineJobManager\*](#)

**retry**(\*\*kwargs: [\*Any\*](#)) → Dict[str, [\*Any\*](#)] | Response

Retry the job.

**Parameters**

**\*\*kwargs** ([\*Any\*](#)) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabPipelineRetryError** – If the request failed

**Return type**

Dict[str, [\*Any\*](#)] | Response

**test\_report:** [\*ProjectPipelineTestReportManager\*](#)

**test\_report\_summary:** [\*ProjectPipelineTestReportSummaryManager\*](#)

**variables:** [\*ProjectPipelineVariableManager\*](#)

**class** gitlab.v4.objects.**ProjectPipelineBridge**(manager: [\*RESTManager\*](#), attrs: Dict[str, [\*Any\*](#)], \*,  
created\_from\_list: bool = False, lazy: bool = False)

Bases: [\*RESTObject\*](#)

**Parameters**

- **manager** ([\*RESTManager\*](#))
- **attrs** (Dict[str, [\*Any\*](#)])
- **created\_from\_list** (bool)
- **lazy** (bool)

**class** gitlab.v4.objects.**ProjectPipelineBridgeManager**(gl: [\*Gitlab\*](#), parent: [\*RESTObject\*](#) | None =  
None)

Bases: [\*ListMixin\*](#), [\*RESTManager\*](#)

**Object listing filters**

- scope

**Object Creation**

**Object update**



**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
class gitlab.v4.objects.ProjectPipelineJob(manager: RESTManager, attrs: Dict[str, Any], *,
                                           created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectPipelineJobManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [RESTManager](#)

**Object listing filters**

- scope
- include\_retried

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
class gitlab.v4.objects.ProjectPipelineManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [RetrieveMixin](#), [CreateMixin](#), [DeleteMixin](#), [RESTManager](#)

**Object listing filters**

- scope
- status
- source
- ref
- sha
- yaml\_errors
- name
- username
- order\_by
- sort

**Object Creation**

Required attributes for object create:

- ref

## Object update

### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**create**(*data*: *Dict[str, Any]* | *None* = *None*, *\*\*kwargs*: *Any*) → *ProjectPipeline*

Creates a new object.

### Parameters

- **data** (*Dict[str, Any]* | *None*) – Parameters to send to the server to create the resource
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

### Returns

A new instance of the managed object class build with the data sent by the server

### Return type

[ProjectPipeline](#)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectPipeline*

Retrieve a single object.

### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

### Returns

The generated RESTObject.

### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

### Return type

[ProjectPipeline](#)

**class** gitlab.v4.objects.**ProjectPipelineSchedule**(*manager*: [RESTManager](#), *attrs*: *Dict[str, Any]*, \*, *created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)

- **lazy** (*bool*)

**pipelines:** *ProjectPipelineSchedulePipelineManager*

**play**(\*\*kwargs: *Any*) → Dict[str, *Any*]

Trigger a new scheduled pipeline, which runs immediately. The next scheduled run of this pipeline is not affected.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabPipelinePlayError** – If the request failed

**Return type**

Dict[str, *Any*]

**take\_ownership**(\*\*kwargs: *Any*) → None

Update the owner of a pipeline schedule.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabOwnershipError** – If the request failed

**Return type**

None

**variables:** *ProjectPipelineScheduleVariableManager*

**class** gitlab.v4.objects.**ProjectPipelineScheduleManager**(gl: *Gitlab*, parent: *RESTObject* | None = None)

Bases: *CRUDMixin*, *RESTManager*

**Object Creation**

Required attributes for object create:

- description
- ref
- cron

Optional attributes for object create:

- cron\_timezone
- active

**Object update**

Optional attributes for object update:

- description
- ref
- cron

- `cron_timezone`
- `active`

#### Parameters

- `gl` (`Gitlab`)
- `parent` (`RESTObject` | `None`)

**get**(*id*: `str` | `int`, *lazy*: `bool` = `False`, *\*\*kwargs*: `Any`) → `ProjectPipelineSchedule`

Retrieve a single object.

#### Parameters

- `id` (`str` | `int`) – ID of the object to retrieve
- `lazy` (`bool`) – If `True`, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- *\*\*kwargs* (`Any`) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated `RESTObject`.

#### Raises

- `GitlabAuthenticationError` – If authentication is not correct
- `GitlabGetError` – If the server cannot perform the request

#### Return type

`ProjectPipelineSchedule`

```
class gitlab.v4.objects.ProjectPipelineSchedulePipeline(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: `RESTObject`

#### Parameters

- `manager` (`RESTManager`)
- `attrs` (`Dict[str, Any]`)
- `created_from_list` (`bool`)
- `lazy` (`bool`)

```
class gitlab.v4.objects.ProjectPipelineSchedulePipelineManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: `ListMixin`, `RESTManager`

#### Object Creation

#### Object update

#### Parameters

- `gl` (`Gitlab`)
- `parent` (`RESTObject` | `None`)

```
class gitlab.v4.objects.ProjectPipelineScheduleVariable(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

```
class gitlab.v4.objects.ProjectPipelineScheduleVariableManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [CreateMixin](#), [UpdateMixin](#), [DeleteMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **key**
- **value**

#### Object update

Required attributes for object update:

- **key**
- **value**

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | [None](#))

```
class gitlab.v4.objects.ProjectPipelineTestReport(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

```
class gitlab.v4.objects.ProjectPipelineTestReportManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [GetWithoutIdMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(\*\*kwargs: *Any*) → *ProjectPipelineTestReport*

Retrieve a single object.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated [RESTObject](#)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

*ProjectPipelineTestReport*

```
class gitlab.v4.objects.ProjectPipelineTestReportSummary(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectPipelineTestReportSummaryManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [GetWithoutIdMixin](#), [RESTManager](#)

**Object Creation**

**Object update**

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(\*\*kwargs: *Any*) → *ProjectPipelineTestReportSummary*

Retrieve a single object.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated [RESTObject](#)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**`ProjectPipelineTestReportSummary`

```
class gitlab.v4.objects.ProjectPipelineVariable(manager: RESTManager, attrs: Dict[str, Any], *,
                                              created_from_list: bool = False, lazy: bool = False)
```

Bases: `RESTObject`**Parameters**

- **manager** (`RESTManager`)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectPipelineVariableManager(gl: Gitlab, parent: RESTObject | None =
                                                    None)
```

Bases: `ListMixin`, `RESTManager`**Object Creation****Object update****Parameters**

- **gl** (`Gitlab`)
- **parent** (`RESTObject` | `None`)

```
class gitlab.v4.objects.ProjectProtectedBranch(manager: RESTManager, attrs: Dict[str, Any], *,
                                              created_from_list: bool = False, lazy: bool = False)
```

Bases: `SaveMixin`, `ObjectDeleteMixin`, `RESTObject`**Parameters**

- **manager** (`RESTManager`)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectProtectedBranchManager(gl: Gitlab, parent: RESTObject | None =
                                                    None)
```

Bases: `CRUDMixin`, `RESTManager`**Object Creation**

Required attributes for object create:

- **name**

Optional attributes for object create:

- **push\_access\_level**
- **merge\_access\_level**
- **unprotect\_access\_level**
- **allow\_force\_push**
- **allowed\_to\_push**

- `allowed_to_merge`
- `allowed_to_unprotect`
- `code_owner_approval_required`

### Object update

#### Parameters

- `gl` (`Gitlab`)
- `parent` (`RESTObject` | `None`)

`get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectProtectedBranch`

Retrieve a single object.

#### Parameters

- `id` (`str` | `int`) – ID of the object to retrieve
- `lazy` (`bool`) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- `**kwargs` (`Any`) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated `RESTObject`.

#### Raises

- `GitlabAuthenticationError` – If authentication is not correct
- `GitlabGetError` – If the server cannot perform the request

#### Return type

`ProjectProtectedBranch`

```
class gitlab.v4.objects.ProjectProtectedEnvironment(manager: RESTManager, attrs: Dict[str, Any],
*, created_from_list: bool = False, lazy: bool = False)
```

Bases: `ObjectDeleteMixin`, `RESTObject`

#### Parameters

- `manager` (`RESTManager`)
- `attrs` (`Dict[str, Any]`)
- `created_from_list` (`bool`)
- `lazy` (`bool`)

```
class gitlab.v4.objects.ProjectProtectedEnvironmentManager(gl: Gitlab, parent: RESTObject | None
= None)
```

Bases: `RetrieveMixin`, `CreateMixin`, `DeleteMixin`, `RESTManager`

### Object Creation

Required attributes for object create:

- `name`
- `deploy_access_levels`

Optional attributes for object create:

- `required_approval_count`



- `approval_rules`

### Object update

#### Parameters

- `gl` (`Gitlab`)
- `parent` (`RESTObject` | `None`)

`get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectProtectedEnvironment`

Retrieve a single object.

#### Parameters

- `id` (`str` | `int`) – ID of the object to retrieve
- `lazy` (`bool`) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- `**kwargs` (`Any`) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated `RESTObject`.

#### Raises

- `GitlabAuthenticationError` – If authentication is not correct
- `GitlabGetError` – If the server cannot perform the request

#### Return type

`ProjectProtectedEnvironment`

```
class gitlab.v4.objects.ProjectProtectedTag(manager: RESTManager, attrs: Dict[str, Any], *,
                                             created_from_list: bool = False, lazy: bool = False)
```

Bases: `ObjectDeleteMixin`, `RESTObject`

#### Parameters

- `manager` (`RESTManager`)
- `attrs` (`Dict[str, Any]`)
- `created_from_list` (`bool`)
- `lazy` (`bool`)

```
class gitlab.v4.objects.ProjectProtectedTagManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: `NoUpdateMixin`, `RESTManager`

### Object Creation

Required attributes for object create:

- `name`

Optional attributes for object create:

- `create_access_level`

### Object update

#### Parameters

- `gl` (`Gitlab`)
- `parent` (`RESTObject` | `None`)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectProtectedTag*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*ProjectProtectedTag*

```
class gitlab.v4.objects.ProjectPushRules(manager: RESTManager, attrs: Dict[str, Any], *,
                                          created_from_list: bool = False, lazy: bool = False)
```

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectPushRulesManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *GetWithoutIdMixin*, *CreateMixin*, *UpdateMixin*, *DeleteMixin*, *RESTManager*

#### Object Creation

Optional attributes for object create:

- **author\_email\_regex**
- **branch\_name\_regex**
- **commit\_committer\_check**
- **commit\_message\_negative\_regex**
- **commit\_message\_regex**
- **deny\_delete\_tag**
- **file\_name\_regex**
- **max\_file\_size**
- **member\_check**
- **prevent\_secrets**
- **reject\_unsigned\_commits**

## Object update

Optional attributes for object update:

- `author_email_regex`
- `branch_name_regex`
- `commit_committer_check`
- `commit_message_negative_regex`
- `commit_message_regex`
- `deny_delete_tag`
- `file_name_regex`
- `max_file_size`
- `member_check`
- `prevent_secrets`
- `reject_unsigned_commits`

### Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) / `None`)

**get**(*\*\*kwargs*: [Any](#)) → [ProjectPushRules](#)

Retrieve a single object.

### Parameters

**\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. `sudo`)

### Returns

The generated [RESTObject](#)

### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

### Return type

[ProjectPushRules](#)

```
class gitlab.v4.objects.ProjectRegistryRepository(manager: RESTManager, attrs: Dict\[str, Any\], *,
          created_from_list: bool = False, lazy: bool =
          False)
```

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

### Parameters

- `manager` ([RESTManager](#))
- `attrs` ([Dict\[str, Any\]](#))
- `created_from_list` ([bool](#))
- `lazy` ([bool](#))

tags: [ProjectRegistryTagManager](#)

```
class gitlab.v4.objects.ProjectRegistryRepositoryManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [DeleteMixin](#), [ListMixin](#), [RESTManager](#)

Object Creation

Object update

Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
class gitlab.v4.objects.ProjectRegistryTag(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectRegistryTagManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [DeleteMixin](#), [RetrieveMixin](#), [RESTManager](#)

Object Creation

Object update

Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
delete_in_bulk(name_regex_delete: str, **kwargs: Any) → None
```

Delete Tag in bulk

Parameters

- **name\_regex\_delete** (*str*) – The regex of the name to delete. To delete all tags specify `.*`.
- **keep\_n** – The amount of latest tags of given name to keep.
- **name\_regex\_keep** – The regex of the name to keep. This value overrides any matches from `name_regex`.
- **older\_than** – Tags to delete that are older than the given time, written in human readable form 1h, 1d, 1month.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo`)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the server cannot perform the request

Return type

*None*

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectRegistryTag*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*ProjectRegistryTag*

```
class gitlab.v4.objects.ProjectRelease(manager: RESTManager, attrs: Dict[str, Any], *,
                                       created_from_list: bool = False, lazy: bool = False)
```

Bases: *SaveMixin*, *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**links:** *ProjectReleaseLinkManager*

```
class gitlab.v4.objects.ProjectReleaseLink(manager: RESTManager, attrs: Dict[str, Any], *,
                                           created_from_list: bool = False, lazy: bool = False)
```

Bases: *ObjectDeleteMixin*, *SaveMixin*, *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectReleaseLinkManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: *CRUDMixin*, *RESTManager*

#### Object Creation

Required attributes for object create:

- **name**
- **url**

Optional attributes for object create:

- `filepath`
- `direct_asset_path`
- `link_type`

### Object update

Optional attributes for object update:

- `name`
- `url`
- `filepath`
- `direct_asset_path`
- `link_type`

### Parameters

- `gl` (`Gitlab`)
- `parent` (`RESTObject` | `None`)

`get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectReleaseLink`

Retrieve a single object.

### Parameters

- `id` (`str` | `int`) – ID of the object to retrieve
- `lazy` (`bool`) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- `**kwargs` (`Any`) – Extra options to send to the server (e.g. `sudo`)

### Returns

The generated `RESTObject`.

### Raises

- `GitlabAuthenticationError` – If authentication is not correct
- `GitlabGetError` – If the server cannot perform the request

### Return type

`ProjectReleaseLink`

`class gitlab.v4.objects.ProjectReleaseManager(gl: Gitlab, parent: RESTObject | None = None)`

Bases: `CRUDMixin`, `RESTManager`

### Object listing filters

- `order_by`
- `sort`
- `include_html_description`

### Object Creation

Required attributes for object create:

- `tag_name`

Optional attributes for object create:

- name
- description
- ref
- assets

### Object update

Optional attributes for object update:

- name
- description
- milestones
- released\_at

### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [ProjectRelease](#)

Retrieve a single object.

### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

### Returns

The generated [RESTObject](#).

### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

### Return type

[ProjectRelease](#)

```
class gitlab.v4.objects.ProjectRemoteMirror(manager: RESTManager, attrs: Dict[str, Any], *,
                                             created_from_list: bool = False, lazy: bool = False)
```

Bases: [ObjectDeleteMixin](#), [SaveMixin](#), [RESTObject](#)

### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**ProjectRemoteMirrorManager**(gl: Gitlab, parent: RESTObject | None = None)

Bases: [ListMixin](#), [CreateMixin](#), [UpdateMixin](#), [DeleteMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- url

Optional attributes for object create:

- enabled
- only\_protected\_branches

#### Object update

Optional attributes for object update:

- enabled
- only\_protected\_branches

#### Parameters

- gl ([Gitlab](#))
- parent ([RESTObject](#) | None)

**class** gitlab.v4.objects.**ProjectResourceGroup**(manager: [RESTManager](#), attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False)

Bases: [SaveMixin](#), [RESTObject](#)

#### Parameters

- manager ([RESTManager](#))
- attrs (Dict[str, Any])
- created\_from\_list (bool)
- lazy (bool)

upcoming\_jobs: [ProjectResourceGroupUpcomingJobManager](#)

**class** gitlab.v4.objects.**ProjectResourceGroupManager**(gl: Gitlab, parent: RESTObject | None = None)

Bases: [RetrieveMixin](#), [UpdateMixin](#), [RESTManager](#)

#### Object listing filters

- order\_by
- sort
- include\_html\_description

#### Object Creation

#### Object update

Optional attributes for object update:

- process\_mode

#### Parameters

- gl ([Gitlab](#))



- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectResourceGroup*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated *RESTObject*.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*ProjectResourceGroup*

```
class gitlab.v4.objects.ProjectResourceGroupUpcomingJob(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectResourceGroupUpcomingJobManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
class gitlab.v4.objects.ProjectRunner(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)

- **lazy** (*bool*)

**class** gitlab.v4.objects.**ProjectRunnerManager**(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | *None* = *None*)

Bases: [CreateMixin](#), [DeleteMixin](#), [ListMixin](#), [RESTManager](#)

#### Object listing filters

- **scope**
- **tag\_list**

#### Object Creation

Required attributes for object create:

- **runner\_id**

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**class** gitlab.v4.objects.**ProjectSecureFile**(*manager*: [RESTManager](#), *attrs*: *Dict*[*str*, *Any*], \*,  
*created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

##### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**download**(*streamed*: *bool* = *False*, *action*: *Callable*[[*bytes*], *None*] | *None* = *None*, *chunk\_size*: *int* = 1024, \*,  
*iterator*: *bool* = *False*, **\*\*kwargs**: *Any*) → *bytes* | *Iterator*[*Any*] | *None*

Download the secure file.

##### Parameters

- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk\_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable*[[*bytes*], *None*] | *None*) – Callable responsible of dealing with chunk of data
- **chunk\_size** (*int*) – Size of each chunk
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the artifacts could not be retrieved

##### Returns

The artifacts if *streamed* is False, None otherwise.

##### Return type

*bytes* | *Iterator*[*Any*] | *None*

**class** gitlab.v4.objects.**ProjectSecureFileManager**(gl: [Gitlab](#), parent: [RESTObject](#) | None = None)

Bases: [NoUpdateMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- name
- file

#### Object update

##### Parameters

- gl ([Gitlab](#))
- parent ([RESTObject](#) | None)

**get**(id: str | int, lazy: bool = False, \*\*kwargs: Any) → [ProjectSecureFile](#)

Retrieve a single object.

##### Parameters

- id (str | int) – ID of the object to retrieve
- lazy (bool) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- \*\*kwargs (Any) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated [RESTObject](#).

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

##### Return type

[ProjectSecureFile](#)

**class** gitlab.v4.objects.**ProjectService**(manager: [RESTManager](#), attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False)

Bases: [ProjectIntegration](#)

##### Parameters

- manager ([RESTManager](#))
- attrs (Dict[str, Any])
- created\_from\_list (bool)
- lazy (bool)

**class** gitlab.v4.objects.**ProjectServiceManager**(gl: [Gitlab](#), parent: [RESTObject](#) | None = None)

Bases: [ProjectIntegrationManager](#)

#### Object Creation

#### Object update

##### Parameters

- gl ([Gitlab](#))

- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectService*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. *sudo*)

#### Returns

The generated *RESTObject*.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*ProjectService*

```
class gitlab.v4.objects.ProjectSnippet(manager: RESTManager, attrs: Dict[str, Any], *,  
                                         created_from_list: bool = False, lazy: bool = False)
```

Bases: *UserAgentDetailMixin*, *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

awardemojis: *ProjectSnippetAwardEmojiManager*

```
content(streamed: bool = False, action: Callable[[...], Any] | None = None, chunk_size: int = 1024, *,  
         iterator: bool = False, **kwargs: Any) → bytes | Iterator[Any] | None
```

Return the content of a snippet.

#### Parameters

- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk\_size* and each chunk is passed to *action* for treatment.
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable*[*[...]*, *Any*] | *None*) – Callable responsible of dealing with chunk of data
- **chunk\_size** (*int*) – Size of each chunk
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. *sudo*)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the content could not be retrieved

**Returns**

The snippet content

**Return type**

bytes | *Iterator*[*Any*] | None

**discussions:** [ProjectSnippetDiscussionManager](#)

**notes:** [ProjectSnippetNoteManager](#)

```
class gitlab.v4.objects.ProjectSnippetAwardEmoji(manager: RESTManager, attrs: Dict[str, Any], *,
                                                  created_from_list: bool = False, lazy: bool = False)
```

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectSnippetAwardEmojiManager(gl: Gitlab, parent: RESTObject | None =
                                                         None)
```

Bases: [NoUpdateMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- **name**

**Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectSnippetAwardEmoji
```

Retrieve a single object.

**Parameters**

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated [RESTObject](#).

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectSnippetAwardEmoji](#)

```
class gitlab.v4.objects.ProjectSnippetDiscussion(manager: RESTManager, attrs: Dict[str, Any], *,
                                                  created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

**notes:** [ProjectSnippetDiscussionNoteManager](#)

```
class gitlab.v4.objects.ProjectSnippetDiscussionManager(gl: Gitlab, parent: RESTObject | None =
                                                         None)
```

Bases: [RetrieveMixin](#), [CreateMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **body**

Optional attributes for object create:

- **created\_at**

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | [None](#))

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectSnippetDiscussion
```

Retrieve a single object.

##### Parameters

- **id** ([str](#) | [int](#)) – ID of the object to retrieve
- **lazy** ([bool](#)) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated [RESTObject](#).

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

##### Return type

[ProjectSnippetDiscussion](#)

```
class gitlab.v4.objects.ProjectSnippetDiscussionNote(manager: RESTManager, attrs: Dict[str, Any],
                                                       *, created_from_list: bool = False, lazy: bool =
                                                       False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectSnippetDiscussionNoteManager(gl: Gitlab, parent: RESTObject |
                                                         None = None)
```

Bases: [GetMixin](#), [CreateMixin](#), [UpdateMixin](#), [DeleteMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- **body**

Optional attributes for object create:

- **created\_at**

**Object update**

Required attributes for object update:

- **body**

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectSnippetDiscussionNote
```

Retrieve a single object.

**Parameters**

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated [RESTObject](#).

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectSnippetDiscussionNote](#)

```
class gitlab.v4.objects.ProjectSnippetManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [CRUDMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- `title`
- `visibility`

Optional attributes for object create:

- `description`
- `content`

Mutually exclusive attributes for object create:

- `files`
- `file_name`

### Object update

Optional attributes for object update:

- `title`
- `files`
- `file_name`
- `content`
- `visibility`
- `description`

#### Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) | `None`)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectSnippet*

Retrieve a single object.

#### Parameters

- `id` (*str* | *int*) – ID of the object to retrieve
- `lazy` (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- `**kwargs` (*Any*) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[ProjectSnippet](#)

```
class gitlab.v4.objects.ProjectSnippetNote(manager: RESTManager, attrs: Dict[str, Any], *,
                                             created_from_list: bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters



- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

awardemojis: [ProjectSnippetNoteAwardEmojiManager](#)

```
class gitlab.v4.objects.ProjectSnippetNoteAwardEmoji(manager: RESTManager, attrs: Dict[str, Any],
*, created_from_list: bool = False, lazy: bool =
False)
```

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectSnippetNoteAwardEmojiManager(gl: Gitlab, parent: RESTObject |
None = None)
```

Bases: [NoUpdateMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **name**

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectSnippetNoteAwardEmoji
```

Retrieve a single object.

##### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated [RESTObject](#).

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

##### Return type

[ProjectSnippetNoteAwardEmoji](#)

**class** gitlab.v4.objects.**ProjectSnippetNoteManager**(gl: [Gitlab](#), parent: [RESTObject](#) | *None* = *None*)

Bases: [CRUDMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **body**

#### Object update

Required attributes for object update:

- **body**

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(id: *str* | *int*, lazy: *bool* = *False*, \*\*kwargs: *Any*) → [ProjectSnippetNote](#)

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[ProjectSnippetNote](#)

**class** gitlab.v4.objects.**ProjectStorage**(manager: [RESTManager](#), attrs: *Dict*[*str*, *Any*], \*, created\_from\_list: *bool* = *False*, lazy: *bool* = *False*)

Bases: [RefreshMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**ProjectStorageManager**(gl: [Gitlab](#), parent: [RESTObject](#) | *None* = *None*)

Bases: [GetWithoutIdMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(\*\*kwargs: *Any*) → *ProjectStorage*

Retrieve a single object.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated [RESTObject](#)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectStorage](#)

```
class gitlab.v4.objects.ProjectTag(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectTagManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [NoUpdateMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- tag\_name
- ref

Optional attributes for object create:

- message

**Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(id: *str* | *int*, lazy: bool = False, \*\*kwargs: *Any*) → *ProjectTag*

Retrieve a single object.

**Parameters**

- **id** (*str* | *int*) – ID of the object to retrieve

- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[ProjectTag](#)

```
class gitlab.v4.objects.ProjectTrigger(manager: RESTManager, attrs: Dict[str, Any], *,
                                       created_from_list: bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectTriggerManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [CRUDMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- **description**

**Object update**

Required attributes for object update:

- **description**

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → ProjectTrigger
```

Retrieve a single object.

**Parameters**

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**[ProjectTrigger](#)

```
class gitlab.v4.objects.ProjectUser(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectUserManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [RESTManager](#)**Object listing filters**

- **search**
- **skip\_users**

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
class gitlab.v4.objects.ProjectVariable(manager: RESTManager, attrs: Dict[str, Any], *,
    created_from_list: bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.ProjectVariableManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [CRUDMixin](#), [RESTManager](#)**Object Creation**

Required attributes for object create:

- **key**
- **value**

Optional attributes for object create:

- `protected`
- `variable_type`
- `masked`
- `environment_scope`

### Object update

Required attributes for object update:

- `key`
- `value`

Optional attributes for object update:

- `protected`
- `variable_type`
- `masked`
- `environment_scope`

### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) / `None`)

**get**(*id: str | int, lazy: bool = False, \*\*kwargs: Any*) → *ProjectVariable*

Retrieve a single object.

### Parameters

- **id** (*str* / *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo`)

### Returns

The generated [RESTObject](#).

### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

### Return type

[ProjectVariable](#)

```
class gitlab.v4.objects.ProjectWiki(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [UploadMixin](#), [RESTObject](#)

### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)

- **lazy** (*bool*)

**class** gitlab.v4.objects.**ProjectWikiManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: [CRUDMixin](#), [RESTManager](#)

#### Object listing filters

- **with\_content**

#### Object Creation

Required attributes for object create:

- **title**
- **content**

Optional attributes for object create:

- **format**

#### Object update

Optional attributes for object update:

- **title**
- **content**
- **format**

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id: str | int, lazy: bool = False, \*\*kwargs: Any*) → [ProjectWiki](#)

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[ProjectWiki](#)

**class** gitlab.v4.objects.**RegistryRepository**(*manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False*)

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** `gitlab.v4.objects.RegistryRepositoryManager`(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | *None* = *None*)

Bases: [GetMixin](#), [RESTManager](#)

**Object Creation**

**Object update**

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [RegistryRepository](#)

Retrieve a single object.

**Parameters**

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo`)

**Returns**

The generated [RESTObject](#).

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[RegistryRepository](#)

**class** `gitlab.v4.objects.RepositoryMixin`

Bases: `object`

**delete\_merged\_branches**(*\*\*kwargs*: *Any*) → *None*

Delete merged branches.

**Parameters**

- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. `sudo`)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the server failed to perform the request

**Return type**

*None*



**repository\_archive**(*sha*: *str* | *None* = *None*, *streamed*: *bool* = *False*, *action*: *Callable*[*...*, *Any*] | *None* = *None*, *chunk\_size*: *int* = 1024, *format*: *str* | *None* = *None*, *path*: *str* | *None* = *None*, \*, *iterator*: *bool* = *False*, *\*\*kwargs*: *Any*) → *bytes* | *Iterator*[*Any*] | *None*

Return an archive of the repository.

#### Parameters

- **sha** (*str* | *None*) – ID of the commit (default branch by default)
- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk\_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable*[*...*, *Any*] | *None*) – Callable responsible of dealing with chunk of data
- **chunk\_size** (*int*) – Size of each chunk
- **format** (*str* | *None*) – file format (tar.gz by default)
- **path** (*str* | *None*) – The subpath of the repository to download (all files by default)
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the server failed to perform the request

#### Returns

The binary data of the archive

#### Return type

*bytes* | *Iterator*[*Any*] | *None*

**repository\_blob**(*sha*: *str*, *\*\*kwargs*: *Any*) → *Dict*[*str*, *Any*] | *Response*

Return a file by blob SHA.

#### Parameters

- **sha** (*str*) – ID of the blob
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server failed to perform the request

#### Returns

The blob content and metadata

#### Return type

*Dict*[*str*, *Any*] | *Response*

**repository\_compare**(*from\_*: *str*, *to*: *str*, *\*\*kwargs*: *Any*) → *Dict*[*str*, *Any*] | *Response*

Return a diff between two branches/commits.

#### Parameters

- **from** – Source branch/SHA
- **to** (*str*) – Destination branch/SHA

- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)
- **from\_** (*str*)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server failed to perform the request

**Returns**

The diff

**Return type**

*Dict[str, [Any](#)] | Response*

**repository\_contributors** (\*\*kwargs: [Any](#)) → *GitlabList* | List[Dict[str, [Any](#)]]

Return a list of contributors for the project.

**Parameters**

- **all** – If True, return all the items, without pagination
- **per\_page** – Number of items to retrieve per request
- **page** – ID of the page to return (starts with page 1)
- **iterator** – If set to True and no pagination option is defined, return a generator instead of a list
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server failed to perform the request

**Returns**

The contributors

**Return type**

*GitlabList* | List[Dict[str, [Any](#)]]

**repository\_merge\_base**(refs: List[str], \*\*kwargs: [Any](#)) → Dict[str, [Any](#)] | Response

Return a diff between two branches/commits.

**Parameters**

- **refs** (*List[str]*) – The refs to find the common ancestor of. Multiple refs can be passed.
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server failed to perform the request

**Returns**

The common ancestor commit (*not* a RESTObject)

**Return type**

*Dict[str, [Any](#)] | Response*

**repository\_raw\_blob**(*sha*: str, *streamed*: bool = False, *action*: Callable[[...], Any] | None = None, *chunk\_size*: int = 1024, \*, *iterator*: bool = False, \*\**kwargs*: Any) → bytes | Iterator[Any] | None

Return the raw file contents for a blob.

#### Parameters

- **sha** (str) – ID of the blob
- **streamed** (bool) – If True the data will be processed by chunks of *chunk\_size* and each chunk is passed to *action* for treatment
- **iterator** (bool) – If True directly return the underlying response iterator
- **action** (Callable[[...], Any] | None) – Callable responsible of dealing with chunk of data
- **chunk\_size** (int) – Size of each chunk
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server failed to perform the request

#### Returns

The blob content if streamed is False, None otherwise

#### Return type

bytes | Iterator[Any] | None

**repository\_tree**(*path*: str = "", *ref*: str = "", *recursive*: bool = False, \*\**kwargs*: Any) → GitlabList | List[Dict[str, Any]]

Return a list of files in the repository.

#### Parameters

- **path** (str) – Path of the top folder (/ by default)
- **ref** (str) – Reference to a commit or branch
- **recursive** (bool) – Whether to get the tree recursively
- **all** – If True, return all the items, without pagination
- **per\_page** – Number of items to retrieve per request
- **page** – ID of the page to return (starts with page 1)
- **iterator** – If set to True and no pagination option is defined, return a generator instead of a list
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server failed to perform the request

#### Returns

The representation of the tree

#### Return type

GitlabList | List[Dict[str, Any]]

**update\_submodule**(*submodule: str, branch: str, commit\_sha: str, \*\*kwargs: Any*) → Dict[str, Any] | Response

Update a project submodule

#### Parameters

- **submodule** (*str*) – Full path to the submodule
- **branch** (*str*) – Name of the branch to commit into
- **commit\_sha** (*str*) – Full commit SHA to update the submodule to
- **commit\_message** – Commit message. If no message is provided, a default one will be set (optional)
- **kwargs** (*Any*)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabPutError** – If the submodule could not be updated

#### Return type

Dict[str, Any] | Response

**class** gitlab.v4.objects.**Runner**(*manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False*)

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (Dict[str, Any])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**jobs:** [RunnerJobManager](#)

**class** gitlab.v4.objects.**RunnerAll**(*manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False*)

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (Dict[str, Any])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**RunnerAllManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: [ListMixin](#), [RESTManager](#)

#### Object listing filters

- scope
- type
- status

- paused
- tag\_list

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
class gitlab.v4.objects.RunnerJob(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

##### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.RunnerJobManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [RESTManager](#)

#### Object listing filters

- status

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
class gitlab.v4.objects.RunnerManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [CRUDMixin](#), [RESTManager](#)

#### Object listing filters

- scope
- type
- status
- paused
- tag\_list

#### Object Creation

Required attributes for object create:

- token

Optional attributes for object create:

- description

- `info`
- `active`
- `locked`
- `run_untagged`
- `tag_list`
- `access_level`
- `maximum_timeout`

### Object update

Optional attributes for object update:

- `description`
- `active`
- `tag_list`
- `run_untagged`
- `locked`
- `access_level`
- `maximum_timeout`

### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) / `None`)

**all**(*scope*: `str` | `None` = `None`, **\*\*kwargs**: [Any](#)) → `List[Runner]`

List all the runners.

### Parameters

- **scope** (`str` / `None`) – The scope of runners to show, one of: `specific`, `shared`, `active`, `paused`, `online`
- **all** – If `True`, return all the items, without pagination
- **per\_page** – Number of items to retrieve per request
- **page** – ID of the page to return (starts with page 1)
- **iterator** – If set to `True` and no pagination option is defined, return a generator instead of a list
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. `sudo`)

### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the server failed to perform the request

### Returns

A list of runners matching the scope.

### Return type

`List[Runner]`

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *Runner*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*Runner*

**verify**(*token*: *str*, *\*\*kwargs*: *Any*) → *None*

Validates authentication credentials for a registered Runner.

#### Parameters

- **token** (*str*) – The runner's authentication token
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabVerifyError** – If the server failed to verify the token

#### Return type

*None*

**class** gitlab.v4.objects.SharedProject(*manager*: *RESTManager*, *attrs*: *Dict*[*str*, *Any*], \*, *created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.SharedProjectManager(*gl*: *Gitlab*, *parent*: *RESTObject* | *None* = *None*)

Bases: *ListMixin*, *RESTManager*

#### Object listing filters

- archived
- visibility
- order\_by

- `sort`
- `search`
- `simple`
- `starred`
- `with_issues_enabled`
- `with_merge_requests_enabled`
- `min_access_level`
- `with_custom_attributes`

#### Object Creation

#### Object update

##### Parameters

- `gl` (`Gitlab`)
- `parent` (`RESTObject` | `None`)

**class** `gitlab.v4.objects.SidekiqManager`(`gl`: `Gitlab`, `parent`: `RESTObject` | `None` = `None`)

Bases: `RESTManager`

Manager for the Sidekiq methods.

This manager doesn't actually manage objects but provides helper function for the sidekiq metrics API.

#### Object Creation

#### Object update

##### Parameters

- `gl` (`Gitlab`)
- `parent` (`RESTObject` | `None`)

**compound\_metrics**(*\*\*kwargs*: `Any`) → `Dict[str, Any]` | `Response`

Return all available metrics and statistics.

##### Parameters

**\*\*kwargs** (`Any`) – Extra options to send to the server (e.g. `sudo`)

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the information couldn't be retrieved

##### Returns

All available Sidekiq metrics and statistics

##### Return type

`Dict[str, Any]` | `Response`

**job\_stats**(*\*\*kwargs*: `Any`) → `Dict[str, Any]` | `Response`

Return statistics about the jobs performed.

##### Parameters

**\*\*kwargs** (`Any`) – Extra options to send to the server (e.g. `sudo`)

##### Raises



- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the information couldn't be retrieved

**Returns**

Statistics about the Sidekiq jobs performed

**Return type**

*Dict[str, Any] | Response*

**process\_metrics**(\*\*kwargs: *Any*) → *Dict[str, Any] | Response*

Return the registered sidekiq workers.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the information couldn't be retrieved

**Returns**

Information about the register Sidekiq worker

**Return type**

*Dict[str, Any] | Response*

**queue\_metrics**(\*\*kwargs: *Any*) → *Dict[str, Any] | Response*

Return the registered queues information.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the information couldn't be retrieved

**Returns**

Information about the Sidekiq queues

**Return type**

*Dict[str, Any] | Response*

**class** gitlab.v4.objects.**Snippet**(manager: *RESTManager*, attrs: *Dict[str, Any]*, \*, created\_from\_list: *bool* = *False*, lazy: *bool* = *False*)

Bases: *UserAgentDetailMixin, SaveMixin, ObjectDeleteMixin, RESTObject*

**Parameters**

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**content**(streamed: *bool* = *False*, action: *Callable[[...], Any] | None* = *None*, chunk\_size: *int* = *1024*, \*, iterator: *bool* = *False*, \*\*kwargs: *Any*) → *bytes | Iterator[Any] | None*

Return the content of a snippet.

**Parameters**

- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk\_size* and each chunk is passed to *action* for treatment.
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable[[...], Any] | None*) – Callable responsible of dealing with chunk of data
- **chunk\_size** (*int*) – Size of each chunk
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the content could not be retrieved

**Returns**

The snippet content

**Return type**

bytes | *Iterator*[*Any*] | None

**class** gitlab.v4.objects.**SnippetManager**(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | None = None)

Bases: [CRUDMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- title

Optional attributes for object create:

- description
- content
- visibility

Mutually exclusive attributes for object create:

- files
- file\_name

**Object update**

Optional attributes for object update:

- title
- files
- file\_name
- content
- visibility
- description

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *Snippet*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated RESTObject.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*Snippet*

**public**(*\*\*kwargs*: *Any*) → *RESTObjectList* | *List[RESTObject]*

List all the public snippets.

#### Parameters

- **all** – If True the returned object will be a list
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

**GitlabListError** – If the list could not be retrieved

#### Returns

A generator for the snippets list

#### Return type

*RESTObjectList* | *List[RESTObject]*

**class** gitlab.v4.objects.StarredProject(*manager*: *RESTManager*, *attrs*: *Dict[str, Any]*, \*, *created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.StarredProjectManager(*gl*: *Gitlab*, *parent*: *RESTObject* | *None* = *None*)

Bases: *ListMixin*, *RESTManager*

#### Object listing filters

- archived
- membership
- min\_access\_level

- `order_by`
- `owned`
- `search`
- `simple`
- `sort`
- `starred`
- `statistics`
- `visibility`
- `with_custom_attributes`
- `with_issues_enabled`
- `with_merge_requests_enabled`

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | `None`)

```
class gitlab.v4.objects.Todo(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool = False, lazy: bool = False)
```

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

##### Parameters

- **manager** ([RESTManager](#))
- **attrs** (`Dict[str, Any]`)
- **created\_from\_list** (`bool`)
- **lazy** (`bool`)

```
mark_as_done(**kwargs: Any) → Dict[str, Any]
```

Mark the todo as done.

##### Parameters

**\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. `sudo`)

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabTodoError** – If the server failed to perform the request

##### Returns

A dict with the result

##### Return type

`Dict[str, Any]`

```
class gitlab.v4.objects.TodoManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [DeleteMixin](#), [RESTManager](#)

#### Object listing filters

- action
- author\_id
- project\_id
- state
- type

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**mark\_all\_as\_done**(\*\*kwargs: *Any*) → *None*

Mark all the todos as done.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabTodoError** – If the server failed to perform the request

**Returns**

The number of todos marked done

**Return type**

*None*

**class** gitlab.v4.objects.**Topic**(manager: [RESTManager](#), attrs: *Dict[str, Any]*, \*, created\_from\_list: *bool* = *False*, lazy: *bool* = *False*)

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**TopicManager**(gl: [Gitlab](#), parent: [RESTObject](#) | *None* = *None*)

Bases: [CRUDMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- name

Optional attributes for object create:

- avatar
- description
- title

### Object update

Optional attributes for object update:

- `avatar`
- `description`
- `name`

#### Parameters

- `gl` ([Gitlab](#))
- `parent` ([RESTObject](#) / `None`)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *Topic*

Retrieve a single object.

#### Parameters

- `id` (*str* / *int*) – ID of the object to retrieve
- `lazy` (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- *\*\*kwargs* (*Any*) – Extra options to send to the server (e.g. `sudo`)

#### Returns

The generated `RESTObject`.

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

*Topic*

**merge**(*source\_topic\_id*: *int* | *str*, *target\_topic\_id*: *int* | *str*, *\*\*kwargs*: *Any*) → `Dict[str, Any]`

Merge two topics, assigning all projects to the target topic.

#### Parameters

- `source_topic_id` (*int* / *str*) – ID of source project topic
- `target_topic_id` (*int* / *str*) – ID of target project topic
- *\*\*kwargs* (*Any*) – Extra options to send to the server (e.g. `sudo`)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabTopicMergeError** – If the merge failed

#### Returns

The merged topic data (*not* a `RESTObject`)

#### Return type

`Dict[str, Any]`

```
class gitlab.v4.objects.User(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool =
    False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

**activate**(\*\*kwargs: [Any](#)) → [Dict](#)[[str](#), [Any](#)] | [Response](#)

Activate the user.

#### Parameters

**\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabActivateError** – If the user could not be activated

#### Returns

Whether the user status has been changed

#### Return type

[Dict](#)[[str](#), [Any](#)] | [Response](#)

**approve**(\*\*kwargs: [Any](#)) → [Dict](#)[[str](#), [Any](#)] | [Response](#)

Approve a user creation request.

#### Parameters

**\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUserApproveError** – If the user could not be activated

#### Returns

The new object data (*not* a [RESTObject](#))

#### Return type

[Dict](#)[[str](#), [Any](#)] | [Response](#)

**ban**(\*\*kwargs: [Any](#)) → [Dict](#)[[str](#), [Any](#)] | [Response](#)

Ban the user.

#### Parameters

**\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabBanError** – If the user could not be banned

#### Returns

Whether the user has been banned

**Return type***Dict[str, Any] | Response***block**(\*\*kwargs: Any) → bool | None

Block the user.

**Parameters****\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabBlockError** – If the user could not be blocked

**Returns**

Whether the user status has been changed

**Return type**

bool | None

**customattributes:** *UserCustomAttributeManager***deactivate**(\*\*kwargs: Any) → Dict[str, Any] | Response

Deactivate the user.

**Parameters****\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeactivateError** – If the user could not be deactivated

**Returns**

Whether the user status has been changed

**Return type***Dict[str, Any] | Response***emails:** *UserEmailManager***events:** *UserEventManager***follow**(\*\*kwargs: Any) → Dict[str, Any] | Response

Follow the user.

**Parameters****\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabFollowError** – If the user could not be followed

**Returns**The new object data (*not* a RESTObject)**Return type***Dict[str, Any] | Response***followers\_users:** *UserFollowersManager*



**following\_users:** *UserFollowingManager*

**gpgkeys:** *UserGPGKeyManager*

**identityproviders:** *UserIdentityProviderManager*

**impersonationtokens:** *UserImpersonationTokenManager*

**keys:** *UserKeyManager*

**memberships:** *UserMembershipManager*

**personal\_access\_tokens:** *UserPersonalAccessTokenManager*

**projects:** *UserProjectManager*

**reject**(\*\*kwargs: *Any*) → Dict[str, *Any*] | Response

Reject a user creation request.

#### Parameters

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUserRejectError** – If the user could not be rejected

#### Returns

The new object data (*not* a RESTObject)

#### Return type

Dict[str, *Any*] | Response

**starred\_projects:** *StarredProjectManager*

**status:** *UserStatusManager*

**unban**(\*\*kwargs: *Any*) → Dict[str, *Any*] | Response

Unban the user.

#### Parameters

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUnbanError** – If the user could not be unbanned

#### Returns

Whether the user has been unbanned

#### Return type

Dict[str, *Any*] | Response

**unblock**(\*\*kwargs: *Any*) → bool | None

Unblock the user.

#### Parameters

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUnblockError** – If the user could not be unblocked

**Returns**

Whether the user status has been changed

**Return type**

bool | None

**unfollow**(\*\*kwargs: Any) → Dict[str, Any] | Response

Unfollow the user.

**Parameters**

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUnfollowError** – If the user could not be followed

**Returns**

The new object data (*not* a RESTObject)

**Return type**

Dict[str, Any] | Response

**class** gitlab.v4.objects.**UserActivities**(manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False)

Bases: [RESTObject](#)

**Parameters**

- **manager** (RESTManager)
- **attrs** (Dict[str, Any])
- **created\_from\_list** (bool)
- **lazy** (bool)

**class** gitlab.v4.objects.**UserActivitiesManager**(gl: Gitlab, parent: RESTObject | None = None)

Bases: [ListMixin](#), [RESTManager](#)

**Object Creation****Object update****Parameters**

- **gl** (Gitlab)
- **parent** (RESTObject | None)

**class** gitlab.v4.objects.**UserCustomAttribute**(manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False)

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

**Parameters**

- **manager** (RESTManager)
- **attrs** (Dict[str, Any])
- **created\_from\_list** (bool)

- **lazy** (*bool*)

**class** gitlab.v4.objects.**UserCustomAttributeManager**(*gl*: Gitlab, *parent*: RESTObject | None = None)

Bases: *RetrieveMixin*, *SetMixin*, *DeleteMixin*, *RESTManager*

#### Object Creation

#### Object update

##### Parameters

- **gl** (Gitlab)
- **parent** (RESTObject | None)

**get**(*id*: str | int, *lazy*: bool = False, *\*\*kwargs*: Any) → *UserCustomAttribute*

Retrieve a single object.

##### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated RESTObject.

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

##### Return type

*UserCustomAttribute*

**class** gitlab.v4.objects.**UserEmail**(*manager*: RESTManager, *attrs*: Dict[str, Any], \*, *created\_from\_list*: bool = False, *lazy*: bool = False)

Bases: *ObjectDeleteMixin*, *RESTObject*

##### Parameters

- **manager** (RESTManager)
- **attrs** (Dict[str, Any])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**UserEmailManager**(*gl*: Gitlab, *parent*: RESTObject | None = None)

Bases: *RetrieveMixin*, *CreateMixin*, *DeleteMixin*, *RESTManager*

#### Object Creation

Required attributes for object create:

- email

#### Object update

##### Parameters

- **gl** (Gitlab)

- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *UserEmail*

Retrieve a single object.

#### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. *sudo*)

#### Returns

The generated [RESTObject](#).

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

#### Return type

[UserEmail](#)

```
class gitlab.v4.objects.UserEvent(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: [Event](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict*[*str*, *Any*])
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.UserEventManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [EventManager](#)

#### Object listing filters

- *action*
- *target\_type*
- *before*
- *after*
- *sort*
- *scope*

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

```
class gitlab.v4.objects.UserGPGKey(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[[str](#), [Any](#)])
- **created\_from\_list** ([bool](#))
- **lazy** ([bool](#))

```
class gitlab.v4.objects.UserGPGKeyManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [RetrieveMixin](#), [CreateMixin](#), [DeleteMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **key**

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | [None](#))

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → UserGPGKey
```

Retrieve a single object.

##### Parameters

- **id** ([str](#) | [int](#)) – ID of the object to retrieve
- **lazy** ([bool](#)) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** ([Any](#)) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated [RESTObject](#).

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

##### Return type

[UserGPGKey](#)

```
class gitlab.v4.objects.UserIdentityProviderManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [DeleteMixin](#), [RESTManager](#)

Manager for user identities.

This manager does not actually manage objects but enables functionality for deletion of user identities by provider.

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**class** gitlab.v4.objects.**UserImpersonationToken**(*manager*: [RESTManager](#), *attrs*: *Dict[str, Any]*, \*,  
*created\_from\_list*: *bool* = *False*, *lazy*: *bool* = *False*)

Bases: [ObjectDeleteMixin](#), [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**class** gitlab.v4.objects.**UserImpersonationTokenManager**(*gl*: [Gitlab](#), *parent*: [RESTObject](#) | *None* =  
*None*)

Bases: [NoUpdateMixin](#), [RESTManager](#)

#### Object listing filters

- **state**

#### Object Creation

Required attributes for object create:

- **name**
- **scopes**

Optional attributes for object create:

- **expires\_at**

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | *None*)

**get**(*id*: *str* | *int*, *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [UserImpersonationToken](#)

Retrieve a single object.

##### Parameters

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

##### Returns

The generated [RESTObject](#).

##### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**`UserImpersonationToken`

```
class gitlab.v4.objects.UserKey(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool
                               = False, lazy: bool = False)
```

Bases: `ObjectDeleteMixin`, `RESTObject`

**Parameters**

- **manager** (`RESTManager`)
- **attrs** (`Dict[str, Any]`)
- **created\_from\_list** (`bool`)
- **lazy** (`bool`)

```
class gitlab.v4.objects.UserKeyManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: `RetrieveMixin`, `CreateMixin`, `DeleteMixin`, `RESTManager`

**Object Creation**

Required attributes for object create:

- **title**
- **key**

**Object update****Parameters**

- **gl** (`Gitlab`)
- **parent** (`RESTObject` | `None`)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → UserKey
```

Retrieve a single object.

**Parameters**

- **id** (`str` | `int`) – ID of the object to retrieve
- **lazy** (`bool`) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (`Any`) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated `RESTObject`.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**`UserKey`

```
class gitlab.v4.objects.UserManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: `CRUDMixin`, `RESTManager`

**Object listing filters**

- **active**

- blocked
- username
- extern\_uid
- provider
- external
- search
- custom\_attributes
- status
- two\_factor

### Object Creation

Optional attributes for object create:

- email
- username
- name
- password
- reset\_password
- skype
- linkedin
- twitter
- projects\_limit
- extern\_uid
- provider
- bio
- admin
- can\_create\_group
- website\_url
- skip\_confirmation
- external
- organization
- location
- avatar
- public\_email
- private\_profile
- color\_scheme\_id
- theme\_id



### Object update

Required attributes for object update:

- email
- username
- name

Optional attributes for object update:

- password
- skype
- linkedin
- twitter
- projects\_limit
- extern\_uid
- provider
- bio
- admin
- can\_create\_group
- website\_url
- skip\_reconfirmation
- external
- organization
- location
- avatar
- public\_email
- private\_profile
- color\_scheme\_id
- theme\_id

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) / *None*)

**get**(*id: str | int, lazy: bool = False, \*\*kwargs: Any*) → *User*

Retrieve a single object.

#### Parameters

- **id** (*str* / *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[User](#)

```
class gitlab.v4.objects.UserMembership(manager: RESTManager, attrs: Dict[str, Any], *,
                                       created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

```
class gitlab.v4.objects.UserMembershipManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

**Object listing filters**

- **type**

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → UserMembership
```

Retrieve a single object.

**Parameters**

- **id** (*str* | *int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

[UserMembership](#)

```
class gitlab.v4.objects.UserPersonalAccessToken(manager: RESTManager, attrs: Dict[str, Any], *,
                                                created_from_list: bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (Dict[str, Any])
- **created\_from\_list** (bool)
- **lazy** (bool)

```
class gitlab.v4.objects.UserPersonalAccessTokenManager(gl: Gitlab, parent: RESTObject | None =
                                                         None)
```

Bases: [CreateMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- name
- scopes

Optional attributes for object create:

- expires\_at

#### Object update

##### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
class gitlab.v4.objects.UserProject(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
                                      bool = False, lazy: bool = False)
```

Bases: [RESTObject](#)

#### Parameters

- **manager** ([RESTManager](#))
- **attrs** (Dict[str, Any])
- **created\_from\_list** (bool)
- **lazy** (bool)

```
class gitlab.v4.objects.UserProjectManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [ListMixin](#), [CreateMixin](#), [RESTManager](#)

#### Object listing filters

- archived
- visibility
- order\_by
- sort
- search
- simple

- owned
- membership
- starred
- statistics
- with\_issues\_enabled
- with\_merge\_requests\_enabled
- with\_custom\_attributes
- with\_programming\_language
- wiki\_checksum\_failed
- repository\_checksum\_failed
- min\_access\_level
- id\_after
- id\_before

### Object Creation

Required attributes for object create:

- name

Optional attributes for object create:

- default\_branch
- issues\_enabled
- wall\_enabled
- merge\_requests\_enabled
- wiki\_enabled
- snippets\_enabled
- squash\_option
- public
- visibility
- description
- builds\_enabled
- public\_builds
- import\_url
- only\_allow\_merge\_if\_build\_succeeds

### Object update

#### Parameters

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) / *None*)

**list**(\*\*kwargs: Any) → *RESTObjectList* | List[*RESTObject*]

Retrieve a list of objects.

#### Parameters

- **all** – If True, return all the items, without pagination
- **per\_page** – Number of items to retrieve per request
- **page** – ID of the page to return (starts with page 1)
- **iterator** – If set to True and no pagination option is defined, return a generator instead of a list
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

#### Returns

The list of objects, or a generator if *iterator* is True

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the server cannot perform the request

#### Return type

*RESTObjectList* | List[*RESTObject*]

**class** gitlab.v4.objects.**UserStatus**(manager: *RESTManager*, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False)

Bases: *RESTObject*

#### Parameters

- **manager** (*RESTManager*)
- **attrs** (Dict[str, Any])
- **created\_from\_list** (bool)
- **lazy** (bool)

**class** gitlab.v4.objects.**UserStatusManager**(gl: *Gitlab*, parent: *RESTObject* | None = None)

Bases: *GetWithoutIdMixin*, *RESTManager*

#### Object Creation

#### Object update

#### Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject* | None)

**get**(\*\*kwargs: Any) → *UserStatus*

Retrieve a single object.

#### Parameters

- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

#### Returns

The generated *RESTObject*

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct

- **GitlabGetError** – If the server cannot perform the request

**Return type**

[UserStatus](#)

```
class gitlab.v4.objects.Variable(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list:
    bool = False, lazy: bool = False)
```

Bases: [SaveMixin](#), [ObjectDeleteMixin](#), [RESTObject](#)

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** ([Dict](#)[str, Any])
- **created\_from\_list** (bool)
- **lazy** (bool)

```
class gitlab.v4.objects.VariableManager(gl: Gitlab, parent: RESTObject | None = None)
```

Bases: [CRUDMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- key
- value

Optional attributes for object create:

- protected
- variable\_type
- masked

**Object update**

Required attributes for object update:

- key
- value

Optional attributes for object update:

- protected
- variable\_type
- masked

**Parameters**

- **gl** ([Gitlab](#))
- **parent** ([RESTObject](#) | None)

```
get(id: str | int, lazy: bool = False, **kwargs: Any) → Variable
```

Retrieve a single object.

**Parameters**

- **id** (str | int) – ID of the object to retrieve

- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

Variable

**Module contents****7.6.3 Submodules****7.6.4 gitlab.base module**

**class** gitlab.base.RESTManager(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: object

Base class for CRUD operations on objects.

Derived class must define `_path` and `_obj_cls`.

`_path`: Base URL path on which requests will be sent (e.g. `/projects`) `_obj_cls`: The class of objects that will be created

**Parameters**

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

**gitlab:** *Gitlab*

**property** parent\_attrs: *Dict[str, Any] | None*

**property** path: *str | None*

**class** gitlab.base.RESTObject(*manager: RESTManager, attrs: Dict[str, Any], \*, created\_from\_list: bool = False, lazy: bool = False*)

Bases: object

Represents an object built from server data.

It holds the attributes know from the server, and the updated attributes in another. This allows smart updates, if the object allows it.

You can redefine `_id_attr` in child classes to specify which attribute must be used as the unique ID. None means that the object can be updated without ID in the url.

Likewise, you can define a `_repr_attr` in subclasses to specify which attribute should be added as a human-readable identifier when called in the object's `__repr__()` method.

**Parameters**

- **manager** ([RESTManager](#))
- **attrs** (*Dict[str, Any]*)
- **created\_from\_list** (*bool*)
- **lazy** (*bool*)

**asdict**(\*, *with\_parent\_attrs: bool = False*) → *Dict[str, Any]*

**Parameters**

**with\_parent\_attrs** (*bool*)

**Return type**

*Dict[str, Any]*

**property attributes:** *Dict[str, Any]*

**property encoded\_id:** *int | str | None*

Ensure that the ID is url-encoded so that it can be safely used in a URL path

**get\_id**() → *int | str | None*

Returns the id of the resource.

**Return type**

*int | str | None*

**manager:** [RESTManager](#)

**pformat**() → *str*

**Return type**

*str*

**pprint**() → *None*

**Return type**

*None*

**to\_json**(\*, *with\_parent\_attrs: bool = False, \*\*kwargs: Any*) → *str*

**Parameters**

- **with\_parent\_attrs** (*bool*)
- **kwargs** (*Any*)

**Return type**

*str*

**class** `gitlab.base.RESTObjectList`(*manager: RESTManager, obj\_cls: Type[RESTObject], \_list: GitlabList*)

Bases: `object`

Generator object representing a list of `RESTObject`'s.

This generator uses the Gitlab pagination system to fetch new data when required.

Note: you should not instantiate such objects, they are returned by calls to `RESTManager.list()`

**Parameters**

- **manager** ([RESTManager](#)) – Manager to attach to the created objects
- **obj\_cls** (*Type[RESTObject]*) – Type of objects to create from the json data
- **\_list** ([GitlabList](#)) – A `GitlabList` object



**property current\_page:** `int`

The current page number.

**next()** → *RESTObject*

**Return type**

*RESTObject*

**property next\_page:** `int | None`

The next page number.

If None, the current page is the last.

**property per\_page:** `int | None`

The number of items per page.

**property prev\_page:** `int | None`

The previous page number.

If None, the current page is the first.

**property total:** `int | None`

The total number of items.

**property total\_pages:** `int | None`

The total number of pages.

### 7.6.5 gitlab.cli module

**class** `gitlab.cli.VerticalHelpFormatter`(*prog*, *indent\_increment=2*, *max\_help\_position=24*, *width=None*)

Bases: `HelpFormatter`

**format\_help()** → `str`

**Return type**

`str`

`gitlab.cli.cls_to_gitlab_resource`(*cls*: *RESTObject*) → `str`

**Parameters**

**cls** (*RESTObject*)

**Return type**

`str`

`gitlab.cli.die`(*msg*: *str*, *e*: *Exception* | *None* = *None*) → `None`

**Parameters**

- **msg** (*str*)
- **e** (*Exception* | *None*)

**Return type**

`None`

`gitlab.cli.docs`() → `ArgumentParser`

Provide a statically generated parser for sphinx only, so we don't need to provide dummy gitlab config for readthedocs.

**Return type***ArgumentParser*`gitlab.cli.gitlab_resource_to_cls(gitlab_resource: str, namespace: ModuleType) → Type[RESTObject]`**Parameters**

- **gitlab\_resource** (*str*)
- **namespace** (*ModuleType*)

**Return type***Type[RESTObject]*`gitlab.cli.main() → None`**Return type***None*`gitlab.cli.register_custom_action(cls_names: str | Tuple[str, ...], mandatory: Tuple[str, ...] = (), optional: Tuple[str, ...] = (), custom_action: str | None = None) → Callable[[__F], __F]`**Parameters**

- **cls\_names** (*str | Tuple[str, ...]*)
- **mandatory** (*Tuple[str, ...]*)
- **optional** (*Tuple[str, ...]*)
- **custom\_action** (*str | None*)

**Return type***Callable[[\_\_F], \_\_F]*

## 7.6.6 gitlab.config module

**exception** `gitlab.config.ConfigError`Bases: *Exception***exception** `gitlab.config.GitlabConfigHelperError`Bases: *ConfigError***exception** `gitlab.config.GitlabConfigMissingError`Bases: *ConfigError***class** `gitlab.config.GitlabConfigParser(gitlab_id: str | None = None, config_files: List[str] | None = None)`Bases: *object***Parameters**

- **gitlab\_id** (*str | None*)
- **config\_files** (*List[str] | None*)

**exception** `gitlab.config.GitlabDataError`Bases: *ConfigError***exception** `gitlab.config.GitlabIDError`Bases: *ConfigError*

### 7.6.7 gitlab.const module

```
class gitlab.const.AccessLevel(value, names=None, *, module=None, qualname=None, type=None, start=1, boundary=None)
```

Bases: IntEnum

**ADMIN:** int = 60

**DEVELOPER:** int = 30

**GUEST:** int = 10

**MAINTAINER:** int = 40

**MINIMAL\_ACCESS:** int = 5

**NO\_ACCESS:** int = 0

**OWNER:** int = 50

**REPORTER:** int = 20

```
class gitlab.const.NotificationLevel(value, names=None, *, module=None, qualname=None, type=None, start=1, boundary=None)
```

Bases: GitlabEnum

**CUSTOM:** str = 'custom'

**DISABLED:** str = 'disabled'

**GLOBAL:** str = 'global'

**MENTION:** str = 'mention'

**PARTICIPATING:** str = 'participating'

**WATCH:** str = 'watch'

```
class gitlab.const.SearchScope(value, names=None, *, module=None, qualname=None, type=None, start=1, boundary=None)
```

Bases: GitlabEnum

**BLOBS:** str = 'blobs'

**COMMITTS:** str = 'commits'

**GLOBAL\_SNIPPET\_TITLES:** str = 'snippet\_titles'

**ISSUES:** str = 'issues'

**MERGE\_REQUESTS:** str = 'merge\_requests'

**MILESTONES:** str = 'milestones'

**PROJECTS:** str = 'projects'

**PROJECT\_NOTES:** str = 'notes'

**USERS:** str = 'users'

```
WIKI_BLOBS: str = 'wiki_blobs'
```

```
class gitlab.const.Visibility(value, names=None, *, module=None, qualname=None, type=None, start=1,
                             boundary=None)
```

Bases: `GitlabEnum`

```
INTERNAL: str = 'internal'
```

```
PRIVATE: str = 'private'
```

```
PUBLIC: str = 'public'
```

## 7.6.8 gitlab.exceptions module

```
exception gitlab.exceptions.GitlabActivateError(error_message: str | bytes = "", response_code: int |
                                                None = None, response_body: bytes | None = None)
```

Bases: `GitlabOperationError`

### Parameters

- **error\_message** (`str` | `bytes`)
- **response\_code** (`int` | `None`)
- **response\_body** (`bytes` | `None`)

### Return type

`None`

```
exception gitlab.exceptions.GitlabAttachFileError(error_message: str | bytes = "", response_code: int |
                                                  None = None, response_body: bytes | None =
                                                  None)
```

Bases: `GitlabOperationError`

### Parameters

- **error\_message** (`str` | `bytes`)
- **response\_code** (`int` | `None`)
- **response\_body** (`bytes` | `None`)

### Return type

`None`

```
exception gitlab.exceptions.GitlabAuthenticationError(error_message: str | bytes = "",
                                                        response_code: int | None = None,
                                                        response_body: bytes | None = None)
```

Bases: `GitlabError`

### Parameters

- **error\_message** (`str` | `bytes`)
- **response\_code** (`int` | `None`)
- **response\_body** (`bytes` | `None`)

### Return type

`None`

**exception** gitlab.exceptions.GitlabBanError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabBlockError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabBuildCancelError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabCancelError](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabBuildEraseError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabRetryError](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.exceptions.GitlabBuildPlayError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabRetryError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.exceptions.GitlabBuildRetryError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabRetryError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.exceptions.GitlabCancelError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.exceptions.GitlabCherryPickError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.exceptions.GitlabCiLintError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.exceptions.GitlabConnectionError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.exceptions.GitlabCreateError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.exceptions.GitlabDeactivateError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabDeleteError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabDeploymentApprovalError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [Exception](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabFollowError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabGetError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)



**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabGroupTransferError(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabHeadError(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabHousekeepingError(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabHttpError(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [GitlabError](#)

**Parameters**

- **error\_message** (*str* | *bytes*)

- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** gitlab.exceptions.**GitlabImportError**(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** gitlab.exceptions.**GitlabInvitationError**(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** gitlab.exceptions.**GitlabJobCancelError**(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [\*GitlabCancelError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** gitlab.exceptions.**GitlabJobEraseError**(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [\*GitlabRetryError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabJobPlayError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabRetryError](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabJobRetryError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabRetryError](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabLicenseError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabListError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabMRApprovalError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabMRClosedError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabMRForbiddenError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabMROnBuildSuccessError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.exceptions.GitlabMRRebaseError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

#### Parameters

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

#### Return type

None

**exception** `gitlab.exceptions.GitlabMRResetApprovalError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

#### Parameters

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

#### Return type

None

**exception** `gitlab.exceptions.GitlabMarkdownError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

#### Parameters

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

#### Return type

None

**exception** `gitlab.exceptions.GitlabOperationError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabError`](#)

#### Parameters

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

#### Return type

None

**exception** `gitlab.exceptions.GitlabOwnershipError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.exceptions.GitlabParsingError`(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [\*GitlabError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.exceptions.GitlabPipelineCancelError`(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [\*GitlabCancelError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.exceptions.GitlabPipelinePlayError`(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [\*GitlabRetryError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.exceptions.GitlabPipelineRetryError`(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [\*GitlabRetryError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** `gitlab.exceptions.GitlabProjectDeployKeyError`(*error\_message: str* | *bytes* = "",  
*response\_code: int* | *None* = *None*,  
*response\_body: bytes* | *None* = *None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** `gitlab.exceptions.GitlabPromoteError`(*error\_message: str* | *bytes* = "", *response\_code: int* |  
*None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** `gitlab.exceptions.GitlabProtectError`(*error\_message: str* | *bytes* = "", *response\_code: int* |  
*None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** `gitlab.exceptions.GitlabRenderError`(*error\_message: str* | *bytes* = "", *response\_code: int* |  
*None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** gitlab.exceptions.**GitlabRepairError**(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.**GitlabRestoreError**(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.**GitlabRetryError**(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.**GitlabRevertError**(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None



**exception** `gitlab.exceptions.GitlabRotateError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.exceptions.GitlabSearchError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.exceptions.GitlabSetError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.exceptions.GitlabStopError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** `gitlab.exceptions.GitlabSubscribeError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.exceptions.GitlabTimeTrackingError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.exceptions.GitlabTodoError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.exceptions.GitlabTopicMergeError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

*None*

**exception** `gitlab.exceptions.GitlabTransferProjectError`(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*str* | *bytes*)

- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabUnbanError(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabUnblockError(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabUnfollowError(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabUnsubscribeError(*error\_message: str* | *bytes* = "", *response\_code: int* | *None* = *None*, *response\_body: bytes* | *None* = *None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*str* | *bytes*)
- **response\_code** (*int* | *None*)
- **response\_body** (*bytes* | *None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabUpdateError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabUploadError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabUserApproveError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.GitlabUserRejectError(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)**Parameters**

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

**Return type**

None

**exception** gitlab.exceptions.**GitlabVerifyError**(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

#### Return type

None

**exception** gitlab.exceptions.**RedirectError**(*error\_message: str | bytes = "", response\_code: int | None = None, response\_body: bytes | None = None*)

Bases: [GitlabError](#)

#### Parameters

- **error\_message** (*str | bytes*)
- **response\_code** (*int | None*)
- **response\_body** (*bytes | None*)

#### Return type

None

## 7.6.9 gitlab.mixins module

**class** gitlab.mixins.**AccessRequestMixin**

Bases: object

**approve**(*access\_level: int = 30, \*\*kwargs: Any*) → None

Approve an access request.

#### Parameters

- **access\_level** (*int*) – The access level for the user
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUpdateError** – If the server fails to perform the request

#### Return type

None

**manager:** [RESTManager](#)

**class** gitlab.mixins.**BadgeRenderMixin**

Bases: object

**render**(*link\_url: str, image\_url: str, \*\*kwargs: Any*) → Dict[str, Any]

Preview link\_url and image\_url after interpolation.

#### Parameters

- **link\_url** (*str*) – URL of the badge link
- **image\_url** (*str*) – URL of the badge image
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabRenderError** – If the rendering failed

**Returns**

The rendering properties

**Return type**

*Dict[str, Any]*

**class** gitlab.mixins.CRUDMixin

Bases: *GetMixin, ListMixin, CreateMixin, UpdateMixin, DeleteMixin*

**gitlab:** *Gitlab*

**class** gitlab.mixins.CreateMixin

Bases: object

**create**(*data: Dict[str, Any] | None = None, \*\*kwargs: Any*) → *RESTObject*

Create a new object.

**Parameters**

- **data** (*Dict[str, Any] | None*) – parameters to send to the server to create the resource
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

A new instance of the managed object class built with the data sent by the server

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

**Return type**

*RESTObject*

**gitlab:** *Gitlab*

**class** gitlab.mixins.DeleteMixin

Bases: object

**delete**(*id: int | str | None = None, \*\*kwargs: Any*) → None

Delete an object on the server.

**Parameters**

- **id** (*int | str | None*) – ID of the object to delete
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the server cannot perform the request

**Return type**

None

**gitlab:** *Gitlab***class** gitlab.mixins.DownloadMixin

Bases: object

**download**(*streamed: bool = False, action: Callable[[bytes], None] | None = None, chunk\_size: int = 1024, \*, iterator: bool = False, \*\*kwargs: Any*) → bytes | Iterator[Any] | None

Download the archive of a resource export.

**Parameters**

- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk\_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable[[bytes], None] | None*) – Callable responsible of dealing with chunk of data
- **chunk\_size** (*int*) – Size of each chunk
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server failed to perform the request

**Returns**

The blob content if streamed is False, None otherwise

**Return type**

bytes | Iterator[Any] | None

**manager:** *RESTManager***class** gitlab.mixins.GetMixin

Bases: HeadMixin, object

**get**(*id: str | int, lazy: bool = False, \*\*kwargs: Any*) → *RESTObject*

Retrieve a single object.

**Parameters**

- **id** (*str | int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Returns**

The generated RESTObject.

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct

- **GitlabGetError** – If the server cannot perform the request

**Return type**

`RESTObject`

**gitlab:** `Gitlab`

**class** `gitlab.mixins.GetWithoutIdMixin`

Bases: `HeadMixin`, `object`

**get**(*\*\*kwargs*: `Any`) → `RESTObject`

Retrieve a single object.

**Parameters**

*\*\*kwargs* (`Any`) – Extra options to send to the server (e.g. `sudo`)

**Returns**

The generated `RESTObject`

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**

`RESTObject`

**gitlab:** `Gitlab`

**class** `gitlab.mixins.ListMixin`

Bases: `HeadMixin`, `object`

**gitlab:** `Gitlab`

**list**(*\*\*kwargs*: `Any`) → `RESTObjectList` | `List[RESTObject]`

Retrieve a list of objects.

**Parameters**

- **all** – If `True`, return all the items, without pagination
- **per\_page** – Number of items to retrieve per request
- **page** – ID of the page to return (starts with page 1)
- **iterator** – If set to `True` and no pagination option is defined, return a generator instead of a list
- *\*\*kwargs* (`Any`) – Extra options to send to the server (e.g. `sudo`)

**Returns**

The list of objects, or a generator if *iterator* is `True`

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the server cannot perform the request

**Return type**

`RESTObjectList` | `List[RESTObject]`

**class** `gitlab.mixins.NoUpdateMixin`

Bases: `GetMixin`, `ListMixin`, `CreateMixin`, `DeleteMixin`



**gitlab:** *Gitlab*

**class** gitlab.mixins.ObjectDeleteMixin

Bases: object

Mixin for RESTObject's that can be deleted.

**delete**(\*\*kwargs: *Any*) → None

Delete the object from the server.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the server cannot perform the request

**Return type**

None

**manager:** *RESTManager*

**class** gitlab.mixins.ParticipantsMixin

Bases: object

**manager:** *RESTManager*

**participants**(\*\*kwargs: *Any*) → Dict[str, *Any*]

List the participants.

**Parameters**

- **all** – If True, return all the items, without pagination
- **per\_page** – Number of items to retrieve per request
- **page** – ID of the page to return (starts with page 1)
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the list could not be retrieved

**Returns**

The list of participants

**Return type**

Dict[str, *Any*]

**class** gitlab.mixins.RefreshMixin

Bases: object

**manager:** *RESTManager*

**refresh**(\*\*kwargs: *Any*) → None

Refresh a single object from server.

**Parameters**

**\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Return type**

None

Returns None (updates the object)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Parameters****kwargs** (*Any*)**Return type**

None

**class** gitlab.mixins.RetrieveMixinBases: *ListMixin*, *GetMixin***gitlab:** *Gitlab***class** gitlab.mixins.SaveMixin

Bases: object

Mixin for RESTObject's that can be updated.

**manager:** *RESTManager***save**(\*\*kwargs: *Any*) → Dict[str, *Any*] | None

Save the changes made to the object to the server.

The object is updated to match what the server returns.

**Parameters****\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)**Returns**The new object data (*not* a RESTObject)**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUpdateError** – If the server cannot perform the request

**Return type**Dict[str, *Any*] | None**class** gitlab.mixins.SetMixin

Bases: object

**gitlab:** *Gitlab***set**(key: str, value: str, \*\*kwargs: *Any*) → *RESTObject*

Create or update the object.

**Parameters**

- **key** (*str*) – The key of the object to create/update
- **value** (*str*) – The value to set for the object
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabSetError** – If an error occurred

**Returns**

The created/updated attribute

**Return type**

`RESTObject`

**class** gitlab.mixins.SubscribableMixin

Bases: object

**manager:** `RESTManager`

**subscribe**(\*\*kwargs: `Any`) → None

Subscribe to the object notifications.

**Parameters**

**\*\*kwargs** (`Any`) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabSubscribeError** – If the subscription cannot be done

**Return type**

None

**unsubscribe**(\*\*kwargs: `Any`) → None

Unsubscribe from the object notifications.

**Parameters**

**\*\*kwargs** (`Any`) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUnsubscribeError** – If the unsubscription cannot be done

**Return type**

None

**class** gitlab.mixins.TimeTrackingMixin

Bases: object

**add\_spent\_time**(duration: `str`, \*\*kwargs: `Any`) → Dict[`str`, `Any`]

Add time spent working on the object.

**Parameters**

- **duration** (`str`) – Duration in human format (e.g. 3h30)
- **\*\*kwargs** (`Any`) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabTimeTrackingError** – If the time tracking update cannot be done

**Return type**

`Dict[str, Any]`

**manager:** [\*RESTManager\*](#)

**reset\_spent\_time**(\*\*kwargs: [\*Any\*](#)) → Dict[str, [\*Any\*](#)]

Resets the time spent working on the object.

**Parameters**

**\*\*kwargs** ([\*Any\*](#)) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabTimeTrackingError** – If the time tracking update cannot be done

**Return type**

Dict[str, [\*Any\*](#)]

**reset\_time\_estimate**(\*\*kwargs: [\*Any\*](#)) → Dict[str, [\*Any\*](#)]

Resets estimated time for the object to 0 seconds.

**Parameters**

**\*\*kwargs** ([\*Any\*](#)) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabTimeTrackingError** – If the time tracking update cannot be done

**Return type**

Dict[str, [\*Any\*](#)]

**time\_estimate**(duration: str, \*\*kwargs: [\*Any\*](#)) → Dict[str, [\*Any\*](#)]

Set an estimated time of work for the object.

**Parameters**

- **duration** (str) – Duration in human format (e.g. 3h30)
- **\*\*kwargs** ([\*Any\*](#)) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabTimeTrackingError** – If the time tracking update cannot be done

**Return type**

Dict[str, [\*Any\*](#)]

**time\_stats**(\*\*kwargs: [\*Any\*](#)) → Dict[str, [\*Any\*](#)]

Get time stats for the object.

**Parameters**

**\*\*kwargs** ([\*Any\*](#)) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabTimeTrackingError** – If the time tracking update cannot be done

**Return type**

Dict[str, [\*Any\*](#)]

```
class gitlab.mixins.TodoMixin
```

Bases: object

**manager:** [\*RESTManager\*](#)

**todo**(\*\*kwargs: [\*Any\*](#)) → None

Create a todo associated to the object.

**Parameters**

**\*\*kwargs** ([\*Any\*](#)) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabTodoError** – If the todo cannot be set

**Return type**

None

```
class gitlab.mixins.UpdateMixin
```

Bases: object

**gitlab:** [\*Gitlab\*](#)

**update**(id: int | str | None = None, new\_data: Dict[str, [\*Any\*](#)] | None = None, \*\*kwargs: [\*Any\*](#)) → Dict[str, [\*Any\*](#)]

Update an object on the server.

**Parameters**

- **id** (int | str | None) – ID of the object to update (can be None if not required)
- **new\_data** (Dict[str, [\*Any\*](#)] | None) – the update data for the object
- **\*\*kwargs** ([\*Any\*](#)) – Extra options to send to the server (e.g. sudo)

**Returns**

The new object data (*not* a RESTObject)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUpdateError** – If the server cannot perform the request

**Return type**

Dict[str, [\*Any\*](#)]

```
class gitlab.mixins.UserAgentDetailMixin
```

Bases: object

**manager:** [\*RESTManager\*](#)

**user\_agent\_detail**(\*\*kwargs: [\*Any\*](#)) → Dict[str, [\*Any\*](#)]

Get the user agent detail.

**Parameters**

**\*\*kwargs** ([\*Any\*](#)) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

**Return type**  
*Dict[str, Any]*

### 7.6.10 gitlab.utils module

**class** gitlab.utils.**EncodedId**(value: str | int | [EncodedId](#))

Bases: str

A custom *str* class that will return the URL-encoded value of the string.

- Using it recursively will only url-encode the value once.
- Can accept either *str* or *int* as input value.
- Can be used in an f-string and output the URL-encoded string.

Reference to documentation on why this is necessary.

See:

<https://docs.gitlab.com/ee/api/index.html#namespaced-path-encoding>  
<https://docs.gitlab.com/ee/api/index.html#path-parameters>

**Parameters**  
**value** (str | int | [EncodedId](#))

**Return type**  
[EncodedId](#)

**class** gitlab.utils.**MaskingFormatter**(fmt: str | None = '%(levelname)s:%(name)s:%(message)s', datefmt: str | None = None, style: [Literal](#)['%', '{', '\$'] = '%', validate: bool = True, masked: str | None = None)

Bases: [Formatter](#)

A logging formatter that can mask credentials

**Parameters**

- **fmt** (str | None)
- **datefmt** (str | None)
- **style** ([Literal](#)['%', '{', '\$'])
- **validate** (bool)
- **masked** (str | None)

**format**(record: [LogRecord](#)) → str

Format the specified record as text.

The record's attribute dictionary is used as the operand to a string formatting operation which yields the returned string. Before formatting the dictionary, a couple of preparatory steps are carried out. The message attribute of the record is computed using [LogRecord.getMessage\(\)](#). If the formatting string uses the time (as determined by a call to [usesTime\(\)](#)), [formatTime\(\)](#) is called to format the event time. If there is exception information, it is formatted using [formatException\(\)](#) and appended to the message.

**Parameters**  
**record** ([LogRecord](#))

**Return type**

str

```
gitlab.utils.copy_dict(*, src: Dict[str, Any], dest: Dict[str, Any]) → None
```

**Parameters**

- **src** (*Dict[str, Any]*)
- **dest** (*Dict[str, Any]*)

**Return type**

None

```
gitlab.utils.get_content_type(content_type: str | None) → str
```

**Parameters**

**content\_type** (*str | None*)

**Return type**

str

```
gitlab.utils.remove_none_from_dict(data: Dict[str, Any]) → Dict[str, Any]
```

**Parameters**

**data** (*Dict[str, Any]*)

**Return type**

*Dict[str, Any]*

```
gitlab.utils.response_content(response: Response, streamed: bool, action: Callable[[bytes], None] | None,
                               chunk_size: int, *, iterator: bool) → bytes | Iterator[Any] | None
```

**Parameters**

- **response** (*Response*)
- **streamed** (*bool*)
- **action** (*Callable[[bytes], None] | None*)
- **chunk\_size** (*int*)
- **iterator** (*bool*)

**Return type**

*bytes | Iterator[Any] | None*

```
gitlab.utils.warn(message: str, *, category: Type[Warning] | None = None, source: Any | None = None) → None
```

This `warnings.warn` wrapper function attempts to show the location causing the warning in the user code that called the library.

It does this by walking up the stack trace to find the first frame located outside the `gitlab/` directory. This is helpful to users as it shows them their code that is causing the warning.

**Parameters**

- **message** (*str*)
- **category** (*Type[Warning] | None*)
- **source** (*Any | None*)

**Return type**

None

## 7.7 CLI reference (gitlab command)

**Warning:** The following is a complete, auto-generated list of subcommands available via the **gitlab** command-line tool. Some of the actions may currently not work as expected or lack functionality available via the API.

Please see the existing [list of CLI related issues](#), or open a new one if it is not already listed there.

### 7.7.1 gitlab

GitLab API Command Line Interface

```
usage: gitlab [-h] [--version] [-v] [-d] [-c CONFIG_FILE] [-g GITLAB]
             [-o {json,legacy,yaml}] [-f FIELDS] [--server-url SERVER_URL]
             [--ssl-verify SSL_VERIFY | --no-ssl-verify] [--timeout TIMEOUT]
             [--api-version API_VERSION] [--per-page PER_PAGE]
             [--pagination PAGINATION] [--order-by ORDER_BY]
             [--user-agent USER_AGENT]
             [--private-token PRIVATE_TOKEN | --oauth-token OAUTH_TOKEN | --job-token_
↪JOB_TOKEN]
             [--skip-login]
application
application-appearance
application-settings
application-statistics
audit-event
broadcast-message
bulk-import
bulk-import-all-entity
bulk-import-entity
ci-lint
current-user
current-user-email
current-user-gpg-key
current-user-key
current-user-runner
current-user-status
deploy-key
deploy-token
dockerfile
event
feature
generic-package
geo-node
gitignore
gitlabciyaml
group
group-access-request
group-access-token
group-audit-event
group-badge
group-billable-member
```

(continues on next page)



(continued from previous page)

```
group-billable-member-membership
group-board
group-board-list
group-cluster
group-custom-attribute
group-deploy-token
group-descendant-group
group-epic
group-epic-award-emoji
group-epic-discussion-note
group-epic-issue
group-epic-note
group-epic-note-award-emoji
group-epic-resource-label-event
group-export
group-hook
group-import
group-invitation
group-issue
group-issues-statistics
group-iteration
group-ldap-group-link
group-label
group-member
group-member-all
group-merge-request
group-milestone
group-notification-settings
group-package
group-project
group-push-rules
group-runner
group-saml-group-link
group-subgroup
group-variable
group-wiki
hook
issue
issues-statistics
key
ldap-group
license
merge-request
namespace
notification-settings
pages-domain
personal-access-token
project
project-access-request
project-access-token
project-additional-statistics
project-approval
```

(continues on next page)

(continued from previous page)

```
project-approval-rule
project-artifact
project-audit-event
project-badge
project-board
project-board-list
project-branch
project-ci-lint
project-cluster
project-commit
project-commit-comment
project-commit-discussion
project-commit-discussion-note
project-commit-status
project-custom-attribute
project-deploy-token
project-deployment
project-deployment-merge-request
project-environment
project-event
project-export
project-file
project-fork
project-hook
project-import
project-integration
project-invitation
project-issue
project-issue-award-emoji
project-issue-discussion
project-issue-discussion-note
project-issue-link
project-issue-note
project-issue-note-award-emoji
project-issue-resource-iteration-event
project-issue-resource-label-event
project-issue-resource-milestone-event
project-issue-resource-state-event
project-issue-resource-weight-event
project-issues-statistics
project-job
project-job-token-scope
project-key
project-label
project-member
project-member-all
project-merge-request
project-merge-request-approval
project-merge-request-approval-rule
project-merge-request-approval-state
project-merge-request-award-emoji
project-merge-request-diff
```

(continues on next page)

(continued from previous page)

```
project-merge-request-discussion
project-merge-request-discussion-note
project-merge-request-draft-note
project-merge-request-note
project-merge-request-note-award-emoji
project-merge-request-pipeline
project-merge-request-resource-label-event
project-merge-request-resource-milestone-event
project-merge-request-resource-state-event
project-merge-request-reviewer-detail
project-merge-train
project-milestone
project-note
project-notification-settings
project-package
project-package-file
project-package-pipeline
project-pages-domain
project-pipeline
project-pipeline-bridge
project-pipeline-job
project-pipeline-schedule
project-pipeline-schedule-pipeline
project-pipeline-schedule-variable
project-pipeline-test-report
project-pipeline-test-report-summary
project-pipeline-variable
project-protected-branch
project-protected-environment
project-protected-tag
project-push-rules
project-registry-repository
project-registry-tag
project-release
project-release-link
project-remote-mirror
project-resource-group
project-resource-group-upcoming-job
project-runner
project-secure-file
project-service
project-snippet
project-snippet-award-emoji
project-snippet-discussion
project-snippet-discussion-note
project-snippet-note
project-snippet-note-award-emoji
project-storage
project-tag
project-trigger
project-user
project-variable
```

(continues on next page)

(continued from previous page)

```

project-wiki
registry-repository
runner
runner-all
runner-job
shared-project
snippet
starred-project
todo
topic
user
user-activities
user-custom-attribute
user-email
user-event
user-gpg-key
user-impersonation-token
user-key
user-membership
user-personal-access-token
user-project
user-status
variable
...

```

**-h, --help**

show this help message and exit

**--version**

Display the version.

**-v, --verbose, --fancy**

Verbose mode (legacy format only) [env var: GITLAB\_VERBOSE]

**-d, --debug**

Debug mode (display HTTP requests) [env var: GITLAB\_DEBUG]

**-c <config\_file>, --config-file <config\_file>**

Configuration file to use. Can be used multiple times. [env var: PYTHON\_GITLAB\_CFG]

**-g <gitlab>, --gitlab <gitlab>**

Which configuration section should be used. If not defined, the default selection will be used.

**-o {json,legacy,yaml}, --output {json,legacy,yaml}**

Output format (v4 only): json|legacy|yaml

**-f <fields>, --fields <fields>**

Fields to display in the output (comma separated). Not used with legacy output

**--server-url <server\_url>**

GitLab server URL [env var: GITLAB\_URL]

**--ssl-verify <ssl\_verify>**

Path to a CA\_BUNDLE file or directory with certificates of trusted CAs. [env var: GITLAB\_SSL\_VERIFY]

**--no-ssl-verify**

Disable SSL verification

**--timeout** <timeout>

Timeout to use for requests to the GitLab server. [env var: GITLAB\_TIMEOUT]

**--api-version** <api\_version>

GitLab API version [env var: GITLAB\_API\_VERSION]

**--per-page** <per\_page>

Number of entries to return per page in the response. [env var: GITLAB\_PER\_PAGE]

**--pagination** <pagination>

Whether to use keyset or offset pagination [env var: GITLAB\_PAGINATION]

**--order-by** <order\_by>

Set order\_by globally [env var: GITLAB\_ORDER\_BY]

**--user-agent** <user\_agent>

The user agent to send to GitLab with the HTTP request. [env var: GITLAB\_USER\_AGENT]

**--private-token** <private\_token>

GitLab private access token [env var: GITLAB\_PRIVATE\_TOKEN]

**--oauth-token** <oauth\_token>

GitLab OAuth token [env var: GITLAB\_OAUTH\_TOKEN]

**--job-token** <job\_token>

GitLab CI job token [env var: CI\_JOB\_TOKEN]

**--skip-login**

Skip initial authenticated API call to the current user endpoint. This may be useful when invoking the CLI in scripts. [env var: GITLAB\_SKIP\_LOGIN]

**gitlab application**

```
usage: gitlab application [-h] {list,create,delete} ...
```

**-h, --help**

show this help message and exit

**gitlab application create**

```
usage: gitlab application create [-h] [--sudo SUDO] --name NAME --redirect-uri
                                REDIRECT_URI --scopes SCOPES
                                [--confidential CONFIDENTIAL]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--name** <name>

**--redirect-uri** <redirect\_uri>  
**--scopes** <scopes>  
**--confidential** <confidential>

### gitlab application delete

```
usage: gitlab application delete [-h] [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--id** <id>

### gitlab application list

```
usage: gitlab application list [-h] [--sudo SUDO] [--page PAGE]
                               [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--page** <page>  
**--per-page** <per\_page>  
**--get-all**  
Return all items from the server, without pagination.

### gitlab application-appearance

```
usage: gitlab application-appearance [-h] {get,update} ...
```

**-h, --help**  
show this help message and exit

### gitlab application-appearance get

```
usage: gitlab application-appearance get [-h] [--sudo SUDO]
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>

## gitlab application-appearance update

```
usage: gitlab application-appearance update [-h] [--sudo SUDO] [--title TITLE]
                                           [--description DESCRIPTION]
                                           [--logo LOGO]
                                           [--header-logo HEADER_LOGO]
                                           [--favicon FAVICON]
                                           [--new-project-guidelines NEW_PROJECT_
↳GUIDELINES]
                                           [--header-message HEADER_MESSAGE]
                                           [--footer-message FOOTER_MESSAGE]
                                           [--message-background-color MESSAGE_
↳BACKGROUND_COLOR]
                                           [--message-font-color MESSAGE_FONT_COLOR]
                                           [--email-header-and-footer-enabled EMAIL_
↳HEADER_AND_FOOTER_ENABLED]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--title** <title>

**--description** <description>

**--logo** <logo>

**--header-logo** <header\_logo>

**--favicon** <favicon>

**--new-project-guidelines** <new\_project\_guidelines>

**--header-message** <header\_message>

**--footer-message** <footer\_message>

**--message-background-color** <message\_background\_color>

**--message-font-color** <message\_font\_color>

**--email-header-and-footer-enabled** <email\_header\_and\_footer\_enabled>

## gitlab application-settings

```
usage: gitlab application-settings [-h] {get,update} ...
```

**-h, --help**

show this help message and exit

## gitlab application-settings get

```
usage: gitlab application-settings get [-h] [--sudo SUDO]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

## gitlab application-settings update

```
usage: gitlab application-settings update [-h] [--sudo SUDO] [--id ID]
                                         [--default-projects-limit DEFAULT_PROJECTS_
↳LIMIT]
                                         [--signup-enabled SIGNUP_ENABLED]
                                         [--password-authentication-enabled-for-web-
↳PASSWORD_AUTHENTICATION_ENABLED_FOR_WEB]
                                         [--gravatar-enabled GRAVATAR_ENABLED]
                                         [--sign-in-text SIGN_IN_TEXT]
                                         [--created-at CREATED_AT]
                                         [--updated-at UPDATED_AT]
                                         [--home-page-url HOME_PAGE_URL]
                                         [--default-branch-protection DEFAULT_BRANCH_
↳PROTECTION]
                                         [--restricted-visibility-levels RESTRICTED_
↳VISIBILITY_LEVELS]
                                         [--max-attachment-size MAX_ATTACHMENT_SIZE]
                                         [--session-expire-delay SESSION_EXPIRE_DELAY]
                                         [--default-project-visibility DEFAULT_PROJECT_
↳VISIBILITY]
                                         [--default-snippet-visibility DEFAULT_SNIPPET_
↳VISIBILITY]
                                         [--default-group-visibility DEFAULT_GROUP_
↳VISIBILITY]
                                         [--outbound-local-requests-whitelist OUTBOUND_
↳LOCAL_REQUESTS_WHITELIST]
                                         [--disabled-oauth-sign-in-sources DISABLED_
↳OAUTH_SIGN_IN_SOURCES]
                                         [--domain-whitelist DOMAIN_WHITELIST]
                                         [--domain-blacklist-enabled DOMAIN_BLACKLIST_
↳ENABLED]
                                         [--domain-blacklist DOMAIN_BLACKLIST]
                                         [--domain-allowlist DOMAIN_ALLOWLIST]
                                         [--domain-denylist-enabled DOMAIN_DENYLIST_
↳ENABLED]
                                         [--domain-denylist DOMAIN_DENYLIST]
                                         [--external-authorization-service-enabled-
↳EXTERNAL_AUTHORIZATION_SERVICE_ENABLED]
                                         [--external-authorization-service-url EXTERNAL_
↳AUTHORIZATION_SERVICE_URL]
                                         [--external-authorization-service-default-
↳label EXTERNAL_AUTHORIZATION_SERVICE_DEFAULT_LABEL]
```

(continues on next page)



(continued from previous page)

```

↪EXTERNAL_AUTHORIZATION_SERVICE_TIMEOUT] [--external-authorization-service-timeout↵
[--import-sources IMPORT_SOURCES]
↪APPLICATIONS] [--user-oauth-applications USER_OAUTH_

[--after-sign-out-path AFTER_SIGN_OUT_PATH]
↪CONTAINER_REGISTRY_TOKEN_EXPIRE_DELAY] [--container-registry-token-expire-delay↵

[--repository-storages REPOSITORY_STORAGES]
[--plantuml-enabled PLANTUML_ENABLED]
[--plantuml-url PLANTUML_URL]
↪SESSION_TIME] [--terminal-max-session-time TERMINAL_MAX_

[--polling-interval-multiplier POLLING_
↪INTERVAL_MULTIPLIER]

[--rsa-key-restriction RSA_KEY_RESTRICTION]
[--dsa-key-restriction DSA_KEY_RESTRICTION]
[--ecdsa-key-restriction ECDSA_KEY_RESTRICTION]
↪RESTRICTION] [--ed25519-key-restriction ED25519_KEY_

[--first-day-of-week FIRST_DAY_OF_WEEK]
[--enforce-terms ENFORCE_TERMS]
[--terms TERMS]
↪PERFORMANCE_BAR_ALLOWED_GROUP_ID] [--performance-bar-allowed-group-id↵

[--instance-statistics-visibility-private↵
↪INSTANCE_STATISTICS_VISIBILITY_PRIVATE]

[--user-show-add-ssh-key-message USER_SHOW_ADD_
↪SSH_KEY_MESSAGE]

[--file-template-project-id FILE_TEMPLATE_
↪PROJECT_ID]

[--local-markdown-version LOCAL_MARKDOWN_
↪VERSION]

[--asset-proxy-enabled ASSET_PROXY_ENABLED]
[--asset-proxy-url ASSET_PROXY_URL]
[--asset-proxy-whitelist ASSET_PROXY_WHITELIST]
[--asset-proxy-allowlist ASSET_PROXY_ALLOWLIST]
[--geo-node-allowed-ips GEO_NODE_ALLOWED_IPS]
[--allow-local-requests-from-hooks-and-
↪services ALLOW_LOCAL_REQUESTS_FROM_HOOKS_AND_SERVICES]
[--allow-local-requests-from-web-hooks-and-
↪services ALLOW_LOCAL_REQUESTS_FROM_WEB_HOOKS_AND_SERVICES]
[--allow-local-requests-from-system-hooks↵
↪ALLOW_LOCAL_REQUESTS_FROM_SYSTEM_HOOKS]

```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--id** <id>**--default-projects-limit** <default\_projects\_limit>

```
--signup-enabled <signup_enabled>
--password-authentication-enabled-for-web <password_authentication_enabled_for_web>
--gravatar-enabled <gravatar_enabled>
--sign-in-text <sign_in_text>
--created-at <created_at>
--updated-at <updated_at>
--home-page-url <home_page_url>
--default-branch-protection <default_branch_protection>
--restricted-visibility-levels <restricted_visibility_levels>
--max-attachment-size <max_attachment_size>
--session-expire-delay <session_expire_delay>
--default-project-visibility <default_project_visibility>
--default-snippet-visibility <default_snippet_visibility>
--default-group-visibility <default_group_visibility>
--outbound-local-requests-whitelist <outbound_local_requests_whitelist>
--disabled-oauth-sign-in-sources <disabled_oauth_sign_in_sources>
--domain-whitelist <domain_whitelist>
--domain-blacklist-enabled <domain_blacklist_enabled>
--domain-blacklist <domain_blacklist>
--domain-allowlist <domain_allowlist>
--domain-denylist-enabled <domain_denylist_enabled>
--domain-denylist <domain_denylist>
--external-authorization-service-enabled <external_authorization_service_enabled>
--external-authorization-service-url <external_authorization_service_url>
--external-authorization-service-default-label <external_authorization_service_default_label>
--external-authorization-service-timeout <external_authorization_service_timeout>
--import-sources <import_sources>
--user-oauth-applications <user_oauth_applications>
--after-sign-out-path <after_sign_out_path>
--container-registry-token-expire-delay <container_registry_token_expire_delay>
--repository-storages <repository_storages>
```

```

--plantuml-enabled <plantuml_enabled>
--plantuml-url <plantuml_url>
--terminal-max-session-time <terminal_max_session_time>
--polling-interval-multiplier <polling_interval_multiplier>
--rsa-key-restriction <rsa_key_restriction>
--dsa-key-restriction <dsa_key_restriction>
--ecdsa-key-restriction <ecdsa_key_restriction>
--ed25519-key-restriction <ed25519_key_restriction>
--first-day-of-week <first_day_of_week>
--enforce-terms <enforce_terms>
--terms <terms>
--performance-bar-allowed-group-id <performance_bar_allowed_group_id>
--instance-statistics-visibility-private <instance_statistics_visibility_private>
--user-show-add-ssh-key-message <user_show_add_ssh_key_message>
--file-template-project-id <file_template_project_id>
--local-markdown-version <local_markdown_version>
--asset-proxy-enabled <asset_proxy_enabled>
--asset-proxy-url <asset_proxy_url>
--asset-proxy-whitelist <asset_proxy_whitelist>
--asset-proxy-allowlist <asset_proxy_allowlist>
--geo-node-allowed-ips <geo_node_allowed_ips>
--allow-local-requests-from-hooks-and-services <allow_local_requests_from_hooks_and_services>
--allow-local-requests-from-web-hooks-and-services <allow_local_requests_from_web_hooks_and_services>
--allow-local-requests-from-system-hooks <allow_local_requests_from_system_hooks>

```

### gitlab application-statistics

```
usage: gitlab application-statistics [-h] {get} ...
```

**-h, --help**

show this help message and exit

### gitlab application-statistics get

```
usage: gitlab application-statistics get [-h] [--sudo SUDO]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

### gitlab audit-event

```
usage: gitlab audit-event [-h] {list,get} ...
```

**-h, --help**  
show this help message and exit

### gitlab audit-event get

```
usage: gitlab audit-event get [-h] [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--id** <id>

### gitlab audit-event list

```
usage: gitlab audit-event list [-h] [--sudo SUDO]
                                [--created-after CREATED_AFTER]
                                [--created-before CREATED_BEFORE]
                                [--entity-type ENTITY_TYPE]
                                [--entity-id ENTITY_ID] [--page PAGE]
                                [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--created-after** <created\_after>

**--created-before** <created\_before>

**--entity-type** <entity\_type>

**--entity-id** <entity\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

**gitlab broadcast-message**

```
usage: gitlab broadcast-message [-h] {list,get,create,update,delete} ...
```

**-h, --help**

show this help message and exit

**gitlab broadcast-message create**

```
usage: gitlab broadcast-message create [-h] [--sudo SUDO] --message MESSAGE
                                         [--starts-at STARTS_AT]
                                         [--ends-at ENDS_AT] [--color COLOR]
                                         [--font FONT]
                                         [--target-access-levels TARGET_ACCESS_LEVELS]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--message** <message>

**--starts-at** <starts\_at>

**--ends-at** <ends\_at>

**--color** <color>

**--font** <font>

**--target-access-levels** <target\_access\_levels>

**gitlab broadcast-message delete**

```
usage: gitlab broadcast-message delete [-h] [--sudo SUDO] --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>

### gitlab broadcast-message get

```
usage: gitlab broadcast-message get [-h] [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--id** <id>

### gitlab broadcast-message list

```
usage: gitlab broadcast-message list [-h] [--sudo SUDO] [--page PAGE]
                                     [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab broadcast-message update

```
usage: gitlab broadcast-message update [-h] [--sudo SUDO] --id ID
                                     [--message MESSAGE]
                                     [--starts-at STARTS_AT]
                                     [--ends-at ENDS_AT] [--color COLOR]
                                     [--font FONT]
                                     [--target-access-levels TARGET_ACCESS_LEVELS]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--id** <id>

**--message** <message>

**--starts-at** <starts\_at>

**--ends-at** <ends\_at>

**--color** <color>

**--font** <font>

**--target-access-levels** <target\_access\_levels>

## gitlab bulk-import

```
usage: gitlab bulk-import [-h] {list,get,create} ...
```

**-h, --help**

show this help message and exit

## gitlab bulk-import create

```
usage: gitlab bulk-import create [-h] [--sudo SUDO] --configuration
                                CONFIGURATION --entities ENTITIES
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--configuration** <configuration>

**--entities** <entities>

## gitlab bulk-import get

```
usage: gitlab bulk-import get [-h] [--sudo SUDO] --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>

## gitlab bulk-import list

```
usage: gitlab bulk-import list [-h] [--sudo SUDO] [--sort SORT]
                                [--status STATUS] [--page PAGE]
                                [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--sort** <sort>

**--status** <status>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab bulk-import-all-entity

```
usage: gitlab bulk-import-all-entity [-h] {list} ...
```

**-h, --help**  
show this help message and exit

### gitlab bulk-import-all-entity list

```
usage: gitlab bulk-import-all-entity list [-h] [--sudo SUDO] [--sort SORT]
                                           [--status STATUS] [--page PAGE]
                                           [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--sort** <sort>

**--status** <status>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab bulk-import-entity

```
usage: gitlab bulk-import-entity [-h] {list,get} ...
```

**-h, --help**  
show this help message and exit

### gitlab bulk-import-entity get

```
usage: gitlab bulk-import-entity get [-h] [--sudo SUDO] --bulk-import-id
BULK_IMPORT_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--bulk-import-id** <bulk\_import\_id>

**--id** <id>



### gitlab bulk-import-entity list

```
usage: gitlab bulk-import-entity list [-h] [--sudo SUDO] --bulk-import-id
                                     BULK_IMPORT_ID [--sort SORT]
                                     [--status STATUS] [--page PAGE]
                                     [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo <sudo>****--bulk-import-id <bulk\_import\_id>****--sort <sort>****--status <status>****--page <page>****--per-page <per\_page>****--get-all**

Return all items from the server, without pagination.

### gitlab ci-lint

```
usage: gitlab ci-lint [-h] {create,validate} ...
```

**-h, --help**

show this help message and exit

### gitlab ci-lint create

```
usage: gitlab ci-lint create [-h] [--sudo SUDO] --content CONTENT
                             [--include-merged-yaml INCLUDE_MERGED_YAML]
                             [--include-jobs INCLUDE_JOBS]
```

**-h, --help**

show this help message and exit

**--sudo <sudo>****--content <content>****--include-merged-yaml <include\_merged\_yaml>****--include-jobs <include\_jobs>**

### gitlab ci-lint validate

```
usage: gitlab ci-lint validate [-h] --content CONTENT
                               [--include-merged-yaml INCLUDE_MERGED_YAML]
                               [--include-jobs INCLUDE_JOBS]
```

**-h, --help**  
show this help message and exit

**--content** <content>

**--include-merged-yaml** <include\_merged\_yaml>

**--include-jobs** <include\_jobs>

### gitlab current-user

```
usage: gitlab current-user [-h] {get} ...
```

**-h, --help**  
show this help message and exit

### gitlab current-user get

```
usage: gitlab current-user get [-h] [--sudo SUDO]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

### gitlab current-user-email

```
usage: gitlab current-user-email [-h] {list,get,create,delete} ...
```

**-h, --help**  
show this help message and exit

### gitlab current-user-email create

```
usage: gitlab current-user-email create [-h] [--sudo SUDO] --email EMAIL
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--email** <email>

### gitlab current-user-email delete

```
usage: gitlab current-user-email delete [-h] [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--id** <id>

### gitlab current-user-email get

```
usage: gitlab current-user-email get [-h] [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--id** <id>

### gitlab current-user-email list

```
usage: gitlab current-user-email list [-h] [--sudo SUDO] [--page PAGE]
                                     [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab current-user-gpg-key

```
usage: gitlab current-user-gpg-key [-h] {list,get,create,delete} ...
```

**-h, --help**  
show this help message and exit

### gitlab current-user-gpg-key create

```
usage: gitlab current-user-gpg-key create [-h] [--sudo SUDO] --key KEY
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--key** <key>

### gitlab current-user-gpg-key delete

```
usage: gitlab current-user-gpg-key delete [-h] [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--id** <id>

### gitlab current-user-gpg-key get

```
usage: gitlab current-user-gpg-key get [-h] [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--id** <id>

### gitlab current-user-gpg-key list

```
usage: gitlab current-user-gpg-key list [-h] [--sudo SUDO] [--page PAGE]
                                         [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab current-user-key

```
usage: gitlab current-user-key [-h] {list,get,create,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab current-user-key create

```
usage: gitlab current-user-key create [-h] [--sudo SUDO] --title TITLE --key  
KEY
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--title** <title>

**--key** <key>

### gitlab current-user-key delete

```
usage: gitlab current-user-key delete [-h] [--sudo SUDO] --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>

### gitlab current-user-key get

```
usage: gitlab current-user-key get [-h] [--sudo SUDO] --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>

### gitlab current-user-key list

```
usage: gitlab current-user-key list [-h] [--sudo SUDO] [--page PAGE]
                                     [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab current-user-runner

```
usage: gitlab current-user-runner [-h] {create} ...
```

**-h, --help**

show this help message and exit

### gitlab current-user-runner create

```
usage: gitlab current-user-runner create [-h] [--sudo SUDO] --runner-type
                                         RUNNER_TYPE [--group-id GROUP_ID]
                                         [--project-id PROJECT_ID]
                                         [--description DESCRIPTION]
                                         [--paused PAUSED] [--locked LOCKED]
                                         [--run-untagged RUN_UNTAGGED]
                                         [--tag-list TAG_LIST]
                                         [--access-level ACCESS_LEVEL]
                                         [--maximum-timeout MAXIMUM_TIMEOUT]
                                         [--maintenance-note MAINTENANCE_NOTE]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--runner-type** <runner\_type>

**--group-id** <group\_id>

**--project-id** <project\_id>

**--description** <description>

**--paused** <paused>

**--locked** <locked>

```

--run-untagged <run_untagged>
--tag-list <tag_list>
--access-level <access_level>
--maximum-timeout <maximum_timeout>
--maintenance-note <maintenance_note>

```

### gitlab current-user-status

```
usage: gitlab current-user-status [-h] {get,update} ...
```

```

-h, --help
    show this help message and exit

```

#### gitlab current-user-status get

```
usage: gitlab current-user-status get [-h] [--sudo SUDO]
```

```

-h, --help
    show this help message and exit

--sudo <sudo>

```

#### gitlab current-user-status update

```
usage: gitlab current-user-status update [-h] [--sudo SUDO] [--emoji EMOJI]
                                         [--message MESSAGE]
```

```

-h, --help
    show this help message and exit

--sudo <sudo>

--emoji <emoji>

--message <message>

```

### gitlab deploy-key

```
usage: gitlab deploy-key [-h] {list} ...
```

```

-h, --help
    show this help message and exit

```

### gitlab deploy-key list

```
usage: gitlab deploy-key list [-h] [--sudo SUDO] [--page PAGE]
                             [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab deploy-token

```
usage: gitlab deploy-token [-h] {list} ...
```

**-h, --help**

show this help message and exit

### gitlab deploy-token list

```
usage: gitlab deploy-token list [-h] [--sudo SUDO] [--page PAGE]
                             [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab dockerfile

```
usage: gitlab dockerfile [-h] {list,get} ...
```

**-h, --help**

show this help message and exit



### gitlab dockerfile get

```
usage: gitlab dockerfile get [-h] [--sudo SUDO] --name NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--name** <name>

### gitlab dockerfile list

```
usage: gitlab dockerfile list [-h] [--sudo SUDO] [--page PAGE]
                             [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab event

```
usage: gitlab event [-h] {list} ...
```

**-h, --help**  
show this help message and exit

### gitlab event list

```
usage: gitlab event list [-h] [--sudo SUDO] [--action ACTION]
                        [--target-type TARGET_TYPE] [--before BEFORE]
                        [--after AFTER] [--sort SORT] [--scope SCOPE]
                        [--page PAGE] [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--action** <action>

**--target-type** <target\_type>

**--before** <before>

**--after** <after>

**--sort** <sort>

**--scope** <scope>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab feature

```
usage: gitlab feature [-h] {list,delete} ...
```

**-h, --help**

show this help message and exit

## gitlab feature delete

```
usage: gitlab feature delete [-h] [--sudo SUDO] --name NAME
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--name** <name>

## gitlab feature list

```
usage: gitlab feature list [-h] [--sudo SUDO] [--page PAGE]
                        [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab generic-package

```
usage: gitlab generic-package [-h] {upload,download} ...
```

### **-h, --help**

show this help message and exit

## gitlab generic-package download

```
usage: gitlab generic-package download [-h] --project-id PROJECT_ID
                                         [--sudo SUDO] --package-name
                                         PACKAGE_NAME --package-version
                                         PACKAGE_VERSION --file-name FILE_NAME
```

### **-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--package-name** <package\_name>

**--package-version** <package\_version>

**--file-name** <file\_name>

## gitlab generic-package upload

```
usage: gitlab generic-package upload [-h] --project-id PROJECT_ID
                                       [--sudo SUDO] --package-name PACKAGE_NAME
                                       --package-version PACKAGE_VERSION
                                       --file-name FILE_NAME --path PATH
```

### **-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--package-name** <package\_name>

**--package-version** <package\_version>

**--file-name** <file\_name>

**--path** <path>

## gitlab geo-node

```
usage: gitlab geo-node [-h]
      list
      get
      update
      delete
      repair
      status
      current-failures
      ...
```

**-h, --help**  
show this help message and exit

## gitlab geo-node current-failures

```
usage: gitlab geo-node current-failures [-h] --id ID
```

**-h, --help**  
show this help message and exit

**--id** <id>

## gitlab geo-node delete

```
usage: gitlab geo-node delete [-h] [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--id** <id>

## gitlab geo-node get

```
usage: gitlab geo-node get [-h] [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--id** <id>

### gitlab geo-node list

```
usage: gitlab geo-node list [-h] [--sudo SUDO] [--page PAGE]
                             [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab geo-node repair

```
usage: gitlab geo-node repair [-h] --id ID
```

**-h, --help**

show this help message and exit

**--id** <id>

### gitlab geo-node status

```
usage: gitlab geo-node status [-h] --id ID
```

**-h, --help**

show this help message and exit

**--id** <id>

### gitlab geo-node update

```
usage: gitlab geo-node update [-h] [--sudo SUDO] --id ID [--enabled ENABLED]
                               [--url URL]
                               [--files-max-capacity FILES_MAX_CAPACITY]
                               [--repos-max-capacity REPOS_MAX_CAPACITY]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>

**--enabled** <enabled>

**--url** <url>

**--files-max-capacity** <files\_max\_capacity>

**--repos-max-capacity** <repos\_max\_capacity>

## gitlab gitignore

```
usage: gitlab gitignore [-h] {list,get} ...
```

**-h, --help**

show this help message and exit

## gitlab gitignore get

```
usage: gitlab gitignore get [-h] [--sudo SUDO] --name NAME
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--name** <name>

## gitlab gitignore list

```
usage: gitlab gitignore list [-h] [--sudo SUDO] [--page PAGE]
                             [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab gitlabciyaml

```
usage: gitlab gitlabciyaml [-h] {list,get} ...
```

**-h, --help**

show this help message and exit

### gitlab gitlabciyaml get

```
usage: gitlab gitlabciyaml get [-h] [--sudo SUDO] --name NAME
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--name** <name>

### gitlab gitlabciyaml list

```
usage: gitlab gitlabciyaml list [-h] [--sudo SUDO] [--page PAGE]
                                [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab group

```
usage: gitlab group [-h]
  list
  get
  create
  update
  delete
  transfer-project
  transfer
  search
  ldap-sync
  share
  unshare
  restore
  ...
```

**-h, --help**

show this help message and exit

## gitlab group create

```
usage: gitlab group create [-h] [--sudo SUDO] --name NAME --path PATH
                           [--description DESCRIPTION]
                           [--membership-lock MEMBERSHIP_LOCK]
                           [--visibility VISIBILITY]
                           [--share-with-group-lock SHARE_WITH_GROUP_LOCK]
                           [--require-two-factor-authentication REQUIRE_TWO_FACTOR_
↪AUTHENTICATION]
                           [--two-factor-grace-period TWO_FACTOR_GRACE_PERIOD]
                           [--project-creation-level PROJECT_CREATION_LEVEL]
                           [--auto-devops-enabled AUTO_DEVOPS_ENABLED]
                           [--subgroup-creation-level SUBGROUP_CREATION_LEVEL]
                           [--emails-disabled EMAILS_DISABLED]
                           [--avatar AVATAR]
                           [--mentions-disabled MENTIONS_DISABLED]
                           [--lfs-enabled LFS_ENABLED]
                           [--request-access-enabled REQUEST_ACCESS_ENABLED]
                           [--parent-id PARENT_ID]
                           [--default-branch-protection DEFAULT_BRANCH_PROTECTION]
                           [--shared-runners-minutes-limit SHARED_RUNNERS_MINUTES_LIMIT]
                           [--extra-shared-runners-minutes-limit EXTRA_SHARED_RUNNERS_
↪MINUTES_LIMIT]
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--name** <name>

**--path** <path>

**--description** <description>

**--membership-lock** <membership\_lock>

**--visibility** <visibility>

**--share-with-group-lock** <share\_with\_group\_lock>

**--require-two-factor-authentication** <require\_two\_factor\_authentication>

**--two-factor-grace-period** <two\_factor\_grace\_period>

**--project-creation-level** <project\_creation\_level>

**--auto-devops-enabled** <auto\_devops\_enabled>

**--subgroup-creation-level** <subgroup\_creation\_level>

**--emails-disabled** <emails\_disabled>

**--avatar** <avatar>

**--mentions-disabled** <mentions\_disabled>



```

--lfs-enabled <lfs_enabled>
--request-access-enabled <request_access_enabled>
--parent-id <parent_id>
--default-branch-protection <default_branch_protection>
--shared-runners-minutes-limit <shared_runners_minutes_limit>
--extra-shared-runners-minutes-limit <extra_shared_runners_minutes_limit>

```

### gitlab group delete

```
usage: gitlab group delete [-h] [--sudo SUDO] --id ID
```

```

-h, --help
    show this help message and exit
--sudo <sudo>
--id <id>

```

### gitlab group get

```
usage: gitlab group get [-h] [--sudo SUDO] --id ID
```

```

-h, --help
    show this help message and exit
--sudo <sudo>
--id <id>

```

### gitlab group ldap-sync

```
usage: gitlab group ldap-sync [-h] --id ID
```

```

-h, --help
    show this help message and exit
--id <id>

```

### gitlab group list

```

usage: gitlab group list [-h] [--sudo SUDO] [--skip-groups SKIP_GROUPS]
                        [--all-available ALL_AVAILABLE] [--search SEARCH]
                        [--order-by ORDER_BY] [--sort SORT]
                        [--statistics STATISTICS] [--owned OWNED]
                        [--with-custom-attributes WITH_CUSTOM_ATTRIBUTES]
                        [--min-access-level MIN_ACCESS_LEVEL]

```

(continues on next page)

(continued from previous page)

```
[--top-level-only TOP_LEVEL_ONLY] [--page PAGE]
[--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--skip-groups** <skip\_groups>

**--all-available** <all\_available>

**--search** <search>

**--order-by** <order\_by>

**--sort** <sort>

**--statistics** <statistics>

**--owned** <owned>

**--with-custom-attributes** <with\_custom\_attributes>

**--min-access-level** <min\_access\_level>

**--top-level-only** <top\_level\_only>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab group restore

```
usage: gitlab group restore [-h] --id ID
```

**-h, --help**  
show this help message and exit

**--id** <id>

### gitlab group search

```
usage: gitlab group search [-h] --id ID --scope SCOPE --search SEARCH
```

**-h, --help**  
show this help message and exit

**--id** <id>

**--scope** <scope>

**--search** <search>

### gitlab group share

```
usage: gitlab group share [-h] --id ID --group-id GROUP_ID --group-access
                           GROUP_ACCESS [--expires-at EXPIRES_AT]
```

**-h, --help**  
show this help message and exit

**--id <id>**

**--group-id <group\_id>**

**--group-access <group\_access>**

**--expires-at <expires\_at>**

### gitlab group transfer

```
usage: gitlab group transfer [-h] --id ID [--group-id GROUP_ID]
```

**-h, --help**  
show this help message and exit

**--id <id>**

**--group-id <group\_id>**

### gitlab group transfer-project

```
usage: gitlab group transfer-project [-h] --id ID --project-id PROJECT_ID
```

**-h, --help**  
show this help message and exit

**--id <id>**

**--project-id <project\_id>**

### gitlab group unshare

```
usage: gitlab group unshare [-h] --id ID --group-id GROUP_ID
```

**-h, --help**  
show this help message and exit

**--id <id>**

**--group-id <group\_id>**

## gitlab group update

```
usage: gitlab group update [-h] [--sudo SUDO] --id ID [--name NAME]
                             [--path PATH] [--description DESCRIPTION]
                             [--membership-lock MEMBERSHIP_LOCK]
                             [--share-with-group-lock SHARE_WITH_GROUP_LOCK]
                             [--visibility VISIBILITY]
                             [--require-two-factor-authentication REQUIRE_TWO_FACTOR_
↪AUTHENTICATION]
                             [--two-factor-grace-period TWO_FACTOR_GRACE_PERIOD]
                             [--project-creation-level PROJECT_CREATION_LEVEL]
                             [--auto-devops-enabled AUTO_DEVOPS_ENABLED]
                             [--subgroup-creation-level SUBGROUP_CREATION_LEVEL]
                             [--emails-disabled EMAILS_DISABLED]
                             [--avatar AVATAR]
                             [--mentions-disabled MENTIONS_DISABLED]
                             [--lfs-enabled LFS_ENABLED]
                             [--request-access-enabled REQUEST_ACCESS_ENABLED]
                             [--default-branch-protection DEFAULT_BRANCH_PROTECTION]
                             [--file-template-project-id FILE_TEMPLATE_PROJECT_ID]
                             [--shared-runners-minutes-limit SHARED_RUNNERS_MINUTES_LIMIT]
↪MINUTES_LIMIT]
                             [--extra-shared-runners-minutes-limit EXTRA_SHARED_RUNNERS_
↪GROUP]
                             [--prevent-forking-outside-group PREVENT_FORKING_OUTSIDE_
                             [--shared-runners-setting SHARED_RUNNERS_SETTING]
```

### -h, --help

show this help message and exit

**--sudo** <sudo>

**--id** <id>

**--name** <name>

**--path** <path>

**--description** <description>

**--membership-lock** <membership\_lock>

**--share-with-group-lock** <share\_with\_group\_lock>

**--visibility** <visibility>

**--require-two-factor-authentication** <require\_two\_factor\_authentication>

**--two-factor-grace-period** <two\_factor\_grace\_period>

**--project-creation-level** <project\_creation\_level>

**--auto-devops-enabled** <auto\_devops\_enabled>

**--subgroup-creation-level** <subgroup\_creation\_level>

```

--emails-disabled <emails_disabled>
--avatar <avatar>
--mentions-disabled <mentions_disabled>
--lfs-enabled <lfs_enabled>
--request-access-enabled <request_access_enabled>
--default-branch-protection <default_branch_protection>
--file-template-project-id <file_template_project_id>
--shared-runners-minutes-limit <shared_runners_minutes_limit>
--extra-shared-runners-minutes-limit <extra_shared_runners_minutes_limit>
--prevent-forking-outside-group <prevent_forking_outside_group>
--shared-runners-setting <shared_runners_setting>

```

### gitlab group-access-request

```
usage: gitlab group-access-request [-h] {list,create,delete,approve} ...
```

**-h, --help**  
show this help message and exit

### gitlab group-access-request approve

```
usage: gitlab group-access-request approve [-h] --group-id GROUP_ID
                                           [--sudo SUDO] --id ID
                                           [--access-level ACCESS_LEVEL]
```

**-h, --help**  
show this help message and exit

**--group-id** <group\_id>

**--sudo** <sudo>

**--id** <id>

**--access-level** <access\_level>

### gitlab group-access-request create

```
usage: gitlab group-access-request create [-h] [--sudo SUDO] --group-id
                                         GROUP_ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

### gitlab group-access-request delete

```
usage: gitlab group-access-request delete [-h] [--sudo SUDO] --group-id
                                         GROUP_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

### gitlab group-access-request list

```
usage: gitlab group-access-request list [-h] [--sudo SUDO] --group-id GROUP_ID
                                         [--page PAGE] [--per-page PER_PAGE]
                                         [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab group-access-token

```
usage: gitlab group-access-token [-h] {list,get,create,delete} ...
```

**-h, --help**

show this help message and exit

## gitlab group-access-token create

```
usage: gitlab group-access-token create [-h] [--sudo SUDO] --group-id GROUP_ID
                                         --name NAME --scopes SCOPES
                                         [--access-level ACCESS_LEVEL]
                                         [--expires-at EXPIRES_AT]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--name** <name>

**--scopes** <scopes>

**--access-level** <access\_level>

**--expires-at** <expires\_at>

## gitlab group-access-token delete

```
usage: gitlab group-access-token delete [-h] [--sudo SUDO] --group-id GROUP_ID
                                         --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

## gitlab group-access-token get

```
usage: gitlab group-access-token get [-h] [--sudo SUDO] --group-id GROUP_ID
                                       --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>  
**--id** <id>

### gitlab group-access-token list

```
usage: gitlab group-access-token list [-h] [--sudo SUDO] --group-id GROUP_ID
                                     [--page PAGE] [--per-page PER_PAGE]
                                     [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab group-audit-event

```
usage: gitlab group-audit-event [-h] {list,get} ...
```

**-h, --help**  
show this help message and exit

### gitlab group-audit-event get

```
usage: gitlab group-audit-event get [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>



### gitlab group-audit-event list

```
usage: gitlab group-audit-event list [-h] [--sudo SUDO] --group-id GROUP_ID
                                     [--created-after CREATED_AFTER]
                                     [--created-before CREATED_BEFORE]
                                     [--page PAGE] [--per-page PER_PAGE]
                                     [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--group-id** <group\_id>**--created-after** <created\_after>**--created-before** <created\_before>**--page** <page>**--per-page** <per\_page>**--get-all**

Return all items from the server, without pagination.

### gitlab group-badge

```
usage: gitlab group-badge [-h] {list,get,create,update,delete,render} ...
```

**-h, --help**

show this help message and exit

### gitlab group-badge create

```
usage: gitlab group-badge create [-h] [--sudo SUDO] --group-id GROUP_ID
                                  --link-url LINK_URL --image-url IMAGE_URL
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--group-id** <group\_id>**--link-url** <link\_url>**--image-url** <image\_url>

### gitlab group-badge delete

```
usage: gitlab group-badge delete [-h] [--sudo SUDO] --group-id GROUP_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

### gitlab group-badge get

```
usage: gitlab group-badge get [-h] [--sudo SUDO] --group-id GROUP_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

### gitlab group-badge list

```
usage: gitlab group-badge list [-h] [--sudo SUDO] --group-id GROUP_ID
                                [--page PAGE] [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab group-badge render

```
usage: gitlab group-badge render [-h] --group-id GROUP_ID [--sudo SUDO] --id
                                ID --link-url LINK_URL --image-url IMAGE_URL
```

**-h, --help**  
show this help message and exit

**--group-id** <group\_id>

**--sudo** <sudo>

**--id** <id>

**--link-url** <link\_url>

**--image-url** <image\_url>

### gitlab group-badge update

```
usage: gitlab group-badge update [-h] [--sudo SUDO] --group-id GROUP_ID --id
                                ID [--link-url LINK_URL]
                                [--image-url IMAGE_URL]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

**--link-url** <link\_url>

**--image-url** <image\_url>

### gitlab group-billable-member

```
usage: gitlab group-billable-member [-h] {list,delete} ...
```

**-h, --help**  
show this help message and exit

### gitlab group-billable-member delete

```
usage: gitlab group-billable-member delete [-h] [--sudo SUDO] --group-id
                                           GROUP_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>  
**--group-id** <group\_id>  
**--id** <id>

### gitlab group-billable-member list

```
usage: gitlab group-billable-member list [-h] [--sudo SUDO] --group-id
                                         GROUP_ID [--search SEARCH]
                                         [--sort SORT] [--page PAGE]
                                         [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--search** <search>

**--sort** <sort>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab group-billable-member-membership

```
usage: gitlab group-billable-member-membership [-h] {list} ...
```

**-h, --help**  
show this help message and exit

### gitlab group-billable-member-membership list

```
usage: gitlab group-billable-member-membership list [-h] [--sudo SUDO]
                                                    --group-id GROUP_ID
                                                    --user-id USER_ID
                                                    [--page PAGE]
                                                    [--per-page PER_PAGE]
                                                    [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>  
**--user-id** <user\_id>  
**--page** <page>  
**--per-page** <per\_page>  
**--get-all**  
Return all items from the server, without pagination.

### gitlab group-board

```
usage: gitlab group-board [-h] {list,get,create,update,delete} ...
```

**-h, --help**  
show this help message and exit

#### gitlab group-board create

```
usage: gitlab group-board create [-h] [--sudo SUDO] --group-id GROUP_ID --name  
NAME
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--group-id** <group\_id>  
**--name** <name>

#### gitlab group-board delete

```
usage: gitlab group-board delete [-h] [--sudo SUDO] --group-id GROUP_ID --id  
ID
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--group-id** <group\_id>  
**--id** <id>

### gitlab group-board get

```
usage: gitlab group-board get [-h] [--sudo SUDO] --group-id GROUP_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

### gitlab group-board list

```
usage: gitlab group-board list [-h] [--sudo SUDO] --group-id GROUP_ID
                               [--page PAGE] [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab group-board update

```
usage: gitlab group-board update [-h] [--sudo SUDO] --group-id GROUP_ID --id
                                   ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

## gitlab group-board-list

```
usage: gitlab group-board-list [-h] {list,get,create,update,delete} ...
```

**-h, --help**

show this help message and exit

## gitlab group-board-list create

```
usage: gitlab group-board-list create [-h] [--sudo SUDO] --group-id GROUP_ID
                                         --board-id BOARD_ID
                                         [--label-id LABEL_ID | --assignee-id ASSIGNEE_ID |
↳ --milestone-id MILESTONE_ID]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--board-id** <board\_id>

**--label-id** <label\_id>

**--assignee-id** <assignee\_id>

**--milestone-id** <milestone\_id>

## gitlab group-board-list delete

```
usage: gitlab group-board-list delete [-h] [--sudo SUDO] --group-id GROUP_ID
                                         --board-id BOARD_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--board-id** <board\_id>

**--id** <id>

### gitlab group-board-list get

```
usage: gitlab group-board-list get [-h] [--sudo SUDO] --group-id GROUP_ID
                                   --board-id BOARD_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--board-id** <board\_id>

**--id** <id>

### gitlab group-board-list list

```
usage: gitlab group-board-list list [-h] [--sudo SUDO] --group-id GROUP_ID
                                   --board-id BOARD_ID [--page PAGE]
                                   [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--board-id** <board\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab group-board-list update

```
usage: gitlab group-board-list update [-h] [--sudo SUDO] --group-id GROUP_ID
                                   --board-id BOARD_ID --id ID --position
                                   POSITION
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--board-id** <board\_id>

**--id** <id>

**--position** <position>



## gitlab group-cluster

```
usage: gitlab group-cluster [-h] {list,get,create,update,delete} ...
```

**-h, --help**

show this help message and exit

## gitlab group-cluster create

```
usage: gitlab group-cluster create [-h] [--sudo SUDO] --group-id GROUP_ID
                                   --name NAME
                                   --platform-kubernetes-attributes
                                   PLATFORM_KUBERNETES_ATTRIBUTES
                                   [--domain DOMAIN] [--enabled ENABLED]
                                   [--managed MANAGED]
                                   [--environment-scope ENVIRONMENT_SCOPE]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--name** <name>

**--platform-kubernetes-attributes** <platform\_kubernetes\_attributes>

**--domain** <domain>

**--enabled** <enabled>

**--managed** <managed>

**--environment-scope** <environment\_scope>

## gitlab group-cluster delete

```
usage: gitlab group-cluster delete [-h] [--sudo SUDO] --group-id GROUP_ID --id
                                   ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

### gitlab group-cluster get

```
usage: gitlab group-cluster get [-h] [--sudo SUDO] --group-id GROUP_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

### gitlab group-cluster list

```
usage: gitlab group-cluster list [-h] [--sudo SUDO] --group-id GROUP_ID
                                [--page PAGE] [--per-page PER_PAGE]
                                [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab group-cluster update

```
usage: gitlab group-cluster update [-h] [--sudo SUDO] --group-id GROUP_ID --id
                                   ID [--name NAME] [--domain DOMAIN]
                                   [--management-project-id MANAGEMENT_PROJECT_ID]
                                   [--platform-kubernetes-attributes PLATFORM_KUBERNETES_
↪ATTRIBUTES]
                                   [--environment-scope ENVIRONMENT_SCOPE]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

**--name** <name>

**--domain** <domain>

```
--management-project-id <management_project_id>
--platform-kubernetes-attributes <platform_kubernetes_attributes>
--environment-scope <environment_scope>
```

### gitlab group-custom-attribute

```
usage: gitlab group-custom-attribute [-h] {list,get,delete} ...
```

**-h, --help**  
show this help message and exit

### gitlab group-custom-attribute delete

```
usage: gitlab group-custom-attribute delete [-h] [--sudo SUDO] --group-id
      GROUP_ID --key KEY
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--key** <key>

### gitlab group-custom-attribute get

```
usage: gitlab group-custom-attribute get [-h] [--sudo SUDO] --group-id
      GROUP_ID --key KEY
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--key** <key>

### gitlab group-custom-attribute list

```
usage: gitlab group-custom-attribute list [-h] [--sudo SUDO] --group-id
      GROUP_ID [--page PAGE]
      [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab group-deploy-token

```
usage: gitlab group-deploy-token [-h] {list,get,create,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab group-deploy-token create

```
usage: gitlab group-deploy-token create [-h] [--sudo SUDO] --group-id GROUP_ID
                                         --name NAME --scopes SCOPES
                                         [--expires-at EXPIRES_AT]
                                         [--username USERNAME]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--name** <name>

**--scopes** <scopes>

**--expires-at** <expires\_at>

**--username** <username>

### gitlab group-deploy-token delete

```
usage: gitlab group-deploy-token delete [-h] [--sudo SUDO] --group-id GROUP_ID
                                         --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

### gitlab group-deploy-token get

```
usage: gitlab group-deploy-token get [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

### gitlab group-deploy-token list

```
usage: gitlab group-deploy-token list [-h] [--sudo SUDO] --group-id GROUP_ID
                                     [--scopes SCOPES] [--page PAGE]
                                     [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--scopes** <scopes>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab group-descendant-group

```
usage: gitlab group-descendant-group [-h] {list} ...
```

**-h, --help**

show this help message and exit

### gitlab group-descendant-group list

```
usage: gitlab group-descendant-group list [-h] [--sudo SUDO] --group-id
GROUP_ID [--skip-groups SKIP_GROUPS]
         [--all-available ALL_AVAILABLE]
         [--search SEARCH]
         [--order-by ORDER_BY] [--sort SORT]
         [--statistics STATISTICS]
```

(continues on next page)

(continued from previous page)

```
↪ATTRIBUTES]
[--owned OWNED]
[--with-custom-attributes WITH_CUSTOM_

[--min-access-level MIN_ACCESS_LEVEL]
[--page PAGE] [--per-page PER_PAGE]
[--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--group-id** <group\_id>**--skip-groups** <skip\_groups>**--all-available** <all\_available>**--search** <search>**--order-by** <order\_by>**--sort** <sort>**--statistics** <statistics>**--owned** <owned>**--with-custom-attributes** <with\_custom\_attributes>**--min-access-level** <min\_access\_level>**--page** <page>**--per-page** <per\_page>**--get-all**

Return all items from the server, without pagination.

## gitlab group-epic

```
usage: gitlab group-epic [-h] {list,get,create,update,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab group-epic create

```
usage: gitlab group-epic create [-h] [--sudo SUDO] --group-id GROUP_ID --title
                                TITLE [--labels LABELS]
                                [--description DESCRIPTION]
                                [--start-date START_DATE]
                                [--end-date END_DATE]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--title** <title>

**--labels** <labels>

**--description** <description>

**--start-date** <start\_date>

**--end-date** <end\_date>

### gitlab group-epic delete

```
usage: gitlab group-epic delete [-h] [--sudo SUDO] --group-id GROUP_ID --iid
                                IID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--iid** <iid>

### gitlab group-epic get

```
usage: gitlab group-epic get [-h] [--sudo SUDO] --group-id GROUP_ID --iid IID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--iid** <iid>

## gitlab group-epic list

```
usage: gitlab group-epic list [-h] [--sudo SUDO] --group-id GROUP_ID
                               [--author-id AUTHOR_ID] [--labels LABELS]
                               [--order-by ORDER_BY] [--sort SORT]
                               [--search SEARCH] [--page PAGE]
                               [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--author-id** <author\_id>

**--labels** <labels>

**--order-by** <order\_by>

**--sort** <sort>

**--search** <search>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab group-epic update

```
usage: gitlab group-epic update [-h] [--sudo SUDO] --group-id GROUP_ID --iid
                                IID [--title TITLE] [--labels LABELS]
                                [--description DESCRIPTION]
                                [--start-date START_DATE]
                                [--end-date END_DATE]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--iid** <iid>

**--title** <title>

**--labels** <labels>

**--description** <description>

**--start-date** <start\_date>

**--end-date** <end\_date>



## gitlab group-epic-award-emoji

```
usage: gitlab group-epic-award-emoji [-h] {list,get,create,delete} ...
```

**-h, --help**

show this help message and exit

## gitlab group-epic-award-emoji create

```
usage: gitlab group-epic-award-emoji create [-h] [--sudo SUDO] --group-id
                                             GROUP_ID --epic-iid EPIC_IID
                                             --name NAME
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--epic-iid** <epic\_iid>

**--name** <name>

## gitlab group-epic-award-emoji delete

```
usage: gitlab group-epic-award-emoji delete [-h] [--sudo SUDO] --group-id
                                             GROUP_ID --epic-iid EPIC_IID --id
                                             ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--epic-iid** <epic\_iid>

**--id** <id>

## gitlab group-epic-award-emoji get

```
usage: gitlab group-epic-award-emoji get [-h] [--sudo SUDO] --group-id
                                           GROUP_ID --epic-iid EPIC_IID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--epic-iid** <epic\_iid>

**--id** <id>

### gitlab group-epic-award-emoji list

```
usage: gitlab group-epic-award-emoji list [-h] [--sudo SUDO] --group-id
                                           GROUP_ID --epic-iid EPIC_IID
                                           [--page PAGE] [--per-page PER_PAGE]
                                           [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--epic-iid** <epic\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab group-epic-discussion-note

```
usage: gitlab group-epic-discussion-note [-h] {get,create,update,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab group-epic-discussion-note create

```
usage: gitlab group-epic-discussion-note create [-h] [--sudo SUDO] --group-id
                                                  GROUP_ID --epic-id EPIC_ID
                                                  --discussion-id DISCUSSION_ID
                                                  --body BODY
                                                  [--created-at CREATED_AT]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--epic-id** <epic\_id>

**--discussion-id** <discussion\_id>

**--body** <body>  
**--created-at** <created\_at>

### gitlab group-epic-discussion-note delete

```
usage: gitlab group-epic-discussion-note delete [-h] [--sudo SUDO] --group-id
        GROUP_ID --epic-id EPIC_ID
        --discussion-id DISCUSSION_ID
        --id ID
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--group-id** <group\_id>  
**--epic-id** <epic\_id>  
**--discussion-id** <discussion\_id>  
**--id** <id>

### gitlab group-epic-discussion-note get

```
usage: gitlab group-epic-discussion-note get [-h] [--sudo SUDO] --group-id
        GROUP_ID --epic-id EPIC_ID
        --discussion-id DISCUSSION_ID
        --id ID
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--group-id** <group\_id>  
**--epic-id** <epic\_id>  
**--discussion-id** <discussion\_id>  
**--id** <id>

### gitlab group-epic-discussion-note update

```
usage: gitlab group-epic-discussion-note update [-h] [--sudo SUDO] --group-id
        GROUP_ID --epic-id EPIC_ID
        --discussion-id DISCUSSION_ID
        --id ID --body BODY
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>  
**--group-id** <group\_id>  
**--epic-id** <epic\_id>  
**--discussion-id** <discussion\_id>  
**--id** <id>  
**--body** <body>

### gitlab group-epic-issue

```
usage: gitlab group-epic-issue [-h] {list,create,update,delete} ...
```

**-h, --help**  
show this help message and exit

### gitlab group-epic-issue create

```
usage: gitlab group-epic-issue create [-h] [--sudo SUDO] --group-id GROUP_ID  
--epic-iid EPIC_IID --issue-id ISSUE_ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>  
**--group-id** <group\_id>  
**--epic-iid** <epic\_iid>  
**--issue-id** <issue\_id>

### gitlab group-epic-issue delete

```
usage: gitlab group-epic-issue delete [-h] [--sudo SUDO] --group-id GROUP_ID  
--epic-iid EPIC_IID --epic-issue-id  
EPIC_ISSUE_ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>  
**--group-id** <group\_id>  
**--epic-iid** <epic\_iid>  
**--epic-issue-id** <epic\_issue\_id>

### gitlab group-epic-issue list

```
usage: gitlab group-epic-issue list [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --epic-iid EPIC_IID [--page PAGE]
                                     [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--group-id** <group\_id>**--epic-iid** <epic\_iid>**--page** <page>**--per-page** <per\_page>**--get-all**

Return all items from the server, without pagination.

### gitlab group-epic-issue update

```
usage: gitlab group-epic-issue update [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --epic-iid EPIC_IID --epic-issue-id
                                     EPIC_ISSUE_ID
                                     [--move-before-id MOVE_BEFORE_ID]
                                     [--move-after-id MOVE_AFTER_ID]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--group-id** <group\_id>**--epic-iid** <epic\_iid>**--epic-issue-id** <epic\_issue\_id>**--move-before-id** <move\_before\_id>**--move-after-id** <move\_after\_id>

### gitlab group-epic-note

```
usage: gitlab group-epic-note [-h] {list,get,create,update,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab group-epic-note create

```
usage: gitlab group-epic-note create [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --epic-id EPIC_ID --body BODY
                                     [--created-at CREATED_AT]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--epic-id** <epic\_id>

**--body** <body>

**--created-at** <created\_at>

### gitlab group-epic-note delete

```
usage: gitlab group-epic-note delete [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --epic-id EPIC_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--epic-id** <epic\_id>

**--id** <id>

### gitlab group-epic-note get

```
usage: gitlab group-epic-note get [-h] [--sudo SUDO] --group-id GROUP_ID
                                   --epic-id EPIC_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--epic-id** <epic\_id>

**--id** <id>

### gitlab group-epic-note list

```
usage: gitlab group-epic-note list [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --epic-id EPIC_ID [--page PAGE]
                                     [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--group-id** <group\_id>**--epic-id** <epic\_id>**--page** <page>**--per-page** <per\_page>**--get-all**

Return all items from the server, without pagination.

### gitlab group-epic-note update

```
usage: gitlab group-epic-note update [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --epic-id EPIC_ID --id ID --body BODY
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--group-id** <group\_id>**--epic-id** <epic\_id>**--id** <id>**--body** <body>

### gitlab group-epic-note-award-emoji

```
usage: gitlab group-epic-note-award-emoji [-h] {list,get,create,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab group-epic-note-award-emoji create

```
usage: gitlab group-epic-note-award-emoji create [-h] [--sudo SUDO] --group-id
                                                GROUP_ID --epic-iid EPIC_IID
                                                --note-id NOTE_ID --name NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--epic-iid** <epic\_iid>

**--note-id** <note\_id>

**--name** <name>

### gitlab group-epic-note-award-emoji delete

```
usage: gitlab group-epic-note-award-emoji delete [-h] [--sudo SUDO] --group-id
                                                GROUP_ID --epic-iid EPIC_IID
                                                --note-id NOTE_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--epic-iid** <epic\_iid>

**--note-id** <note\_id>

**--id** <id>

### gitlab group-epic-note-award-emoji get

```
usage: gitlab group-epic-note-award-emoji get [-h] [--sudo SUDO] --group-id
                                                GROUP_ID --epic-iid EPIC_IID
                                                --note-id NOTE_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--epic-iid** <epic\_iid>

**--note-id** <note\_id>

**--id** <id>



### gitlab group-epic-note-award-emoji list

```
usage: gitlab group-epic-note-award-emoji list [-h] [--sudo SUDO] --group-id
                                                GROUP_ID --epic-iid EPIC_IID
                                                --note-id NOTE_ID [--page PAGE]
                                                [--per-page PER_PAGE]
                                                [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--group-id** <group\_id>**--epic-iid** <epic\_iid>**--note-id** <note\_id>**--page** <page>**--per-page** <per\_page>**--get-all**

Return all items from the server, without pagination.

### gitlab group-epic-resource-label-event

```
usage: gitlab group-epic-resource-label-event [-h] {list,get} ...
```

**-h, --help**

show this help message and exit

### gitlab group-epic-resource-label-event get

```
usage: gitlab group-epic-resource-label-event get [-h] [--sudo SUDO]
                                                    --group-id GROUP_ID
                                                    --epic-id EPIC_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--group-id** <group\_id>**--epic-id** <epic\_id>**--id** <id>

### gitlab group-epic-resource-label-event list

```
usage: gitlab group-epic-resource-label-event list [-h] [--sudo SUDO]
                                                    --group-id GROUP_ID
                                                    --epic-id EPIC_ID
                                                    [--page PAGE]
                                                    [--per-page PER_PAGE]
                                                    [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--epic-id** <epic\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab group-export

```
usage: gitlab group-export [-h] {get,create,download} ...
```

**-h, --help**  
show this help message and exit

#### gitlab group-export create

```
usage: gitlab group-export create [-h] [--sudo SUDO] --group-id GROUP_ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

#### gitlab group-export download

```
usage: gitlab group-export download [-h] --group-id GROUP_ID [--sudo SUDO]
```

**-h, --help**  
show this help message and exit

**--group-id** <group\_id>

**--sudo** <sudo>

## gitlab group-export get

```
usage: gitlab group-export get [-h] [--sudo SUDO] --group-id GROUP_ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

## gitlab group-hook

```
usage: gitlab group-hook [-h] {list,get,create,update,delete} ...
```

**-h, --help**  
show this help message and exit

## gitlab group-hook create

```
usage: gitlab group-hook create [-h] [--sudo SUDO] --group-id GROUP_ID --url
                                URL [--push-events PUSH_EVENTS]
                                [--issues-events ISSUES_EVENTS]
                                [--confidential-issues-events CONFIDENTIAL_ISSUES_EVENTS]
                                [--merge-requests-events MERGE_REQUESTS_EVENTS]
                                [--tag-push-events TAG_PUSH_EVENTS]
                                [--note-events NOTE_EVENTS]
                                [--confidential-note-events CONFIDENTIAL_NOTE_EVENTS]
                                [--job-events JOB_EVENTS]
                                [--pipeline-events PIPELINE_EVENTS]
                                [--wiki-page-events WIKI_PAGE_EVENTS]
                                [--deployment-events DEPLOYMENT_EVENTS]
                                [--releases-events RELEASES_EVENTS]
                                [--subgroup-events SUBGROUP_EVENTS]
                                [--enable-ssl-verification ENABLE_SSL_VERIFICATION]
                                [--token TOKEN]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--url** <url>

**--push-events** <push\_events>

**--issues-events** <issues\_events>

**--confidential-issues-events** <confidential\_issues\_events>

**--merge-requests-events** <merge\_requests\_events>

```
--tag-push-events <tag_push_events>
--note-events <note_events>
--confidential-note-events <confidential_note_events>
--job-events <job_events>
--pipeline-events <pipeline_events>
--wiki-page-events <wiki_page_events>
--deployment-events <deployment_events>
--releases-events <releases_events>
--subgroup-events <subgroup_events>
--enable-ssl-verification <enable_ssl_verification>
--token <token>
```

### gitlab group-hook delete

```
usage: gitlab group-hook delete [-h] [--sudo SUDO] --group-id GROUP_ID --id ID
```

```
-h, --help
    show this help message and exit
--sudo <sudo>
--group-id <group_id>
--id <id>
```

### gitlab group-hook get

```
usage: gitlab group-hook get [-h] [--sudo SUDO] --group-id GROUP_ID --id ID
```

```
-h, --help
    show this help message and exit
--sudo <sudo>
--group-id <group_id>
--id <id>
```

## gitlab group-hook list

```
usage: gitlab group-hook list [-h] [--sudo SUDO] --group-id GROUP_ID
                               [--page PAGE] [--per-page PER_PAGE] [--get-all]
```

### -h, --help

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--page** <page>

**--per-page** <per\_page>

### --get-all

Return all items from the server, without pagination.

## gitlab group-hook update

```
usage: gitlab group-hook update [-h] [--sudo SUDO] --group-id GROUP_ID --id ID
                                --url URL [--push-events PUSH_EVENTS]
                                [--issues-events ISSUES_EVENTS]
                                [--confidential-issues-events CONFIDENTIAL_ISSUES_EVENTS]
                                [--merge-requests-events MERGE_REQUESTS_EVENTS]
                                [--tag-push-events TAG_PUSH_EVENTS]
                                [--note-events NOTE_EVENTS]
                                [--confidential-note-events CONFIDENTIAL_NOTE_EVENTS]
                                [--job-events JOB_EVENTS]
                                [--pipeline-events PIPELINE_EVENTS]
                                [--wiki-page-events WIKI_PAGE_EVENTS]
                                [--deployment-events DEPLOYMENT_EVENTS]
                                [--releases-events RELEASES_EVENTS]
                                [--subgroup-events SUBGROUP_EVENTS]
                                [--enable-ssl-verification ENABLE_SSL_VERIFICATION]
                                [--token TOKEN]
```

### -h, --help

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

**--url** <url>

**--push-events** <push\_events>

**--issues-events** <issues\_events>

**--confidential-issues-events** <confidential\_issues\_events>

```
--merge-requests-events <merge_requests_events>
--tag-push-events <tag_push_events>
--note-events <note_events>
--confidential-note-events <confidential_note_events>
--job-events <job_events>
--pipeline-events <pipeline_events>
--wiki-page-events <wiki_page_events>
--deployment-events <deployment_events>
--releases-events <releases_events>
--subgroup-events <subgroup_events>
--enable-ssl-verification <enable_ssl_verification>
--token <token>
```

### gitlab group-import

```
usage: gitlab group-import [-h] {get} ...
```

**-h, --help**  
show this help message and exit

#### gitlab group-import get

```
usage: gitlab group-import get [-h] [--sudo SUDO] --group-id GROUP_ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

### gitlab group-invitation

```
usage: gitlab group-invitation [-h] {list,get,create,update,delete} ...
```

**-h, --help**  
show this help message and exit

### gitlab group-invitation create

```
usage: gitlab group-invitation create [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --access-level ACCESS_LEVEL
                                     [--expires-at EXPIRES_AT]
                                     [--invite-source INVITE_SOURCE]
                                     [--tasks-to-be-done TASKS_TO_BE_DONE]
                                     [--tasks-project-id TASKS_PROJECT_ID]
                                     [--email EMAIL | --user-id USER_ID]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--group-id** <group\_id>**--access-level** <access\_level>**--expires-at** <expires\_at>**--invite-source** <invite\_source>**--tasks-to-be-done** <tasks\_to\_be\_done>**--tasks-project-id** <tasks\_project\_id>**--email** <email>**--user-id** <user\_id>

### gitlab group-invitation delete

```
usage: gitlab group-invitation delete [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --email EMAIL
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--group-id** <group\_id>**--email** <email>

### gitlab group-invitation get

```
usage: gitlab group-invitation get [-h] [--sudo SUDO] --group-id GROUP_ID
                                   --email EMAIL
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

```
--group-id <group_id>
--email <email>
```

### gitlab group-invitation list

```
usage: gitlab group-invitation list [-h] [--sudo SUDO] --group-id GROUP_ID
                                     [--query QUERY] [--page PAGE]
                                     [--per-page PER_PAGE] [--get-all]
```

```
-h, --help
    show this help message and exit

--sudo <sudo>

--group-id <group_id>

--query <query>

--page <page>

--per-page <per_page>

--get-all
    Return all items from the server, without pagination.
```

### gitlab group-invitation update

```
usage: gitlab group-invitation update [-h] [--sudo SUDO] --group-id GROUP_ID
                                       --email EMAIL
                                       [--access-level ACCESS_LEVEL]
                                       [--expires-at EXPIRES_AT]
```

```
-h, --help
    show this help message and exit

--sudo <sudo>

--group-id <group_id>

--email <email>

--access-level <access_level>

--expires-at <expires_at>
```



## gitlab group-issue

```
usage: gitlab group-issue [-h] {list} ...
```

### -h, --help

show this help message and exit

## gitlab group-issue list

```
usage: gitlab group-issue list [-h] [--sudo SUDO] --group-id GROUP_ID
                                [--state STATE] [--labels LABELS]
                                [--milestone MILESTONE] [--order-by ORDER_BY]
                                [--sort SORT] [--iids IIDS]
                                [--author-id AUTHOR_ID]
                                [--iteration-id ITERATION_ID]
                                [--assignee-id ASSIGNEE_ID]
                                [--my-reaction-emoji MY_REACTION_EMOJI]
                                [--search SEARCH]
                                [--created-after CREATED_AFTER]
                                [--created-before CREATED_BEFORE]
                                [--updated-after UPDATED_AFTER]
                                [--updated-before UPDATED_BEFORE] [--page PAGE]
                                [--per-page PER_PAGE] [--get-all]
```

### -h, --help

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--state** <state>

**--labels** <labels>

**--milestone** <milestone>

**--order-by** <order\_by>

**--sort** <sort>

**--iids** <iids>

**--author-id** <author\_id>

**--iteration-id** <iteration\_id>

**--assignee-id** <assignee\_id>

**--my-reaction-emoji** <my\_reaction\_emoji>

**--search** <search>

**--created-after** <created\_after>

**--created-before** <created\_before>  
**--updated-after** <updated\_after>  
**--updated-before** <updated\_before>  
**--page** <page>  
**--per-page** <per\_page>  
**--get-all**  
Return all items from the server, without pagination.

### gitlab group-issues-statistics

```
usage: gitlab group-issues-statistics [-h] {get} ...
```

**-h, --help**  
show this help message and exit

### gitlab group-issues-statistics get

```
usage: gitlab group-issues-statistics get [-h] [--sudo SUDO] --group-id
                                           GROUP_ID
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--group-id** <group\_id>

### gitlab group-iteration

```
usage: gitlab group-iteration [-h] {list} ...
```

**-h, --help**  
show this help message and exit

### gitlab group-iteration list

```
usage: gitlab group-iteration list [-h] [--sudo SUDO] --group-id GROUP_ID
                                   [--state STATE] [--search SEARCH]
                                   [--include-ancestors INCLUDE_ANCESTORS]
                                   [--page PAGE] [--per-page PER_PAGE]
                                   [--get-all]
```

**-h, --help**  
show this help message and exit

```

--sudo <sudo>
--group-id <group_id>
--state <state>
--search <search>
--include-ancestors <include_ancestors>
--page <page>
--per-page <per_page>
--get-all
    Return all items from the server, without pagination.

```

### gitlab group-label

```

usage: gitlab group-label [-h]
    list
    get
    create
    update
    delete
    subscribe
    unsubscribe
    ...

```

```

-h, --help
    show this help message and exit

```

### gitlab group-label create

```

usage: gitlab group-label create [-h] [--sudo SUDO] --group-id GROUP_ID --name
                                NAME --color COLOR
                                [--description DESCRIPTION]
                                [--priority PRIORITY]

```

```

-h, --help
    show this help message and exit

--sudo <sudo>
--group-id <group_id>
--name <name>
--color <color>
--description <description>
--priority <priority>

```

### gitlab group-label delete

```
usage: gitlab group-label delete [-h] [--sudo SUDO] --group-id GROUP_ID --name
                                NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--name** <name>

### gitlab group-label get

```
usage: gitlab group-label get [-h] [--sudo SUDO] --group-id GROUP_ID --name
                                NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--name** <name>

### gitlab group-label list

```
usage: gitlab group-label list [-h] [--sudo SUDO] --group-id GROUP_ID
                                [--page PAGE] [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab group-label subscribe

```
usage: gitlab group-label subscribe [-h] --group-id GROUP_ID [--sudo SUDO]
                                   --name NAME
```

**-h, --help**  
show this help message and exit

**--group-id** <group\_id>

**--sudo** <sudo>

**--name** <name>

### gitlab group-label unsubscribe

```
usage: gitlab group-label unsubscribe [-h] --group-id GROUP_ID [--sudo SUDO]
                                   --name NAME
```

**-h, --help**  
show this help message and exit

**--group-id** <group\_id>

**--sudo** <sudo>

**--name** <name>

### gitlab group-label update

```
usage: gitlab group-label update [-h] [--sudo SUDO] --group-id GROUP_ID --name
                                   NAME [--new-name NEW_NAME] [--color COLOR]
                                   [--description DESCRIPTION]
                                   [--priority PRIORITY]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--name** <name>

**--new-name** <new\_name>

**--color** <color>

**--description** <description>

**--priority** <priority>

## gitlab group-ldap-group-link

```
usage: gitlab group-ldap-group-link [-h] {list,create,delete} ...
```

**-h, --help**

show this help message and exit

## gitlab group-ldap-group-link create

```
usage: gitlab group-ldap-group-link create [-h] [--sudo SUDO] --group-id
                                           GROUP_ID --provider PROVIDER
                                           --group-access GROUP_ACCESS
                                           [--cn CN | --filter FILTER]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--provider** <provider>

**--group-access** <group\_access>

**--cn** <cn>

**--filter** <filter>

## gitlab group-ldap-group-link delete

```
usage: gitlab group-ldap-group-link delete [-h] [--sudo SUDO] --group-id
                                           GROUP_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

## gitlab group-ldap-group-link list

```
usage: gitlab group-ldap-group-link list [-h] [--sudo SUDO] --group-id
                                           GROUP_ID [--page PAGE]
                                           [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab group-member

```
usage: gitlab group-member [-h] {list,get,create,update,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab group-member create

```
usage: gitlab group-member create [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --access-level ACCESS_LEVEL --user-id
                                     USER_ID [--expires-at EXPIRES_AT]
                                     [--tasks-to-be-done TASKS_TO_BE_DONE]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--access-level** <access\_level>

**--user-id** <user\_id>

**--expires-at** <expires\_at>

**--tasks-to-be-done** <tasks\_to\_be\_done>

### gitlab group-member delete

```
usage: gitlab group-member delete [-h] [--sudo SUDO] --group-id GROUP_ID --id
                                     ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

### gitlab group-member get

```
usage: gitlab group-member get [-h] [--sudo SUDO] --group-id GROUP_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

### gitlab group-member list

```
usage: gitlab group-member list [-h] [--sudo SUDO] --group-id GROUP_ID  
                                [--page PAGE] [--per-page PER_PAGE]  
                                [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab group-member update

```
usage: gitlab group-member update [-h] [--sudo SUDO] --group-id GROUP_ID --id  
ID --access-level ACCESS_LEVEL  
[--expires-at EXPIRES_AT]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

**--access-level** <access\_level>

**--expires-at** <expires\_at>



## gitlab group-member-all

```
usage: gitlab group-member-all [-h] {list,get} ...
```

**-h, --help**

show this help message and exit

## gitlab group-member-all get

```
usage: gitlab group-member-all get [-h] [--sudo SUDO] --group-id GROUP_ID --id
                                     ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

## gitlab group-member-all list

```
usage: gitlab group-member-all list [-h] [--sudo SUDO] --group-id GROUP_ID
                                     [--page PAGE] [--per-page PER_PAGE]
                                     [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab group-merge-request

```
usage: gitlab group-merge-request [-h] {list} ...
```

**-h, --help**

show this help message and exit

## gitlab group-merge-request list

```
usage: gitlab group-merge-request list [-h] [--sudo SUDO] --group-id GROUP_ID
                                         [--state STATE] [--order-by ORDER_BY]
                                         [--sort SORT] [--milestone MILESTONE]
                                         [--view VIEW] [--labels LABELS]
                                         [--created-after CREATED_AFTER]
                                         [--created-before CREATED_BEFORE]
                                         [--updated-after UPDATED_AFTER]
                                         [--updated-before UPDATED_BEFORE]
                                         [--scope SCOPE] [--author-id AUTHOR_ID]
                                         [--assignee-id ASSIGNEE_ID]
                                         [--approver-ids APPROVER_IDS]
                                         [--approved-by-ids APPROVED_BY_IDS]
                                         [--my-reaction-emoji MY_REACTION_EMOJI]
                                         [--source-branch SOURCE_BRANCH]
                                         [--target-branch TARGET_BRANCH]
                                         [--search SEARCH] [--wip WIP]
                                         [--page PAGE] [--per-page PER_PAGE]
                                         [--get-all]
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--state** <state>

**--order-by** <order\_by>

**--sort** <sort>

**--milestone** <milestone>

**--view** <view>

**--labels** <labels>

**--created-after** <created\_after>

**--created-before** <created\_before>

**--updated-after** <updated\_after>

**--updated-before** <updated\_before>

**--scope** <scope>

**--author-id** <author\_id>

**--assignee-id** <assignee\_id>

**--approver-ids** <approver\_ids>

**--approved-by-ids** <approved\_by\_ids>

```

--my-reaction-emoji <my_reaction_emoji>
--source-branch <source_branch>
--target-branch <target_branch>
--search <search>
--wip <wip>
--page <page>
--per-page <per_page>
--get-all
    Return all items from the server, without pagination.

```

### gitlab group-milestone

```

usage: gitlab group-milestone [-h]
    list
    get
    create
    update
    delete
    issues
    merge-requests
    ...

```

```

-h, --help
    show this help message and exit

```

### gitlab group-milestone create

```

usage: gitlab group-milestone create [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --title TITLE [--description DESCRIPTION]
                                     [--due-date DUE_DATE]
                                     [--start-date START_DATE]

```

```

-h, --help
    show this help message and exit
--sudo <sudo>
--group-id <group_id>
--title <title>
--description <description>
--due-date <due_date>
--start-date <start_date>

```

### gitlab group-milestone delete

```
usage: gitlab group-milestone delete [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

### gitlab group-milestone get

```
usage: gitlab group-milestone get [-h] [--sudo SUDO] --group-id GROUP_ID --id
                                   ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

### gitlab group-milestone issues

```
usage: gitlab group-milestone issues [-h] --group-id GROUP_ID [--sudo SUDO]
                                     --id ID
```

**-h, --help**  
show this help message and exit

**--group-id** <group\_id>

**--sudo** <sudo>

**--id** <id>

### gitlab group-milestone list

```
usage: gitlab group-milestone list [-h] [--sudo SUDO] --group-id GROUP_ID
                                   [--iids IIDS] [--state STATE]
                                   [--search SEARCH] [--page PAGE]
                                   [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--iids** <iids>

**--state** <state>

**--search** <search>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab group-milestone merge-requests

```
usage: gitlab group-milestone merge-requests [-h] --group-id GROUP_ID
                                              [--sudo SUDO] --id ID
```

**-h, --help**

show this help message and exit

**--group-id** <group\_id>

**--sudo** <sudo>

**--id** <id>

### gitlab group-milestone update

```
usage: gitlab group-milestone update [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --id ID [--title TITLE]
                                     [--description DESCRIPTION]
                                     [--due-date DUE_DATE]
                                     [--start-date START_DATE]
                                     [--state-event STATE_EVENT]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--id** <id>

**--title** <title>

**--description** <description>

**--due-date** <due\_date>

**--start-date** <start\_date>

**--state-event** <state\_event>

## gitlab group-notification-settings

```
usage: gitlab group-notification-settings [-h] {get,update} ...
```

**-h, --help**

show this help message and exit

## gitlab group-notification-settings get

```
usage: gitlab group-notification-settings get [-h] [--sudo SUDO] --group-id
GROUP_ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

## gitlab group-notification-settings update

```
usage: gitlab group-notification-settings update [-h] [--sudo SUDO] --group-id
GROUP_ID [--level LEVEL]
[--notification-email NOTIFICATION_
↪EMAIL]
[--new-note NEW_NOTE]
[--new-issue NEW_ISSUE]
[--reopen-issue REOPEN_ISSUE]
[--close-issue CLOSE_ISSUE]
[--reassign-issue REASSIGN_ISSUE]
[--new-merge-request NEW_MERGE_REQUEST]
[--reopen-merge-request REOPEN_MERGE_
↪REQUEST]
[--close-merge-request CLOSE_MERGE_
↪REQUEST]
[--reassign-merge-request REASSIGN_
↪MERGE_REQUEST]
[--merge-merge-request MERGE_MERGE_
↪REQUEST]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

```

--level <level>
--notification-email <notification_email>
--new-note <new_note>
--new-issue <new_issue>
--reopen-issue <reopen_issue>
--close-issue <close_issue>
--reassign-issue <reassign_issue>
--new-merge-request <new_merge_request>
--reopen-merge-request <reopen_merge_request>
--close-merge-request <close_merge_request>
--reassign-merge-request <reassign_merge_request>
--merge-merge-request <merge_merge_request>

```

## gitlab group-package

```
usage: gitlab group-package [-h] {list} ...
```

**-h, --help**  
show this help message and exit

## gitlab group-package list

```
usage: gitlab group-package list [-h] [--sudo SUDO] --group-id GROUP_ID
                                [--exclude-subgroups EXCLUDE_SUBGROUPS]
                                [--order-by ORDER_BY] [--sort SORT]
                                [--package-type PACKAGE_TYPE]
                                [--package-name PACKAGE_NAME] [--page PAGE]
                                [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

```

--sudo <sudo>
--group-id <group_id>
--exclude-subgroups <exclude_subgroups>
--order-by <order_by>
--sort <sort>
--package-type <package_type>

```

**--package-name** <package\_name>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab group-project

```
usage: gitlab group-project [-h] {list} ...
```

**-h, --help**

show this help message and exit

## gitlab group-project list

```
usage: gitlab group-project list [-h] [--sudo SUDO] --group-id GROUP_ID
                                [--archived ARCHIVED]
                                [--visibility VISIBILITY]
                                [--order-by ORDER_BY] [--sort SORT]
                                [--search SEARCH] [--simple SIMPLE]
                                [--owned OWNED] [--starred STARRED]
                                [--with-custom-attributes WITH_CUSTOM_ATTRIBUTES]
                                [--include-subgroups INCLUDE_SUBGROUPS]
                                [--with-issues-enabled WITH_ISSUES_ENABLED]
                                --with-merge-requests-enabled WITH_MERGE_REQUESTS_
↳ENABLED]
                                [--with-shared WITH_SHARED]
                                [--min-access-level MIN_ACCESS_LEVEL]
                                [--with-security-reports WITH_SECURITY_REPORTS]
                                [--page PAGE] [--per-page PER_PAGE]
                                [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--archived** <archived>

**--visibility** <visibility>

**--order-by** <order\_by>

**--sort** <sort>

**--search** <search>

**--simple** <simple>



```

--owned <owned>
--starred <starred>
--with-custom-attributes <with_custom_attributes>
--include-subgroups <include_subgroups>
--with-issues-enabled <with_issues_enabled>
--with-merge-requests-enabled <with_merge_requests_enabled>
--with-shared <with_shared>
--min-access-level <min_access_level>
--with-security-reports <with_security_reports>
--page <page>
--per-page <per_page>
--get-all

```

Return all items from the server, without pagination.

## gitlab group-push-rules

```
usage: gitlab group-push-rules [-h] {get,create,update,delete} ...
```

**-h, --help**

show this help message and exit

## gitlab group-push-rules create

```

usage: gitlab group-push-rules create [-h] [--sudo SUDO] --group-id GROUP_ID
                                     [--deny-delete-tag DENY_DELETE_TAG]
                                     [--member-check MEMBER_CHECK]
                                     [--prevent-secrets PREVENT_SECRETS]
                                     [--commit-message-regex COMMIT_MESSAGE_REGEX]
                                     [--commit-message-negative-regex COMMIT_MESSAGE_
↳NEGATIVE_REGEX]
                                     [--branch-name-regex BRANCH_NAME_REGEX]
                                     [--author-email-regex AUTHOR_EMAIL_REGEX]
                                     [--file-name-regex FILE_NAME_REGEX]
                                     [--max-file-size MAX_FILE_SIZE]
                                     [--commit-committer-check COMMIT_COMMITTER_CHECK]
                                     [--reject-unsigned-commits REJECT_UNSIGNED_COMMITS]

```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

```
--deny-delete-tag <deny_delete_tag>
--member-check <member_check>
--prevent-secrets <prevent_secrets>
--commit-message-regex <commit_message_regex>
--commit-message-negative-regex <commit_message_negative_regex>
--branch-name-regex <branch_name_regex>
--author-email-regex <author_email_regex>
--file-name-regex <file_name_regex>
--max-file-size <max_file_size>
--commit-committer-check <commit_committer_check>
--reject-unsigned-commits <reject_unsigned_commits>
```

### gitlab group-push-rules delete

```
usage: gitlab group-push-rules delete [-h] [--sudo SUDO] --group-id GROUP_ID
```

```
-h, --help
    show this help message and exit
--sudo <sudo>
--group-id <group_id>
```

### gitlab group-push-rules get

```
usage: gitlab group-push-rules get [-h] [--sudo SUDO] --group-id GROUP_ID
```

```
-h, --help
    show this help message and exit
--sudo <sudo>
--group-id <group_id>
```

### gitlab group-push-rules update

```
usage: gitlab group-push-rules update [-h] [--sudo SUDO] --group-id GROUP_ID
                                     [--deny-delete-tag DENY_DELETE_TAG]
                                     [--member-check MEMBER_CHECK]
                                     [--prevent-secrets PREVENT_SECRETS]
                                     [--commit-message-regex COMMIT_MESSAGE_REGEX]
                                     [--commit-message-negative-regex COMMIT_MESSAGE_
↪NEGATIVE_REGEX]
```

(continues on next page)

(continued from previous page)

```

[--branch-name-regex BRANCH_NAME_REGEX]
[--author-email-regex AUTHOR_EMAIL_REGEX]
[--file-name-regex FILE_NAME_REGEX]
[--max-file-size MAX_FILE_SIZE]
[--commit-committer-check COMMIT_COMMITTER_CHECK]
[--reject-unsigned-commits REJECT_UNSIGNED_COMMITS]

```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--group-id** <group\_id>**--deny-delete-tag** <deny\_delete\_tag>**--member-check** <member\_check>**--prevent-secrets** <prevent\_secrets>**--commit-message-regex** <commit\_message\_regex>**--commit-message-negative-regex** <commit\_message\_negative\_regex>**--branch-name-regex** <branch\_name\_regex>**--author-email-regex** <author\_email\_regex>**--file-name-regex** <file\_name\_regex>**--max-file-size** <max\_file\_size>**--commit-committer-check** <commit\_committer\_check>**--reject-unsigned-commits** <reject\_unsigned\_commits>

## gitlab group-runner

```
usage: gitlab group-runner [-h] {list} ...
```

**-h, --help**

show this help message and exit

## gitlab group-runner list

```
usage: gitlab group-runner list [-h] [--sudo SUDO] --group-id GROUP_ID
                                [--scope SCOPE] [--tag-list TAG_LIST]
                                [--page PAGE] [--per-page PER_PAGE]
                                [--get-all]

```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--scope** <scope>

**--tag-list** <tag\_list>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab group-saml-group-link

```
usage: gitlab group-saml-group-link [-h] {list,get,create,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab group-saml-group-link create

```
usage: gitlab group-saml-group-link create [-h] [--sudo SUDO] --group-id
                                           GROUP_ID --saml-group-name
                                           SAML_GROUP_NAME --access-level
                                           ACCESS_LEVEL
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--saml-group-name** <saml\_group\_name>

**--access-level** <access\_level>

### gitlab group-saml-group-link delete

```
usage: gitlab group-saml-group-link delete [-h] [--sudo SUDO] --group-id
                                           GROUP_ID --name NAME
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--name** <name>

### gitlab group-saml-group-link get

```
usage: gitlab group-saml-group-link get [-h] [--sudo SUDO] --group-id GROUP_ID
                                         --name NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--name** <name>

### gitlab group-saml-group-link list

```
usage: gitlab group-saml-group-link list [-h] [--sudo SUDO] --group-id
                                         GROUP_ID [--page PAGE]
                                         [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab group-subgroup

```
usage: gitlab group-subgroup [-h] {list} ...
```

**-h, --help**  
show this help message and exit

### gitlab group-subgroup list

```
usage: gitlab group-subgroup list [-h] [--sudo SUDO] --group-id GROUP_ID
                                   [--skip-groups SKIP_GROUPS]
                                   [--all-available ALL_AVAILABLE]
                                   [--search SEARCH] [--order-by ORDER_BY]
                                   [--sort SORT] [--statistics STATISTICS]
                                   [--owned OWNED]
                                   [--with-custom-attributes WITH_CUSTOM_ATTRIBUTES]
                                   [--min-access-level MIN_ACCESS_LEVEL]
```

(continues on next page)

(continued from previous page)

```
[--page PAGE] [--per-page PER_PAGE]
[--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--skip-groups** <skip\_groups>

**--all-available** <all\_available>

**--search** <search>

**--order-by** <order\_by>

**--sort** <sort>

**--statistics** <statistics>

**--owned** <owned>

**--with-custom-attributes** <with\_custom\_attributes>

**--min-access-level** <min\_access\_level>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

## gitlab group-variable

```
usage: gitlab group-variable [-h] {list,get,create,update,delete} ...
```

**-h, --help**  
show this help message and exit

## gitlab group-variable create

```
usage: gitlab group-variable create [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --key KEY --value VALUE
                                     [--protected PROTECTED]
                                     [--variable-type VARIABLE_TYPE]
                                     [--masked MASKED]
```

**-h, --help**  
show this help message and exit

```

--sudo <sudo>
--group-id <group_id>
--key <key>
--value <value>
--protected <protected>
--variable-type <variable_type>
--masked <masked>

```

### gitlab group-variable delete

```
usage: gitlab group-variable delete [-h] [--sudo SUDO] --group-id GROUP_ID
                                   --key KEY

```

```

-h, --help
    show this help message and exit
--sudo <sudo>
--group-id <group_id>
--key <key>

```

### gitlab group-variable get

```
usage: gitlab group-variable get [-h] [--sudo SUDO] --group-id GROUP_ID --key
                                KEY

```

```

-h, --help
    show this help message and exit
--sudo <sudo>
--group-id <group_id>
--key <key>

```

### gitlab group-variable list

```
usage: gitlab group-variable list [-h] [--sudo SUDO] --group-id GROUP_ID
                                   [--page PAGE] [--per-page PER_PAGE]
                                   [--get-all]

```

```

-h, --help
    show this help message and exit
--sudo <sudo>

```

**--group-id** <group\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab group-variable update

```
usage: gitlab group-variable update [-h] [--sudo SUDO] --group-id GROUP_ID
                                   --key KEY --value VALUE
                                   [--protected PROTECTED]
                                   [--variable-type VARIABLE_TYPE]
                                   [--masked MASKED]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--key** <key>

**--value** <value>

**--protected** <protected>

**--variable-type** <variable\_type>

**--masked** <masked>

### gitlab group-wiki

```
usage: gitlab group-wiki [-h] {list,get,create,update,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab group-wiki create

```
usage: gitlab group-wiki create [-h] [--sudo SUDO] --group-id GROUP_ID --title
                                TITLE --content CONTENT [--format FORMAT]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>



**--title** <title>  
**--content** <content>  
**--format** <format>

### gitlab group-wiki delete

```
usage: gitlab group-wiki delete [-h] [--sudo SUDO] --group-id GROUP_ID --slug
                                SLUG
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--group-id** <group\_id>  
**--slug** <slug>

### gitlab group-wiki get

```
usage: gitlab group-wiki get [-h] [--sudo SUDO] --group-id GROUP_ID --slug
                              SLUG
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--group-id** <group\_id>  
**--slug** <slug>

### gitlab group-wiki list

```
usage: gitlab group-wiki list [-h] [--sudo SUDO] --group-id GROUP_ID
                               [--with-content WITH_CONTENT] [--page PAGE]
                               [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--group-id** <group\_id>  
**--with-content** <with\_content>  
**--page** <page>  
**--per-page** <per\_page>  
**--get-all**  
Return all items from the server, without pagination.

### gitlab group-wiki update

```
usage: gitlab group-wiki update [-h] [--sudo SUDO] --group-id GROUP_ID --slug
                                SLUG [--title TITLE] [--content CONTENT]
                                [--format FORMAT]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--slug** <slug>

**--title** <title>

**--content** <content>

**--format** <format>

### gitlab hook

```
usage: gitlab hook [-h] {list,get,create,delete} ...
```

**-h, --help**  
show this help message and exit

### gitlab hook create

```
usage: gitlab hook create [-h] [--sudo SUDO] --url URL
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--url** <url>

### gitlab hook delete

```
usage: gitlab hook delete [-h] [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--id** <id>

### gitlab hook get

```
usage: gitlab hook get [-h] [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--id** <id>

### gitlab hook list

```
usage: gitlab hook list [-h] [--sudo SUDO] [--page PAGE] [--per-page PER_PAGE]
                        [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab issue

```
usage: gitlab issue [-h] {list,get} ...
```

**-h, --help**  
show this help message and exit

### gitlab issue get

```
usage: gitlab issue get [-h] [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--id** <id>

## gitlab issue list

```
usage: gitlab issue list [-h] [--sudo SUDO] [--state STATE] [--labels LABELS]
                        [--milestone MILESTONE] [--scope SCOPE]
                        [--author-id AUTHOR_ID] [--iteration-id ITERATION_ID]
                        [--assignee-id ASSIGNEE_ID]
                        [--my-reaction-emoji MY_REACTION_EMOJI] [--iids IIDS]
                        [--order-by ORDER_BY] [--sort SORT] [--search SEARCH]
                        [--created-after CREATED_AFTER]
                        [--created-before CREATED_BEFORE]
                        [--updated-after UPDATED_AFTER]
                        [--updated-before UPDATED_BEFORE] [--page PAGE]
                        [--per-page PER_PAGE] [--get-all]
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--state** <state>

**--labels** <labels>

**--milestone** <milestone>

**--scope** <scope>

**--author-id** <author\_id>

**--iteration-id** <iteration\_id>

**--assignee-id** <assignee\_id>

**--my-reaction-emoji** <my\_reaction\_emoji>

**--iids** <iids>

**--order-by** <order\_by>

**--sort** <sort>

**--search** <search>

**--created-after** <created\_after>

**--created-before** <created\_before>

**--updated-after** <updated\_after>

**--updated-before** <updated\_before>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab issues-statistics

```
usage: gitlab issues-statistics [-h] {get} ...
```

**-h, --help**

show this help message and exit

### gitlab issues-statistics get

```
usage: gitlab issues-statistics get [-h] [--sudo SUDO]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

### gitlab key

```
usage: gitlab key [-h] {get} ...
```

**-h, --help**

show this help message and exit

### gitlab key get

```
usage: gitlab key get [-h] [--sudo SUDO] --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>

### gitlab ldap-group

```
usage: gitlab ldap-group [-h] {list} ...
```

**-h, --help**

show this help message and exit

## gitlab ldap-group list

```
usage: gitlab ldap-group list [-h] [--sudo SUDO] [--search SEARCH]
                             [--provider PROVIDER] [--page PAGE]
                             [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--search** <search>

**--provider** <provider>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab license

```
usage: gitlab license [-h] {list,get} ...
```

**-h, --help**

show this help message and exit

## gitlab license get

```
usage: gitlab license get [-h] [--sudo SUDO] --key KEY [--project PROJECT]
                          [--fullname FULLNAME]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--key** <key>

**--project** <project>

**--fullname** <fullname>

## gitlab license list

```
usage: gitlab license list [-h] [--sudo SUDO] [--popular POPULAR]
                          [--page PAGE] [--per-page PER_PAGE] [--get-all]
```

### -h, --help

show this help message and exit

### --sudo <sudo>

### --popular <popular>

### --page <page>

### --per-page <per\_page>

### --get-all

Return all items from the server, without pagination.

## gitlab merge-request

```
usage: gitlab merge-request [-h] {list} ...
```

### -h, --help

show this help message and exit

## gitlab merge-request list

```
usage: gitlab merge-request list [-h] [--sudo SUDO] [--state STATE]
                                [--order-by ORDER_BY] [--sort SORT]
                                [--milestone MILESTONE] [--view VIEW]
                                [--labels LABELS]
                                [--with-labels-details WITH_LABELS_DETAILS]
                                [--with-merge-status-recheck WITH_MERGE_STATUS_RECHECK]
                                [--created-after CREATED_AFTER]
                                [--created-before CREATED_BEFORE]
                                [--updated-after UPDATED_AFTER]
                                [--updated-before UPDATED_BEFORE]
                                [--scope SCOPE] [--author-id AUTHOR_ID]
                                [--author-username AUTHOR_USERNAME]
                                [--assignee-id ASSIGNEE_ID]
                                [--approver-ids APPROVER_IDS]
                                [--approved-by-ids APPROVED_BY_IDS]
                                [--reviewer-id REVIEWER_ID]
                                [--reviewer-username REVIEWER_USERNAME]
                                [--my-reaction-emoji MY_REACTION_EMOJI]
                                [--source-branch SOURCE_BRANCH]
                                [--target-branch TARGET_BRANCH]
                                [--search SEARCH] [--in IN] [--wip WIP]
                                [--not NOT] [--environment ENVIRONMENT]
                                [--deployed-before DEPLOYED_BEFORE]
                                [--deployed-after DEPLOYED_AFTER]
```

(continues on next page)

(continued from previous page)

```
[--page PAGE] [--per-page PER_PAGE]
[--get-all]
```

```
-h, --help
    show this help message and exit

--sudo <sudo>

--state <state>

--order-by <order_by>

--sort <sort>

--milestone <milestone>

--view <view>

--labels <labels>

--with-labels-details <with_labels_details>

--with-merge-status-recheck <with_merge_status_recheck>

--created-after <created_after>

--created-before <created_before>

--updated-after <updated_after>

--updated-before <updated_before>

--scope <scope>

--author-id <author_id>

--author-username <author_username>

--assignee-id <assignee_id>

--approver-ids <approver_ids>

--approved-by-ids <approved_by_ids>

--reviewer-id <reviewer_id>

--reviewer-username <reviewer_username>

--my-reaction-emoji <my_reaction_emoji>

--source-branch <source_branch>

--target-branch <target_branch>

--search <search>

--in <in>

--wip <wip>
```



```

--not <not>
--environment <environment>
--deployed-before <deployed_before>
--deployed-after <deployed_after>
--page <page>
--per-page <per_page>
--get-all
    Return all items from the server, without pagination.

```

## gitlab namespace

```
usage: gitlab namespace [-h] {list,get,exists} ...
```

```

-h, --help
    show this help message and exit

```

## gitlab namespace exists

```
usage: gitlab namespace exists [-h] --id ID --namespace NAMESPACE --parent-id
                                PARENT_ID
```

```

-h, --help
    show this help message and exit

--id <id>
--namespace <namespace>
--parent-id <parent_id>

```

## gitlab namespace get

```
usage: gitlab namespace get [-h] [--sudo SUDO] --id ID
```

```

-h, --help
    show this help message and exit

--sudo <sudo>
--id <id>

```

## gitlab namespace list

```
usage: gitlab namespace list [-h] [--sudo SUDO] [--search SEARCH]
                             [--page PAGE] [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--search** <search>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab notification-settings

```
usage: gitlab notification-settings [-h] {get,update} ...
```

**-h, --help**

show this help message and exit

## gitlab notification-settings get

```
usage: gitlab notification-settings get [-h] [--sudo SUDO]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

## gitlab notification-settings update

```
usage: gitlab notification-settings update [-h] [--sudo SUDO] [--level LEVEL]
                                           [--notification-email NOTIFICATION_EMAIL]
                                           [--new-note NEW_NOTE]
                                           [--new-issue NEW_ISSUE]
                                           [--reopen-issue REOPEN_ISSUE]
                                           [--close-issue CLOSE_ISSUE]
                                           [--reassign-issue REASSIGN_ISSUE]
                                           [--new-merge-request NEW_MERGE_REQUEST]
                                           [--reopen-merge-request REOPEN_MERGE_REQUEST]
                                           [--close-merge-request CLOSE_MERGE_REQUEST]
                                           [--reassign-merge-request REASSIGN_MERGE_
↩REQUEST]
                                           [--merge-merge-request MERGE_MERGE_REQUEST]
```

```

-h, --help
    show this help message and exit

--sudo <sudo>

--level <level>

--notification-email <notification_email>

--new-note <new_note>

--new-issue <new_issue>

--reopen-issue <reopen_issue>

--close-issue <close_issue>

--reassign-issue <reassign_issue>

--new-merge-request <new_merge_request>

--reopen-merge-request <reopen_merge_request>

--close-merge-request <close_merge_request>

--reassign-merge-request <reassign_merge_request>

--merge-merge-request <merge_merge_request>

```

## gitlab pages-domain

```
usage: gitlab pages-domain [-h] {list} ...
```

```

-h, --help
    show this help message and exit

```

## gitlab pages-domain list

```
usage: gitlab pages-domain list [-h] [--sudo SUDO] [--page PAGE]
                                [--per-page PER_PAGE] [--get-all]
```

```

-h, --help
    show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
    Return all items from the server, without pagination.

```

## gitlab personal-access-token

```
usage: gitlab personal-access-token [-h] {list,get,delete} ...
```

**-h, --help**

show this help message and exit

## gitlab personal-access-token delete

```
usage: gitlab personal-access-token delete [-h] [--sudo SUDO] --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>

## gitlab personal-access-token get

```
usage: gitlab personal-access-token get [-h] [--sudo SUDO] --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>

## gitlab personal-access-token list

```
usage: gitlab personal-access-token list [-h] [--sudo SUDO]
                                         [--user-id USER_ID] [--page PAGE]
                                         [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab project

```
usage: gitlab project [-h]
  list
  get
  create
  update
  delete
  upload
  update-submodule
  repository-tree
  repository-blob
  repository-raw-blob
  repository-compare
  repository-contributors
  repository-archive
  repository-merge-base
  delete-merged-branches
  create-fork-relation
  delete-fork-relation
  languages
  star
  unstar
  archive
  unarchive
  share
  unshare
  trigger-pipeline
  housekeeping
  restore
  snapshot
  search
  mirror-pull
  mirror-pull-details
  transfer
  ...
```

### **-h, --help**

show this help message and exit

## gitlab project archive

```
usage: gitlab project archive [-h] --id ID
```

### **-h, --help**

show this help message and exit

**--id** <id>

## gitlab project create

```
usage: gitlab project create [-h] [--sudo SUDO] [--name NAME] [--path PATH]
                             [--allow-merge-on-skipped-pipeline ALLOW_MERGE_ON_SKIPPED_
↳ PIPELINE]
                             [--only-allow-merge-if-all-status-checks-passed ONLY_ALLOW_
↳ MERGE_IF_ALL_STATUS_CHECKS_PASSED]
                             [--analytics-access-level ANALYTICS_ACCESS_LEVEL]
                             [--approvals-before-merge APPROVALS_BEFORE_MERGE]
                             [--auto-cancel-pending-pipelines AUTO_CANCEL_PENDING_
↳ PIPELINES]
                             [--auto-devops-deploy-strategy AUTO_DEVOPS_DEPLOY_STRATEGY]
                             [--auto-devops-enabled AUTO_DEVOPS_ENABLED]
                             [--autoclose-referenced-issues AUTOCLOSE_REFERENCED_ISSUES]
                             [--avatar AVATAR]
                             [--build-coverage-regex BUILD_COVERAGE_REGEX]
                             [--build-git-strategy BUILD_GIT_STRATEGY]
                             [--build-timeout BUILD_TIMEOUT]
                             [--builds-access-level BUILDS_ACCESS_LEVEL]
                             [--ci-config-path CI_CONFIG_PATH]
                             [--container-expiration-policy-attributes CONTAINER_
↳ EXPIRATION_POLICY_ATTRIBUTES]
                             [--container-registry-access-level CONTAINER_REGISTRY_
↳ ACCESS_LEVEL]
                             [--container-registry-enabled CONTAINER_REGISTRY_ENABLED]
                             [--default-branch DEFAULT_BRANCH]
                             [--description DESCRIPTION]
                             [--emails-disabled EMAILS_DISABLED]
                             [--external-authorization-classification-label EXTERNAL_
↳ AUTHORIZATION_CLASSIFICATION_LABEL]
                             [--forking-access-level FORKING_ACCESS_LEVEL]
                             [--group-with-project-templates-id GROUP_WITH_PROJECT_
↳ TEMPLATES_ID]
                             [--import-url IMPORT_URL]
                             [--initialize-with-readme INITIALIZE_WITH_README]
                             [--issues-access-level ISSUES_ACCESS_LEVEL]
                             [--issues-enabled ISSUES_ENABLED]
                             [--jobs-enabled JOBS_ENABLED]
                             [--lfs-enabled LFS_ENABLED]
                             [--merge-method MERGE_METHOD]
                             [--merge-pipelines-enabled MERGE_PIPELINES_ENABLED]
                             [--merge-requests-access-level MERGE_REQUESTS_ACCESS_LEVEL]
                             [--merge-requests-enabled MERGE_REQUESTS_ENABLED]
                             [--mirror-trigger-builds MIRROR_TRIGGER_BUILDS]
                             [--mirror MIRROR] [--namespace-id NAMESPACE_ID]
                             [--operations-access-level OPERATIONS_ACCESS_LEVEL]
                             [--only-allow-merge-if-all-discussions-are-resolved ONLY_
↳ ALLOW_MERGE_IF_ALL_DISCUSSIONS_ARE_RESOLVED]
                             [--only-allow-merge-if-pipeline-succeeds ONLY_ALLOW_MERGE_
↳ IF_PIPELINE_SUCCEEDS]
                             [--packages-enabled PACKAGES_ENABLED]
                             [--pages-access-level PAGES_ACCESS_LEVEL]
                             [--requirements-access-level REQUIREMENTS_ACCESS_LEVEL]
```

(continues on next page)

(continued from previous page)

```

↪REQUEST_LINK_ENABLED]      [--printing-merge-request-link-enabled PRINTING_MERGE_
                             [--public-builds PUBLIC_BUILDS]
                             [--releases-access-level RELEASES_ACCESS_LEVEL]
                             [--environments-access-level ENVIRONMENTS_ACCESS_LEVEL]
                             [--feature-flags-access-level FEATURE_FLAGS_ACCESS_LEVEL]
                             [--infrastructure-access-level INFRASTRUCTURE_ACCESS_LEVEL]
                             [--monitor-access-level MONITOR_ACCESS_LEVEL]
                             [--remove-source-branch-after-merge REMOVE_SOURCE_BRANCH_
↪AFTER_MERGE]               [--repository-access-level REPOSITORY_ACCESS_LEVEL]
                             [--repository-storage REPOSITORY_STORAGE]
                             [--request-access-enabled REQUEST_ACCESS_ENABLED]
                             [--resolve-outdated-diff-discussions RESOLVE_OUTDATED_DIFF_
↪DISCUSSIONS]               [--security-and-compliance-access-level SECURITY_AND_
↪COMPLIANCE_ACCESS_LEVEL]   [--shared-runners-enabled SHARED_RUNNERS_ENABLED]
                             [--show-default-award-emojis SHOW_DEFAULT_AWARD_EMOJIS]
                             [--snippets-access-level SNIPPETS_ACCESS_LEVEL]
                             [--snippets-enabled SNIPPETS_ENABLED]
                             [--squash-option SQUASH_OPTION]
                             [--tag-list TAG_LIST] [--topics TOPICS]
                             [--template-name TEMPLATE_NAME]
                             [--template-project-id TEMPLATE_PROJECT_ID]
                             [--use-custom-template USE_CUSTOM_TEMPLATE]
                             [--visibility VISIBILITY]
                             [--wiki-access-level WIKI_ACCESS_LEVEL]
                             [--wiki-enabled WIKI_ENABLED]

```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--name** <name>**--path** <path>**--allow-merge-on-skipped-pipeline** <allow\_merge\_on\_skipped\_pipeline>**--only-allow-merge-if-all-status-checks-passed** <only\_allow\_merge\_if\_all\_status\_checks\_passed>**--analytics-access-level** <analytics\_access\_level>**--approvals-before-merge** <approvals\_before\_merge>**--auto-cancel-pending-pipelines** <auto\_cancel\_pending\_pipelines>**--auto-devops-deploy-strategy** <auto\_devops\_deploy\_strategy>**--auto-devops-enabled** <auto\_devops\_enabled>**--autoclose-referenced-issues** <autoclose\_referenced\_issues>

```
--avatar <avatar>
--build-coverage-regex <build_coverage_regex>
--build-git-strategy <build_git_strategy>
--build-timeout <build_timeout>
--builds-access-level <builds_access_level>
--ci-config-path <ci_config_path>
--container-expiration-policy-attributes <container_expiration_policy_attributes>
--container-registry-access-level <container_registry_access_level>
--container-registry-enabled <container_registry_enabled>
--default-branch <default_branch>
--description <description>
--emails-disabled <emails_disabled>
--external-authorization-classification-label <external_authorization_classification_label>
--forking-access-level <forking_access_level>
--group-with-project-templates-id <group_with_project_templates_id>
--import-url <import_url>
--initialize-with-readme <initialize_with_readme>
--issues-access-level <issues_access_level>
--issues-enabled <issues_enabled>
--jobs-enabled <jobs_enabled>
--lfs-enabled <lfs_enabled>
--merge-method <merge_method>
--merge-pipelines-enabled <merge_pipelines_enabled>
--merge-requests-access-level <merge_requests_access_level>
--merge-requests-enabled <merge_requests_enabled>
--mirror-trigger-builds <mirror_trigger_builds>
--mirror <mirror>
--namespace-id <namespace_id>
--operations-access-level <operations_access_level>
--only-allow-merge-if-all-discussions-are-resolved <only_allow_merge_if_all_discussions_are_resolved>
--only-allow-merge-if-pipeline-succeeds <only_allow_merge_if_pipeline_succeeds>
```



```
--packages-enabled <packages_enabled>
--pages-access-level <pages_access_level>
--requirements-access-level <requirements_access_level>
--printing-merge-request-link-enabled <printing_merge_request_link_enabled>
--public-builds <public_builds>
--releases-access-level <releases_access_level>
--environments-access-level <environments_access_level>
--feature-flags-access-level <feature_flags_access_level>
--infrastructure-access-level <infrastructure_access_level>
--monitor-access-level <monitor_access_level>
--remove-source-branch-after-merge <remove_source_branch_after_merge>
--repository-access-level <repository_access_level>
--repository-storage <repository_storage>
--request-access-enabled <request_access_enabled>
--resolve-outdated-diff-discussions <resolve_outdated_diff_discussions>
--security-and-compliance-access-level <security_and_compliance_access_level>
--shared-runners-enabled <shared_runners_enabled>
--show-default-award-emojis <show_default_award_emojis>
--snippets-access-level <snippets_access_level>
--snippets-enabled <snippets_enabled>
--squash-option <squash_option>
--tag-list <tag_list>
--topics <topics>
--template-name <template_name>
--template-project-id <template_project_id>
--use-custom-template <use_custom_template>
--visibility <visibility>
--wiki-access-level <wiki_access_level>
--wiki-enabled <wiki_enabled>
```

### gitlab project create-fork-relation

```
usage: gitlab project create-fork-relation [-h] --id ID --forked-from-id
      FORKED_FROM_ID
```

**-h, --help**  
show this help message and exit

**--id <id>**

**--forked-from-id <forked\_from\_id>**

### gitlab project delete

```
usage: gitlab project delete [-h] [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--sudo <sudo>**

**--id <id>**

### gitlab project delete-fork-relation

```
usage: gitlab project delete-fork-relation [-h] --id ID
```

**-h, --help**  
show this help message and exit

**--id <id>**

### gitlab project delete-merged-branches

```
usage: gitlab project delete-merged-branches [-h] --id ID
```

**-h, --help**  
show this help message and exit

**--id <id>**

### gitlab project get

```
usage: gitlab project get [-h] [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--sudo <sudo>**

**--id <id>**

## gitlab project housekeeping

```
usage: gitlab project housekeeping [-h] --id ID
```

**-h, --help**

show this help message and exit

**--id** <id>

## gitlab project languages

```
usage: gitlab project languages [-h] --id ID
```

**-h, --help**

show this help message and exit

**--id** <id>

## gitlab project list

```
usage: gitlab project list [-h] [--sudo SUDO] [--archived ARCHIVED]
                           [--id-after ID_AFTER] [--id-before ID_BEFORE]
                           [--last-activity-after LAST_ACTIVITY_AFTER]
                           [--last-activity-before LAST_ACTIVITY_BEFORE]
                           [--membership MEMBERSHIP]
                           [--min-access-level MIN_ACCESS_LEVEL]
                           [--order-by ORDER_BY] [--owned OWNED]
                           [--repository-checksum-failed REPOSITORY_CHECKSUM_FAILED]
                           [--repository-storage REPOSITORY_STORAGE]
                           [--search-namespaces SEARCH_NAMESPACES]
                           [--search SEARCH] [--simple SIMPLE] [--sort SORT]
                           [--starred STARRED] [--statistics STATISTICS]
                           [--topic TOPIC] [--visibility VISIBILITY]
                           [--wiki-checksum-failed WIKI_CHECKSUM_FAILED]
                           [--with-custom-attributes WITH_CUSTOM_ATTRIBUTES]
                           [--with-issues-enabled WITH_ISSUES_ENABLED]
                           [--with-merge-requests-enabled WITH_MERGE_REQUESTS_ENABLED]
                           [--with-programming-language WITH_PROGRAMMING_LANGUAGE]
                           [--page PAGE] [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--archived** <archived>

**--id-after** <id\_after>

**--id-before** <id\_before>

**--last-activity-after** <last\_activity\_after>

**--last-activity-before** <last\_activity\_before>  
**--membership** <membership>  
**--min-access-level** <min\_access\_level>  
**--order-by** <order\_by>  
**--owned** <owned>  
**--repository-checksum-failed** <repository\_checksum\_failed>  
**--repository-storage** <repository\_storage>  
**--search-namespaces** <search\_namespaces>  
**--search** <search>  
**--simple** <simple>  
**--sort** <sort>  
**--starred** <starred>  
**--statistics** <statistics>  
**--topic** <topic>  
**--visibility** <visibility>  
**--wiki-checksum-failed** <wiki\_checksum\_failed>  
**--with-custom-attributes** <with\_custom\_attributes>  
**--with-issues-enabled** <with\_issues\_enabled>  
**--with-merge-requests-enabled** <with\_merge\_requests\_enabled>  
**--with-programming-language** <with\_programming\_language>  
**--page** <page>  
**--per-page** <per\_page>  
**--get-all**

Return all items from the server, without pagination.

### gitlab project mirror-pull

```
usage: gitlab project mirror-pull [-h] --id ID
```

**-h, --help**  
show this help message and exit  
**--id** <id>

### gitlab project mirror-pull-details

```
usage: gitlab project mirror-pull-details [-h] --id ID
```

**-h, --help**

show this help message and exit

**--id** <id>

### gitlab project repository-archive

```
usage: gitlab project repository-archive [-h] --id ID [--sha SHA]
                                         [--format FORMAT]
```

**-h, --help**

show this help message and exit

**--id** <id>

**--sha** <sha>

**--format** <format>

### gitlab project repository-blob

```
usage: gitlab project repository-blob [-h] --id ID --sha SHA
```

**-h, --help**

show this help message and exit

**--id** <id>

**--sha** <sha>

### gitlab project repository-compare

```
usage: gitlab project repository-compare [-h] --id ID --from- FROM_ --to TO
```

**-h, --help**

show this help message and exit

**--id** <id>

**--from-** <from\_>

**--to** <to>

### gitlab project repository-contributors

```
usage: gitlab project repository-contributors [-h] --id ID
```

**-h, --help**  
show this help message and exit

**--id** <id>

### gitlab project repository-merge-base

```
usage: gitlab project repository-merge-base [-h] --id ID --refs REFS
```

**-h, --help**  
show this help message and exit

**--id** <id>

**--refs** <refs>

### gitlab project repository-raw-blob

```
usage: gitlab project repository-raw-blob [-h] --id ID --sha SHA
```

**-h, --help**  
show this help message and exit

**--id** <id>

**--sha** <sha>

### gitlab project repository-tree

```
usage: gitlab project repository-tree [-h] --id ID [--path PATH] [--ref REF]
                                     [--recursive RECURSIVE]
```

**-h, --help**  
show this help message and exit

**--id** <id>

**--path** <path>

**--ref** <ref>

**--recursive** <recursive>

### gitlab project restore

```
usage: gitlab project restore [-h] --id ID
```

**-h, --help**  
show this help message and exit

**--id** <id>

### gitlab project search

```
usage: gitlab project search [-h] --id ID --scope SCOPE --search SEARCH
```

**-h, --help**  
show this help message and exit

**--id** <id>

**--scope** <scope>

**--search** <search>

### gitlab project share

```
usage: gitlab project share [-h] --id ID --group-id GROUP_ID --group-access  
GROUP_ACCESS [--expires-at EXPIRES_AT]
```

**-h, --help**  
show this help message and exit

**--id** <id>

**--group-id** <group\_id>

**--group-access** <group\_access>

**--expires-at** <expires\_at>

### gitlab project snapshot

```
usage: gitlab project snapshot [-h] --id ID [--wiki WIKI]
```

**-h, --help**  
show this help message and exit

**--id** <id>

**--wiki** <wiki>

### gitlab project star

```
usage: gitlab project star [-h] --id ID
```

**-h, --help**  
show this help message and exit

**--id <id>**

### gitlab project transfer

```
usage: gitlab project transfer [-h] --id ID --to-namespace TO_NAMESPACE
```

**-h, --help**  
show this help message and exit

**--id <id>**

**--to-namespace <to\_namespace>**

### gitlab project trigger-pipeline

```
usage: gitlab project trigger-pipeline [-h] --id ID --ref REF --token TOKEN
```

**-h, --help**  
show this help message and exit

**--id <id>**

**--ref <ref>**

**--token <token>**

### gitlab project unarchive

```
usage: gitlab project unarchive [-h] --id ID
```

**-h, --help**  
show this help message and exit

**--id <id>**



## gitlab project unshare

```
usage: gitlab project unshare [-h] --id ID --group-id GROUP_ID
```

**-h, --help**

show this help message and exit

**--id <id>**

**--group-id <group\_id>**

## gitlab project unstar

```
usage: gitlab project unstar [-h] --id ID
```

**-h, --help**

show this help message and exit

**--id <id>**

## gitlab project update

```
usage: gitlab project update [-h] [--sudo SUDO] --id ID
                                [--allow-merge-on-skipped-pipeline ALLOW_MERGE_ON_SKIPPED_
↪ PIPELINE]
                                [--only-allow-merge-if-all-status-checks-passed ONLY_ALLOW_
↪ MERGE_IF_ALL_STATUS_CHECKS_PASSED]
                                [--analytics-access-level ANALYTICS_ACCESS_LEVEL]
                                [--approvals-before-merge APPROVALS_BEFORE_MERGE]
                                [--auto-cancel-pending-pipelines AUTO_CANCEL_PENDING_
↪ PIPELINES]
                                [--auto-devops-deploy-strategy AUTO_DEVOPS_DEPLOY_STRATEGY]
                                [--auto-devops-enabled AUTO_DEVOPS_ENABLED]
                                [--autoclose-referenced-issues AUTOCLOSE_REFERENCED_ISSUES]
                                [--avatar AVATAR]
                                [--build-coverage-regex BUILD_COVERAGE_REGEX]
                                [--build-git-strategy BUILD_GIT_STRATEGY]
                                [--build-timeout BUILD_TIMEOUT]
                                [--builds-access-level BUILDS_ACCESS_LEVEL]
                                [--ci-config-path CI_CONFIG_PATH]
                                [--ci-default-git-depth CI_DEFAULT_GIT_DEPTH]
                                [--ci-forward-deployment-enabled CI_FORWARD_DEPLOYMENT_
↪ ENABLED]
                                [--ci-allow-fork-pipelines-to-run-in-parent-project CI_
↪ ALLOW_FORK_PIPELINES_TO_RUN_IN_PARENT_PROJECT]
                                [--ci-separated-caches CI_SEPARATED_CACHES]
                                [--container-expiration-policy-attributes CONTAINER_
↪ EXPIRATION_POLICY_ATTRIBUTES]
                                [--container-registry-access-level CONTAINER_REGISTRY_
↪ ACCESS_LEVEL]
                                [--container-registry-enabled CONTAINER_REGISTRY_ENABLED]
```

(continues on next page)

(continued from previous page)

```

[--default-branch DEFAULT_BRANCH]
[--description DESCRIPTION]
[--emails-disabled EMAILS_DISABLED]
[--enforce-auth-checks-on-uploads ENFORCE_AUTH_CHECKS_ON_
↪ UPLOADS]
[--external-authorization-classification-label EXTERNAL_
↪ AUTHORIZATION_CLASSIFICATION_LABEL]
[--forking-access-level FORKING_ACCESS_LEVEL]
[--import-url IMPORT_URL]
[--issues-access-level ISSUES_ACCESS_LEVEL]
[--issues-enabled ISSUES_ENABLED]
[--issues-template ISSUES_TEMPLATE]
[--jobs-enabled JOBS_ENABLED]
[--keep-latest-artifact KEEP_LATEST_ARTIFACT]
[--lfs-enabled LFS_ENABLED]
[--merge-commit-template MERGE_COMMIT_TEMPLATE]
[--merge-method MERGE_METHOD]
[--merge-pipelines-enabled MERGE_PIPELINES_ENABLED]
[--merge-requests-access-level MERGE_REQUESTS_ACCESS_LEVEL]
[--merge-requests-enabled MERGE_REQUESTS_ENABLED]
[--merge-requests-template MERGE_REQUESTS_TEMPLATE]
[--merge-trains-enabled MERGE_TRAINS_ENABLED]
[--mirror-overwrites-diverged-branches MIRROR_OVERWRITES_
↪ DIVERGED_BRANCHES]
[--mirror-trigger-builds MIRROR_TRIGGER_BUILDS]
[--mirror-user-id MIRROR_USER_ID]
[--mirror MIRROR]
[--mr-default-target-self MR_DEFAULT_TARGET_SELF]
[--name NAME]
[--operations-access-level OPERATIONS_ACCESS_LEVEL]
[--only-allow-merge-if-all-discussions-are-resolved ONLY_
↪ ALLOW_MERGE_IF_ALL_DISCUSSIONS_ARE_RESOLVED]
[--only-allow-merge-if-pipeline-succeeds ONLY_ALLOW_MERGE_
↪ IF_PIPELINE_SUCCEEDS]
[--only-mirror-protected-branches ONLY_MIRROR_PROTECTED_
↪ BRANCHES]
[--packages-enabled PACKAGES_ENABLED]
[--pages-access-level PAGES_ACCESS_LEVEL]
[--requirements-access-level REQUIREMENTS_ACCESS_LEVEL]
[--restrict-user-defined-variables RESTRICT_USER_DEFINED_
↪ VARIABLES]
[--path PATH] [--public-builds PUBLIC_BUILDS]
[--releases-access-level RELEASES_ACCESS_LEVEL]
[--environments-access-level ENVIRONMENTS_ACCESS_LEVEL]
[--feature-flags-access-level FEATURE_FLAGS_ACCESS_LEVEL]
[--infrastructure-access-level INFRASTRUCTURE_ACCESS_LEVEL]
[--monitor-access-level MONITOR_ACCESS_LEVEL]
[--remove-source-branch-after-merge REMOVE_SOURCE_BRANCH_
↪ AFTER_MERGE]
[--repository-access-level REPOSITORY_ACCESS_LEVEL]
[--repository-storage REPOSITORY_STORAGE]
[--request-access-enabled REQUEST_ACCESS_ENABLED]

```

(continues on next page)

(continued from previous page)

```

[↪DISCUSSIONS]                [--resolve-outdated-diff-discussions RESOLVE_OUTDATED_DIFF_]
                                [--security-and-compliance-access-level SECURITY_AND_]
[↪COMPLIANCE_ACCESS_LEVEL]    [--service-desk-enabled SERVICE_DESK_ENABLED]
                                [--shared-runners-enabled SHARED_RUNNERS_ENABLED]
                                [--show-default-award-emojis SHOW_DEFAULT_AWARD_EMOJIS]
                                [--snippets-access-level SNIPPETS_ACCESS_LEVEL]
                                [--snippets-enabled SNIPPETS_ENABLED]
                                [--issue-branch-template ISSUE_BRANCH_TEMPLATE]
                                [--squash-commit-template SQUASH_COMMIT_TEMPLATE]
                                [--squash-option SQUASH_OPTION]
                                [--suggestion-commit-message SUGGESTION_COMMIT_MESSAGE]
                                [--tag-list TAG_LIST] [--topics TOPICS]
                                [--visibility VISIBILITY]
                                [--wiki-access-level WIKI_ACCESS_LEVEL]
                                [--wiki-enabled WIKI_ENABLED]

```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--id** <id>**--allow-merge-on-skipped-pipeline** <allow\_merge\_on\_skipped\_pipeline>**--only-allow-merge-if-all-status-checks-passed** <only\_allow\_merge\_if\_all\_status\_checks\_passed>**--analytics-access-level** <analytics\_access\_level>**--approvals-before-merge** <approvals\_before\_merge>**--auto-cancel-pending-pipelines** <auto\_cancel\_pending\_pipelines>**--auto-devops-deploy-strategy** <auto\_devops\_deploy\_strategy>**--auto-devops-enabled** <auto\_devops\_enabled>**--autoclose-referenced-issues** <autoclose\_referenced\_issues>**--avatar** <avatar>**--build-coverage-regex** <build\_coverage\_regex>**--build-git-strategy** <build\_git\_strategy>**--build-timeout** <build\_timeout>**--builds-access-level** <builds\_access\_level>**--ci-config-path** <ci\_config\_path>**--ci-default-git-depth** <ci\_default\_git\_depth>**--ci-forward-deployment-enabled** <ci\_forward\_deployment\_enabled>

```
--ci-allow-fork-pipelines-to-run-in-parent-project <ci_allow_fork_pipelines_to_run_in_parent_project>
--ci-separated-caches <ci_separated_caches>
--container-expiration-policy-attributes <container_expiration_policy_attributes>
--container-registry-access-level <container_registry_access_level>
--container-registry-enabled <container_registry_enabled>
--default-branch <default_branch>
--description <description>
--emails-disabled <emails_disabled>
--enforce-auth-checks-on-uploads <enforce_auth_checks_on_uploads>
--external-authorization-classification-label <external_authorization_classification_label>
--forking-access-level <forking_access_level>
--import-url <import_url>
--issues-access-level <issues_access_level>
--issues-enabled <issues_enabled>
--issues-template <issues_template>
--jobs-enabled <jobs_enabled>
--keep-latest-artifact <keep_latest_artifact>
--lfs-enabled <lfs_enabled>
--merge-commit-template <merge_commit_template>
--merge-method <merge_method>
--merge-pipelines-enabled <merge_pipelines_enabled>
--merge-requests-access-level <merge_requests_access_level>
--merge-requests-enabled <merge_requests_enabled>
--merge-requests-template <merge_requests_template>
--merge-trains-enabled <merge_trains_enabled>
--mirror-overwrites-diverged-branches <mirror_overwrites_diverged_branches>
--mirror-trigger-builds <mirror_trigger_builds>
--mirror-user-id <mirror_user_id>
--mirror <mirror>
--mr-default-target-self <mr_default_target_self>
--name <name>
```

```
--operations-access-level <operations_access_level>
--only-allow-merge-if-all-discussions-are-resolved <only_allow_merge_if_all_discussions_are_resolved>
--only-allow-merge-if-pipeline-succeeds <only_allow_merge_if_pipeline_succeeds>
--only-mirror-protected-branches <only_mirror_protected_branches>
--packages-enabled <packages_enabled>
--pages-access-level <pages_access_level>
--requirements-access-level <requirements_access_level>
--restrict-user-defined-variables <restrict_user_defined_variables>
--path <path>
--public-builds <public_builds>
--releases-access-level <releases_access_level>
--environments-access-level <environments_access_level>
--feature-flags-access-level <feature_flags_access_level>
--infrastructure-access-level <infrastructure_access_level>
--monitor-access-level <monitor_access_level>
--remove-source-branch-after-merge <remove_source_branch_after_merge>
--repository-access-level <repository_access_level>
--repository-storage <repository_storage>
--request-access-enabled <request_access_enabled>
--resolve-outdated-diff-discussions <resolve_outdated_diff_discussions>
--security-and-compliance-access-level <security_and_compliance_access_level>
--service-desk-enabled <service_desk_enabled>
--shared-runners-enabled <shared_runners_enabled>
--show-default-award-emojis <show_default_award_emojis>
--snippets-access-level <snippets_access_level>
--snippets-enabled <snippets_enabled>
--issue-branch-template <issue_branch_template>
--squash-commit-template <squash_commit_template>
--squash-option <squash_option>
--suggestion-commit-message <suggestion_commit_message>
--tag-list <tag_list>
```

**--topics** <topics>  
**--visibility** <visibility>  
**--wiki-access-level** <wiki\_access\_level>  
**--wiki-enabled** <wiki\_enabled>

### gitlab project update-submodule

```
usage: gitlab project update-submodule [-h] --id ID --submodule SUBMODULE
                                         --branch BRANCH --commit-sha COMMIT_SHA
```

**-h, --help**  
show this help message and exit  
**--id** <id>  
**--submodule** <submodule>  
**--branch** <branch>  
**--commit-sha** <commit\_sha>

### gitlab project upload

```
usage: gitlab project upload [-h] --id ID --filename FILENAME --filepath
                             FILEPATH
```

**-h, --help**  
show this help message and exit  
**--id** <id>  
**--filename** <filename>  
**--filepath** <filepath>

### gitlab project-access-request

```
usage: gitlab project-access-request [-h] {list,create,delete,approve} ...
```

**-h, --help**  
show this help message and exit

### gitlab project-access-request approve

```
usage: gitlab project-access-request approve [-h] --project-id PROJECT_ID
                                           [--sudo SUDO] --id ID
                                           [--access-level ACCESS_LEVEL]
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

**--access-level** <access\_level>

### gitlab project-access-request create

```
usage: gitlab project-access-request create [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

### gitlab project-access-request delete

```
usage: gitlab project-access-request delete [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-access-request list

```
usage: gitlab project-access-request list [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID [--page PAGE]
                                           [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-access-token

```
usage: gitlab project-access-token [-h] {list,get,create,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab project-access-token create

```
usage: gitlab project-access-token create [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --name NAME --scopes
                                           SCOPES [--access-level ACCESS_LEVEL]
                                           [--expires-at EXPIRES_AT]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

**--scopes** <scopes>

**--access-level** <access\_level>

**--expires-at** <expires\_at>

### gitlab project-access-token delete

```
usage: gitlab project-access-token delete [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>



### gitlab project-access-token get

```
usage: gitlab project-access-token get [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-access-token list

```
usage: gitlab project-access-token list [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID [--page PAGE]
                                         [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-additional-statistics

```
usage: gitlab project-additional-statistics [-h] {get} ...
```

**-h, --help**  
show this help message and exit

#### gitlab project-additional-statistics get

```
usage: gitlab project-additional-statistics get [-h] [--sudo SUDO]
                                                  --project-id PROJECT_ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

## gitlab project-approval

```
usage: gitlab project-approval [-h] {get,update} ...
```

**-h, --help**

show this help message and exit

## gitlab project-approval get

```
usage: gitlab project-approval get [-h] [--sudo SUDO] --project-id PROJECT_ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

## gitlab project-approval update

```
usage: gitlab project-approval update [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID
                                         [--approvals-before-merge APPROVALS_BEFORE_MERGE]
                                         [--reset-approvals-on-push RESET_APPROVALS_ON_PUSH]
                                         [--disable-overriding-approvers-per-merge-request_
↳DISABLE_OVERRIDING_APPROVERS_PER_MERGE_REQUEST]
                                         [--merge-requests-author-approval MERGE_REQUESTS_
↳AUTHOR_APPROVAL]
                                         [--merge-requests-disable-committers-approval_
↳MERGE_REQUESTS_DISABLE_COMMITTERS_APPROVAL]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--approvals-before-merge** <approvals\_before\_merge>

**--reset-approvals-on-push** <reset\_approvals\_on\_push>

**--disable-overriding-approvers-per-merge-request** <disable\_overriding\_approvers\_per\_merge\_request>

**--merge-requests-author-approval** <merge\_requests\_author\_approval>

**--merge-requests-disable-committers-approval** <merge\_requests\_disable\_committers\_approval>

## gitlab project-approval-rule

```
usage: gitlab project-approval-rule [-h] {list,create,update,delete} ...
```

**-h, --help**

show this help message and exit

## gitlab project-approval-rule create

```
usage: gitlab project-approval-rule create [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --name NAME
                                           --approvals-required
                                           APPROVALS_REQUIRED
                                           [--user-ids USER_IDS]
                                           [--group-ids GROUP_IDS]
                                           [--protected-branch-ids PROTECTED_BRANCH_IDS]
                                           [--usernames USERNAMES]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

**--approvals-required** <approvals\_required>

**--user-ids** <user\_ids>

**--group-ids** <group\_ids>

**--protected-branch-ids** <protected\_branch\_ids>

**--usernames** <usernames>

## gitlab project-approval-rule delete

```
usage: gitlab project-approval-rule delete [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-approval-rule list

```
usage: gitlab project-approval-rule list [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID [--page PAGE]
                                           [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-approval-rule update

```
usage: gitlab project-approval-rule update [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-artifact

```
usage: gitlab project-artifact [-h] {delete,download,raw} ...
```

**-h, --help**  
show this help message and exit

### gitlab project-artifact delete

```
usage: gitlab project-artifact delete [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --ref-name REF_NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--ref-name** <ref\_name>

## gitlab project-artifact download

```
usage: gitlab project-artifact download [-h] --project-id PROJECT_ID
                                         [--sudo SUDO] --ref-name REF_NAME
                                         --job JOB [--job-token JOB_TOKEN]
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--ref-name** <ref\_name>

**--job** <job>

**--job-token** <job\_token>

## gitlab project-artifact raw

```
usage: gitlab project-artifact raw [-h] --project-id PROJECT_ID [--sudo SUDO]
                                   --ref-name REF_NAME --artifact-path
                                   ARTIFACT_PATH --job JOB
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--ref-name** <ref\_name>

**--artifact-path** <artifact\_path>

**--job** <job>

## gitlab project-audit-event

```
usage: gitlab project-audit-event [-h] {list,get} ...
```

**-h, --help**  
show this help message and exit

### gitlab project-audit-event get

```
usage: gitlab project-audit-event get [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-audit-event list

```
usage: gitlab project-audit-event list [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID
                                     [--created-after CREATED_AFTER]
                                     [--created-before CREATED_BEFORE]
                                     [--page PAGE] [--per-page PER_PAGE]
                                     [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--created-after** <created\_after>

**--created-before** <created\_before>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-badge

```
usage: gitlab project-badge [-h] {list,get,create,update,delete,render} ...
```

**-h, --help**  
show this help message and exit

### gitlab project-badge create

```
usage: gitlab project-badge create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --link-url LINK_URL --image-url IMAGE_URL
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--link-url** <link\_url>

**--image-url** <image\_url>

### gitlab project-badge delete

```
usage: gitlab project-badge delete [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-badge get

```
usage: gitlab project-badge get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-badge list

```
usage: gitlab project-badge list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                [--page PAGE] [--per-page PER_PAGE]
                                [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-badge render

```
usage: gitlab project-badge render [-h] --project-id PROJECT_ID [--sudo SUDO]
                                   --id ID --link-url LINK_URL --image-url
                                   IMAGE_URL
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

**--link-url** <link\_url>

**--image-url** <image\_url>

### gitlab project-badge update

```
usage: gitlab project-badge update [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --id ID [--link-url LINK_URL]
                                   [--image-url IMAGE_URL]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>



**--link-url** <link\_url>  
**--image-url** <image\_url>

### gitlab project-board

```
usage: gitlab project-board [-h] {list,get,create,update,delete} ...
```

**-h, --help**  
show this help message and exit

### gitlab project-board create

```
usage: gitlab project-board create [-h] [--sudo SUDO] --project-id PROJECT_ID  
--name NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

### gitlab project-board delete

```
usage: gitlab project-board delete [-h] [--sudo SUDO] --project-id PROJECT_ID  
--id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-board get

```
usage: gitlab project-board get [-h] [--sudo SUDO] --project-id PROJECT_ID  
--id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-board list

```
usage: gitlab project-board list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                [--page PAGE] [--per-page PER_PAGE]
                                [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--project-id** <project\_id>**--page** <page>**--per-page** <per\_page>**--get-all**

Return all items from the server, without pagination.

### gitlab project-board update

```
usage: gitlab project-board update [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--project-id** <project\_id>**--id** <id>

### gitlab project-board-list

```
usage: gitlab project-board-list [-h] {list,get,create,update,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab project-board-list create

```
usage: gitlab project-board-list create [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --board-id BOARD_ID
                                         [--label-id LABEL_ID | --assignee-id ASSIGNEE_ID]
↳ | --milestone-id MILESTONE_ID]
```

**-h, --help**

show this help message and exit

```

--sudo <sudo>
--project-id <project_id>
--board-id <board_id>
--label-id <label_id>
--assignee-id <assignee_id>
--milestone-id <milestone_id>

```

### gitlab project-board-list delete

```
usage: gitlab project-board-list delete [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --board-id BOARD_ID --id ID

```

```

-h, --help
    show this help message and exit
--sudo <sudo>
--project-id <project_id>
--board-id <board_id>
--id <id>

```

### gitlab project-board-list get

```
usage: gitlab project-board-list get [-h] [--sudo SUDO] --project-id
                                       PROJECT_ID --board-id BOARD_ID --id ID

```

```

-h, --help
    show this help message and exit
--sudo <sudo>
--project-id <project_id>
--board-id <board_id>
--id <id>

```

### gitlab project-board-list list

```
usage: gitlab project-board-list list [-h] [--sudo SUDO] --project-id
                                       PROJECT_ID --board-id BOARD_ID
                                       [--page PAGE] [--per-page PER_PAGE]
                                       [--get-all]

```

```

-h, --help
    show this help message and exit

```

**--sudo** <sudo>

**--project-id** <project\_id>

**--board-id** <board\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-board-list update

```
usage: gitlab project-board-list update [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --board-id BOARD_ID --id ID
                                         --position POSITION
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--board-id** <board\_id>

**--id** <id>

**--position** <position>

### gitlab project-branch

```
usage: gitlab project-branch [-h] {list,get,create,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab project-branch create

```
usage: gitlab project-branch create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                     --branch BRANCH --ref REF
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--branch** <branch>

**--ref** <ref>

### gitlab project-branch delete

```
usage: gitlab project-branch delete [-h] [--sudo SUDO] --project-id PROJECT_ID
                                     --name NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

### gitlab project-branch get

```
usage: gitlab project-branch get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --name NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

### gitlab project-branch list

```
usage: gitlab project-branch list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   [--page PAGE] [--per-page PER_PAGE]
                                   [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

## gitlab project-ci-lint

```
usage: gitlab project-ci-lint [-h] {get,create,validate} ...
```

**-h, --help**

show this help message and exit

## gitlab project-ci-lint create

```
usage: gitlab project-ci-lint create [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID --content CONTENT
                                     [--dry-run DRY_RUN]
                                     [--include-jobs INCLUDE_JOBS] [--ref REF]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--content** <content>

**--dry-run** <dry\_run>

**--include-jobs** <include\_jobs>

**--ref** <ref>

## gitlab project-ci-lint get

```
usage: gitlab project-ci-lint get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   [--dry-run DRY_RUN]
                                   [--include-jobs INCLUDE_JOBS] [--ref REF]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--dry-run** <dry\_run>

**--include-jobs** <include\_jobs>

**--ref** <ref>

## gitlab project-ci-lint validate

```
usage: gitlab project-ci-lint validate [-h] --project-id PROJECT_ID
                                     [--sudo SUDO] --content CONTENT
                                     [--dry-run DRY_RUN]
                                     [--include-jobs INCLUDE_JOBS]
                                     [--ref REF]
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--content** <content>

**--dry-run** <dry\_run>

**--include-jobs** <include\_jobs>

**--ref** <ref>

## gitlab project-cluster

```
usage: gitlab project-cluster [-h] {list,get,create,update,delete} ...
```

**-h, --help**  
show this help message and exit

## gitlab project-cluster create

```
usage: gitlab project-cluster create [-h] [--sudo SUDO] --project-id
PROJECT_ID --name NAME
--platform-kubernetes-attributes
PLATFORM_KUBERNETES_ATTRIBUTES
[--domain DOMAIN] [--enabled ENABLED]
[--managed MANAGED]
[--environment-scope ENVIRONMENT_SCOPE]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

**--platform-kubernetes-attributes** <platform\_kubernetes\_attributes>

**--domain** <domain>

**--enabled** <enabled>  
**--managed** <managed>  
**--environment-scope** <environment\_scope>

### gitlab project-cluster delete

```
usage: gitlab project-cluster delete [-h] [--sudo SUDO] --project-id  
PROJECT_ID --id ID
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--project-id** <project\_id>  
**--id** <id>

### gitlab project-cluster get

```
usage: gitlab project-cluster get [-h] [--sudo SUDO] --project-id PROJECT_ID  
--id ID
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--project-id** <project\_id>  
**--id** <id>

### gitlab project-cluster list

```
usage: gitlab project-cluster list [-h] [--sudo SUDO] --project-id PROJECT_ID  
[--page PAGE] [--per-page PER_PAGE]  
[--get-all]
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--project-id** <project\_id>  
**--page** <page>  
**--per-page** <per\_page>  
**--get-all**  
Return all items from the server, without pagination.



## gitlab project-cluster update

```
usage: gitlab project-cluster update [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID --id ID [--name NAME]
                                     [--domain DOMAIN]
                                     [--management-project-id MANAGEMENT_PROJECT_ID]
                                     [--platform-kubernetes-attributes PLATFORM_
↪KUBERNETES_ATTRIBUTES]
                                     [--environment-scope ENVIRONMENT_SCOPE]
```

### -h, --help

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

**--name** <name>

**--domain** <domain>

**--management-project-id** <management\_project\_id>

**--platform-kubernetes-attributes** <platform\_kubernetes\_attributes>

**--environment-scope** <environment\_scope>

## gitlab project-commit

```
usage: gitlab project-commit [-h]
                             list
                             get
                             create
                             diff
                             cherry-pick
                             refs
                             merge-requests
                             revert
                             signature
                             ...
```

### -h, --help

show this help message and exit

### gitlab project-commit cherry-pick

```
usage: gitlab project-commit cherry-pick [-h] --project-id PROJECT_ID
                                         [--sudo SUDO] --id ID --branch BRANCH
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

**--branch** <branch>

### gitlab project-commit create

```
usage: gitlab project-commit create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                     --branch BRANCH --commit-message
                                     COMMIT_MESSAGE --actions ACTIONS
                                     [--author-email AUTHOR_EMAIL]
                                     [--author-name AUTHOR_NAME]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--branch** <branch>

**--commit-message** <commit\_message>

**--actions** <actions>

**--author-email** <author\_email>

**--author-name** <author\_name>

### gitlab project-commit diff

```
usage: gitlab project-commit diff [-h] --project-id PROJECT_ID [--sudo SUDO]
                                   --id ID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

## gitlab project-commit get

```
usage: gitlab project-commit get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

## gitlab project-commit list

```
usage: gitlab project-commit list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                [--all ALL] [--ref-name REF_NAME]
                                [--since SINCE] [--until UNTIL]
                                [--path PATH] [--with-stats WITH_STATS]
                                [--first-parent FIRST_PARENT]
                                [--order ORDER] [--trailers TRAILERS]
                                [--page PAGE] [--per-page PER_PAGE]
                                [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--all** <all>

**--ref-name** <ref\_name>

**--since** <since>

**--until** <until>

**--path** <path>

**--with-stats** <with\_stats>

**--first-parent** <first\_parent>

**--order** <order>

**--trailers** <trailers>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-commit merge-requests

```
usage: gitlab project-commit merge-requests [-h] --project-id PROJECT_ID
                                           [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

### gitlab project-commit refs

```
usage: gitlab project-commit refs [-h] --project-id PROJECT_ID [--sudo SUDO]
                                   --id ID [--type TYPE]
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

**--type** <type>

### gitlab project-commit revert

```
usage: gitlab project-commit revert [-h] --project-id PROJECT_ID [--sudo SUDO]
                                     --id ID --branch BRANCH
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

**--branch** <branch>

### gitlab project-commit signature

```
usage: gitlab project-commit signature [-h] --project-id PROJECT_ID
                                     [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

### gitlab project-commit-comment

```
usage: gitlab project-commit-comment [-h] {list,create} ...
```

**-h, --help**  
show this help message and exit

#### gitlab project-commit-comment create

```
usage: gitlab project-commit-comment create [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --commit-id COMMIT_ID
                                           --note NOTE [--path PATH]
                                           [--line LINE]
                                           [--line-type LINE_TYPE]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--commit-id** <commit\_id>

**--note** <note>

**--path** <path>

**--line** <line>

**--line-type** <line\_type>

### gitlab project-commit-comment list

```
usage: gitlab project-commit-comment list [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --commit-id COMMIT_ID
                                           [--page PAGE] [--per-page PER_PAGE]
                                           [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--commit-id** <commit\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-commit-discussion

```
usage: gitlab project-commit-discussion [-h] {list,get,create} ...
```

**-h, --help**

show this help message and exit

### gitlab project-commit-discussion create

```
usage: gitlab project-commit-discussion create [-h] [--sudo SUDO] --project-id
                                                PROJECT_ID --commit-id
                                                COMMIT_ID --body BODY
                                                [--created-at CREATED_AT]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--commit-id** <commit\_id>

**--body** <body>

**--created-at** <created\_at>

### gitlab project-commit-discussion get

```
usage: gitlab project-commit-discussion get [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --commit-id COMMIT_ID
                                           --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--commit-id** <commit\_id>

**--id** <id>

### gitlab project-commit-discussion list

```
usage: gitlab project-commit-discussion list [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --commit-id COMMIT_ID
                                           [--page PAGE]
                                           [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--commit-id** <commit\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-commit-discussion-note

```
usage: gitlab project-commit-discussion-note [-h]
get
create
update
delete
...
```

**-h, --help**

show this help message and exit

### gitlab project-commit-discussion-note create

```
usage: gitlab project-commit-discussion-note create [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --commit-id COMMIT_ID
                                                    --discussion-id
DISCUSSION_ID --body BODY
                                                    [--created-at CREATED_AT]
                                                    [--position POSITION]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--commit-id** <commit\_id>

**--discussion-id** <discussion\_id>

**--body** <body>

**--created-at** <created\_at>

**--position** <position>

### gitlab project-commit-discussion-note delete

```
usage: gitlab project-commit-discussion-note delete [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --commit-id COMMIT_ID
                                                    --discussion-id
DISCUSSION_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--commit-id** <commit\_id>

**--discussion-id** <discussion\_id>

**--id** <id>



### gitlab project-commit-discussion-note get

```
usage: gitlab project-commit-discussion-note get [-h] [--sudo SUDO]
                                                --project-id PROJECT_ID
                                                --commit-id COMMIT_ID
                                                --discussion-id DISCUSSION_ID
                                                --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--project-id** <project\_id>**--commit-id** <commit\_id>**--discussion-id** <discussion\_id>**--id** <id>

### gitlab project-commit-discussion-note update

```
usage: gitlab project-commit-discussion-note update [-h] [--sudo SUDO]
                                                --project-id PROJECT_ID
                                                --commit-id COMMIT_ID
                                                --discussion-id
DISCUSSION_ID --id ID
                                                --body BODY
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--project-id** <project\_id>**--commit-id** <commit\_id>**--discussion-id** <discussion\_id>**--id** <id>**--body** <body>

### gitlab project-commit-status

```
usage: gitlab project-commit-status [-h] {list,create} ...
```

**-h, --help**

show this help message and exit

### gitlab project-commit-status create

```
usage: gitlab project-commit-status create [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --commit-id COMMIT_ID
                                           --state STATE
                                           [--description DESCRIPTION]
                                           [--name NAME] [--context CONTEXT]
                                           [--ref REF]
                                           [--target-url TARGET_URL]
                                           [--coverage COVERAGE]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--commit-id** <commit\_id>

**--state** <state>

**--description** <description>

**--name** <name>

**--context** <context>

**--ref** <ref>

**--target-url** <target\_url>

**--coverage** <coverage>

### gitlab project-commit-status list

```
usage: gitlab project-commit-status list [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --commit-id COMMIT_ID
                                           [--page PAGE] [--per-page PER_PAGE]
                                           [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--commit-id** <commit\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-custom-attribute

```
usage: gitlab project-custom-attribute [-h] {list,get,delete} ...
```

**-h, --help**  
show this help message and exit

### gitlab project-custom-attribute delete

```
usage: gitlab project-custom-attribute delete [-h] [--sudo SUDO] --project-id  
PROJECT_ID --key KEY
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--key** <key>

### gitlab project-custom-attribute get

```
usage: gitlab project-custom-attribute get [-h] [--sudo SUDO] --project-id  
PROJECT_ID --key KEY
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--key** <key>

### gitlab project-custom-attribute list

```
usage: gitlab project-custom-attribute list [-h] [--sudo SUDO] --project-id  
PROJECT_ID [--page PAGE]  
[--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

## gitlab project-deploy-token

```
usage: gitlab project-deploy-token [-h] {list,get,create,delete} ...
```

**-h, --help**

show this help message and exit

## gitlab project-deploy-token create

```
usage: gitlab project-deploy-token create [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --name NAME --scopes
                                           SCOPES [--expires-at EXPIRES_AT]
                                           [--username USERNAME]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

**--scopes** <scopes>

**--expires-at** <expires\_at>

**--username** <username>

## gitlab project-deploy-token delete

```
usage: gitlab project-deploy-token delete [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

## gitlab project-deploy-token get

```
usage: gitlab project-deploy-token get [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-deploy-token list

```
usage: gitlab project-deploy-token list [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID [--scopes SCOPES]
                                         [--page PAGE] [--per-page PER_PAGE]
                                         [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--scopes** <scopes>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-deployment

```
usage: gitlab project-deployment [-h] {list,get,create,update,approval} ...
```

**-h, --help**

show this help message and exit

### gitlab project-deployment approval

```
usage: gitlab project-deployment approval [-h] --project-id PROJECT_ID
                                           [--sudo SUDO] --id ID --status
                                           STATUS [--comment COMMENT]
                                           [--represented-as REPRESENTED_AS]
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

**--status** <status>

**--comment** <comment>

**--represented-as** <represented\_as>

### gitlab project-deployment create

```
usage: gitlab project-deployment create [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --sha SHA --ref REF --tag
                                         TAG --status STATUS --environment
                                         ENVIRONMENT
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--sha** <sha>

**--ref** <ref>

**--tag** <tag>

**--status** <status>

**--environment** <environment>

### gitlab project-deployment get

```
usage: gitlab project-deployment get [-h] [--sudo SUDO] --project-id
                                       PROJECT_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-deployment list

```
usage: gitlab project-deployment list [-h] [--sudo SUDO] --project-id
                                       PROJECT_ID [--order-by ORDER_BY]
                                       [--sort SORT]
                                       [--updated-after UPDATED_AFTER]
                                       [--updated-before UPDATED_BEFORE]
                                       [--environment ENVIRONMENT]
                                       [--status STATUS] [--page PAGE]
                                       [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

```

--project-id <project_id>
--order-by <order_by>
--sort <sort>
--updated-after <updated_after>
--updated-before <updated_before>
--environment <environment>
--status <status>
--page <page>
--per-page <per_page>
--get-all
    Return all items from the server, without pagination.

```

### gitlab project-deployment update

```
usage: gitlab project-deployment update [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --id ID

```

```

-h, --help
    show this help message and exit
--sudo <sudo>
--project-id <project_id>
--id <id>

```

### gitlab project-deployment-merge-request

```
usage: gitlab project-deployment-merge-request [-h] {list} ...

```

```

-h, --help
    show this help message and exit

```

### gitlab project-deployment-merge-request list

```

usage: gitlab project-deployment-merge-request list [-h] [--sudo SUDO]
                                                    --deployment-id
                                                    DEPLOYMENT_ID --project-id
                                                    PROJECT_ID [--state STATE]
                                                    [--order-by ORDER_BY]
                                                    [--sort SORT]
                                                    [--milestone MILESTONE]
                                                    [--view VIEW]
                                                    [--labels LABELS]

```

(continues on next page)

(continued from previous page)

```

↪DETAILS]                                [--with-labels-details WITH_LABELS_]
                                           [--with-merge-status-recheck WITH_]
↪MERGE_STATUS_RECHECK]                  [--created-after CREATED_AFTER]
                                           [--created-before CREATED_BEFORE]
                                           [--updated-after UPDATED_AFTER]
                                           [--updated-before UPDATED_BEFORE]
                                           [--scope SCOPE]
                                           [--author-id AUTHOR_ID]
                                           [--author-username AUTHOR_USERNAME]
                                           [--assignee-id ASSIGNEE_ID]
                                           [--approver-ids APPROVER_IDS]
                                           [--approved-by-ids APPROVED_BY_IDS]
                                           [--reviewer-id REVIEWER_ID]
                                           [--reviewer-username REVIEWER_]
↪USERNAME]                               [--my-reaction-emoji MY_REACTION_]
↪EMOJI]                                  [--source-branch SOURCE_BRANCH]
                                           [--target-branch TARGET_BRANCH]
                                           [--search SEARCH]
                                           [--in IN] [--wip WIP]
                                           [--not NOT]
                                           [--environment ENVIRONMENT]
                                           [--deployed-before DEPLOYED_BEFORE]
                                           [--deployed-after DEPLOYED_AFTER]
                                           [--page PAGE]
                                           [--per-page PER_PAGE]
                                           [--get-all]

```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--deployment-id** <deployment\_id>**--project-id** <project\_id>**--state** <state>**--order-by** <order\_by>**--sort** <sort>**--milestone** <milestone>**--view** <view>**--labels** <labels>**--with-labels-details** <with\_labels\_details>**--with-merge-status-recheck** <with\_merge\_status\_recheck>



```

--created-after <created_after>
--created-before <created_before>
--updated-after <updated_after>
--updated-before <updated_before>
--scope <scope>
--author-id <author_id>
--author-username <author_username>
--assignee-id <assignee_id>
--approver-ids <approver_ids>
--approved-by-ids <approved_by_ids>
--reviewer-id <reviewer_id>
--reviewer-username <reviewer_username>
--my-reaction-emoji <my_reaction_emoji>
--source-branch <source_branch>
--target-branch <target_branch>
--search <search>
--in <in>
--wip <wip>
--not <not>
--environment <environment>
--deployed-before <deployed_before>
--deployed-after <deployed_after>
--page <page>
--per-page <per_page>
--get-all

```

Return all items from the server, without pagination.

### gitlab project-environment

```

usage: gitlab project-environment [-h]
  list
  get
  create
  update

```

(continues on next page)

(continued from previous page)

```
delete
stop
...
```

**-h, --help**

show this help message and exit

**gitlab project-environment create**

```
usage: gitlab project-environment create [-h] [--sudo SUDO] --project-id
PROJECT_ID --name NAME
[--external-url EXTERNAL_URL]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--project-id** <project\_id>**--name** <name>**--external-url** <external\_url>**gitlab project-environment delete**

```
usage: gitlab project-environment delete [-h] [--sudo SUDO] --project-id
PROJECT_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--project-id** <project\_id>**--id** <id>**gitlab project-environment get**

```
usage: gitlab project-environment get [-h] [--sudo SUDO] --project-id
PROJECT_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--project-id** <project\_id>**--id** <id>

## gitlab project-environment list

```
usage: gitlab project-environment list [-h] [--sudo SUDO] --project-id
PROJECT_ID [--name NAME]
      [--search SEARCH] [--states STATES]
      [--page PAGE] [--per-page PER_PAGE]
      [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--project-id** <project\_id>**--name** <name>**--search** <search>**--states** <states>**--page** <page>**--per-page** <per\_page>**--get-all**

Return all items from the server, without pagination.

## gitlab project-environment stop

```
usage: gitlab project-environment stop [-h] --project-id PROJECT_ID
      [--sudo SUDO] --id ID
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>**--sudo** <sudo>**--id** <id>

## gitlab project-environment update

```
usage: gitlab project-environment update [-h] [--sudo SUDO] --project-id
PROJECT_ID --id ID [--name NAME]
      [--external-url EXTERNAL_URL]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--project-id** <project\_id>

**--id** <id>  
**--name** <name>  
**--external-url** <external\_url>

## gitlab project-event

```
usage: gitlab project-event [-h] {list} ...
```

**-h, --help**  
show this help message and exit

## gitlab project-event list

```
usage: gitlab project-event list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                [--action ACTION] [--target-type TARGET_TYPE]
                                [--before BEFORE] [--after AFTER]
                                [--sort SORT] [--scope SCOPE] [--page PAGE]
                                [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--action** <action>

**--target-type** <target\_type>

**--before** <before>

**--after** <after>

**--sort** <sort>

**--scope** <scope>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

## gitlab project-export

```
usage: gitlab project-export [-h] {get,create,download} ...
```

**-h, --help**

show this help message and exit

## gitlab project-export create

```
usage: gitlab project-export create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                     [--description DESCRIPTION]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--description** <description>

## gitlab project-export download

```
usage: gitlab project-export download [-h] --project-id PROJECT_ID
                                       [--sudo SUDO]
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

## gitlab project-export get

```
usage: gitlab project-export get [-h] [--sudo SUDO] --project-id PROJECT_ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

## gitlab project-file

```
usage: gitlab project-file [-h] {get,create,update,delete,raw,blame} ...
```

**-h, --help**

show this help message and exit

## gitlab project-file blame

```
usage: gitlab project-file blame [-h] --project-id PROJECT_ID [--sudo SUDO]
                                --file-path FILE_PATH --ref REF
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--file-path** <file\_path>

**--ref** <ref>

## gitlab project-file create

```
usage: gitlab project-file create [-h] --project-id PROJECT_ID [--sudo SUDO]
                                --file-path FILE_PATH --branch BRANCH
                                --content CONTENT --commit-message
                                COMMIT_MESSAGE [--encoding ENCODING]
                                [--author-email AUTHOR_EMAIL]
                                [--author-name AUTHOR_NAME]
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--file-path** <file\_path>

**--branch** <branch>

**--content** <content>

**--commit-message** <commit\_message>

**--encoding** <encoding>

**--author-email** <author\_email>

**--author-name** <author\_name>

### gitlab project-file delete

```
usage: gitlab project-file delete [-h] --project-id PROJECT_ID [--sudo SUDO]
                                   --file-path FILE_PATH --branch BRANCH
                                   --commit-message COMMIT_MESSAGE
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--file-path** <file\_path>

**--branch** <branch>

**--commit-message** <commit\_message>

### gitlab project-file get

```
usage: gitlab project-file get [-h] --project-id PROJECT_ID [--sudo SUDO]
                               --file-path FILE_PATH --ref REF
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--file-path** <file\_path>

**--ref** <ref>

### gitlab project-file raw

```
usage: gitlab project-file raw [-h] --project-id PROJECT_ID [--sudo SUDO]
                               --file-path FILE_PATH --ref REF
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--file-path** <file\_path>

**--ref** <ref>

### gitlab project-file update

```
usage: gitlab project-file update [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --file-path FILE_PATH --branch BRANCH
                                   --content CONTENT --commit-message
                                   COMMIT_MESSAGE [--encoding ENCODING]
                                   [--author-email AUTHOR_EMAIL]
                                   [--author-name AUTHOR_NAME]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--file-path** <file\_path>

**--branch** <branch>

**--content** <content>

**--commit-message** <commit\_message>

**--encoding** <encoding>

**--author-email** <author\_email>

**--author-name** <author\_name>

### gitlab project-fork

```
usage: gitlab project-fork [-h] {list,create} ...
```

**-h, --help**

show this help message and exit

### gitlab project-fork create

```
usage: gitlab project-fork create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   [--namespace NAMESPACE]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--namespace** <namespace>



**gitlab project-fork list**

```
usage: gitlab project-fork list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                [--archived ARCHIVED]
                                [--visibility VISIBILITY]
                                [--order-by ORDER_BY] [--sort SORT]
                                [--search SEARCH] [--simple SIMPLE]
                                [--owned OWNED] [--membership MEMBERSHIP]
                                [--starred STARRED] [--statistics STATISTICS]
                                [--with-custom-attributes WITH_CUSTOM_ATTRIBUTES]
                                [--with-issues-enabled WITH_ISSUES_ENABLED]
                                [--with-merge-requests-enabled WITH_MERGE_REQUESTS_
↪ENABLED]
                                [--page PAGE] [--per-page PER_PAGE]
                                [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--archived** <archived>

**--visibility** <visibility>

**--order-by** <order\_by>

**--sort** <sort>

**--search** <search>

**--simple** <simple>

**--owned** <owned>

**--membership** <membership>

**--starred** <starred>

**--statistics** <statistics>

**--with-custom-attributes** <with\_custom\_attributes>

**--with-issues-enabled** <with\_issues\_enabled>

**--with-merge-requests-enabled** <with\_merge\_requests\_enabled>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab project-hook

```
usage: gitlab project-hook [-h] {list,get,create,update,delete} ...
```

### -h, --help

show this help message and exit

## gitlab project-hook create

```
usage: gitlab project-hook create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --url URL [--push-events PUSH_EVENTS]
                                   [--issues-events ISSUES_EVENTS]
                                   [--confidential-issues-events CONFIDENTIAL_ISSUES_
↳EVENTS]
                                   [--merge-requests-events MERGE_REQUESTS_EVENTS]
                                   [--tag-push-events TAG_PUSH_EVENTS]
                                   [--note-events NOTE_EVENTS]
                                   [--job-events JOB_EVENTS]
                                   [--pipeline-events PIPELINE_EVENTS]
                                   [--wiki-page-events WIKI_PAGE_EVENTS]
                                   [--enable-ssl-verification ENABLE_SSL_VERIFICATION]
                                   [--token TOKEN]
```

### -h, --help

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--url** <url>

**--push-events** <push\_events>

**--issues-events** <issues\_events>

**--confidential-issues-events** <confidential\_issues\_events>

**--merge-requests-events** <merge\_requests\_events>

**--tag-push-events** <tag\_push\_events>

**--note-events** <note\_events>

**--job-events** <job\_events>

**--pipeline-events** <pipeline\_events>

**--wiki-page-events** <wiki\_page\_events>

**--enable-ssl-verification** <enable\_ssl\_verification>

**--token** <token>

### gitlab project-hook delete

```
usage: gitlab project-hook delete [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-hook get

```
usage: gitlab project-hook get [-h] [--sudo SUDO] --project-id PROJECT_ID --id
                                ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-hook list

```
usage: gitlab project-hook list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                [--page PAGE] [--per-page PER_PAGE]
                                [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

## gitlab project-hook update

```
usage: gitlab project-hook update [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --id ID --url URL
                                   [--push-events PUSH_EVENTS]
                                   [--issues-events ISSUES_EVENTS]
                                   [--confidential-issues-events CONFIDENTIAL_ISSUES_
↪EVENTS]
                                   [--merge-requests-events MERGE_REQUESTS_EVENTS]
                                   [--tag-push-events TAG_PUSH_EVENTS]
                                   [--note-events NOTE_EVENTS]
                                   [--job-events JOB_EVENTS]
                                   [--pipeline-events PIPELINE_EVENTS]
                                   [--wiki-events WIKI_EVENTS]
                                   [--enable-ssl-verification ENABLE_SSL_VERIFICATION]
                                   [--token TOKEN]
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

**--url** <url>

**--push-events** <push\_events>

**--issues-events** <issues\_events>

**--confidential-issues-events** <confidential\_issues\_events>

**--merge-requests-events** <merge\_requests\_events>

**--tag-push-events** <tag\_push\_events>

**--note-events** <note\_events>

**--job-events** <job\_events>

**--pipeline-events** <pipeline\_events>

**--wiki-events** <wiki\_events>

**--enable-ssl-verification** <enable\_ssl\_verification>

**--token** <token>

## gitlab project-import

```
usage: gitlab project-import [-h] {get} ...
```

**-h, --help**  
show this help message and exit

## gitlab project-import get

```
usage: gitlab project-import get [-h] [--sudo SUDO] --project-id PROJECT_ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

## gitlab project-integration

```
usage: gitlab project-integration [-h] {list,get,update,delete,available} ...
```

**-h, --help**  
show this help message and exit

## gitlab project-integration available

```
usage: gitlab project-integration available [-h] --project-id PROJECT_ID
                                           [--sudo SUDO] --slug SLUG
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--slug** <slug>

## gitlab project-integration delete

```
usage: gitlab project-integration delete [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --slug SLUG
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--slug** <slug>

### gitlab project-integration get

```
usage: gitlab project-integration get [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID --slug SLUG
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--slug** <slug>

### gitlab project-integration list

```
usage: gitlab project-integration list [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID [--page PAGE]
                                     [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-integration update

```
usage: gitlab project-integration update [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID --slug SLUG
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--slug** <slug>

## gitlab project-invitation

```
usage: gitlab project-invitation [-h] {list,get,create,update,delete} ...
```

### -h, --help

show this help message and exit

## gitlab project-invitation create

```
usage: gitlab project-invitation create [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --access-level ACCESS_LEVEL
                                         [--expires-at EXPIRES_AT]
                                         [--invite-source INVITE_SOURCE]
                                         [--tasks-to-be-done TASKS_TO_BE_DONE]
                                         [--tasks-project-id TASKS_PROJECT_ID]
                                         [--email EMAIL | --user-id USER_ID]
```

### -h, --help

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--access-level** <access\_level>

**--expires-at** <expires\_at>

**--invite-source** <invite\_source>

**--tasks-to-be-done** <tasks\_to\_be\_done>

**--tasks-project-id** <tasks\_project\_id>

**--email** <email>

**--user-id** <user\_id>

## gitlab project-invitation delete

```
usage: gitlab project-invitation delete [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --email EMAIL
```

### -h, --help

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--email** <email>

### gitlab project-invitation get

```
usage: gitlab project-invitation get [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID --email EMAIL
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--email** <email>

### gitlab project-invitation list

```
usage: gitlab project-invitation list [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID [--query QUERY] [--page PAGE]
                                     [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--query** <query>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-invitation update

```
usage: gitlab project-invitation update [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID --email EMAIL
                                     [--access-level ACCESS_LEVEL]
                                     [--expires-at EXPIRES_AT]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--email** <email>

**--access-level** <access\_level>

**--expires-at** <expires\_at>



## gitlab project-issue

```
usage: gitlab project-issue [-h]
    list
    get
    create
    update
    delete
    user-agent-detail
    subscribe
    unsubscribe
    todo
    time-stats
    time-estimate
    reset-time-estimate
    add-spent-time
    reset-spent-time
    participants
    move
    reorder
    related-merge-requests
    closed-by
    ...
```

### **-h, --help**

show this help message and exit

## gitlab project-issue add-spent-time

```
usage: gitlab project-issue add-spent-time [-h] --project-id PROJECT_ID
                                           [--sudo SUDO] --iid IID --duration
                                           DURATION
```

### **-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

**--duration** <duration>

### gitlab project-issue closed-by

```
usage: gitlab project-issue closed-by [-h] --project-id PROJECT_ID
                                     [--sudo SUDO] --iid IID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-issue create

```
usage: gitlab project-issue create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --title TITLE [--description DESCRIPTION]
                                   [--confidential CONFIDENTIAL]
                                   [--assignee-ids ASSIGNEE_IDS]
                                   [--assignee-id ASSIGNEE_ID]
                                   [--milestone-id MILESTONE_ID]
                                   [--labels LABELS] [--created-at CREATED_AT]
                                   [--due-date DUE_DATE]
                                   [--merge-request-to-resolve-discussions-of MERGE_
↳REQUEST_TO_RESOLVE_DISCUSSIONS_OF]
                                   [--discussion-to-resolve DISCUSSION_TO_RESOLVE]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--title** <title>

**--description** <description>

**--confidential** <confidential>

**--assignee-ids** <assignee\_ids>

**--assignee-id** <assignee\_id>

**--milestone-id** <milestone\_id>

**--labels** <labels>

**--created-at** <created\_at>

**--due-date** <due\_date>

**--merge-request-to-resolve-discussions-of** <merge\_request\_to\_resolve\_discussions\_of>

**--discussion-to-resolve** <discussion\_to\_resolve>

### gitlab project-issue delete

```
usage: gitlab project-issue delete [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --iid IID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--iid** <iid>

### gitlab project-issue get

```
usage: gitlab project-issue get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --iid IID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--iid** <iid>

### gitlab project-issue list

```
usage: gitlab project-issue list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   [--iids IIDS] [--state STATE]
                                   [--labels LABELS] [--milestone MILESTONE]
                                   [--scope SCOPE] [--author-id AUTHOR_ID]
                                   [--iteration-id ITERATION_ID]
                                   [--assignee-id ASSIGNEE_ID]
                                   [--my-reaction-emoji MY_REACTION_EMOJI]
                                   [--order-by ORDER_BY] [--sort SORT]
                                   [--search SEARCH]
                                   [--created-after CREATED_AFTER]
                                   [--created-before CREATED_BEFORE]
                                   [--updated-after UPDATED_AFTER]
                                   [--updated-before UPDATED_BEFORE]
                                   [--page PAGE] [--per-page PER_PAGE]
                                   [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

```
--iids <iids>
--state <state>
--labels <labels>
--milestone <milestone>
--scope <scope>
--author-id <author_id>
--iteration-id <iteration_id>
--assignee-id <assignee_id>
--my-reaction-emoji <my_reaction_emoji>
--order-by <order_by>
--sort <sort>
--search <search>
--created-after <created_after>
--created-before <created_before>
--updated-after <updated_after>
--updated-before <updated_before>
--page <page>
--per-page <per_page>
--get-all
    Return all items from the server, without pagination.
```

### gitlab project-issue move

```
usage: gitlab project-issue move [-h] --project-id PROJECT_ID [--sudo SUDO]
                                --iid IID --to-project-id TO_PROJECT_ID
```

```
-h, --help
    show this help message and exit
--project-id <project_id>
--sudo <sudo>
--iid <iid>
--to-project-id <to_project_id>
```

### gitlab project-issue participants

```
usage: gitlab project-issue participants [-h] --project-id PROJECT_ID
                                         [--sudo SUDO] --iid IID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-issue related-merge-requests

```
usage: gitlab project-issue related-merge-requests [-h] --project-id
                                                    PROJECT_ID [--sudo SUDO]
                                                    --iid IID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-issue reorder

```
usage: gitlab project-issue reorder [-h] --project-id PROJECT_ID [--sudo SUDO]
                                     --iid IID --move-after-id MOVE_AFTER_ID
                                     --move-before-id MOVE_BEFORE_ID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

**--move-after-id** <move\_after\_id>

**--move-before-id** <move\_before\_id>

### gitlab project-issue reset-spent-time

```
usage: gitlab project-issue reset-spent-time [-h] --project-id PROJECT_ID
                                           [--sudo SUDO] --iid IID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-issue reset-time-estimate

```
usage: gitlab project-issue reset-time-estimate [-h] --project-id PROJECT_ID
                                                [--sudo SUDO] --iid IID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-issue subscribe

```
usage: gitlab project-issue subscribe [-h] --project-id PROJECT_ID
                                       [--sudo SUDO] --iid IID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-issue time-estimate

```
usage: gitlab project-issue time-estimate [-h] --project-id PROJECT_ID
                                           [--sudo SUDO] --iid IID --duration
                                           DURATION
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>  
**--iid** <iid>  
**--duration** <duration>

### gitlab project-issue time-stats

```
usage: gitlab project-issue time-stats [-h] --project-id PROJECT_ID
                                     [--sudo SUDO] --iid IID
```

**-h, --help**  
show this help message and exit  
**--project-id** <project\_id>  
**--sudo** <sudo>  
**--iid** <iid>

### gitlab project-issue todo

```
usage: gitlab project-issue todo [-h] --project-id PROJECT_ID [--sudo SUDO]
                                --iid IID
```

**-h, --help**  
show this help message and exit  
**--project-id** <project\_id>  
**--sudo** <sudo>  
**--iid** <iid>

### gitlab project-issue unsubscribe

```
usage: gitlab project-issue unsubscribe [-h] --project-id PROJECT_ID
                                       [--sudo SUDO] --iid IID
```

**-h, --help**  
show this help message and exit  
**--project-id** <project\_id>  
**--sudo** <sudo>  
**--iid** <iid>

## gitlab project-issue update

```
usage: gitlab project-issue update [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --iid IID [--title TITLE]
                                   [--description DESCRIPTION]
                                   [--confidential CONFIDENTIAL]
                                   [--assignee-ids ASSIGNEE_IDS]
                                   [--assignee-id ASSIGNEE_ID]
                                   [--milestone-id MILESTONE_ID]
                                   [--labels LABELS]
                                   [--state-event STATE_EVENT]
                                   [--updated-at UPDATED_AT]
                                   [--due-date DUE_DATE]
                                   [--discussion-locked DISCUSSION_LOCKED]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--iid** <iid>

**--title** <title>

**--description** <description>

**--confidential** <confidential>

**--assignee-ids** <assignee\_ids>

**--assignee-id** <assignee\_id>

**--milestone-id** <milestone\_id>

**--labels** <labels>

**--state-event** <state\_event>

**--updated-at** <updated\_at>

**--due-date** <due\_date>

**--discussion-locked** <discussion\_locked>

## gitlab project-issue user-agent-detail

```
usage: gitlab project-issue user-agent-detail [-h] --project-id PROJECT_ID
                                              [--sudo SUDO] --iid IID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>



**--sudo** <sudo>

**--iid** <iid>

### gitlab project-issue-award-emoji

```
usage: gitlab project-issue-award-emoji [-h] {list,get,create,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab project-issue-award-emoji create

```
usage: gitlab project-issue-award-emoji create [-h] [--sudo SUDO] --project-id
PROJECT_ID --issue-iid
ISSUE_IID --name NAME
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--name** <name>

### gitlab project-issue-award-emoji delete

```
usage: gitlab project-issue-award-emoji delete [-h] [--sudo SUDO] --project-id
PROJECT_ID --issue-iid
ISSUE_IID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--id** <id>

### gitlab project-issue-award-emoji get

```
usage: gitlab project-issue-award-emoji get [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --issue-iid ISSUE_IID
                                           --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--id** <id>

### gitlab project-issue-award-emoji list

```
usage: gitlab project-issue-award-emoji list [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --issue-iid ISSUE_IID
                                           [--page PAGE]
                                           [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-issue-discussion

```
usage: gitlab project-issue-discussion [-h] {list,get,create} ...
```

**-h, --help**  
show this help message and exit

### gitlab project-issue-discussion create

```
usage: gitlab project-issue-discussion create [-h] [--sudo SUDO] --project-id
                                             PROJECT_ID --issue-iid ISSUE_IID
                                             --body BODY
                                             [--created-at CREATED_AT]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--body** <body>

**--created-at** <created\_at>

### gitlab project-issue-discussion get

```
usage: gitlab project-issue-discussion get [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --issue-iid ISSUE_IID
                                           --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--id** <id>

### gitlab project-issue-discussion list

```
usage: gitlab project-issue-discussion list [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --issue-iid ISSUE_IID
                                           [--page PAGE]
                                           [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-issue-discussion-note

```
usage: gitlab project-issue-discussion-note [-h]
get
create
update
delete
...
```

**-h, --help**

show this help message and exit

### gitlab project-issue-discussion-note create

```
usage: gitlab project-issue-discussion-note create [-h] [--sudo SUDO]
--project-id PROJECT_ID
--issue-iid ISSUE_IID
--discussion-id
DISCUSSION_ID --body BODY
[--created-at CREATED_AT]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--discussion-id** <discussion\_id>

**--body** <body>

**--created-at** <created\_at>

### gitlab project-issue-discussion-note delete

```
usage: gitlab project-issue-discussion-note delete [-h] [--sudo SUDO]
--project-id PROJECT_ID
--issue-iid ISSUE_IID
--discussion-id
DISCUSSION_ID --id ID
```

**-h, --help**

show this help message and exit

```
--sudo <sudo>
--project-id <project_id>
--issue-iid <issue_iid>
--discussion-id <discussion_id>
--id <id>
```

### gitlab project-issue-discussion-note get

```
usage: gitlab project-issue-discussion-note get [-h] [--sudo SUDO]
                                                --project-id PROJECT_ID
                                                --issue-iid ISSUE_IID
                                                --discussion-id DISCUSSION_ID
                                                --id ID
```

**-h, --help**  
show this help message and exit

```
--sudo <sudo>
--project-id <project_id>
--issue-iid <issue_iid>
--discussion-id <discussion_id>
--id <id>
```

### gitlab project-issue-discussion-note update

```
usage: gitlab project-issue-discussion-note update [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --issue-iid ISSUE_IID
                                                    --discussion-id
DISCUSSION_ID --id ID
                                                    --body BODY
```

**-h, --help**  
show this help message and exit

```
--sudo <sudo>
--project-id <project_id>
--issue-iid <issue_iid>
--discussion-id <discussion_id>
--id <id>
--body <body>
```

## gitlab project-issue-link

```
usage: gitlab project-issue-link [-h] {list,create,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab project-issue-link create

```
usage: gitlab project-issue-link create [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --issue-iid ISSUE_IID
                                         --target-project-id TARGET_PROJECT_ID
                                         --target-issue-iid TARGET_ISSUE_IID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--target-project-id** <target\_project\_id>

**--target-issue-iid** <target\_issue\_iid>

### gitlab project-issue-link delete

```
usage: gitlab project-issue-link delete [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --issue-iid ISSUE_IID
                                         --issue-link-id ISSUE_LINK_ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--issue-link-id** <issue\_link\_id>

### gitlab project-issue-link list

```
usage: gitlab project-issue-link list [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID --issue-iid ISSUE_IID
                                     [--page PAGE] [--per-page PER_PAGE]
                                     [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-issue-note

```
usage: gitlab project-issue-note [-h] {list,get,create,update,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab project-issue-note create

```
usage: gitlab project-issue-note create [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID --issue-iid ISSUE_IID
                                     --body BODY [--created-at CREATED_AT]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--body** <body>

**--created-at** <created\_at>

### gitlab project-issue-note delete

```
usage: gitlab project-issue-note delete [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --issue-iid ISSUE_IID --id
                                         ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--id** <id>

### gitlab project-issue-note get

```
usage: gitlab project-issue-note get [-h] [--sudo SUDO] --project-id
                                       PROJECT_ID --issue-iid ISSUE_IID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--id** <id>

### gitlab project-issue-note list

```
usage: gitlab project-issue-note list [-h] [--sudo SUDO] --project-id
                                       PROJECT_ID --issue-iid ISSUE_IID
                                       [--page PAGE] [--per-page PER_PAGE]
                                       [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.



### gitlab project-issue-note update

```
usage: gitlab project-issue-note update [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --issue-iid ISSUE_IID --id
                                         ID --body BODY
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--id** <id>

**--body** <body>

### gitlab project-issue-note-award-emoji

```
usage: gitlab project-issue-note-award-emoji [-h] {list,get,create,delete} ...
```

**-h, --help**  
show this help message and exit

### gitlab project-issue-note-award-emoji create

```
usage: gitlab project-issue-note-award-emoji create [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --issue-iid ISSUE_IID
                                                    --note-id NOTE_ID --name
                                                    NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--note-id** <note\_id>

**--name** <name>

### gitlab project-issue-note-award-emoji delete

```
usage: gitlab project-issue-note-award-emoji delete [-h] [--sudo SUDO]
                                         --project-id PROJECT_ID
                                         --issue-iid ISSUE_IID
                                         --note-id NOTE_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--note-id** <note\_id>

**--id** <id>

### gitlab project-issue-note-award-emoji get

```
usage: gitlab project-issue-note-award-emoji get [-h] [--sudo SUDO]
                                         --project-id PROJECT_ID
                                         --issue-iid ISSUE_IID
                                         --note-id NOTE_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--note-id** <note\_id>

**--id** <id>

### gitlab project-issue-note-award-emoji list

```
usage: gitlab project-issue-note-award-emoji list [-h] [--sudo SUDO]
                                         --project-id PROJECT_ID
                                         --issue-iid ISSUE_IID
                                         --note-id NOTE_ID
                                         [--page PAGE]
                                         [--per-page PER_PAGE]
                                         [--get-all]
```

**-h, --help**  
show this help message and exit

```

--sudo <sudo>
--project-id <project_id>
--issue-iid <issue_iid>
--note-id <note_id>
--page <page>
--per-page <per_page>
--get-all
    Return all items from the server, without pagination.

```

### gitlab project-issue-resource-iteration-event

```
usage: gitlab project-issue-resource-iteration-event [-h] {list,get} ...
```

```

-h, --help
    show this help message and exit

```

### gitlab project-issue-resource-iteration-event get

```

usage: gitlab project-issue-resource-iteration-event get [-h] [--sudo SUDO]
                                                    --project-id
                                                    PROJECT_ID
                                                    --issue-iid ISSUE_IID
                                                    --id ID

```

```

-h, --help
    show this help message and exit

--sudo <sudo>
--project-id <project_id>
--issue-iid <issue_iid>
--id <id>

```

### gitlab project-issue-resource-iteration-event list

```

usage: gitlab project-issue-resource-iteration-event list [-h] [--sudo SUDO]
                                                    --project-id
                                                    PROJECT_ID
                                                    --issue-iid
                                                    ISSUE_IID
                                                    [--page PAGE]
                                                    [--per-page PER_PAGE]
                                                    [--get-all]

```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-issue-resource-label-event

```
usage: gitlab project-issue-resource-label-event [-h] {list,get} ...
```

**-h, --help**

show this help message and exit

### gitlab project-issue-resource-label-event get

```
usage: gitlab project-issue-resource-label-event get [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --issue-iid ISSUE_IID
                                                    --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--id** <id>

### gitlab project-issue-resource-label-event list

```
usage: gitlab project-issue-resource-label-event list [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --issue-iid ISSUE_IID
                                                    [--page PAGE]
                                                    [--per-page PER_PAGE]
                                                    [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-issue-resource-milestone-event

```
usage: gitlab project-issue-resource-milestone-event [-h] {list,get} ...
```

**-h, --help**

show this help message and exit

### gitlab project-issue-resource-milestone-event get

```
usage: gitlab project-issue-resource-milestone-event get [-h] [--sudo SUDO]
                                     --project-id
                                     PROJECT_ID
                                     --issue-iid ISSUE_IID
                                     --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--id** <id>

### gitlab project-issue-resource-milestone-event list

```
usage: gitlab project-issue-resource-milestone-event list [-h] [--sudo SUDO]
                                                           --project-id
                                                           PROJECT_ID
                                                           --issue-iid
                                                           ISSUE_IID
                                                           [--page PAGE]
                                                           [--per-page PER_PAGE]
                                                           [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-issue-resource-state-event

```
usage: gitlab project-issue-resource-state-event [-h] {list,get} ...
```

**-h, --help**

show this help message and exit

### gitlab project-issue-resource-state-event get

```
usage: gitlab project-issue-resource-state-event get [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --issue-iid ISSUE_IID
                                                    --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--id** <id>

### gitlab project-issue-resource-state-event list

```
usage: gitlab project-issue-resource-state-event list [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --issue-iid ISSUE_IID
                                                    [--page PAGE]
                                                    [--per-page PER_PAGE]
                                                    [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-issue-resource-weight-event

```
usage: gitlab project-issue-resource-weight-event [-h] {list,get} ...
```

**-h, --help**

show this help message and exit

### gitlab project-issue-resource-weight-event get

```
usage: gitlab project-issue-resource-weight-event get [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --issue-iid ISSUE_IID
                                                    --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--id** <id>

### gitlab project-issue-resource-weight-event list

```
usage: gitlab project-issue-resource-weight-event list [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --issue-iid ISSUE_IID
                                                    [--page PAGE]
                                                    [--per-page PER_PAGE]
                                                    [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--issue-iid** <issue\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-issues-statistics

```
usage: gitlab project-issues-statistics [-h] {get} ...
```

**-h, --help**

show this help message and exit

### gitlab project-issues-statistics get

```
usage: gitlab project-issues-statistics get [-h] [--sudo SUDO] --project-id
      PROJECT_ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

### gitlab project-job

```
usage: gitlab project-job [-h]
      list
      get
      cancel
      retry
      play
      erase
      keep-artifacts
      delete-artifacts
      artifacts
      artifact
      trace
      ...
```

**-h, --help**

show this help message and exit



### gitlab project-job artifact

```
usage: gitlab project-job artifact [-h] --project-id PROJECT_ID [--sudo SUDO]
                                   --id ID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

### gitlab project-job artifacts

```
usage: gitlab project-job artifacts [-h] --project-id PROJECT_ID [--sudo SUDO]
                                   --id ID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

### gitlab project-job cancel

```
usage: gitlab project-job cancel [-h] --project-id PROJECT_ID [--sudo SUDO]
                                   --id ID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

### gitlab project-job delete-artifacts

```
usage: gitlab project-job delete-artifacts [-h] --project-id PROJECT_ID
                                           [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

### gitlab project-job erase

```
usage: gitlab project-job erase [-h] --project-id PROJECT_ID [--sudo SUDO]
                               --id ID
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

### gitlab project-job get

```
usage: gitlab project-job get [-h] [--sudo SUDO] --project-id PROJECT_ID --id
                               ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-job keep-artifacts

```
usage: gitlab project-job keep-artifacts [-h] --project-id PROJECT_ID
                                           [--sudo SUDO] --id ID
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

### gitlab project-job list

```
usage: gitlab project-job list [-h] [--sudo SUDO] --project-id PROJECT_ID
                               [--scope SCOPE] [--page PAGE]
                               [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--project-id** <project\_id>**--scope** <scope>**--page** <page>**--per-page** <per\_page>**--get-all**

Return all items from the server, without pagination.

### gitlab project-job play

```
usage: gitlab project-job play [-h] --project-id PROJECT_ID [--sudo SUDO] --id
                               ID
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>**--sudo** <sudo>**--id** <id>

### gitlab project-job retry

```
usage: gitlab project-job retry [-h] --project-id PROJECT_ID [--sudo SUDO]
                               --id ID
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>**--sudo** <sudo>**--id** <id>

### gitlab project-job trace

```
usage: gitlab project-job trace [-h] --project-id PROJECT_ID [--sudo SUDO]
                                --id ID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

### gitlab project-job-token-scope

```
usage: gitlab project-job-token-scope [-h] {get,update} ...
```

**-h, --help**  
show this help message and exit

#### gitlab project-job-token-scope get

```
usage: gitlab project-job-token-scope get [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

#### gitlab project-job-token-scope update

```
usage: gitlab project-job-token-scope update [-h] [--sudo SUDO] --project-id
                                              PROJECT_ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

## gitlab project-key

```
usage: gitlab project-key [-h] {list,get,create,update,delete,enable} ...
```

**-h, --help**

show this help message and exit

## gitlab project-key create

```
usage: gitlab project-key create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                --title TITLE --key KEY [--can-push CAN_PUSH]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--title** <title>

**--key** <key>

**--can-push** <can\_push>

## gitlab project-key delete

```
usage: gitlab project-key delete [-h] [--sudo SUDO] --project-id PROJECT_ID
                                --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

## gitlab project-key enable

```
usage: gitlab project-key enable [-h] --project-id PROJECT_ID [--sudo SUDO]
                                --id ID --key-id KEY_ID
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

**--key-id** <key\_id>

### gitlab project-key get

```
usage: gitlab project-key get [-h] [--sudo SUDO] --project-id PROJECT_ID --id
                               ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-key list

```
usage: gitlab project-key list [-h] [--sudo SUDO] --project-id PROJECT_ID
                               [--page PAGE] [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-key update

```
usage: gitlab project-key update [-h] [--sudo SUDO] --project-id PROJECT_ID
                                --id ID [--title TITLE] [--can-push CAN_PUSH]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

**--title** <title>

**--can-push** <can\_push>

## gitlab project-label

```
usage: gitlab project-label [-h]
    list
    get
    create
    update
    delete
    subscribe
    unsubscribe
    ...
```

### -h, --help

show this help message and exit

## gitlab project-label create

```
usage: gitlab project-label create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --name NAME --color COLOR
                                   [--description DESCRIPTION]
                                   [--priority PRIORITY]
```

### -h, --help

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

**--color** <color>

**--description** <description>

**--priority** <priority>

## gitlab project-label delete

```
usage: gitlab project-label delete [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --name NAME
```

### -h, --help

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

### gitlab project-label get

```
usage: gitlab project-label get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                --name NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

### gitlab project-label list

```
usage: gitlab project-label list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                [--page PAGE] [--per-page PER_PAGE]
                                [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-label subscribe

```
usage: gitlab project-label subscribe [-h] --project-id PROJECT_ID
                                       [--sudo SUDO] --name NAME
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--name** <name>



### gitlab project-label unsubscribe

```
usage: gitlab project-label unsubscribe [-h] --project-id PROJECT_ID
                                         [--sudo SUDO] --name NAME
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--name** <name>

### gitlab project-label update

```
usage: gitlab project-label update [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --name NAME [--new-name NEW_NAME]
                                   [--color COLOR] [--description DESCRIPTION]
                                   [--priority PRIORITY]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

**--new-name** <new\_name>

**--color** <color>

**--description** <description>

**--priority** <priority>

### gitlab project-member

```
usage: gitlab project-member [-h] {list,get,create,update,delete} ...
```

**-h, --help**  
show this help message and exit

### gitlab project-member create

```
usage: gitlab project-member create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                     --access-level ACCESS_LEVEL --user-id
                                     USER_ID [--expires-at EXPIRES_AT]
                                     [--tasks-to-be-done TASKS_TO_BE_DONE]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--access-level** <access\_level>

**--user-id** <user\_id>

**--expires-at** <expires\_at>

**--tasks-to-be-done** <tasks\_to\_be\_done>

### gitlab project-member delete

```
usage: gitlab project-member delete [-h] [--sudo SUDO] --project-id PROJECT_ID
                                     --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-member get

```
usage: gitlab project-member get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-member list

```
usage: gitlab project-member list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   [--page PAGE] [--per-page PER_PAGE]
                                   [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo <sudo>**

**--project-id <project\_id>**

**--page <page>**

**--per-page <per\_page>**

**--get-all**

Return all items from the server, without pagination.

### gitlab project-member update

```
usage: gitlab project-member update [-h] [--sudo SUDO] --project-id PROJECT_ID
                                     --id ID --access-level ACCESS_LEVEL
                                     [--expires-at EXPIRES_AT]
```

**-h, --help**

show this help message and exit

**--sudo <sudo>**

**--project-id <project\_id>**

**--id <id>**

**--access-level <access\_level>**

**--expires-at <expires\_at>**

### gitlab project-member-all

```
usage: gitlab project-member-all [-h] {list,get} ...
```

**-h, --help**

show this help message and exit

### gitlab project-member-all get

```
usage: gitlab project-member-all get [-h] [--sudo SUDO] --project-id
PROJECT_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-member-all list

```
usage: gitlab project-member-all list [-h] [--sudo SUDO] --project-id
PROJECT_ID [--page PAGE]
[--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-merge-request

```
usage: gitlab project-merge-request [-h]
list
get
create
update
delete
subscribe
unsubscribe
todo
time-stats
time-estimate
reset-time-estimate
add-spent-time
reset-spent-time
participants
cancel-merge-when-pipeline-succeeds
```

(continues on next page)

(continued from previous page)

```

closes-issues
commits
changes
approve
unapprove
rebase
reset-approvals
merge-ref
merge

```

...

**-h, --help**

show this help message and exit

**gitlab project-merge-request add-spent-time**

```

usage: gitlab project-merge-request add-spent-time [-h] --project-id
                                                    PROJECT_ID [--sudo SUDO]
                                                    --iid IID --duration
                                                    DURATION

```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>**--sudo** <sudo>**--iid** <iid>**--duration** <duration>**gitlab project-merge-request approve**

```

usage: gitlab project-merge-request approve [-h] --project-id PROJECT_ID
                                              [--sudo SUDO] --iid IID
                                              [--sha SHA]

```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>**--sudo** <sudo>**--iid** <iid>**--sha** <sha>

### gitlab project-merge-request cancel-merge-when-pipeline-succeeds

```
usage: gitlab project-merge-request cancel-merge-when-pipeline-succeeds
       [-h] --project-id PROJECT_ID [--sudo SUDO] --iid IID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-merge-request changes

```
usage: gitlab project-merge-request changes [-h] --project-id PROJECT_ID
                                             [--sudo SUDO] --iid IID
                                             [--access-raw-diffs ACCESS_RAW_DIFFS]
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

**--access-raw-diffs** <access\_raw\_diffs>

### gitlab project-merge-request closes-issues

```
usage: gitlab project-merge-request closes-issues [-h] --project-id PROJECT_ID
                                                    [--sudo SUDO] --iid IID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

## gitlab project-merge-request commits

```
usage: gitlab project-merge-request commits [-h] --project-id PROJECT_ID
                                           [--sudo SUDO] --iid IID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

## gitlab project-merge-request create

```
usage: gitlab project-merge-request create [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --source-branch
                                           SOURCE_BRANCH --target-branch
                                           TARGET_BRANCH --title TITLE
                                           [--allow-collaboration ALLOW_COLLABORATION]
                                           [--allow-maintainer-to-push ALLOW_MAINTAINER_
↪TO_PUSH]
                                           [--approvals-before-merge APPROVALS_BEFORE_
↪MERGE]
                                           [--assignee-id ASSIGNEE_ID]
                                           [--assignee-ids ASSIGNEE_IDS]
                                           [--description DESCRIPTION]
                                           [--labels LABELS]
                                           [--milestone-id MILESTONE_ID]
                                           [--remove-source-branch REMOVE_SOURCE_BRANCH]
                                           [--reviewer-ids REVIEWER_IDS]
                                           [--squash SQUASH]
                                           [--target-project-id TARGET_PROJECT_ID]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--source-branch** <source\_branch>

**--target-branch** <target\_branch>

**--title** <title>

**--allow-collaboration** <allow\_collaboration>

**--allow-maintainer-to-push** <allow\_maintainer\_to\_push>

**--approvals-before-merge** <approvals\_before\_merge>

**--assignee-id** <assignee\_id>

**--assignee-ids** <assignee\_ids>  
**--description** <description>  
**--labels** <labels>  
**--milestone-id** <milestone\_id>  
**--remove-source-branch** <remove\_source\_branch>  
**--reviewer-ids** <reviewer\_ids>  
**--squash** <squash>  
**--target-project-id** <target\_project\_id>

### gitlab project-merge-request delete

```
usage: gitlab project-merge-request delete [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --iid IID
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--project-id** <project\_id>  
**--iid** <iid>

### gitlab project-merge-request get

```
usage: gitlab project-merge-request get [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --iid IID
                                         [--render-html RENDER_HTML]
                                         [--include-diverged-commits-count INCLUDE_
↪DIVERGED_COMMITS_COUNT]
                                         [--include-rebase-in-progress INCLUDE_REBASE_IN_
↪PROGRESS]
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--project-id** <project\_id>  
**--iid** <iid>  
**--render-html** <render\_html>  
**--include-diverged-commits-count** <include\_diverged\_commits\_count>  
**--include-rebase-in-progress** <include\_rebase\_in\_progress>



**gitlab project-merge-request list**

```
usage: gitlab project-merge-request list [-h] [--sudo SUDO] --project-id
PROJECT_ID [--state STATE]
[--order-by ORDER_BY] [--sort SORT]
[--milestone MILESTONE] [--view VIEW]
[--labels LABELS]
[--created-after CREATED_AFTER]
[--created-before CREATED_BEFORE]
[--updated-after UPDATED_AFTER]
[--updated-before UPDATED_BEFORE]
[--scope SCOPE] [--iids IIDS]
[--author-id AUTHOR_ID]
[--assignee-id ASSIGNEE_ID]
[--approver-ids APPROVER_IDS]
[--approved-by-ids APPROVED_BY_IDS]
[--my-reaction-emoji MY_REACTION_EMOJI]
[--source-branch SOURCE_BRANCH]
[--target-branch TARGET_BRANCH]
[--search SEARCH] [--wip WIP]
[--page PAGE] [--per-page PER_PAGE]
[--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--state** <state>

**--order-by** <order\_by>

**--sort** <sort>

**--milestone** <milestone>

**--view** <view>

**--labels** <labels>

**--created-after** <created\_after>

**--created-before** <created\_before>

**--updated-after** <updated\_after>

**--updated-before** <updated\_before>

**--scope** <scope>

**--iids** <iids>

**--author-id** <author\_id>

**--assignee-id** <assignee\_id>

**--approver-ids** <approver\_ids>  
**--approved-by-ids** <approved\_by\_ids>  
**--my-reaction-emoji** <my\_reaction\_emoji>  
**--source-branch** <source\_branch>  
**--target-branch** <target\_branch>  
**--search** <search>  
**--wip** <wip>  
**--page** <page>  
**--per-page** <per\_page>  
**--get-all**  
Return all items from the server, without pagination.

### gitlab project-merge-request merge

```
usage: gitlab project-merge-request merge [-h] --project-id PROJECT_ID
                                         [--sudo SUDO] --iid IID
                                         [--merge-commit-message MERGE_COMMIT_MESSAGE]
                                         [--should-remove-source-branch SHOULD_REMOVE_]
                                         ↪SOURCE_BRANCH]
                                         [--merge-when-pipeline-succeeds MERGE_WHEN_]
                                         ↪PIPELINE_SUCCEEDS]
```

**-h, --help**  
show this help message and exit  
**--project-id** <project\_id>  
**--sudo** <sudo>  
**--iid** <iid>  
**--merge-commit-message** <merge\_commit\_message>  
**--should-remove-source-branch** <should\_remove\_source\_branch>  
**--merge-when-pipeline-succeeds** <merge\_when\_pipeline\_succeeds>

### gitlab project-merge-request merge-ref

```
usage: gitlab project-merge-request merge-ref [-h] --project-id PROJECT_ID
                                              [--sudo SUDO] --iid IID
```

**-h, --help**  
show this help message and exit  
**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-merge-request participants

```
usage: gitlab project-merge-request participants [-h] --project-id PROJECT_ID
                                              [--sudo SUDO] --iid IID
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-merge-request rebase

```
usage: gitlab project-merge-request rebase [-h] --project-id PROJECT_ID
                                           [--sudo SUDO] --iid IID
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-merge-request reset-approvals

```
usage: gitlab project-merge-request reset-approvals [-h] --project-id
                                                    PROJECT_ID [--sudo SUDO]
                                                    --iid IID
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-merge-request reset-spent-time

```
usage: gitlab project-merge-request reset-spent-time [-h] --project-id
                                                    PROJECT_ID [--sudo SUDO]
                                                    --iid IID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-merge-request reset-time-estimate

```
usage: gitlab project-merge-request reset-time-estimate [-h] --project-id
                                                         PROJECT_ID
                                                         [--sudo SUDO] --iid
                                                         IID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-merge-request subscribe

```
usage: gitlab project-merge-request subscribe [-h] --project-id PROJECT_ID
                                              [--sudo SUDO] --iid IID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-merge-request time-estimate

```
usage: gitlab project-merge-request time-estimate [-h] --project-id PROJECT_ID
                                                  [--sudo SUDO] --iid IID
                                                  --duration DURATION
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

**--duration** <duration>

### gitlab project-merge-request time-stats

```
usage: gitlab project-merge-request time-stats [-h] --project-id PROJECT_ID
                                                [--sudo SUDO] --iid IID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-merge-request todo

```
usage: gitlab project-merge-request todo [-h] --project-id PROJECT_ID
                                          [--sudo SUDO] --iid IID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-merge-request unapprove

```
usage: gitlab project-merge-request unapprove [-h] --project-id PROJECT_ID
                                              [--sudo SUDO] --iid IID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-merge-request unsubscribe

```
usage: gitlab project-merge-request unsubscribe [-h] --project-id PROJECT_ID
                                              [--sudo SUDO] --iid IID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--iid** <iid>

### gitlab project-merge-request update

```
usage: gitlab project-merge-request update [-h] [--sudo SUDO] --project-id
PROJECT_ID --iid IID
      [--target-branch TARGET_BRANCH]
      [--assignee-id ASSIGNEE_ID]
      [--title TITLE]
      [--description DESCRIPTION]
      [--state-event STATE_EVENT]
      [--labels LABELS]
      [--milestone-id MILESTONE_ID]
      [--remove-source-branch REMOVE_SOURCE_BRANCH]
      [--discussion-locked DISCUSSION_LOCKED]
      [--allow-maintainer-to-push ALLOW_MAINTAINER_
↪ TO_PUSH]
      [--squash SQUASH]
      [--reviewer-ids REVIEWER_IDS]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

```

--iid <iid>
--target-branch <target_branch>
--assignee-id <assignee_id>
--title <title>
--description <description>
--state-event <state_event>
--labels <labels>
--milestone-id <milestone_id>
--remove-source-branch <remove_source_branch>
--discussion-locked <discussion_locked>
--allow-maintainer-to-push <allow_maintainer_to_push>
--squash <squash>
--reviewer-ids <reviewer_ids>

```

### gitlab project-merge-request-approval

```
usage: gitlab project-merge-request-approval [-h] {get,update} ...
```

```

-h, --help
    show this help message and exit

```

### gitlab project-merge-request-approval get

```

usage: gitlab project-merge-request-approval get [-h] [--sudo SUDO]
                                         --project-id PROJECT_ID
                                         --mr-iid MR_IID

```

```

-h, --help
    show this help message and exit

--sudo <sudo>
--project-id <project_id>
--mr-iid <mr_iid>

```

### gitlab project-merge-request-approval update

```
usage: gitlab project-merge-request-approval update [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --mr-iid MR_IID
                                                    --approvals-required
                                                    APPROVALS_REQUIRED
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--approvals-required** <approvals\_required>

### gitlab project-merge-request-approval-rule

```
usage: gitlab project-merge-request-approval-rule [-h]
list
get
create
update
delete
...
```

**-h, --help**

show this help message and exit

### gitlab project-merge-request-approval-rule create

```
usage: gitlab project-merge-request-approval-rule create [-h] [--sudo SUDO]
                                                         --project-id
                                                         PROJECT_ID --mr-iid
                                                         MR_IID --id ID
                                                         --merge-request-iid
                                                         MERGE_REQUEST_IID
                                                         --name NAME
                                                         --approvals-required
                                                         APPROVALS_REQUIRED
                                                         [--approval-project-rule-id_
→APPROVAL_PROJECT_RULE_ID]
                                                         [--user-ids USER_IDS]
                                                         [--group-ids GROUP_IDS]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>



```

--project-id <project_id>
--mr-iid <mr_iid>
--id <id>
--merge-request-iid <merge_request_iid>
--name <name>
--approvals-required <approvals_required>
--approval-project-rule-id <approval_project_rule_id>
--user-ids <user_ids>
--group-ids <group_ids>

```

### gitlab project-merge-request-approval-rule delete

```

usage: gitlab project-merge-request-approval-rule delete [-h] [--sudo SUDO]
                                                    --project-id
                                                    PROJECT_ID --mr-iid
                                                    MR_IID --id ID

```

```

-h, --help
    show this help message and exit
--sudo <sudo>
--project-id <project_id>
--mr-iid <mr_iid>
--id <id>

```

### gitlab project-merge-request-approval-rule get

```

usage: gitlab project-merge-request-approval-rule get [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --mr-iid MR_IID --id ID

```

```

-h, --help
    show this help message and exit
--sudo <sudo>
--project-id <project_id>
--mr-iid <mr_iid>
--id <id>

```

### gitlab project-merge-request-approval-rule list

```
usage: gitlab project-merge-request-approval-rule list [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --mr-iid MR_IID
                                                    [--page PAGE]
                                                    [--per-page PER_PAGE]
                                                    [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-merge-request-approval-rule update

```
usage: gitlab project-merge-request-approval-rule update [-h] [--sudo SUDO]
                                                    --project-id
                                                    PROJECT_ID --mr-iid
                                                    MR_IID --id ID
                                                    --merge-request-iid
                                                    MERGE_REQUEST_IID
                                                    --approval-rule-id
                                                    APPROVAL_RULE_ID
                                                    --name NAME
                                                    --approvals-required
                                                    APPROVALS_REQUIRED
                                                    [--user-ids USER_IDS]
                                                    [--group-ids GROUP_IDS]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--id** <id>

**--merge-request-iid** <merge\_request\_iid>

**--approval-rule-id** <approval\_rule\_id>

```

--name <name>
--approvals-required <approvals_required>
--user-ids <user_ids>
--group-ids <group_ids>

```

### gitlab project-merge-request-approval-state

```
usage: gitlab project-merge-request-approval-state [-h] {get} ...
```

**-h, --help**  
show this help message and exit

#### gitlab project-merge-request-approval-state get

```
usage: gitlab project-merge-request-approval-state get [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --mr-iid MR_IID --id ID
```

**-h, --help**  
show this help message and exit

```

--sudo <sudo>
--project-id <project_id>
--mr-iid <mr_iid>
--id <id>

```

### gitlab project-merge-request-award-emoji

```
usage: gitlab project-merge-request-award-emoji [-h]
list
get
create
delete
...
```

**-h, --help**  
show this help message and exit

### gitlab project-merge-request-award-emoji create

```
usage: gitlab project-merge-request-award-emoji create [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --mr-iid MR_IID --name
                                                    NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--name** <name>

### gitlab project-merge-request-award-emoji delete

```
usage: gitlab project-merge-request-award-emoji delete [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --mr-iid MR_IID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--id** <id>

### gitlab project-merge-request-award-emoji get

```
usage: gitlab project-merge-request-award-emoji get [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --mr-iid MR_IID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--id** <id>

### gitlab project-merge-request-award-emoji list

```
usage: gitlab project-merge-request-award-emoji list [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --mr-iid MR_IID
                                                    [--page PAGE]
                                                    [--per-page PER_PAGE]
                                                    [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-merge-request-diff

```
usage: gitlab project-merge-request-diff [-h] {list,get} ...
```

**-h, --help**

show this help message and exit

### gitlab project-merge-request-diff get

```
usage: gitlab project-merge-request-diff get [-h] [--sudo SUDO] --project-id
PROJECT_ID --mr-iid MR_IID --id
ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--id** <id>

### gitlab project-merge-request-diff list

```
usage: gitlab project-merge-request-diff list [-h] [--sudo SUDO] --project-id
                                             PROJECT_ID --mr-iid MR_IID
                                             [--page PAGE]
                                             [--per-page PER_PAGE]
                                             [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-merge-request-discussion

```
usage: gitlab project-merge-request-discussion [-h]
list
get
create
update
...
```

**-h, --help**

show this help message and exit

### gitlab project-merge-request-discussion create

```
usage: gitlab project-merge-request-discussion create [-h] [--sudo SUDO]
--project-id PROJECT_ID
--mr-iid MR_IID --body
BODY
[--created-at CREATED_AT]
[--position POSITION]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--body** <body>  
**--created-at** <created\_at>  
**--position** <position>

### gitlab project-merge-request-discussion get

```
usage: gitlab project-merge-request-discussion get [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --mr-iid MR_IID --id ID
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--project-id** <project\_id>  
**--mr-iid** <mr\_iid>  
**--id** <id>

### gitlab project-merge-request-discussion list

```
usage: gitlab project-merge-request-discussion list [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --mr-iid MR_IID
                                                    [--page PAGE]
                                                    [--per-page PER_PAGE]
                                                    [--get-all]
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--project-id** <project\_id>  
**--mr-iid** <mr\_iid>  
**--page** <page>  
**--per-page** <per\_page>  
**--get-all**  
Return all items from the server, without pagination.

### gitlab project-merge-request-discussion update

```
usage: gitlab project-merge-request-discussion update [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --mr-iid MR_IID --id ID
                                                    --resolved RESOLVED
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--id** <id>

**--resolved** <resolved>

### gitlab project-merge-request-discussion-note

```
usage: gitlab project-merge-request-discussion-note [-h]
get
create
update
delete
...
```

**-h, --help**  
show this help message and exit

### gitlab project-merge-request-discussion-note create

```
usage: gitlab project-merge-request-discussion-note create [-h] [--sudo SUDO]
                                                    --project-id
PROJECT_ID --mr-iid
MR_IID
--discussion-id
DISCUSSION_ID
--body BODY
[--created-at CREATED_AT]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--discussion-id** <discussion\_id>



**--body** <body>

**--created-at** <created\_at>

### gitlab project-merge-request-discussion-note delete

```
usage: gitlab project-merge-request-discussion-note delete [-h] [--sudo SUDO]
                                     --project-id
                                     PROJECT_ID --mr-iid
                                     MR_IID
                                     --discussion-id
                                     DISCUSSION_ID --id
                                     ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--discussion-id** <discussion\_id>

**--id** <id>

### gitlab project-merge-request-discussion-note get

```
usage: gitlab project-merge-request-discussion-note get [-h] [--sudo SUDO]
                                     --project-id
                                     PROJECT_ID --mr-iid
                                     MR_IID --discussion-id
                                     DISCUSSION_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--discussion-id** <discussion\_id>

**--id** <id>

### gitlab project-merge-request-discussion-note update

```
usage: gitlab project-merge-request-discussion-note update [-h] [--sudo SUDO]
                                     --project-id
                                     PROJECT_ID --mr-iid
                                     MR_IID
                                     --discussion-id
                                     DISCUSSION_ID --id
                                     ID --body BODY
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--discussion-id** <discussion\_id>

**--id** <id>

**--body** <body>

### gitlab project-merge-request-draft-note

```
usage: gitlab project-merge-request-draft-note [-h]
  list
  get
  create
  update
  delete
```

...

**-h, --help**

show this help message and exit

### gitlab project-merge-request-draft-note create

```
usage: gitlab project-merge-request-draft-note create [-h] [--sudo SUDO]
                                     --project-id PROJECT_ID
                                     --mr-iid MR_IID --note
                                     NOTE
                                     [--commit-id COMMIT_ID]
                                     [--in-reply-to-discussion-id IN_
↳REPLY_TO_DISCUSSION_ID]
                                     [--position POSITION]
                                     [--resolve-discussion RESOLVE_
↳DISCUSSION]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--note** <note>

**--commit-id** <commit\_id>

**--in-reply-to-discussion-id** <in\_reply\_to\_discussion\_id>

**--position** <position>

**--resolve-discussion** <resolve\_discussion>

### gitlab project-merge-request-draft-note delete

```
usage: gitlab project-merge-request-draft-note delete [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --mr-iid MR_IID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--id** <id>

### gitlab project-merge-request-draft-note get

```
usage: gitlab project-merge-request-draft-note get [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --mr-iid MR_IID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--id** <id>

### gitlab project-merge-request-draft-note list

```
usage: gitlab project-merge-request-draft-note list [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --mr-iid MR_IID
                                                    [--page PAGE]
                                                    [--per-page PER_PAGE]
                                                    [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-merge-request-draft-note update

```
usage: gitlab project-merge-request-draft-note update [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --mr-iid MR_IID --id ID
                                                    [--position POSITION]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--id** <id>

**--position** <position>

### gitlab project-merge-request-note

```
usage: gitlab project-merge-request-note [-h]
list
get
create
update
delete
...
```

**-h, --help**

show this help message and exit

### gitlab project-merge-request-note create

```
usage: gitlab project-merge-request-note create [-h] [--sudo SUDO]
                                              --project-id PROJECT_ID
                                              --mr-iid MR_IID --body BODY
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--body** <body>

### gitlab project-merge-request-note delete

```
usage: gitlab project-merge-request-note delete [-h] [--sudo SUDO]
                                              --project-id PROJECT_ID
                                              --mr-iid MR_IID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--id** <id>

### gitlab project-merge-request-note get

```
usage: gitlab project-merge-request-note get [-h] [--sudo SUDO] --project-id
                                              PROJECT_ID --mr-iid MR_IID --id
                                              ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--id** <id>

### gitlab project-merge-request-note list

```
usage: gitlab project-merge-request-note list [-h] [--sudo SUDO] --project-id
                                             PROJECT_ID --mr-iid MR_IID
                                             [--page PAGE]
                                             [--per-page PER_PAGE]
                                             [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-merge-request-note update

```
usage: gitlab project-merge-request-note update [-h] [--sudo SUDO]
                                                --project-id PROJECT_ID
                                                --mr-iid MR_IID --id ID --body
                                                BODY
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--id** <id>

**--body** <body>

### gitlab project-merge-request-note-award-emoji

```
usage: gitlab project-merge-request-note-award-emoji [-h]
list
get
create
delete
...
```

**-h, --help**

show this help message and exit

### gitlab project-merge-request-note-award-emoji create

```
usage: gitlab project-merge-request-note-award-emoji create
       [-h] [--sudo SUDO] --project-id PROJECT_ID --mr-iid MR_IID --note-id
       NOTE_ID --name NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--note-id** <note\_id>

**--name** <name>

### gitlab project-merge-request-note-award-emoji delete

```
usage: gitlab project-merge-request-note-award-emoji delete
       [-h] [--sudo SUDO] --project-id PROJECT_ID --mr-iid MR_IID --note-id
       NOTE_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--note-id** <note\_id>

**--id** <id>

### gitlab project-merge-request-note-award-emoji get

```
usage: gitlab project-merge-request-note-award-emoji get [-h] [--sudo SUDO]
       --project-id PROJECT_ID --mr-iid MR_IID --note-id NOTE_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--note-id** <note\_id>

**--id** <id>

### gitlab project-merge-request-note-award-emoji list

```
usage: gitlab project-merge-request-note-award-emoji list [-h] [--sudo SUDO]
                                                           --project-id
                                                           PROJECT_ID --mr-iid
                                                           MR_IID --note-id
                                                           NOTE_ID
                                                           [--page PAGE]
                                                           [--per-page PER_PAGE]
                                                           [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--note-id** <note\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-merge-request-pipeline

```
usage: gitlab project-merge-request-pipeline [-h] {list,create} ...
```

**-h, --help**

show this help message and exit

### gitlab project-merge-request-pipeline create

```
usage: gitlab project-merge-request-pipeline create [-h] [--sudo SUDO]
                                                           --project-id PROJECT_ID
                                                           --mr-iid MR_IID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>



### gitlab project-merge-request-pipeline list

```
usage: gitlab project-merge-request-pipeline list [-h] [--sudo SUDO]
                                                --project-id PROJECT_ID
                                                --mr-iid MR_IID
                                                [--page PAGE]
                                                [--per-page PER_PAGE]
                                                [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--project-id** <project\_id>**--mr-iid** <mr\_iid>**--page** <page>**--per-page** <per\_page>**--get-all**

Return all items from the server, without pagination.

### gitlab project-merge-request-resource-label-event

```
usage: gitlab project-merge-request-resource-label-event [-h] {list,get} ...
```

**-h, --help**

show this help message and exit

### gitlab project-merge-request-resource-label-event get

```
usage: gitlab project-merge-request-resource-label-event get
       [-h] [--sudo SUDO] --project-id PROJECT_ID --mr-iid MR_IID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--project-id** <project\_id>**--mr-iid** <mr\_iid>**--id** <id>

### gitlab project-merge-request-resource-label-event list

```
usage: gitlab project-merge-request-resource-label-event list
       [-h] [--sudo SUDO] --project-id PROJECT_ID --mr-iid MR_IID
       [--page PAGE] [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-merge-request-resource-milestone-event

```
usage: gitlab project-merge-request-resource-milestone-event
       [-h] {list,get} ...
```

**-h, --help**

show this help message and exit

### gitlab project-merge-request-resource-milestone-event get

```
usage: gitlab project-merge-request-resource-milestone-event get
       [-h] [--sudo SUDO] --project-id PROJECT_ID --mr-iid MR_IID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--id** <id>

### gitlab project-merge-request-resource-milestone-event list

```
usage: gitlab project-merge-request-resource-milestone-event list
       [-h] [--sudo SUDO] --project-id PROJECT_ID --mr-iid MR_IID
       [--page PAGE] [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-merge-request-resource-state-event

```
usage: gitlab project-merge-request-resource-state-event [-h] {list,get} ...
```

**-h, --help**

show this help message and exit

### gitlab project-merge-request-resource-state-event get

```
usage: gitlab project-merge-request-resource-state-event get
       [-h] [--sudo SUDO] --project-id PROJECT_ID --mr-iid MR_IID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--id** <id>

### gitlab project-merge-request-resource-state-event list

```
usage: gitlab project-merge-request-resource-state-event list
       [-h] [--sudo SUDO] --project-id PROJECT_ID --mr-iid MR_IID
       [--page PAGE] [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-merge-request-reviewer-detail

```
usage: gitlab project-merge-request-reviewer-detail [-h] {list} ...
```

**-h, --help**

show this help message and exit

### gitlab project-merge-request-reviewer-detail list

```
usage: gitlab project-merge-request-reviewer-detail list [-h] [--sudo SUDO]
                                     --project-id
                                     PROJECT_ID --mr-iid
                                     MR_IID [--page PAGE]
                                     [--per-page PER_PAGE]
                                     [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--mr-iid** <mr\_iid>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab project-merge-train

```
usage: gitlab project-merge-train [-h] {list} ...
```

### -h, --help

show this help message and exit

## gitlab project-merge-train list

```
usage: gitlab project-merge-train list [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID [--scope SCOPE]
                                         [--page PAGE] [--per-page PER_PAGE]
                                         [--get-all]
```

### -h, --help

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--scope** <scope>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab project-milestone

```
usage: gitlab project-milestone [-h]
    list
    get
    create
    update
    delete
    issues
    merge-requests
    ...
```

### -h, --help

show this help message and exit

### gitlab project-milestone create

```
usage: gitlab project-milestone create [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --title TITLE
                                         [--description DESCRIPTION]
                                         [--due-date DUE_DATE]
                                         [--start-date START_DATE]
                                         [--state-event STATE_EVENT]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--title** <title>

**--description** <description>

**--due-date** <due\_date>

**--start-date** <start\_date>

**--state-event** <state\_event>

### gitlab project-milestone delete

```
usage: gitlab project-milestone delete [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-milestone get

```
usage: gitlab project-milestone get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                     --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-milestone issues

```
usage: gitlab project-milestone issues [-h] --project-id PROJECT_ID
                                         [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

### gitlab project-milestone list

```
usage: gitlab project-milestone list [-h] [--sudo SUDO] --project-id
PROJECT_ID [--iids IIDS] [--state STATE]
        [--search SEARCH] [--page PAGE]
        [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--iids** <iids>

**--state** <state>

**--search** <search>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-milestone merge-requests

```
usage: gitlab project-milestone merge-requests [-h] --project-id PROJECT_ID
                                                [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

### gitlab project-milestone update

```
usage: gitlab project-milestone update [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --id ID [--title TITLE]
                                         [--description DESCRIPTION]
                                         [--due-date DUE_DATE]
                                         [--start-date START_DATE]
                                         [--state-event STATE_EVENT]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

**--title** <title>

**--description** <description>

**--due-date** <due\_date>

**--start-date** <start\_date>

**--state-event** <state\_event>

### gitlab project-note

```
usage: gitlab project-note [-h] {list,get} ...
```

**-h, --help**  
show this help message and exit

#### gitlab project-note get

```
usage: gitlab project-note get [-h] [--sudo SUDO] --project-id PROJECT_ID --id
                                ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>



## gitlab project-note list

```
usage: gitlab project-note list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                [--page PAGE] [--per-page PER_PAGE]
                                [--get-all]
```

### -h, --help

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

### --get-all

Return all items from the server, without pagination.

## gitlab project-notification-settings

```
usage: gitlab project-notification-settings [-h] {get,update} ...
```

### -h, --help

show this help message and exit

## gitlab project-notification-settings get

```
usage: gitlab project-notification-settings get [-h] [--sudo SUDO]
                                                --project-id PROJECT_ID
```

### -h, --help

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

## gitlab project-notification-settings update

```
usage: gitlab project-notification-settings update [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    [--level LEVEL]
                                                    [--notification-email NOTIFICATION_
↪EMAIL]
                                                    [--new-note NEW_NOTE]
                                                    [--new-issue NEW_ISSUE]
                                                    [--reopen-issue REOPEN_ISSUE]
                                                    [--close-issue CLOSE_ISSUE]
                                                    [--reassign-issue REASSIGN_ISSUE]
```

(continues on next page)

(continued from previous page)

<code>↪REQUEST]</code>	<code>[--new-merge-request NEW_MERGE_</code>
<code>↪REQUEST]</code>	<code>[--reopen-merge-request REOPEN_MERGE_</code>
<code>↪REQUEST]</code>	<code>[--close-merge-request CLOSE_MERGE_</code>
<code>↪MERGE_REQUEST]</code>	<code>[--reassign-merge-request REASSIGN_</code>
<code>↪REQUEST]</code>	<code>[--merge-merge-request MERGE_MERGE_</code>

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--project-id** <project\_id>**--level** <level>**--notification-email** <notification\_email>**--new-note** <new\_note>**--new-issue** <new\_issue>**--reopen-issue** <reopen\_issue>**--close-issue** <close\_issue>**--reassign-issue** <reassign\_issue>**--new-merge-request** <new\_merge\_request>**--reopen-merge-request** <reopen\_merge\_request>**--close-merge-request** <close\_merge\_request>**--reassign-merge-request** <reassign\_merge\_request>**--merge-merge-request** <merge\_merge\_request>**gitlab project-package**

usage: gitlab project-package [-h] {list,get,delete} ...

**-h, --help**

show this help message and exit

### gitlab project-package delete

```
usage: gitlab project-package delete [-h] [--sudo SUDO] --project-id
PROJECT_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-package get

```
usage: gitlab project-package get [-h] [--sudo SUDO] --project-id PROJECT_ID
--id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-package list

```
usage: gitlab project-package list [-h] [--sudo SUDO] --project-id PROJECT_ID
[--order-by ORDER_BY] [--sort SORT]
[--package-type PACKAGE_TYPE]
[--package-name PACKAGE_NAME] [--page PAGE]
[--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--order-by** <order\_by>

**--sort** <sort>

**--package-type** <package\_type>

**--package-name** <package\_name>

**--page** <page>

**--per-page** <per\_page>

### **--get-all**

Return all items from the server, without pagination.

## **gitlab project-package-file**

```
usage: gitlab project-package-file [-h] {list,delete} ...
```

### **-h, --help**

show this help message and exit

## **gitlab project-package-file delete**

```
usage: gitlab project-package-file delete [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --package-id PACKAGE_ID
                                           --id ID
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--package-id** <package\_id>

**--id** <id>

## **gitlab project-package-file list**

```
usage: gitlab project-package-file list [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --package-id PACKAGE_ID
                                           [--page PAGE] [--per-page PER_PAGE]
                                           [--get-all]
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--package-id** <package\_id>

**--page** <page>

**--per-page** <per\_page>

### **--get-all**

Return all items from the server, without pagination.

## gitlab project-package-pipeline

```
usage: gitlab project-package-pipeline [-h] {list} ...
```

**-h, --help**

show this help message and exit

## gitlab project-package-pipeline list

```
usage: gitlab project-package-pipeline list [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --package-id PACKAGE_ID
                                           [--page PAGE]
                                           [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--package-id** <package\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab project-pages-domain

```
usage: gitlab project-pages-domain [-h] {list,get,create,update,delete} ...
```

**-h, --help**

show this help message and exit

## gitlab project-pages-domain create

```
usage: gitlab project-pages-domain create [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --domain DOMAIN
                                           [--certificate CERTIFICATE]
                                           [--key KEY]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--domain** <domain>  
**--certificate** <certificate>  
**--key** <key>

### gitlab project-pages-domain delete

```
usage: gitlab project-pages-domain delete [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --domain DOMAIN
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--project-id** <project\_id>  
**--domain** <domain>

### gitlab project-pages-domain get

```
usage: gitlab project-pages-domain get [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --domain DOMAIN
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--project-id** <project\_id>  
**--domain** <domain>

### gitlab project-pages-domain list

```
usage: gitlab project-pages-domain list [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID [--page PAGE]
                                         [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--project-id** <project\_id>  
**--page** <page>  
**--per-page** <per\_page>  
**--get-all**  
Return all items from the server, without pagination.

### gitlab project-pages-domain update

```
usage: gitlab project-pages-domain update [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --domain DOMAIN
                                           [--certificate CERTIFICATE]
                                           [--key KEY]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--domain** <domain>

**--certificate** <certificate>

**--key** <key>

### gitlab project-pipeline

```
usage: gitlab project-pipeline [-h] {list,get,create,delete,cancel,retry} ...
```

**-h, --help**  
show this help message and exit

### gitlab project-pipeline cancel

```
usage: gitlab project-pipeline cancel [-h] --project-id PROJECT_ID
                                       [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

### gitlab project-pipeline create

```
usage: gitlab project-pipeline create [-h] [--sudo SUDO] --project-id
                                       PROJECT_ID --ref REF
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--ref** <ref>

### gitlab project-pipeline delete

```
usage: gitlab project-pipeline delete [-h] [--sudo SUDO] --project-id
PROJECT_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-pipeline get

```
usage: gitlab project-pipeline get [-h] [--sudo SUDO] --project-id PROJECT_ID
--id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-pipeline list

```
usage: gitlab project-pipeline list [-h] [--sudo SUDO] --project-id PROJECT_ID
[--scope SCOPE] [--status STATUS]
[--source SOURCE] [--ref REF] [--sha SHA]
[--yaml-errors YAML_ERRORS] [--name NAME]
[--username USERNAME]
[--order-by ORDER_BY] [--sort SORT]
[--page PAGE] [--per-page PER_PAGE]
[--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--scope** <scope>

**--status** <status>

**--source** <source>

**--ref** <ref>



**--sha** <sha>  
**--yaml-errors** <yaml\_errors>  
**--name** <name>  
**--username** <username>  
**--order-by** <order\_by>  
**--sort** <sort>  
**--page** <page>  
**--per-page** <per\_page>  
**--get-all**

Return all items from the server, without pagination.

### gitlab project-pipeline retry

```
usage: gitlab project-pipeline retry [-h] --project-id PROJECT_ID
                                     [--sudo SUDO] --id ID
```

**-h, --help**  
 show this help message and exit  
**--project-id** <project\_id>  
**--sudo** <sudo>  
**--id** <id>

### gitlab project-pipeline-bridge

```
usage: gitlab project-pipeline-bridge [-h] {list} ...
```

**-h, --help**  
 show this help message and exit

### gitlab project-pipeline-bridge list

```
usage: gitlab project-pipeline-bridge list [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --pipeline-id
                                           PIPELINE_ID [--scope SCOPE]
                                           [--page PAGE] [--per-page PER_PAGE]
                                           [--get-all]
```

**-h, --help**  
 show this help message and exit  
**--sudo** <sudo>

**--project-id** <project\_id>

**--pipeline-id** <pipeline\_id>

**--scope** <scope>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-pipeline-job

```
usage: gitlab project-pipeline-job [-h] {list} ...
```

**-h, --help**

show this help message and exit

### gitlab project-pipeline-job list

```
usage: gitlab project-pipeline-job list [-h] [--sudo SUDO] --project-id
PROJECT_ID --pipeline-id PIPELINE_ID
[--scope SCOPE]
[--include-retried INCLUDE_RETRIED]
[--page PAGE] [--per-page PER_PAGE]
[--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--pipeline-id** <pipeline\_id>

**--scope** <scope>

**--include-retried** <include\_retried>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab project-pipeline-schedule

```
usage: gitlab project-pipeline-schedule [-h]
      list
      get
      create
      update
      delete
      take-ownership
      play
      ...
```

**-h, --help**  
show this help message and exit

## gitlab project-pipeline-schedule create

```
usage: gitlab project-pipeline-schedule create [-h] [--sudo SUDO] --project-id
      PROJECT_ID --description
      DESCRIPTION --ref REF --cron
      CRON
      [--cron-timezone CRON_TIMEZONE]
      [--active ACTIVE]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--description** <description>

**--ref** <ref>

**--cron** <cron>

**--cron-timezone** <cron\_timezone>

**--active** <active>

## gitlab project-pipeline-schedule delete

```
usage: gitlab project-pipeline-schedule delete [-h] [--sudo SUDO] --project-id
      PROJECT_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-pipeline-schedule get

```
usage: gitlab project-pipeline-schedule get [-h] [--sudo SUDO] --project-id
      PROJECT_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-pipeline-schedule list

```
usage: gitlab project-pipeline-schedule list [-h] [--sudo SUDO] --project-id
      PROJECT_ID [--page PAGE]
      [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-pipeline-schedule play

```
usage: gitlab project-pipeline-schedule play [-h] --project-id PROJECT_ID
      [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

### gitlab project-pipeline-schedule take-ownership

```
usage: gitlab project-pipeline-schedule take-ownership [-h] --project-id
                                                    PROJECT_ID
                                                    [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

### gitlab project-pipeline-schedule update

```
usage: gitlab project-pipeline-schedule update [-h] [--sudo SUDO] --project-id
                                                    PROJECT_ID --id ID
                                                    [--description DESCRIPTION]
                                                    [--ref REF] [--cron CRON]
                                                    [--cron-timezone CRON_TIMEZONE]
                                                    [--active ACTIVE]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

**--description** <description>

**--ref** <ref>

**--cron** <cron>

**--cron-timezone** <cron\_timezone>

**--active** <active>

### gitlab project-pipeline-schedule-pipeline

```
usage: gitlab project-pipeline-schedule-pipeline [-h] {list} ...
```

**-h, --help**  
show this help message and exit

### gitlab project-pipeline-schedule-pipeline list

```
usage: gitlab project-pipeline-schedule-pipeline list [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --pipeline-schedule-id
PIPELINE_SCHEDULE_ID
                                                    [--page PAGE]
                                                    [--per-page PER_PAGE]
                                                    [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--pipeline-schedule-id** <pipeline\_schedule\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-pipeline-schedule-variable

```
usage: gitlab project-pipeline-schedule-variable [-h]
create
update
delete
...
```

**-h, --help**

show this help message and exit

### gitlab project-pipeline-schedule-variable create

```
usage: gitlab project-pipeline-schedule-variable create [-h] [--sudo SUDO]
                                                    --project-id
PROJECT_ID
                                                    --pipeline-schedule-id
PIPELINE_SCHEDULE_ID
                                                    --key KEY --value
VALUE
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>  
**--pipeline-schedule-id** <pipeline\_schedule\_id>  
**--key** <key>  
**--value** <value>

### gitlab project-pipeline-schedule-variable delete

```
usage: gitlab project-pipeline-schedule-variable delete [-h] [--sudo SUDO]
                                                    --project-id
                                                    PROJECT_ID
                                                    --pipeline-schedule-id
                                                    PIPELINE_SCHEDULE_ID
                                                    --key KEY
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--project-id** <project\_id>  
**--pipeline-schedule-id** <pipeline\_schedule\_id>  
**--key** <key>

### gitlab project-pipeline-schedule-variable update

```
usage: gitlab project-pipeline-schedule-variable update [-h] [--sudo SUDO]
                                                    --project-id
                                                    PROJECT_ID
                                                    --pipeline-schedule-id
                                                    PIPELINE_SCHEDULE_ID
                                                    --key KEY --value
                                                    VALUE
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--project-id** <project\_id>  
**--pipeline-schedule-id** <pipeline\_schedule\_id>  
**--key** <key>  
**--value** <value>

### gitlab project-pipeline-test-report

```
usage: gitlab project-pipeline-test-report [-h] {get} ...
```

**-h, --help**

show this help message and exit

### gitlab project-pipeline-test-report get

```
usage: gitlab project-pipeline-test-report get [-h] [--sudo SUDO] --project-id
                                                PROJECT_ID --pipeline-id
                                                PIPELINE_ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--pipeline-id** <pipeline\_id>

### gitlab project-pipeline-test-report-summary

```
usage: gitlab project-pipeline-test-report-summary [-h] {get} ...
```

**-h, --help**

show this help message and exit

### gitlab project-pipeline-test-report-summary get

```
usage: gitlab project-pipeline-test-report-summary get [-h] [--sudo SUDO]
                                                         --project-id PROJECT_ID
                                                         --pipeline-id
                                                         PIPELINE_ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--pipeline-id** <pipeline\_id>



## gitlab project-pipeline-variable

```
usage: gitlab project-pipeline-variable [-h] {list} ...
```

**-h, --help**

show this help message and exit

## gitlab project-pipeline-variable list

```
usage: gitlab project-pipeline-variable list [-h] [--sudo SUDO] --project-id
      PROJECT_ID --pipeline-id
      PIPELINE_ID [--page PAGE]
      [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--pipeline-id** <pipeline\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab project-protected-branch

```
usage: gitlab project-protected-branch [-h]
      list
      get
      create
      update
      delete
      ...
```

**-h, --help**

show this help message and exit

### gitlab project-protected-branch create

```
usage: gitlab project-protected-branch create [-h] [--sudo SUDO] --project-id
PROJECT_ID --name NAME
[--push-access-level PUSH_ACCESS_LEVEL]
[--merge-access-level MERGE_ACCESS_LEVEL]
[--unprotect-access-level UNPROTECT_ACCESS_
↪LEVEL]
[--allow-force-push ALLOW_FORCE_PUSH]
[--allowed-to-push ALLOWED_TO_PUSH]
[--allowed-to-merge ALLOWED_TO_MERGE]
[--allowed-to-unprotect ALLOWED_TO_
↪UNPROTECT]
[--code-owner-approval-required CODE_OWNER_
↪APPROVAL_REQUIRED]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

**--push-access-level** <push\_access\_level>

**--merge-access-level** <merge\_access\_level>

**--unprotect-access-level** <unprotect\_access\_level>

**--allow-force-push** <allow\_force\_push>

**--allowed-to-push** <allowed\_to\_push>

**--allowed-to-merge** <allowed\_to\_merge>

**--allowed-to-unprotect** <allowed\_to\_unprotect>

**--code-owner-approval-required** <code\_owner\_approval\_required>

### gitlab project-protected-branch delete

```
usage: gitlab project-protected-branch delete [-h] [--sudo SUDO] --project-id
PROJECT_ID --name NAME
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

### gitlab project-protected-branch get

```
usage: gitlab project-protected-branch get [-h] [--sudo SUDO] --project-id
PROJECT_ID --name NAME
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

### gitlab project-protected-branch list

```
usage: gitlab project-protected-branch list [-h] [--sudo SUDO] --project-id
PROJECT_ID [--page PAGE]
[--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-protected-branch update

```
usage: gitlab project-protected-branch update [-h] [--sudo SUDO] --project-id
PROJECT_ID --name NAME
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

## gitlab project-protected-environment

```
usage: gitlab project-protected-environment [-h] {list,get,create,delete} ...
```

**-h, --help**

show this help message and exit

## gitlab project-protected-environment create

```
usage: gitlab project-protected-environment create [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --name NAME
                                                    --deploy-access-levels
DEPLOY_ACCESS_LEVELS
                                                    [--required-approval-count REQUIRED_
↪APPROVAL_COUNT]
                                                    [--approval-rules APPROVAL_RULES]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

**--deploy-access-levels** <deploy\_access\_levels>

**--required-approval-count** <required\_approval\_count>

**--approval-rules** <approval\_rules>

## gitlab project-protected-environment delete

```
usage: gitlab project-protected-environment delete [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --name NAME
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

### gitlab project-protected-environment get

```
usage: gitlab project-protected-environment get [-h] [--sudo SUDO]
                                              --project-id PROJECT_ID --name
                                              NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

### gitlab project-protected-environment list

```
usage: gitlab project-protected-environment list [-h] [--sudo SUDO]
                                              --project-id PROJECT_ID
                                              [--page PAGE]
                                              [--per-page PER_PAGE]
                                              [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-protected-tag

```
usage: gitlab project-protected-tag [-h] {list,get,create,delete} ...
```

**-h, --help**  
show this help message and exit

### gitlab project-protected-tag create

```
usage: gitlab project-protected-tag create [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --name NAME
                                           [--create-access-level CREATE_ACCESS_LEVEL]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

**--create-access-level** <create\_access\_level>

### gitlab project-protected-tag delete

```
usage: gitlab project-protected-tag delete [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --name NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

### gitlab project-protected-tag get

```
usage: gitlab project-protected-tag get [-h] [--sudo SUDO] --project-id
                                          PROJECT_ID --name NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

### gitlab project-protected-tag list

```
usage: gitlab project-protected-tag list [-h] [--sudo SUDO] --project-id
PROJECT_ID [--page PAGE]
      [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-push-rules

```
usage: gitlab project-push-rules [-h] {get,create,update,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab project-push-rules create

```
usage: gitlab project-push-rules create [-h] [--sudo SUDO] --project-id
PROJECT_ID
      [--author-email-regex AUTHOR_EMAIL_REGEX]
      [--branch-name-regex BRANCH_NAME_REGEX]
      [--commit-committer-check COMMIT_COMMITTER_CHECK]
      [--commit-message-negative-regex COMMIT_MESSAGE_
↪NEGATIVE_REGEX]
      [--commit-message-regex COMMIT_MESSAGE_REGEX]
      [--deny-delete-tag DENY_DELETE_TAG]
      [--file-name-regex FILE_NAME_REGEX]
      [--max-file-size MAX_FILE_SIZE]
      [--member-check MEMBER_CHECK]
      [--prevent-secrets PREVENT_SECRETS]
      [--reject-unsigned-commits REJECT_UNSIGNED_
↪COMMITTS]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

```
--author-email-regex <author_email_regex>
--branch-name-regex <branch_name_regex>
--commit-committer-check <commit_committer_check>
--commit-message-negative-regex <commit_message_negative_regex>
--commit-message-regex <commit_message_regex>
--deny-delete-tag <deny_delete_tag>
--file-name-regex <file_name_regex>
--max-file-size <max_file_size>
--member-check <member_check>
--prevent-secrets <prevent_secrets>
--reject-unsigned-commits <reject_unsigned_commits>
```

### gitlab project-push-rules delete

```
usage: gitlab project-push-rules delete [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID
```

```
-h, --help
    show this help message and exit
--sudo <sudo>
--project-id <project_id>
```

### gitlab project-push-rules get

```
usage: gitlab project-push-rules get [-h] [--sudo SUDO] --project-id
                                       PROJECT_ID
```

```
-h, --help
    show this help message and exit
--sudo <sudo>
--project-id <project_id>
```



## gitlab project-push-rules update

```
usage: gitlab project-push-rules update [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID
                                         [--author-email-regex AUTHOR_EMAIL_REGEX]
                                         [--branch-name-regex BRANCH_NAME_REGEX]
                                         [--commit-committer-check COMMIT_COMMITTER_CHECK]
                                         [--commit-message-negative-regex COMMIT_MESSAGE_
↪NEGATIVE_REGEX]
                                         [--commit-message-regex COMMIT_MESSAGE_REGEX]
                                         [--deny-delete-tag DENY_DELETE_TAG]
                                         [--file-name-regex FILE_NAME_REGEX]
                                         [--max-file-size MAX_FILE_SIZE]
                                         [--member-check MEMBER_CHECK]
                                         [--prevent-secrets PREVENT_SECRETS]
                                         [--reject-unsigned-commits REJECT_UNSIGNED_
↪COMMITTS]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--author-email-regex** <author\_email\_regex>

**--branch-name-regex** <branch\_name\_regex>

**--commit-committer-check** <commit\_committer\_check>

**--commit-message-negative-regex** <commit\_message\_negative\_regex>

**--commit-message-regex** <commit\_message\_regex>

**--deny-delete-tag** <deny\_delete\_tag>

**--file-name-regex** <file\_name\_regex>

**--max-file-size** <max\_file\_size>

**--member-check** <member\_check>

**--prevent-secrets** <prevent\_secrets>

**--reject-unsigned-commits** <reject\_unsigned\_commits>

## gitlab project-registry-repository

```
usage: gitlab project-registry-repository [-h] {list,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab project-registry-repository delete

```
usage: gitlab project-registry-repository delete [-h] [--sudo SUDO]
                                                --project-id PROJECT_ID --id
                                                ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-registry-repository list

```
usage: gitlab project-registry-repository list [-h] [--sudo SUDO] --project-id
                                                PROJECT_ID [--page PAGE]
                                                [--per-page PER_PAGE]
                                                [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-registry-tag

```
usage: gitlab project-registry-tag [-h] {list,get,delete,delete-in-bulk} ...
```

**-h, --help**  
show this help message and exit

### gitlab project-registry-tag delete

```
usage: gitlab project-registry-tag delete [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --repository-id
                                           REPOSITORY_ID --name NAME
```

**-h, --help**  
show this help message and exit

```
--sudo <sudo>
--project-id <project_id>
--repository-id <repository_id>
--name <name>
```

### gitlab project-registry-tag delete-in-bulk

```
usage: gitlab project-registry-tag delete-in-bulk [-h] --project-id PROJECT_ID
--repository-id REPOSITORY_ID [--sudo SUDO]
--name NAME
--name-regex-delete NAME_REGEX_DELETE
[--keep-n KEEP_N]
[--name-regex-keep NAME_REGEX_KEEP]
[--older-than OLDER_THAN]
```

```
-h, --help
    show this help message and exit
--project-id <project_id>
--repository-id <repository_id>
--sudo <sudo>
--name <name>
--name-regex-delete <name_regex_delete>
--keep-n <keep_n>
--name-regex-keep <name_regex_keep>
--older-than <older_than>
```

### gitlab project-registry-tag get

```
usage: gitlab project-registry-tag get [-h] [--sudo SUDO] --project-id
PROJECT_ID --repository-id
REPOSITORY_ID --name NAME
```

```
-h, --help
    show this help message and exit
--sudo <sudo>
--project-id <project_id>
--repository-id <repository_id>
--name <name>
```

### gitlab project-registry-tag list

```
usage: gitlab project-registry-tag list [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --repository-id
                                         REPOSITORY_ID [--page PAGE]
                                         [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--repository-id** <repository\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-release

```
usage: gitlab project-release [-h] {list,get,create,update,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab project-release create

```
usage: gitlab project-release create [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --tag-name TAG_NAME
                                         [--name NAME] [--description DESCRIPTION]
                                         [--ref REF] [--assets ASSETS]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--tag-name** <tag\_name>

**--name** <name>

**--description** <description>

**--ref** <ref>

**--assets** <assets>

### gitlab project-release delete

```
usage: gitlab project-release delete [-h] [--sudo SUDO] --project-id
PROJECT_ID --tag-name TAG_NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--tag-name** <tag\_name>

### gitlab project-release get

```
usage: gitlab project-release get [-h] [--sudo SUDO] --project-id PROJECT_ID
--tag-name TAG_NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--tag-name** <tag\_name>

### gitlab project-release list

```
usage: gitlab project-release list [-h] [--sudo SUDO] --project-id PROJECT_ID
[--order-by ORDER_BY] [--sort SORT]
[--include-html-description INCLUDE_HTML_DESCRIPTION]
[--page PAGE] [--per-page PER_PAGE]
[--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--order-by** <order\_by>

**--sort** <sort>

**--include-html-description** <include\_html\_description>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-release update

```
usage: gitlab project-release update [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID --tag-name TAG_NAME
                                     [--name NAME] [--description DESCRIPTION]
                                     [--milestones MILESTONES]
                                     [--released-at RELEASED_AT]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--tag-name** <tag\_name>

**--name** <name>

**--description** <description>

**--milestones** <milestones>

**--released-at** <released\_at>

### gitlab project-release-link

```
usage: gitlab project-release-link [-h] {list,get,create,update,delete} ...
```

**-h, --help**  
show this help message and exit

#### gitlab project-release-link create

```
usage: gitlab project-release-link create [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --tag-name TAG_NAME
                                           --name NAME --url URL
                                           [--filepath FILEPATH]
                                           [--direct-asset-path DIRECT_ASSET_PATH]
                                           [--link-type LINK_TYPE]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--tag-name** <tag\_name>

**--name** <name>

**--url** <url>

**--filepath** <filepath>  
**--direct-asset-path** <direct\_asset\_path>  
**--link-type** <link\_type>

### gitlab project-release-link delete

```
usage: gitlab project-release-link delete [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --tag-name TAG_NAME --id
                                           ID
```

**-h, --help**  
 show this help message and exit  
**--sudo** <sudo>  
**--project-id** <project\_id>  
**--tag-name** <tag\_name>  
**--id** <id>

### gitlab project-release-link get

```
usage: gitlab project-release-link get [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --tag-name TAG_NAME --id ID
```

**-h, --help**  
 show this help message and exit  
**--sudo** <sudo>  
**--project-id** <project\_id>  
**--tag-name** <tag\_name>  
**--id** <id>

### gitlab project-release-link list

```
usage: gitlab project-release-link list [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --tag-name TAG_NAME
                                         [--page PAGE] [--per-page PER_PAGE]
                                         [--get-all]
```

**-h, --help**  
 show this help message and exit  
**--sudo** <sudo>  
**--project-id** <project\_id>

**--tag-name** <tag\_name>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-release-link update

```
usage: gitlab project-release-link update [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --tag-name TAG_NAME --id
                                           ID [--name NAME] [--url URL]
                                           [--filepath FILEPATH]
                                           [--direct-asset-path DIRECT_ASSET_PATH]
                                           [--link-type LINK_TYPE]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--tag-name** <tag\_name>

**--id** <id>

**--name** <name>

**--url** <url>

**--filepath** <filepath>

**--direct-asset-path** <direct\_asset\_path>

**--link-type** <link\_type>

### gitlab project-remote-mirror

```
usage: gitlab project-remote-mirror [-h] {list,create,update,delete} ...
```

**-h, --help**

show this help message and exit



### gitlab project-remote-mirror create

```
usage: gitlab project-remote-mirror create [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --url URL
                                           [--enabled ENABLED]
                                           [--only-protected-branches ONLY_PROTECTED_
↪BRANCHES]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--project-id** <project\_id>**--url** <url>**--enabled** <enabled>**--only-protected-branches** <only\_protected\_branches>

### gitlab project-remote-mirror delete

```
usage: gitlab project-remote-mirror delete [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--project-id** <project\_id>**--id** <id>

### gitlab project-remote-mirror list

```
usage: gitlab project-remote-mirror list [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID [--page PAGE]
                                           [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--project-id** <project\_id>**--page** <page>**--per-page** <per\_page>**--get-all**

Return all items from the server, without pagination.

### gitlab project-remote-mirror update

```
usage: gitlab project-remote-mirror update [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --id ID
                                           [--enabled ENABLED]
                                           [--only-protected-branches ONLY_PROTECTED_
↪BRANCHES]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

**--enabled** <enabled>

**--only-protected-branches** <only\_protected\_branches>

### gitlab project-resource-group

```
usage: gitlab project-resource-group [-h] {list,get,update} ...
```

**-h, --help**

show this help message and exit

### gitlab project-resource-group get

```
usage: gitlab project-resource-group get [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --key KEY
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--key** <key>

### gitlab project-resource-group list

```
usage: gitlab project-resource-group list [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID [--order-by ORDER_BY]
                                           [--sort SORT]
                                           [--include-html-description INCLUDE_HTML_
↪DESCRIPTION]
                                           [--page PAGE] [--per-page PER_PAGE]
                                           [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--order-by** <order\_by>

**--sort** <sort>

**--include-html-description** <include\_html\_description>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-resource-group update

```
usage: gitlab project-resource-group update [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --key KEY
                                           [--process-mode PROCESS_MODE]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--key** <key>

**--process-mode** <process\_mode>

### gitlab project-resource-group-upcoming-job

```
usage: gitlab project-resource-group-upcoming-job [-h] {list} ...
```

**-h, --help**  
show this help message and exit

### gitlab project-resource-group-upcoming-job list

```
usage: gitlab project-resource-group-upcoming-job list [-h] [--sudo SUDO]
                                                       --project-id PROJECT_ID
                                                       --resource-group-key
                                                       RESOURCE_GROUP_KEY
                                                       [--page PAGE]
                                                       [--per-page PER_PAGE]
                                                       [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--resource-group-key** <resource\_group\_key>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab project-runner

```
usage: gitlab project-runner [-h] {list,create,delete} ...
```

**-h, --help**

show this help message and exit

## gitlab project-runner create

```
usage: gitlab project-runner create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                     --runner-id RUNNER_ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--runner-id** <runner\_id>

## gitlab project-runner delete

```
usage: gitlab project-runner delete [-h] [--sudo SUDO] --project-id PROJECT_ID
                                     --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

## gitlab project-runner list

```
usage: gitlab project-runner list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                [--scope SCOPE] [--tag-list TAG_LIST]
                                [--page PAGE] [--per-page PER_PAGE]
                                [--get-all]
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--scope** <scope>

**--tag-list** <tag\_list>

**--page** <page>

**--per-page** <per\_page>

### **--get-all**

Return all items from the server, without pagination.

## gitlab project-secure-file

```
usage: gitlab project-secure-file [-h] {list,get,create,delete,download} ...
```

### **-h, --help**

show this help message and exit

## gitlab project-secure-file create

```
usage: gitlab project-secure-file create [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --name NAME --file FILE
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

**--file** <file>

### gitlab project-secure-file delete

```
usage: gitlab project-secure-file delete [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-secure-file download

```
usage: gitlab project-secure-file download [-h] --project-id PROJECT_ID
                                           [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

### gitlab project-secure-file get

```
usage: gitlab project-secure-file get [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-secure-file list

```
usage: gitlab project-secure-file list [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID [--page PAGE]
                                         [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab project-service

```
usage: gitlab project-service [-h] {list,get,update,delete,available} ...
```

**-h, --help**

show this help message and exit

## gitlab project-service available

```
usage: gitlab project-service available [-h] --project-id PROJECT_ID
                                         [--sudo SUDO] --slug SLUG
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--slug** <slug>

## gitlab project-service delete

```
usage: gitlab project-service delete [-h] [--sudo SUDO] --project-id
PROJECT_ID --slug SLUG
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--slug** <slug>

### gitlab project-service get

```
usage: gitlab project-service get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --slug SLUG
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--slug** <slug>

### gitlab project-service list

```
usage: gitlab project-service list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   [--page PAGE] [--per-page PER_PAGE]
                                   [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-service update

```
usage: gitlab project-service update [-h] [--sudo SUDO] --project-id
PROJECT_ID --slug SLUG
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--slug** <slug>



## gitlab project-snippet

```
usage: gitlab project-snippet [-h]
    list
    get
    create
    update
    delete
    user-agent-detail
    content
    ...
```

### -h, --help

show this help message and exit

## gitlab project-snippet content

```
usage: gitlab project-snippet content [-h] --project-id PROJECT_ID
                                     [--sudo SUDO] --id ID
```

### -h, --help

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

## gitlab project-snippet create

```
usage: gitlab project-snippet create [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID --title TITLE --visibility
                                     VISIBILITY [--description DESCRIPTION]
                                     [--content CONTENT]
                                     [--files FILES | --file-name FILE_NAME]
```

### -h, --help

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--title** <title>

**--visibility** <visibility>

**--description** <description>

**--content** <content>

**--files** <files>

**--file-name** <file\_name>

### gitlab project-snippet delete

```
usage: gitlab project-snippet delete [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-snippet get

```
usage: gitlab project-snippet get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-snippet list

```
usage: gitlab project-snippet list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   [--page PAGE] [--per-page PER_PAGE]
                                   [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

## gitlab project-snippet update

```
usage: gitlab project-snippet update [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID --id ID [--title TITLE]
                                     [--files FILES] [--file-name FILE_NAME]
                                     [--content CONTENT]
                                     [--visibility VISIBILITY]
                                     [--description DESCRIPTION]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

**--title** <title>

**--files** <files>

**--file-name** <file\_name>

**--content** <content>

**--visibility** <visibility>

**--description** <description>

## gitlab project-snippet user-agent-detail

```
usage: gitlab project-snippet user-agent-detail [-h] --project-id PROJECT_ID
                                                [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>

**--id** <id>

## gitlab project-snippet-award-emoji

```
usage: gitlab project-snippet-award-emoji [-h] {list,get,create,delete} ...
```

**-h, --help**  
show this help message and exit

### gitlab project-snippet-award-emoji create

```
usage: gitlab project-snippet-award-emoji create [-h] [--sudo SUDO]
                                                --project-id PROJECT_ID
                                                --snippet-id SNIPPET_ID
                                                --name NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--snippet-id** <snippet\_id>

**--name** <name>

### gitlab project-snippet-award-emoji delete

```
usage: gitlab project-snippet-award-emoji delete [-h] [--sudo SUDO]
                                                --project-id PROJECT_ID
                                                --snippet-id SNIPPET_ID --id
                                                ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--snippet-id** <snippet\_id>

**--id** <id>

### gitlab project-snippet-award-emoji get

```
usage: gitlab project-snippet-award-emoji get [-h] [--sudo SUDO] --project-id
                                                PROJECT_ID --snippet-id
                                                SNIPPET_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--snippet-id** <snippet\_id>

**--id** <id>

### gitlab project-snippet-award-emoji list

```
usage: gitlab project-snippet-award-emoji list [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --snippet-id
                                           SNIPPET_ID [--page PAGE]
                                           [--per-page PER_PAGE]
                                           [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--snippet-id** <snippet\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-snippet-discussion

```
usage: gitlab project-snippet-discussion [-h] {list,get,create} ...
```

**-h, --help**

show this help message and exit

### gitlab project-snippet-discussion create

```
usage: gitlab project-snippet-discussion create [-h] [--sudo SUDO]
                                           --project-id PROJECT_ID
                                           --snippet-id SNIPPET_ID --body
                                           BODY [--created-at CREATED_AT]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--snippet-id** <snippet\_id>

**--body** <body>

**--created-at** <created\_at>

### gitlab project-snippet-discussion get

```
usage: gitlab project-snippet-discussion get [-h] [--sudo SUDO] --project-id
      PROJECT_ID --snippet-id
      SNIPPET_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--snippet-id** <snippet\_id>

**--id** <id>

### gitlab project-snippet-discussion list

```
usage: gitlab project-snippet-discussion list [-h] [--sudo SUDO] --project-id
      PROJECT_ID --snippet-id
      SNIPPET_ID [--page PAGE]
      [--per-page PER_PAGE]
      [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--snippet-id** <snippet\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-snippet-discussion-note

```
usage: gitlab project-snippet-discussion-note [-h]
      get
      create
      update
      delete
      ...
```

**-h, --help**  
show this help message and exit

### gitlab project-snippet-discussion-note create

```
usage: gitlab project-snippet-discussion-note create [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --snippet-id SNIPPET_ID
                                                    --discussion-id
DISCUSSION_ID --body BODY
                                                    [--created-at CREATED_AT]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--snippet-id** <snippet\_id>

**--discussion-id** <discussion\_id>

**--body** <body>

**--created-at** <created\_at>

### gitlab project-snippet-discussion-note delete

```
usage: gitlab project-snippet-discussion-note delete [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --snippet-id SNIPPET_ID
                                                    --discussion-id
DISCUSSION_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--snippet-id** <snippet\_id>

**--discussion-id** <discussion\_id>

**--id** <id>

### gitlab project-snippet-discussion-note get

```
usage: gitlab project-snippet-discussion-note get [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --snippet-id SNIPPET_ID
                                                    --discussion-id
DISCUSSION_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--snippet-id** <snippet\_id>

**--discussion-id** <discussion\_id>

**--id** <id>

### gitlab project-snippet-discussion-note update

```
usage: gitlab project-snippet-discussion-note update [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --snippet-id SNIPPET_ID
                                                    --discussion-id
DISCUSSION_ID --id ID
                                                    --body BODY
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--snippet-id** <snippet\_id>

**--discussion-id** <discussion\_id>

**--id** <id>

**--body** <body>

### gitlab project-snippet-note

```
usage: gitlab project-snippet-note [-h] {list,get,create,update,delete} ...
```

**-h, --help**  
show this help message and exit

### gitlab project-snippet-note create

```
usage: gitlab project-snippet-note create [-h] [--sudo SUDO] --project-id
PROJECT_ID --snippet-id SNIPPET_ID
--body BODY
```

**-h, --help**  
show this help message and exit



```
--sudo <sudo>
--project-id <project_id>
--snippet-id <snippet_id>
--body <body>
```

### gitlab project-snippet-note delete

```
usage: gitlab project-snippet-note delete [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --snippet-id SNIPPET_ID
                                           --id ID
```

```
-h, --help
    show this help message and exit
--sudo <sudo>
--project-id <project_id>
--snippet-id <snippet_id>
--id <id>
```

### gitlab project-snippet-note get

```
usage: gitlab project-snippet-note get [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --snippet-id SNIPPET_ID --id
                                         ID
```

```
-h, --help
    show this help message and exit
--sudo <sudo>
--project-id <project_id>
--snippet-id <snippet_id>
--id <id>
```

### gitlab project-snippet-note list

```
usage: gitlab project-snippet-note list [-h] [--sudo SUDO] --project-id
                                         PROJECT_ID --snippet-id SNIPPET_ID
                                         [--page PAGE] [--per-page PER_PAGE]
                                         [--get-all]
```

```
-h, --help
    show this help message and exit
```

**--sudo** <sudo>

**--project-id** <project\_id>

**--snippet-id** <snippet\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-snippet-note update

```
usage: gitlab project-snippet-note update [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID --snippet-id SNIPPET_ID
                                           --id ID --body BODY
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--snippet-id** <snippet\_id>

**--id** <id>

**--body** <body>

### gitlab project-snippet-note-award-emoji

```
usage: gitlab project-snippet-note-award-emoji [-h]
list
get
create
delete
...
```

**-h, --help**

show this help message and exit

### gitlab project-snippet-note-award-emoji create

```
usage: gitlab project-snippet-note-award-emoji create [-h] [--sudo SUDO]
                                                       --project-id PROJECT_ID
                                                       --snippet-id SNIPPET_ID
                                                       --note-id NOTE_ID --name
                                                       NAME
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--snippet-id** <snippet\_id>

**--note-id** <note\_id>

**--name** <name>

### gitlab project-snippet-note-award-emoji delete

```
usage: gitlab project-snippet-note-award-emoji delete [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --snippet-id SNIPPET_ID
                                                    --note-id NOTE_ID --id
                                                    ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--snippet-id** <snippet\_id>

**--note-id** <note\_id>

**--id** <id>

### gitlab project-snippet-note-award-emoji get

```
usage: gitlab project-snippet-note-award-emoji get [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --snippet-id SNIPPET_ID
                                                    --note-id NOTE_ID --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--snippet-id** <snippet\_id>

**--note-id** <note\_id>

**--id** <id>

### gitlab project-snippet-note-award-emoji list

```
usage: gitlab project-snippet-note-award-emoji list [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --snippet-id SNIPPET_ID
                                                    --note-id NOTE_ID
                                                    [--page PAGE]
                                                    [--per-page PER_PAGE]
                                                    [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--snippet-id** <snippet\_id>

**--note-id** <note\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-storage

```
usage: gitlab project-storage [-h] {get} ...
```

**-h, --help**

show this help message and exit

### gitlab project-storage get

```
usage: gitlab project-storage get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

## gitlab project-tag

```
usage: gitlab project-tag [-h] {list,get,create,delete} ...
```

**-h, --help**

show this help message and exit

## gitlab project-tag create

```
usage: gitlab project-tag create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                --tag-name TAG_NAME --ref REF
                                [--message MESSAGE]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--tag-name** <tag\_name>

**--ref** <ref>

**--message** <message>

## gitlab project-tag delete

```
usage: gitlab project-tag delete [-h] [--sudo SUDO] --project-id PROJECT_ID
                                --name NAME
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

## gitlab project-tag get

```
usage: gitlab project-tag get [-h] [--sudo SUDO] --project-id PROJECT_ID
                              --name NAME
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--name** <name>

### gitlab project-tag list

```
usage: gitlab project-tag list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                [--page PAGE] [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-trigger

```
usage: gitlab project-trigger [-h] {list,get,create,update,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab project-trigger create

```
usage: gitlab project-trigger create [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID --description DESCRIPTION
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--description** <description>

### gitlab project-trigger delete

```
usage: gitlab project-trigger delete [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-trigger get

```
usage: gitlab project-trigger get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

### gitlab project-trigger list

```
usage: gitlab project-trigger list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   [--page PAGE] [--per-page PER_PAGE]
                                   [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab project-trigger update

```
usage: gitlab project-trigger update [-h] [--sudo SUDO] --project-id
PROJECT_ID --id ID --description
DESCRIPTION
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--id** <id>

**--description** <description>

## gitlab project-user

```
usage: gitlab project-user [-h] {list} ...
```

**-h, --help**

show this help message and exit

## gitlab project-user list

```
usage: gitlab project-user list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                [--search SEARCH] [--skip-users SKIP_USERS]
                                [--page PAGE] [--per-page PER_PAGE]
                                [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--search** <search>

**--skip-users** <skip\_users>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab project-variable

```
usage: gitlab project-variable [-h] {list,get,create,update,delete} ...
```

**-h, --help**

show this help message and exit

## gitlab project-variable create

```
usage: gitlab project-variable create [-h] [--sudo SUDO] --project-id
PROJECT_ID --key KEY --value VALUE
                                [--protected PROTECTED]
                                [--variable-type VARIABLE_TYPE]
                                [--masked MASKED]
                                [--environment-scope ENVIRONMENT_SCOPE]
```

**-h, --help**

show this help message and exit



```

--sudo <sudo>
--project-id <project_id>
--key <key>
--value <value>
--protected <protected>
--variable-type <variable_type>
--masked <masked>
--environment-scope <environment_scope>

```

### gitlab project-variable delete

```
usage: gitlab project-variable delete [-h] [--sudo SUDO] --project-id
PROJECT_ID --key KEY
```

```

-h, --help
    show this help message and exit
--sudo <sudo>
--project-id <project_id>
--key <key>

```

### gitlab project-variable get

```
usage: gitlab project-variable get [-h] [--sudo SUDO] --project-id PROJECT_ID
--key KEY
```

```

-h, --help
    show this help message and exit
--sudo <sudo>
--project-id <project_id>
--key <key>

```

### gitlab project-variable list

```
usage: gitlab project-variable list [-h] [--sudo SUDO] --project-id PROJECT_ID
[--page PAGE] [--per-page PER_PAGE]
[--get-all]
```

```

-h, --help
    show this help message and exit

```

**--sudo** <sudo>

**--project-id** <project\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-variable update

```
usage: gitlab project-variable update [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID --key KEY --value VALUE
                                     [--protected PROTECTED]
                                     [--variable-type VARIABLE_TYPE]
                                     [--masked MASKED]
                                     [--environment-scope ENVIRONMENT_SCOPE]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--key** <key>

**--value** <value>

**--protected** <protected>

**--variable-type** <variable\_type>

**--masked** <masked>

**--environment-scope** <environment\_scope>

### gitlab project-wiki

```
usage: gitlab project-wiki [-h] {list,get,create,update,delete,upload} ...
```

**-h, --help**

show this help message and exit

### gitlab project-wiki create

```
usage: gitlab project-wiki create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --title TITLE --content CONTENT
                                   [--format FORMAT]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--title** <title>

**--content** <content>

**--format** <format>

### gitlab project-wiki delete

```
usage: gitlab project-wiki delete [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --slug SLUG
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--slug** <slug>

### gitlab project-wiki get

```
usage: gitlab project-wiki get [-h] [--sudo SUDO] --project-id PROJECT_ID
                               --slug SLUG
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--slug** <slug>

### gitlab project-wiki list

```
usage: gitlab project-wiki list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                [--with-content WITH_CONTENT] [--page PAGE]
                                [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--with-content** <with\_content>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab project-wiki update

```
usage: gitlab project-wiki update [-h] [--sudo SUDO] --project-id PROJECT_ID
                                --slug SLUG [--title TITLE]
                                [--content CONTENT] [--format FORMAT]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--slug** <slug>

**--title** <title>

**--content** <content>

**--format** <format>

### gitlab project-wiki upload

```
usage: gitlab project-wiki upload [-h] --project-id PROJECT_ID [--sudo SUDO]
                                --slug SLUG --filename FILENAME --filepath
                                FILEPATH
```

**-h, --help**

show this help message and exit

**--project-id** <project\_id>

**--sudo** <sudo>  
**--slug** <slug>  
**--filename** <filename>  
**--filepath** <filepath>

### gitlab registry-repository

```
usage: gitlab registry-repository [-h] {get} ...
```

**-h, --help**  
show this help message and exit

### gitlab registry-repository get

```
usage: gitlab registry-repository get [-h] [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--id** <id>

### gitlab runner

```
usage: gitlab runner [-h] {list,get,create,update,delete,all,verify} ...
```

**-h, --help**  
show this help message and exit

### gitlab runner all

```
usage: gitlab runner all [-h] --id ID [--scope SCOPE]
```

**-h, --help**  
show this help message and exit  
**--id** <id>  
**--scope** <scope>

## gitlab runner create

```
usage: gitlab runner create [-h] [--sudo SUDO] --token TOKEN
                             [--description DESCRIPTION] [--info INFO]
                             [--active ACTIVE] [--locked LOCKED]
                             [--run-untagged RUN_UNTAGGED]
                             [--tag-list TAG_LIST]
                             [--access-level ACCESS_LEVEL]
                             [--maximum-timeout MAXIMUM_TIMEOUT]
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--token** <token>

**--description** <description>

**--info** <info>

**--active** <active>

**--locked** <locked>

**--run-untagged** <run\_untagged>

**--tag-list** <tag\_list>

**--access-level** <access\_level>

**--maximum-timeout** <maximum\_timeout>

## gitlab runner delete

```
usage: gitlab runner delete [-h] [--sudo SUDO] --id ID
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>

## gitlab runner get

```
usage: gitlab runner get [-h] [--sudo SUDO] --id ID
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>

## gitlab runner list

```
usage: gitlab runner list [-h] [--sudo SUDO] [--scope SCOPE] [--type TYPE]
                        [--status STATUS] [--paused PAUSED]
                        [--tag-list TAG_LIST] [--page PAGE]
                        [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--scope** <scope>**--type** <type>**--status** <status>**--paused** <paused>**--tag-list** <tag\_list>**--page** <page>**--per-page** <per\_page>**--get-all**

Return all items from the server, without pagination.

## gitlab runner update

```
usage: gitlab runner update [-h] [--sudo SUDO] --id ID
                        [--description DESCRIPTION] [--active ACTIVE]
                        [--tag-list TAG_LIST]
                        [--run-untagged RUN_UNTAGGED] [--locked LOCKED]
                        [--access-level ACCESS_LEVEL]
                        [--maximum-timeout MAXIMUM_TIMEOUT]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--id** <id>**--description** <description>**--active** <active>**--tag-list** <tag\_list>**--run-untagged** <run\_untagged>**--locked** <locked>**--access-level** <access\_level>**--maximum-timeout** <maximum\_timeout>

### gitlab runner verify

```
usage: gitlab runner verify [-h] --id ID --token TOKEN
```

**-h, --help**

show this help message and exit

**--id** <id>

**--token** <token>

### gitlab runner-all

```
usage: gitlab runner-all [-h] {list} ...
```

**-h, --help**

show this help message and exit

### gitlab runner-all list

```
usage: gitlab runner-all list [-h] [--sudo SUDO] [--scope SCOPE] [--type TYPE]
                                [--status STATUS] [--paused PAUSED]
                                [--tag-list TAG_LIST] [--page PAGE]
                                [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--scope** <scope>

**--type** <type>

**--status** <status>

**--paused** <paused>

**--tag-list** <tag\_list>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.



## gitlab runner-job

```
usage: gitlab runner-job [-h] {list} ...
```

### -h, --help

show this help message and exit

## gitlab runner-job list

```
usage: gitlab runner-job list [-h] [--sudo SUDO] --runner-id RUNNER_ID
                                [--status STATUS] [--page PAGE]
                                [--per-page PER_PAGE] [--get-all]
```

### -h, --help

show this help message and exit

**--sudo** <sudo>

**--runner-id** <runner\_id>

**--status** <status>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab shared-project

```
usage: gitlab shared-project [-h] {list} ...
```

### -h, --help

show this help message and exit

## gitlab shared-project list

```
usage: gitlab shared-project list [-h] [--sudo SUDO] --group-id GROUP_ID
                                [--archived ARCHIVED]
                                [--visibility VISIBILITY]
                                [--order-by ORDER_BY] [--sort SORT]
                                [--search SEARCH] [--simple SIMPLE]
                                [--starred STARRED]
                                [--with-issues-enabled WITH_ISSUES_ENABLED]
                                [--with-merge-requests-enabled WITH_MERGE_REQUESTS_
↳ENABLED]
                                [--min-access-level MIN_ACCESS_LEVEL]
                                [--with-custom-attributes WITH_CUSTOM_ATTRIBUTES]
                                [--page PAGE] [--per-page PER_PAGE]
                                [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--archived** <archived>

**--visibility** <visibility>

**--order-by** <order\_by>

**--sort** <sort>

**--search** <search>

**--simple** <simple>

**--starred** <starred>

**--with-issues-enabled** <with\_issues\_enabled>

**--with-merge-requests-enabled** <with\_merge\_requests\_enabled>

**--min-access-level** <min\_access\_level>

**--with-custom-attributes** <with\_custom\_attributes>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab snippet

```
usage: gitlab snippet [-h]
  list
  get
  create
  update
  delete
  user-agent-detail
  content
  public
  ...
```

**-h, --help**

show this help message and exit

### gitlab snippet content

```
usage: gitlab snippet content [-h] --id ID
```

**-h, --help**

show this help message and exit

**--id** <id>

### gitlab snippet create

```
usage: gitlab snippet create [-h] [--sudo SUDO] --title TITLE
                             [--description DESCRIPTION] [--content CONTENT]
                             [--visibility VISIBILITY]
                             [--files FILES | --file-name FILE_NAME]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--title** <title>

**--description** <description>

**--content** <content>

**--visibility** <visibility>

**--files** <files>

**--file-name** <file\_name>

### gitlab snippet delete

```
usage: gitlab snippet delete [-h] [--sudo SUDO] --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>

### gitlab snippet get

```
usage: gitlab snippet get [-h] [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--id** <id>

### gitlab snippet list

```
usage: gitlab snippet list [-h] [--sudo SUDO] [--page PAGE]
                        [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab snippet public

```
usage: gitlab snippet public [-h] --id ID
```

**-h, --help**  
show this help message and exit

**--id** <id>

### gitlab snippet update

```
usage: gitlab snippet update [-h] [--sudo SUDO] --id ID [--title TITLE]
                        [--files FILES] [--file-name FILE_NAME]
                        [--content CONTENT] [--visibility VISIBILITY]
                        [--description DESCRIPTION]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--id** <id>

**--title** <title>

```

--files <files>
--file-name <file_name>
--content <content>
--visibility <visibility>
--description <description>

```

### gitlab snippet user-agent-detail

```
usage: gitlab snippet user-agent-detail [-h] --id ID
```

```

-h, --help
    show this help message and exit
--id <id>

```

### gitlab starred-project

```
usage: gitlab starred-project [-h] {list} ...
```

```

-h, --help
    show this help message and exit

```

### gitlab starred-project list

```

usage: gitlab starred-project list [-h] [--sudo SUDO] --user-id USER_ID
                                     [--archived ARCHIVED]
                                     [--membership MEMBERSHIP]
                                     [--min-access-level MIN_ACCESS_LEVEL]
                                     [--order-by ORDER_BY] [--owned OWNED]
                                     [--search SEARCH] [--simple SIMPLE]
                                     [--sort SORT] [--starred STARRED]
                                     [--statistics STATISTICS]
                                     [--visibility VISIBILITY]
                                     [--with-custom-attributes WITH_CUSTOM_ATTRIBUTES]
                                     [--with-issues-enabled WITH_ISSUES_ENABLED]
                                     [--with-merge-requests-enabled WITH_MERGE_REQUESTS_
↳ENABLED]
                                     [--page PAGE] [--per-page PER_PAGE]
                                     [--get-all]

```

```

-h, --help
    show this help message and exit
--sudo <sudo>
--user-id <user_id>

```

**--archived** <archived>  
**--membership** <membership>  
**--min-access-level** <min\_access\_level>  
**--order-by** <order\_by>  
**--owned** <owned>  
**--search** <search>  
**--simple** <simple>  
**--sort** <sort>  
**--starred** <starred>  
**--statistics** <statistics>  
**--visibility** <visibility>  
**--with-custom-attributes** <with\_custom\_attributes>  
**--with-issues-enabled** <with\_issues\_enabled>  
**--with-merge-requests-enabled** <with\_merge\_requests\_enabled>  
**--page** <page>  
**--per-page** <per\_page>  
**--get-all**  
Return all items from the server, without pagination.

### gitlab todo

```
usage: gitlab todo [-h] {list,delete,mark-as-done,mark-all-as-done} ...
```

**-h, --help**  
show this help message and exit

### gitlab todo delete

```
usage: gitlab todo delete [-h] [--sudo SUDO] --id ID
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--id** <id>

## gitlab todo list

```
usage: gitlab todo list [-h] [--sudo SUDO] [--action ACTION]
                        [--author-id AUTHOR_ID] [--project-id PROJECT_ID]
                        [--state STATE] [--type TYPE] [--page PAGE]
                        [--per-page PER_PAGE] [--get-all]
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--action** <action>

**--author-id** <author\_id>

**--project-id** <project\_id>

**--state** <state>

**--type** <type>

**--page** <page>

**--per-page** <per\_page>

### **--get-all**

Return all items from the server, without pagination.

## gitlab todo mark-all-as-done

```
usage: gitlab todo mark-all-as-done [-h] --id ID
```

### **-h, --help**

show this help message and exit

**--id** <id>

## gitlab todo mark-as-done

```
usage: gitlab todo mark-as-done [-h] --id ID
```

### **-h, --help**

show this help message and exit

**--id** <id>

## gitlab topic

```
usage: gitlab topic [-h] {list,get,create,update,delete,merge} ...
```

### **-h, --help**

show this help message and exit

## gitlab topic create

```
usage: gitlab topic create [-h] [--sudo SUDO] --name NAME [--avatar AVATAR]
                             [--description DESCRIPTION] [--title TITLE]
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--name** <name>

**--avatar** <avatar>

**--description** <description>

**--title** <title>

## gitlab topic delete

```
usage: gitlab topic delete [-h] [--sudo SUDO] --id ID
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>

## gitlab topic get

```
usage: gitlab topic get [-h] [--sudo SUDO] --id ID
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>



### gitlab topic list

```
usage: gitlab topic list [-h] [--sudo SUDO] [--page PAGE]
                        [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab topic merge

```
usage: gitlab topic merge [-h] --id ID --source-topic-id SOURCE_TOPIC_ID
                        --target-topic-id TARGET_TOPIC_ID
```

**-h, --help**

show this help message and exit

**--id** <id>

**--source-topic-id** <source\_topic\_id>

**--target-topic-id** <target\_topic\_id>

### gitlab topic update

```
usage: gitlab topic update [-h] [--sudo SUDO] --id ID [--avatar AVATAR]
                        [--description DESCRIPTION] [--name NAME]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>

**--avatar** <avatar>

**--description** <description>

**--name** <name>

## gitlab user

```
usage: gitlab user [-h]
    list
    get
    create
    update
    delete
    block
    follow
    unfollow
    unblock
    deactivate
    activate
    approve
    reject
    ban
    unban
    ...
```

### -h, --help

show this help message and exit

## gitlab user activate

```
usage: gitlab user activate [-h] --id ID
```

### -h, --help

show this help message and exit

--id <id>

## gitlab user approve

```
usage: gitlab user approve [-h] --id ID
```

### -h, --help

show this help message and exit

--id <id>

## gitlab user ban

```
usage: gitlab user ban [-h] --id ID
```

### -h, --help

show this help message and exit

--id <id>

## gitlab user block

```
usage: gitlab user block [-h] --id ID
```

**-h, --help**

show this help message and exit

**--id** <id>

## gitlab user create

```
usage: gitlab user create [-h] [--sudo SUDO] [--email EMAIL]
                        [--username USERNAME] [--name NAME]
                        [--password PASSWORD]
                        [--reset-password RESET_PASSWORD] [--skype SKYPE]
                        [--linkedin LINKEDIN] [--twitter TWITTER]
                        [--projects-limit PROJECTS_LIMIT]
                        [--extern-uid EXTERN_UID] [--provider PROVIDER]
                        [--bio BIO] [--admin ADMIN]
                        [--can-create-group CAN_CREATE_GROUP]
                        [--website-url WEBSITE_URL]
                        [--skip-confirmation SKIP_CONFIRMATION]
                        [--external EXTERNAL] [--organization ORGANIZATION]
                        [--location LOCATION] [--avatar AVATAR]
                        [--public-email PUBLIC_EMAIL]
                        [--private-profile PRIVATE_PROFILE]
                        [--color-scheme-id COLOR_SCHEME_ID]
                        [--theme-id THEME_ID]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--email** <email>

**--username** <username>

**--name** <name>

**--password** <password>

**--reset-password** <reset\_password>

**--skype** <skype>

**--linkedin** <linkedin>

**--twitter** <twitter>

**--projects-limit** <projects\_limit>

**--extern-uid** <extern\_uid>

**--provider** <provider>

```
--bio <bio>
--admin <admin>
--can-create-group <can_create_group>
--website-url <website_url>
--skip-confirmation <skip_confirmation>
--external <external>
--organization <organization>
--location <location>
--avatar <avatar>
--public-email <public_email>
--private-profile <private_profile>
--color-scheme-id <color_scheme_id>
--theme-id <theme_id>
```

### gitlab user deactivate

```
usage: gitlab user deactivate [-h] --id ID
```

```
-h, --help
    show this help message and exit
--id <id>
```

### gitlab user delete

```
usage: gitlab user delete [-h] [--sudo SUDO] --id ID
```

```
-h, --help
    show this help message and exit
--sudo <sudo>
--id <id>
```

### gitlab user follow

```
usage: gitlab user follow [-h] --id ID
```

**-h, --help**

show this help message and exit

**--id** <id>

### gitlab user get

```
usage: gitlab user get [-h] [--sudo SUDO] --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>

### gitlab user list

```
usage: gitlab user list [-h] [--sudo SUDO] [--active ACTIVE]
                        [--blocked BLOCKED] [--username USERNAME]
                        [--extern-uid EXTERN_UID] [--provider PROVIDER]
                        [--external EXTERNAL] [--search SEARCH]
                        [--custom-attributes CUSTOM_ATTRIBUTES]
                        [--status STATUS] [--two-factor TWO_FACTOR]
                        [--page PAGE] [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--active** <active>

**--blocked** <blocked>

**--username** <username>

**--extern-uid** <extern\_uid>

**--provider** <provider>

**--external** <external>

**--search** <search>

**--custom-attributes** <custom\_attributes>

**--status** <status>

**--two-factor** <two\_factor>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab user reject

```
usage: gitlab user reject [-h] --id ID
```

**-h, --help**

show this help message and exit

**--id** <id>

### gitlab user unban

```
usage: gitlab user unban [-h] --id ID
```

**-h, --help**

show this help message and exit

**--id** <id>

### gitlab user unblock

```
usage: gitlab user unblock [-h] --id ID
```

**-h, --help**

show this help message and exit

**--id** <id>

### gitlab user unfollow

```
usage: gitlab user unfollow [-h] --id ID
```

**-h, --help**

show this help message and exit

**--id** <id>

**gitlab user update**

```
usage: gitlab user update [-h] [--sudo SUDO] --id ID --email EMAIL --username
                        USERNAME --name NAME [--password PASSWORD]
                        [--skype SKYPE] [--linkedin LINKEDIN]
                        [--twitter TWITTER]
                        [--projects-limit PROJECTS_LIMIT]
                        [--extern-uid EXTERN_UID] [--provider PROVIDER]
                        [--bio BIO] [--admin ADMIN]
                        [--can-create-group CAN_CREATE_GROUP]
                        [--website-url WEBSITE_URL]
                        [--skip-reconfirmation SKIP_RECONFIRMATION]
                        [--external EXTERNAL] [--organization ORGANIZATION]
                        [--location LOCATION] [--avatar AVATAR]
                        [--public-email PUBLIC_EMAIL]
                        [--private-profile PRIVATE_PROFILE]
                        [--color-scheme-id COLOR_SCHEME_ID]
                        [--theme-id THEME_ID]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>

**--email** <email>

**--username** <username>

**--name** <name>

**--password** <password>

**--skype** <skype>

**--linkedin** <linkedin>

**--twitter** <twitter>

**--projects-limit** <projects\_limit>

**--extern-uid** <extern\_uid>

**--provider** <provider>

**--bio** <bio>

**--admin** <admin>

**--can-create-group** <can\_create\_group>

**--website-url** <website\_url>

**--skip-reconfirmation** <skip\_reconfirmation>

**--external** <external>

**--organization** <organization>  
**--location** <location>  
**--avatar** <avatar>  
**--public-email** <public\_email>  
**--private-profile** <private\_profile>  
**--color-scheme-id** <color\_scheme\_id>  
**--theme-id** <theme\_id>

### gitlab user-activities

```
usage: gitlab user-activities [-h] {list} ...
```

**-h, --help**  
show this help message and exit

### gitlab user-activities list

```
usage: gitlab user-activities list [-h] [--sudo SUDO] [--page PAGE]
                                     [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

### gitlab user-custom-attribute

```
usage: gitlab user-custom-attribute [-h] {list,get,delete} ...
```

**-h, --help**  
show this help message and exit



### gitlab user-custom-attribute delete

```
usage: gitlab user-custom-attribute delete [-h] [--sudo SUDO] --user-id
USER_ID --key KEY
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--key** <key>

### gitlab user-custom-attribute get

```
usage: gitlab user-custom-attribute get [-h] [--sudo SUDO] --user-id USER_ID
--key KEY
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--key** <key>

### gitlab user-custom-attribute list

```
usage: gitlab user-custom-attribute list [-h] [--sudo SUDO] --user-id USER_ID
[--page PAGE] [--per-page PER_PAGE]
[--get-all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

## gitlab user-email

```
usage: gitlab user-email [-h] {list,get,create,delete} ...
```

**-h, --help**

show this help message and exit

## gitlab user-email create

```
usage: gitlab user-email create [-h] [--sudo SUDO] --user-id USER_ID --email  
                                EMAIL
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--email** <email>

## gitlab user-email delete

```
usage: gitlab user-email delete [-h] [--sudo SUDO] --user-id USER_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--id** <id>

## gitlab user-email get

```
usage: gitlab user-email get [-h] [--sudo SUDO] --user-id USER_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--id** <id>

### gitlab user-email list

```
usage: gitlab user-email list [-h] [--sudo SUDO] --user-id USER_ID
                             [--page PAGE] [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--user-id** <user\_id>**--page** <page>**--per-page** <per\_page>**--get-all**

Return all items from the server, without pagination.

### gitlab user-event

```
usage: gitlab user-event [-h] {list} ...
```

**-h, --help**

show this help message and exit

### gitlab user-event list

```
usage: gitlab user-event list [-h] [--sudo SUDO] --user-id USER_ID
                             [--action ACTION] [--target-type TARGET_TYPE]
                             [--before BEFORE] [--after AFTER] [--sort SORT]
                             [--scope SCOPE] [--page PAGE]
                             [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--user-id** <user\_id>**--action** <action>**--target-type** <target\_type>**--before** <before>**--after** <after>**--sort** <sort>**--scope** <scope>**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab user-gpg-key

```
usage: gitlab user-gpg-key [-h] {list,get,create,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab user-gpg-key create

```
usage: gitlab user-gpg-key create [-h] [--sudo SUDO] --user-id USER_ID --key
                                   KEY
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--key** <key>

### gitlab user-gpg-key delete

```
usage: gitlab user-gpg-key delete [-h] [--sudo SUDO] --user-id USER_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--id** <id>

### gitlab user-gpg-key get

```
usage: gitlab user-gpg-key get [-h] [--sudo SUDO] --user-id USER_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--id** <id>

### gitlab user-gpg-key list

```
usage: gitlab user-gpg-key list [-h] [--sudo SUDO] --user-id USER_ID
                                [--page PAGE] [--per-page PER_PAGE]
                                [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--user-id** <user\_id>**--page** <page>**--per-page** <per\_page>**--get-all**

Return all items from the server, without pagination.

### gitlab user-impersonation-token

```
usage: gitlab user-impersonation-token [-h] {list,get,create,delete} ...
```

**-h, --help**

show this help message and exit

### gitlab user-impersonation-token create

```
usage: gitlab user-impersonation-token create [-h] [--sudo SUDO] --user-id
                                              USER_ID --name NAME --scopes
                                              SCOPES [--expires-at EXPIRES_AT]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>**--user-id** <user\_id>**--name** <name>**--scopes** <scopes>**--expires-at** <expires\_at>

### gitlab user-impersonation-token delete

```
usage: gitlab user-impersonation-token delete [-h] [--sudo SUDO] --user-id
USER_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--id** <id>

### gitlab user-impersonation-token get

```
usage: gitlab user-impersonation-token get [-h] [--sudo SUDO] --user-id
USER_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--id** <id>

### gitlab user-impersonation-token list

```
usage: gitlab user-impersonation-token list [-h] [--sudo SUDO] --user-id
USER_ID [--state STATE]
[--page PAGE]
[--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--state** <state>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

## gitlab user-key

```
usage: gitlab user-key [-h] {list,get,create,delete} ...
```

**-h, --help**

show this help message and exit

## gitlab user-key create

```
usage: gitlab user-key create [-h] [--sudo SUDO] --user-id USER_ID --title  
                               TITLE --key KEY
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--title** <title>

**--key** <key>

## gitlab user-key delete

```
usage: gitlab user-key delete [-h] [--sudo SUDO] --user-id USER_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--id** <id>

## gitlab user-key get

```
usage: gitlab user-key get [-h] [--sudo SUDO] --user-id USER_ID --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--id** <id>

### gitlab user-key list

```
usage: gitlab user-key list [-h] [--sudo SUDO] --user-id USER_ID [--page PAGE]
                             [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab user-membership

```
usage: gitlab user-membership [-h] {list,get} ...
```

**-h, --help**

show this help message and exit

### gitlab user-membership get

```
usage: gitlab user-membership get [-h] [--sudo SUDO] --user-id USER_ID
                                   --source-id SOURCE_ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--source-id** <source\_id>

### gitlab user-membership list

```
usage: gitlab user-membership list [-h] [--sudo SUDO] --user-id USER_ID
                                   [--type TYPE] [--page PAGE]
                                   [--per-page PER_PAGE] [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>



**--type** <type>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

### gitlab user-personal-access-token

```
usage: gitlab user-personal-access-token [-h] {create} ...
```

**-h, --help**

show this help message and exit

### gitlab user-personal-access-token create

```
usage: gitlab user-personal-access-token create [-h] [--sudo SUDO] --user-id
                                                USER_ID --name NAME --scopes
                                                SCOPES
                                                [--expires-at EXPIRES_AT]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--name** <name>

**--scopes** <scopes>

**--expires-at** <expires\_at>

### gitlab user-project

```
usage: gitlab user-project [-h] {list,create} ...
```

**-h, --help**

show this help message and exit

## gitlab user-project create

```
usage: gitlab user-project create [-h] [--sudo SUDO] --user-id USER_ID --name
                                NAME [--default-branch DEFAULT_BRANCH]
                                [--issues-enabled ISSUES_ENABLED]
                                [--wall-enabled WALL_ENABLED]
                                [--merge-requests-enabled MERGE_REQUESTS_ENABLED]
                                [--wiki-enabled WIKI_ENABLED]
                                [--snippets-enabled SNIPPETS_ENABLED]
                                [--squash-option SQUASH_OPTION]
                                [--public PUBLIC] [--visibility VISIBILITY]
                                [--description DESCRIPTION]
                                [--builds-enabled BUILDS_ENABLED]
                                [--public-builds PUBLIC_BUILDS]
                                [--import-url IMPORT_URL]
                                [--only-allow-merge-if-build-succeeds ONLY_ALLOW_MERGE_
↪IF_BUILD_SUCCEEDS]
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--name** <name>

**--default-branch** <default\_branch>

**--issues-enabled** <issues\_enabled>

**--wall-enabled** <wall\_enabled>

**--merge-requests-enabled** <merge\_requests\_enabled>

**--wiki-enabled** <wiki\_enabled>

**--snippets-enabled** <snippets\_enabled>

**--squash-option** <squash\_option>

**--public** <public>

**--visibility** <visibility>

**--description** <description>

**--builds-enabled** <builds\_enabled>

**--public-builds** <public\_builds>

**--import-url** <import\_url>

**--only-allow-merge-if-build-succeeds** <only\_allow\_merge\_if\_build\_succeeds>

**gitlab user-project list**

```
usage: gitlab user-project list [-h] [--sudo SUDO] --user-id USER_ID
                                [--archived ARCHIVED]
                                [--visibility VISIBILITY]
                                [--order-by ORDER_BY] [--sort SORT]
                                [--search SEARCH] [--simple SIMPLE]
                                [--owned OWNED] [--membership MEMBERSHIP]
                                [--starred STARRED] [--statistics STATISTICS]
                                [--with-issues-enabled WITH_ISSUES_ENABLED]
                                --with-merge-requests-enabled WITH_MERGE_REQUESTS_
                                ↪ENABLED]
                                [--with-custom-attributes WITH_CUSTOM_ATTRIBUTES]
                                [--with-programming-language WITH_PROGRAMMING_LANGUAGE]
                                [--wiki-checksum-failed WIKI_CHECKSUM_FAILED]
                                [--repository-checksum-failed REPOSITORY_CHECKSUM_FAILED]
                                [--min-access-level MIN_ACCESS_LEVEL]
                                [--id-after ID_AFTER] [--id-before ID_BEFORE]
                                [--page PAGE] [--per-page PER_PAGE]
                                [--get-all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--user-id** <user\_id>

**--archived** <archived>

**--visibility** <visibility>

**--order-by** <order\_by>

**--sort** <sort>

**--search** <search>

**--simple** <simple>

**--owned** <owned>

**--membership** <membership>

**--starred** <starred>

**--statistics** <statistics>

**--with-issues-enabled** <with\_issues\_enabled>

**--with-merge-requests-enabled** <with\_merge\_requests\_enabled>

**--with-custom-attributes** <with\_custom\_attributes>

**--with-programming-language** <with\_programming\_language>

**--wiki-checksum-failed** <wiki\_checksum\_failed>

**--repository-checksum-failed** <repository\_checksum\_failed>  
**--min-access-level** <min\_access\_level>  
**--id-after** <id\_after>  
**--id-before** <id\_before>  
**--page** <page>  
**--per-page** <per\_page>  
**--get-all**  
Return all items from the server, without pagination.

### gitlab user-status

```
usage: gitlab user-status [-h] {get} ...
```

**-h, --help**  
show this help message and exit

#### gitlab user-status get

```
usage: gitlab user-status get [-h] [--sudo SUDO] --user-id USER_ID
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--user-id** <user\_id>

### gitlab variable

```
usage: gitlab variable [-h] {list,get,create,update,delete} ...
```

**-h, --help**  
show this help message and exit

#### gitlab variable create

```
usage: gitlab variable create [-h] [--sudo SUDO] --key KEY --value VALUE  
                                [--protected PROTECTED]  
                                [--variable-type VARIABLE_TYPE]  
                                [--masked MASKED]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>  
**--key** <key>  
**--value** <value>  
**--protected** <protected>  
**--variable-type** <variable\_type>  
**--masked** <masked>

### gitlab variable delete

```
usage: gitlab variable delete [-h] [--sudo SUDO] --key KEY
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--key** <key>

### gitlab variable get

```
usage: gitlab variable get [-h] [--sudo SUDO] --key KEY
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--key** <key>

### gitlab variable list

```
usage: gitlab variable list [-h] [--sudo SUDO] [--page PAGE]
                             [--per-page PER_PAGE] [--get-all]
```

**-h, --help**  
show this help message and exit  
**--sudo** <sudo>  
**--page** <page>  
**--per-page** <per\_page>  
**--get-all**  
Return all items from the server, without pagination.

## gitlab variable update

```
usage: gitlab variable update [-h] [--sudo SUDO] --key KEY --value VALUE
                               [--protected PROTECTED]
                               [--variable-type VARIABLE_TYPE]
                               [--masked MASKED]
```

### **-h, --help**

show this help message and exit

### **--sudo <sudo>**

### **--key <key>**

### **--value <value>**

### **--protected <protected>**

### **--variable-type <variable\_type>**

### **--masked <masked>**

## 7.8 Lower-level APIs

python-gitlab's API levels provide different degrees of convenience, control and stability.

### 7.8.1 Main interface - Gitlab, managers and objects

As shown in previous sections and examples, the high-level API interface wraps GitLab's API endpoints and makes them available from the `Gitlab` instance via managers that create objects you can manipulate.

This is what most users will want to use, as it covers most of GitLab's API endpoints, and allows you to write idiomatic Python code when interacting with the API.

### 7.8.2 Lower-level API - HTTP methods

**Danger:** At this point, python-gitlab will no longer take care of URL-encoding and other transformations needed to correctly pass API parameter types. You have to construct these correctly yourself.

However, you still benefit from many of the client's *Features* such as authentication, requests and retry handling.

---

**Important:** If you've found yourself at this section because of an endpoint not yet implemented in the library - please consider opening a pull request implementing the resource or at least filing an issue so we can track progress.

High-quality pull requests for standard endpoints that pass CI and include unit tests and documentation are easy to review, and can land quickly with monthly releases. If you ask, we can also trigger a new release, so you and everyone benefits from the contribution right away!

---

Managers and objects call specific HTTP methods to fetch or send data to the server. These methods can be invoked directly to access endpoints not currently implemented by the client. This essentially gives you some level of usability for any endpoint the moment it is available on your GitLab instance.

These methods can be accessed directly via the Gitlab instance (e.g. `gl.http_get()`), or via an object's manager (e.g. `project.manager.gitlab.http_get()`), if the Gitlab instance is not available in the current context.

For example, if you'd like to access GitLab's [undocumented latest pipeline endpoint](#), you can do so by calling `http_get()` with the path to the endpoint:

```
>>> gl = gitlab.Gitlab(private_token=private_token)
>>>
>>> pipeline = gl.http_get("/projects/gitlab-org%2Fgitlab/pipelines/latest")
>>> pipeline["id"]
449070256
```

The available methods are:

- `http_get()`
- `http_post()`
- `http_put()`
- `http_patch()`
- `http_delete()`
- `http_list()` (a wrapper around `http_get` handling pagination, including with lazy generators)
- `http_head()` (only returns the header dictionary)

### 7.8.3 Lower-level API - HTTP requests

**Important:** This is mostly intended for internal use in python-gitlab and may have a less stable interface than higher-level APIs. To lessen the chances of a change to the interface impacting your code, we recommend using keyword arguments when calling the interfaces.

At the lowest level, HTTP methods call `http_request()`, which performs the actual request and takes care of details such as timeouts, retries, and handling rate-limits.

This method can be invoked directly to or customize this behavior for a single request, or to call custom HTTP methods not currently implemented in the library - while still making use of all of the client's options and authentication methods.

For example, if for whatever reason you want to fetch allowed methods for an endpoint at runtime:

```
>>> gl = gitlab.Gitlab(private_token=private_token)
>>>
>>> response = gl.http_request(verb="OPTIONS", path="/projects")
>>> response.headers["Allow"]
'OPTIONS, GET, POST, HEAD'
```

Or get the total number of a user's events with a customized HEAD request:

```
>>> response = gl.http_request(
    verb="HEAD",
    path="/events",
```

(continues on next page)

(continued from previous page)

```
        query_params={"sudo": "some-user"},
        timeout=10
    )
>>> response.headers["X-Total"]
'123'
```

## 7.9 CHANGELOG

### 7.9.1 v4.4.0 (2024-01-15)

#### Chore

- chore(deps): update all non-major dependencies ([550f935](#))
- chore(deps): update pre-commit hook pycqa/flake8 to v7 ([9a199b6](#))
- chore(deps): update dependency jinja2 to v3.1.3 [security] ([880913b](#))
- chore(deps): update dependency flake8 to v7 ([20243c5](#))
- chore(deps): update all non-major dependencies ([cbc13a6](#))
- chore(ci): align upload and download action versions ([dcca59d](#))
- chore(deps): update actions/upload-artifact action to v4 ([7114af3](#))
- chore(ci): add Python 3.13 development CI job

Add a job to test the development versions of Python 3.13. ([ff0c11b](#))

- chore(deps): update all non-major dependencies ([369a595](#))

#### Feature

- feat(api): add reviewer\_details manager for mergerequest to get reviewers of merge request

Those changes implements ‘GET /projects/:id/merge\_requests/:merge\_request\_iid/reviewers’ gitlab API call. Naming for call is not reviewers because reviewers attribute already presen in merge request response ([adbd90c](#))

- feat(api): support access token rotate API ([b13971d](#))
- feat(api): support single resource access token get API ([dae9e52](#))

#### Fix

- fix(cli): support binary files with @ notation

Support binary files being used in the CLI with arguments using the @ notation. For example `--avatar @/path/to/avatar.png`

Also explicitly catch the common OSError exception, which is the parent exception for things like: FileNotFoundError, PermissionError and more exceptions.

Remove the bare exception handling. We would rather have the full traceback of any exceptions that we don’t know about and add them later if needed.

Closes: [#2752](#) ([57749d4](#))



## 7.9.2 v4.3.0 (2023-12-28)

### Chore

- chore(deps): update all non-major dependencies ([d7bdb02](#))
- chore(deps): update actions/stale action to v9 ([c01988b](#))
- chore(deps): update all non-major dependencies ([9e067e5](#))
- chore(deps): update actions/setup-python action to v5 ([fad1441](#))
- chore(deps): update all non-major dependencies ([bb2af7b](#))
- chore(deps): update all non-major dependencies ([5ef1b4a](#))
- chore(deps): update dependency types-setuptools to v69 ([de11192](#))

### Documentation

- docs: fix rst link typo in CONTRIBUTING.rst ([2b6da6e](#))

### Feature

- feat(api): add support for the Draft notes API (#2728)
- feat(api): add support for the Draft notes API
- fix(client): handle empty 204 responses in PUT requests ([ebf9d82](#))

### Fix

- fix(cli): add ability to disable SSL verification

Add a `--no-ssl-verify` option to disable SSL verification

Closes: #2714 ([3fe9fa6](#))

## 7.9.3 v4.2.0 (2023-11-28)

### Chore

- chore(deps): update all non-major dependencies ([8aeb853](#))
- chore(deps): update dessant/lock-threads action to v5 ([f4ce867](#))
- chore(deps): update all non-major dependencies ([9fe2335](#))
- chore(deps): update all non-major dependencies ([91e66e9](#))
- chore(deps): update all non-major dependencies ([d0546e0](#))

## Feature

- feat: add pipeline status as Enum

<https://docs.gitlab.com/ee/api/pipelines.html> (4954bbc)

- feat(api): add support for wiki attachments (#2722)

Added UploadMixin in mixin module Added UploadMixin dependency for Project, ProjectWiki, GroupWiki Added api tests for wiki upload Added unit test for mixin Added docs sections to wikis.rst (7b864b8)

## 7.9.4 v4.1.1 (2023-11-03)

### Chore

- chore(ci): add release id to workflow step (9270e10)
- chore(deps): update all non-major dependencies (32954fb)

### Documentation

- docs(users): add missing comma in v4 API create runner examples

The examples which show usage of new runner registration api endpoint are missing commas. This change adds the missing commas. (b1b2edf)

### Fix

- fix(build): include py.typed in dists (b928639)

## 7.9.5 v4.1.0 (2023-10-28)

### Chore

- chore(deps): update all non-major dependencies (bf68485)
- chore(CHANGELOG): re-add v4.0.0 changes using old format (258a751)
- chore(CHANGELOG): revert python-semantic-release format change (b5517e0)
- chore: add source label to container image (7b19278)
- chore(rtd): revert to python 3.11 (#2694) (1113742)

### Ci

- ci: remove unneeded GitLab auth (fd7bbfc)

## Feature

- feat: add Merge Request `merge_status` and `detailed_merge_status` values as constants ([e18a424](#))

## Fix

- fix: remove deprecated `MergeStatus` ([c6c012b](#))

## 7.9.6 v4.0.0 (2023-10-17)

### Breaking

- docs(advanced): document new netrc behavior

BREAKING CHANGE: python-gitlab now explicitly passes `auth` to requests, meaning it will only read netrc credentials if no token is provided, fixing a bug where netrc credentials took precedence over OAuth tokens. This also affects the CLI, where all environment variables now take precedence over netrc files. ([45b8930](#))

- refactor(build): build project using PEP 621

BREAKING CHANGE: python-gitlab now stores metadata in `pyproject.toml` as per PEP 621, with `setup.py` removed. pip version v21.1 or higher is required if you want to perform an editable install. ([71fca8c](#))

- refactor(const): remove deprecated global constant import

BREAKING CHANGE: Constants defined in `gitlab.const` can no longer be imported globally from `gitlab`. Import them from `gitlab.const` instead. ([e4a1f6e](#))

- refactor(list): `as_list` support is removed.

In `list()` calls support for the `as_list` argument has been removed. `as_list` was previously deprecated and now the use of `iterator` will be required if wanting to have same functionality as using `as_list`

BREAKING CHANGE: Support for the deprecated `as_list` argument in `list()` calls has been removed. Use `iterator` instead. ([9b6d89e](#))

- refactor(lint): remove deprecated `lint()` in favor of `ci_lint.create()`

BREAKING CHANGE: The deprecated `lint()` method is no longer available. Use `ci_lint.create()` instead. ([0b17a2d](#))

- refactor(artifacts): remove deprecated `artifact()` in favor of `artifacts.raw()`

BREAKING CHANGE: The deprecated `project.artifact()` method is no longer available. Use `project.artifacts.raw()` instead. ([90134c9](#))

- refactor(artifacts): remove deprecated `artifacts()` in favor of `artifacts.download()`

BREAKING CHANGE: The deprecated `project.artifacts()` method is no longer available. Use `project.artifacts.download()` instead. ([42639f3](#))

- refactor(groups): remove deprecated LDAP group link add/delete methods

BREAKING CHANGE: The deprecated `group.add_ldap_group_link()` and `group.delete_ldap_group_link()` methods are no longer available. Use `group.ldap_group_links.create()` and `group.ldap_group_links.delete()` instead. ([5c8b7c1](#))

- refactor(projects): remove deprecated `project.transfer_project()` in favor of `project.transfer()`

BREAKING CHANGE: The deprecated `project.transfer_project()` method is no longer available. Use `project.transfer()` instead. ([27ed490](#))

- fix(cli): remove deprecated `--all` option in favor of `--get-all`

BREAKING CHANGE: The `--all` option is no longer available in the CLI. Use `--get-all` instead. (e9d48cf)

- feat: remove support for Python 3.7, require 3.8 or higher

Python 3.8 is End-of-Life (EOL) as of 2023-06-27 as stated in <https://devguide.python.org/versions/> and <https://peps.python.org/pep-0537/>

By dropping support for Python 3.7 and requiring Python 3.8 or higher it allows python-gitlab to take advantage of new features in Python 3.8, which are documented at: <https://docs.python.org/3/whatsnew/3.8.html>

BREAKING CHANGE: As of python-gitlab 4.0.0, Python 3.7 is no longer supported. Python 3.8 or higher is required. (058d5a5)

## Chore

- chore(ci): follow upstream config for release build\_command (3e20a76)
- chore(ci): update release build for python-semantic-release v8 (#2692) (bf050d1)
- chore(deps): update pre-commit hook pycqa/pylint to v3 (0f4a346)
- chore(deps): update all non-major dependencies (1348a04)
- chore: add package pipelines API link (2a2404f)
- chore(ci): fix pre-commit deps and python version (1e7f257)
- chore(ci): remove Python 3.13 dev job (e8c50f2)
- chore(helpers): fix previously undetected flake8 issue (bf8bd73)
- chore: fix test names (f1654b8)
- chore: make linters happy (3b83d5d)
- chore: change `_update_uses` to `_update_method` and use an Enum

Change the name of the `_update_uses` attribute to `_update_method` and store an Enum in the attribute to indicate which type of HTTP method to use. At the moment it supports POST and PUT. But can in the future support PATCH. (7073a2d)

- chore(deps): update all non-major dependencies (ff45124)
- chore(deps): update dependency pylint to v3 (491350c)
- chore(deps): update pre-commit hook maxbrunet/pre-commit-renovate to v37 (b4951cd)
- chore(deps): update all non-major dependencies (0d49164)
- chore(deps): update dependency commitizen to v3.10.0 (becd8e2)
- chore(deps): update pre-commit hook commitizen-tools/commitizen to v3.10.0 (626c2f8)
- chore(deps): update all non-major dependencies (6093dbc)
- chore(deps): update all non-major dependencies (bb728b1)
- chore(deps): update all non-major dependencies to v23.9.1 (a16b732)
- chore(deps): update actions/checkout action to v4 (af13914)
- chore(deps): update all non-major dependencies (9083787)
- chore(deps): update dependency build to v1 (2e856f2)
- chore(deps): update all non-major dependencies (b6a3db1)

- chore(rtd): use readthedocs v2 syntax ([6ce2149](#))
- chore(rtd): fix docs build on readthedocs.io (#2654) ([3d7139b](#))
- chore(ci): adapt release workflow and config for v8 ([827fe9e](#))
- chore(deps): update relekang/python-semantic-release action to v8 ([c57c85d](#))
- chore(deps): update all non-major dependencies ([16f2d34](#))
- chore(deps): update all non-major dependencies ([5b33ade](#))
- chore(deps): update pre-commit hook maxbrunet/pre-commit-renovate to v36 ([db58cca](#))
- chore(deps): update dependency ubuntu to v22 ([8865552](#))
- chore(deps): update all non-major dependencies ([3732841](#))
- chore(deps): update dependency pytest-docker to v2 ([b87bb0d](#))
- chore: switch to docker-compose v2

Closes: #2625 ([713b5ca](#))

- chore: update PyYAML to 6.0.1

Fixes issue with CI having error: `AttributeError: cython_sources`

Closes: #2624 ([3b8939d](#))

- chore(deps): update all non-major dependencies ([511f45c](#))
- chore(deps): update all non-major dependencies ([d4a7410](#))
- chore(deps): update all non-major dependencies ([12846cf](#))
- chore(deps): update all non-major dependencies ([33d2aa2](#))
- chore(deps): update dependency types-setuptools to v68 ([bdd4eb6](#))
- chore(deps): update actions/upload-artifact action to v3 ([b78d6bf](#))
- chore(deps): update dependency setuptools to v68 ([0f06082](#))
- chore(deps): bring myst-parser up to date with sphinx 7 ([da03e9c](#))
- chore(deps): bring furo up to date with sphinx ([a15c927](#))
- chore(deps): update dependency sphinx to v7 ([2918dfd](#))
- chore(deps): update actions/checkout action to v3 ([e2af1e8](#))
- chore(deps): update actions/setup-python action to v4 ([e0d6783](#))
- chore(deps): update all non-major dependencies ([5ff56d8](#))
- chore(deps): pin pytest-console-scripts for 3.7 ([6d06630](#))
- chore(deps): update all non-major dependencies ([7586a5c](#))

## Documentation

- docs: correct error with back-ticks (#2653)

New linting package update detected the issue. ([0b98dd3](#))

- docs(access\_token): adopt token docs to 16.1

expires\_at is now required Upstream MR: [https://gitlab.com/gitlab-org/gitlab/-/merge\\_requests/124964](https://gitlab.com/gitlab-org/gitlab/-/merge_requests/124964) ([fe7a971](#))

- docs(files): fix minor typo in variable declaration ([118ce42](#))

## Feature

- feat(client): mask tokens by default when logging ([1611d78](#))
- feat(api): add ProjectPackagePipeline

Add ProjectPackagePipeline, which is scheduled to be included in GitLab 16.0 ([5b4addd](#))

- feat: officially support Python 3.12 ([2a69c0e](#))
- feat(packages): Allow uploading bytes and files

This commit adds a keyword argument to GenericPackageManager.upload() to allow uploading bytes and file-like objects to the generic package registry. That necessitates changing file path to be a keyword argument as well, which then cascades into a whole slew of checks to not allow passing both and to not allow uploading file-like objects as JSON data.

Closes <https://github.com/python-gitlab/python-gitlab/issues/1815> ([61e0fae](#))

- feat: Use requests AuthBase classes ([5f46cfd](#))
- feat(api): add support for job token scope settings ([59d6a88](#))
- feat(api): support project remote mirror deletion ([d900910](#))
- feat(api): add optional GET attrs for /projects/:id/ci/lint ([40a102d](#))
- feat(api): add support for new runner creation API (#2635)

Co-authored-by: Nejc Habjan <[hab.nejc@gmail.com](mailto:hab.nejc@gmail.com)> ([4abcd17](#))

- feat(releases): Add support for direct\_asset\_path

This commit adds support for the “new” alias for filepath: direct\_asset\_path (added in 15.10) in release links API. ([d054917](#))

- feat: Added iteration to issue and group filters ([8d2d297](#))

## Fix

- fix(cli): add \_from\_parent\_attrs to user-project manager (#2558) ([016d90c](#))
- fix(cli): fix action display in -help when there are few actions

fixes #2656 ([b22d662](#))

- fix(client): support empty 204 responses in http\_patch ([e15349c](#))
- fix(snippets): allow passing list of files ([31c3c5e](#))

## Test

- test: add tests for token masking ([163bfcf](#))
- test(cli): add test for user-project list ([a788cff](#))
- test: correct calls to `script_runner.run()`

Warnings were being raised. Resolve those warnings. ([cd04315](#))

- test: fix failing tests that use 204 (No Content) plus content

`urllib3>=2` now checks for expected content length. Also codes 204 and 304 are set to expect a content length of 0 [1]

So in the unit tests stop setting content to return in these situations.

[1] <https://github.com/urllib3/urllib3/blob/88a707290b655394aade060a8b7eae83152dc8b/src/urllib3/response.py#L691-L693> ([3074f52](#))

## Unknown

- chore(deps): update dependency requests to v2.31.0 [security]

Also update dependency `responses==0.23.3` as it provides support for `urllib3>=2`

Closes: [#2626](#) ([988a6e7](#))

## 7.9.7 v3.15.0 (2023-06-09)

### Chore

- chore(deps): update pre-commit hook maxbrunet/pre-commit-renovate to v35 ([8202e3f](#))
- chore: update sphinx from 5.3.0 to 6.2.1 ([c44a290](#))
- chore: update copyright year to include 2023 ([511c6e5](#))
- chore(deps): update all non-major dependencies ([e3de6ba](#))
- chore(deps): update pre-commit hook commitizen-tools/commitizen to v3 ([1591e33](#))
- chore(deps): update dependency types-setuptools to v67 ([c562424](#))
- chore(deps): update dependency requests-toolbelt to v1 ([86eba06](#))
- chore(deps): update dependency myst-parser to v1 ([9c39848](#))
- chore(deps): update dependency commitizen to v3 ([784d59e](#))
- chore(ci): use OIDC trusted publishing for pypi.org ([#2559](#))
- chore(ci): use OIDC trusted publishing for pypi.org
- chore(ci): explicitly install setuptools in tests ([7be09e5](#))

## Documentation

- docs: remove exclusive EE about issue links ([e0f6f18](#))

## Feature

- feat: add support for `select=&#34;package_file&#34;` in package upload

Add ability to use `select=&#34;package_file&#34;` when uploading a generic package as described in: [https://docs.gitlab.com/ee/user/packages/generic\\_packages/index.html](https://docs.gitlab.com/ee/user/packages/generic_packages/index.html)

Closes: [#2557](#) ([3a49f09](#))

- feat(api): add support for events scope parameter ([348f56e](#))
- feat: usernames support for MR approvals

This can be used instead of ‘user\_ids’

See: [https://docs.gitlab.com/ee/api/merge\\_request\\_approvals.html#create-project-level-rule](https://docs.gitlab.com/ee/api/merge_request_approvals.html#create-project-level-rule) ([a2b8c8c](#))

## 7.9.8 v3.14.0 (2023-04-11)

### Chore

- chore(ci): wait for all coverage reports in CI status ([511764d](#))
- chore(setup): depend on typing-extensions for 3.7 until EOL ([3abc557](#))
- chore: add Contributor Covenant 2.1 as Code of Conduct

See [https://www.contributor-covenant.org/version/2/1/code\\_of\\_conduct/](https://www.contributor-covenant.org/version/2/1/code_of_conduct/) ([fe334c9](#))

- chore(deps): update all non-major dependencies ([8b692e8](#))
- chore(deps): update dependency furo to v2023 ([7a1545d](#))
- chore(deps): update actions/stale action to v8 ([7ac4b86](#))
- chore(pre-commit): Bumping versions ([e973729](#))
- chore(.github): actually make PR template the default ([7a8a862](#))
- chore: use a dataclass to return values from `prepare_send_data`

I found the tuple of three values confusing. So instead use a dataclass to return the three values. It is still confusing but a little bit less so.

Also add some unit tests ([f2b5e4f](#))

- chore(contributing): refresh development docs ([d387d91](#))
- chore(github): add default pull request template ([bf46c67](#))
- chore(deps): update all non-major dependencies ([#2493](#))
- chore(deps): update all non-major dependencies
- chore(fixtures): downgrade GitLab for now
- chore(deps): ungroup typing deps, group gitlab instead
- chore(deps): downgrade argcomplete for now



Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com> Co-authored-by: Nejc Habjan <nejc.habjan@siemens.com> ([07d03dc](#))

- chore(deps): update dependency pre-commit to v3 (#2508)

Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com> ([7d779c8](#))

- chore(deps): update dependency coverage to v7 (#2501)

Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com> ([aee73d0](#))

- chore(deps): update dependency flake8 to v6 (#2502)

Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com> ([3d4596e](#))

- chore(renovate): swith to gitlab-ee ([8da48ee](#))
- chore(renovate): bring back custom requirements pattern ([ae0b21c](#))
- chore(deps): update mypy (1.0.0) and responses (0.22.0)

Update the `requirements-*` files.

In order to update `mypy==1.0.0` we need to also update `responses==0.22.0`

Fix one issue found by mypy

Leaving updates for `precommit` to be done in a separate commit by someone. ([9c24657](#))

- chore(renovate): do not ignore tests dir ([5b8744e](#))
- chore(deps): update all non-major dependencies ([2f06999](#))
- chore(deps): update pre-commit hook psf/black to v23 ([217a787](#))
- chore(deps): update black (23.1.0) and commitizen (2.40.0) (#2479)

Update the dependency versions: black: 23.1.0 commitizen: 2.40.0

They needed to be updated together as just updating `black` caused a dependency conflict.

Updated files by running `black` and committing the changes. ([44786ef](#))

- chore: add SECURITY.md ([572ca3b](#))
- chore: remove `pre-commit` as a default `tox` environment (#2470)

For users who use `tox` having `pre-commit` as part of the default environment list is redundant as it will run the same tests again that are being run in other environments. For example: `black`, `flake8`, `pylint`, and more. ([fde2495](#))

- chore: add Python 3.12 testing

Add a unit test for Python 3.12. This will use the latest version of Python 3.12 that is available from <https://github.com/actions/python-versions/>

At this time it is 3.12.0-alpha.4 but will move forward over time until the final 3.12 release and updates. So 3.12.0, 3.12.1, ... will be matched. ([0867564](#))

## Documentation

- docs(objects): fix typo in pipeline schedules ([3057f45](#))
- docs(advanced): clarify netrc, proxy behavior with requests ([1da7c53](#))
- docs: fix update badge behaviour

docs: fix update badge behaviour

Earlier: `badge.image_link = new_link`

Now: `badge.image_url = new_image_url` `badge.link_url = new_link_url` ([3d7ca1c](#))

- docs(advanced): fix typo in Gitlab examples ([1992790](#))

## Feature

- feat(projects): allow importing additional items from GitHub ([ce84f2e](#))
- feat(objects): support fetching PATs via id or `self` endpoint ([19b38bd](#))
- feat: add `resource_weight_event` for `ProjectIssue` ([6e5ef55](#))
- feat(backends): use PEP544 protocols for structural subtyping ([#2442](#))

The purpose of this change is to track API changes described in <https://github.com/python-gitlab/python-gitlab/blob/main/docs/api-levels.rst>, for example, for package versioning and breaking change announcements in case of protocol changes.

This is MVP implementation to be used by [#2435](#). ([4afeaff](#))

- feat(client): add `http_patch` method ([#2471](#))

In order to support some new API calls we need to support the HTTP PATCH method.

Closes: [#2469](#) ([f711d9e](#))

- feat(cli): add setting of `allow_force_push` for protected branch

For the CLI: add `allow_force_push` as an optional argument for creating a protected branch.

API reference: [https://docs.gitlab.com/ee/api/protected\\_branches.html#protect-repository-branches](https://docs.gitlab.com/ee/api/protected_branches.html#protect-repository-branches)

Closes: [#2466](#) ([929e07d](#))

## Fix

- fix(cli): warn user when no fields are displayed ([8bf53c8](#))
- fix(client): properly parse content-type when charset is present ([76063c3](#))
- fix: support int for `parent_id` in `import_group`

This will also fix other use cases where an integer is passed in to `MultipartEncoder`.

Added unit tests to show it works.

Closes: [#2506](#) ([90f96ac](#))

- fix(cli): add ability to escape at-prefixed parameter ([#2513](#))
- fix(cli): Add ability to escape at-prefixed parameter ([#2511](#))

Co-authored-by: Nejc Habjan <hab.nejc@gmail.com> (4f7c784)

- fix(cli): display items when iterator is returned (33a04e7)
- fix: typo fixed in docs (ee5f444)

## Refactor

- refactor(client): let mypy know http\_password is set (2dd177b)

## Test

- test(unit): increase V4 CLI coverage (5748d37)
- test(unit): split the last remaining unittest-based classes into modules” (14e0f65)
- test(unit): remove redundant package (4a9e3ee)
- test(unit): consistently use inline fixtures (1bc56d1)
- test(meta): move meta suite into unit tests

They’re always run with it anyway, so it makes no difference. (847004b)

- test(functional): clarify MR fixture factory name (d8fd1a8)

## Unknown

- Merge pull request #2465 from valentingregoire/typos

docs: fix typo in issue docs (43f5ac5)

- Merge branch ‘main’ into typos (3cfd390)

## 7.9.9 v3.13.0 (2023-01-30)

### Chore

- chore: make backends private (1e629af)
- chore(deps): update all non-major dependencies (ea7010b)
- chore: add a UserWarning if both iterator=True and page=X are used (#2462)

If a caller calls a `list()` method with both `iterator=True` (or `as_list=False`) and `page=X` then emit a `UserWarning` as the options are mutually exclusive. (8e85791)

- chore: remove tox envdir values

tox > 4 no longer will re-use the tox directory :( What this means is that with the previous config if you ran: `$ tox -e mypy; tox -e isort; tox -e mypy` It would recreate the tox environment each time :(

By removing the `envdir` values it will have the tox environments in separate directories and not recreate them.

They have an FAQ entry about this: <https://tox.wiki/en/latest/upgrading.html#re-use-of-environments> (3c7c7fc)

- chore: update attributes for create and update projects (aa44f2a)

- chore(deps): update all non-major dependencies ([122988c](#))
- chore(deps): update all non-major dependencies ([49c0233](#))
- chore(deps): update all non-major dependencies ([10c4f31](#))
- chore(deps): update all non-major dependencies ([bbd01e8](#))
- chore(deps): update actions/stale action to v7 ([76eb024](#))
- chore(ci): complete all unit tests even if one has failed (#2438) ([069c6c3](#))
- chore: add test, docs, and helper for 409 retries ([3e1c625](#))
- chore(deps): update all non-major dependencies ([6682808](#))
- chore(deps): update all non-major dependencies ([1816107](#))
- chore(deps): update pre-commit hook pycqa/flake8 to v6 ([82c61e1](#))
- chore: add docs for schedule pipelines ([9a9a6a9](#))
- chore(tox): ensure test envs have all dependencies ([63cf4e4](#))
- chore(deps): update pre-commit hook maxbrunet/pre-commit-renovate to v34.48.4 ([985b971](#))
- chore(deps): update dessant/lock-threads action to v4 ([337b25c](#))
- chore: Use SPDX license expression in project metadata ([acb3a4a](#))
- chore(deps): update actions/download-artifact action to v3 ([64ca597](#))
- chore(deps): update all non-major dependencies ([21e767d](#))

## Documentation

- docs(faq): describe and group common errors ([4c9a072](#))

## Feature

- feat(group): add support for group restore API ([9322db6](#))
- feat(client): automatically retry on HTTP 409 Resource lock

Fixes: [#2325](#) ([dced76a](#))

- feat(api): add support for bulk imports API ([043de2d](#))
- feat(api): add support for resource groups ([5f8b8f5](#))
- feat(api): support listing pipelines triggered by pipeline schedules ([865fa41](#))
- feat: allow filtering pipelines by source

See: <https://docs.gitlab.com/ee/api/pipelines.html#list-project-pipelines> Added in GitLab 14.3 ([b6c0872](#))

- feat(client): bootstrap the http backends concept (#2391) ([91a665f](#))
- feat: add resource iteration events (see [https://docs.gitlab.com/ee/api/resource\\_iteration\\_events.html](https://docs.gitlab.com/ee/api/resource_iteration_events.html)) ([ef5feb4](#))
- feat: allow passing kwargs to Gitlab class when instantiating with `from_config` (#2392) ([e88d34e](#))
- feat: add `keep_base_url` when getting configuration from file ([50a0301](#))

## Fix

- fix(client): regression - do not automatically get\_next if page=# and iterator=True/as\_list=False are used

This fix a regression introduced on commit <https://github.com/python-gitlab/python-gitlab/commit/1339d645ce58a2e1198b898b9549ba5917b1ff12>

If page is used, then get\_next should be false.

This was found on the mesa ci project, after upgrading the python-gitlab version, the script that monitors the ci was getting killed by consuming too much memory. ([585e3a8](#))

- fix: change return value to “None” in case getattr returns None to prevent error ([3f86d36](#))
- fix(deps): bump requests-toolbelt to fix deprecation warning ([faf842e](#))
- fix: Use the ProjectIterationManager within the Project object

The Project object was previously using the GroupIterationManager resulting in the incorrect API endpoint being used. Utilize the correct ProjectIterationManager instead.

Resolves #2403 ([44f05dc](#))

- fix(api): Make description optional for releases ([5579750](#))

## Refactor

- refactor: add reason property to RequestsResponse (#2439) ([b59b7bd](#))
- refactor: remove unneeded requests.utils import (#2426) ([6fca651](#))
- refactor: Migrate MultipartEncoder to RequestsBackend (#2421) ([43b369f](#))
- refactor: move Response object to backends (#2420) ([7d9ce0d](#))
- refactor: move the request call to the backend (#2413) ([283e7cc](#))
- refactor: Moving RETRYABLE\_TRANSIENT\_ERROR\_CODES to const ([887852d](#))

## Test

- test(functional): do not require config file ([43c2dda](#))
- test(unit): expand tests for pipeline schedules ([c7cf0d1](#))

### 7.9.10 v3.12.0 (2022-11-28)

## Chore

- chore: validate httpx package is not installed by default ([0ecf3bb](#))
- chore(deps): update all non-major dependencies ([d8a657b](#))
- chore(deps): update pre-commit hook maxbrunet/pre-commit-renovate to v34.24.0 ([a0553c2](#))
- chore: correct website for pylint

Use <https://github.com/PyCQA/pylint> as the website for pylint. ([fcd72fe](#))

- chore(deps): update pre-commit hook maxbrunet/pre-commit-renovate to v34.20.0 ([e6f1bd6](#))
- chore(deps): update all non-major dependencies ([b2c6d77](#))

- chore(deps): update pre-commit hook maxbrunet/pre-commit-renovate to v34 (623e768)

## Documentation

- docs: Use the term “log file” for getting a job log file

The GitLab docs refer to it as a log file: <https://docs.gitlab.com/ee/api/jobs.html#get-a-log-file>

“trace” is the endpoint name but not a common term people will think of for a “log file” (9d2b1ad)

- docs(groups): describe GitLab.com group creation limitation (9bd433a)
- docs(api): pushrules remove saying None is returned when not found

In `groups.pushrules.get()`, GitLab does not return None when no rules are found. GitLab returns a 404.

Update docs to not say it will return None

Also update docs in `project.pushrules.get()` to be consistent. Not 100% sure if it returns None or returns a 404, but we don’t need to document that.

Closes: #2368 (c3600b4)

## Feature

- feat: add support for SAML group links (#2367) (1020ce9)
- feat(groups): add LDAP link manager and deprecate old API endpoints (3a61f60)
- feat(groups): add support for listing ldap\_group\_links (#2371) (ad7c8fa)
- feat: implement secure files API (d0a0348)
- feat(ci): Re-Run Tests on PR Comment workflow (034cde3)
- feat(api): add support for getting a project’s pull mirror details

Add the ability to get a project’s pull mirror details. This was added in GitLab 15.5 and is a PREMIUM feature.

<https://docs.gitlab.com/ee/api/projects.html#get-a-projects-pull-mirror-details> (060cfe1)

- feat(api): add support for remote project import from AWS S3 (#2357) (892281e)
- feat(api): add support for remote project import (#2348) (e5dc72d)
- feat(api): add application statistics (6fcf3b6)

## Fix

- fix(cli): Enable debug before doing auth

Authentication issues are currently hard to debug since `--debug` only has effect after `gl.auth()` has been called.

For example, a 401 error is printed without any details about the actual HTTP request being sent:

```
$ gitlab --debug --server-url https://gitlab.com current-user get
401: 401 Unauthorized
```

By moving the call to `gl.enable_debug()` the usual debug logs get printed before the final error message.

Signed-off-by: Emanuele Aina <[emanuele.aina@collabora.com](mailto:emanuele.aina@collabora.com)> (65abb85)

- fix(cli): expose missing `mr_default_target_self` project attribute

Example::

```
gitlab project update -id 616 --mr-default-target-self 1
```

References:

- [https://gitlab.com/gitlab-org/gitlab/-/merge\\_requests/58093](https://gitlab.com/gitlab-org/gitlab/-/merge_requests/58093)
- [https://gitlab.com/gitlab-org/gitlab/-/blob/v13.11.0-ee/doc/user/project/merge\\_requests/creating\\_merge\\_requests.md#new-merge-request-from-a-fork](https://gitlab.com/gitlab-org/gitlab/-/blob/v13.11.0-ee/doc/user/project/merge_requests/creating_merge_requests.md#new-merge-request-from-a-fork)
- <https://gitlab.com/gitlab-org/gitlab/-/blob/v14.7.0-ee/doc/api/projects.md#get-single-project> ([12aea32](#))
- fix: use POST method and return dict in `cancel_merge_when_pipeline_succeeds()` (#2350)
- Call was incorrectly using a PUT method when should have used a POST method.
  - Changed return type to a dict as GitLab only returns { 'status': 'success' } on success. Since the function didn't work previously, this should not impact anyone.
  - Updated the test fixture `merge_request` to add ability to create a pipeline.
  - Added functional test for `mr.cancel_merge_when_pipeline_succeeds()`

Fixes: #2349 ([bd82d74](#))

## Refactor

- refactor: explicitly use `ProjectSecureFile` ([0c98b2d](#))

## Test

- test(api): fix flaky test `test_cancel_merge_when_pipeline_succeeds`

This is an attempt to fix the flaky test `test_cancel_merge_when_pipeline_succeeds`. Were seeing a: 405 Method Not Allowed error when setting the MR to `merge_when_pipeline_succeeds`.

Closes: #2383 ([6525c17](#))

## Unknown

- Merge pull request #2347 from Shreya-7/issue-2264-add-application-statistics

feat(api): add application statistics ([31ec146](#))

- Merge pull request #2351 from python-gitlab/renovate/all-minor-patch

chore(deps): update all non-major dependencies ([2974966](#))

- Merge pull request #2352 from python-gitlab/renovate/maxbrunet-pre-commit-renovate-34.x

chore(deps): update pre-commit hook maxbrunet/pre-commit-renovate to v34 ([c3d9820](#))

### 7.9.11 v3.11.0 (2022-10-28)

#### Chore

- chore: add responses to pre-commit deps ([4b8ddc7](#))
- chore: add basic type checks to functional/api tests ([5b642a5](#))
- chore: add basic typing to functional tests ([ee143c9](#))
- chore: narrow type hints for license API ([50731c1](#))
- chore: add basic type checks to meta tests ([545d6d6](#))
- chore: add basic typing to smoke tests ([64e8c31](#))
- chore: add basic typing to test root ([0b2f6bc](#))
- chore(deps): update pre-commit hook maxbrunet/pre-commit-renovate to v33 ([932bbde](#))
- chore(deps): update all non-major dependencies ([dde3642](#))
- chore: add not-callable to pylint ignore list

The not-callable error started showing up. Ignore this error as it is invalid. Also mypy tests for these issues.

Closes: [#2334](#) ([f0c02a5](#))

- chore: revert compose upgrade

This reverts commit [f825d70e25feae8cd9da84e768ec6075edbc2200](#). ([dd04e8e](#))

- chore(deps): update all non-major dependencies ([2966234](#))
- chore: use kwargs for http\_request docs ([124abab](#))
- chore(deps): pin GitHub Actions ([8dbaa5c](#))
- chore(deps): group non-major upgrades to reduce noise ([37d14bd](#))
- chore(deps): pin and clean up test dependencies ([60b9197](#))
- chore(deps): pin dependencies ([953f38d](#))
- chore: topic functional tests ([d542eba](#))
- chore: renovate and precommit cleanup ([153d373](#))
- chore(deps): update black to v22.10.0 ([531ee05](#))
- chore(deps): update dependency types-requests to v2.28.11.2 ([d47c0f0](#))
- chore: fix flaky test ([fdd4114](#))
- chore: update the issue templates
- Have an option to go to the discussions
- Have an option to go to the Gitter chat
- Move the bug/issue template into the .github/ISSUE\_TEMPLATE/ directory ([c15bd33](#))
- chore: simplify wait\_for\_sidekiq usage

Simplify usage of wait\_for\_sidekiq by putting the assert if it timed out inside the function rather than after calling it. ([196538b](#))

- chore(deps): update dependency pylint to v2.15.3 ([6627a60](#))
- chore(deps): update dependency mypy to v0.981 ([da48849](#))



- chore(deps): update dependency commitizen to v2.35.0 ([4ce9559](#))
- chore(deps): update typing dependencies ([81285fa](#))

## Documentation

- docs(advanced): add hint on type narrowing ([a404152](#))
- docs: add minimal docs about the `enable_debug()` method

Add some minimal documentation about the `enable_debug()` method. ([b4e9ab7](#))

- docs(commits): fix commit create example for binary content ([bcc1eb4](#))
- docs(readme): add a basic feature list ([b4d53f1](#))
- docs(api): describe use of lower-level methods ([b7a6874](#))
- docs(api): describe the `list()` and `all()` runners' functions ([b6cc3f2](#))
- docs(api): Update `merge_requests.rst`: `mr_id` to `mr_iid`

Typo: Author probably meant `mr_iid` (i.e. project-specific MR ID) and **not** `mr_id` (i.e. server-wide MR ID)

Closes: <https://github.com/python-gitlab/python-gitlab/issues/2295>

Signed-off-by: Stavros Ntentos <133706+stdedos@users.noreply.github.com> ([b32234d](#))

## Feature

- feat(build): officially support Python 3.11 ([74f66c7](#))
- feat(api): add support for topics merge API ([9a6d197](#))

## Fix

- fix: remove `project.approvals.set_approvals()` method

The `project.approvals.set_approvals()` method used the `/projects/:id/approvers` end point. That end point was removed from GitLab in the 13.11 release, on 2-Apr-2021 in commit [27dc2f2fe81249bbdc25f7bd8fe799752aac05e6](#) via merge commit [e482597a8cf1bae8e27abd6774b684fb90491835](#). It was deprecated on 19-Aug-2019.

See merge request: [https://gitlab.com/gitlab-org/gitlab/-/merge\\_requests/57473](https://gitlab.com/gitlab-org/gitlab/-/merge_requests/57473) ([91f08f0](#))

- fix: use epic id instead of iid for epic notes ([97cae38](#))
- fix(cli): handle list response for json/yaml output

Handle the case with the CLI where a list response is returned from GitLab and json/yaml output is requested.

Add a functional CLI test to validate it works.

Closes: [#2287](#) ([9b88132](#))

- fix: intermittent failure in `test_merge_request_reset_approvals`

Have been seeing intermittent failures in the test: `tests/functional/api/test_merge_requests.py::test_merge_request_reset_approvals`

Also saw a failure in: `tests/functional/cli/test_cli_v4.py::test_accept_request_merge[subprocess]`

Add a call to `wait_for_sidekiq()` to hopefully resolve the issues. ([3dde36e](#))

## Refactor

- refactor: pre-commit trigger from tox ([6e59c12](#))
- refactor: migrate legacy EE tests to pytest ([88c2505](#))
- refactor: pytest-docker fixtures ([3e4781a](#))
- refactor(deps): drop compose v1 dependency in favor of v2 ([f825d70](#))

## Test

- test: fix `test_project_push_rules` test

Make the `test_project_push_rules` test work. ([8779cf6](#))

- test: enable skipping tests per GitLab plan ([01d5f68](#))
- test: use `false` instead of `/usr/bin/false`

On Debian systems `false` is located at `/bin/false` (`coreutils` package). This fixes unit test failure on Debian system:

`FileNotFoundError: [Errno 2] No such file or directory: '/usr/bin/false' /usr/lib/python3.10/subprocess.py:1845:`  
`FileNotFoundError` ([51964b3](#))

## Unknown

- Merge pull request #2345 from python-gitlab/jlvillal/enable\_debug

docs: add minimal docs about the `enable_debug()` method ([8f74a33](#))

- Merge pull request #2343 from python-gitlab/feat/python-3-11

feat(build): officially support Python 3.11 ([a3b4824](#))

- Merge pull request #2341 from python-gitlab/renovate/maxbrunet-pre-commit-renovate-33.x

chore(deps): update pre-commit hook maxbrunet/pre-commit-renovate to v33 ([31a39e1](#))

- Merge pull request #2320 from lmlbaum/refactoring

refactor: pre-commit triggered from tox ([eec6c02](#))

- Merge pull request #2333 from python-gitlab/jlvillal/remove\_approvers\_endpoint

fix: remove `project.approvals.set_approvals()` method ([eb54adf](#))

- Merge pull request #2332 from python-gitlab/jlvillal/fix\_test

test: fix `test_project_push_rules` test ([c676b43](#))

- Merge pull request #2322 from AndreySV/fix-test-with-false

test: use `false` instead of `/usr/bin/false` ([4eca9b9](#))

- Merge pull request #2318 from python-gitlab/renovate/all-minor-patch

chore(deps): update all non-major dependencies ([9410acb](#))

## 7.9.12 v3.10.0 (2022-09-28)

### Chore

- chore: bump GitLab docker image to 15.4.0-ee.0
- Use `settings.delayed_group_deletion=False` as that is the recommended method to turn off the delayed group deletion now.
- Change test to look for `default` as `pages` is not mentioned in the docs[1]

[1] [https://docs.gitlab.com/ee/api/sidekiq\\_metrics.html#get-the-current-queue-metrics](https://docs.gitlab.com/ee/api/sidekiq_metrics.html#get-the-current-queue-metrics) (b87a2bc)

- chore(deps): update black to v22.8.0 (86b0e40)
- chore(deps): update dependency types-requests to v2.28.10 (5dde7d4)
- chore(deps): update dependency pytest to v7.1.3 (ec7f26c)
- chore(deps): update dependency commitizen to v2.32.5 (e180f14)
- chore(deps): update pre-commit hook commitizen-tools/commitizen to v2.32.2 (31ba64f)
- chore(deps): update dependency commitizen to v2.32.2 (31aea28)

### Feature

- feat: Add `reset_approvals` api

Added the newly added `reset_approvals` merge request api.

Signed-off-by: Lucas Zampieri <lzampier@redhat.com> (88693ff)

### Fix

- fix(cli): add missing attributes for creating MRs (1714d0a)
- fix(cli): add missing attribute for MR changes (20c46a0)

### Unknown

- Merge pull request #2280 from python-gitlab/jlvillal/docker\_image

chore: bump GitLab docker image to 15.4.0-ee.0 (fceeabc)

- Merge pull request #2261 from python-gitlab/renovate/commitizen-2.x

chore(deps): update dependency commitizen to v2.32.2 (336ee21)

- Merge pull request #2262 from python-gitlab/renovate/commitizen-tools-commitizen-2.x

chore(deps): update pre-commit hook commitizen-tools/commitizen to v2.32.2 (89bf581)

- Merge pull request #2254 from python-gitlab/jlvillal/deploy\_approve

feat: add support for deployment approval endpoint (56fbe02)

### 7.9.13 v3.9.0 (2022-08-28)

#### Chore

- chore: Only check for our UserWarning

The GitHub CI is showing a ResourceWarning, causing our test to fail.

Update test to only look for our UserWarning which should not appear.

What was seen when debugging the GitHub CI: {message: ResourceWarning( “unclosed <socket.socket fd=12, family=AddressFamily.AF\_INET, type=SocketKind.SOCK\_STREAM, proto=6, laddr=(‘127.0.0.1’, 50862), raddr=(‘127.0.0.1’, 8080)>” ), category: ‘ResourceWarning’, filename: ‘/home/runner/work/python-gitlab/python-gitlab/.tox/api\_func\_v4/lib/python3.10/site-packages/urllib3/poolmanager.py’, lineno: 271, line: None } ([bd4dfb4](#))

- chore: fix issue if only run test\_gitlab.py func test

Make it so can run just the test\_gitlab.py functional test.

For example: `$ tox -e api_func_v4 -- -k test_gitlab.py` ([98f1956](#))

- chore(ci): make pytest annotations work ([f67514e](#))
- chore(deps): update pre-commit hook commitizen-tools/commitizen to v2.32.1 ([cdd6efe](#))
- chore(deps): update dependency commitizen to v2.32.1 ([9787c5c](#))
- chore(deps): update dependency types-requests to v2.28.9 ([be932f6](#))
- chore(deps): update dependency types-setuptools to v64 ([4c97f26](#))
- chore(deps): update pre-commit hook pycqa/flake8 to v5 ([835d884](#))
- chore(deps): update dependency types-requests to v2.28.8 ([8e5b86f](#))
- chore(deps): update pre-commit hook commitizen-tools/commitizen to v2.31.0 ([71d37d9](#))
- chore(deps): update dependency commitizen to v2.31.0 ([4ff0894](#))

#### Feature

- feat: add support for deployment approval endpoint

Add support for the deployment approval endpoint[1]

[1] <https://docs.gitlab.com/ee/api/deployments.html#approve-or-reject-a-blocked-deployment> Closes: [#2253](#) ([9c9eeb9](#))

- feat: add support for merge\_base API ([dd4fbd5](#))

#### Unknown

- Merge pull request #2255 from python-gitlab/jlvillal/noop

chore: fix issue if only run test\_gitlab.py func test ([e095735](#))

- Merge pull request #2241 from python-gitlab/renovate/pycqa-flake8-5.x

chore(deps): update pre-commit hook pycqa/flake8 to v5 ([13d4927](#))

- Merge pull request #2239 from python-gitlab/renovate/commitizen-tools-commitizen-2.x

chore(deps): update pre-commit hook commitizen-tools/commitizen to v2.31.0 ([9381a44](#))

- Merge pull request #2238 from python-gitlab/renovate/commitizen-2.x

chore(deps): update dependency commitizen to v2.31.0 ([b432e47](#))

### 7.9.14 v3.8.1 (2022-08-10)

#### Chore

- chore(deps): update dependency commitizen to v2.29.5 ([181390a](#))
- chore(deps): update dependency flake8 to v5.0.4 ([50a4fec](#))
- chore(deps): update dependency sphinx to v5 ([3f3396e](#))
- chore: remove broad Exception catching from `config.py`

Change “except Exception:” catching to more granular exceptions.

A step in enabling the “broad-except” check in pylint. ([0abc90b](#))

- chore: add license badge to readme ([9aecc9e](#))
- chore: consolidate license and authors ([366665e](#))

#### Fix

- fix(client): do not assume user attrs returned for `auth()`

This is mostly relevant for people mocking the API in tests. ([a07547c](#))

#### Unknown

- Merge pull request #2233 from python-gitlab/fix/do-not-require-web-url ([99d580a](#))
- Merge pull request #2153 from python-gitlab/renovate/sphinx-5.x

chore(deps): update dependency sphinx to v5 ([1e12eaf](#))

- Merge pull request #2212 from python-gitlab/jlvillal/config

chore: remove broad Exception catching from `config.py` ([70e67bf](#))

### 7.9.15 v3.8.0 (2022-08-04)

#### Chore

- chore: use `urlunparse` instead of string replace

Use the `urlunparse()` function to reconstruct the URL without the query parameters. ([6d1b62d](#))

- chore(ci): bump semantic-release for fixed commit parser ([1e063ae](#))
- chore: enable mypy check `disallow_any_generics` ([24d17b4](#))
- chore: enable mypy check `no_implicit_optional` ([64b208e](#))
- chore(deps): update dependency types-requests to v2.28.6 ([54dd4c3](#))
- chore(deps): update dependency flake8 to v5 ([cdc384b](#))

- chore(deps): update pre-commit hook commitizen-tools/commitizen to v2.29.2 ([4988c02](#))
- chore(deps): update dependency commitizen to v2.29.2 ([30274ea](#))
- chore: change `_repr_attr` for Project to be `path_with_namespace`

Previously `_repr_attr` was `path` but that only gives the basename of the path. So `https://gitlab.com/gitlab-org/gitlab` would only show “gitlab”. Using `path_with_namespace` it will now show “gitlab-org/gitlab” ([7cccefe](#))

- chore: make code PEP597 compliant

Use `encoding='utf-8'` in `open()` and open-like functions.

<https://peps.python.org/pep-0597/> ([433dba0](#))

- chore: enable mypy check `warn_return_any`

Update code so that the `warn_return_any` check passes. ([76ec4b4](#))

- chore(clusters): deprecate clusters support

Cluster support was deprecated in GitLab 14.5 [1]. And disabled by default in GitLab 15.0 [2]

- Update docs to mark clusters as deprecated
- Remove testing of clusters

[1] [https://docs.gitlab.com/ee/api/project\\_clusters.html](https://docs.gitlab.com/ee/api/project_clusters.html) [2] <https://gitlab.com/groups/gitlab-org/configure/-/epics/8> ([b46b379](#))

- chore(topics): ‘title’ is required when creating a topic

In GitLab `>= 15.0` `title` is required when creating a topic. ([271f688](#))

## Documentation

- docs: describe self-revoking personal access tokens ([5ea48fc](#))

## Feature

- feat(client): warn user on misconfigured URL in `auth()` ([0040b43](#))
- feat: Support downloading archive subpaths ([cadb0e5](#))

## Fix

- fix(client): ensure encoded query params are never duplicated ([1398426](#))
- fix: optionally keep user-provided base URL for pagination (#2149) ([e2ea8b8](#))

## Refactor

- refactor(client): factor out URL check into a helper (af21a18)
- refactor(client): remove handling for incorrect link header

This was a quirk only present in GitLab 13.0 and fixed with 13.1. See [https://gitlab.com/gitlab-org/gitlab/-/merge\\_requests/33714](https://gitlab.com/gitlab-org/gitlab/-/merge_requests/33714) and <https://gitlab.com/gitlab-org/gitlab/-/issues/218504> for more context. (77c04b1)

## Test

- test(unit): reproduce duplicate encoded query params (6f71c66)
- test: attempt to make functional test startup more reliable

The functional tests have been erratic. Current theory is that we are starting the tests before the GitLab container is fully up and running.

- Add checking of the Health Check[1] endpoints.
- Add a 20 second delay after we believe it is up and running.
- Increase timeout from 300 to 400 seconds

[1] [https://docs.gitlab.com/ee/user/admin\\_area/monitoring/health\\_check.html](https://docs.gitlab.com/ee/user/admin_area/monitoring/health_check.html) (67508e8)

## Unknown

- Merge pull request #2221 from python-gitlab/jlvillal/unparse chore: use `urlunparse` instead of string replace (9e0b60f)
- Merge pull request #2219 from python-gitlab/fix/no-duplicate-params fix(client): ensure encoded query params are never duplicated (d263f57)
- Merge pull request #2220 from python-gitlab/chore/bump-semantic-release chore(ci): bump semantic-release for fixed commit parser (2ebfc70)
- Merge pull request #2211 from python-gitlab/jlvillal/mypy\_step\_step chore: enable mypy check `disallow_any_generics` (1136b17)
- Merge pull request #2208 from python-gitlab/renovate/commitizen-tools-commitizen-2.x chore(deps): update pre-commit hook commitizen-tools/commitizen to v2.29.2 (6cedbc8)
- Merge pull request #2210 from python-gitlab/jlvillal/mypy\_step\_by\_step chore: enable mypy check `no_implicit_optional` (1c91b24)
- Merge pull request #2209 from python-gitlab/renovate/flake8-5.x chore(deps): update dependency flake8 to v5 (d81cec3)
- Merge pull request #2203 from python-gitlab/jlvillal/project\_repr chore: change `_repr_attr` for Project to be `path_with_namespace` (98bdb98)
- Merge pull request #2188 from python-gitlab/jlvillal/fix\_functional\_ci test: attempt to make functional test startup more reliable (17414f7)
- Merge pull request #2199 from orf/patch-1

Support downloading archive subpaths ([5e1df65](#))

- Merge pull request #2157 from python-gitlab/jlvillal/mypy\_step\_by\_step

chore: enable mypy check `warn_return_any` ([b8be32a](#))

- Merge pull request #2201 from python-gitlab/jlvillal/encoding\_warning

chore: make code PEP597 compliant ([1b7cd31](#))

- Merge pull request #2194 from python-gitlab/jlvillal/update-gitlab

test(functional): bump GitLab docker image to 15.2.0-ee.0 ([7a53c69](#))

## 7.9.16 v3.7.0 (2022-07-28)

### Chore

- chore: revert “test(functional): simplify token creation”

This reverts commit 67ab24fe5ae10a9f8cc9122b1a08848e8927635d. ([4b798fc](#))

- chore: enable using GitLab EE in functional tests

Enable using GitLab Enterprise Edition (EE) in the functional tests. This will allow us to add functional tests for EE only features in the functional tests. ([17c01ea](#))

- chore(deps): update dependency mypy to v0.971 ([7481d27](#))
- chore(deps): update pre-commit hook commitizen-tools/commitizen to v2.29.0 ([ad8d62a](#))
- chore(deps): update typing dependencies ([f2209a0](#))
- chore(deps): update dependency commitizen to v2.29.0 ([c365be1](#))
- chore(authors): fix email and do the ABC ([9833632](#))
- chore: make `reset_gitlab()` better

Saw issues in the CI where `reset_gitlab()` would fail. It would fail to delete the group that is created when GitLab starts up. Extending the timeout didn't fix the issue.

Changed the code to use the new `helpers.safe_delete()` function. Which will delete the resource and then make sure it is deleted before returning.

Also added some logging functionality that can be seen if logging is turned on in pytest. ([d87d6b1](#))

- chore: fixtures: after `delete()` wait to verify deleted

In our fixtures that create:

- groups
- project merge requests
- projects
- users

They delete the created objects after use. Now wait to ensure the objects are deleted before continuing as having unexpected objects existing can impact some of our tests. ([1f73b6b](#))

- chore: add a lazy boolean attribute to `RESTObject`

This can be used to tell if a `RESTObject` was created using `lazy=True`.

Add a message to the `AttributeError` if attribute access fails for an instance created with `lazy=True`. ([a7e8cfb](#))



- chore: enable mypy check `strict_equality`

Enable the mypy `strict_equality` check. ([a29cd6c](#))

- chore: change name of API functional test to `api_func_v4`

The CLI test is `cli_func_v4` and using `api_func_v4` matches with that naming convention. ([8cf5cd9](#))

- chore(deps): update typing dependencies ([e772248](#))
- chore(deps): update pre-commit hook `pycqa/pylint` to v2.14.5 ([c75a1d8](#))
- chore(deps): update dependency `pylint` to v2.14.5 ([e153636](#))
- chore(deps): update pre-commit hook `commitizen-tools/commitizen` to v2.28.0 ([d238e1b](#))
- chore(deps): update dependency `commitizen` to v2.28.0 ([8703dd3](#))
- chore(deps): update `black` to v22.6.0 ([82bd596](#))
- chore(deps): update pre-commit hook `pycqa/pylint` to v2.14.4 ([5cd39be](#))
- chore(ci\_lint): add create attributes ([6e1342f](#))
- chore(deps): update dependency `requests` to v2.28.1 ([be33245](#))
- chore(deps): update dependency `pylint` to v2.14.4 ([2cee2d4](#))
- chore: simplify multi-nested try blocks

Instead of have a multi-nested series of try blocks. Convert it to a more readable series of `if` statements. ([e734470](#))

- chore(docs): convert tabs to spaces

Some tabs snuck into the documentation. Convert them to 4-spaces. ([9ea5520](#))

- chore: fix misspelling ([2d08fc8](#))

## Documentation

- docs(cli): showcase use of token scopes ([4a6f8d6](#))
- docs(projects): document export with upload to URL ([03f5484](#))
- docs: describe fetching existing export status ([9c5b8d5](#))
- docs(authors): add John ([e2afb84](#))
- docs: document CI Lint usage ([d5de4b1](#))
- docs(users): add docs about listing a user's projects

Add docs about listing a user's projects.

Update docs on the membership API to update the URL to the upstream docs and also add a note that it requires Administrator access to use. ([065a1a5](#))

- docs: update return type of `pushrules`

Update the return type of `pushrules` to surround `None` with back-ticks to make it code-formatted. ([53cbecc](#))

- docs: describe ROPC flow in place of password authentication ([91c17b7](#))
- docs(readme): Remove redundant `-v` that breaks the command Remove redundant `-v` that breaks the command ([c523e18](#))

## Feature

- feat: allow sort/ordering for project releases

See: <https://docs.gitlab.com/ee/api/releases/#list-releases> (b1dd284)

- feat(cli): add a custom help formatter

Add a custom argparse help formatter that overrides the output format to list items vertically.

The formatter is derived from `argparse.HelpFormatter` with minimal changes.

Co-authored-by: John Villalovos <[john@sodarock.com](mailto:john@sodarock.com)> Co-authored-by: Nejc Habjan <[nejc.habjan@siemens.com](mailto:nejc.habjan@siemens.com)> (005ba93)

- feat: add support for iterations API (194ee01)
- feat(groups): add support for shared projects API (66461ba)
- feat(issues): add support for issue reorder API (8703324)
- feat(namespaces): add support for namespace existence API (4882cb2)
- feat: add support for group and project invitations API (7afd340)
- feat(projects): add support for project restore API (4794ecc)
- feat: add support for filtering jobs by scope

See: ‘scope’ here: <https://docs.gitlab.com/ee/api/jobs.html#list-project-jobs> (0e1c0dd)

- feat: add `asdict()` and `to_json()` methods to Gitlab Objects

Add an `asdict()` method that returns a dictionary representation copy of the Gitlab Object. This is a copy and changes made to it will have no impact on the Gitlab Object.

The `asdict()` method name was chosen as both the `dataclasses` and `attrs` libraries have an `asdict()` function which has the similar purpose of creating a dictionary representation of an object.

Also add a `to_json()` method that returns a JSON string representation of the object.

Closes: #1116 (08ac071)

- feat(api): add support for instance-level registry repositories (284d739)
- feat(groups): add support for group-level registry repositories (70148c6)
- feat: Add ‘merge\_pipelines\_enabled’ project attribute

Boolean. Enable or disable merge pipelines.

See: <https://docs.gitlab.com/ee/api/projects.html#edit-project> [https://docs.gitlab.com/ee/ci/pipelines/merged\\_results\\_pipelines.html](https://docs.gitlab.com/ee/ci/pipelines/merged_results_pipelines.html) (fc33c93)

- feat: support validating CI lint results (3b1ede4)
- feat(cli): add support for global CI lint (3f67c4b)
- feat(objects): add Project CI Lint support

Add support for validating a project’s CI configuration [1]

[1] <https://docs.gitlab.com/ee/api/lint.html> (b213dd3)

- feat: add support for group push rules

Add the `GroupPushRules` and `GroupPushRulesManager` classes.

Closes: #1259 (b5cdc09)

- feat(api): add support for get for a MR approval rule

In GitLab 14.10 they added support to get a single merge request approval rule [1]

Add support for it to ProjectMergeRequestApprovalRuleManager

[1] [https://docs.gitlab.com/ee/api/merge\\_request\\_approvals.html#get-a-single-merge-request-level-rule](https://docs.gitlab.com/ee/api/merge_request_approvals.html#get-a-single-merge-request-level-rule) (89c18c6)

## Fix

- fix: support array types for most resources (d9126cd)
- fix: use the [] after key names for array variables in params

1. If a value is of type ArrayAttribute then append '[' to the name of the value for query parameters (params).

This is step 3 in a series of steps of our goal to add full support for the GitLab API data types[1]:

- array
- hash
- array of hashes

Step one was: commit 5127b1594c00c7364e9af15e42d2e2f2d909449b Step two was: commit a57334f1930752c70ea15847a39324fa94042460

Fixes: #1698

[1] <https://docs.gitlab.com/ee/api/#encoding-api-parameters-of-array-and-hash-types> (1af44ce)

- fix(runners): fix listing for /runners/all (c6dd57c)
- fix(config): raise error when gitlab id provided but no config section found (1ef7018)
- fix(config): raise error when gitlab id provided but no config file found (ac46c1c)
- fix: add get\_all param (and --get-all) to allow passing all to API (7c71d5d)
- fix: results returned by attributes property to show updates

Previously the attributes method would show the original values in a Gitlab Object even if they had been updated. Correct this so that the updated value will be returned.

Also use copy.deepcopy() to ensure that modifying the dictionary returned can not also modify the object. (e5affc8)

- fix: Enable epic notes

Add the notes attribute to GroupEpic (5fc3216)

- fix(cli): remove irrelevant MR approval rule list filters (0daec5f)
- fix: ensure path elements are escaped

Ensure the path elements that are passed to the server are escaped. For example a "/" will be changed to "%2F"

Closes: #2116 (5d9c198)

## Refactor

- refactor: migrate services to integrations ([a428051](#))
- refactor(objects): move ci lint to separate file ([6491f1b](#))
- refactor(test-projects): apply suggestions and use fixtures ([a51f848](#))

## Test

- test(cli): add tests for token scopes ([263fe3d](#))
- test: add test to show issue fixed

<https://github.com/python-gitlab/python-gitlab/issues/1698> has been fixed. Add test to show that. ([75bec7d](#))

- test(functional): bump GitLab docker image to 15.2.0-ee.0

Use the GitLab docker image 15.2.0-ee.0 in the functional testing. ([69014e9](#))

- test: always ensure clean config environment ([8d4f13b](#))
- test(ee): add an EE specific test ([10987b3](#))
- test(functional): simplify token creation ([67ab24f](#))
- test: fix broken test if user had config files

Use `monkeypatch` to ensure that no config files are reported for the test.

Closes: [#2172](#) ([864fc12](#))

- test: allow `podman` users to run functional tests

Users of `podman` will likely have `DOCKER_HOST` set to something like `unix:///run/user/1000/podman/podman.sock`

Pass this environment variable so that it will be used during the functional tests. ([ff215b7](#))

- test(api\_func\_v4): catch deprecation warning for `gl.lint()`

Catch the deprecation warning for the call to `gl.lint()`, so it won't show up in the log. ([95fe924](#))

- test(functional): use both `get_all` and `all` in `list()` tests ([201298d](#))
- test: add more tests for container registries ([f6b6e18](#))
- test(functional): replace `len()` calls with list membership checks ([97e0eb9](#))

## Unknown

- Merge pull request [#2198](#) from [nickbroon/nickbroon-release-sort-order](#)

feat: allow sort/ordering for project releases ([c33cb20](#))

- Merge pull request [#2195](#) from [python-gitlab/jlvillal/array\\_test](#)

test: add test to show issue fixed ([1cf5932](#))

- Merge pull request [#1699](#) from [python-gitlab/jlvillal/arrays](#)

fix: use the `[]` after key names for array variables in `params` ([510ec30](#))

- Merge pull request [#1778](#) from [python-gitlab/jlvillal/gitlab-ee](#)

chore: enable using GitLab EE in functional tests ([b661003](#))

- Merge pull request #2184 from python-gitlab/renovate/commitizen-tools-commitizen-2.x

chore(deps): update pre-commit hook commitizen-tools/commitizen to v2.29.0 ([346cf76](#))

- Merge pull request #2182 from python-gitlab/renovate/commitizen-2.x

chore(deps): update dependency commitizen to v2.29.0 ([e2ca0b4](#))

- Merge pull request #2173 from python-gitlab/jlvillal/config\_test\_fix

test: fix broken test if user had config files ([1ecbc7c](#))

- Merge pull request #2166 from python-gitlab/jlvillal/podman

test: allow podman users to run functional tests ([0b02b95](#))

- Merge pull request #1785 from python-gitlab/jlvillal/reset\_gitlab

chore: make reset\_gitlab() better ([789ef81](#))

- Merge pull request #1784 from python-gitlab/jlvillal/sidekiq

chore: fixtures: after delete() wait to verify deleted ([916b1db](#))

- Merge pull request #2163 from python-gitlab/jlvillal/lint\_warning

test(api\_func\_v4): catch deprecation warning for `gl.lint()` ([1855279](#))

- Merge pull request #2161 from nickbroon/nickbroon-jobs\_scope

feat: add support for filtering jobs by scope ([0549afa](#))

- Merge pull request #2160 from python-gitlab/docs-author-add-john

docs(authors): add John ([ead9f15](#))

- Merge pull request #1872 from python-gitlab/jlvillal/as\_dict

feat: add `asdict()` and `to_json()` methods to Gitlab Objects ([fcbced8](#))

- Merge pull request #2082 from python-gitlab/jlvillal/mark\_lazy\_state

chore: add a lazy boolean attribute to `RESTObject` ([2c90fd0](#))

- Merge pull request #2146 from python-gitlab/jlvillal/mypy\_strict\_step\_by\_step

chore: enable mypy check `strict_equality` ([c84379d](#))

- Merge pull request #2147 from python-gitlab/jlvillal/api\_func\_v4

chore: change name of API functional test to `api_func_v4` ([ed110bd](#))

- Merge pull request #2141 from nickbroon/nickbroon-merge\_pipelines\_enabled

feat: Add 'merge\_pipelines\_enabled' project attribute ([e409811](#))

- Merge pull request #2125 from python-gitlab/jlvillal/user\_docs

docs(users): add docs about listing a user's projects ([1fbfb22](#))

- Merge pull request #1896 from python-gitlab/jlvillal/ci\_lint

feat: add Project CI Lint support ([d15fea0](#))

- Merge pull request #2126 from python-gitlab/jlvillal/push\_rules

docs: update return type of `pushrules` ([88a1535](#))

- Merge pull request #1266 from gokeefe/gokeefe/group\_push\_rules

#1259 Add `GroupPushRules` and `GroupPushRulesManager` classes ([768890a](#))

- Merge pull request #2114 from python-gitlab/jlvillal/remove\_trys

chore: simplify multi-nested try blocks ([3df404c](#))

- Merge pull request #2117 from python-gitlab/jlvillal/encodedid\_path

fix: ensure path elements are escaped ([04c6063](#))

- Merge pull request #2113 from tuxiqae/patch-1

Remove redundant -v that breaks the command ([ca3b438](#))

- Merge pull request #2069 from antoineauger/test/unit-tests-projects

test(projects): add unit tests for projects ([0e4db56](#))

- Merge pull request #2110 from python-gitlab/jlvillal/mr\_approval\_rules

feat(api): add support for get for a MR approval rule ([389e1e6](#))

- Merge pull request #2111 from python-gitlab/jlvillal/meta\_fix

chore: fix misspelling ([6486566](#))

- Merge pull request #2112 from python-gitlab/jlvillal/doc\_remove\_tabs

chore(docs): convert tabs to spaces ([8771ad8](#))

## 7.9.17 v3.6.0 (2022-06-28)

### Chore

- chore(deps): ignore python-semantic-release updates ([f185b17](#))
- chore(workflows): explicitly use python-version ([eb14475](#))
- chore(deps): update typing dependencies ([acc5c39](#))
- chore(deps): update actions/setup-python action to v4 ([77c1f03](#))
- chore(deps): update pre-commit hook pycqa/pylint to v2.14.3 ([d1fe838](#))
- chore(ci): increase timeout for docker container to come online

Have been seeing timeout issues more and more. Increase timeout from 200 seconds to 300 seconds (5 minutes). ([bda020b](#))

- chore(docs): ignore nitpicky warnings ([1c3efb5](#))
- chore: patch sphinx for explicit re-exports ([06871ee](#))
- chore: bump mypy pre-commit hook ([0bbcad7](#))
- chore(gitlab): fix implicit re-exports for mpyy ([981b844](#))
- chore: add link to Commitizen in Github workflow

Add a link to the Commitizen website in the Github workflow. Hopefully this will help people when their job fails. ([d08d07d](#))

- chore(deps): update dependency pylint to v2.14.3 ([9a16bb1](#))
- chore(deps): update pre-commit hook commitizen-tools/commitizen to v2.27.1 ([22c5db4](#))
- chore(deps): update dependency requests to v2.28.0 ([d361f4b](#))
- chore: fix issue found with pylint==2.14.3

A new error was reported when running pylint==2.14.3: gitlab/client.py:488:0: W1404: Implicit string concatenation found in call (implicit-str-concat)

Fixed this issue. ([eeab035](#))

- chore(deps): update dependency mypy to v0.961 ([f117b2f](#))
- chore(deps): update typing dependencies ([aebf9c8](#))
- chore(deps): update dependency mypy to v0.960 ([8c016c7](#))
- chore(cli): rename “object” to “GitLab resource”

Make the parser name more user friendly by renaming from generic “object” to “GitLab resource” ([62e64a6](#))

- chore: use multiple processors when running PyLint

Use multiple processors when running PyLint. On my system it took about 10.3 seconds to run PyLint before this change. After this change it takes about 5.8 seconds to run PyLint. ([7f2240f](#))

- chore: rename `whaction` and `action` to `resource_action` in CLI

Rename the variables `whaction` and `action` to `resource_action` to improve code-readability. ([fb3f28a](#))

- chore: enable pylint check: “redefined-outer-name”,

Enable the pylint check “redefined-outer-name” and fix the errors detected. ([1324ce1](#))

- chore: enable pylint check: “no-self-use”

Enable the pylint check “no-self-use” and fix the errors detected. ([80aadaaf](#))

- chore: enable pylint check: “no-else-return”

Enable the pylint check “no-else-return” and fix the errors detected. ([d0b0811](#))

- chore: enable pylint check: “attribute-defined-outside-init”

Enable the pylint check: “attribute-defined-outside-init” and fix errors detected. ([d6870a9](#))

- chore: enable pylint check “raise-missing-from”

Enable the pylint check “raise-missing-from” and fix errors detected. ([1a2781e](#))

- chore: enable pylint checks which require no changes

Enabled the pylint checks that don’t require any code changes. Previously these checks were disabled. ([50fdbc4](#))

- chore: enable pylint checks

Enable the pylint checks:

- unnecessary-pass
- unspecified-encoding

Update code to resolve errors found ([1e89164](#))

- chore: rename `what` to `gitlab_resource`

Naming a variable `what` makes it difficult to understand what it is used for.

Rename it to `gitlab_resource` as that is what is being stored.

The Gitlab documentation talks about them being resources: [https://docs.gitlab.com/ee/api/api\\_resources.html](https://docs.gitlab.com/ee/api/api_resources.html)

This will improve code readability. ([c86e471](#))

- chore: rename `__call__()` to `run()` in `GitlabCLI`

Less confusing to have it be a normal method. ([6189437](#))

- chore: enable ‘consider-using-sys-exit’ pylint check

Enable the ‘consider-using-sys-exit’ pylint check and fix errors raised. ([0afcc3e](#))

- chore: require f-strings

We previously converted all string formatting to use f-strings. Enable pylint check to enforce this. ([96e994d](#))

- chore(ci): pin 3.11 to beta.1 ([7119f2d](#))
- chore(cli): ignore coverage on exceptions triggering cli.die ([98ccc3c](#))
- chore: move `utils._validate_attrs` inside `types.RequiredOptional`

Move the `validate_attrs` function to be inside the `RequiredOptional` class. It makes sense for it to be part of the class as it is working on data related to the class. ([9d629bb](#))

- chore: remove use of ‘%’ string formatter in `gitlab/utils.py`

Replace usage with f-string ([0c5a121](#))

- chore: have `EncodedId` creation always return `EncodedId`

There is no reason to return an `int` as we can always return a `str` version of the `int`

Change `EncodedId` to always return an `EncodedId`. This removes the need to have `mypy` ignore the error raised. ([a1a246f](#))

- chore: move `RequiredOptional` to the `gitlab.types` module

By having `RequiredOptional` in the `gitlab.base` module it makes it difficult with circular imports. Move it to the `gitlab.types` module which has no dependencies on any other `gitlab` module. ([7d26530](#))

- chore: update type-hints return signature for `GetWithoutIdMixin` methods

Commit `f0152dc3cc9a42aa4dc3c0014b4c29381e9b39d6` removed situation where `get()` in a `GetWithoutIdMixin` based class could return `None`

Update the type-hints to no longer return `Optional` AKA `None` ([aa972d4](#))

- chore(deps): update dependency commitizen to v2.27.1 ([456f9f1](#))
- chore(deps): update typing dependencies ([f3f79c1](#))
- chore: correct `ModuleNotFoundError()` arguments

Previously in commit `233b79ed442aac66faf9eb4b0087ea126d6dffc5` I had used the `name` argument for `ModuleNotFoundError()`. This basically is the equivalent of not passing any message to `ModuleNotFoundError()`. So when the exception was raised it wasn’t very helpful.

Correct that and add a unit-test that shows we get the message we expect. ([0b7933c](#))

- chore(mixins): remove `None` check as `http_get` always returns value ([f0152dc](#))

## Documentation

- docs(api): add separate section for advanced usage ([22ae101](#))
- docs(api): document usage of `head()` methods ([f555bfb](#))
- docs(projects): provide more detailed import examples ([8f8611a](#))
- docs(projects): document 404 gotcha with unactivated integrations ([522ecff](#))
- docs(variables): instruct users to follow GitLab rules for values ([194b6be](#))
- docs(api): stop linking to `python-requests.org` ([49c7e83](#))



- docs(api): fix incorrect docs for merge\_request\_approvals (#2094)
- docs(api): fix incorrect docs for merge\_request\_approvals

The `set_approvers()` method is on the `ProjectApprovalManager` class. It is not part of the `ProjectApproval` class.

The docs were previously showing to call `set_approvers` using a `ProjectApproval` instance, which would fail. Correct the documentation.

This was pointed out by a question on the Gitter channel.

Co-authored-by: Nejc Habjan <nejc.habjan@siemens.com> (5583eaa)

- docs(api-usage): add import os in example (2194a44)
- docs: drop deprecated setuptools build\_sphinx (048d66a)
- docs(usage): refer to upstream docs instead of custom attributes (ae7d3b0)
- docs(ext): fix rendering for RequiredOptional dataclass (4d431e5)
- docs: documentation updates to reflect addition of mutually exclusive attributes (24b720e)
- docs: use `as_list=False` or `all=True` in Getting started

In the “Getting started with the API” section of the documentation, use either `as_list=False` or `all=True` in the example usages of the `list()` method.

Also add a warning about the fact that `list()` by default does not return all items. (de8c6e8)

## Feature

- feat(downloads): allow streaming downloads access to response iterator (#1956)
- feat(downloads): allow streaming downloads access to response iterator

Allow access to the underlying response iterator when downloading in streaming mode by specifying `iterator=True`.

Update type annotations to support this change.

- docs(api-docs): add iterator example to artifact download

Document the usage of the `iterator=True` option when downloading artifacts

- test(packages): add tests for streaming downloads (b644721)
- feat(api): support head() method for get and list endpoints (ce9216c)
- feat(api): implement HEAD method (90635a7)
- feat(users): add approve and reject methods to User

As requested in #1604.

Co-authored-by: John Villalovos <john@sodarock.com> (f57139d)

- feat(api): convert gitlab.const to Enums

This allows accessing the elements by value, i.e.:

```
import gitlab.const
gitlab.const.AccessLevel(20) (c3c6086)
```

- feat: Add support for Protected Environments
- [https://docs.gitlab.com/ee/api/protected\\_environments.html](https://docs.gitlab.com/ee/api/protected_environments.html)
- <https://github.com/python-gitlab/python-gitlab/issues/1130>

no write operation are implemented yet as I have no use case right now and am not sure how it should be done ([1dc9d0f](#))

- feat(users): add ban and unban methods ([0d44b11](#))
- feat(docker): provide a Debian-based slim image ([384031c](#))
- feat: support mutually exclusive attributes and consolidate validation to fix board lists (#2037)

add exclusive tuple to RequiredOptional data class to support for mutually exclusive attributes

consolidate `_check_missing_create_attrs` and `_check_missing_update_attrs` from `mixins.py` into `_validate_attrs` in `utils.py`

change `_create_attrs` in board list manager classes from `required=('label_id',)` to `exclusive=('label_id','assignee_id','milestone_id')`

closes <https://github.com/python-gitlab/python-gitlab/issues/1897> ([3fa330c](#))

- feat(client): introduce `iterator=True` and deprecate `as_list=False` in `list()`

`as_list=False` is confusing as it doesn't explain what is being returned. Replace it with `iterator=True` which more clearly explains to the user that an iterator/generator will be returned.

This maintains backward compatibility with `as_list` but does issue a `DeprecationWarning` if `as_list` is set. ([cdc6605](#))

## Fix

- fix(base): do not fail `repr()` on lazy objects ([1efb123](#))
- fix(cli): project-merge-request-approval-rule

Using the CLI the command: `gitlab project-merge-request-approval-rule list --mr-iid 1 --project-id foo/bar`

Would raise an exception. This was due to the fact that `_id_attr` and `_repr_attr` were set for keys which are not returned in the response.

Add a unit test which shows the `repr` function now works. Before it did not.

This is an EE feature so we can't functional test it.

Closes: #2065 ([15a242c](#))

- fix(cli): fix project export download for CLI

Since [ac1c619cae6481833f5df91862624bf0380fef67](#) we delete parent arg keys from the args dict so this has been trying to access the wrong attribute. ([5d14867](#))

## Refactor

- refactor: do not recommend plain `gitlab.const` constants ([d652133](#))
- refactor: avoid possible breaking change in iterator (#2107)

Commit [b6447211754e126f64e12fc735ad74fe557b7fb4](#) inadvertently introduced a possible breaking change as it added a new argument `iterator` and added it in between existing (potentially positional) arguments.

This moves the `iterator` argument to the end of the argument list and requires it to be a keyword-only argument. ([212ddfc](#))

- refactor: remove no-op id argument in `GetWithoutIdMixin` ([0f2a602](#))
- refactor(test-projects): remove `test_restore_project` ([9be0875](#))
- refactor(mixins): extract custom type transforms into `utils` ([09b3b22](#))

## Test

- test: add tests and clean up usage for new enums ([323ab3c](#))
- test(pylint): enable pylint “unused-argument” check

Enable the pylint “unused-argument” check and resolve issues it found.

- Quite a few functions were accepting `**kwargs` but not then passing them on through to the next level. Now pass `**kwargs` to next level.
- Other functions had no reason to accept `**kwargs`, so remove it
- And a few other fixes. ([23feae9](#))
- test(api): add tests for HEAD method ([b0f02fa](#))
- test(projects): add unit tests for projects ([67942f0](#))
- test: move back to using latest Python 3.11 version ([8c34781](#))
- test: increase client coverage ([00aec96](#))
- test: add more tests for RequiredOptional ([ce40fde](#))
- test(cli): improve coverage for custom actions ([7327f78](#))
- test(gitlab): increase unit test coverage ([df072e1](#))

## Unknown

- Merge pull request #2105 from python-gitlab/renovate/actions-setup-python-4.x

chore(deps): update actions/setup-python action to v4 ([ebd5795](#))

- Merge pull request #2100 from python-gitlab/jlvillal/pylint\_2022-06-26

test(pylint): enable pylint “unused-argument” check ([3c3f865](#))

- Merge pull request #2061 from bgamari/patch-1

feat(users): add approve and reject methods to User ([f9b7c7b](#))

- Merge pull request #2089 from python-gitlab/jlvillal/more\_time

chore(ci): increase timeout for docker container to come online ([a825844](#))

- Merge pull request #1688 from jspricke/enum

feat(api): Convert gitlab.const to Enums ([f0ac3cd](#))

- Merge pull request #2084 from calve/protected-environments

feat: Add support for Protected Environments ([1feabc0](#))

- Merge pull request #2083 from python-gitlab/jlvillal/cz

chore: add link to Commitizen in Github workflow ([8342f53](#))

- Merge pull request #2066 from python-gitlab/jlvillal/approval\_rule\_id

fix(cli): project-merge-request-approval-rule ([41ceaca](#))

- Merge pull request #2076 from python-gitlab/renovate/pylint-2.x

chore(deps): update dependency pylint to v2.14.3 ([e48ad91](#))

- Merge pull request #2077 from python-gitlab/jlvillal/pylint

chore: fix issue found with pylint==2.14.3 ([8e0cd8b](#))

- Merge pull request #2064 from antoineauger/feat/user-ban-unban

feat(users): add ban and unban methods ([ca98d88](#))

- Merge pull request #2057 from walterrowe/main

docs: update docs to reflect addition of mutually exclusive attributes ([0f607f6](#))

- Merge pull request #2055 from python-gitlab/jlvillal/resource

chore(cli): rename “object” to “GitLab resource” ([0e3c461](#))

- Merge pull request #2045 from python-gitlab/jlvillal/test\_validate\_attrs

test: add more tests for RequiredOptional ([40c9b4f](#))

- Merge pull request #2056 from python-gitlab/jlvillal/pylint\_job

chore: use multiple processors when running PyLint ([7b9bb3c](#))

- Merge pull request #2051 from python-gitlab/jlvillal/more\_more\_pylint

chore: enable more pylint checks ([7a5923c](#))

- Merge pull request #2054 from python-gitlab/jlvillal/resource

chore: rename whaction and action to resource\_action in CLI ([61b8beb](#))

- Merge pull request #2049 from python-gitlab/jlvillal/python\_311

test: move back to using latest Python 3.11 version ([6cdccd9](#))

- Merge pull request #2053 from python-gitlab/jlvillal/resource

chore: rename what to gitlab\_resource ([8d30b15](#))

- Merge pull request #2052 from python-gitlab/jlvillal/cli\_minior\_clean

chore: rename \_\_call\_\_() to run() in GitlabCLI ([4eb5bad](#))

- Merge pull request #2050 from python-gitlab/jlvillal/more\_pylint

chore: enable ‘consider-using-sys-exit’ pylint check ([9ab3c10](#))

- Merge pull request #2043 from python-gitlab/jlvillal/f-string

chore: require f-strings ([3d000d3](#))

- Merge pull request #2042 from python-gitlab/jlvillal/exclusive

Clean-up the validate\_attrs method/function ([28cf3c3](#))

- Merge pull request #2040 from python-gitlab/jlvillal/type\_alias

chore: have EncodedId creation always return EncodedId ([dea9435](#))

- Merge pull request #2039 from python-gitlab/jlvillal/required\_optional

chore: move RequiredOptional to the gitlab.types module ([37eb8e0](#))

- Merge pull request #2036 from python-gitlab/jlvillal/get\_not\_none

chore: update type-hints return signature for GetWithoutIdMixin methods ([1f17349](#))

- Merge pull request #2033 from python-gitlab/jlvillal/module\_not\_found\_error

chore: correct ModuleNotFoundError() arguments ([b2e6f3b](#))

- Merge pull request #2034 from python-gitlab/renovate/typing-dependencies

chore(deps): update typing dependencies ([38218e5](#))

- Merge pull request #2035 from python-gitlab/renovate/commitizen-2.x

chore(deps): update dependency commitizen to v2.27.1 ([91e60ba](#))

- Merge pull request #2032 from python-gitlab/jlvillal/i\_hate\_as\_list

feat(client): introduce `iterator=True` and deprecate `as_list=False` in `list()` ([c51b538](#))

- Merge pull request #1884 from python-gitlab/jlvillal/list\_docs

docs: use `as_list=False` or `all=True` in Getting started ([5ae18d0](#))

## 7.9.18 v3.5.0 (2022-05-28)

### Chore

- chore(ci): fix prefix for action version ([1c02189](#))
- chore(ci): pin semantic-release version ([0ea61cc](#))
- chore(deps): update pre-commit hook pycqa/pylint to v2.13.9 ([1e22790](#))
- chore(deps): update dependency pylint to v2.13.9 ([4224950](#))
- chore: run the `pylint` check by default in tox

Since we require `pylint` to pass in the CI. Let's run it by default in tox. ([55ace1d](#))

- chore: rename the test which runs `flake8` to be `flake8`

Previously the test was called `pep8`. The test only runs `flake8` so call it `flake8` to be more precise. ([78b4f99](#))

- chore(deps): update pre-commit hook pycqa/pylint to v2.13.8 ([1835593](#))
- chore(deps): update dependency pylint to v2.13.8 ([b235bb0](#))
- chore: add `cz` to default tox environment list and `skip_missing_interpreters`

Add the `cz` (`comittizen`) check by default.

Set `skip_missing_interpreters = True` so that when a user runs tox and doesn't have a specific version of Python it doesn't mark it as an error. ([ba8c052](#))

- chore: exclude `build/` directory from mypy check

The `build/` directory is created by the tox environment `twine-check`. When the `build/` directory exists mypy will have an error. ([989a12b](#))

- chore(deps): update dependency types-requests to v2.27.25 ([d6ea47a](#))
- chore(ci): replace commitlint with commitizen ([b8d15fe](#))
- chore(renovate): set schedule to reduce noise ([882fe7a](#))
- chore(deps): update dependency types-requests to v2.27.24 ([f88e3a6](#))
- chore(deps): update dependency types-requests to v2.27.23 ([a6fed8b](#))

## Documentation

- docs: update issue example and extend API usage docs ([aad71d2](#))
- docs(CONTRIBUTING.rst): fix link to conventional-changelog commit format documentation ([2373a4f](#))
- docs: add missing Admin access const value

As shown here, Admin access is set to 60: [https://docs.gitlab.com/ee/api/protected\\_branches.html#protected-branches-api](https://docs.gitlab.com/ee/api/protected_branches.html#protected-branches-api) ([3e0d4d9](#))

- docs(merge\_requests): add new possible merge request state and link to the upstream docs

The actual documentation do not mention the locked state for a merge request ([e660fa8](#))

## Feature

- feat(objects): support get project storage endpoint ([8867ee5](#))
- feat: display human-readable attribute in `repr()` if present ([6b47c26](#))
- feat(ux): display `project.name_with_namespace` on project repr

This change the repr from:

```
$ gitlab.projects.get(id=some_id) <Project id:some_id>
```

To:

```
$ gitlab.projects.get(id=some_id) <Project id:some_id name_with_namespace:"group_name / project_name">
```

This is especially useful when working on random projects or listing of projects since users generally don't remember projects ids. ([e598762](#))

## Fix

- fix(cli): changed default `allow_abbrev` value to fix arguments collision problem (#2013)

fix(cli): change default `allow_abbrev` value to fix argument collision ([d68cacf](#))

- fix: duplicate subparsers being added to argparse

Python 3.11 added an additional check in the argparse library which detected duplicate subparsers being added. We had duplicate subparsers being added.

Make sure we don't add duplicate subparsers.

Closes: #2015 ([f553fd3](#))

## Test

- test(projects): add tests for list project methods ([fa47829](#))

## Unknown

- Merge pull request #2022 from MichaelSweikata/feat/documentation-update  
docs: update issue example and extend API usage docs ([792cee9](#))
- Merge pull request #2012 from rnoberger/rnoberger/test-projects  
test: increase projects coverage ([fd9154e](#))
- Merge pull request #2019 from python-gitlab/jlvillal/tox  
Some improvements to our tox environment defaults ([d121d2d](#))
- Merge pull request #2018 from python-gitlab/renovate/pycqa-pylint-2.x  
chore(deps): update pre-commit hook pycqa/pylint to v2.13.8 ([9e64645](#))
- Merge pull request #2017 from python-gitlab/renovate/pylint-2.x  
chore(deps): update dependency pylint to v2.13.8 ([0049f83](#))
- Merge pull request #2014 from python-gitlab/jlvillal/python3.11beta1  
fix: duplicate subparsers being added to argparse ([7d5a0c9](#))
- Merge pull request #1996 from Psychojoker/project-name-in-repr  
feat(ux): display project.name\_with\_namespace on project repr ([82c9a07](#))
- Merge pull request #2008 from python-gitlab/jlvillal/mypy-tox  
chore: exclude build/ directory from mypy check ([d169983](#))
- Merge pull request #2009 from python-gitlab/jlvillal/tox-env  
chore: add cz to default tox environment list and skip\_missing\_interpreters ([5cb2859](#))
- Merge pull request #2005 from carlosduelo/patch-1  
merge request can have state locked ([ef207da](#))
- Merge pull request #2001 from python-gitlab/renovate/typing-dependencies  
chore(deps): update dependency types-requests to v2.27.25 ([04e0d24](#))
- Merge pull request #1994 from python-gitlab/renovate/typing-dependencies  
chore(deps): update dependency types-requests to v2.27.24 ([79b903d](#))

## 7.9.19 v3.4.0 (2022-04-28)

### Chore

- chore(deps): update dependency mypy to v0.950 ([241e626](#))
- chore(deps): update dependency types-requests to v2.27.22 ([22263e2](#))
- chore(deps): update dependency types-requests to v2.27.21 ([0fb0955](#))
- chore(deps): update dependency pytest to v7.1.2 ([fd3fa23](#))
- chore(deps): update typing dependencies ([c12466a](#))
- chore(deps): update pre-commit hook pycqa/pylint to v2.13.7 ([1396221](#))
- chore(deps): update dependency pylint to v2.13.7 ([5fb2234](#))

- chore(deps): update typing dependencies ([d27cc6a](#))
- chore(deps): update dependency pylint to v2.13.5 ([5709675](#))
- chore(deps): update pre-commit hook pycqa/pylint to v2.13.5 ([17d5c6c](#))
- chore(deps): update codecov/codecov-action action to v3 ([292e91b](#))
- chore(deps): update dependency types-setuptools to v57.4.12 ([6551353](#))
- chore(client): remove duplicate code ([5cbbf26](#))
- chore(deps): update dependency types-requests to v2.27.16 ([ad799fc](#))
- chore(deps): upgrade gitlab-ce to 14.9.2-ce.0 ([d508b18](#))
- chore(deps): update pre-commit hook pycqa/pylint to v2.13.4 ([9d0b252](#))
- chore(deps): update dependency pylint to v2.13.4 ([a9a9392](#))
- chore(deps): update pre-commit hook pycqa/pylint to v2.13.3 ([8f0a3af](#))
- chore(deps): update dependency pylint to v2.13.3 ([0ae3d20](#))
- chore(deps): update black to v22.3.0 ([8d48224](#))

## Documentation

- docs(api-docs): docs fix for application scopes ([e1ad93d](#))

## Feature

- feat: emit a warning when using a `list()` method returns max

A common cause of issues filed and questions raised is that a user will call a `list()` method and only get 20 items. As this is the default maximum of items that will be returned from a `list()` method.

To help with this we now emit a warning when the result from a `list()` method is greater-than or equal to 20 (or the specified `per_page` value) and the user is not using either `all=True`, `all=False`, `as_list=False`, or `page=X`. ([1339d64](#))

- feat(objects): support getting project/group deploy tokens by id ([fcd37fe](#))
- feat(user): support getting user SSH key by id ([6f93c05](#))
- feat(api): re-add topic delete endpoint

This reverts commit [e3035a799a484f8d6c460f57e57d4b59217cd6de](#). ([d1d96bd](#))

## Fix

- fix: add `ChunkedEncodingError` to list of retryable exceptions ([7beb20f](#))
- fix: avoid passing redundant arguments to API ([3431887](#))
- fix(cli): add missing filters for project commit list ([149d244](#))
- fix: add 52x range to retry transient failures and tests ([c3ef1b5](#))
- fix: also retry HTTP-based transient errors ([3b49e4d](#))



## Unknown

- Merge pull request #1988 from python-gitlab/renovate/mypy-0.x  
chore(deps): update dependency mypy to v0.950 ([0c0035e](#))
- Merge pull request #1986 from python-gitlab/renovate/typing-dependencies  
chore(deps): update dependency types-requests to v2.27.21 ([ab8352e](#))
- Merge pull request #1983 from python-gitlab/renovate/pytest-7.x  
chore(deps): update dependency pytest to v7.1.2 ([4a6bd90](#))
- Merge pull request #1981 from python-gitlab/renovate/typing-dependencies  
chore(deps): update typing dependencies ([cf2953e](#))
- Merge pull request #1980 from python-gitlab/renovate/pycqa-pylint-2.x  
chore(deps): update pre-commit hook pycqa/pylint to v2.13.7 ([0305979](#))
- Merge pull request #1979 from python-gitlab/renovate/pylint-2.x  
chore(deps): update dependency pylint to v2.13.7 ([396e30c](#))
- Merge pull request #1965 from python-gitlab/fix/redundant-args-api  
fix: avoid passing redundant arguments to API ([ba7692a](#))
- Merge pull request #1974 from Sineaggi/add-chunked-to-list-of-retryable-exceptions  
Add ChunkedEncodingError to list of retryable exceptions ([07a16af](#))
- Merge pull request #1963 from python-gitlab/feat/deploy-token-get  
feat(objects): support getting project/group deploy tokens by id ([69ace2d](#))
- Merge pull request #1962 from python-gitlab/feat/user-ssh-key  
feat(user): support getting user SSH key by id ([68bf5d8](#))
- Merge pull request #1875 from python-gitlab/jlvillal/list\_warning  
feat: emit a warning when using a `list()` method returns max ([4d6f125](#))
- Merge pull request #1971 from python-gitlab/renovate/pycqa-pylint-2.x  
chore(deps): update pre-commit hook pycqa/pylint to v2.13.5 ([5370979](#))
- Merge pull request #1964 from python-gitlab/fix/missing-commit-list-filters  
fix(cli): add missing filters for project commit list ([3b0806e](#))
- Merge pull request #1968 from python-gitlab/renovate/codecov-codecov-action-3.x  
chore(deps): update codecov/codecov-action action to v3 ([de8cfd9](#))
- Merge pull request #1967 from python-gitlab/renovate/typing-dependencies  
chore(deps): update dependency types-setuptools to v57.4.12 ([3be5ac2](#))
- Merge pull request #1904 from Sineaggi/retry-additional-http-transient-errors  
Retry additional http transient errors ([0353bd4](#))
- Merge pull request #1961 from python-gitlab/renovate/typing-dependencies  
chore(deps): update dependency types-requests to v2.27.16 ([19ab07d](#))
- Merge pull request #1959 from python-gitlab/renovate/pycqa-pylint-2.x

chore(deps): update pre-commit hook pycqa/pylint to v2.13.4 ([8db6841](#))

- Merge pull request #1958 from python-gitlab/renovate/pylint-2.x

chore(deps): update dependency pylint to v2.13.4 ([400d8e5](#))

- Merge pull request #1951 from wacuuu/main

docs: small docs fix-up for application scopes ([8e241e4](#))

- Merge pull request #1954 from python-gitlab/renovate/pycqa-pylint-2.x

chore(deps): update pre-commit hook pycqa/pylint to v2.13.3 ([eee173e](#))

- Merge pull request #1953 from python-gitlab/renovate/pylint-2.x

chore(deps): update dependency pylint to v2.13.3 ([5498f9e](#))

- Merge pull request #1952 from python-gitlab/renovate/black

chore(deps): update black to v22.3.0 ([f942e65](#))

## 7.9.20 v3.3.0 (2022-03-28)

### Chore

- chore(deps): update dependency sphinx to v4.5.0 ([36ab769](#))
- chore(deps): update pre-commit hook pycqa/pylint to v2.13.2 ([14d367d](#))
- chore(deps): update dependency pylint to v2.13.2 ([10f15a6](#))
- chore(deps): update dependency types-requests to v2.27.15 ([2e8ecf5](#))
- chore(deps): update pre-commit hook pycqa/pylint to v2.13.1 ([1d0c6d4](#))
- chore(deps): update dependency types-requests to v2.27.14 ([be6b54c](#))
- chore(deps): update dependency pylint to v2.13.1 ([eefd724](#))
- chore(deps): update pre-commit hook pycqa/pylint to v2.13.0 ([9fe60f7](#))
- chore(deps): update dependency pylint to v2.13.0 ([5fa403b](#))
- chore(deps): update dependency mypy to v0.942 ([8ba0f8c](#))
- chore(deps): update dependency pytest-console-scripts to v1.3.1 ([da392e3](#))
- chore(deps): update dependency pytest to v7.1.1 ([e31f2ef](#))
- chore(deps): update typing dependencies ([21e7c37](#))
- chore(deps): update dependency mypy to v0.941 ([3a9d4f1](#))
- chore(deps): update dependency pytest to v7.1.0 ([27c7e33](#))
- chore(deps): update dependency types-requests to v2.27.12 ([8cd668e](#))
- chore(deps): update dependency mypy to v0.940 ([dd11084](#))
- chore(deps): update dependency types-setuptools to v57.4.10 ([b37fc41](#))
- chore(deps): update dependency pytest to v7 ([ae8d70d](#))
- chore(deps): update actions/upload-artifact action to v3 ([18a0eae](#))
- chore(deps): update actions/stale action to v5 ([d841185](#))

- chore(deps): update actions/setup-python action to v3 ([7f845f7](#))
- chore(deps): update pre-commit hook alessandrojcm/commitlint-pre-commit-hook to v8 ([5440780](#))
- chore(deps): update black to v22 ([3f84f1b](#))
- chore(deps): update dependency sphinx to v4.4.0 ([425d161](#))
- chore(deps): update actions/checkout action to v3 ([7333cbb](#))
- chore(deps): update dependency pytest-console-scripts to v1.3 ([9c202dd](#))
- chore(deps): update dependency mypy to v0.931 ([33646c1](#))
- chore(deps): update typing dependencies ([37a7c40](#))
- chore(deps): update dependency requests to v2.27.1 ([95dad55](#))

## Documentation

- docs: fix typo and incorrect style ([2828b10](#))
- docs: add pipeline test report summary support ([d78afb3](#))
- docs(chore): include docs .js files in sdist ([3010b40](#))

## Feature

- feat(object): add pipeline test report summary support ([a97e0cf](#))

## Style

- style: reformat for black v22 ([93d4403](#))

## Unknown

- Merge pull request #1947 from python-gitlab/renovate/typing-dependencies  
chore(deps): update dependency types-requests to v2.27.15 ([59ae16c](#))
- Merge pull request #1895 from python-gitlab/jlvillal/rate-limit  
fix: support RateLimit-Reset header ([114958e](#))
- Merge pull request #1905 from derekschrock/docs-static  
docs(chore): Include docs .js files in sdist ([363bc87](#))
- Merge pull request #1917 from python-gitlab/renovate/major-black  
chore(deps): update black to v22 (major) ([a4e76eb](#))
- Merge pull request #1919 from python-gitlab/renovate/alessandrojcm-commitlint-pre-commit-hook-8.x  
chore(deps): update pre-commit hook alessandrojcm/commitlint-pre-commit-hook to v8 ([71ebee4](#))
- Merge pull request #1915 from kinbald/test-report-summary  
feat: add support for test report summary ([7966584](#))

### 7.9.21 v3.2.0 (2022-02-28)

#### Chore

- chore: create a custom `warnings.warn` wrapper

Create a custom `warnings.warn` wrapper that will walk the stack trace to find the first frame outside of the `gitlab/` path to print the warning against. This will make it easier for users to find where in their code the error is generated from ([6ca9aa2](#))

- chore: correct type-hints for `per_page` attribute

There are occasions where a GitLab `list()` call does not return the `x-per-page` header. For example the listing of custom attributes.

Update the type-hints to reflect that. ([e825653](#))

- chore: require kwargs for `utils.copy_dict()`

The non-keyword arguments were a tiny bit confusing as the destination was first and the source was second.

Change the order and require key-word only arguments to ensure we don't silently break anyone. ([7cf35b2](#))

- chore: create new `ArrayAttribute` class

Create a new `ArrayAttribute` class. This is to indicate types which are sent to the GitLab server as arrays <https://docs.gitlab.com/ee/api/#array>

At this stage it is identical to the `CommaSeparatedListAttribute` class but will be used later to support the array types sent to GitLab.

This is the second step in a series of steps of our goal to add full support for the GitLab API data types[1]:

- array
- hash
- array of hashes

Step one was: commit [5127b1594c00c7364e9af15e42d2e2f2d909449b](#)

[1] <https://docs.gitlab.com/ee/api/#encoding-api-parameters-of-array-and-hash-types>

Related: [#1698](#) ([a57334f](#))

- chore(ci): do not run release workflow in forks ([2b6edb9](#))

#### Documentation

- docs: enable gitter chat directly in docs ([bd1ecdd](#))
- docs: add delete methods for runners and project artifacts ([5e711fd](#))
- docs: add `retry_transient` infos

Co-authored-by: Nejc Habjan <[hab.nejc@gmail.com](mailto:hab.nejc@gmail.com)> ([bb1f054](#))

- docs: add transient errors retry info ([b7a1266](#))
- docs(artifacts): deprecate `artifacts()` and `artifact()` methods ([64d01ef](#))
- docs: revert “chore: add temporary banner for v3” (#1864)

This reverts commit [a349793307e3a975bb51f864b48e5e9825f70182](#).

Co-authored-by: Wadim Klincov <[wadim.klincov@siemens.com](mailto:wadim.klincov@siemens.com)> ([7a13b9b](#))

## Feature

- feat(merge\_request\_approvals): add support for deleting MR approval rules ([85a734f](#))
- feat(artifacts): add support for project artifacts delete API ([c01c034](#))
- feat(mixins): allow deleting resources without IDs ([0717517](#))
- feat(objects): add a complete artifacts manager ([c8c2fa7](#))

## Fix

- fix: support RateLimit-Reset header

Some endpoints are not returning the `Retry-After` header when rate-limiting occurs. In those cases use the `RateLimit-Reset [1]` header, if available.

Closes: [#1889](#)

[1] [https://docs.gitlab.com/ee/user/admin\\_area/settings/user\\_and\\_ip\\_rate\\_limits.html#response-headers](https://docs.gitlab.com/ee/user/admin_area/settings/user_and_ip_rate_limits.html#response-headers) ([4060146](#))

- fix(services): use slug for id\_attr instead of custom methods ([e30f39d](#))
- fix: remove custom `delete` method for labels

The usage of deleting was incorrect according to the current API. Remove custom `delete()` method as not needed.

Add tests to show it works with labels needing to be encoded.

Also enable the `test_group_labels()` test function. Previously it was disabled.

Add ability to do a `get()` for group labels.

Closes: [#1867](#) ([0841a2a](#))

## Style

- style(objects): add spacing to docstrings ([700d25d](#))

## Test

- test(unit): clean up MR approvals fixtures ([0eb4f7f](#))
- test(runners): add test for deleting runners by auth token ([14b88a1](#))
- test(functional): fix GitLab configuration to support pagination

When pagination occurs python-gitlab uses the URL provided by the GitLab server to use for the next request.

We had previously set the GitLab server configuraiton to say its URL was `http://gitlab.test` which is not in DNS. Set the hostname in the URL to `http://127.0.0.1:8080` which is the correct URL for the GitLab server to be accessed while doing functional tests.

Closes: [#1877](#) ([5b7d00d](#))

- test(services): add functional tests for services ([2fea2e6](#))
- test(objects): add tests for project artifacts ([8ce0336](#))

## Unknown

- Merge pull request #1882 from python-gitlab/jlvillal/custom\_warn  
chore: create a custom warnings.warn wrapper ([5beda3b](#))
- Merge pull request #1881 from python-gitlab/jlvillal/easy2  
test(functional): fix GitLab configuration to support pagination ([4cb7d92](#))
- Merge pull request #1880 from python-gitlab/jlvillal/easy  
chore: correct type-hints for per\_page attribute ([5e19694](#))
- Merge pull request #1876 from emirot/patch-1  
docs: add transient errors retry info ([9897c98](#))
- Merge pull request #1871 from python-gitlab/jlvillal/copy\_dict  
chore: require kwargs for utils.copy\_dict() ([2adf31d](#))
- Merge pull request #1868 from python-gitlab/jlvillal/delete\_label  
fix: remove custom delete method for labels ([0ab0fc1](#))
- Merge pull request #1866 from python-gitlab/jlvillal/arrays\_2  
chore: create new ArrayAttribute class ([7646360](#))

## 7.9.22 v3.1.1 (2022-01-28)

### Chore

- chore: use dataclass for RequiredOptional ([30117a3](#))
- chore: remove redundant list comprehension ([271cfd3](#))
- chore: consistently use open() encoding and file descriptor ([dc32d54](#))
- chore: don't explicitly pass args to super() ([618267c](#))
- chore: always use context manager for file IO ([e8031f4](#))
- chore: remove old-style classes ([ae2a015](#))
- chore: rename types.ListAttribute to types.CommaSeparatedListAttribute

This name more accurately describes what the type is. Also this is the first step in a series of steps of our goal to add full support for the GitLab API data types[1]:

- array
- hash
- array of hashes

[1] <https://docs.gitlab.com/ee/api/#encoding-api-parameters-of-array-and-hash-types> ([5127b15](#))

- chore: rename gitlab/\_\_version\_\_.py -> gitlab/\_version.py

It is confusing to have a `gitlab/__version__.py` because we also create a variable `gitlab.__version__` which can conflict with `gitlab/_version.py`.

For example in `gitlab/const.py` we have to know that `gitlab.__version__` is a module and not the variable due to the ordering of imports. But in most other usage `gitlab.__version__` is a version string.

To reduce confusion make the name of the version file `gitlab/_version.py`. ([b981ce7](#))

- chore: create return type-hints for `get_id()` & `encoded_id`

Create return type-hints for `RESTObject.get_id()` and `RESTObject.encoded_id`. Previously was saying they return `Any`. Be more precise in saying they can return either: `None`, `str`, or `int`. ([0c3a1d1](#))

- chore(tests): use method `projects.transfer()`

When doing the functional tests use the new function `projects.transfer` instead of the deprecated function `projects.transfer_project()` ([e5af2a7](#))

## Documentation

- docs: enhance release docs for `CI_JOB_TOKEN` usage ([5d973de](#))
- docs(changelog): add missing changelog items ([01755fb](#))

## Fix

- fix(cli): make ‘per\_page’ and ‘page’ type explicit ([d493a5e](#))
- fix(cli): make ‘timeout’ type explicit ([bbb7df5](#))
- fix(cli): allow custom methods in managers ([8dfed0c](#))
- fix(objects): make resource access tokens and repos available in CLI ([e0a3a41](#))

## Style

- style: use f-strings where applicable ([cfed622](#))
- style: use literals to declare data structures ([019a40f](#))

## Test

- test: add a meta test to make sure that `v4/objects/` files are imported

Add a test to make sure that all of the `gitlab/v4/objects/` files are imported in `gitlab/v4/objects/__init__.py` ([9c8c804](#))

- test: convert usage of `match_querystring` to `match`

In the `responses` library the usage of `match_querystring` is deprecated. Convert to using `match` ([d16e41b](#))

- test: remove usage of `httpmock` library

Convert all usage of the `httpmock` library to using the `responses` library. ([5254f19](#))

- test: use ‘responses’ in `test_mixins_methods.py`

Convert from `httpmock` to `responses` in `test_mixins_methods.py`

This leaves only one file left to convert ([208da04](#))

## Unknown

- Merge pull request #1862 from thomasgl-orange/cli-fix-timeout  
fix(cli): make 'timeout', 'per\_page' and 'page' type explicit ([3fb4486](#))
- Merge pull request #1858 from python-gitlab/jlvillal/attribute\_rename  
chore: rename `types.ListAttribute` to `types.CommaSeparatedListAttribute` ([39e7435](#))
- Merge pull request #1848 from python-gitlab/jlvillal/objects\_imported  
test: add a meta test to make sure that v4/objects/ files are imported ([07539c9](#))
- Merge pull request #1854 from MRigal/docs/small-releases-additions  
Enhance releases API docs for `CI_JOB_TOKEN` usage ([ff04900](#))
- Merge pull request #1845 from python-gitlab/jlvillal/rm\_httmock  
Remove usage of `httmock` and clean up deprecations ([ff4b1cc](#))
- Merge pull request #1838 from python-gitlab/jlvillal/version\_mv  
chore: rename `gitlab/__version__.py` to `gitlab/_version.py` ([8af403c](#))
- Merge pull request #1843 from python-gitlab/jlvillal/rm\_httmock  
test: use 'responses' in `test_mixins_methods.py` ([fe14dd5](#))
- Merge pull request #1841 from python-gitlab/jlvillal/get\_id  
chore: create return type-hints for `get_id()` & `encoded_id` ([1ac982a](#))
- Merge pull request #1840 from python-gitlab/docs/missing-changelog-items  
docs(changelog): add missing changelog items ([a1dbe86](#))
- Merge pull request #1839 from python-gitlab/jlvillal/catch\_warnings  
chore(tests): use method `projects.transfer()` ([48b06a9](#))

## 7.9.23 v3.1.0 (2022-01-14)

### Chore

- chore(groups): use `encoded_id` for group path ([868f243](#))
  - chore(objects): use `self.encoded_id` where applicable
- Updated a few remaining usages of `self.id` to use `self.encoded_id` as for the most part we shouldn't be using `self.id`
- There are now only a few (4 lines of code) remaining uses of `self.id`, most of which seem that they should stay that way. ([75758bf](#))
- chore(objects): use `self.encoded_id` where could be a string
- Updated a few remaining usages of `self.id` to use `self.encoded_id` where it could be a string value. ([c3c3a91](#))
- chore(projects): fix typing for transfer method
- Co-authored-by: John Villalovos <[john@sodarock.com](mailto:john@sodarock.com)> ([0788fe6](#))
- chore: ignore intermediate coverage artifacts ([110ae91](#))
  - chore: replace usage of `utils._url_encode()` with `utils.EncodedId()`



`utils.EncodedId()` has basically the same functionality of using `utils.url_encode()`. So remove `utils.url_encode()` as we don't need it. ([b07eece](#))

- chore: add `EncodedId` string class to use to hold URL-encoded paths

Add `EncodedId` string class. This class returns a URL-encoded string but ensures it will only URL-encode it once even if recursively called.

Also added some functional tests of 'lazy' objects to make sure they work. ([a2e7c38](#))

- chore: add `pprint()` and `pformat()` methods to `RESTObject`

This is useful in debugging and testing. As can easily print out the values from an instance in a more human-readable form. ([d69ba04](#))

- chore: add logging to `tests/functional/conftest.py`

I have found trying to debug issues in the functional tests can be difficult. Especially when trying to figure out failures in the CI running on Github.

Add logging to `tests/functional/conftest.py` to have a better understanding of what is happening during a test run which is useful when trying to troubleshoot issues in the CI. ([a1ac9ae](#))

- chore(docs): use admonitions consistently ([55c67d1](#))
- chore: fix functional test failure if config present

Previously `c8256a5933d745f70c7eea0a7d6230b51bac0fbc` was done to fix this but it missed two other failures. ([c9ed3dd](#))

- chore: fix missing comma

There was a missing comma which meant the strings were concatenated instead of being two separate strings. ([7c59fac](#))

- chore(dist): add docs \*.md files to `sdist`

`build_sphinx` to fail due to `setup.cfg` `warning-is-error` ([d9457d8](#))

- chore: add a stale workflow

Use the stale action to close issues and pull-requests with no activity.

Issues: It will mark them as stale after 60 days and then close them once they have been stale for 15 days.

Pull-Requests: It will mark pull-requests as stale after 90 days and then close them once they have been stale for 15 days.

<https://github.com/actions/stale>

Closes: #1649 ([2c036a9](#))

- chore: add temporary banner for v3 ([a349793](#))

## Ci

- ci: don't fail CI if unable to upload the code coverage data

If a CI job can't upload coverage results to `codecov.com` it causes the CI to fail and code can't be merged. ([d5b3744](#))

## Documentation

- docs: update project access token API reference link ([73ae955](#))
- docs(cli): make examples more easily navigable by generating TOC ([f33c523](#))

## Feature

- feat: add support for Groups API method `transfer()` ([0007006](#))
- feat(api): add `project.transfer()` and deprecate `transfer_project()` ([259668a](#))
- feat(api): return result from `SaveMixin.save()`

Return the new object data when calling `SaveMixin.save()`.

Also remove check for `None` value when calling `self.manager.update()` as that method only returns a dictionary.

Closes: [#1081](#) ([e6258a4](#))

- feat: add support for Group Access Token API

See [https://docs.gitlab.com/ee/api/group\\_access\\_tokens.html](https://docs.gitlab.com/ee/api/group_access_tokens.html) ([c01b7c4](#))

## Fix

- fix(cli): add missing list filters for environments ([6f64d40](#))
- fix: use url-encoded ID in all paths

Make sure all usage of the ID in the URL path is encoded. Normally it isn't an issue as most IDs are integers or strings which don't contain a slash (`'/'`). But when the ID is a string with a slash character it will break things.

Add a test case that shows this fixes wikis issue with subpages which use the slash character.

Closes: [#1079](#) ([12435d7](#))

- fix(members): use new `*All` objects for `*AllManager` managers

Change it so that:

`GroupMemberAllManager` uses `GroupMemberAll` object `ProjectMemberAllManager` uses `ProjectMemberAll` object

Create `GroupMemberAll` and `ProjectMemberAll` objects that do not support any `Mixin` type methods. Previously we were using `GroupMember` and `ProjectMember` which support the `save()` and `delete()` methods but those methods will not work with objects retrieved using the `/members/all/` API calls.

`list()` API calls: [1] `GET /groups/:id/members/all` `GET /projects/:id/members/all`

`get()` API calls: [2] `GET /groups/:id/members/all/:user_id` `GET /projects/:id/members/all/:user_id`

Closes: [#1825](#) Closes: [#848](#)

[1] <https://docs.gitlab.com/ee/api/members.html#list-all-members-of-a-group-or-project-including-inherited-and-invited-members> [2] <https://docs.gitlab.com/ee/api/members.html#get-a-member-of-a-group-or-project-including-inherited-and-invited-members> ([755e0a3](#))

- fix(api): services: add missing `lazy` parameter

Commit [8da0b758c589f608a6ae4eeb74b3f306609ba36d](#) added the `lazy` parameter to the `services.get()` method but missed then using the `lazy` parameter when it called `super(...).get(...)`

Closes: [#1828](#) ([888f332](#))

- fix: broken URL for FAQ about attribute-error-list

The URL was missing a ‘v’ before the version number and thus the page did not exist.

Previously the URL for python-gitlab 3.0.0 was: <https://python-gitlab.readthedocs.io/en/3.0.0/faq.html#attribute-error-list>

Which does not exist.

Change it to: <https://python-gitlab.readthedocs.io/en/v3.0.0/faq.html#attribute-error-list> add the ‘v’  
 \_\_\_\_\_^ (1863f30)

- fix: remove custom URL encoding

We were using `str.replace()` calls to take care of URL encoding issues.

Switch them to use our `utils._url_encode()` function which itself uses `urllib.parse.quote()`

Closes: #1356 (3d49e5e)

- fix: remove default arguments for `mergerequests.merge()`

The arguments `should_remove_source_branch` and `merge_when_pipeline_succeeds` are optional arguments. We should not be setting any default value for them.

[https://docs.gitlab.com/ee/api/merge\\_requests.html#accept-mr](https://docs.gitlab.com/ee/api/merge_requests.html#accept-mr)

Closes: #1750 (8e589c4)

- fix(cli): url-encode path components of the URL

In the CLI we need to make sure the components put into the path portion of the URL are url-encoded. Otherwise they will be interpreted as part of the path. For example can specify the project ID as a path, but in the URL it must be url-encoded or it doesn’t work.

Also stop adding the components of the path as query parameters in the URL.

Closes: #783 Closes: #1498 (ac1c619)

- fix: change to `http_list` for some `ProjectCommit` methods

Fix the type-hints and use `http_list()` for the `ProjectCommits` methods:

- `diff()`
- `merge_requests()`
- `refs()`

This will enable using the pagination support we have for lists.

Closes: #1805 Closes: #1231 (497e860)

## Test

- test(groups): enable group transfer tests (57bb67a)

## Unknown

- Merge pull request #1836 from python-gitlab/jlvillal/id\_to\_encodedid chore(objects): use `self.encoded_id` where applicable ([2c62d91](#))
- Merge pull request #1835 from python-gitlab/jlvillal/id\_to\_encodedid chore(objects): use `self.encoded_id` where could be a string ([34110dd](#))
- Merge pull request #1832 from python-gitlab/jlvillal/return\_save feat(api): return result from `SaveMixin.save()` ([27e0742](#))
- Merge pull request #1834 from python-gitlab/jlvillal/cover\_no\_fail ci: don't fail CI if unable to upload the code coverage data ([da30753](#))
- Merge pull request #1831 from python-gitlab/chore/ignore-coverage chore: ignore intermediate coverage artifacts ([8b14ff0](#))
- Merge pull request #1819 from python-gitlab/jlvillal/encoded\_id fix: use url-encoded ID in all paths ([bc48840](#))
- Merge pull request #1827 from python-gitlab/jlvillal/all\_objects fix: members: use new `*All` objects for `*AllManager` managers ([58e5b25](#))
- Merge pull request #1829 from python-gitlab/jlvillal/lazy\_service fix(api): services: add missing `lazy` parameter ([824151c](#))
- Merge pull request #1823 from python-gitlab/jlvillal/fix\_url fix: broken URL for FAQ about attribute-error-list ([4a000b6](#))
- Merge pull request #1812 from python-gitlab/jlvillal/pprint chore: add `pprint()` and `pformat()` methods to `RESTObject` ([bdc19b1](#))
- Merge pull request #1786 from python-gitlab/jlvillal/logging test: add logging to `tests/functional/conftest.py` ([ac81272](#))
- Merge pull request #1816 from python-gitlab/jlvillal/remove\_replace fix: remove custom URL encoding ([24d2766](#))
- Merge pull request #1818 from python-gitlab/jlvillal/merge\_request\_merge\_defaults fix: remove default arguments for `mergerequests.merge()` ([0dba899](#))
- Merge pull request #1790 from python-gitlab/jlvillal/parent\_attrs fix(cli): url-encode path components of the URL ([22a1516](#))
- Merge pull request #1809 from python-gitlab/jlvillal/list\_api fix: change to `http_list` for some `ProjectCommit` methods ([d45b59e](#))
- Merge pull request #1813 from derekschrock/missing-dist chore(dist): add docs `*.md` files to `sdist` ([4861883](#))
- Merge pull request #1814 from python-gitlab/jlvillal/missing\_comma chore: fix missing comma ([fd523b3](#))
- Merge pull request #1789 from python-gitlab/jlvillal/stale

chore: add a stale workflow ([9896340](#))

- Merge pull request #1803 from python-gitlab/jlvillal/test\_1425

chore: add functional test of mergerequest.get() ([bc6c6e6](#))

## 7.9.24 v3.0.0 (2022-01-05)

### Breaking

- feat(cli): allow options from args and environment variables

**BREAKING-CHANGE:** The gitlab CLI will now accept CLI arguments and environment variables for its global options in addition to configuration file options. This may change behavior for some workflows such as running inside GitLab CI and with certain environment variables configured. ([ca58008](#))

- fix: stop encoding '.' to '%2E'

Forcing the encoding of '.' to '%2E' causes issues. It also goes against the RFC: <https://datatracker.ietf.org/doc/html/rfc3986.html#section-2.3>

From the RFC: For consistency, percent-encoded octets in the ranges of ALPHA (%41-%5A and %61-%7A), DIGIT (%30-%39), hyphen (%2D), period (%2E), underscore (%5F), or tilde (%7E) should not be created by URI producers...

Closes #1006 Related #1356 Related #1561

**BREAKING CHANGE:** stop encoding '.' to '%2E'. This could potentially be a breaking change for users who have incorrectly configured GitLab servers which don't handle period '.' characters correctly. ([702e41d](#))

- feat(cli): do not require config file to run CLI

**BREAKING CHANGE:** A config file is no longer needed to run the CLI. python-gitlab will default to <https://gitlab.com> with no authentication if there is no config file provided. python-gitlab will now also only look for configuration in the provided PYTHON\_GITLAB\_CFG path, instead of merging it with user- and system-wide config files. If the environment variable is defined and the file cannot be opened, python-gitlab will now explicitly fail. ([92a893b](#))

- feat: remove support for Python 3.6, require 3.7 or higher

Python 3.6 is End-of-Life (EOL) as of 2021-12 as stated in <https://www.python.org/dev/peps/pep-0494/>

By dropping support for Python 3.6 and requiring Python 3.7 or higher it allows python-gitlab to take advantage of new features in Python 3.7, which are documented at: <https://docs.python.org/3/whatsnew/3.7.html>

Some of these new features that may be useful to python-gitlab are:

- PEP 563, postponed evaluation of type annotations.
- dataclasses: PEP 557 – Data Classes
- importlib.resources
- PEP 562, customization of access to module attributes.
- PEP 560, core support for typing module and generic types.
- PEP 565, improved DeprecationWarning handling

**BREAKING CHANGE:** As of python-gitlab 3.0.0, Python 3.6 is no longer supported. Python 3.7 or higher is required. ([414009d](#))

- chore: rename master branch to main

**BREAKING CHANGE:** As of python-gitlab 3.0.0, the default branch for development has changed from master to main. ([545f8ed](#))

- refactor(objects): remove deprecated branch protect methods

BREAKING CHANGE: remove deprecated branch protect methods in favor of the more complete protected branches API. ([9656a16](#))

- fix(api): replace deprecated attribute in delete\_in\_bulk() (#1536)

BREAKING CHANGE: The deprecated `name_regex` attribute has been removed in favor of `name_regex_delete`. (see <https://gitlab.com/gitlab-org/gitlab/-/commit/ce99813cf54>) ([c59fbdb](#))

- fix(objects): rename confusing `to_project_id` argument

BREAKING CHANGE: rename confusing `to_project_id` argument in `transfer_project` to `project_id` (`--project-id` in CLI). This is used for the source project, not for the target namespace. ([ce4bc0d](#))

- refactor(objects): remove deprecated constants defined in objects

BREAKING CHANGE: remove deprecated constants defined in `gitlab.v4.objects`, and use only `gitlab.const` module ([3f320af](#))

- refactor(objects): remove deprecated tag release API

BREAKING CHANGE: remove deprecated tag release API. This was removed in GitLab 14.0 ([2b8a94a](#))

- refactor(objects): remove deprecated `project.issuesstatistics`

BREAKING CHANGE: remove deprecated `project.issuesstatistics` in favor of `project.issues_statistics` ([ca7777e](#))

- refactor(objects): remove deprecated `members.all()` method

BREAKING CHANGE: remove deprecated `members.all()` method in favor of `members_all.list()` ([4d7b848](#))

- refactor(objects): remove deprecated `pipelines()` method

BREAKING CHANGE: remove deprecated `pipelines()` methods in favor of `pipelines.list()` ([c4f5ec6](#))

- feat: default to `gitlab.com` if no URL given

BREAKING CHANGE: python-gitlab will now default to `gitlab.com` if no URL is given ([8236281](#))

- fix!: raise error if there is a 301/302 redirection

Before we raised an error if there was a 301, 302 redirect but only from an http URL to an https URL. But we didn't raise an error for any other redirects.

This caused two problems:

1. PUT requests that are redirected get changed to GET requests which don't perform the desired action but raise no error. This is because the GET response succeeds but since it wasn't a PUT it doesn't update. See issue: <https://github.com/python-gitlab/python-gitlab/issues/1432>
2. POST requests that are redirected also got changed to GET requests. They also caused hard to debug tracebacks for the user. See issue: <https://github.com/python-gitlab/python-gitlab/issues/1477>

Correct this by always raising a `RedirectError` exception and improve the exception message to let them know what was redirected.

Closes: #1485 Closes: #1432 Closes: #1477 ([d56a434](#))

## Chore

- chore: fix typo in MR documentation ([2254222](#))
- chore: add functional test of `mergerequest.get()`

Add a functional test of `test mergerequest.get()` and `mergerequest.get(..., lazy=True)`

Closes: #1425 ([a92b55b](#))

- chore(deps): update dependency `argcomplete` to v2 ([c6d7e9a](#))
- chore(deps): update dependency `requests` to v2.27.0 ([f8c3d00](#))
- chore: add test case to show branch name with period works

Add a test case to show that a branch name with a period can be fetched with a `get()`

Closes: #1715 ([ea97d7a](#))

- chore(deps): update typing dependencies ([1f95613](#))
- chore(deps): update dependency `mypy` to v0.930 ([ccf8190](#))
- chore(deps): upgrade `mypy` pre-commit hook ([e19e4d7](#))
- chore: fix functional test failure if config present

Fix functional test failure if config present and configured with token.

Closes: #1791 ([c8256a5](#))

- chore: ensure `reset_gitlab()` succeeds

Ensure `reset_gitlab()` succeeds by waiting to make sure everything has been deleted as expected. If the timeout is exceeded fail the test.

Not using `wait_for_sidekiq` as it didn't work. During testing I didn't see any `sidekiq` processes as being busy even though not everything was deleted. ([0aa0b27](#))

- chore: skip a functional test if not using `>= py3.9`

One of the tests requires Python 3.9 or higher to run. Mark the test to be skipped if running Python less than 3.9. ([ac9b595](#))

- chore: update version in `docker-compose.yml`

When running with `docker-compose` on Ubuntu 20.04 I got the error:

```
$ docker-compose up ERROR: The Compose file './docker-compose.yml' is invalid because: networks.gitlab-network value Additional properties are not allowed ('name' was unexpected)
```

Changing the version in the `docker-compose.yml` file from '3' to '3.5' resolved the issue. ([79321aa](#))

- chore: add and document optional parameters for `get MR`

Add and document (some of the) optional parameters that can be done for a `project.merge_requests.get()`

Closes #1775 ([bfa3dbe](#))

- chore: generate artifacts for the docs build in the CI

When building the docs store the created documentation as an artifact so that it can be viewed.

This will create a `html-docs.zip` file which can be downloaded containing the contents of the `build/sphinx/html/` directory. It can be downloaded, extracted, and then viewed. This can be useful in reviewing changes to the documentation.

See <https://github.com/actions/upload-artifact> for more information on how this works. ([85b43ae](#))

- chore(deps): update pre-commit hook pycqa/flake8 to v4 ([98a5592](#))
- chore(deps): update pre-commit hook psf/black to v21 ([b86e819](#))
- chore(deps): update pre-commit hook pycqa/isort to v5.10.1 ([8ac4f4a](#))
- chore: remove ‘# type: ignore’ for new mypy version

mypy 0.920 now understands the type of ‘http.client.HTTPConnection.debuglevel’ so we remove the ‘type: ignore’ comment to make mypy pass ([34a5f22](#))

- chore(deps): update dependency mypy to v0.920 ([a519b2f](#))
- chore(deps): update pre-commit hook alessandrojcm/commitlint-pre-commit-hook to v6 ([fb9110b](#))
- chore(ci): enable renovate for pre-commit ([1ac4329](#))
- chore: fix unit test if config file exists locally

Closes #1764 ([c80b3b7](#))

- chore(deps): update dependency sphinx to v4.3.2 ([2210e56](#))
- chore: add .env as a file that search tools should not ignore

The .env file was not set as a file that should not be ignored by search tools. We want to have the search tools search any .env files. ([c9318a9](#))

- chore(deps): update dependency types-requests to v2.26.2 ([ac7e329](#))
- chore: add Python 3.11 testing

Add a unit test for Python 3.11. This will use the latest version of Python 3.11 that is available from <https://github.com/actions/python-versions/>

At this time it is 3.11.0-alpha.2 but will move forward over time until the final 3.11 release and updates. So 3.11.0, 3.11.1, ... will be matched. ([b5ec192](#))

- chore(api): temporarily remove topic delete endpoint

It is not yet available upstream. ([e3035a7](#))

- chore: fix renovate setup for gitlab docker image ([49af15b](#))
- chore: add get() methods for GetWithoutIdMixin based classes

Add the get() methods for the GetWithoutIdMixin based classes.

Update the tests/meta/test\_ensure\_type\_hints.py tests to check to ensure that the get methods are defined with the correct return type. ([d27c50a](#))

- chore: github workflow: cancel prior running jobs on new push

If new new push is done to a pull-request, then cancel any already running github workflow jobs in order to conserve resources. ([fd81569](#))

- chore: fix pylint error “expression-not-assigned”

Fix pylint error “expression-not-assigned” and remove check from the disabled list.

And I personally think it is much more readable now and is less lines of code. ([a90eb23](#))

- chore: add running unit tests on windows/macOS

Add running the unit tests on windows-latest and macOS-latest with Python 3.10. ([ad5d60c](#))

- chore: set pre-commit mypy args to empty list



<https://github.com/pre-commit/mirrors-mypy/blob/master/.pre-commit-hooks.yaml>

Sets some default args which seem to be interfering with things. Plus we set all of our args in the `pyproject.toml` file. ([b67a6ad](#))

- chore: run pre-commit on changes to the config file

If `.pre-commit-config.yaml` or `.github/workflows/pre_commit.yml` are updated then run pre-commit. ([5f10b3b](#))

- chore: add initial pylint check

Initial pylint check is added. A LONG list of disabled checks is also added. In the future we should work through the list and resolve the errors or disable them on a more granular level. ([041091f](#))

- chore: enable ‘warn\_redundant\_casts’ for mypy

Enable ‘warn\_redundant\_casts’ for mypy and resolve one issue. ([f40e9b3](#))

- chore: enable subset of the ‘mypy –strict’ options that work

Enable the subset of the ‘mypy –strict’ options that work with no changes to the code. ([a86d049](#))

- chore(deps): update dependency black to v21.12b0 ([ab841b8](#))
- chore: use constants from gitlab.const module

Have code use constants from the `gitlab.const` module instead of from the top-level `gitlab` module. ([6b8067e](#))

- chore: attempt to be more informative for missing attributes

A commonly reported issue from users on Gitter is that they get an `AttributeError` for an attribute that should be present. This is often caused due to the fact that they used the `list()` method to retrieve the object and objects retrieved this way often only have a subset of the full data.

Add more details in the `AttributeError` message that explains the situation to users. This will hopefully allow them to resolve the issue.

Update the FAQ in the docs to add a section discussing the issue.

Closes #1138 ([1839c9e](#))

- chore(docs): link to main, not master ([af0cb4d](#))
- chore(docs): use builtin autodoc hints ([5e9c943](#))
- chore(docs): load autodoc-typehints module ([bd366ab](#))
- chore(tests): apply review suggestions ([381c748](#))
- chore(deps): update dependency sphinx to v4.3.1 ([93a3893](#))
- chore: remove pytest-console-scripts specific config

Remove the `pytest-console-scripts` specific config from the global ‘[pytest]’ config section.

Use the command line option `--script-launch-mode=subprocess`

Closes #1713 ([e80dcb1](#))

- chore(deps): update typing dependencies ([8d4c953](#))
- chore: remove duplicate/no-op tests from `meta/test_ensure_type_hints`

Before we were generating 725 tests for the `meta/test_ensure_type_hints.py` tests. Which isn’t a huge concern as it was fairly fast. But when we had a failure we would usually get two failures for each problem as the same test was being run multiple times.

Changed it so that:

1. Don't add tests that are not for \*Manager classes
2. Use a set so that we don't have duplicate tests.

After doing that our generated test count in `meta/test_ensure_type_hints.py` went from 725 to 178 tests.

Additionally removed the parsing of `pyproject.toml` to generate files to ignore as we have finished adding type-hints to all files in `gitlab/v4/objects/`. This also means we no longer use the `toml` library so remove installation of `types-toml`.

To determine the test count the following command was run: `$ tox -e py39 -- -k test_ensure_type_hints (a2f59f4)`

- chore(deps): update dependency types-setuptools to v57.4.3 (ec2c68b)
- chore(deps): update dependency black to v21 (5bca87c)
- chore: have renovate upgrade black version (#1700)

renovate is not upgrading the `black` package. There is an open issue<sup>[1]</sup> about this.

Also change `.commitlintrc.json` to allow 200 character footer lines in the commit message. Otherwise would be forced to split the URL across multiple lines making it un-clickable :(

Use suggested work-arounds from: <https://github.com/renovatebot/renovate/issues/7167#issuecomment-904106838>  
<https://github.com/scop/bash-completion/blob/e7497f6ee8232065ec11450a52a1f244f345e2c6/renovate.json#L34-L38>

[1] <https://github.com/renovatebot/renovate/issues/7167> (21228cd)

- chore: add type-hints to `gitlab/v4/objects/files.py` (0c22bd9)
- chore: add type-hints to `gitlab/v4/objects/labels.py` (d04e557)
- chore: add type-hints to `gitlab/v4/objects/sidekiq.py` (a91a303)
- chore: add type-hints to `gitlab/v4/objects/services.py` (8da0b75)
- chore: add type-hints to `gitlab/v4/objects/repositories.py` (00d7b20)
- chore: add type-hints to `gitlab/v4/objects/pipelines.py` (cb3ad6c)
- chore: add type-hints to `gitlab/v4/objects/milestones.py` (8b6078f)
- chore: add type-hints to `gitlab/v4/objects/jobs.py` (e8884f2)
- chore: add type-hints to `gitlab/v4/objects/issues.py` (93e39a2)
- chore: add type-hints to `gitlab/v4/objects/geo_nodes.py` (13243b7)
- chore: add type-hints to `gitlab/v4/objects/epics.py` (d4adf8d)
- chore: fix issue with adding type-hints to 'manager' attribute

When attempting to add type-hints to the the 'manager' attribute into a `RESTObject` derived class it would break things.

This was because our auto-manager creation code would automatically add the specified annotated manager to the 'manager' attribute. This breaks things.

Now check in our auto-manager creation if our attribute is called 'manager'. If so we ignore it. (9a451a8)

- chore: correct `test_groups.py` test

The test was checking twice if the same `group3` was not in the returned list. Should have been checking for `group3` and `group4`.

Also added a test that only skipped one group and checked that the group was not in the returned list and a non-skipped group was in the list. (9c878a4)

- chore: add type-hints to `gitlab/v4/objects/merge_request_approvals.py` (cf3a99a)

- chore: enable mypy for tests/meta/\* ([ba7707f](#))
- chore: check setup.py with mypy

Prior commit [06184daafd5010ba40bb39a0768540b7e98bd171](#) fixed the type-hints for setup.py. But missed removing 'setup' from the exclude list in pyproject.toml for mypy checks.

Remove 'setup' from the exclude list in pyproject.toml from mypy checks. ([77cb7a8](#))

- chore: ensure get() methods have correct type-hints

Fix classes which don't have correct 'get()' methods for classes derived from GetMixin.

Add a unit test which verifies that classes have the correct return type in their 'get()' method. ([46773a8](#))

- chore: create a 'tests/meta/' directory and put test\_mro.py in it

The 'test\_mro.py' file is not really a unit test but more of a 'meta' check on the validity of the code base. ([94feb8a](#))

- chore: add type-hints to setup.py and check with mypy ([06184da](#))
- chore: add type-hints to gitlab/v4/objects/snippets.py ([f256d4f](#))
- chore(deps): update dependency types-pyyaml to v6.0.1 ([a544cd5](#))
- chore(deps): update dependency sphinx to v4.3.0 ([57283fc](#))
- chore(deps): update dependency types-requests to v2.26.0 ([7528d84](#))
- chore(deps): update dependency isort to v5.10.1 ([2012975](#))
- chore(deps): update dependency types-requests to v2.25.12 ([205ad5f](#))
- chore: enforce type-hints on most files in gitlab/v4/objects/
  - Add type-hints to some of the files in gitlab/v4/objects/
  - Fix issues detected when adding type-hints
  - Changed mypy exclusion to explicitly list the 13 files that have not yet had type-hints added. ([7828ba2](#))
- chore: add type hints for gitlab/v4/objects/commits.py ([dc096a2](#))
- chore(ci): add workflow to lock old issues ([a7d64fe](#))
- chore: add type-hints to multiple files in gitlab/v4/objects/

Add and/or check type-hints for the following files `gitlab.v4.objects.access_requests` `gitlab.v4.objects.applications` `gitlab.v4.objects.broadcast_messages` `gitlab.v4.objects.deployments` `gitlab.v4.objects.keys` `gitlab.v4.objects.merge_trains` `gitlab.v4.objects.namespaces` `gitlab.v4.objects.pages` `gitlab.v4.objects.personal_access_tokens` `gitlab.v4.objects.project_access_tokens` `gitlab.v4.objects.tags` `gitlab.v4.objects.templates` `gitlab.v4.objects.triggers`

Add a 'get' method with the correct type for Managers derived from GetMixin. ([8b75a77](#))

- chore: add type-hints to gitlab/v4/objects/groups.py
- Add type-hints to gitlab/v4/objects/groups.py
- Have share() function update object attributes.
- Add 'get()' method so that type-checkers will understand that getting a group is of type Group. ([94dcb06](#))
- chore: add type-hints to gitlab/v4/objects/merge\_requests.py
- Add type-hints to gitlab/v4/objects/merge\_requests.py
- Add return value to cancel\_merge\_when\_pipeline\_succeeds() function as GitLab docs show it returns a value.
- Add return value to approve() function as GitLab docs show it returns a value.

- Add ‘get()’ method so that type-checkers will understand that getting a project merge request is of type ProjectMergeRequest. ([f9c0ad9](#))
- chore(deps): update dependency isort to v5.10.0 ([ae62468](#))
- chore(ci): wait for all coverage jobs before posting comment ([c7fdad4](#))
- chore(deps): update dependency types-pyyaml to v6 ([0b53c0a](#))
- chore(deps): update typing dependencies ([4170dbe](#))
- chore(deps): update dependency flake8 to v4 ([79785f0](#))
- chore(deps): update typing dependencies ([4eb8ec8](#))
- chore(deps): upgrade gitlab-ce to 14.3.2-ce.0 ([5a1678f](#))
- chore(objects): remove non-existing trigger ownership method ([8dc7f40](#))
- chore(deps): update dependency types-requests to v2.25.9 ([e3912ca](#))
- chore: fix type-check issue shown by new requests-types

types-requests==2.25.9 changed a type-hint. Update code to handle this change. ([0ee9aa4](#))

- chore(deps): update dependency sphinx to v4 ([73745f7](#))
- chore(deps): update python docker tag to v3.10 ([b3d6d91](#))
- chore: clean up install docs ([a5d8b7f](#))
- chore: attempt to fix flaky functional test

Add an additional check to attempt to solve the flakiness of the test\_merge\_request\_should\_remove\_source\_branch() test. ([487b9a8](#))

- chore: convert to using type-annotations for managers

Convert our manager usage to be done via type annotations.

Now to define a manager to be used in a RESTObject subclass can simply do: class ExampleClass(CRUDMixin, RESTObject): my\_manager: MyManager

Any type-annotation that annotates it to be of type \*Manager (with the exception of RESTManager) will cause the manager to be created on the object. ([d8de4dc](#))

- chore: add type-hints to gitlab/v4/objects/users.py

Adding type-hints to gitlab/v4/objects/users.py ([88988e3](#))

- chore: improve type-hinting for managers

The ‘managers’ are dynamically created. This unfortunately means that we don’t have any type-hints for them and so editors which understand type-hints won’t know that they are valid attributes.

- Add the type-hints for the managers we define.
- Add a unit test that makes sure that the type-hints and the ‘\_managers’ attribute are kept in sync with each other.
- Add unit test that makes sure specified managers in ‘\_managers’ have a name ending in ‘Managers’ to keep with current convention.
- Make RESTObject.\_managers always present with a default value of None.
- Fix a type-issue revealed now that mypy knows what the type is ([c9b5d3b](#))
- chore(deps): update dependency types-pyyaml to v5.4.10 ([bdb6cb9](#))

## Documentation

- docs: switch to Furo and refresh introduction pages ([ee6b024](#))
- docs: correct documentation for updating discussion note

Closes #1777 ([ee66f4a](#))

- docs: rename documentation files to match names of code files

Rename the merge request related documentation files to match the code files. This will make it easier to find the documentation quickly.

Rename: `docs/gl_objects/mrs.rst` -> `docs/gl_objects/merge_requests.rst`  
`docs/gl_objects/mr_approvals.rst` -> `docs/gl_objects/merge_request_approvals.rst` ([ee3f865](#))

- docs(project): remove redundant encoding parameter ([fed613f](#))
- docs: use annotations for return types ([79e785e](#))
- docs: update docs to use `gitlab.const` for constants

Update the docs to use `gitlab.const` to access constants. ([b3b0b5f](#))

- docs: only use type annotations for documentation ([b7dde0d](#))
- docs: add links to the GitLab API docs

Add links to the GitLab API docs for `merge_requests.py` as it contains code which spans two different API documentation pages. ([e3b5d27](#))

- docs: fix API delete key example ([b31bb05](#))
- docs(pipelines): document `take_ownership` method ([69461f6](#))
- docs(api): document the update method for project variables ([7992911](#))
- docs(api): clarify job token usage with `auth()`

See issue #1620 ([3f423ef](#))

- docs: fix a few typos

There are small typos in:

- `docs/gl_objects/deploy_tokens.rst`
- `gitlab/base.py`
- `gitlab/mixins.py`
- `gitlab/v4/objects/features.py`
- `gitlab/v4/objects/groups.py`
- `gitlab/v4/objects/packages.py`
- `gitlab/v4/objects/projects.py`
- `gitlab/v4/objects/sidekiq.py`
- `gitlab/v4/objects/todos.py`

Fixes:

- Should read `treatment` rather than `reatment`.
- Should read `transferred` rather than `transferred`.
- Should read `registered` rather than `registred`.

- Should read occurred rather than occured.
- Should read overridden rather than overriden.
- Should read marked rather than maked.
- Should read instantiate rather than instanciate.
- Should read function rather than fonction. ([7ea4ddc](#))
- docs: consolidate changelogs and remove v3 API docs ([90da8ba](#))
- docs: correct documented return type

repository\_archive() returns 'bytes' not 'str'

<https://docs.gitlab.com/ee/api/repositories.html#get-file-archive>

Fixes: [#1584](#) ([acabf63](#))

## Feature

- feat(docker): remove custom entrypoint from image

This is no longer needed as all of the configuration is handled by the CLI and can be passed as arguments. ([80754a1](#))

- feat(api): support file format for repository archive ([83dcabf](#))
- feat: add support for squash\_option in Projects

There is an optional squash\_option parameter which can be used when creating Projects and UserProjects.

Closes [#1744](#) ([a246ce8](#))

- feat(api): add support for Topics API ([e7559bf](#))
- feat: add delete on package\_file object ([124667b](#))
- feat: add support for projects.groups.list()

Add support for projects.groups.list() endpoint.

Closes [#1717](#) ([68ff595](#))

- feat(api): add support for epic notes

Added support for notes on group epics

Signed-off-by: Raimund Hook <[raimund.hook@exfo.com](mailto:raimund.hook@exfo.com)> ([7f4edb5](#))

- feat(api): add project milestone promotion

Adds promotion to Project Milestones

Signed-off-by: Raimund Hook <[raimund.hook@exfo.com](mailto:raimund.hook@exfo.com)> ([f068520](#))

- feat(api): add merge trains

Add support for merge trains ([fd73a73](#))

- feat(api): add merge request approval state

Add support for merge request approval state ([f41b093](#))

- feat(api): add project label promotion

Adds a mixin that allows the /promote endpoint to be called.

Signed-off-by: Raimund Hook <[raimund.hook@exfo.com](mailto:raimund.hook@exfo.com)> ([6d7c88a](#))

- feat(objects): support delete package files API ([4518046](#))
- feat(objects): list starred projects of a user ([47a5606](#))
- feat(build): officially support and test python 3.10 ([c042ddc](#))
- feat(objects): support Create and Revoke personal access token API ([e19314d](#))
- feat: allow global `retry_transient_errors` setup

`retry_transient_errors` can now be set through the Gitlab instance and global configuration

Documentation for API usage has been updated and missing tests have been added. ([3b1d3a4](#))

## Fix

- fix: handle situation where GitLab does not return values

If a query returns more than 10,000 records than the following values are NOT returned: `x.total_pages` `x.total`

Modify the code to allow no value to be set for these values. If there is not a value returned the functions will now return `None`.

Update unit test so no longer `xfail`

[https://docs.gitlab.com/ee/user/gitlab\\_com/index.html#pagination-response-headers](https://docs.gitlab.com/ee/user/gitlab_com/index.html#pagination-response-headers)

Closes [#1686](#) ([cb824a4](#))

- fix(build): do not include docs in wheel package ([68a97ce](#))
- fix(api): delete invalid ‘project-runner get’ command ([#1628](#))
- fix(api): delete ‘group-runner get’ and ‘group-runner delete’ commands

Co-authored-by: Léo GATELLIER <[git@leogatellier.fr](mailto:git@leogatellier.fr)> ([905781b](#))

- fix(build): do not package tests in wheel ([969dccc](#))

## Refactor

- refactor: deprecate accessing constants from top-level namespace

We are planning on adding enumerated constants into `gitlab/const.py`, but if we do that than they will end up being added to the top-level `gitlab` namespace. We really want to get users to start using `gitlab.const.` to access the constant values in the future.

Add the currently defined constants to a list that should not change. Use a module level `getattr` function so that we can deprecate access to the top-level constants.

Add a unit test which verifies we generate a warning when accessing the top-level constants. ([c0aa0e1](#))

- refactor: use new-style formatting for named placeholders ([c0d8810](#))
- refactor: use f-strings for string formatting ([7925c90](#))

## Test

- test: reproduce missing pagination headers in tests ([501f9a1](#))
- test: drop httmock dependency in test\_gitlab.py ([c764bee](#))
- test(api): fix current user mail count in newer gitlab ([af33aff](#))
- test(cli): improve basic CLI coverage ([6b892e3](#))
- test(build): add smoke tests for sdist & wheel package ([b8a47ba](#))

## Unknown

- Merge pull request #1804 from mlegner/patch-1
- chore: fix typo in MR documentation ([1582387](#))
- Merge pull request #1800 from python-gitlab/jlvillal/dot\_branch
- chore: add test case to show branch name with period works ([896a8c7](#))
- Merge pull request #1799 from python-gitlab/renovate/mypy-0.x
- chore(deps): update dependency mypy to v0.930 ([2323a7c](#))
- Merge pull request #1792 from python-gitlab/jlvillal/cli\_test
- chore: fix functional test failure if config present ([2ac2a68](#))
- Merge pull request #1773 from python-gitlab/jlvillal/pagination
- fix: handle situation where gitlab.com does not return values ([a3eafab](#))
- Merge pull request #1783 from python-gitlab/jlvillal/sidekiq
- chore: ensure reset\_gitlab() succeeds ([f26bf7d](#))
- Merge pull request #1782 from python-gitlab/jlvillal/repository\_func\_tests
- chore: skip a functional test if not using >= py3.9 ([d65ce36](#))
- Merge pull request #1781 from python-gitlab/jlvillal/docker\_compose
- chore: update version in docker-compose.yml ([171df89](#))
- Merge pull request #1774 from python-gitlab/jlvillal/doc\_artifacts
- chore: generate artifacts for the docs build in the CI ([3cb2352](#))
- Merge pull request #1776 from python-gitlab/jlvillal/rebase\_in\_progress
- Add some docs for getting the status of a merge\_request rebase ([e7d4d91](#))
- Merge pull request #1766 from python-gitlab/jlvillal/leave\_dot
- fix: stop encoding ‘.’ to ‘%2E’ ([eef8059](#))
- Merge pull request #1770 from python-gitlab/renovate/alessandrojcm-commitlint-pre-commit-hook-6.x
- chore(deps): update pre-commit hook alessandrojcm/commitlint-pre-commit-hook to v6 ([182ab92](#))
- Merge pull request #1753 from python-gitlab/renovate/mypy-0.x
- chore(deps): update dependency mypy to v0.920 ([5ea5392](#))
- Merge pull request #1765 from python-gitlab/jlvillal/unit\_test\_config
- chore: fix unit test if config file exists locally ([ccefef80](#))



- Merge pull request #1757 from python-gitlab/jlvillal/gitignore  
chore: add .env as a file that search tools should not ignore ([3ee061c](#))
- Merge pull request #1746 from python-gitlab/jlvillal/squash\_option  
feat: add support for `squash_option` in Projects ([7799cb9](#))
- Merge pull request #1743 from python-gitlab/feat/cli-without-config-file  
feat(cli): do not require config file to run CLI ([170a4d9](#))
- Merge pull request #1742 from python-gitlab/jlvillal/py311\_alpha  
chore: add Python 3.11 testing ([74d4e4b](#))
- Merge pull request #1710 from python-gitlab/jlvillal/get\_without\_id  
chore: add `get()` methods for `GetWithoutIdMixin` based classes ([ac5defa](#))
- Merge pull request #1733 from simonisateur/fix-package-file-delete  
feat: package file delete on package file object ([2f37ccb](#))
- Merge pull request #1736 from python-gitlab/jlvillal/workflow  
chore: github workflow: cancel prior running jobs on new push ([4945353](#))
- Merge pull request #1726 from python-gitlab/jlvillal/windows  
chore: add running unit tests on windows/macos ([83f36d6](#))
- Merge pull request #1738 from python-gitlab/jlvillal/pylint\_fixes  
chore: fix pylint error “expression-not-assigned” ([3679591](#))
- Merge pull request #1729 from python-gitlab/jlvillal/pylint  
chore: add initial pylint check ([3a7d6f6](#))
- Merge pull request #1727 from python-gitlab/jlvillal/mypy\_strict\_two\_steps  
Enable more strict mypy checking ([1c33080](#))
- Merge pull request #1709 from python-gitlab/docs/sphinx-annotations  
docs: only use type annotations for documentation ([2708f91](#))
- Merge pull request #1702 from python-gitlab/jlvillal/attribute\_help  
chore: attempt to be more informative for missing attributes ([387e59f](#))
- Merge pull request #1694 from python-gitlab/jlvillal/const\_explicit  
refactor: explicitly import `gitlab.const` values into top-level namespace ([e6582a3](#))
- Merge pull request #1721 from python-gitlab/test/cli-coverage  
test(cli): improve basic CLI coverage ([09a973e](#))
- Merge pull request #1714 from python-gitlab/jlvillal/pytest\_script\_launch\_mode  
chore: remove `pytest-console-scripts` specific config ([1badfeb](#))
- Merge pull request #1712 from StingRayZA/Epicnotes  
feat(api): add support for epic notes ([70b9870](#))
- Merge pull request #1718 from python-gitlab/jlvillal/project\_groups  
feat: add support for `projects.groups.list()` ([64f2360](#))

- Merge pull request #1707 from python-gitlab/jlvillal/reduce\_meta\_tests  
chore: remove duplicate/no-op tests from meta/test\_ensure\_type\_hints (3225f2c)
- Merge pull request #1695 from python-gitlab/jlvillal/mypy\_epics  
chore: add type-hints to remaining gitlab/v4/objects/\*.py files (7ba5995)
- Merge pull request #1705 from python-gitlab/jlvillal/drop\_py\_36  
feat: remove support for Python 3.6, require 3.7 or higher (a390ec3)
- Merge pull request #1693 from python-gitlab/jlvillal/mypy\_test\_meta  
chore: enable mypy for tests/meta/\* (9b78c10)
- Merge pull request #1701 from python-gitlab/jlvillal/func\_test  
chore: correct test\_groups.py test (178ec1a)
- Merge pull request #1696 from python-gitlab/jlvillal/mypy\_merge\_request\_approvals  
chore: add type-hints to gitlab/v4/objects/merge\_request\_approvals.py (2cd15ac)
- Merge pull request #1692 from python-gitlab/jlvillal/mypy\_setup  
chore: check setup.py with mypy (500895a)
- Merge pull request #1681 from python-gitlab/jlvillal/mypy\_ensure\_type\_hints  
Ensure get() methods have correct type-hints (0951989)
- Merge pull request #1683 from python-gitlab/jlvillal/mypy\_setup  
chore: add type-hints to setup.py and check with mypy (a553ee7)
- Merge pull request #1691 from python-gitlab/jlvillal/mypy\_snippets  
chore: add type-hints to gitlab/v4/objects/snippets.py (f775668)
- Merge pull request #1680 from python-gitlab/jlvillal/mypy\_small\_files\_1  
chore: enforce type-hints on most files in gitlab/v4/objects/ (472b300)
- Merge pull request #1678 from python-gitlab/jlvillal/mypy\_commits  
chore: add type hints for gitlab/v4/objects/commits.py (9a2f54c)
- Merge pull request #1677 from python-gitlab/chore/ci-lock-threads  
chore(ci): add workflow to lock old issues (0e6fb5e)
- Merge pull request #1674 from python-gitlab/jlvillal/mypy\_small\_files\_1  
chore: add type-hints to multiple files in gitlab/v4/objects/ (cf801d8)
- Merge pull request #1668 from python-gitlab/jlvillal/mypy\_groups  
chore: add type-hints to gitlab/v4/objects/groups.py (f3688dc)
- Merge pull request #1673 from python-gitlab/jlvillal/mypy\_merge\_requests  
chore: add type-hints to gitlab/v4/objects/merge\_requests.py (32ea954)
- Merge pull request #1670 from python-gitlab/jlvillal/merge\_requests\_api  
docs: add links to the GitLab API docs (4ab9e92)
- Merge pull request #1665 from python-gitlab/renovate/isort-5.x  
chore(deps): update dependency isort to v5.10.0 (f51d9be)

- Merge pull request #1646 from JacobHenner/add-merge-trains  
feat(api): add merge trains ([ed88bce](#))
- Merge pull request #1655 from StingRayZA/add-milestone-promote  
feat(api): add project milestone promotion ([5ce3b17](#))
- Merge pull request #1641 from JacobHenner/add-merge-request-approval-state  
feat(api): add merge request approval state ([422309f](#))
- Merge pull request #1610 from StingRayZA/add-label-promote  
feat(api): add project label promotion ([853d850](#))
- Merge pull request #1629 from python-gitlab/chore/master-to-main  
chore: rename master branch to main ([63b2070](#))
- Merge pull request #1616 from lmmx/patch-1  
Document the update method for project variables ([e851eed](#))
- Merge pull request #1624 from axl89/docs-clarification  
Clarified CI Job Token auth() caveats ([49fae96](#))
- Merge pull request #1515 from JohnVillalovos/jlvillal/mypy\_v4\_obj\_users  
chore: add type-hints to gitlab/v4/objects/users.py ([7753fa2](#))
- Merge pull request #1621 from JohnVillalovos/jlvillal/mypy\_dep  
chore: fix type-check issue shown by new requests-types ([e93f84b](#))
- Merge pull request #1619 from python-gitlab/renovate/python-3.x  
chore(deps): update python docker tag to v3.10 ([d97f79d](#))
- Merge pull request #1617 from python-gitlab/feat/support-3.10  
feat(build): officially support and test python 3.10 ([5c17c36](#))
- Merge pull request #1450 from python-gitlab/renovate/sphinx-4.x  
chore(deps): update dependency sphinx to v4 ([6ce56c2](#))
- Merge pull request #1603 from timgates42/bugfix\_typos  
docs: fix a few typos ([227607c](#))
- Merge pull request #1486 from JohnVillalovos/jlvillal/prohibit\_redirection  
fix!: raise error if there is a 301/302 redirection ([3742405](#))
- Merge pull request #1512 from JohnVillalovos/jlvillal/type\_managers  
chore: improve type-hinting for managers ([5247e8b](#))
- Merge pull request #1585 from JohnVillalovos/jlvillal/archive\_type  
docs: correct documented return type ([557c7d2](#))
- Merge pull request #1565 from javatarz/master  
feat: allow global retry\_transient\_errors ([d98d948](#))

## 7.9.25 v2.10.1 (2021-08-28)

### Chore

- chore(deps): update dependency types-pyyaml to v5.4.8 ([2ae1dd7](#))
- chore(deps): update dependency types-pyyaml to v5.4.7 ([ec8be67](#))
- chore(deps): update codecov/codecov-action action to v2 ([44f4fb7](#))
- chore(deps): update typing dependencies ([34fc210](#))
- chore: define root dir in mypy, not tox ([7a64e67](#))
- chore(deps): group typing requirements with mypy additional\_dependencies ([38597e7](#))
- chore: fix mypy pre-commit hook ([bd50df6](#))
- chore(deps): update dependency types-requests to v2.25.2 ([4782678](#))
- chore(deps): update wagoid/commitlint-github-action action to v4 ([ae97196](#))
- chore(deps): update dependency types-requests to v2.25.1 ([a2d133a](#))
- chore(deps): update precommit hook pycqa/isort to v5.9.3 ([e1954f3](#))
- chore(deps): update dependency isort to v5.9.3 ([ab46e31](#))

### Documentation

- docs(mergequests): gl.mergequests.list documentation was misleading ([5b5a7bc](#))

### Fix

- fix(mixins): improve deprecation warning

Also note what should be changed ([57e0187](#))

- fix(deps): upgrade requests to 2.25.0 (see CVE-2021-33503) ([ce995b2](#))

### Unknown

- Merge pull request #1550 from python-gitlab/renovate/codecov-codecov-action-2.x

chore(deps): update codecov/codecov-action action to v2 ([e54832a](#))

- Merge pull request #1566 from Psychojoker/doc/mergequest\_list\_misleading\_doc

docs(mergequests): gl.mergequests.list documentation was misleading ([8e27721](#))

- Merge pull request #1571 from python-gitlab/fix-mixings-improve-deprecation-warning

fix(mixins): improve deprecation warning ([e2fdffb](#))

## 7.9.26 v2.10.0 (2021-07-28)

### Chore

- chore(deps): update dependency requests to v2.26.0 ([d3ea203](#))
- chore(deps): update precommit hook pycqa/isort to v5.9.2 ([521cddd](#))
- chore(deps): update dependency isort to v5.9.2 ([d5dcf1c](#))

### Documentation

- docs(readme): move contributing docs to CONTRIBUTING.rst

Move the Contributing section of README.rst to CONTRIBUTING.rst, so it is recognized by GitHub and shown when new contributors make pull requests. ([edf49a3](#))

- docs: add example for mr.merge\_ref

Signed-off-by: Matej Focko <mfocko@redhat.com> ([b30b8ac](#))

- docs(project): add example on getting a single project using name with namespace ([ef16a97](#))

### Feature

- feat(api): add merge\_ref for merge requests

Support merge\_ref on merge requests that returns commit of attempted merge of the MR.

Signed-off-by: Matej Focko <mfocko@redhat.com> ([1e24ab2](#))

- feat(api): add name\_regex\_keep attribute in delete\_in\_bulk() ([e49ff3f](#))

### Fix

- fix(api): do not require Release name for creation

Stop requiring a name attribute for creating a Release, since a release name has not been required since GitLab 12.5. ([98cd03b](#))

### Test

- test(functional): add mr.merge\_ref tests
- Add test for using merge\_ref on non-merged MR
- Add test for using merge\_ref on MR with conflicts

Signed-off-by: Matej Focko <mfocko@redhat.com> ([a9924f4](#))

## Unknown

- Merge pull request #1537 from antti-mikael/feat/registry-deleteinbulk-keepregex
- feat(api): add name\_regex\_keep attribute in delete\_in\_bulk() ([85713bb](#))

## 7.9.27 v2.9.0 (2021-06-28)

### Chore

- chore: skip EE test case in functional tests ([953f207](#))
  - chore(deps): update dependency mypy to v0.910 ([02a56f3](#))
  - chore(deps): update dependency types-requests to v2 ([a81a926](#))
  - chore(deps): update precommit hook pycqa/isort to v5.9.1 ([c57ffe3](#))
  - chore(deps): update dependency isort to v5.9.1 ([0479dba](#))
  - chore(deps): update dependency types-requests to v0.1.13 ([c3ddae2](#))
  - chore(deps): update dependency types-requests to v0.1.12 ([f84c2a8](#))
  - chore(deps): update dependency types-pyyaml to v5 ([5c22634](#))
  - chore(deps): update dependency types-pyyaml to v0.1.9 ([1f5b3c0](#))
  - chore(deps): update dependency types-pyyaml to v0.1.8 ([e566767](#))
  - chore(deps): update dependency mypy to v0.902 ([19c9736](#))
  - chore(deps): update dependency types-requests to v0.1.11 ([6ba629c](#))
  - chore: add type-hints to gitlab/v4/objects/projects.py
- Adding type-hints to gitlab/v4/objects/projects.py ([872dd6d](#))

### Documentation

- docs(tags): remove deprecated functions ([1b1a827](#))
- docs(release): add update example ([6254a5f](#))
- docs: make Gitlab class usable for intersphinx ([8753add](#))

### Feature

- feat(release): allow to update release
- Release API now supports PUT. ([b4c4787](#))
- feat(api): add group hooks ([4a7e9b8](#))
  - feat(api): remove responsibility for API inconsistencies for MR reviewers ([3d985ee](#))
  - feat(api): add MR pipeline manager in favor of pipelines() method ([954357c](#))

## Test

- test(releases): integration for release PUT ([13bf61d](#))
- test(releases): add unit-tests for release update ([5b68a5a](#))

## Unknown

- Merge pull request #1533 from sugonyak/add-group-hooks  
feat(api): add group hooks ([6abf13a](#))
- Merge pull request #1522 from PPaques/1521-releases-edit  
Support Release Update API ([33d3428](#))
- Merge pull request #1396 from spyoungtech/merge\_request\_reviewers  
feat(api): add support for creating/editing reviewers in project MRs ([2c86003](#))
- Merge pull request #1528 from python-gitlab/renovate/types-requests-2.x  
chore(deps): update dependency types-requests to v2 ([af7aae7](#))
- Merge pull request #1323 from python-gitlab/feat/mr-pipeline-manager  
feat(api): add merge request pipeline manager and deprecate mr.pipelines() method ([e77554c](#))
- Merge pull request #1513 from python-gitlab/renovate/types-pyyaml-0.x  
chore(deps): update dependency types-pyyaml to v0.1.8 ([e3aa023](#))
- Merge pull request #1514 from python-gitlab/renovate/types-requests-0.x  
chore(deps): update dependency types-requests to v0.1.11 ([82973ce](#))
- Merge pull request #1504 from python-gitlab/renovate/mypy-0.x  
chore(deps): update dependency mypy to v0.902 ([387e147](#))
- Merge pull request #1505 from JohnVillalovos/jlvillal/mypy-deps  
chore: add new required type packages for mypy ([5446423](#))
- Merge pull request #1511 from JohnVillalovos/jlvillal/testing-type-hints  
chore: add type-hints to gitlab/v4/objects/projects.py ([8e6aaf5](#))

## 7.9.28 v2.8.0 (2021-06-10)

### Chore

- chore: add new required type packages for mypy

New version of mypy flagged errors for missing types. Install the recommended type-\* packages that resolve the issues. ([a7371e1](#))

- chore: sync create and update attributes for Projects

Sync the create attributes with: <https://docs.gitlab.com/ee/api/projects.html#create-project>

Sync the update attributes with documentation at: <https://docs.gitlab.com/ee/api/projects.html#edit-project>

As a note the ordering of the attributes was done to match the ordering of the attributes in the documentation.

Closes: #1497 ([0044bd2](#))

- chore: add missing linters to pre-commit and pin versions ([85bbd1a](#))
- chore(ci): use admin PAT for release workflow ([d175d41](#))
- chore: add missing optional create parameter for approval\_rules

Add missing optional create parameter ('protected\_branch\_ids') to the project approvalrules.

[https://docs.gitlab.com/ee/api/merge\\_request\\_approvals.html#create-project-level-rule](https://docs.gitlab.com/ee/api/merge_request_approvals.html#create-project-level-rule) ([06a6001](#))

- chore: apply typing suggestions

Co-authored-by: John Villalovos <[john@sodarock.com](mailto:john@sodarock.com)> ([a11623b](#))

- chore: add type-hints to gitlab/v4/cli.py
  - Add type-hints to gitlab/v4/cli.py
  - Add required type-hints to other files based on adding type-hints to gitlab/v4/cli.py ([2673af0](#))
- chore(ci): ignore .python-version from pyenv ([149953d](#))
- chore: apply suggestions ([fe7d19d](#))
- chore: clean up tox, pre-commit and requirements ([237b97c](#))
- chore: make certain dotfiles searchable by ripgrep

By explicitly NOT excluding the dotfiles we care about to the .gitignore file we make those files searchable by tools like ripgrep.

By default dotfiles are ignored by ripgrep and other search tools (not grep) ([e4ce078](#))

- chore: use built-in function isinstance() instead of getmro()

Code was using inspect.getmro() to replicate the functionality of the built-in function isinstance()

Switch to using isinstance() ([81f6386](#))

- chore: correct a type-hint ([046607c](#))
- chore: move 'gitlab/tests/' dir to 'tests/unit/'

Move the 'gitlab/tests/' directory to 'tests/unit/' so we have all the tests located under the 'tests/' directory. ([1ac0722](#))

- chore: rename 'tools/functional/' to 'tests/functional/'

Rename the 'tools/functional/' directory to 'tests/functional/'

This makes more sense as these are functional tests and not tools.

This was discussed in: <https://github.com/python-gitlab/python-gitlab/discussions/1468> ([502715d](#))

- chore: add a merge\_request() pytest fixture and use it

Added a pytest.fixture for merge\_request(). Use this fixture in tools/functional/api/test\_merge\_requests.py ([8be2838](#))

- chore: simplify functional tests

Add a helper function to have less code duplication in the functional testing. ([df9b5f9](#))

- chore: add functional test mr.merge() with long commit message

Functional test to show that <https://github.com/python-gitlab/python-gitlab/issues/1452> is fixed.

Added a functional test to ensure that we can use large commit message (10\_000+ bytes) in mr.merge()

Related to: #1452 ([cd5993c](#))



- chore: add a functional test for issue #1120

Going to switch to putting parameters from in the query string to having them in the 'data' body section. Add a functional test to make sure that we don't break anything.

<https://github.com/python-gitlab/python-gitlab/issues/1120> (7d66115)

- chore: fix import ordering using isort

Fix the import ordering using isort.

<https://pycqa.github.io/isort/> (f3afd34)

- chore: add an isort tox environment and run isort in CI
  - Add an isort tox environment
  - Run the isort tox environment using `–check` in the Github CI

<https://pycqa.github.io/isort/> (dda646e)

- chore(deps): update precommit hook alessandrojcm/commitlint-pre-commit-hook to v5 (9ff349d)
- chore(deps): update gitlab/gitlab-ce docker tag to v13.11.4-ce.0 (4223269)
- chore(deps): update dependency docker-compose to v1.29.2 (fc241e1)
- chore(ci): ignore debug and type\_checking in coverage (885b608)
- chore(ci): automate releases (0ef497e)
- chore(docs): fix import order for readthedocs build (c3de1fb)
- chore(deps): update gitlab/gitlab-ce docker tag to v13.11.3-ce.0 (f0b52d8)
- chore: have black run at the top-level

This will ensure everything is formatted with black, including setup.py. (429d6c5)

- chore: have flake8 check the entire project

Have flake8 run at the top-level of the projects instead of just the gitlab directory. (ab343ef)

- chore(deps): update gitlab/gitlab-ce docker tag to v13.11.2-ce.0 (434d15d)
- chore: mypy: Disallow untyped definitions

Be more strict and don't allow untyped definitions on the files we check.

Also this adds type-hints for two of the decorators so that now functions/methods decorated by them will have their types be revealed correctly. (6aef2da)

- chore: remove commented-out print (0357c37)

## Documentation

- docs: fix typo in http\_delete docstring (5226f09)
- docs(api): add behavior in local attributes when updating objects (38f65e8)
- docs: fail on warnings during sphinx build

This is useful when docs aren't included in the toctree and don't show up on RTD. (cbd4d52)

## Feature

- feat: add keys endpoint ([a81525a](#))
- feat(objects): add support for Group wikis (#1484)

feat(objects): add support for Group wikis ([74f5e62](#))

- feat(objects): add support for generic packages API ([79d88bd](#))
- feat(api): add support for creating/editing reviewers in project merge requests ([676d1f6](#))
- feat(api): add deployment mergerequests interface ([fbbc0d4](#))
- feat(objects): support all issues statistics endpoints ([f731707](#))
- feat(objects): add support for descendant groups API ([1b70580](#))
- feat(objects): add pipeline test report support ([ee9f96e](#))
- feat(objects): add support for billable members ([fb0b083](#))
- feat: add feature to get inherited member for project/group ([e444b39](#))
- feat: add code owner approval as attribute

The python API was missing the field `code_owner_approval_required` as implemented in the GitLab REST API. ([fdc46ba](#))

- feat: indicate that we are a typed package

By adding the file: `py.typed` it indicates that python-gitlab is a typed package and contains type-hints.

<https://www.python.org/dev/peps/pep-0561/> ([e4421ca](#))

## Fix

- fix: catch invalid type used to initialize RESTObject

Sometimes we have errors where we don't get a dictionary passed to `RESTObject.init()` method. This breaks things but in confusing ways.

Check in the `init()` method and raise an exception if it occurs. ([c7bcc25](#))

- fix: functional project service test (#1500)

chore: fix functional project service test ([093db9d](#))

- fix: ensure kwargs are passed appropriately for `ObjectDeleteMixin` ([4e690c2](#))
- fix(cli): add missing list filter for jobs ([b3d1c26](#))
- fix: change `mr.merge()` to use `'post_data'`

MR <https://github.com/python-gitlab/python-gitlab/pull/1121> changed `mr.merge()` to use `'query_data'`. This appears to have been wrong.

From the Gitlab docs they state it should be sent in a payload body <https://docs.gitlab.com/ee/api/README.html#request-payload> since `mr.merge()` is a PUT request.

> Request Payload

> API Requests can use parameters sent as query strings or as a > payload body. GET requests usually send a query string, while PUT > or POST requests usually send the payload body

Fixes: #1452 Related to: #1120 ([cb6a3c6](#))

- fix(cli): fix parsing CLI objects to classnames ([4252070](#))
- fix(objects): return server data in cancel/retry methods ([9fed061](#))
- fix(objects): add missing group attributes ([d20ff4f](#))
- fix(objects): allow lists for filters for in all objects ([603a351](#))
- fix: iids not working as a list in projects.issues.list()

Set the 'iids' values as type ListAttribute so it will pass the list as a comma-separated string, instead of a list.

Add a functional test.

Closes: #1407 ([45f806c](#))

## Style

- style: clean up test run config ([dfa40c1](#))

## Test

- test(functional): force delete users on reset

Timing issues between requesting group deletion and GitLab enacting that deletion resulted in errors while attempting to delete a user which was the sole owner of said group (see: test\_groups). Pass the 'hard\_delete' parameter to ensure user deletion. ([8f81456](#))

- test(api): fix issues test

Was incorrectly using the issue 'id' vs 'iid'. ([8e5b0de](#))

- test(functional): explicitly remove deploy tokens on reset

Deploy tokens would remain in the instance if the respective project or group was deleted without explicitly revoking the deploy tokens first. ([19a55d8](#))

- test(cli): replace assignment expression

This is a feature added in 3.8, removing it allows for the test to run with lower python versions. ([11ae11b](#))

- test(functional): optionally keep containers running post-tests

Additionally updates token creation to make use of `first_or_create()`, to avoid errors from the script caused by GitLab constraints preventing duplicate tokens with the same value. ([4c475ab](#))

- test(cli): add more real class scenarios ([8cf5031](#))
- test(functional): add test for skip\_groups list filter ([a014774](#))
- test(functional): start tracking functional test coverage ([f875786](#))

## Unknown

- Merge pull request #1487 from JohnVillalovos/jlvillal/check\_attrs  
fix: catch invalid type used to initialize RESTObject ([600a2c1](#))
- Merge pull request #1489 from python-gitlab/chore/release-action-gh-token  
chore(ci): use PAT for release workflow ([161bb0b](#))
- Merge pull request #1499 from JohnVillalovos/jlvillal/projects\_attrs  
chore: sync create and update attributes for Projects ([f91b72a](#))
- Merge pull request #1490 from benjamb/benbrown/keys  
feat: add keys endpoint ([d3fac50](#))
- Merge pull request #1478 from benjamb/benbrown/keep-containers  
Optionally keep containers after running integration tests ([d981956](#))
- Merge pull request #1483 from JohnVillalovos/jlvillal/mypy\_cli  
chore: add type-hints to gitlab/v4/cli.py ([55ae61a](#))
- Merge pull request #1488 from JohnVillalovos/jlvillal/add\_missing\_option  
chore: add missing optional create parameter for approval\_rules ([ac92205](#))
- Merge pull request #1249 from rmonat/master  
feat: add pipeline test report support ([fb7174e](#))
- Merge pull request #1475 from JohnVillalovos/jlvillal/gitignore  
chore: make certain dotfiles searchable by ripgrep ([861d3d2](#))
- Merge pull request #1481 from JohnVillalovos/jlvillal/no\_getmro  
chore: use built-in function issubclass() instead of getmro() ([489b0d3](#))
- Merge pull request #1474 from JohnVillalovos/jlvillal/mv\_unit\_tests  
chore: move 'gitlab/tests/' dir to 'tests/unit/' ([56770ce](#))
- Merge pull request #1480 from JohnVillalovos/jlvillal/fix\_hint  
chore: correct a type-hint ([8eb911d](#))
- Merge pull request #1469 from JohnVillalovos/jlvillal/test\_directory  
chore: rename 'tools/functional/' to 'tests/functional/' ([90ecf2f](#))
- Merge pull request #1465 from JohnVillalovos/jlvillal/fix\_1452\_query\_parameters  
Switch mr.merge() to use post\_data (was using query\_data) ([9beff0d](#))
- Merge pull request #1456 from python-gitlab/feat/billable-members  
feat(objects): add support for billable members ([184b94b](#))
- Merge pull request #1463 from JohnVillalovos/jlvillal/isort  
chore: add isort as a checker ([7824811](#))
- Merge pull request #1290 from python-gitlab/fix/parse-cli-objects-camelcase  
fix(cli): fix parsing CLI objects to classnames ([1508eb7](#))
- Merge pull request #1459 from python-gitlab/renovate/alessandrojcm-commitlint-pre-commit-hook-5.x

chore(deps): update precommit hook alessandrojcm/commitlint-pre-commit-hook to v5 ([ce0e642](#))

- Merge pull request #1376 from Shkurupii/feat-get-inherited-members

feat: get inherited member for project/group ([f35c73e](#))

- Merge pull request #1455 from python-gitlab/renovate/gitlab-gitlab-ce-13.x

chore(deps): update gitlab/gitlab-ce docker tag to v13.11.4-ce.0 ([c4979a8](#))

- Merge pull request #1451 from python-gitlab/renovate/docker-compose-1.x

chore(deps): update dependency docker-compose to v1.29.2 ([3628949](#))

- Merge pull request #1427 from python-gitlab/chore/automate-releases

chore(ci): automate releases ([25695d9](#))

- Merge pull request #1448 from python-gitlab/docs/local-object-attributes

docs(api): add behavior in local attributes when updating objects ([b0b2113](#))

- Merge pull request #1449 from python-gitlab/chore/ignore-typing-coverage

chore(ci): ignore debug and type\_checking in coverage ([62b544d](#))

- Merge pull request #1440 from python-gitlab/test/functional-test-coverage

test(functional): start tracking functional test coverage ([0d3b8ae](#))

- Merge pull request #1420 from python-gitlab/fix/missing-list-attributes

fix(objects): make lists work for filters in all objects ([45edae9](#))

- Merge pull request #1444 from python-gitlab/fix/return-retry-cancel-output

fix(objects): return server data in cancel/retry methods ([1ddb54a](#))

- Merge pull request #1409 from JohnVillalovos/jlvillal/untyped\_defs

chore: mypy: Disallow untyped definitions ([562fbbd](#))

- Merge pull request #1442 from python-gitlab/chore/fix-readthedocs

chore(docs): fix import order for readthedocs build ([b563cdc](#))

- Merge pull request #1441 from python-gitlab/docs/no-manpages-warnings

docs: fail on warnings during sphinx build ([e46cacf](#))

- Merge pull request #1438 from python-gitlab/fix/missing-group-attributes

fix(objects): add missing group attributes ([5061972](#))

- Merge pull request #1434 from python-gitlab/renovate/docker-gitlab-gitlab-ce-13.x

chore(deps): update gitlab/gitlab-ce docker tag to v13.11.3-ce.0 ([1e6305e](#))

- Merge pull request #1437 from daniellanner/feat/api-code-owner-approval

feat: add code owner approval as attribute ([d61e669](#))

- Merge pull request #1433 from JohnVillalovos/jlvillal/black

chore: have black run at the top-level ([09ef8d4](#))

- Merge pull request #1429 from JohnVillalovos/jlvillal/flake8

chore: have flake8 check the entire project ([b498ebd](#))

- Merge pull request #1421 from JohnVillalovos/jlvillal/typed\_gitlab

feat: indicate that we are a typed package ([98891eb](#))

- Merge pull request #1413 from JohnVillalovos/jlvillal/1407

fix: iids not working as a list in projects.issues.list() ([a6b6cd4](#))

- Merge pull request #1352 from JohnVillalovos/jlvillal/fix\_mro

fix: add a check to ensure the MRO is correct ([909aa9a](#))

- Merge pull request #1415 from JohnVillalovos/jlvillal/list\_attribute\_int

feat: add support for lists of integers to ListAttribute ([dde01c7](#))

- Merge pull request #1412 from JohnVillalovos/jlvillal/optional\_get\_attrs

chore: make Get.\*Mixin.\_optional\_get\_attrs always present ([5b81d7d](#))

## 7.9.29 v2.7.1 (2021-04-26)

### Feature

- feat: add support for lists of integers to ListAttribute

Previously ListAttribute only support lists of integers. Now be more flexible and support lists of items which can be coerced into strings, for example integers.

This will help us fix issue #1407 by using ListAttribute for the ‘iids’ field. ([115938b](#))

### Fix

- fix(files): do not url-encode file paths twice ([8e25cec](#))

### Unknown

- Merge pull request #1418 from python-gitlab/fix/urlencode-file-paths

fix(files): do not url-encode filepaths twice ([37af229](#))

## 7.9.30 v2.7.0 (2021-04-25)

### Chore

- chore(objects): remove noisy deprecation warning for audit events

It’s mostly an internal thing anyway and can be removed in 3.0.0 ([2953642](#))

- chore: make Get.\*Mixin.\_optional\_get\_attrs always present

Always create GetMixin/GetWithoutIdMixin.\_optional\_get\_attrs attribute with a default value of tuple()

This way we don’t need to use hasattr() and we will know the type of the attribute. ([3c1a0b3](#))

- chore: make ListMixin.\_list\_filters always present

Always create ListMixin.\_list\_filters attribute with a default value of tuple().

This way we don’t need to use hasattr() and we will know the type of the attribute. ([8933113](#))

- chore: make RESTObject.\_short\_print\_attrs always present

Always create `RESTObject._short_print_attrs` with a default value of `None`.

This way we don't need to use `hasattr()` and we will know the type of the attribute. (6d55120)

- chore: bump version to 2.7.0 (34c4052)
- chore(deps): update gitlab/gitlab-ce docker tag to v13.11.1-ce.0 (3088714)
- chore(deps): update gitlab/gitlab-ce docker tag to v13.11.0-ce.0 (711896f)
- chore: remove unused function `sanitize_parameters()`

The function `sanitize_parameters()` was used when the v3 API was in use. Since v3 API support has been removed there are no more users of this function. (443b934)

- chore: fix F841 errors reported by flake8

Local variable name is assigned to but never used

<https://www.flake8rules.com/rules/F841.html> (40f4ab2)

- chore: fix F401 errors reported by flake8

F401: Module imported but unused

<https://www.flake8rules.com/rules/F401.html> (ff21eb6)

- chore: fix E711 error reported by flake8

E711: Comparison to none should be 'if cond is none:'

<https://www.flake8rules.com/rules/E711.html> (630901b)

- chore: fix E712 errors reported by flake8

E712: Comparison to true should be 'if cond is true:' or 'if cond:'

<https://www.flake8rules.com/rules/E712.html> (83670a4)

- chore: fix E741/E742 errors reported by flake8

Fixes to resolve errors for: <https://www.flake8rules.com/rules/E741.html> Do not use variables named 'I', 'O', or 'l' (E741)

<https://www.flake8rules.com/rules/E742.html>

Do not define classes named `&#39;I&#39;`, `&#39;O&#39;`, or `&#39;l&#39;`; (E742)   
 ↪(['380f227'](<https://github.com/python-gitlab/python-gitlab/commit/380f227a1ecffd5e22ae7aefed95af3b5d830994>))

- chore: fix typo in mr events (c5e6fb3)
- chore(config): allow simple commands without external script (91ffb8e)
- chore: have `_create_attrs` & `_update_attrs` be a namedtuple

Convert `_create_attrs` and `_update_attrs` to use a `NamedTuple` (`RequiredOptional`) to help with code readability. Update all code to use the `NamedTuple`. (aee1f49)

- chore(deps): update dependency docker-compose to v1.29.1 (a89ec43)
- chore(deps): update gitlab/gitlab-ce docker tag to v13.10.3-ce.0 (eabe091)
- chore(deps): update dependency sphinx to v3.5.4 (a886d28)
- chore(deps): update gitlab/gitlab-ce docker tag to v13.10.1-ce.0 (1995361)
- chore(deps): update dependency docker-compose to v1.28.6 (46b05d5)
- chore(deps): update gitlab/gitlab-ce docker tag to v13.10.0-ce.0 (5221e33)

- chore(deps): update gitlab/gitlab-ce docker tag to v13.9.4-ce.0 ([939f769](#))
- chore: remove usage of getattr()

Remove usage of getattr(self, “\_update\_uses\_post”, False)

Instead add it to class and set default value to False.

Add a tests that shows it is set to True for the ProjectMergeRequestApprovalManager and ProjectApprovalManager classes. ([2afd18a](#))

- chore: fix package file test naming ([8c80268](#))
- chore: add \_create\_attrs & \_update\_attrs to RESTManager

Add the attributes: \_create\_attrs and \_update\_attrs to the RESTManager class. This is so that we stop using getattr() if we don’t need to.

This also helps with type-hints being available for these attributes. ([147f05d](#))

- chore(deps): update gitlab/gitlab-ce docker tag to v13.9.3-ce.0 ([2ddf45f](#))
- chore: make \_types always present in RESTManager

We now create \_types = {} in RESTManager class.

By making \_types always present in RESTManager it makes the code simpler. We no longer have to do: types = getattr(self, “\_types”, {})

And the type checker now understands the type. ([924f83e](#))

- chore: make lint happy ([7a7c9fd](#))
- chore: make lint happy ([b5f43c8](#))
- chore: make lint happy ([732e49c](#))
- chore: import audit events in objects ([35a190c](#))
- chore: add type-hints for gitlab/mixins.py
  - Added type-hints for gitlab/mixins.py
  - Changed use of filter with a lambda expression to list-comprehension. mypy was not able to understand the previous code. Also list-comprehension is better :) ([baea721](#))
- chore(deps): update dependency sphinx to v3.5.2 ([9dee5c4](#))
- chore: add test ([f8cf1e1](#))
- chore(deps): update gitlab/gitlab-ce docker tag to v13.9.2-ce.0 ([933ba52](#))
- chore: put assert statements inside ‘if TYPE\_CHECKING:’

To be safe that we don’t assert while running, put the assert statements, which are used by mypy to check that types are correct, inside an ‘if TYPE\_CHECKING:’ block.

Also, instead of asserting that the item is a dict, instead assert that it is not a requests.Response object. Theoretically the JSON could return as a list or dict, though at this time we are assuming a dict. ([b562458](#))

- chore: add type hints to gitlab/base.py:RESTManager

Add some additional type hints to gitlab/base.py ([9c55593](#))

- chore: del ‘import \*’ in gitlab/v4/objects/project\_access\_tokens.py

Remove usage of ‘import \*’ in gitlab/v4/objects/project\_access\_tokens.py. ([9efbe12](#))

- chore: disallow incomplete type defs



Don't allow a partially annotated function definition. Either none of the function is annotated or all of it must be.

Update code to ensure no-more partially annotated functions.

Update gitlab/cli.py with better type-hints. Changed `Tuple[Any, ...]` to `Tuple[str, ...]` (907634f)

- chore(api): move repository endpoints into separate module (1ed154c)
- chore: add additional type-hints for gitlab/base.py

Add type-hints for the variables which are set via `self.dict`

mypy doesn't see them when they are assigned via `self.dict`. So declare them in the class definition. (ad72ef3)

- chore: add and fix some type-hints in gitlab/client.py

Was able to figure out better type-hints for gitlab/client.py (8837207)

- chore: remove import of gitlab.utils from `init.py`

Initially when extracting out the gitlab/client.py code we tried to remove this but functional tests failed.

Later we fixed the functional test that was failing, so now remove the unneeded import. (39b9183)

- chore: add type-hints to gitlab/client.py

Adding some initial type-hints to gitlab/client.py (c9e5b4f)

- chore: improve type-hints for gitlab/base.py

Determined the base class for `obj_cls` and adding type-hints for it. (cbd43d0)

- chore(deps): update dependency docker-compose to v1.28.5 (f4ab558)
- chore: add type-hints to gitlab/cli.py (10b7b83)
- chore: add type-hints to gitlab/config.py (213e563)
- chore: add type hints to gitlab/utils.py (acd9294)
- chore: add type-hints to gitlab/const.py (a10a777)
- chore(deps): update wagoid/commitlint-github-action action to v3 (b3274cf)
- chore(deps): update gitlab/gitlab-ce docker tag to v13.9.1-ce.0 (f6fd995)
- chore: remove usage of 'from ... import \*'

In gitlab/v4/objects/\*.py remove usage of:

- from gitlab.base import \*
- from gitlab.mixins import \*

Change them to:

- from gitlab.base import CLASS\_NAME
- from gitlab.mixins import CLASS\_NAME

Programmatically update code to explicitly import needed classes only.

After the change the output of: `$ flake8 gitlab/v4/objects/*.py | grep 'REST|Mixin'`

Is empty. Before many messages about unable to determine if it was a valid name. (c83eaf4)

- chore: remove unused function `_construct_url()`

The function `_construct_url()` was used by the v3 API. All usage of the function was removed in commit fe89b949922c028830dd49095432ba627d330186 (009d369)

- chore: add type hints to gitlab/base.py (3727cbd)
- chore: remove usage of ‘from ... import \*’ in client.py

In gitlab/client.py remove usage of:

- from gitlab.const import \*
- from gitlab.exceptions import \*

Change them to:

- import gitlab.const
- import gitlab.exceptions

Update code to explicitly reference things in gitlab.const and gitlab.exceptions

A flake8 run no longer lists any undefined variables. Before it listed possible undefined variables. (bf0c8c5)

- chore(deps): update gitlab/gitlab-ce docker tag to v13.9.0-ce.0 (3aef19c)
- chore: explicitly import gitlab.v4.objects/cli

As we only support the v4 Gitlab API, explicitly import gitlab.v4.objects and gitlab.v4.clie instead of dynamically importing it depending on the API version.

This has the added benefit of mypy being able to type check the Gitlab `init()` function as currently it will fail if we enable type checking of `init()` it will fail.

Also, this also helps by not confusing tools like pyinstaller/cx\_freeze with dynamic imports so you don’t need hooks for standalone executables. And according to <https://docs.gitlab.com/ee/api/>,

&#34;GraphQL co-exists with the current v4 REST API. If we have a v5 API, this should be a compatibility layer on top of GraphQL.&#34; ([`233b79e`](https://github.com/python-gitlab/python-gitlab/commit/233b79ed442aac66faf9eb4b0087ea126d6dffc5))

- chore(objects): make Project refreshable

Helps getting the real state of the project from the server. (958a6aa)

- chore(tests): remove unused URL segment (66f0b6c)
- chore(deps): update dependency docker-compose to v1.28.4 (8938484)
- chore(deps): update dependency docker-compose to v1.28.3 (2358d48)
- chore(deps): update dependency sphinx to v3.5.1 (f916f09)
- chore: remove unused ALLOWED\_KEYSET\_ENDPOINTS variable

The variable ALLOWED\_KEYSET\_ENDPOINTS was added in commit f86ef3bbdb5bffa1348a802e62b281d3f31d33ad.

Then most of that commit was removed in commit e71fe16b47835aa4db2834e98c7ffc6bdec36723, but ALLOWED\_KEYSET\_ENDPOINTS was missed. (3d5d5d8)

- chore: remove Python 2 code

httplib is a Python 2 library. It was renamed to http.client in Python 3.

<https://docs.python.org/2.7/library/httplib.html> (b5d4e40)

- chore(deps): update dependency sphinx to v3.5.0 (188c5b6)
- chore(deps): update gitlab/gitlab-ce docker tag to v13.8.4-ce.0 (832cb88)
- chore(ci): deduplicate PR jobs (63918c3)
- chore(deps): update gitlab/gitlab-ce docker tag to v13.8.3-ce.0 (e6c20f1)

- chore(deps): update precommit hook alessandrojcm/commitlint-pre-commit-hook to v4 ([505a8b8](#))
- chore(deps): update gitlab/gitlab-ce docker tag to v13.8.2-ce.0 ([7c12038](#))
- chore(deps): update dependency sphinx to v3.4.3 ([37c992c](#))

## Documentation

- docs(api): add examples for resource state events ([4d00c12](#))
- docs: add information about the gitter community

Add a section in the README.rst about the gitter community. The badge already exists and is useful but very easy to miss. ([6ff67e7](#))

- docs(api): add release links API docs ([36d65f0](#))
- docs: add docs and examples for custom user agent ([a69a214](#))
- docs: change travis-ci badge to githubactions ([2ba5ba2](#))

## Feature

- feat(objects): add support for resource state events API ([d4799c4](#))
- feat: add ProjectPackageFile

Add ProjectPackageFile and the ability to list project package package\_files.

Fixes #1372 ([b9d469b](#))

- feat(objects): add support for group audit events API ([2a0fbdf](#))
- feat: option to add a helper to lookup token ([8ecf559](#))
- feat(users): add follow/unfollow API ([e456869](#))
- feat(projects): add project access token api ([1becef0](#))
- feat: add an initial mypy test to tox.ini

Add an initial mypy test to test gitlab/base.py and gitlab/init.py ([fdec039](#))

- feat(objects): add Release Links API support ([28d7518](#))
- feat: add project audit endpoint ([6660dbe](#))
- feat: add personal access token API

See: [https://docs.gitlab.com/ee/api/personal\\_access\\_tokens.html](https://docs.gitlab.com/ee/api/personal_access_tokens.html) ([2bb16fa](#))

- feat(issues): add missing get verb to IssueManager ([f78ebe0](#))
- feat: import from bitbucket server

I'd like to use this library to automate importing Bitbucket Server repositories into GitLab. There is a [GitLab API endpoint](#) to do this, but it is not exposed through this library.

- Add an `import_bitbucket_server` method to the `ProjectManager`. This method calls this GitLab API endpoint: <https://docs.gitlab.com/ee/api/import.html#import-repository-from-bitbucket-server>
- Modify `import_gitlab` method docstring for python3 compatibility
- Add a skipped stub test for the existing `import_github` method ([ff3013a](#))
- feat(api,cli): make user agent configurable ([4bb201b](#))

## Fix

- fix: add a check to ensure the MRO is correct

Add a check to ensure the MRO (Method Resolution Order) is correct for classes in `gitlab.v4.objects` when doing type-checking.

An example of an incorrect definition: `class ProjectPipeline(RESTObject, RefreshMixin, ObjectDeleteMixin):`  
^^^^^^^^^^ This should be at the end.

Correct way would be: `class ProjectPipeline(RefreshMixin, ObjectDeleteMixin, RESTObject):` Correctly at the end  
^^^^^^^^^^

Also fix classes which have the issue. ([565d548](#))

- fix: correct `ProjectFile.decode()` documentation

`ProjectFile.decode()` returns 'bytes' and not 'str'.

Update the method's doc-string and add a type-hint.

`ProjectFile.decode()` returns the result of a call to `base64.b64decode()`

The docs for that function state it returns 'bytes': <https://docs.python.org/3/library/base64.html#base64.b64decode>

Fixes: [#1403 \(b180baf\)](#)

- fix: update user's bool data and avatar

If we want to update email, avatar and do not send email confirmation change (`skip_reconfirmation = True`), `MultipartEncoder` will try to encode everything except None and bytes. So it tries to encode bools. Casting bool's values to their stringified int representation fix it. ([3ba27ff](#))

- fix: argument type was not a tuple as expected

While adding type-hints mypy flagged this as an issue. The third argument to `register_custom_action` is supposed to be a tuple. It was being passed as a string rather than a tuple of strings. ([062f8f6](#))

- fix: correct some type-hints in `gitlab/mixins.py`

Commit [baea7215bbbe07c06b2ca0f97a1d3d482668d887](#) introduced type-hints for `gitlab/mixins.py`.

After starting to add type-hints to `gitlab/v4/objects/users.py` discovered a few errors.

Main error was using '=' instead of ':'. For example: `_parent = Optional[...]` should be `_parent: Optional[...]`

Resolved those issues. ([8bd3124](#))

- fix: only append kwargs as query parameters

Some arguments to `http_request` were being read from kwargs, but kwargs is where this function creates query parameters from, by default. In the absence of a `query_parameters` param, the function would construct URLs with query parameters such as `retry_transient_errors=True` despite those parameters having no meaning to the API to which the request was sent.

This change names those arguments that are specific to `http_request` so that they do not end up as query parameters read from kwargs. ([b9ecc9a](#))

- fix: only add `query_parameters` to `GitlabList` once

Fixes [#1386 \(ca2c3c9\)](#)

- fix(types): prevent `dir` from producing duplicates ([5bf7525](#))
- fix: checking if `RESTManager._from_parent_attrs` is set

Prior to commit 3727cbd21fc40b312573ca8da56e0f6cf9577d08 `RESTManager._from_parent_attrs` did not exist unless it was explicitly set. But commit 3727cbd21fc40b312573ca8da56e0f6cf9577d08 set it to a default value of `{}`.

So the checks using `hasattr()` were no longer valid.

Update the checks to check if `RESTManager._from_parent_attrs` has a value. (8224b40)

- fix: handling config value in `_get_values_from_helper` (9dfb4cd)
- fix: update doc for token helper (3ac6fa1)
- fix: let the `homedir` be expanded in path of helper (fc7387a)
- fix: make secret helper more user friendly (fc2798f)
- fix(objects): add single get endpoint for instance audit events (c3f0a6f)
- fix: linting issues and test (b04dd2c)
- fix: better real life token lookup example (9ef8311)
- fix: handle tags like `debian/2%2.6-21` as identifiers

Git refnames are relatively free-form and can contain all sort for special characters, not just `/` and `#`, see <http://git-scm.com/docs/git-check-ref-format>

In particular, Debian's DEP-14 standard for storing packaging in git repositories mandates the use of the `%` character in tags in some cases like `debian/2%2.6-21`.

Unfortunately python-gitlab currently only escapes `/` to `%2F` and in some cases `#` to `%23`. This means that when using the commit API to retrieve information about the `debian/2%2.6-21` tag only the slash is escaped before being inserted in the URL path and the `%` is left untouched, resulting in something like `/api/v4/projects/123/repository/commits/debian%2F2%2.6-21`. When `urllib3` sees that it detects the invalid `%` escape and then urlencodes the whole string, resulting in `/api/v4/projects/123/repository/commits/debian%252F2%252.6-21`, where the original `/` got escaped twice and produced `%252F`.

To avoid the issue, fully urlencode identifiers and parameters to avoid the `urllib3` auto-escaping in all cases.

Signed-off-by: Emanuele Aina <[emanuele.aina@collabora.com](mailto:emanuele.aina@collabora.com)> (b4dac5c)

- fix: remove duplicate class definitions in `v4/objects/users.py`

The classes `UserStatus` and `UserStatusManager` were each declared twice. Remove the duplicate declarations. (7c4e625)

- fix: wrong variable name

Discovered this when I ran `flake8` on the file. Unfortunately I was the one who introduced this wrong variable name :( (15ec41c)

- fix: tox pep8 target, so that it can run

Previously running the pep8 target would fail as `flake8` was not installed.

Now install `flake8` for the pep8 target.

NOTE: Running the pep8 target fails as there are many warnings/errors. But it does allow us to run it and possibly work on reducing these warnings/errors in the future.

In addition, add two checks to the ignore list as `black` takes care of formatting. The two checks added to the ignore list are:

- E501: line too long
- W503: line break before binary operator (f518e87)
- fix: undefined name errors

Discovered that there were some undefined names. (48ec9e0)

- fix: extend wait timeout for test\_delete\_user()

Have been seeing intermittent failures of the test\_delete\_user() functional test. Have made the following changes to hopefully resolve the issue and if it still fails to know better why the failure occurred.

- Extend the wait timeout for test\_delete\_user() from 30 to 60 tries of 0.5 seconds each.
- Modify wait\_for\_sidekiq() to return True if sidekiq process terminated. Return False if the timeout expired.
- Modify wait\_for\_sidekiq() to loop through all processes instead of assuming there is only one process. If all processes are not busy then return.
- Modify wait\_for\_sidekiq() to sleep at least once before checking for processes being busy.
- Check for True being returned in test\_delete\_user() call to wait\_for\_sidekiq() (19fde8e)
- fix: test\_update\_group() dependency on ordering

Since there are two groups we can't depend on the one we changed to always be the first one returned.

Instead fetch the group we want and then test our assertion against that group. (e78a8d6)

- fix: honor parameter value passed

Gitlab allows setting the defaults for MR to delete the source. Also the inline help of the CLI suggest that a boolean is expected, but no matter what value you set, it will always delete. (c2f8f0e)

## Refactor

- refactor(objects): move instance audit events where they belong (48ba88f)
- refactor: move Gitlab and GitlabList to gitlab/client.py

Move the classes Gitlab and GitlabList from gitlab/**init**.py to the newly created gitlab/client.py file.

Update one test case that was depending on requests being defined in gitlab/**init**.py (53a7645)

- refactor(api): explicitly export classes for star imports (f05c287)
- refactor(v4): split objects and managers per API resource (a5a48ad)

## Test

- test(object): add test for **dir** duplicates (a8e591f)
- test(objects): add tests for resource state events (10225cf)
- test(objects): add unit test for instance audit events (84e3247)
- test: don't add duplicate fixture

Co-authored-by: Nejc Habjan <hab.nejc@gmail.com> (5d94846)

- test(api): add functional test for release links API (ab2a1c8)
- test(api,cli): add tests for custom user agent (c5a37e7)

## Unknown

- Merge pull request #1408 from python-gitlab/chore/bump-to-2-7-0  
chore: bump version to 2.7.0 ([e37de18](#))
- Merge pull request #1411 from JohnVillalovos/jlvillal/list\_filters  
chore: make ListMixin.\_list\_filters always present ([62c75b5](#))
- Merge pull request #1410 from JohnVillalovos/jlvillal/short\_print\_attr  
chore: make RESTObject.\_short\_print\_attrs always present ([09522b3](#))
- Merge pull request #1414 from python-gitlab/chore/remove-noisy-deprecation-warning  
chore(objects): remove noisy deprecation warning for audit events ([0a0fc9f](#))
- Merge pull request #1392 from bbatliner/patch-1  
Improvements to HTTP requests ([cfc42d2](#))
- Merge pull request #1406 from python-gitlab/renovate/docker-gitlab-gitlab-ce-13.x  
chore(deps): update gitlab/gitlab-ce docker tag to v13.11.1-ce.0 ([4f79d9f](#))
- Merge pull request #1405 from JohnVillalovos/jlvillal/returns\_bytes  
fix: correct ProjectFile.decode() documentation ([c055de0](#))
- Merge pull request #1397 from JohnVillalovos/jlvillal/flake8  
Fix all issues reported by running: tox -e pep8 and enable pep8 as a linter check ([976e14d](#))
- Merge pull request #1404 from DylannCordel/fix-upd-user-bool-data-and-avatar  
fix: update user's bool data and avatar ([5fac07a](#))
- Merge pull request #1383 from spyoungtech/dirfix  
fix(types): prevent **dir** in RestObject from producing duplicates ([60c5fd8](#))
- Merge pull request #1400 from JohnVillalovos/jlvillal/sanitize  
chore: remove unused function sanitize\_parameters() ([dd236a0](#))
- Merge pull request #1398 from JohnVillalovos/jlvillal/mypy\_mixins  
fix: correct some type-hints in gitlab/mixins.py ([5b18d20](#))
- Merge pull request #1399 from JohnVillalovos/jlvillal/fix\_custom\_action  
fix: argument type was not a tuple as expected ([fc4f7fd](#))
- Merge pull request #1364 from python-gitlab/feat/resource-state-events  
feat: add support for resource state events API ([916a7fe](#))
- Merge pull request #1359 from klorenz/feat\_token\_lookup  
feat(config): allow using a credential helper to lookup tokens ([af781c1](#))
- Merge pull request #1375 from JohnVillalovos/jlvillal/update\_uses\_post  
chore: remove usage of getattr() ([d236267](#))
- Merge pull request #1366 from JohnVillalovos/jlvillal/create\_attrs  
chore: have \_create\_attrs & \_update\_attrs be a namedtuple ([d1697d4](#))
- Merge pull request #1391 from python-gitlab/renovate/docker-compose-1.x

chore(deps): update dependency docker-compose to v1.29.1 ([a6d3556](#))

- Merge pull request #1380 from python-gitlab/renovate/sphinx-3.x

chore(deps): update dependency sphinx to v3.5.4 ([6b86878](#))

- Merge pull request #1363 from python-gitlab/feat/all-audit-events

Feat: cover all audit events ([02ce49e](#))

- Merge pull request #1382 from python-gitlab/renovate/docker-compose-1.x

chore(deps): update dependency docker-compose to v1.28.6 ([e798c9b](#))

- Merge pull request #1373 from JacobHenner/jacobhenner/add-package\_files

feat: add support for Project Package Files ([8ace76a](#))

- Merge pull request #1371 from JohnVillalovos/jlvillal/create\_attrs\_1

chore: add \_create\_attrs & \_update\_attrs to RESTManager ([8603248](#))

- Merge pull request #1369 from python-gitlab/renovate/docker-gitlab-gitlab-ce-13.x

chore(deps): update gitlab/gitlab-ce docker tag to v13.9.3-ce.0 ([6fde243](#))

- Merge pull request #1367 from JohnVillalovos/jlvillal/from\_parent\_attrs

fix: checking if RESTManager.\_from\_parent\_attrs is set ([f93b9b5](#))

- Merge pull request #1365 from JohnVillalovos/jlvillal/getattr

chore: make \_types always present in RESTManager ([de73ea7](#))

- Merge pull request #1336 from em-/fix/quote-everything

fix: handle tags like debian/2%2.6-21 as identifiers ([48fc907](#))

- Merge pull request #1344 from JohnVillalovos/jlvillal/mixins

chore: add type-hints for gitlab/mixins.py ([63ecd2e](#))

- Merge pull request #1353 from JohnVillalovos/jlvillal/mypy\_base

chore: add type hints to gitlab/base.py:RESTManager ([ebdfec7](#))

- Merge pull request #1350 from JohnVillalovos/jlvillal/isinstance

chore: Put assert statements inside 'if TYPE\_CHECKING:' ([c530f75](#))

- Merge pull request #1361 from python-gitlab/renovate/sphinx-3.x

chore(deps): update dependency sphinx to v3.5.2 ([c7a0669](#))

- Merge pull request #1358 from python-gitlab/renovate/docker-gitlab-gitlab-ce-13.x

chore(deps): update gitlab/gitlab-ce docker tag to v13.9.2-ce.0 ([aa13214](#))

- Merge pull request #1351 from JohnVillalovos/jlvillal/import\_start

chore: del 'import \*' in gitlab/v4/objects/project\_access\_tokens.py ([96d2805](#))

- Merge pull request #1342 from JohnVillalovos/jlvillal/mypy\_incomplete

chore: disallow incomplete type defs ([5f23ed9](#))

- Merge pull request #1347 from python-gitlab/chore/split-repository-methods

chore(api): move repository endpoints into separate module ([d8b8a0a](#))

- Merge pull request #1343 from JohnVillalovos/jlvillal/mypy\_testing\_things



chore: add and fix some type-hints in gitlab/client.py ([f5a65f0](#))

- Merge pull request #1345 from JohnVillalovos/jlvillal/mypy\_base\_fixes

chore: add additional type-hints for gitlab/base.py ([7441455](#))

- Merge pull request #1333 from python-gitlab/feat/user-follow-api

feat(users): add follow/unfollow API ([5bc158d](#))

- Merge pull request #1339 from JohnVillalovos/jlvillal/mypy\_client\_py

chore: add type-hints to gitlab/client.py ([b0d75d9](#))

- Merge pull request #1341 from JohnVillalovos/jlvillal/gitter

doc: add information about the gitter community ([adab83a](#))

- Merge pull request #1340 from JohnVillalovos/jlvillal/gitlab\_init

chore: remove import of gitlab.utils from **init.py** ([d0eb1b5](#))

- Merge pull request #1338 from JohnVillalovos/jlvillal/mypy\_base

Improve type-hints for gitlab/base.py ([cbd4f1e](#))

- Merge pull request #1334 from JohnVillalovos/jlvillal/mypy\_cli

chore: add type-hints to gitlab/cli.py ([f909cae](#))

- Merge pull request #1337 from python-gitlab/renovate/docker-compose-1.x

chore(deps): update dependency docker-compose to v1.28.5 ([bd62fed](#))

- Merge pull request #1335 from JohnVillalovos/jlvillal/remove\_dup\_classes

fix: remove duplicate class definitions in v4/objects/users.py ([ecb686d](#))

- Merge pull request #1328 from python-gitlab/renovate/wagoid-commitlint-github-action-3.x

chore(deps): update wagoid/commitlint-github-action action to v3 ([6662252](#))

- Merge pull request #1329 from JohnVillalovos/jlvillal/mypy\_const

Add type-hints to gitlab/const.py ([ca4c1d9](#))

- Merge pull request #1330 from JohnVillalovos/jlvillal/mypy\_utils

chore: add type hints to gitlab/utils.py ([a1a8bfe](#))

- Merge pull request #1331 from JohnVillalovos/jlvillal/mypy\_config

chore: add type-hints to gitlab/config.py ([d207074](#))

- Merge pull request #1332 from JohnVillalovos/jlvillal/fix\_variable

chore: fix wrong variable name in cli.py ([665c0c3](#))

- Merge pull request #1319 from JohnVillalovos/jlvillal/import\_star

chore: remove usage of 'from ... import \*' ([0b67ca2](#))

- Merge pull request #1327 from python-gitlab/feat/project-access-token-api

feat(projects): add project access token api ([e06c51b](#))

- Merge pull request #1325 from JohnVillalovos/jlvillal/pep8

fix: tox pep8 target, so that it can run ([bb227d3](#))

- Merge pull request #1322 from JohnVillalovos/jlvillal/missing\_vars

fix: undefined name errors in v4 objects ([a7ec67f](#))

- Merge pull request #1321 from JohnVillalovos/jlvillal/remove\_cruft

chore: remove unused function `_construct_url()` ([d90be1e](#))

- Merge pull request #1299 from JohnVillalovos/mypy

Enable mypy type checking and add type hints to `gitlab/base.py` ([a18bc5c](#))

- Merge pull request #1318 from JohnVillalovos/jlvillal/testing

chore: remove usage of `'from ... import *'` in `client.py` ([d9fdf1d](#))

- Merge pull request #1310 from JohnVillalovos/jlvillal/v4\_only

chore: explicitly import `gitlab.v4.objects/cli` ([8c58b07](#))

- Merge pull request #1316 from JohnVillalovos/jlvillal/test\_wait

test: extend wait timeout for `test_delete_user()` ([5cc60d5](#))

- Merge pull request #1314 from python-gitlab/feat/release-links

feat: add release links API support ([2b29776](#))

- Merge pull request #1311 from JohnVillalovos/jlvillal/fix\_functional

fix: `test_update_group()` dependency on ordering ([3381700](#))

- Merge pull request #1307 from python-gitlab/renovate/docker-compose-1.x

chore(deps): update dependency docker-compose to v1.28.4 ([649385c](#))

- Merge pull request #1308 from Sineaggi/add-project-audit-endpoint

feat: add project audit endpoint ([0c5a23e](#))

- Merge pull request #1301 from JohnVillalovos/refactor\_jlvillal

refactor: move `Gitlab` and `GitlabList` to `gitlab/client.py` ([2c4fcf8](#))

- Merge pull request #1305 from python-gitlab/renovate/docker-compose-1.x

chore(deps): update dependency docker-compose to v1.28.3 ([d4e7a03](#))

- Merge pull request #1304 from python-gitlab/feat/personal-access-token-api

feat: add personal access token API ([ef8fcf7](#))

- Merge pull request #1300 from JohnVillalovos/remove\_cruft

chore: remove unused `ALLOWED_KEYSET_ENDPOINTS` variable ([bec2094](#))

- Merge pull request #1303 from python-gitlab/renovate/sphinx-3.x

chore(deps): update dependency sphinx to v3.5.1 ([9f6691d](#))

- Merge pull request #1271 from allcloud-jonathan/feature/honor-bool-for-delete-source

fix: honor parameter value passed ([1552eb5](#))

- Merge pull request #1298 from JohnVillalovos/master

Remove Python 2 code ([07edafe](#))

- Merge pull request #1288 from python-gitlab/refactor/split-objects

refactor(v4): split objects and managers per API resource ([9fcd962](#))

- Merge pull request #1295 from python-gitlab/renovate/sphinx-3.x

chore(deps): update dependency sphinx to v3.5.0 ([76e6f87](#))

- Merge pull request #1292 from python-gitlab/renovate/docker-gitlab-gitlab-ce-13.x

chore(deps): update gitlab/gitlab-ce docker tag to v13.8.4-ce.0 ([45fc49e](#))

- Merge pull request #1223 from python-gitlab/feat/single-issue-api

feat(issues): add missing get verb to IssueManager ([9d6c188](#))

- Merge pull request #1287 from python-gitlab/chore/deduplicate-pr-jobs

chore(ci): deduplicate PR jobs ([9fe506f](#))

- Merge pull request #1283 from fajpunk/import\_bitbucket

feat: import from bitbucket server ([b48563f](#))

- Merge pull request #1244 from python-gitlab/renovate/alessandrojcm-commitlint-pre-commit-hook-4.x

chore(deps): update precommit hook alessandrojcm/commitlint-pre-commit-hook to v4 ([3935baf](#))

- Merge pull request #1281 from python-gitlab/renovate/docker-gitlab-gitlab-ce-13.x

chore(deps): update gitlab/gitlab-ce docker tag to v13.8.2-ce.0 ([1d9155c](#))

- Merge pull request #1225 from python-gitlab/renovate/sphinx-3.x

chore(deps): update dependency sphinx to v3.4.3 ([a985c34](#))

- Merge pull request #1277 from python-gitlab/feat/override-user-agent

feat(api,cli): make user agent configurable ([643454c](#))

- Merge pull request #1276 from python-gitlab/bufferoverflow-patch-1

docs: switch from travis-ci.org to GitHub Actions ([071d699](#))

## 7.9.31 v2.6.0 (2021-01-29)

### Chore

- chore: officially support and test 3.9 ([62dd07d](#))
- chore(deps): pin dependency requests-toolbelt to ==0.9.1 ([4d25f20](#))
- chore(deps): update dependency requests to v2.25.1 ([9c2789e](#))
- chore(deps): update gitlab/gitlab-ce docker tag to v13.8.1-ce.0 ([9854d6d](#))
- chore(ci): add coverage and docs jobs ([2de64cf](#))
- chore(ci): force colors in pytest runs ([1502079](#))
- chore(ci): pin docker-compose install for tests

This ensures python-dotenv with expected behavior for .env processing ([1f7a2ab](#))

- chore(ci): pin os version ([cfa27ac](#))
- chore(ci): fix typo in matrix ([5e1547a](#))
- chore(ci): fix copy/paste oopsie ([c6241e7](#))
- chore(ci): add pytest PR annotations ([8f92230](#))
- chore(ci): replace travis with Actions ([8bb73a3](#))

- chore: move .env into docker-compose dir ([55cbd1c](#))
- chore(deps): update gitlab/gitlab-ce docker tag to v13.5.4-ce.0 ([265dbbd](#))
- chore(deps): update gitlab/gitlab-ce docker tag to v13.5.3-ce.0 ([d1b0b08](#))
- chore(deps): update gitlab/gitlab-ce docker tag to v13.5.2-ce.0 ([4a6831c](#))
- chore(deps): update gitlab/gitlab-ce docker tag to v13.5.1-ce.0 ([348e860](#))
- chore(deps): update gitlab/gitlab-ce docker tag to v13.5.0-ce.0 ([fc205cc](#))
- chore(docs): always edit the file directly on master

There is no way to edit the raw commit ([35e43c5](#))

- chore(deps): update gitlab/gitlab-ce docker tag to v13.4.3-ce.0 ([bc17889](#))
- chore(cli): remove python2 code ([1030e0a](#))
- chore: apply suggestions ([65ce026](#))
- chore(deps): update python docker tag to v3.9 ([1fc65e0](#))
- chore: simplified search scope constants ([16fc048](#))
- chore: added docs for search scopes constants ([7565bf0](#))
- chore: use helper fixtures for test directories ([40ec2f5](#))
- chore: allow overriding docker-compose env vars for tag ([27109ca](#))
- chore: remove unnecessary random function ([d4ee0a6](#))
- chore(deps): pin dependencies ([14d8f77](#))
- chore(deps): update gitlab/gitlab-ce docker tag to v13.3.6-ce.0 ([57b5782](#))
- chore(test): remove hacking dependencies ([9384493](#))
- chore(ci): add .readthedocs.yml ([0ad441e](#))
- chore(ci): reduce renovate PR noise ([f4d7a55](#))
- chore(deps): update gitlab/gitlab-ce docker tag to v13.3.5-ce.0 ([c88d870](#))
- chore(deps): update gitlab/gitlab-ce docker tag to v13.3.4-ce.0 ([e94c4c6](#))
- chore(deps): update gitlab/gitlab-ce docker tag to v13.3.3-ce.0 ([667bf01](#))

## Documentation

- docs(cli-usage): fixed term ([d282a99](#))
- docs(readme): update supported Python versions ([20b1e79](#))
- docs(cli): use inline anonymous references for external links

There doesn't seem to be an obvious way to use an alias for identical text labels that link to different targets. With inline links we can work around this shortcoming. Until we know better. ([f2cf467](#))

- docs(groups): add example for creating subgroups ([92eb4e3](#))
- docs: clean up grammar and formatting in documentation ([aff9bc7](#))
- docs: add Project Merge Request approval rule documentation ([449fc26](#))
- docs(readme): also add hint to delete gitlab-runner-test

Otherwise the whole testsuite will refuse to run (8894f2d)

- docs(issues): add admin, project owner hint

Closes #1101 (609c03b)

- docs(projects): correct fork docs

Closes #1126 (54921db)

- docs(cli): add example for job artifacts download (375b29d)
- docs(cli): add auto-generated CLI reference (6c21fc8)

## Feature

- feat: support multipart uploads (2fa3004)
- feat(tests): test label getter (a41af90)
- feat: add MINIMAL\_ACCESS constant

A “minimal access” access level was introduced in GitLab 13.5. (49eb3ca)

- feat: unit tests added (f37ebf5)
- feat: added support for pipeline bridges (05cbdc2)
- feat: adds support for project merge request approval rules (#1199) (c6fbf39)
- feat(api): added wip filter param for merge requests (d6078f8)
- feat(api): added wip filter param for merge requests (aa6e80d)
- feat(api): add support for user identity provider deletion (e78e121)

## Fix

- fix(api): use RetrieveMixin for ProjectLabelManager

Allows to get a single label from a project, which was missing before even though the GitLab API has the ability to. (1a14395)

- fix(base): really refresh object

This fixes an error, where deleted attributes would not show up

Fixes #1155 (e1e0d8c)

- fix(cli): write binary data to stdout buffer (0733ec6)
- fix(cli): add missing args for project lists (c73e237)
- fix: docs changed using the consts (650b65c)
- fix: typo (9baa905)
- fix(api): add missing runner access\_level param (92669f2)

## Refactor

- refactor(tests): split functional tests ([61e43eb](#))

## Test

- test: ignore failing test for now ([4b4e253](#))
- test: add test\_project\_merge\_request\_approvals.py ([9f6335f](#))
- test(cli): add test for job artifacts download ([f4e7950](#))
- test(env): replace custom scripts with pytest and docker-compose ([79489c7](#))
- test: add unit tests for badges API ([2720b73](#))
- test: add unit tests for resource label events API ([e9a211c](#))

## Unknown

- Merge pull request #1273 from python-gitlab/chore/python3-9  
chore: officially support and test 3.9 ([48cb89b](#))
- Merge pull request #1230 from manuel-91/patch-1  
cli-usage readme: fixed term ([55c8c96](#))
- Merge pull request #1238 from atombrella/install\_version\_doc  
Updated supported Python versions in install doc ([d037a71](#))
- Merge pull request #1255 from bittner/docs/fix-external-links-tokens  
Use inline anonymous references for external links ([762f959](#))
- Merge pull request #1251 from hchouraria/patch-1  
docs(groups): add example for creating subgroups ([5ac309a](#))
- Merge pull request #1272 from python-gitlab/renovate/pin-dependencies  
chore(deps): pin dependency requests-toolbelt to ==0.9.1 ([b567522](#))
- Merge pull request #1233 from python-gitlab/renovate/requests-2.x  
chore(deps): update dependency requests to v2.25.1 ([78a02ce](#))
- Merge pull request #1243 from python-gitlab/renovate/docker-gitlab-gitlab-ce-13.x  
chore(deps): update gitlab/gitlab-ce docker tag to v13.8.1-ce.0 ([cc2dd0c](#))
- Merge pull request #1252 from python-gitlab/feat/multipart-uploads  
feat: support multipart uploads ([4f8d901](#))
- Merge pull request #1269 from nejch/fix/test-env  
chore(ci): bring test environment back to life ([fd179d4](#))
- Merge pull request #1250 from JacobHenner/feature/add-minimal-access  
feat: Add MINIMAL\_ACCESS constant ([fac0874](#))
- Merge pull request #1263 from ePirat/epirat-fix-get-label

fix(api): add missing GetMixin to ProjectLabelManager ([e61a0f2](#))

- Merge pull request #1200 from robinson96/feature/project\_merge\_request\_approval\_rules

Feature/project merge request approval rules ([6035ca8](#))

- Merge pull request #1213 from python-gitlab/fix/delete-attr

fix(base): really refresh object ([af965d4](#))

- Merge pull request #1212 from python-gitlab/chore/docs-edit-on-master

chore(docs): always edit the file directly on master ([ce8460f](#))

- Merge pull request #1211 from python-gitlab/docs/admin-project-owner

docs(issues): add admin, project owner hint ([e5b047d](#))

- Merge pull request #1214 from python-gitlab/docs/readme-gitlab-runner-test

docs(readme): also add hint to delete gitlab-runner-test ([53b4c2f](#))

- Merge pull request #1210 from python-gitlab/docs/projects-correct-fork

docs(projects): correct fork docs ([77ac8c3](#))

- Merge pull request #1202 from python-gitlab/fix/cli-binary-data

fix(cli): write binary data to stdout buffer ([3a38c6d](#))

- Merge pull request #1209 from python-gitlab/docs/cli-reference-page

docs(cli): add auto-generated CLI reference ([9054a3b](#))

- Merge pull request #1131 from valentingregoire/master

feat: added constants for search API ([8cb8040](#))

- Merge pull request #1205 from python-gitlab/refactor/split-functional-tests

refactor(tests): split functional tests ([68a4162](#))

- Merge pull request #1204 from python-gitlab/renovate/docker-python-3.x

chore(deps): update python docker tag to v3.9 ([2002098](#))

- Merge pull request #1203 from intostern/feat/bridge

Added support for pipeline bridges ([c303dab](#))

- Merge pull request #1206 from python-gitlab/fix/cli-project-list-args

fix(cli): add missing args for project lists ([be0afdd](#))

- Merge pull request #1178 from python-gitlab/test/cleanup-env

test(env): replace custom scripts with pytest and docker-compose ([266030a](#))

- Merge pull request #1177 from python-gitlab/renovate/pin-dependencies

chore(deps): pin dependencies ([0d89a6e](#))

- Merge pull request #1189 from python-gitlab/chore/test-hacking-dependency

chore(test): remove hacking dependencies ([d51042a](#))

- Merge pull request #1180 from Shkurupii/add-unittests-for-project-badges

test: add unit tests for badges API ([6bdb7f8](#))

- Merge pull request #1184 from python-gitlab/fix/runner-access-level

fix(api): add missing runner access\_level param ([a7c20ca](#))

- Merge pull request #1182 from jlpospisil/allow-mr-search-by-wip

Added MR wip filter param ([74c1e4f](#))

- Merge pull request #1181 from python-gitlab/feat/delete-user-identities

feat(api): add support for user identity provider deletion ([35f9cb8](#))

- Merge pull request #1179 from python-gitlab/chore/readthedocs-yml

chore(ci): add .readthedocs.yml ([49a0032](#))

- Merge pull request #1175 from python-gitlab/chore/noisy-renovate

chore(ci): reduce renovate PR noise ([8d662ab](#))

- Merge pull request #1174 from Shkurupii/add-unittests-for-resource-label-events

Add unit tests for resource label events API ([b1c2045](#))

- Merge pull request #1173 from python-gitlab/renovate/docker-gitlab-gitlab-ce-13.x

chore(deps): update gitlab/gitlab-ce docker tag to v13.3.5-ce.0 ([24fdbef](#))

- Merge pull request #1172 from python-gitlab/renovate/docker-gitlab-gitlab-ce-13.x

chore(deps): update gitlab/gitlab-ce docker tag to v13.3.4-ce.0 ([2a6801e](#))

- Merge pull request #1171 from python-gitlab/renovate/docker-gitlab-gitlab-ce-13.x

chore(deps): update gitlab/gitlab-ce docker tag to v13.3.3-ce.0 ([769367c](#))

## 7.9.32 v2.5.0 (2020-09-01)

### Chore

- chore: bump python-gitlab to 2.5.0 ([56fef01](#))
- chore(deps): update python docker tag to v3.8 ([a8070f2](#))
- chore(deps): update gitlab/gitlab-ce docker tag to v13.3.2-ce.0 ([9fd778b](#))
- chore(test): use pathlib for paths ([5a56b6b](#))
- chore(ci): pin gitlab-ce version for renovate ([cb79fb7](#))
- chore(env): add pre-commit and commit-msg hooks ([82070b2](#))
- chore(ci): use fixed black version ([9565684](#))
- chore: make latest black happy with existing code ([6961479](#))
- chore: update tools dir for latest black version ([f245ffb](#))
- chore: make latest black happy with existing code ([d299753](#))
- chore: update tools dir for latest black version ([c2806d8](#))
- chore: remove unnecessary import ([f337b7a](#))
- chore: make latest black happy with existing code ([4039c8c](#))
- chore: run unittest2pytest on all unit tests ([11383e7](#))
- chore: remove remnants of python2 imports ([402566a](#))



## Documentation

- docs(variables): add docs for instance-level variables ([ad4b87c](#))
- docs(packages): add examples for Packages API and cli usage ([a47dfcd](#))
- docs(api): add example for latest pipeline job artifacts ([d20f022](#))
- docs(cli): add examples for group-project list ([af86dcd](#))
- docs: additional project file delete example

Showing how to delete without having to pull the file ([9e94b75](#))

## Feature

- feat(api): add support for instance variables ([4492fc4](#))
- feat(api): add support for Packages API ([71495d1](#))
- feat(api): add endpoint for latest ref artifacts ([b7a07fc](#))
- feat: add support to resource milestone events

Fixes #1154 ([88f8cc7](#))

- feat: add share/unshare group with group ([7c6e541](#))

## Fix

- fix: wrong reconfirmation parameter when updating user's email

Since version 10.3 (and later), param to not send (re)confirmation when updating an user is `skip_reconfirmation` (and not `skip_confirmation`).

See:

- [https://gitlab.com/gitlab-org/gitlab-foss/-/merge\\_requests/15175?tab=](https://gitlab.com/gitlab-org/gitlab-foss/-/merge_requests/15175?tab=)
- <https://docs.gitlab.com/11.11/ee/api/users.html#user-modification>
- <https://docs.gitlab.com/ee/api/users.html#user-modification> ([b5c267e](#))
- fix: tests fail when using REUSE\_CONTAINER option

Fixes #1146 ([0078f89](#))

- fix: implement Gitlab's behavior change for owned=True ([9977799](#))

## Refactor

- refactor: turn objects module into a package ([da8af6f](#))
- refactor: rewrite unit tests for objects with responses ([204782a](#))
- refactor: split unit tests by GitLab API resources ([76b2cad](#))

## Test

- test(api): add tests for variables API ([66d108d](#))
- test(packages): add tests for Packages API ([7ea178b](#))
- test: add unit tests for resource milestone events API

Fixes [#1154](#) ([1317f4b](#))

## Unknown

- Merge pull request [#1170](#) from python-gitlab/chore/bump-to-2-5-0  
chore: bump python-gitlab to 2.5.0 ([784cba6](#))
- Merge pull request [#1168](#) from python-gitlab/renovate/docker-python-3.x  
chore(deps): update python docker tag to v3.8 ([cf32499](#))
- Merge pull request [#1163](#) from python-gitlab/feat/instance-variables-api  
Feat: add support for instance variables API ([723ca88](#))
- Merge pull request [#1167](#) from python-gitlab/renovate/docker-gitlab-gitlab-ce-13.x  
chore(deps): update gitlab/gitlab-ce docker tag to v13.3.2-ce.0 ([6e1ed68](#))
- Merge pull request [#1165](#) from DylannCordel/fix-user-email-reconfirmation  
fix: wrong reconfirmation parameter when updating user's email ([97e1dcc](#))
- Merge pull request [#1164](#) from nejch/master  
chore(ci): pin gitlab-ce version for renovate ([e6a9ba9](#))
- Merge pull request [#1162](#) from python-gitlab/chore/pre-commit-config  
chore(env): add pre-commit and commit-msg hooks ([97d8261](#))
- Merge pull request [#1161](#) from python-gitlab/chore/ci-fixed-black-version  
chore(ci): use fixed black version ([28aa17e](#))
- Merge pull request [#1157](#) from Shkurupii/issue-1154  
Add support to resource milestone events ([750f4ee](#))
- Merge pull request [#1159](#) from python-gitlab/feat/project-artifacts  
Feat: Project job artifacts for latest successful pipeline ([26f95f3](#))
- Merge pull request [#1160](#) from python-gitlab/feat/packages-api  
Feat: Add support for packages API ([0f42e32](#))
- Merge branch 'master' into issue-1154 ([fa899d7](#))
- Merge pull request [#1156](#) from python-gitlab/docs/group-projects-list-cli  
docs(cli): add examples for group-project list ([a038e95](#))
- Merge pull request [#1078](#) from python-gitlab/refactor/split-unit-tests  
Refactor: split unit tests by API resources ([a7e44a0](#))
- Merge pull request [#1147](#) from ericfrederich/fix-1146  
fix: tests fail when using REUSE\_CONTAINER option ([e2dc9ec](#))

- Merge pull request #1139 from sathieu/share\_group\_with\_group

feat: add share/unshare the group with a group ([cfa8097](#))

- Merge pull request #1152 from matthew-a-dunlap/doc-project-file-delete-example

docs: additional project file delete example ([5b92de8](#))

### 7.9.33 v2.4.0 (2020-07-09)

#### Chore

- chore: bump version to 2.4.0 ([1606310](#))
- chore: added constants for search API ([8ef53d6](#))

#### Feature

- feat: added NO\_ACCESS const

This constant is useful for cases where no access is granted, e.g. when creating a protected branch.

The NO\_ACCESS const corresponds to the definition in [https://docs.gitlab.com/ee/api/protected\\_branches.html](https://docs.gitlab.com/ee/api/protected_branches.html) ([dab4d0a](#))

#### Fix

- fix: add masked parameter for variables command ([b6339bf](#))
- fix: do not check if kwargs is none

Co-authored-by: Traian Nedelea <[tronlpoint0@pm.me](mailto:tronlpoint0@pm.me)> ([a349b90](#))

- fix: make query kwargs consistent between call in init and next ([72ffa01](#))
- fix: pass kwargs to subsequent queries in gitlab list ([1d011ac](#))
- fix(merge): parse arguments as query\_data ([878098b](#))

#### Unknown

- Merge pull request #1108 from stuartgunter/master

Added NO\_ACCESS const ([3a76d91](#))

- Merge pull request #1092 from aparcar/aparcar-patch-1

Update pipelines\_and\_jobs.rst ([12a40cc](#))

- Merge pull request #1124 from tyates-indeed/fix-1123

Pass kwargs to subsequent queries in GitlabList (fixes: #1123) ([424a8cb](#))

- Merge pull request #1127 from gervasek/master

Add masked parameter for project-variable and group-variable ([bfb5034](#))

- Merge pull request #1121 from ferhat-aram/fix/bad-merge-request-arg-parsing

fix(merge): parse arguments as query\_data ([1d82310](#))

### 7.9.34 v2.3.1 (2020-06-09)

#### Chore

- chore: bump version to 2.3.1 ([870e7ea](#))

#### Fix

- fix: disable default keyset pagination

Instead we set pagination to offset on the other paths ([e71fe16](#))

#### Unknown

- Merge pull request #1115 from python-gitlab/fix/keyset-pagination-revert  
Fix/keyset pagination revert ([3f585ad](#))

### 7.9.35 v2.3.0 (2020-06-08)

#### Chore

- chore: correctly render rst ([f674bf2](#))
- chore: bump to 2.3.0 ([01ff865](#))
- chore(ci): add codecov integration to Travis ([e230568](#))
- chore(test): remove outdated token test ([e6c9fe9](#))
- chore: bring commit signatures up to date with 12.10 ([dc382fe](#))
- chore: fix typo in docstring ([c20f5f1](#))
- chore: remove old builds-email service ([c60e2df](#))
- chore(services): update available service attributes ([7afc357](#))
- chore: use pytest for unit tests and coverage ([9787a40](#))

#### Ci

- ci: lint fixes ([930122b](#))
- ci: add a test for creating and triggering pipeline schedule ([9f04560](#))

## Documentation

- docs(remote\_mirrors): fix create command ([1bb4e42](#))
- docs(remote\_mirrors): fix create command ([bab91fe](#))
- docs(pipelines): simplify download

This uses a context instead of inventing your own stream handler which makes the code simpler and should be fine for most use cases.

Signed-off-by: Paul Spooren <[mail@aparcar.org](mailto:mail@aparcar.org)> ([9a068e0](#))

- docs: update authors ([ac0c84d](#))
- docs(readme): add codecov badge for master ([e21b2c5](#))
- docs(readme): update test docs ([6e2b1ec](#))

## Feature

- feat: add group runners api ([4943991](#))
- feat: add play command to project pipeline schedules

fix: remove version from setup

feat: add pipeline schedule play error exception

docs: add documentation for pipeline schedule play ([07b9988](#))

- feat(api): added support in the GroupManager to upload Group avatars ([28eb7ea](#))
- feat: allow an environment variable to specify config location

It can be useful (especially in scripts) to specify a configuration location via an environment variable. If the “PYTHON\_GITLAB\_CFG” environment variable is defined, treat its value as the path to a configuration file and include it in the set of default configuration locations. ([401e702](#))

- feat(services): add project service list API

Can be used to list available services It was introduced in GitLab 12.7 ([fc52221](#))

- feat(types): add **dir** to RESTObject to expose attributes ([cad134c](#))

## Fix

- fix: use keyset pagination by default for /projects > 50000

Workaround for <https://gitlab.com/gitlab-org/gitlab/-/issues/218504>. Remove this in 13.1 ([f86ef3b](#))

- fix(config): fix duplicate code

Fixes #1094 ([ee2df6f](#))

- fix(project): add missing project parameters ([ad8c67d](#))

## Test

- test: disable test until Gitlab 13.1 ([63ae77a](#))
- test(runners): add all runners unit tests ([127fa5a](#))
- test(cli): convert shell tests to pytest test cases ([c4ab4f5](#))

## Unknown

- Merge pull request #1112 from python-gitlab/fix/rst-renderer  
chore: correctly render rst ([1f7dbc8](#))
- Merge pull request #1111 from python-gitlab/chore/bump-version-2-3-0  
chore: bump to 2.3.0 ([a16ff3f](#))
- Merge pull request #1110 from python-gitlab/fix/keyset-pagination  
Fix keyset pagination in 13.0 ([f10dd38](#))
- Merge pull request #1102 from dotenorio/master  
Update doc for remote\_mirrors ([ef6181b](#))
- Merge branch 'master' of github.com:dotenorio/python-gitlab ([f5f4e12](#))
- Merge pull request #1089 from python-gitlab/feat/group-runners  
feat: add group runners api ([38e9fde](#))
- Merge pull request #1087 from python-gitlab/docs/update-authors  
docs: update authors ([89007c9](#))
- Merge pull request #1099 from python-gitlab/fix/duplicate-code  
fix(config): fix duplicate code ([242cf65](#))
- Merge pull request #1086 from python-gitlab/test/pytest-cli-tests  
test(cli): convert CLI shell tests to pytest test cases ([74b3ddc](#))
- Merge pull request #1085 from python-gitlab/chore/codecov-travis  
chore(ci): add codecov integration to Travis ([91c1c27](#))
- Merge pull request #1082 from python-gitlab/chore/signature-gpg-x509  
chore: bring commit signatures up to date with 12.10 ([5a75310](#))
- Merge pull request #1069 from zillow/feat/add-custom-pipeline-schedule-play  
feat: Add play command to project pipeline schedules ([9d66cb3](#))
- Merge pull request #1077 from Flor1an-dev/master  
feat(api): added support in the GroupManager to upload Group avatars ([7907e5a](#))
- Merge pull request #1075 from python-gitlab/feat/available-services  
feat(services): add project service list API ([dad505c](#))
- Merge pull request #1074 from jeremycline/environment-variable  
feat: Allow an environment variable to specify config location ([0c3b717](#))
- Merge pull request #1073 from python-gitlab/docs/readme-test-docs

docs(readme): update test docs ([70cefe4](#))

- Merge pull request #1072 from spyoungtech/feat/restobject-dir

feat(types): add **dir** to RESTObject to expose attributes ([c7c431a](#))

- Merge pull request #1066 from nejch/chore/pytest-for-unit-tests

chore: use pytest to run unit tests and coverage ([efc6182](#))

- Merge pull request #1067 from python-gitlab/fix/missing-project-attributes

fix(project): add missing project parameters ([29fd95e](#))

### 7.9.36 v2.2.0 (2020-04-07)

#### Chore

- chore: bump to 2.2.0 ([22d4b46](#))
- chore(group): update group\_manager attributes (#1062)
- chore(group): update group\_manager attributes

Co-Authored-By: Nejc Habjan <hab.nejc@gmail.com> ([fa34f5e](#))

- chore: rename ExportMixin to DownloadMixin ([847da60](#))
- chore(mixins): factor out export download into ExportMixin ([6ce5d1f](#))
- chore: use raise..from for chained exceptions (#939) ([79fef26](#))
- chore: fix typo in allow\_failures ([265bbdd](#))
- chore: pass environment variables in tox ([e06d33c](#))
- chore: improve and document testing against different images ([98d3f77](#))
- chore: remove references to python2 in test env ([6e80723](#))
- chore: clean up for black and flake8 ([4fede5d](#))
- chore: flatten test\_import\_github ([b8ea96c](#))
- chore: move test\_import\_github into TestProjectImport ([a881fb7](#))

#### Documentation

- docs: add docs for Group Import/Export API ([8c3d744](#))
- docs: fix comment of prev\_page()

Co-Authored-By: Nejc Habjan <hab.nejc@gmail.com> ([b066b41](#))

- docs: fix comment of prev\_page()

Co-Authored-By: Nejc Habjan <hab.nejc@gmail.com> ([ac6b2da](#))

- docs: fix comment of prev\_page() ([7993c93](#))

## Feature

- feat(api): add support for remote mirrors API (#1056) ([4cfaa2f](#))
- feat(api): add support for Gitlab Deploy Token API ([01de524](#))
- feat(api): add support for Group Import/Export API (#1037) ([6cb9d92](#))
- feat: add support for commit GPG signature API ([da7a809](#))
- feat: add create from template args to ProjectManager

This commit adds the v4 Create project attributes necessary to create a project from a project, instance, or group level template as documented in <https://docs.gitlab.com/ee/api/projects.html#create-project> ([f493b73](#))

## Fix

- fix(types): do not split single value string in ListAttribute ([a26e585](#))
- fix: add missing import\_project param ([9b16614](#))

## Test

- test(api): add tests for group export/import API ([e7b2d6c](#))
- test(types): reproduce get\_for\_api splitting strings (#1057) ([babd298](#))
- test: create separate module for commit tests ([8c03771](#))
- test: move mocks to top of module ([0bff713](#))
- test: add unit tests for Project Import ([f7aad5f](#))
- test: add unit tests for Project Export ([600dc86](#))
- test: prepare base project test class for more tests ([915587f](#))

## Unknown

- Merge pull request #1059 from python-gitlab/fix/raise-from  
chore: use raise..from for chained exceptions (#939) ([6749859](#))
- Merge pull request #1052 from machine424/deploy-tokens-support  
feat(api): add support for Gitlab Deploy Token API ([5979750](#))
- Merge pull request #1064 from python-gitlab/feat/project-remote-mirrors  
feat(api): add support for remote mirrors API (#1056) ([3396aa5](#))
- Merge pull request #1063 from python-gitlab/feat/group-import-export  
Feat: support for group import/export API ([c161852](#))
- Merge pull request #1058 from python-gitlab/fix/listattribute-get-api-splits-string  
Fix: ListAttribute get\_for\_api() splits strings ([50fcd12](#))
- Merge pull request #1053 from lassimus/master  
feat: add create from template args to ProjectManager ([c5904c4](#))
- Merge pull request #1054 from nejch/chore/cleanup-test-env



chore: improve test environment for upcoming features ([8173021](#))

- Merge pull request #1055 from nejch/feat/commit-gpg-signature

feat: add support for commit GPG signature ([1b8e748](#))

- Merge pull request #1049 from donhui/typo-fix
- docs: fix comment of prev\_page() ([82deb7d](#))
- Merge pull request #1040 from nejch/test/project-export-import

test: update tests and params for project export/import ([4ffaf1d](#))

### 7.9.37 v2.1.2 (2020-03-09)

#### Chore

- chore: bump version to 2.1.2 ([ad7e2bf](#))

#### Unknown

- Merge pull request #1045 from python-gitlab/revert-1003-feat/all-keyset-pagination

Revert “feat: use keyset pagination by default for all=True” ([6d941bd](#))

- Revert “feat: use keyset pagination by default for all=True” ([6f843b6](#))

### 7.9.38 v2.1.1 (2020-03-09)

#### Chore

- chore: bump version to 2.1.1 ([6c5458a](#))
- chore(user): update user attributes to 12.8 ([666f880](#))

#### Fix

- fix(docs): additional project statistics example ([5ae5a06](#))

#### Unknown

- Merge pull request #1043 from python-gitlab/chore/update-user-attributes

chore(user): update user attributes to 12.8 ([8c44bb6](#))

- Merge pull request #1042 from khuedoan98/patch-1

fix(docs): additional project statistics example ([be5b15e](#))

### 7.9.39 v2.1.0 (2020-03-08)

#### Chore

- chore: bump version to 2.1.0 ([47cb58c](#))
- chore: fix broken requests links

Another case of the double slash rewrite. ([b392c21](#))

- chore: ensure developers use same gitlab image as Travis ([fab17fc](#))

#### Documentation

- docs: add reference for REQUESTS\_CA\_BUNDLE ([37e8d5d](#))
- docs(pagination): clear up pagination docs

Co-Authored-By: Mitar <mitar.git@tnode.com> ([1609824](#))

#### Feature

- feat(api): add support for GitLab OAuth Applications API ([4e12356](#))
- feat: add support for user memberships API (#1009) ([c313c2b](#))
- feat: add support for commit revert API (#991) ([5298964](#))
- feat: add capability to control GitLab features per project or group ([7f192b4](#))

#### Fix

- fix(projects): correct copy-paste error ([adc9101](#))
- fix: do not require empty data dict for create() ([99d959f](#))
- fix: remove trailing slashes from base URL (#913) ([2e396e4](#))
- fix(docs): fix typo in user memberships example ([33889bc](#))
- fix: return response with commit data ([b77b945](#))
- fix(objects): add default name data and use http post

Updating approvers new api needs a POST call. Also It needs a name of the new rule, defaulting this to 'name'. ([70c0cfb](#))

- fix: remove null values from features POST data, because it fails with HTTP 500 ([1ec1816](#))

## Performance

- perf: prepare environment when gitlab is reconfigured ([3834d9c](#))

## Style

- style: fix black violations ([ad3e833](#))

## Test

- test: add unit tests for base URLs with trailing slashes ([32844c7](#))
- test: remove duplicate resp\_get\_project ([cb43695](#))
- test: use lazy object in unit tests ([31c6562](#))
- test: add unit tests for revert commit API ([d7a3066](#))

## Unknown

- Merge pull request #1039 from python-gitlab/fix/set-approvers  
Fix/set approvers ([481bd4f](#))
- Merge pull request #1038 from nejch/fix/allow-empty-create-data  
Fix: do not require empty data dict for create() ([ca37d23](#))
- Merge pull request #1034 from filipowm/feat/api-oauth-applications  
feat(api): add support for GitLab OAuth Applications using Applications API ([e5afb55](#))
- Merge pull request #1032 from nejch/docs/requests-ca-bundle  
docs: add reference to REQUESTS\_CA\_BUNDLE usage ([fbcc820](#))
- Merge pull request #1003 from python-gitlab/feat/all-keyset-pagination  
feat: use keyset pagination by default for all=True ([3aa9873](#))
- Merge pull request #1022 from nejch/chore/ensure-latest-image  
chore: ensure developers use same gitlab image as CI ([745bdf7](#))
- Merge pull request #1023 from nejch/perf/wait-gitlab-reconfigure  
perf: wait for gitlab to reconfigure instead of using hardcoded sleep ([2b3871d](#))
- Merge pull request #1026 from nejch/feat/user-memberships  
feat: add support for user memberships API (#1009) ([f071390](#))
- Merge pull request #1027 from nejch/fix/base-url-trailing-slash  
Fix: remove trailing slash in base URL ([292dfff](#))
- Merge pull request #1020 from nejch/feat/revert-commit-api  
feat: add support for commit revert API (#991) ([e8f0921](#))
- Merge pull request #1005 from charlesfayal/fix\_set\_approvers  
change path for set\_approvers to new api, with defaulted rule\_type an... ([19242c3](#))

- Merge pull request #1008 from filipowm/feature/feature-flags-additional-config

Add capability to control GitLab features per project or group ([066fc9b](#))

## 7.9.40 v2.0.1 (2020-02-05)

### Chore

- chore: revert to 2.0.1

I've misread the tag ([272db26](#))

- chore: bump to 2.1.0

There are a few more features in there ([a6c0660](#))

- chore: bump version to 2.0.1 ([8287a0d](#))
- chore(user): update user attributes

This also workarounds an GitLab issue, where `private_profile`, would reset to false if not supplied ([27375f6](#))

### Documentation

- docs(auth): remove email/password auth ([c9329bb](#))

### Feature

- feat: use keyset pagination by default for `all=True` ([99b4484](#))

### Fix

- fix(docs): update to new set approvers call for # of approvers

to set the # of approvers for an MR you need to use the same function as for setting the approvers id. ([8e0c526](#))

- fix(objects): update set\_approvers function call

Added a miss paramter update to the set\_approvers function ([65ecadc](#))

- fix(docs and tests): update docs and tests for set\_approvers

Updated the docs with the new set\_approvers arguments, and updated tests with the arg as well. ([2cf12c7](#))

- fix(objects): update to new gitlab api for path, and args

Updated the gitlab path for set\_approvers to `approvers_rules`, added default arg for rule type, and added arg for # of approvals required. ([e512cdd](#))

## Unknown

- Merge pull request #1007 from python-gitlab/chore/user-update

chore(user): update user attributes ([f6d9858](#))

- Merge pull request #1000 from matusf/update-auth-docs

Update auth docs ([7843ace](#))

## 7.9.41 v2.0.0 (2020-01-26)

### Chore

- chore: build\_sphinx needs sphinx >= 1.7.6

Stepping thru Sphinx versions from 1.6.5 to 1.7.5 build\_sphinx fails. Once Sphinx == 1.7.6 build\_sphinx finished. ([528dfab](#))

- chore: enforce python version requirements ([70176db](#))
- chore: bump to 2.0.0

Dropping support for legacy python requires a new major version ([c817dcc](#))

- chore: drop legacy python tests

Support dropped for: 2.7, 3.4, 3.5 ([af8679a](#))

- chore: add PyYaml as extra require ([7ecd518](#))
- chore: bump minimum required requests version

for security reasons ([3f78aa3](#))

### Documentation

- docs: fix snippet get in project ([3a4ff2f](#))
- docs(projects): add raw file download docs

Fixes #969 ([939e9d3](#))

### Feature

- feat: add global order\_by option to ease pagination ([d187925](#))
- feat: support keyset pagination globally ([0b71ba4](#))
- feat: add appearance API ([4c4ac5c](#))
- feat: add autocompletion support ([973cb8b](#))

## Fix

- fix(projects): adjust snippets to match the API ([e104e21](#))

## Refactor

- refactor: support new list filters

This is most likely only useful for the CLI ([bded2de](#))

- refactor: remove six dependency ([9fb4645](#))

## Test

- test: adjust functional tests for project snippets ([ac0ea91](#))
- test: add project snippet tests ([0952c55](#))

## Unknown

- Merge pull request #1001 from python-gitlab/feat/keyset-pagination

Feat/keyset pagination ([df485a9](#))

- Merge pull request #996 from python-gitlab/feat/appearance

feat: add appearance API ([7fd3226](#))

- Merge pull request #988 from jgroom33/patch-3

docs: fix snippet get in project ([afdc43f](#))

- Merge pull request #984 from derekschrock/patch-1

chore: build\_sphinx needs sphinx >= 1.7.6 ([fc2ed13](#))

- Merge pull request #982 from python-gitlab/chore/version-requirements

chore: enforce python version requirements ([83fcd1b](#))

- Merge pull request #980 from python-gitlab/refactor/cleanup-upgrade

Refactor/cleanup upgrade ([5fa0e16](#))

- Merge pull request #979 from python-gitlab/fix/project-snippets

Fix/project snippets ([5a10eb3](#))

- Merge pull request #941 from mchlumsky/feat/autocompletion

feat: add autocompletion support ([ec6e04c](#))

## 7.9.42 v1.15.0 (2019-12-16)

### Chore

- chore: bump version to 1.15.0 (2a01326)
- chore(ci): use correct crane ci (18913dd)

### Documentation

- docs(projects): fix file deletion docs

The function `file.delete()` requires `branch` argument in addition to `commit_message`. (1c4f1c4)

- docs: added docs for statistics (8c84cbf)

### Feature

- feat: allow cfg timeout to be overridden via kwargs

On startup, the `timeout` parameter is loaded from config and stored on the base gitlab object instance. This instance parameter is used as the timeout for all API requests (it's passed into the `session` object when making HTTP calls).

This change allows any API method to specify a `timeout` argument to `**kwargs` that will override the global timeout value. This was somewhat needed / helpful for the `import_github` method.

I have also updated the docs accordingly. (e9a8289)

- feat: add support for `/import/github`

Addresses python-gitlab/python-gitlab#952

This adds a method to the `ProjectManager` called `import_github`, which maps to the `/import/github` API endpoint. Calling `import_github` will trigger an import operation from `<repo_id>` into `<target_namespace>`, using `<personal_access_token>` to authenticate against github. In practice a gitlab server may take many 10's of seconds to respond to this API call, so we also take the liberty of increasing the default timeout (only for this method invocation).

Unfortunately since `import` is a protected keyword in python, I was unable to follow the endpoint structure with the manager namespace. I'm open to suggestions on a more sensible interface.

I'm successfully using this addition to batch-import hundreds of github repositories into gitlab. (aa4d41b)

- feat: nicer stacktrace (697cda2)
- feat: retry transient HTTP errors

Fixes #970 (59fe271)

- feat: access project's issues statistics

Fixes #966 (482e57b)

- feat: adding project stats

Fixes #967 (db0b00a)

- feat: add `variable_type/protected` to projects ci variables

This adds the ci variables types and protected flag for create/update requests.

See [https://docs.gitlab.com/ee/api/project\\_level\\_variables.html#create-variable](https://docs.gitlab.com/ee/api/project_level_variables.html#create-variable) (4724c50)

- feat: add `variable_type` to groups ci variables

This adds the ci variables types for create/update requests.

See [https://docs.gitlab.com/ee/api/group\\_level\\_variables.html#create-variable](https://docs.gitlab.com/ee/api/group_level_variables.html#create-variable) (0986c93)

## Fix

- fix: ignore all parameter, when as\_list=True

Closes #962 (137d72b)

## Style

- style: format with the latest black version (06a8050)

## Test

- test: added tests for statistics (8760efc)
- test: test that all is ignored, when as\_list=False (b5e88f3)

## Unknown

- Merge pull request #959 from andrew-littlebits/feat/import-github

feat: add support for /import/github (97e1fca)

- Merge pull request #973 from mitar/patch-1

Nicer stacktrace (61eaaad2)

- Merge pull request #971 from jooola/ci\_vars\_type

feat: add more options for project/group ci variables manipulation (938fc0a)

- Merge pull request #974 from python-gitlab/docs/file-deletion-docs

docs(projects): fix file deletion docs (59af4e4)

- Merge pull request #968 from mitar/stats

Stats (62b0b62)

- Merge pull request #972 from mitar/http-retry

Retry transient HTTP errors (36bbd37)

- Merge pull request #963 from python-gitlab/fix/as\_list

Fix/as list (3e2d694)



### 7.9.43 v1.14.0 (2019-12-07)

#### Chore

- chore: bump version to 1.14.0 (164fa4f)
- chore(ci): switch to crane docker image (#944) (e0066b6)

#### Documentation

- docs(readme): fix Docker image reference

v1.8.0 is not available.

```
Unable to find image &#39;registry.gitlab.com/python-gitlab/python-gitlab:v1.8.0&#39;;
↳ locally
docker: Error response from daemon: manifest for registry.gitlab.com/python-gitlab/
↳ python-gitlab:v1.8.0 not found: manifest unknown: manifest unknown.
``` ([`b9a40d8`](https://github.com/python-gitlab/python-gitlab/commit/
↳ b9a40d822bcff630a4c92c395c134f8c002ed1cb))

* docs(snippets): fix snippet docs

Fixes #954 ([`bbaa754`](https://github.com/python-gitlab/python-gitlab/commit/
↳ bbaa754673c4a0bffece482fe33e4875ddadc2dc))

* docs: fix typo ([`d9871b1`](https://github.com/python-gitlab/python-gitlab/commit/
↳ d9871b148c7729c9e401f43ff6293a5e65ce1838))

* docs: add project and group cluster examples ([`d15801d`](https://github.com/python-
↳ gitlab/python-gitlab/commit/d15801d7e7742a43ad9517f0ac13b6dba24c6283))

* docs(changelog): add notice for release-notes on Github (#938) ([`de98e57`](https://
↳ github.com/python-gitlab/python-gitlab/commit/
↳ de98e572b003ee4cf2c1ef770a692f442c216247))

### Feature

* feat: add audit endpoint ([`2534020`](https://github.com/python-gitlab/python-gitlab/
↳ commit/2534020b1832f28339ef466d6dd3edc21a521260))

* feat: add project and group clusters ([`ebd053e`](https://github.com/python-gitlab/
↳ python-gitlab/commit/ebd053e7bb695124c8117a95eab0072db185ddf9))

* feat: add support for include_subgroups filter ([`adbcd83`](https://github.com/python-
↳ gitlab/python-gitlab/commit/adbcd83fa172af2f3929ba063a0e780395b102d8))

### Fix

* fix(project-fork): copy create fix from ProjectPipelineManager ([`516307f`](https://
↳ github.com/python-gitlab/python-gitlab/commit/
↳ 516307f1cc9e140c7d85d0ed0c419679b314f80b))
```

(continues on next page)

(continued from previous page)

```

* fix(project-fork): correct path computation for project-fork list ([`44a7c27`](https://
↳github.com/python-gitlab/python-gitlab/commit/
↳44a7c2788dd19c1fe73d7449bd7e1370816fd36d))

* fix(labels): ensure label.save() works

Otherwise, we get:
  File &#34;gitlabracadabra/mixins/labels.py&#34;, line 67, in _process_labels
    current_label.save()
  File &#34;gitlab/exceptions.py&#34;, line 267, in wrapped_f
    return f(*args, **kwargs)
  File &#34;gitlab/v4/objects.py&#34;, line 896, in save
    self._update_attrs(server_data)
  File &#34;gitlab/base.py&#34;, line 131, in _update_attrs
    self.__dict__[&#34;_attrs&#34;].update(new_attrs)
TypeError: &#39;NoneType&#39; object is not iterable

Because server_data is None. ([`727f536`](https://github.com/python-gitlab/python-gitlab/
↳commit/727f53619dba47f0ab770e4e06f1cb774e14f819))

* fix: added missing attributes for project approvals

Reference: https://docs.gitlab.com/ee/api/merge_request_approvals.html#change-
↳configuration

Missing attributes:
* merge_requests_author_approval
* merge_requests_disable_committers_approval ([`460ed63`](https://github.com/python-
↳gitlab/python-gitlab/commit/460ed63c3dc4f966d6aae1415fdad6de125c6327))

### Unknown

* Merge pull request #949 from idanbensha/add_audit_events

feat: add audit endpoint ([`98e1b0a`](https://github.com/python-gitlab/python-gitlab/
↳commit/98e1b0ae77a627d21ce971ee4df813e1955f69a0))

* Merge pull request #958 from vvv/fix-docker-ref

README.rst: fix the upstream Docker image reference ([`f6f5178`](https://github.com/
↳python-gitlab/python-gitlab/commit/f6f5178c1dc7aeb3fdbee19b1768e30b2be4f4f4))

* Merge pull request #955 from python-gitlab/fix/snippet-docs

docs(snippets): fix snippet docs ([`9961aaa`](https://github.com/python-gitlab/python-
↳gitlab/commit/9961aaa1508e08a567c8c66cb194385788b8113e))

* Merge pull request #953 from bmwiedemann/master

Fix doc typo ([`267a9a1`](https://github.com/python-gitlab/python-gitlab/commit/
↳267a9a151ba9f2338f50fbb118513807ebce9704))

```

(continues on next page)

(continued from previous page)

```

* Merge pull request #947 from lundbird/master

docs: add project and group cluster examples ([`e4cad49`](https://github.com/python-
↳ gitlab/python-gitlab/commit/e4cad490b9cd16aa20ea84bb4bd24a6d25b19411))

* Merge pull request #946 from lundbird/master

feat: add project and group clusters ([`da557c9`](https://github.com/python-gitlab/
↳ python-gitlab/commit/da557c931fa6c6d50c373fc022d88acf1431c24a))

* Merge pull request #943 from choyrim/942-project-fork-list-404

#942: fix up path computation for project-fork list ([`ecad2c8`](https://github.com/
↳ python-gitlab/python-gitlab/commit/ecad2c83635c5e5f7003f61502391446ebc631c9))

* Merge pull request #937 from sathieu/fix_labels_save

fix(labels): ensure label.save() works ([`c937338`](https://github.com/python-gitlab/
↳ python-gitlab/commit/c937338b0119b08b358f97b4716c56777ee7bb80))

* Merge pull request #934 from tymonx/fix-missing-attribute-for-project-approvals

Added missing attributes for project approvals ([`9608886`](https://github.com/python-
↳ gitlab/python-gitlab/commit/960888628617beae75392dcdcb6ef5a66abd976d))

* Merge pull request #929 from SVLay/docs/pipeline-variables

docs(pipelines_and_jobs): add pipeline custom variables usage example
↳ ([`4efa6e6`](https://github.com/python-gitlab/python-gitlab/commit/
↳ 4efa6e6e5b9b57a3c4eda9ef20a4194b384055dc))

* Merge pull request #932 from ConorNevin/master

Add support for include_subgroups filter ([`1f18230`](https://github.com/python-gitlab/
↳ python-gitlab/commit/1f182302c206502f5202d1707fef69adf527fea7))

## v1.13.0 (2019-11-02)

### Chore

* chore: bump version to 1.13.0 ([`d0750bc`](https://github.com/python-gitlab/python-
↳ gitlab/commit/d0750bc01ed12952a4d259a13b3917fa404fd435))

* chore(setup): we support 3.8 (#924)

* chore(setup): we support 3.8

* style: format with black ([`6048175`](https://github.com/python-gitlab/python-gitlab/
↳ commit/6048175ef2c21fda298754e9b07515b0a56d66bd))

* chore(ci): update latest docker image for every tag ([`01cbc7a`](https://github.com/

```

(continues on next page)

(continued from previous page)

```

↪python-gitlab/python-gitlab/commit/01cbc7ad04a875bea93a08c0ce563ab5b4fe896b))

* chore(dist): add test data

Closes #907 ([`3133ed7`](https://github.com/python-gitlab/python-gitlab/commit/
↪3133ed7d1df6f49de380b35331bbcc67b585a61b))

### Documentation

* docs(pipelines_and_jobs): add pipeline custom variables usage example
↪([`b275eb0`](https://github.com/python-gitlab/python-gitlab/commit/
↪b275eb03c5954ca24f249efad8125d1eacadd3ac))

* docs: projects get requires id

Also, add an example value for project_id to the other projects.get()
example. ([`5bd8947`](https://github.com/python-gitlab/python-gitlab/commit/
↪5bd8947bd16398aed218f07458aef72e67f2d130))

* docs(project): fix group project example

GroupManager.search is removed since 9a66d78, use list(search=&#39;keyword&#39;) instead
↪([`e680943`](https://github.com/python-gitlab/python-gitlab/commit/
↪e68094317ff6905049e464a59731fe4ab23521de))

### Feature

* feat: add users activate, deactivate functionality

These were introduced in GitLab 12.4 ([`32ad669`](https://github.com/python-gitlab/
↪python-gitlab/commit/32ad66921e408f6553b9d60b6b4833ed3180f549))

* feat: send python-gitlab version as user-agent ([`c22d49d`](https://github.com/python-
↪gitlab/python-gitlab/commit/c22d49d084d1e03426cfab0d394330f8ab4bd85a))

* feat: add deployment creation

Added in GitLab 12.4

Fixes #917 ([`ca256a0`](https://github.com/python-gitlab/python-gitlab/commit/
↪ca256a07a2cdaf77a5c20e307d334b82fd0fe861))

* feat(test): unused unittest2, type -&gt; isinstance ([`33b1801`](https://github.com/
↪python-gitlab/python-gitlab/commit/33b180120f30515d0f76fcf635cb8c76045b1b42))

* feat(doc): remove refs to api v3 in docs ([`6beea9`](https://github.com/python-gitlab/
↪python-gitlab/commit/6beea993f8931d6b7fe682f1afed2bd4c8a4b73))

* feat(auth): remove deprecated session auth ([`b751cdf`](https://github.com/python-
↪gitlab/python-gitlab/commit/b751cdf424454d3859f3f038b58212e441faafaf))

### Fix

```

(continues on next page)

(continued from previous page)

```
* fix(projects): support `approval_rules` endpoint for projects
```

The `approvers` API endpoint is deprecated [1]. GitLab instead uses the `approval\_rules` API endpoint to modify approval settings for merge requests. This adds the functionality for project-level merge request approval settings.

Note that there does not exist an endpoint to `&#39;get&#39;`; a single approval rule at this moment - only `&#39;list&#39;`;

[1] [https://docs.gitlab.com/ee/api/merge\\_request\\_approvals.html](https://docs.gitlab.com/ee/api/merge_request_approvals.html) ([`2cef2bb`](<https://github.com/python-gitlab/python-gitlab/commit/2cef2bb40b1f37b97bb2ee9894ab3b9970cef231>))

```
### Test
```

```
* test(projects): support `approval_rules` endpoint for projects ([`94bac44`](https://github.com/python-gitlab/python-gitlab/commit/94bac4494353e4f597df0251f0547513c011e6de))
```

```
* test: remove warning about open files from test_todo()
```

When running unittests python warns that the json file from test\_todo() was still open. Use with to open, read, and create encoded json data that is used by resp\_get\_todo(). ([`d6419aa`](<https://github.com/python-gitlab/python-gitlab/commit/d6419aa86d6ad385e15d685bf47242bb6c67653e>))

```
### Unknown
```

```
* Merge pull request #931 from python-gitlab/choree/1-13-0
```

chore: bump version to 1.13.0 ([`f39c68f`](<https://github.com/python-gitlab/python-gitlab/commit/f39c68fd0b180ba72dd11e3cbad932d16d4bb484>))

```
* Merge pull request #919 from appian/project-approval-rules
```

fix(projects): support `approval\_rules` endpoint for projects ([`fddc25a`](<https://github.com/python-gitlab/python-gitlab/commit/fddc25adac16a74f61d81871f9ae13c0227d92d6>))

```
* Merge pull request #923 from python-gitlab/feat/users-activate-deactivate
```

feat: add users activate, deactivate functionality ([`912e16b`](<https://github.com/python-gitlab/python-gitlab/commit/912e16b95611715b4df3fae019687f7616af51c1>))

```
* Merge pull request #922 from python-gitlab/chore/latest-docker-image
```

chore(ci): update latest docker image for every tag ([`ac2266b`](<https://github.com/python-gitlab/python-gitlab/commit/ac2266b66553cec11740bd5246e23d649606b5ef>))

```
* Merge pull request #921 from python-gitlab/feat/python-gitlab-agent
```

(continues on next page)

(continued from previous page)

```

feat: send python-gitlab version as user-agent ([`8cb5488`](https://github.com/python-
↳ gitlab/python-gitlab/commit/8cb5488142ca7fc7563fac65b434b672a14369fc))

* Merge pull request #920 from python-gitlab/feat/deployment-create

feat: add deployment creation ([`dad6805`](https://github.com/python-gitlab/python-
↳ gitlab/commit/dad68050c1269519f35dcdb29dc94a03f47532c5))

* Merge pull request #916 from python-gitlab/chore/add-test-to-dist

chore(dist): add test data ([`e790b1e`](https://github.com/python-gitlab/python-gitlab/
↳ commit/e790b1ec40ed690152776a87c15e7f7d5d3b9136))

* Merge pull request #914 from terminalmage/issue913

Remove inaccurate projects.get() example ([`d2c9cee`](https://github.com/python-gitlab/
↳ python-gitlab/commit/d2c9ceec5d6473f286e00963252abbcf1a2a17c))

* Merge pull request #911 from xdavidwu/fix-project-doc

docs(project): fix group project example ([`d853a76`](https://github.com/python-gitlab/
↳ python-gitlab/commit/d853a767ac8835615e0fded3087f55ca8594c1ed))

* Merge pull request #906 from jouve/test-cleanup

unused unittest2, type -&gt; isinstance ([`42a1ba6`](https://github.com/python-gitlab/
↳ python-gitlab/commit/42a1ba6be175a9838c589cb1e40636b3910db505))

* Merge pull request #904 from jouve/remove-cred-auth

remove deprecated session auth ([`67a9c1f`](https://github.com/python-gitlab/python-
↳ gitlab/commit/67a9c1f1c62393b02919d25bcc98c3683d92576a))

* Merge pull request #908 from derekschrock/todo-units-test

Remove warning about open files from test_todo() ([`ff808ee`](https://github.com/python-
↳ gitlab/python-gitlab/commit/ff808ee94a73d65802a21ff1350090885d4befd5))

* Merge pull request #905 from jouve/doc-v3

remove references to api v3 in docs ([`92ba028`](https://github.com/python-gitlab/python-
↳ gitlab/commit/92ba0283b63e562e181061252787e0e46da83a29))

## v1.12.1 (2019-10-07)

### Fix

* fix: fix not working without auth ([`03b7b5b`](https://github.com/python-gitlab/python-
↳ gitlab/commit/03b7b5b07e1fd2872e8968dd6c29bc3161c6c43a))

```

(continues on next page)

(continued from previous page)

```

### Unknown

* Merge pull request #901 from python-gitlab/fix/non-auth

fix: fix not working without auth ([`f4b2927`](https://github.com/python-gitlab/python-
gitlab/commit/f4b29278771e48320e2da4bacc4544d263d1754c))

## v1.12.0 (2019-10-06)

### Chore

* chore: bump to 1.12.0 ([`4648128`](https://github.com/python-gitlab/python-gitlab/
commit/46481283a9985aeb07fe686ec4a34e4a1219b66))

* chore(ci): build test images on tag ([`0256c67`](https://github.com/python-gitlab/
python-gitlab/commit/0256c678ea9593c6371ffff60663f83c423ca872))

### Documentation

* docs(project): add submodule docs ([`b5969a2`](https://github.com/python-gitlab/python-
gitlab/commit/b5969a2dcea77fa608cc29be7a5f39062edd3846))

* docs(projects): add note about project list

Fixes #795 ([`44407c0`](https://github.com/python-gitlab/python-gitlab/commit/
44407c0f59b9602b17cfb93b5e1fa37a84064766))

* docs(repository-tags): fix typo

Closes #879 ([`3024c5d`](https://github.com/python-gitlab/python-gitlab/commit/
3024c5dc8794382e281b83a8266be7061069e83e))

* docs(todo): correct todo docs ([`d64edcb`](https://github.com/python-gitlab/python-
gitlab/commit/d64edcb4851ea62e72e3808daf7d9b4fdaaf548b))

### Feature

* feat(project): implement update_submodule ([`4d1e377`](https://github.com/python-
gitlab/python-gitlab/commit/4d1e3774706f336e87ebe70e1b373ddb37f34b45))

* feat(ci): improve fonctionnal tests ([`eefceac`](https://github.com/python-gitlab/
python-gitlab/commit/eefceace2c2094ef41d3da2bf3c46a58a450dcba))

* feat(project): add file blame api

https://docs.gitlab.com/ee/api/repository_files.html#get-file-blame-from-repository
([`f5b4a11`](https://github.com/python-gitlab/python-gitlab/commit/
f5b4a113a298d33cb72f80c94d85bdfec3c4e149))

* feat: add support for job token

```

(continues on next page)

(continued from previous page)

```

See https://docs.gitlab.com/ee/api/jobs.html#get-job-artifacts for usage.
→([`cef3aa5`](https://github.com/python-gitlab/python-gitlab/commit/
→cef3aa51a6928338c6755c3e6de78605fae8e59e))

* feat(user): add status api ([`62c9fe6`](https://github.com/python-gitlab/
→commit/62c9fe63a47ddde2792a4a5e9cd1c7aa48661492))

### Fix

* fix(cli): fix cli command user-project list ([`c17d7ce`](https://github.com/python-
→gitlab/python-gitlab/commit/c17d7ce14f79c21037808894d8c7ba1117779130))

* fix(labels): don't mangle label name on update ([`1fb6f73`](https://github.com/
→python-gitlab/python-gitlab/commit/1fb6f73f4d501c2b6c86c863d40481e1d7a707fe))

* fix(todo): mark_all_as_done doesn't return anything ([`5066e68`](https://github.
→com/python-gitlab/python-gitlab/commit/5066e68b398039beb5e1966baled7684d97a8f74))

### Refactor

* refactor: remove obsolete test image

Follow up of #896 ([`a14c02e`](https://github.com/python-gitlab/python-gitlab/commit/
→a14c02ef85bd4d273b8c7f0f6bd07680c91955fa))

* refactor: remove unused code, simplify string format ([`c7ff676`](https://github.com/
→python-gitlab/python-gitlab/commit/c7ff676c11303a00da3a570bf2893717d0391f20))

### Style

* style: format with black ([`fef085d`](https://github.com/python-gitlab/python-gitlab/
→commit/fef085dca35d6b60013d53a3723b4cbf121ab2ae))

### Test

* test(submodules): correct test method ([`e59356f`](https://github.com/python-gitlab/
→python-gitlab/commit/e59356f6f90d5b01abbe54153441b6093834aa11))

* test(func): disable commit test

GitLab seems to be randomly failing here ([`c9c76a2`](https://github.com/python-gitlab/
→python-gitlab/commit/c9c76a257d2ed3b394f499253d890c2dd9a01e24))

* test(todo): add unittests ([`7715567`](https://github.com/python-gitlab/python-gitlab/
→commit/77155678a5d8dbbf11d00f3586307694042d3227))

* test(status): add user status test ([`fec4f9c`](https://github.com/python-gitlab/
→python-gitlab/commit/fec4f9c23b8ba33bb49dca05d9c3e45cb727e0af))

* test: re-enabled py_func_v4 test ([`49d84ba`](https://github.com/python-gitlab/python-
→gitlab/commit/49d84ba7e95fa343e622505380b3080279b83f00))

```

(continues on next page)



(continued from previous page)

### Unknown

\* Merge pull request #899 from python-gitlab/chore/package-version

chore: bump to 1.12.0 ([`35cc8c7`](https://github.com/python-gitlab/python-gitlab/commit/35cc8c789fda4977add7f399bf426352b1aa246f))

\* Merge pull request #898 from python-gitlab/feat/update\_submodule

Feat/update submodule ([`6f4332d`](https://github.com/python-gitlab/python-gitlab/commit/6f4332db37b0a609ec4bd5e2c0b7ffc01717599c))

\* Merge pull request #897 from python-gitlab/refactor/remove-obsolete-image

refactor: remove obsolete test image ([`fcea41c`](https://github.com/python-gitlab/python-gitlab/commit/fcea41c61f7776cf57ed5001facbc1e77d2834c4))

\* Merge pull request #892 from godaji/remove-unused-code

Remove unused code and simplify string format. ([`214f7ef`](https://github.com/python-gitlab/python-gitlab/commit/214f7ef5f3b6e99b7756982c5b883e40d3b22657))

\* Merge pull request #896 from joue/fix-fonctionnal-test

improve fonctionnal tests ([`d7d2260`](https://github.com/python-gitlab/python-gitlab/commit/d7d2260945994a9e73fe3f7f9328f3ec9d9c54d4))

\* Merge pull request #894 from minitux/master

feat (project): add file blame api ([`082a624`](https://github.com/python-gitlab/python-gitlab/commit/082a62456deaa68274ed1c44a744c79c5356a622))

\* Merge pull request #891 from python-gitlab/docs/project-note-list

docs(projects): add note about project list ([`ba2b60e`](https://github.com/python-gitlab/python-gitlab/commit/ba2b60e32c12cacf18762a286d05e073529b9898))

\* Merge pull request #886 from LuckySB/cli\_user\_project

fix cli command user-project list ([`88b1833`](https://github.com/python-gitlab/python-gitlab/commit/88b183376de5b8c986eac24955ef129ca4d781cc))

\* Merge pull request #885 from sathieu/patch-1

Don't mangle label name on update ([`ff10726`](https://github.com/python-gitlab/python-gitlab/commit/ff10726d70a62d32ef39398a431def9656c93927))

\* Merge pull request #880 from python-gitlab/docs/tags-fix-typo

docs(repository-tags): fix typo ([`c287bdd`](https://github.com/python-gitlab/python-gitlab/commit/c287bdd0889881d89f991d3929ae513d91b5894c))

(continues on next page)

(continued from previous page)

```

* Merge pull request #878 from python-gitlab/test/todo-unit-test

Test/todo unit test ([`040894d`](https://github.com/python-gitlab/python-gitlab/commit/
↪040894d245709c5c2524d59d2b228a21dd74d1a4))

* Merge pull request #876 from sathieu/job_token

Add support for job token ([`8474829`](https://github.com/python-gitlab/python-gitlab/
↪commit/8474829a3fe40aca8f5d4c1c627908f0830a8f59))

* Merge pull request #875 from python-gitlab/feat/status-api

feat(user): add status api ([`b7f3342`](https://github.com/python-gitlab/python-gitlab/
↪commit/b7f33429c75ed2f464ebd9b4d3c56d3479df3faa))

* Merge pull request #874 from python-gitlab/test/py-fun-test

test: re-enabled py_func_v4 test ([`1490b0e`](https://github.com/python-gitlab/python-
↪gitlab/commit/1490b0e7f175d54cc6d35de7aac6d9e45c0e3d51))

## v1.11.0 (2019-08-31)

### Chore

* chore: bump package version ([`37542cd`](https://github.com/python-gitlab/python-
↪gitlab/commit/37542cd28aa94ba01d5d289d950350ec856745af))

### Feature

* feat: add methods to retrieve an individual project environment ([`29de40e`](https://
↪github.com/python-gitlab/python-gitlab/commit/
↪29de40ee6a20382c293d8cdc8d831b52ad56a657))

* feat: group labels with subscriptable mixin ([`4a9ef9f`](https://github.com/python-
↪gitlab/python-gitlab/commit/4a9ef9f0fa26e01fc6c97cf88b2a162e21f61cce))

### Fix

* fix/projects: avatar uploading for projects ([`558ace9`](https://github.com/python-
↪gitlab/python-gitlab/commit/558ace9b007ff9917734619c05a7c66008a4c3f0))

* fix: remove empty list default arguments

Signed-off-by: Frantisek Lachman <flachman@redhat.com> ([`6e204ce`](https://github.
↪com/python-gitlab/python-gitlab/commit/6e204ce819fc8bdd5359325ed7026a48d63f8103))

* fix: remove empty dict default arguments

Signed-off-by: Frantisek Lachman <flachman@redhat.com> ([`8fc8e35`](https://github.
↪com/python-gitlab/python-gitlab/commit/8fc8e35c63d7ebd80408ae002693618ca16488a7))

```

(continues on next page)

(continued from previous page)

```

* fix: add project and group label update without id to fix cli ([`a3d0d7c`](https://
↳github.com/python-gitlab/python-gitlab/commit/
↳a3d0d7c1e7b259a25d9dc84c0b1de5362c80abb8))

### Test

* test: add group label cli tests ([`f7f24bd`](https://github.com/python-gitlab/python-
↳gitlab/commit/f7f24bd324eaf33aa3d1d5dd12719237e5bf9816))

### Unknown

* Merge pull request #865 from orf/retrieve-environment

feat: add methods to retrieve an individual project environment ([`0389e66`](https://
↳github.com/python-gitlab/python-gitlab/commit/
↳0389e664c0a04021b3df097bacad3940f158607f))

* Merge pull request #861 from ravanscafi/fix-project-avatar

Fix avatar uploading for projects ([`99a9415`](https://github.com/python-gitlab/python-
↳gitlab/commit/99a941526a1845559d92b607fd9e2d86efb7e7b6))

* Merge pull request #860 from lachmanfrantisek/fix-mutable-default-arguments

Fix mutable default arguments ([`e8a3585`](https://github.com/python-gitlab/python-
↳gitlab/commit/e8a3585ed0e7dfa2f64f6c3378a598120f5f8167))

* Merge pull request #847 from sidisel-albertolopez/feat/grouplabels

feat: Add grouplabel support with subscribable mixin ([`edb3359`](https://github.com/
↳python-gitlab/python-gitlab/commit/edb3359fb3a77050d3e162da641445952397279b))

## v1.10.0 (2019-07-22)

### Chore

* chore: bump package version to 1.10.0 ([`c7c8470`](https://github.com/python-gitlab/
↳python-gitlab/commit/c7c847056b6d24ba7a54b93837950b7fdff6c477))

* chore(setup): add 3.7 to supported python versions ([`b1525c9`](https://github.com/
↳python-gitlab/python-gitlab/commit/b1525c9a4ca2d8c6c14d745638b3292a71763aeb))

* chore: move checks back to travis ([`b764525`](https://github.com/python-gitlab/python-
↳gitlab/commit/b7645251a0d073ca413bba80e87884cc236e63f2))

* chore: disable failing travis test ([`515aa9a`](https://github.com/python-gitlab/
↳python-gitlab/commit/515aa9ac2aba132d1dfde0418436ce163fca2313))

* chore(ci): rebuild test image, when something changed ([`2fff260`](https://github.com/
↳python-gitlab/python-gitlab/commit/2fff260a8db69558f865dda56f413627bb70d861))

```

(continues on next page)

(continued from previous page)

```

* chore(ci): update the GitLab version in the test image ([`c410699`](https://github.com/
python-gitlab/python-gitlab/commit/c4106992de392747ccec8c282ac0549932ccd1))

* chore(ci): fix gitlab PyPI publish ([`3e37df1`](https://github.com/python-gitlab/
python-gitlab/commit/3e37df16e2b6a8f1beffc3a595abcb06fd48a17c))

* chore(ci): add automatic GitLab image pushes ([`95c9b6d`](https://github.com/python-
gitlab/python-gitlab/commit/95c9b6dd489fc15c7dfceffca909917f4f3d4312))

* chore: add a tox job to run black

Allow lines to be 88 chars long for flake8. ([`c27fa48`](https://github.com/python-
gitlab/python-gitlab/commit/c27fa486698e441ebc16448ee93e5539cb885ced))

* chore(ci): use reliable ci system ([`724a672`](https://github.com/python-gitlab/python-
gitlab/commit/724a67211bc83d67deef856800af143f1dbd1e78))

* chore(ci): don't try to publish existing release ([`b4e818d`](https://github.com/
python-gitlab/python-gitlab/commit/b4e818db7887ff1ec337aaf392b5719f3931bc61))

* chore: release tags to PyPI automatically

Fixes #609 ([`3133b48`](https://github.com/python-gitlab/python-gitlab/commit/
3133b48a24ce3c9e2547bf2a679d73431dfbefab))

* chore(tests): add rate limit tests ([`e216f06`](https://github.com/python-gitlab/
python-gitlab/commit/e216f06d4d25d37a67239e93a8e2e400552be396))

### Documentation

* docs(snippets): fix project-snippets layout

Fixes #828 ([`7feb97e`](https://github.com/python-gitlab/python-gitlab/commit/
7feb97e9d89b4ef1401d141be3d00b9d0ff6b75c))

* docs(projects): add mention about project listings

Having exactly 20 internal and 5 private projects in the group
spent some time debugging this issue.

Hopefully that helped: https://github.com/python-gitlab/python-gitlab/issues/93

Imho should be definitely mention about `all=True` parameter. ([`f604b25`](https://
github.com/python-gitlab/python-gitlab/commit/
f604b2577b03a6a19641db3f2060f99d24cc7073))

* docs(readme): fix six url

six URL was pointing to 404 ([`0bc30f8`](https://github.com/python-gitlab/python-gitlab/
commit/0bc30f840c9c30dd529ae85bdece6262d2702c94))

* docs: re-order api examples

```

(continues on next page)

(continued from previous page)

```

`Pipelines and Jobs` and `Protected Branches` are out of order in contents and sometimes
↳ hard to find when looking for examples. ([`5d149a2`](https://github.com/python-gitlab/
↳ python-gitlab/commit/5d149a2262653b729f0105639ae5027ae5a109ea))

* docs: add pipeline deletion ([`2bb2571`](https://github.com/python-gitlab/python-
↳ gitlab/commit/2bb257182c237384d60b8d90cbbff5a0598f283b))

* docs(api-usage): fix project group example

Fixes #798 ([`40a1bf3`](https://github.com/python-gitlab/python-gitlab/commit/
↳ 40a1bf36c2df89daa1634e81c0635c1a63831090))

* docs: remove v3 support ([`7927663`](https://github.com/python-gitlab/python-gitlab/
↳ commit/792766319f7c43004460fc9b975549be55430987))

* docs: Add an example of trigger token usage

Closes #752 ([`ea1eefe`](https://github.com/python-gitlab/python-gitlab/commit/
↳ ea1eefef2896420ae4e4d248155e4c5d33b4034e))

* docs(readme): add more info about commitlint, code-format ([`286f703`](https://github.
↳ com/python-gitlab/python-gitlab/commit/286f7031ed542c97fb8792f61012d7448bee2658))

* docs(readme): provide commit message guidelines

Fixes #660 ([`bed8e1b`](https://github.com/python-gitlab/python-gitlab/commit/
↳ bed8e1ba80c73b1d976ec865756b62e66342ce32))

* docs(setup): use proper readme on PyPI ([`6898097`](https://github.com/python-gitlab/
↳ python-gitlab/commit/6898097c45d53a3176882a3d9cb86c0015f8d491))

* docs(groups): fix typo

Fixes #635 ([`ac2d65a`](https://github.com/python-gitlab/python-gitlab/commit/
↳ ac2d65aachba5c19eca857290c5b47ead6bb4356d))

* docs(projects): fix typo in code sample

Fixes #630 ([`b93f2a9`](https://github.com/python-gitlab/python-gitlab/commit/
↳ b93f2a9ea9661521878ac45d70c7bd9a5a470548))

* docs(cli): add PyYAML requirement notice

Fixes #606 ([`d29a489`](https://github.com/python-gitlab/python-gitlab/commit/
↳ d29a48981b521bf31d6f0304b88f39a63185328a))

* docs(readme): add docs build information ([`6585c96`](https://github.com/python-gitlab/
↳ python-gitlab/commit/6585c967732fe2a53c6ad6d4d2ab39aaa68258b0))

* docs: add MR approvals in index ([`0b45afb`](https://github.com/python-gitlab/python-
↳ gitlab/commit/0b45afbeed13745a2f9d8a6ec7d09704a6ab44fb))

```

(continues on next page)

(continued from previous page)

```

* docs(api-usage): add rate limit documentation ([`ad4de20`](https://github.com/python-
gitlab/python-gitlab/commit/ad4de20fe3a2fba2d35d4204bf5b0b7f589d4188))

* docs(projects): fix typo ([`c6bcfe6`](https://github.com/python-gitlab/python-gitlab/
commit/c6bcfe6d372af6557547a408a8b0a39b909f0cdf))

* docs: trigger_pipeline only accept branches and tags as ref

Fixes #430 ([`d63748a`](https://github.com/python-gitlab/python-gitlab/commit/
d63748a41cc22bba93a9adf0812e7eb7b74a0161))

* docs: fix invalid Raise attribute in docstrings ([`95a3fe6`](https://github.com/python-
gitlab/python-gitlab/commit/95a3fe6907676109e1cd2f52ca8f5ad17e0d01d0))

* docs: add missing = ([`391417c`](https://github.com/python-gitlab/python-gitlab/commit/
391417cd47d722760dfdaab577e9f419c5dca0e0))

* docs: remove the build warning about _static ([`764d3ca`](https://github.com/python-
gitlab/python-gitlab/commit/764d3ca0087f0536c48c9e1f60076af211138b9b))

* docs: fix &#34;required&#34; attribute ([`e64d0b9`](https://github.com/python-gitlab/
python-gitlab/commit/e64d0b997776387f400eaec21c37ce6e58d49095))

* docs: add missing requiredCreateAttrs ([`b08d74a`](https://github.com/python-gitlab/
python-gitlab/commit/b08d74ac3efb505961971edb998ce430e430d652))

* docs: add a note for python 3.5 for file content update

The data passed to the JSON serializer must be a string with python 3.
Document this in the examples.

Fix #175 ([`ca014f8`](https://github.com/python-gitlab/python-gitlab/commit/
ca014f8c3e4877a4cc1ae04e1302fb57d39f47c4))

* docs: improve the pagination section ([`29e2efe`](https://github.com/python-gitlab/
python-gitlab/commit/29e2efeae22ce5fa82e3541360b234e0053a65c2))

* docs: notes API ([`3e026d2`](https://github.com/python-gitlab/python-gitlab/commit/
3e026d2ee62eba3ad92ff2cdd53db19f5e0e9f6a))

* docs: snippets API ([`35b7f75`](https://github.com/python-gitlab/python-gitlab/commit/
35b7f750c7e38a39cd4cb27195d9aa4807503b29))

* docs: tags API ([`dd79eda`](https://github.com/python-gitlab/python-gitlab/commit/
dd79eda78f91fc7e1e9a08b1e70ef48e3b4bb06d))

* docs: system hooks API ([`5c51bf3`](https://github.com/python-gitlab/python-gitlab/
commit/5c51bf3d49302afe4725575a83d81a8c9eeb8779))

* docs: add ApplicationSettings API ([`ab7d794`](https://github.com/python-gitlab/python-
gitlab/commit/ab7d794251bcdabfce69b1bde0628cd3b710d784))

```

(continues on next page)

(continued from previous page)

```

* docs: repository files API ([`f00340f`](https://github.com/python-gitlab/python-gitlab/commit/f00340f72935b6fd80df7b62b811644b63049b5a))

* docs: project repository API ([`71a2a4f`](https://github.com/python-gitlab/python-gitlab/commit/71a2a4fb84321e73418fda1ce4e4d47177af928c))

* docs: add milestones API ([`7411907`](https://github.com/python-gitlab/python-gitlab/commit/74119073dae18214df1dd67ded6cd57abda335d4))

* docs: add MR API ([`5614a7c`](https://github.com/python-gitlab/python-gitlab/commit/5614a7c9bf62aede3804469b6781f45d927508ea))

* docs: add licenses API ([`4540614`](https://github.com/python-gitlab/python-gitlab/commit/4540614a38067944c628505225bb15928d8e3c93))

* docs: add labales API ([`31882b8`](https://github.com/python-gitlab/python-gitlab/commit/31882b8a57f3f4c7e4c4b319af436795ebafd3))

* docs: add deploy keys API ([`ea089e0`](https://github.com/python-gitlab/python-gitlab/commit/ea089e092439a8fe95b50c3d0592358550389b51))

* docs: issues API ([`41cbc32`](https://github.com/python-gitlab/python-gitlab/commit/41cbc32621004aab2cae5f7c14fc60005ef7b966))

* docs: commits API ([`07c5594`](https://github.com/python-gitlab/python-gitlab/commit/07c55943eebb302bc1b8feaf482d929c83e9ebe1))

* docs: groups API documentation ([`4d871aa`](https://github.com/python-gitlab/python-gitlab/commit/4d871aadfaa9f57f5ae9f8b49f8367a5ef58545d))

* docs: Add builds-related API docs ([`8e6a944`](https://github.com/python-gitlab/python-gitlab/commit/8e6a9442324926ed1dec0a8bfaf77792e4bdb10f))

* docs: fork relationship API ([`21f48b3`](https://github.com/python-gitlab/python-gitlab/commit/21f48b357130720551d5cccbc62f5275fe970378))

* docs: project search API ([`e4cd04c`](https://github.com/python-gitlab/python-gitlab/commit/e4cd04c225e2160f02a8f292dbd4c0f6350769e4))

* docs: document hooks API ([`b21dca0`](https://github.com/python-gitlab/python-gitlab/commit/b21dca0acb2c12add229a1742e0c552aa50618c1))

* docs: add project members doc ([`dcf31a4`](https://github.com/python-gitlab/python-gitlab/commit/dcf31a425217efebe56d4cbc8250dceb3844b2fa))

* docs: document projects API ([`967595f`](https://github.com/python-gitlab/python-gitlab/commit/967595f504b8de076ae9218a96c3b8dd6273b9d6))

* docs: crossref improvements ([`6f9f42b`](https://github.com/python-gitlab/python-gitlab/commit/6f9f42b64cb82929af60e299c70773af6d406a6e))

```

(continues on next page)



(continued from previous page)

```

* docs: start a FAQ ([`c305459`](https://github.com/python-gitlab/python-gitlab/commit/
↪c3054592f79caa782ec79816501335e9a5c4e9ed))

* docs: do not use the :option: markup ([`368017c`](https://github.com/python-gitlab/
↪python-gitlab/commit/368017c01f15013ab4cc9405c246a86e67f3b067))

### Feature

* feat: add mr rebase method ([`bc4280c`](https://github.com/python-gitlab/python-gitlab/
↪commit/bc4280c2fbff115bd5e29a6f5012ae518610f626))

* feat: get artifact by ref and job ([`cda1174`](https://github.com/python-gitlab/python-
↪gitlab/commit/cda117456791977ad300a1dd26dec56009dac55e))

* feat: add support for board update

Closes #801 ([`908d79f`](https://github.com/python-gitlab/python-gitlab/commit/
↪908d79fa56965e7b3afcfa23236beef457cfa4b4))

* feat: add support for issue.related_merge_requests

Closes #794 ([`90a3631`](https://github.com/python-gitlab/python-gitlab/commit/
↪90a363154067bcf763043124d172eaf705c8fe90))

* feat: bump version to 1.9.0 ([`aaed448`](https://github.com/python-gitlab/python-
↪gitlab/commit/aaed44837869bd2ce22b6f0d2e1196b1d0e626a6))

* feat: implement artifacts deletion

Closes #744 ([`76b6e1f`](https://github.com/python-gitlab/python-gitlab/commit/
↪76b6e1fc0f42ad00f21d284b4ca2c45d6020fd19))

* feat: add endpoint to get the variables of a pipeline

It adds a new endpoint which was released in the Gitlab CE 11.11.

Signed-off-by: Agustin Henze <tin@redhat.com> ([`564de48`](https://github.com/
↪python-gitlab/python-gitlab/commit/564de484f5ef4c76261057d3d2207dc747da020b))

* feat(GitLab Update): delete ProjectPipeline (#736)

* feat(GitLab Update): delete ProjectPipeline

As of Gitlab 11.6 it is now possible to delete a pipeline - https://docs.gitlab.com/ee/
↪api/pipelines.html#delete-a-pipeline ([`768ce19`](https://github.com/python-gitlab/
↪python-gitlab/commit/768ce19c5e5bb197cddd4e3871c175e935c68312))

* feat: Added approve & unapprove method for Mergerequests

Official GitLab API supports this for GitLab EE ([`53f7de7`](https://github.com/python-
↪gitlab/python-gitlab/commit/53f7de7bfe0056950a8e7271632da3f89e3ba3b3))

```

(continues on next page)



(continued from previous page)

```

* feat: obey the rate limit

done by using the retry-after header

Fixes #166 ([`2abf9ab`](https://github.com/python-gitlab/python-gitlab/commit/
↪2abf9abacf834da797f2edf6866e12886d642b9d))

### Fix

* fix: improve pickle support ([`b4b5dec`](https://github.com/python-gitlab/python-
↪gitlab/commit/b4b5dec7e49ac16d98d56547a874fb8f9d5492b))

* fix(cli): allow --recursive parameter in repository tree

Fixes #718
Fixes #731 ([`7969a78`](https://github.com/python-gitlab/python-gitlab/commit/
↪7969a78ce8605c2b0195734e54c7d12086447304))

* fix(cli): don't fail when the short print attr value is None

Fixes #717
Fixes #727 ([`8d1552a`](https://github.com/python-gitlab/python-gitlab/commit/
↪8d1552a0ad137ca5e14fabfc75f7ca034c2a78ca))

* fix(cli): fix update value for key not working ([`b766203`](https://github.com/python-
↪gitlab/python-gitlab/commit/b7662039d191ebb6a4061c276e78999e2da7cd3f))

* fix: convert # to %23 in URLs

Refactor a bit to handle this change, and add unit tests.

Closes #779 ([`14f5385`](https://github.com/python-gitlab/python-gitlab/commit/
↪14f538501bfb47c92e02e615d0817675158db3cf))

* fix: pep8 errors

Errors have not been detected by broken travis runs. ([`334f9ef`](https://github.com/
↪python-gitlab/python-gitlab/commit/334f9efb18c95bb5df3271d26fa0a55b7aec1c7a))

* fix(api): Make *MemberManager.all() return a list of objects

Fixes #699 ([`d74ff50`](https://github.com/python-gitlab/python-gitlab/commit/
↪d74ff506ca0aadaba3221fc54cbebb678240564f))

* fix: use python2 compatible syntax for super ([`b08efcb`](https://github.com/python-
↪gitlab/python-gitlab/commit/b08efcb9d155c20fa938534dd2d912f5191eede6))

* fix: re-add merge request pipelines ([`877ddc0`](https://github.com/python-gitlab/
↪python-gitlab/commit/877ddc0dbb664cd86e870bb81d46ca614770b50e))

* fix(api): Don't try to parse raw downloads

```

(continues on next page)

(continued from previous page)

`http_get` always tries to interpret the retrieved data if the content-type is json. In some cases (artifact download for instance) this is not the expected behavior.

This patch changes `http_get` and download methods to always get the raw data without parsing.

Closes #683 ([`35a6d85`](https://github.com/python-gitlab/python-gitlab/commit/35a6d85acea4776e9c4ad23ff75259481a6bcf8d))

\* `fix(api)`: avoid parameter conflicts with python and gitlab

Provide another way to send data to gitlab with a new `query_parameters` argument. This parameter can be used to explicitly define the dict of items to send to the server, so that `**kwargs` are only used to specify python-gitlab specific parameters.

Closes #566

Closes #629 ([`4bd027a`](https://github.com/python-gitlab/python-gitlab/commit/4bd027aac41c41f7e22af93c7be0058d2faf7fb4))

\* `fix`: remove `decode()` on `error_message` string

The integration tests failed because a test called `decode()` on a string-type variable - the `GitLabException` class handles byte-to-string conversion already in its `__init__`. This commit removes the call to `decode()` in the test.

Traceback (most recent call last): File `"/tools/python_test_v4.py"`, line 801, in `<module>` assert 'Retry later' in `error_message.decode()` AttributeError: 'str' object has no attribute 'decode' `` (16bda20)

- `fix`: handle empty 'Retry-After' header from GitLab

When requests are throttled (HTTP response code 429), python-gitlab assumed that 'Retry-After' existed in the response headers. This is not always the case and so the request fails due to a `KeyError`. The change in this commit adds a rudimentary exponential backoff to the `http_request` method, which defaults to 10 retries but can be set to -1 to retry without bound. (7a3724f)

- `fix(api)`: make `reset_time_estimate()` work again

Closes #672 (cb388d6)

- `fix`: enable use of YAML in the CLI

In order to use the YAML output, `PyYaml` needs to be installed on the docker image. This commit adds the installation to the dockerfile as a separate layer. (ad0b476)

- `fix`: docker entry point argument passing

Fixes the problem of passing spaces in the arguments to the docker entrypoint.

Before this fix, there was virtually no way to pass spaces in arguments such as task description. (67ab637)

- `fix(cli)`: exit on config parse error, instead of crashing
- Exit and hint user about possible errors
- test: adjust test cases to config missing error (6ad9da0)

- fix(docker): use docker image with current sources ([06e8ca8](#))
- fix(cli): print help and usage without config file

Fixes #560 ([6bb4d17](#))

## Refactor

- refactor: format everything black ([318d277](#))
- refactor: rename MASTER\_ACCESS

to MAINTAINER\_ACCESS to follow GitLab 11.0 docs

See: <https://docs.gitlab.com/ce/user/permissions.html#project-members-permissions> ([c38775a](#))

## Style

- style: format with black again ([22b5082](#))

## Test

- test: minor test fixes ([3b523f4](#))
- test: add project releases test

Fixes #762 ([8ff8af0](#))

- test: increase speed by disabling the rate limit faster ([497f56c](#))
- test: always use latest version to test ([82b0fc6](#))
- test: update the tests for GitLab 11.11

Changes in GitLab make the functional tests fail:

- Some actions add new notes and discussions: do not use hardcoded values in related listing asserts
- The feature flag API is buggy (errors 500): disable the tests for now ([622854f](#))

## Unknown

- Merge pull request #842 from python-gitlab/chore/bump-package-version

chore: bump package version to 1.10.0 ([2c1ea56](#))

- Merge pull request #841 from python-gitlab/docs/project-snippets

docs(snippets): fix project-snippets layout ([29d102f](#))

- Merge pull request #840 from python-gitlab/docs/project-docs

docs(projects): add mention about project listings ([de19296](#))

- Merge pull request #838 from python-gitlab/pickle-fix

fix: improve pickle support ([f0e3daa](#))

- Merge pull request #839 from python-gitlab/dajsn-fix-readme-six-url

docs(readme): fix six url ([8d54cf5](#))

- Merge pull request #837 from python-gitlab/PR-bugfix-718

fix(cli): allow `--recursive` parameter in repository tree ([27b9706](#))

- Merge pull request #836 from python-gitlab/test/project-releases

test: add project releases test ([262b222](#))

- Merge pull request #835 from python-gitlab/bugfix-717

fix(cli): don't fail when the short print attr value is None ([0b0a60f](#))

- Merge pull request #833 from python-gitlab/project-variable-update

fix(cli): fix update value for key not working ([6c673c6](#))

- Merge pull request #834 from python-gitlab/chore/setup-supported-versions

chore(setup): add 3.7 to supported python versions ([8306ef2](#))

- Merge pull request #832 from python-gitlab/test/always-latest

test: always use latest version to test ([8f1ed93](#))

- Merge pull request #823 from jeroen92/rebase-mr

Resolve #822, add mr rebase ([b9877b4](#))

- Merge pull request #831 from python-gitlab/chore/move-back-to-travis

chore: move checks back to travis ([c8a7e31](#))

- Merge pull request #830 from python-gitlab/chore/ci-disable-py-func

Chore/ci disable py func ([2e42e28](#))

- Merge pull request #827 from ahaynssen/master

docs: re-order api examples ([5492b71](#))

- Merge pull request #824 from python-gitlab/feat/add-ref-artifacts

feat: get artifact by ref and job ([4a8503d](#))

- Merge pull request #803 from python-gitlab/feat/related\_mr

feat: add support for `issue.related_merge_requests` ([ad1c0dd](#))

- Merge pull request #804 from python-gitlab/feat/update\_board

feat: add support for board update ([f539c36](#))

- Merge pull request #808 from minitux/patch-1 ([2ea8eb8](#))

- Merge pull request #805 from python-gitlab/chore/ci-rebuild-image

chore(ci): rebuild test image, when something changed ([6625a06](#))

- Merge pull request #802 from python-gitlab/chore/gitlab-11.11.3

chore(ci): update the GitLab version in the test image ([51751c5](#))

- Merge pull request #792 from python-gitlab/chore/enable-gitlab-ci-builds

chore(ci): add automatic GitLab image pushes ([50c53c0](#))

- Merge pull request #800 from python-gitlab/chore/ci-publish

chore(ci): fix gitlab PyPI publish ([722a6ef](#))

- Merge pull request #797 from python-gitlab/feat/version-bump

feat: bump version to 1.9.0 ([4b04432](#))

- Merge pull request #799 from python-gitlab/docs/fix-group-access  
docs(api-usage): fix project group example ([b5aaa3e](#))
- Merge pull request #767 from python-gitlab/fix/744/delete\_artifacts  
feature: Implement artifacts deletion ([4e1dd27](#))
- Merge pull request #791 from python-gitlab/tests-for-gl-11.11  
test: update the tests for GitLab 11.11 ([e45a6e2](#))
- Merge pull request #789 from python-gitlab/no-more-v3  
docs: remove v3 support ([e2115b1](#))
- Merge pull request #785 from agustinhzenze/add-pipeline-get-variables-endpoint  
Add new endpoint to get the variables of a pipeline ([463cedc](#))
- Merge pull request #768 from python-gitlab/trigger\_token\_example  
docs: Add an example of trigger token usage ([5af0b52](#))
- Merge pull request #790 from python-gitlab/fix/779  
fix: convert # to %23 in URLs ([101ccd1](#))
- Merge pull request #788 from python-gitlab/black-in-tox  
Add a tox job to run black ([d135e4e](#))
- Merge pull request #778 from python-gitlab/docs/readme-code-format  
docs(readme): add more info about commitlint, code-format ([794d64c](#))
- Merge pull request #775 from python-gitlab/refactor/black  
refactor: format everything black ([290e5ed](#))
- Merge pull request #776 from python-gitlab/revert-760-custom\_cli\_actions\_fix  
Revert “Custom cli actions fix” ([ef32990](#))
- Revert “Custom cli actions fix” ([d3a20c5](#))
- Merge pull request #760 from kkoralsky/custom\_cli\_actions\_fix  
Custom cli actions fix ([e8823e9](#))
- Merge pull request #759 from kkoralsky/registry\_api  
registry api implementation ([84bcd00](#))
- Merge pull request #773 from python-gitlab/chore/ci-reliable-system  
chore(ci): use reliable ci system ([3dc6413](#))
- documentation fix ([2d9078e](#))
- whitespaces ([c91230e](#))
- documentation ([4d31b9c](#))
- fix docstring & improve coding style ([3cede7b](#))
- register cli action for delete\_in\_bulk ([0b79ce9](#))
- fix repository\_id marshaling in cli ([340cd37](#))
- merged new release & registry apis ([910c286](#))

- Merge pull request #769 from python-gitlab/pep-fixes

fix: pep8 errors ([a730598](#))

- Merge pull request #746 from therealgambo/master

add project releases api ([16de1b0](#))

- fix `-/` replacement for `*Manager` custom actions ([6158fd2](#))
- dont ask for id attr if this is `*Manager` originating custom action ([adb6305](#))
- Use `NoUpdateMixin` for now ([8e55a3c](#))
- add project releases api ([3680545](#))
- Merge pull request #714 from jaraco/feature/runpy-invoke

Add runpy hook, allowing invocation with `'python -m gitlab'`. ([a3a7713](#))

- Add runpy hook. Fixes #713.

Allows for invocation with `'python -m gitlab'` ([cd2a14e](#))

- Merge pull request #738 from jeroendecroos/Gitlab\_from\_config\_inheritance

Make `gitlab.Gitlab.from_config` a classmethod ([6bd1902](#))

- Make `gitlab.Gitlab.from_config` a classmethod ([0b70da3](#))
- Merge pull request #732 from hakanf/master

Re-enable command specific help messages ([a6e10f9](#))

- Use `sys.exit` as in rest of code ([6fe2988](#))
- Merge pull request #729 from xarx00/PR-bugfix-716

Fix for #716: `%d` replaced by `%s` ([bc973d4](#))

- Re-enable command specific help messages

This makes sure that the global help message wont be printed instead of the command specific one unless we fail to read the configuration file ([a8caddc](#))

- Fix for #716: `%d` replaced by `%s` ([675f879](#))
- Merge pull request #725 from python-gitlab/fix/699

fix(api): Make `*MemberManager.all()` return a list of objects ([1792442](#))

- Merge pull request #721 from purificant/fix\_typo

fix tiny typo ([b21fa28](#))

- fix tiny typo ([2023875](#))
- Merge pull request #707 from python-gitlab/fix/python-tests

fix: use python2 compatible syntax for `super` ([e58d2a8](#))

- Merge pull request #706 from python-gitlab/chore/ci-existing-release

chore(ci): don't try to publish existing release ([39cb97d](#))

- Merge pull request #702 from jpiron/eq\_hash

Implement `eq` and `hash` methods ([a4ea0fe](#))

- Merge pull request #705 from python-gitlab/release-1.8.0

Release version 1.8.0 ([57fa4e3](#))

- Release version 1.8.0 ([4fce338](#))
- Merge pull request #701 from jpiron/fix\_all\_behaviour

Fix all kwarg behaviour ([8ce4e9e](#))

- Implement **eq** and **hash** methods

To ease lists and sets manipulations. ([3d60850](#))

- Fix all kwarg behaviour

`all` kwarg is used to manage `GitlabList` generator behaviour. However, as it is not popped from kwargs, it is sent to Gitlab API. Some endpoints such as [the project commits](#) one, support a `all` attribute. This means a call like `project.commits.list(all=True, ref_name='master')` won't return all the master commits as one might expect but all the repository's commits. To prevent confusion, the same kwarg shouldn't be used for 2 distinct purposes. Moreover according to [the documentation](#), the `all` project commits API endpoint attribute doesn't seem supported. ([6b2bf5b](#))

- Merge pull request #689 from python-gitlab/fix/wrong-rebase

fix: re-add merge request pipelines ([31bca2f](#))

- Merge pull request #685 from Joustie/master

feat: Added approve method for Mergerequests ([641b80a](#))

- Merge branch 'master' into master ([b51d296](#))
- Merge pull request #687 from python-gitlab/fix/683/raw\_download

fix(api): Don't try to parse raw downloads ([52d7631](#))

- Merge pull request #680 from python-gitlab/chore/automatic-deploy

chore: release tags to PyPI automatically ([ca8c85c](#))

- Merge pull request #681 from python-gitlab/no-param-conflicts

fix(api): avoid parameter conflicts with python and gitlab ([572029c](#))

- Merge pull request #678 from appian/backoff-requests

Fix missing "Retry-After" header and fix integration tests ([89679ce](#))

- Merge pull request #673 from python-gitlab/fix/672

fix(api): make `reset_time_estimate()` work again ([ce2c835](#))

- Merge pull request #664 from python-gitlab/docs/commit-message

docs(readme): provide commit message guidelines ([85ac102](#))

- Merge pull request #659 from python-gitlab/docs/readme-pypi

docs(setup): use proper readme on PyPI ([9be82e1](#))

- Merge pull request #657 from python-gitlab/release-1.7.0

Prepare the 1.7.0 release ([704ca51](#))

- Prepare the 1.7.0 release ([456f3c4](#))
- Merge pull request #656 from esabouraud/feature-protectedbranchesoptions

Issue 653 Add access control options to protected branch creation ([59e3e45](#))

- Add access control options to protected branch creation ([cebbb6f](#))

- Merge pull request #652 from roozbehf/fix/docker-enable-use-of-yaml

fix: enable use of YAML in the CLI ([728f2dd](#))

- Merge pull request #651 from roozbehf/fix/docker-entrypoint-arguments

fix: docker entry point argument passing ([f945910](#))

- Merge pull request #641 from python-gitlab/refactor/excpetion\_msg

Improve error message handling in exceptions ([7bd41cb](#))

- Merge pull request #625 from python-gitlab/fix/611/resource\_label\_event

Add support to resource label events ([20eb7d8](#))

- Merge pull request #642 from python-gitlab/feature/589/member\_all

[feature] Add support for members all() method ([22536f3](#))

- [feature] Add support for members all() method

Closes #589 ([ef1523a](#))

- Improve error message handling in exceptions
- Depending on the request Gitlab has a 'message' or 'error' attribute in the json data, handle both
- Add some consistency by converting messages to unicode or str for exceptions (depending on the python version)

Closes #616 ([1fb1296](#))

- Merge pull request #639 from python-gitlab/fix/628/doc\_typo

[docs] Fix typo in custom attributes example ([011274e](#))

- Merge pull request #638 from python-gitlab/fix/633/milestone\_filter

[docs] Fix the milestone filetring doc (iid -> iids) ([e6df9a8](#))

- [docs] Fix typo in custom attributes example

Closes #628 ([bb251b8](#))

- [docs] Fix the milestone filetring doc (iid -> iids)

Fixes #633 ([0c9a00b](#))

- Add support to resource label events

Closes #611 ([95d0d74](#))

- Merge pull request #634 from python-gitlab/docs/project-typo

docs(projects): fix typo in code sample

Closes #630 ([c8eaeb2](#))

- Merge pull request #636 from python-gitlab/docs/groups-fix-typo

docs(groups): fix typo ([c72a87a](#))

- Merge pull request #627 from nicgrayson/fix-docs-typo

Fix 3 typos in docs ([7f09666](#))

- Fix 3 typos ([a5ab2bb](#))

- Merge pull request #624 from python-gitlab/update/docker-image

Use the pythongitlab/test-python-gitlab docker image for tests ([742243f](#))



- Merge pull request #619 from python-gitlab/issue/595

[docs] Add an example of pipeline schedule vars listing ([fcce7a3](#))

- Merge pull request #626 from python-gitlab/fix/596/maintainer\_wanted

[README] Remove the “maintainer(s) wanted” notice ([bc6ce04](#))

- [README] Remove the “maintainer(s) wanted” notice

Closes #596 ([f51fa19](#))

- [docs] Add an example of pipeline schedule vars listing

Closes #595 ([f7fbfca](#))

- Use the `pythongitlab/test-python-gitlab` docker image for tests

This images is updated to the latest GitLab CE.

Fix the `diff()` test to match the change in the API output. ([2c6c929](#))

- Merge pull request #620 from bittner/patch-1

Add Gitter badge to README ([74623cf](#))

- Add Gitter badge to README ([31d1c5d](#))
- Merge pull request #613 from mkosiarc/fixDoc

[docs] fix discussions typo ([368a34d](#))

- [docs] fix discussions typo ([54b6a54](#))
- Merge pull request #605 from python-gitlab/fix/docker

`fix(docker)`: use docker image with current sources ([c9f7986](#))

- Merge pull request #608 from python-gitlab/ci-output-option

`docs(cli)`: add PyYAML requirement notice ([156a21e](#))

- Merge pull request #607 from python-gitlab/refactor/rename-variable

`refactor`: rename `MASTER_ACCESS` ([5ff2608](#))

- Merge pull request #601 from max-wittig/fix/help-usage

`fix(cli)`: print help and usage without config file ([32b5122](#))

- Merge pull request #600 from hans-d/docker

more flexible docker ([3a8b1a0](#))

- Merge branch ‘master’ into docker ([756c73c](#))
- Add project protected tags management (#581) ([ea71f1d](#))
- more flexible docker ([21d2577](#))
- README: add a note about maintainers ([77f4d3a](#))
- Merge pull request #586 from Halliburton-Landmark/project-issue-create-args

add missing comma in `ProjectIssueManager _create_attrs` ([58d5c0a](#))

- add missing comma in `ProjectIssueManager _create_attrs`

This fixes the argument handling for assignee/milestone ID when for `project-issue create` ([83fb4f9](#))

- [docs] Add a note about `GroupProject` limited API ([9e60364](#))

- Fix the https redirection test ([6f80380](#))
- [docs] add a warning about https://

http to https redirection cause problems. Make notes of this in the docs. ([042b706](#))

- [docs] fix cut and paste leftover ([b02c30f](#))
- Use https:// for gitlab URL ([256518c](#))
- [docs] Fix the owned/starred usage documentation

Closes #579 ([ccf0c2a](#))

- 1.6.0 release ([d8c2488](#))
- [cli] Fix the project-export download

Closes #559 ([facbc8c](#))

- Minor doc updates ([e9506d1](#))
- Add a FAQ ([4d4c8ad](#))
- Raise an exception on https redirects for PUT/POST

POST and PUT requests are modified by clients when redirections happen. A common problem with python-gitlab is a misconfiguration of the server URL: the http to https redirection breaks some requests.

With this change python-gitlab should detect problematic redirections, and raise a proper exception instead of failing with a cryptic error.

Closes #565 ([a221d7b](#))

- [docs] Add/updates notes about read-only objects

MR and issues attached to the root API or groups are not editable. Provide notes describing how to manage this. ([80a68f9](#))

- Merge pull request #572 from btmanm/master

Update projects.rst ([ff6ca5d](#))

- Update projects.rst ([6ada4b0](#))
- Merge pull request #569 from matthias/patch-1

Minor typo “ou” vs. “or” ([0a687d3](#))

- Minor typo “ou” vs. “or”

This change fixes a minor typo in the table of possible values for options in the global section. ([a68f459](#))

- Add support for project transfers from the projects interface. (#561)

See <https://docs.gitlab.com/ee/api/projects.html#transfer-a-project-to-a-new-namespace> ([a1c79d2](#))

- Added support for listing forks of a project (#562) ([b325bd7](#))
- MR: add the squash attribute for create/update

Closes #557 ([35c8c82](#))

- Implement MR.pipelines()

Closes #555 ([32ae924](#))

- Support group and global MR listing

Closes #553 ([0379efa](#))

- Project import: fix the override\_params parameter

Closes #552 ([3461904](#))

- [cli] Fix the case where we have nothing to print ([a139179](#))
- [cli] Output: handle bytes in API responses

Closes #548 ([bbef1f9](#))

- Improve the snippets examples

closes #543 ([bdbec67](#))

- Merge pull request #542 from tpdowes/patch-1

Fix simple typo in identity modification example ([9751ab6](#))

- Fix simple typo in identity modification example ([35fe227](#))
- [docs] don't use hardcoded values for ids ([fe43a28](#))
- 1.5.1 release ([5e6330f](#))
- Improve the protect branch creation example

Closes #536 ([590c41a](#))

- Fix the ProjectPipelineJob base class

Closes #537 ([4cf8118](#))

- Prepare the 1.5.0 release ([eaa4450](#))
- [cli] Fix the non-verbose output of ProjectCommitComment

Closes #433 ([d5289fe](#))

- Use the same description for \*\*kwargs everywhere ([b1c6392](#))
- [docs] Add an example for external identities settings

Fixes #528 ([21e382b](#))

- Revert "make as\_list work for all queries"

This reverts commit 8e787612fa77dc945a4c1327e9faa6eee10c48f2.

This change broke the basic generator usage (Fixes #534) ([1a04634](#))

- Add support for epics API (EE)

Fixes #525 ([ba90e30](#))

- README update ([b2cb700](#))
- ProjectPipelineJob objects can only be listed

And they are not directly related to ProjectJob objects.

Fixes #531 ([e1af0a0](#))

- Add support for the LDAP groups API ([ebf822c](#))
- Add support for the EE license API ([5183069](#))
- Merge pull request #530 from stefancrain/master

Correct session example ([3f88ad0](#))

- Correct session example ([01969c2](#))

- Implement MR-level approvals

Fixes #323 ([59a19ca](#))

- Add push rules tests ([8df6de9](#))
- Add project push rules configuration (#520) ([2c22a34](#))
- [docs] projects.all() doesn't exist in v4

Fixes #526 ([617aa64](#))

- Pull mirroring doesn't return data ([b610d66](#))
- Add support for Project.pull\_mirror (EE) ([ebd6217](#))
- Add support for board creation/deletion (EE) ([f4c4e52](#))
- Add support for LDAP groups ([d6a61af](#))
- Merge pull request #514 from jouve/generator

make as\_list=False work for all=True queries ([a6512f9](#))

- Add support for issue links (EE)

Fixes #422 ([8873eda](#))

- Add geo nodes API support

Fixes #524 ([39c8ad5](#))

- Merge branch 'master' of github.com:python-gitlab/python-gitlab ([5a855fd](#))
- Merge pull request #522 from beyondliu/master

fix #521 change post\_data default value to None ([6dd8774](#))

- Add basic testing forr EE endpoints

Today we don't have a solution for easily deploying an EE instance so using the functional tools is not possible.

This patch provides a testing script that needs to be run against a private EE instance. ([c88333b](#))

- Add support for project-level MR approval configuration ([473dc6f](#))
- fix #521 change post\_data default value to None ([d4c1a8c](#))
- make as\_list work for all queries ([8e78761](#))
- Merge pull request #519 from jouve/silence

silence logs/warnings in unittests ([bbefb99](#))

- silence logs/warnings in unittests ([3fa24ea](#))
- Merge pull request #517 from jouve/dedup

projectpipelinejob was defined twice ([92ca5c4](#))

- projectpipelinejob was defined twice ([17d9354](#))
- Use python 2 on travis for now ([33c2457](#))
- tests: default to python 3

Fix the bytes/str issues ([b3df26e](#))

- Make ProjectCommitStatus.create work with CLI

Fixes #511 ([34c8a03](#))

- `time_stats()`: use an existing attribute if available

A `time_stats` attribute is returned by GitLab when fetching issues and merge requests (on reasonably recent GitLab versions). Use this info instead of making a new API call if possible.

Fixes #510 ([f2223e2](#))

- Update time stats docs ([f8e6b13](#))
- Fix the IssueManager path to avoid redirections ([eae1805](#))
- Add support for group badges

Also consolidate project/group badges tests, and add some docs

Fixes #469 ([9412a5d](#))

- Merge pull request #507 from Miouge1/badges

Add support for Project badges ([01a41ef](#))

- Add support for the gitlab CI lint API ([40b9f4d](#))
- Update the settings attributes ([0cc9828](#))
- Implement runner token validation ([71368e7](#))
- Runners can be created (registered) ([782875a](#))
- Implement runner jobs listing ([0be81cb](#))
- Add missing project attributes ([096d9ec](#))
- Add pipeline listing filters ([51718ea](#))
- Update MR attributes ([2332904](#))
- Implement the markdown rendering API

Testing will be enable when GitLab 11.0 is available. ([9be50be](#))

- Add support for group boards ([fbd2010](#))
- Fix the participants() decorator ([8374bcc](#))
- Issues: add missing attributes and methods ([e901f44](#))
- Update some group attributes ([4ec8975](#))
- Add feature flags deletion support ([f082568](#))
- Add support for environment stop() ([9c19e06](#))
- deploy key: add missing attributes ([6779616](#))
- Deployment: add list filters ([ce7911a](#))
- Add commit.merge\_requests() support ([c19ad90](#))
- Enable mr.participant test ([3c53f7f](#))
- Implement commit.refs() ([32569ea](#))
- Add missing docs file ([63a4c7c](#))
- Implement user\_agent\_detail for snippets

Add a new UserAgentDetail mixin to avoid code duplication. ([7025743](#))

- Add support for Project badges ([e00cad4](#))

- Add support for merged branches deletion ([590ea0d](#))
- Add support for the discussions API

Fixes #501 ([4461139](#))

- Document the global `per_page` setting ([660f0cf](#))
- Merge pull request #505 from [jouve/config\\_per\\_page](#)

add `per_page` config option ([d981904](#))

- add `per_page` config option ([589a9aa](#))
- Add support for the search API

Fixes #470 ([97c8619](#))

- Add support for project import/export

Fixes #471 ([b5f9616](#))

- pep8 fix ([42007ec](#))
- Add support for user avatar upload

Fixes #308 ([174185b](#))

- travis-ci: remove the v3 tests ([175abe9](#))
- [docs] update the sphinx extension for v4 objects ([194ed0b](#))
- [docs] Rework the examples pages
- Get rid of the `.py` files and bring all the python examples in the RST files
- Fix a few things ([5292ffb](#))
- Add release notes for 1.5 ([2c34237](#))
- Drop `GetFromListMixin` ([09d1ec0](#))
- Drop API v3 support

Drop the code, the tests, and update the documentation. ([fe89b94](#))

- `ChangeLog`: fix link ([7011694](#))
- Prepare the 1.4.0 release ([3ad706e](#))
- pep8 fix ([e6ecf65](#))
- Deprecate `GetFromListMixin`

This mixin provides a workaround for `get()` for GitLab objects that don't implement a 'get a single object' API. We are now getting conflicts because GitLab adds GET methods, and this is against the "Implement only what exists in the API" strategy.

Also use the proper GET API call for objects that support it. ([a877514](#))

- longer docker image startup timeout for tests ([5335788](#))
- Add docs for the `files` arg in `http_*` ([79c4682](#))
- Merge pull request #500 from [ericfrederich/efficient\\_get](#)

More efficient `.get()` for group members. ([66d8f30](#))

- More efficient `.get()` for group members.

Fixes #499 ([dabfeb3](#))

- api-usage: bit more detail for listing with all ([4cc9739](#))
- prepare release notes for 1.4 ([68b798b](#))
- [tests] fix functional tests for python3

Fixes #486 ([3dc997f](#))

- [docs] update service.available() example for API v4

Fixes #482 ([6d4ef0f](#))

- [docs] add a code example for listing commits of a MR

Fixes #491 ([037585c](#))

- [docs] move mr samples in rst file ([a643763](#))
- Merge pull request #484 from Matusf/docs-example-raises-attribute-error

Change method for getting content of snippet ([85f2388](#))

- Fix URL encoding on branch methods

Fixes #493 ([736fece](#))

- Add API v3 example ([5c16c8d](#))
- Merge pull request #488 from siemens/feat/rate-limit

feat: obey the rate limit ([86a8251](#))

- Revert “Token scopes are a list”

This reverts commit 32b399af0e506b38a10a2c625338848a03f0b35d. ([25ed8e7](#))

- Merge pull request #483 from ToonMeynen/patch-2

Update projects.py documentation ([2b9ae5c](#))

- Change method for getting content of snippet ([505c749](#))
- Update projects.py

Add missing attributes to file.create in order to make it work. ([629b1e1](#))

- Merge pull request #481 from max-wittig/docs/fix-typo

docs(projects): fix typo ([f3533cd](#))

- Fix the impersonation token deletion example

Fixes #476 ([c5b9676](#))

- [docs] Move notes examples in their own file

Fixes #472 ([f980707](#))

- Expose additional properties for Gitlab objects
- url: the URL provided by the user (from config or constructor)
- api\_url: the computed base endpoint (URL/api/v?)

Fixes #474 ([f09089b](#))

- Token scopes are a list ([32b399a](#))
- [docs] fix GitLab refernce for notes ([33b2b1c](#))
- Provide a basic issue template ([3d8d413](#))

- Get rid of `_sanitize_data`

It was used in one class only, no need for added complexity. (79dc1f1)

- Implement attribute types to handle special cases

Some attributes need to be parsed/modified to work with the API (for instance lists). This patch provides two attribute types that will simplify parts of the code, and fix some CLI bugs.

Fixes #443 (1940fee)

- update docs copyright years (455a8fc)
- [docs] Merge builds.rst and builds.py (78bb6b5)
- Support downloading a single artifact file

Fixes #432 (9080f69)

- pep8 fix (9cb6bbe)
- [cli] Fix listing of strings (cb8ca65)
- Add basic unit tests for v4 CLI (88391bf)
- [cli] Restore the `-help` option behavior

Fixes #381 (7c6be94)

- Add support for recursive tree listing

Fixes #452 (d35a31d)

- [cli] Allow to read args from files

With the `@/file/path` syntax (similar to curl) user can provide values from attributes in files.

Fixes #448 (748d57e)

- [docs] Commits: add an example of binary file creation

Binary files need to be encoded in base64.

Fixes #427 (c7b3f96)

- [docs] Fix the time tracking examples

Fixes #449 (4a2ae8a)

- tests: increase waiting time and hope for the best (2e51332)
- Merge pull request #426 from tardyp/readmixin

introduce RefreshMixin (ee4591d)

- introduce RefreshMixin

RefreshMixin allows to update a REST object so that you can poll on it. This is mostly useful for pipelines and jobs, but could be set on most of other objects, with unknown usecases. (3424333)

- Merge pull request #446 from jwilk-forks/spelling

Fix typos in documentation (6bcc92a)

- Fix typos in documentation (c976fec)
- [cli] `_id_attr` is required on creation (e65dfa3)
- CLI: `display_list` need to support `**kwargs` (5e27bc4)
- [cli] fix listing for json and yaml output



Fixes #438 ([4bdce7a](#))

- Merge pull request #445 from esabouraud/feature-unshare

Add support for unsharing projects with groups ([6c08266](#))

- Add support for unsharing projects to v3 API (untested) ([c8c4b42](#))
- Add support for unsharing projects to v4 API ([5fdd06e](#))
- ProjectKeys can be updated

Closes #444 ([9a30266](#))

- Require requests>=2.4.2

Closes #441 ([a7314ec](#))

- Prepare the 1.3.0 release ([10bd1f4](#))
- Add docs for pipeline schedules ([ac123df](#))
- Move the pipelines doc to builds.rst ([d416238](#))
- Merge pull request #429 from Miouge1/doc-mr-labels

Add documentation about labels update ([0cbd9c6](#))

- Add documentation about labels update ([eb5c149](#))
- pep8 fixes ([e7546de](#))
- Remove pipeline schedules from v3 (not supported) ([0a06779](#))
- Merge branch 'mlq-feature/pipeline-schedules' ([39a0429](#))
- Project pipeline jobs ([fd726cd](#))
- Project pipeline schedules ([31eb913](#))
- Update pipeline schedules code ([6a87d38](#))
- Merge pull request #420 from tardyp/patch-2

make trigger\_pipeline return the pipeline ([70c779c](#))

- Merge pull request #419 from tardyp/patch-1

Simplify the example for streamed artifacts ([6ea7ab7](#))

- fix pep8 ([8134f84](#))
- add a Simplified example for streamed artifacts

Going through an object adds a lot of complication. Adding example for unzipping on the fly ([faeb1c1](#))

- Default to API v4 ([f276f13](#))
- Gitlab can be used as context manager

Fixes #371 ([b4f0317](#))

- config: support api\_version in the global section

Fixes #421 ([29bd813](#))

- make trigger\_pipeline return the pipeline

Trigger\_pipeline returns nothing, which makes it difficult to track the pipeline being trigger.

Next PR will be about updating a pipeline object to get latest status (not sure yet the best way to do it) ([72ade19](#))

- Add Gitlab and User events support

Closes #412 ([1ca3080](#))

- Add support for getting list of user projects

Fixes #403 ([96a1a78](#))

- Add support for MR participants API

Fixes #387 ([08f19b3](#))

- Update the groups documentation

Closes #410 ([638da69](#))

- Fix wrong tag example

Fixes #416 ([e957817](#))

- Add manager for jobs within a pipeline. (#413) ([bdb6d63](#))
- Merge pull request #408 from movermeyer/patch-1

Adding the supported version badge ([5149651](#))

- Merge pull request #409 from movermeyer/patch-2

Clarifying what compatible means ([8a953c2](#))

- Clarifying what supports means ([b980c9f](#))
- Adding the supported version badge ([9253661](#))
- Prepare v1.2.0 ([3a119cd](#))
- Respect content of REQUESTS\_CA\_BUNDLE and \*\_proxy envvars

Explicitly call the requests session.merge\_environment\_settings() method, which will use some environment variables to setup the session properly.

Closes #352 ([6f50447](#))

- Add doc for search by custom attribute ([2e2a78d](#))
- Add support for user/group/project filter by custom attribute

Closes #367 ([65c64eb](#))

- Add support for project and group custom variables

implements parts of #367 ([fa52024](#))

- Add support for features flags

Fixes #360 ([f5850d9](#))

- Add support for pagesdomains

Closes #362 ([c281d95](#))

- Add supported python versions in setup.py ([6923f11](#))
- Add support for subgroups listing

Closes #390 ([928865e](#))

- Add groups listing attributes ([81c9d1f](#))
- Merge pull request #406 from ericfrederich/pagination

Allow `per_page` to be used with generators. ([79d2ca4](#))

- Allow `per_page` to be used with generators.

Fixes #405 ([3b1d1dd](#))

- Update groups tests

Group search in gitlab 10.3 requires a query string with more than 3 characters. Not sure if feature or bug, but let's handle it. ([7efbc30](#))

- Minor doc update (variables)

Fixes #400 ([70e721f](#))

- Update testing tools for /session removal ([8ad4a76](#))
- Remove now-invalid test ([5a5cd74](#))
- `ProjectFile.create()`: don't modify the input data

Fixes #394 ([0a38143](#))

- submanagers: allow having undefined parameters

This might happen in CLI context, where recursion to discover parent attributes is not required (URL gets hardcoded)

Fix should fix the CLI CI. ([6f36f70](#))

- [docs] Add a note about password auth being removed from GitLab

Provide a code snippet demonstrating how to use cookie-based authentication.

Fixes #380 ([e08d3fd](#))

- Add missing doc file ([93f1499](#))
- [docstrings] Explicitly documentation pagination arguments

Fixes #393 ([b0ce3c8](#))

- `mixins.py`: Avoid sending empty update data to `issue.save` (#389) ([0c3a6cb](#))
- Update project services docs for v4

Fixes #396 ([4e048e1](#))

- Add support for award emojis

Fixes #361 ([b33265c](#))

- Make `todo()` raise `GitlabTodoError` on error ([2167409](#))
- Add doc to get issue from iid (#321) ([b775069](#))
- Merge pull request #382 from ptomato/subscribe-response-code

Expected HTTP response for subscribe is 201 ([f624d2e](#))

- Merge pull request #374 from benjamb/typos

Fix typos in docs ([8f3b656](#))

- Update pagination docs for `ProjectCommit`

In v3 pagination starts at page 0 instead of page 1.

Fixes: #377 ([c6c0686](#))

- Expected HTTP response for subscribe is 201

It seems that the GitLab API gives HTTP response code 201 (“created”) when successfully subscribing to an object, not 200. ([0d5f275](#))

- Fix typos in docs ([7c886de](#))
- Fix link to settings API ([be386b8](#))
- Revert “Add unit tests for mixin exceptions”

This reverts commit 4ee139ad5c58006da1f9af93fdd4e70592e6daa0. ([084b905](#))

- Project pipeline jobs ([b861837](#))
- Project pipeline schedules ([34e32a0](#))
- Add support for project housekeeping

Closes #368 ([9ede652](#))

- Add unit tests for mixin exceptions ([4ee139a](#))
- Add a SetMixin

Use it for UserCustomAttribute, will be useful for {Project,Group}CustomAttribute (#367) ([a1b097c](#))

- Add support for user\_agent\_detail (issues)

<https://docs.gitlab.com/ce/api/issues.html#get-user-agent-details> ([397d677](#))

- Merge pull request #369 from Lujeni/fix\_typo\_projects\_documentation

[docs] Bad arguments in projects file documentation ([ad35482](#))

- [docs] Bad arguments in projects file documentation ([5ee4e73](#))
- update user docs with gitlab URLs ([29d8d72](#))
- Add support for user activities ([44a7ef6](#))
- generate coverage reports with tox ([7fadf46](#))
- typo ([2d689f2](#))
- Add support for impersonation tokens API

Closes #363 ([8fec612](#))

- Add missing mocking on unit test ([700e84f](#))
- Add support for oauth and anonymous auth in config/CLI ([0732826](#))
- Rework authentication args handling
- Raise exceptions when conflicting arguments are used
- Build the auth headers when instantiating Gitlab, not on each request
- Enable anonymous Gitlab objects (#364)

Add docs and unit tests ([e9b1583](#))

- Remove deprecated objects/methods ([ba6e09e](#))
- OAuth token support (#357) ([c30121b](#))
- Merge branch ‘master’ of github.com:python-gitlab/python-gitlab ([3bc3e60](#))
- Merge pull request #365 from jeromeroberth/master

[doc] Fix project.triggers.create example with v4 API ([30b1c03](#))

- [doc] Fix project.triggers.create example with v4 API ([6c5ee84](#))
- Merge pull request #342 from matejzero/mattermost

Add mattermost service support ([82897b7](#))

- Add users custom attributes support ([4fb2e43](#))
- 1.1.0 release ([32f7e17](#))
- improve comment in release notes ([9eff543](#))
- [doc] Add sample code for client-side certificates

Closes #23 ([fa89746](#))

- Module's base objects serialization (#359)

Make gitlab objects serializable

With current implementation of API v3 and v4 support, some instances have properties of type module and are not serializable. Handle these properties manually with setstate and getstate methods. ([226e6ce](#))

- Pagination generators: expose more information

Expose the X-\* pagination attributes returned by the Gitlab server when requesting lists.

Closes #304 ([38d4467](#))

- Add a contributed Dockerfile

Thanks oupala!

Closes #295 ([fba7730](#))

- Fix the CLI for objects without ID (API v4)

Fixes #319 ([9dd410f](#))

- Update the repository\_blob documentation

Fixes #312 ([d415cc0](#))

- Add support for wiki pages ([5082879](#))
- Move the ProjectManager class for readability ([4744200](#))
- Add support for GPG keys

Closes #355 ([d0c4118](#))

- Add support for group milestones

Closes #349 ([aba713a](#))

- Move group related code for readability ([cf6767c](#))
- Update the ssl\_verify docstring ([4c3aa23](#))
- Project: add support for printing\_merge\_request\_link\_enabled attr

Closes #353 ([3a8c480](#))

- ProjectFileManager: custom update() method

Closes #340 ([fe5805f](#))

- Document the Gitlab session parameter

Provide a proxy setup example.

Closes #341 ([1b5d480](#))

- [docs] document `get_create_attrs` in the API tutorial ([b23e344](#))
- Change `ProjectUser` and `GroupProject` base class

python-gitlab shouldn't try to provide features that are not existing in the Gitlab API: `GroupProject` and `ProjectUser` objects should not provide unsupported API methods (no `get`, no `create`, no `update`).

This Closes #346 by making explicit that we don't support these non-existent methods. ([8c9ad29](#))

- Remove support for "constructor types" in v4

In v3 we create objects from json dicts when it makes sense. Support for this feature has not been kept in v4, and we didn't get requests for it so let's drop the `_constructor_types` definitions. ([32ea62a](#))

- Snippet notes support all the CRUD methods

Fixes #343 ([dc504ab](#))

- Add mattermost service support ([b5e6a46](#))
- `ProjectFileManager.create`: handle / in file paths

Replace / with `%2F` as is done in other methods.

Fixes #339 ([9d0a479](#))

- Add support for listing project users

<https://docs.gitlab.com/ce/api/projects.html#get-project-users>

Closes #328 ([d6fa94e](#))

- [docs] improve the labels usage documentation

Closes #329 ([5945537](#))

- Drop leftover `pdb` call ([316754d](#))
- Tags release description: support / in tag names ([f3f300c](#))
- [docs] update the file upload samples

Closes #335 ([72664c4](#))

- Make the `delete()` method handle / in ids

Replace the / with the HTTP `%2F` as is done with other methods.

Closes #337 ([8764903](#))

- Fix trigger variables in v4 API (#334)

Fix trigger variables in v4 API

Close #333 ([ac430a3](#))

- Prepare the 1.0.2 release ([9e09cf6](#))
- `ProjectFile`: handle / in path for `delete()` and `save()`

Fixes #326 ([05656bb](#))

- Properly handle the labels attribute in `ProjectMergeRequest`

This should have made it into `e09581fc` but something went wrong (probably a PEBCAK).

Closes #325 ([69f1045](#))

- [docs] remove example usage of submanagers

Closes #324 ([9484106](#))

- 1.0.1 release ([e5f59bd](#))
- Add missing doc file ([a346f92](#))
- Fix a couple listing calls to allow proper pagination

Project.repository\_tree and Project.repository\_contributors return lists, so use http\_list to allow users to use listing features such as all=True.

Closes #314 ([80351ca](#))

- CommitStatus: sha is parent attribute

Fixes #316 ([7d6f3d0](#))

- Merge pull request #317 from mion00/patch-1

Fix http\_get method in get artifacts and job trace ([d321847](#))

- Fix http\_get method in get artifacts and job trace ([bb5a1df](#))
- exception message: mimic v3 API ([05da7ba](#))
- Exceptions: use a proper error message ([e35563e](#))
- Fix the labels attrs on MR and issues

Fixes #306 ([e09581f](#))

- Fix password authentication for v4

Fixes #311 ([89bf53f](#))

- Merge pull request #309 from mkobit/patch-1

Minor typo fix in “Switching to v4” documentation ([9d6036c](#))

- Minor typo fix in “Switching to v4” documentation ([5841070](#))
- adds project upload feature (#239) ([29879d6](#))
- Merge pull request #307 from RobberPhex/fix-tag-api

Fix tag api ([fd40fce](#))

- add list method ([b537b30](#))
- GitlabError filled by response ([4b36786](#))
- Tag can get by id ([cc249ce](#))
- Update changelog, release notes and authors for v1.0 ([3d8df3c](#))
- Switch the version to 1.0.0

The v4 API breaks the compatibility with v3 (at the python-gitlab level), but I believe it is for the greater good. The new code is way easier to read and maintain, and provides more possibilities.

The v3 API will die eventually. ([670217d](#))

- pep8 fix ([d0e2a15](#))
- Improve the docs to make v4 a first class citizen ([60efc83](#))
- [v4] fix CLI for some mixin methods ([0268fc9](#))
- FIX Group.transfer\_project ([947feaf](#))

- tests: faster docker shutdown

Kill the test container violently, no need to wait for a proper shutdown. ([d8db707](#))

- tests: default to v4 API ([0099ff2](#))
- Add support for protected branches

This feature appeared in gitlab 9.5.

Fixes #299 ([c99e399](#))

- Merge branch 'group-variables' ([fcccfd](#))
- Add support for group variables ([eb191df](#))
- [v4] More python functional tests ([0e0d4ae](#))
- update tox/travis for CLI v3/4 tests ([311464b](#))
- [tests] Use -n to not use a venv ([5210956](#))
- [v4] Make sudo the first argument in CLI help ([022a0f6](#))
- Fix the v4 CLI tests (id/iid) ([b0af946](#))
- [v4] Fix the CLI for project files ([cda2d59](#))
- Make CLI tests work for v4 as well ([f00562c](#))
- [v4] Use - instead of \_ in CLI legacy output

This mimics the v3 behavior. ([f762cf6](#))

- make v3 CLI work again ([59550f2](#))
- CLI: yaml and json outputs for v4

Verbose mode only works with the legacy output. Also add support for filtering the output by defining the list of fields that need to be displayed (yaml and json only). ([abade40](#))

- [v4] CLI support is back ([9783207](#))
- [v4] drop unused CurrentUserManager.credentials\_auth method ([a4f0c52](#))
- README: mention v4 support ([b919555](#))
- Update the objects doc/examples for v4 ([4057644](#))
- Fix Args attribute in docstrings ([80eab7b](#))
- [v4] Fix getting projects using full namespace ([72e783d](#))
- Fix URL for branch.unprotect ([279704f](#))
- on\_http\_error: properly wrap the function

This fixes the API docs. ([4ed22b1](#))

- [v4] fix the project attributes for jobs

builds\_enabled and public\_builds are now jobs\_enabled and public\_jobs. ([d1e7cc7](#))

- Fix Gitlab.version()

The method was overwritten by the result of the call. ([45c4aaf](#))

- Merge pull request #294 from wayfair/feature\_internal\_cert\_configuration

Support SSL verification via internal CA bundle ([0e70dd9](#))



- Support SSL verification via internal CA bundle
- Also updates documentation
- See issues #204 and #270 ([4af4748](#))
- Merge pull request #278 from asphaltboy/link-docs-gitlab-token

Docs: Add link to gitlab docs on obtaining a token ([657f011](#))

- update tox/travis test envs ([759f6ed](#))
- Merge branch 'rework\_api' ([3ccdec0](#))
- Docs: Add link to gitlab docs on obtaining a token

I find these sort of links very user friendly ([9b8b806](#))

- Make the project services work in v4 ([2816c1a](#))
- Fix v3 tests ([eee39a3](#))
- Update tests for list() changes ([7592ec5](#))
- remove py3.6 from travis tests ([217dc3e](#))
- Restore the previous listing behavior

Return lists by default : this makes the explicit use of pagination work again.

Use generators only when `as_list` is explicitly set to `False`. ([5a4aafb](#))

- functional tests for v4

Update the python tests for v4, and fix the problems raised when running those tests. ([d7c7911](#))

- Restore correct exceptions

Match the exceptions raised in v3 for v4.

Also update the doc strings with correct information. ([c15ba3b](#))

- Merge pull request #282 from velvetz7/docs\_typo

Fixed repository\_compare examples ([e87835f](#))

- Changed attribution reference ([73be8f9](#))
- Fix merge\_when\_build\_succeeds attribute name

Fixes #285 ([374a6c4](#))

- Fix merge\_when\_build\_succeeds attribute name

Fixes #285 ([67d9a89](#))

- Merge branch 'master' into rework\_api ([274b3bf](#))
- Merge pull request #286 from jonafato/python3.6

Declare support for Python 3.6 ([cb8c1a1](#))

- Merge pull request #287 from guyzmo/features/dependency\_injection

Added dependency injection support for Session ([46b7f48](#))

- Added dependency injection support for Session

fixes #280

Signed-off-by: Guyzmo <guyzmo+github+pub@m0g.net> ([116e3d4](#))

- Declare support for Python 3.6

Add Python 3.6 environments to `tox.ini` and `.travis.yml`. ([4c916b8](#))

- fixed repository\_compare examples ([261db17](#))
- remove useless attributes ([fe3a06c](#))
- Refactor the CLI

v3 and v4 CLI will be very different, so start moving things in their own folders.

For now v4 isn't working at all. ([e3d50b5](#))

- Add missing doc files ([fd5ac4d](#))
- typo ([67be226](#))
- build submanagers for v3 only ([ea79bdc](#))
- Fix GroupProject constructor ([afe4b05](#))
- Merge pull request #276 from elisarver/patch-1

Missing expires\_at in GroupMembers update ([f19681f](#))

- minor doc updates ([6e5a6ec](#))
- Fix changelog and release notes inclusion in sdist ([1922cd5](#))
- Rework documentation ([1a7f672](#))
- Missing expires\_at in GroupMembers update

CreateAttrs was set twice in GroupMember due to possible copy-paste error. ([d41e972](#))

- Remove unused future.division import

We don't do math. ([2a0afc5](#))

- add support for objects delete() ([32c704c](#))
- pep8 fixes ([26c0441](#))
- Document switching to v4 ([186e11a](#))
- 0.10 is old history: remove the upgrade doc ([76e9b12](#))
- 0.21.2 release ([19f1b1a](#))
- Add laziness to get()

The goal is to create empty objects (no API called) but give access to the managers. Using this users can reduce the number of API calls but still use the same API to access children objects.

For example the following will only make one API call but will still get the result right:

```
gl.projects.get(49, lazy=True).issues.get(2, lazy=True).notes.list()
```

This removes the need for more complex managers attributes (e.g. `gl.project_issue_notes`) ([61fba84](#))

- Drop invalid doc about raised exceptions ([197ffd7](#))
- Add new event types to ProjectHook ([a0f215c](#))
- Merge pull request #272 from astronouth7303/patch-1

Add new event types to ProjectHook ([4ce2794](#))

- Fix a few remaining methods ([3488c5c](#))

- Add new event types to ProjectHook

These are being returned in the live API, but can't set them. ([1a58f7e](#))

- tests for objects mixins ([68f4114](#))
- Add tests for managers mixins ([b776c5e](#))
- Basic test for GitlabList ([f2c4a6e](#))
- Fix GitlabList.**len** ([15511bf](#))
- Unit tests for REST\* classes ([0d94ee2](#))
- Merge branch 'rework\_api' of github.com:python-gitlab/python-gitlab into rework\_api ([a5b39a5](#))
- Fixed spelling mistake (#269) ([2b1e0f0](#))
- Tests and fixes for the http\_\* methods ([ff82c88](#))
- make the tests pass ([f754f21](#))
- Migrate all v4 objects to new API

Some things are probably broken. Next step is writing unit and functional tests.

And fix. ([f418767](#))

- Simplify SidekiqManager ([0467f77](#))
- New API: handle gl.auth() and CurrentUser\* classes ([a1c9e2b](#))
- Add support for managers in objects for new API

Convert User\* to the new REST\* API. ([a506902](#))

- pep8 ([9fbdb94](#))
- Move the mixins in their own module ([fb5782e](#))
- Rework the manager and object classes

Add new RESTObject and RESTManager base class, linked to a bunch of Mixin class to implement the actual CRUD methods.

Object are generated by the managers, and special cases are handled in the derived classes.

Both ways (old and new) can be used together, migrate only a few v4 objects to the new method as a POC.

TODO: handle managers on generated objects (have to deal with attributes in the URLs). ([993d576](#))

- pep8 again ([d809fef](#))
- Add lower-level methods for Gitlab()

Multiple goals:

- Support making direct queries to the Gitlab server, without objects and managers.
- Progressively remove the need to know about managers and objects in the Gitlab class; the Gitlab should only be an HTTP proxy to the gitlab server.
- With this the objects gain control on how they should do requests. The complexities of dealing with object specifics will be moved in the object classes where they belong. ([c5ad540](#))
- import urlencode() from six.moves.urllib.parse instead of from urllib (#268)

Fixes AttributeError on Python 3, as urlencode function has been moved to urllib.parse module.

six.moves.urllib.parse.urlencode() is an py2.py3 compatible alias of urllib.parse.urlencode() on Python 3, and of urllib.urlencode() on Python 2. (88900e0)

- Tests and fixes for the http\_\* methods (904c9fa)
- make the tests pass (d0a9334)
- Migrate all v4 objects to new API

Some things are probably broken. Next step is writing unit and functional tests.

And fix. (6be990c)

- Simplify SidekiqManager (230b567)
- New API: handle gl.auth() and CurrentUser\* classes (7193034)
- Add support for managers in objects for new API

Convert User\* to the new REST\* API. (5319d0d)

- pep8 (29cb0e4)
- Move the mixins in their own module (0748c89)
- Rework the manager and object classes

Add new RESTObject and RESTManager base class, linked to a bunch of Mixin class to implement the actual CRUD methods.

Object are generated by the managers, and special cases are handled in the derived classes.

Both ways (old and new) can be used together, migrate only a few v4 objects to the new method as a POC.

TODO: handle managers on generated objects (have to deal with attributes in the URLs). (29e0bae)

- pep8 again (b7298de)
- Add lower-level methods for Gitlab()

Multiple goals:

- Support making direct queries to the Gitlab server, without objects and managers.
- Progressively remove the need to know about managers and objects in the Gitlab class; the Gitlab should only be an HTTP proxy to the gitlab server.
- With this the objects gain control on how they should do requests. The complexities of dealing with object specifics will be moved in the object classes where they belong. (7ddb5e)
- Prepare for v4 API testing (38bff3e)
- [v4] Make project issues work properly
- Use iids instead of ids
- Add required duration argument for time\_estimate() and add\_spent\_time() (ac3aef6)
- Remove extra\_attrs argument from \_raw\_list (unneded) (f3b2855)
- pep8 fix (0663184)
- Fix python functional tests (f733ffb)
- [v4] Make MR work properly
- Use iids instead of ids (Fixes #266)

- Add required duration argument for `time_estimate()` and `add_spent_time()` ([1ab9ff0](#))
- install doc: use sudo for system commands

Fixes #267 ([a3b8858](#))

- Changelog update ([4bf251c](#))
- Update API docs for v4 ([1ac66bc](#))
- Fix broken docs examples ([746846c](#))
- Prepare the 0.21.1 release ([3ff7d9b](#))
- move changelog and release notes at the end of index ([d75e565](#))
- [v4] Fix the jobs manager attribute in Project ([4f1b952](#))
- Prepare the 0.21 release ([cd9194b](#))
- update copyright years ([ba41e5e](#))
- [v4] Add support for dockerfiles API ([0aa38c1](#))
- [v4] Builds have been renamed to Jobs ([deac5a8](#))
- [v4] Triggers: update object
- Add support for the description attribute
- Add `take_ownership` support
- Triggers now use `id` as identifier ([29e735d](#))
- add a warning about the upcoming v4 as default ([5ea7f84](#))
- Add v4 support to docs ([f2b94a7](#))
- Update release notes for v4 ([627a6aa](#))
- pep8 fix ([03ac8da](#))
- Merge branch 'v4\_support' ([766efe6](#))
- [v4] User: drop the manager filters ([dcb501](#))
- [v4] Remove deprecated objects methods and classes ([8e4b65f](#))
- Deprecate parameter related methods in `gitlab.Gitlab`

These methods change the auth information and URL, and might have some unwanted side effects.

Users should create a new Gitlab instance to change the URL and authentication information. ([7ac1e4c](#))

- pep8 fixes ([441244b](#))
- [v4] Users confirm attribute renamed `skip_confirmation` ([cd98903](#))
- [v4] repository tree: `s/ref_name/ref/` ([2dd84e8](#))
- [v4] Try to make the files `raw()` method work ([0d1ace1](#))
- [v4] Update triggers endpoint and attrs ([0c3fe39](#))
- [v4] Milestones: `iid => iids` ([449f607](#))
- 202 is expected on some delete operations ([b9eb10a](#))
- [v4] Rename the `ACCESS*` variables ([cd18aee](#))
- [v4] `GroupManager.search` is not needed ([9a66d78](#))

- [v4] Rename the visibility attribute

Also change the value of the `VISIBILITY_*` consts, and move them to the `objects` module root.

TODO: deal the numerical value used by v3. ([27c1e95](#))

- [v4] Remove public attribute for projects ([9b625f0](#))
- [v4] MR `s/build/pipeline/` in attributes ([9de53bf](#))
- [v4] Rename `branch_name` to `branch` ([8b75bc8](#))
- [v4] Update (un)subscription endpoints ([90c8958](#))
- [v4] Update user (un)block HTTP methods ([5c8cb29](#))
- [v4] Drop `ProjectKeyManager.enable()` ([41f141d](#))
- [v4] Add `projects.list()` attributes

All the `ProjectManager` filter methods can now be handled by `projects.list()`. ([e789cee](#))

- [v4] Update project fork endpoint ([6684c13](#))
- [v4] Update the licenses templates endpoint ([206be8f](#))
- [v4] Update project unstar endpoint ([76ca234](#))
- [v4] Update project keys endpoint ([d71800b](#))
- [v4] Update iid attr for issues and MRs ([9259041](#))
- [v4] `projects.search()` has been removed ([af70ec3](#))
- [v4] Drop teams support ([17dffdf](#))
- Add missing `base.py` file ([3f7e5f3](#))
- Duplicate the `v3/objects.py` in `v4/`

Using imports from `v3/objects.py` in `v4/objects.py` will have side effects. Duplication is not the most elegant choice but `v4` is the future and `v3` will die eventually. ([3aa6b48](#))

- Reorganise the code to handle `v3` and `v4` objects

Having objects managing both versions will only make the code more complicated, with lots of tests everywhere. This solution might generate some code duplication, but it should be maintainable. ([e853a30](#))

- Update Gitlab `init` docstring ([f373885](#))
- [v4] Update project search API
- `projects.search()` is not implemented in `v4`
- add the ‘search’ attribute to `projects.list()` ([deecf17](#))
- Initial, non-functional `v4` support ([c02dabd](#))
- Add ‘search’ attribute to `projects.list()`

`projects.search()` has been deprecated by Gitlab ([ce3dd0d](#))

- Update URLs to reflect the github changes ([7def297](#))
- Fixed `repository_tree` and `repository_blob` path encoding (#265) ([f3dfa6a](#))
- MR: add support for time tracking features

Fixes #248 ([324f81b](#))

- Available services: return a list

The method returned a JSON string, which made no sense...

Fixes #258 ([5b90061](#))

- Make GroupProjectManager a subclass of ProjectManager

Fixes #255 ([468246c](#))

- Add support for nested groups (#257) ([5afeeb7](#))
- Add support for priority attribute in labels

Fixes #256 ([5901a1c](#))

- Support milestone start date (#251) ([9561b81](#))
- Merge pull request #249 from guikcd/patch-1

docs: s/correspnding/corresponding/ ([f03613d](#))

- s/correspnding/corresponding/ ([e5c7246](#))
- Feature/milestone merge requests (#247)

Added milestone.merge\_requests() API ([34c7a23](#))

- Update User options for creation and update

Fixes #246 ([9d80699](#))

- Merge pull request #245 from TimNN/mr-time-stats

Add time\_stats to ProjectMergeRequest ([f05a24b](#))

- add time\_stats to ProjectMergeRequest ([63a11f5](#))
- Prepare 0.20 release ([c545504](#))
- Merge pull request #244 from gpocentek/issue/209

Make GroupProject inherit from Project ([20d6678](#))

- Stop listing if recursion limit is hit (#234) ([989f3b7](#))
- Provide API wrapper for cherry picking commits (#236) ([22bf128](#))
- add 'delete source branch' option when creating MR (#241) ([8677f1c](#))
- Merge pull request #243 from DmytroLitvinov/fix/bug\_242

Change to correct logic of functions ([cc4fe78](#))

- Change to correct logic of functions ([889bbe5](#))
- Make GroupProject inherit from Project

Fixes #209 ([380bcc4](#))

- Implement pipeline creation API (#237) ([8c27e70](#))
- Properly handle extra args when listing with all=True

Fixes #233 ([3b38844](#))

- Add support for merge request notes deletion

Fixes #227 ([e39d7ea](#))

- Add DeployKey{,Manager} classes

They are the same as Key and KeyManager but the name makes more sense.

Fixes #212 ([a3f2ab1](#))

- Include chanlog and release notes in docs ([ea0759d](#))
- Minor changelog formatting update ([99e6f65](#))
- Make sure that manager objects are never overwritten

Group.projects (manager) can be replaced by a list of Project objects when creating/updating objects. The GroupObject API is more consistent and closer to the GitLab API, so make sure it is always used.

Fixes #209 ([35339d6](#))

- Changelog: improvements. Fixes #229 (#230)
- change indentation so bullet points are not treated as quote
- add links to releases
- add dates to releases
- use releases as headers ([37ee7ea](#))
- Time tracking (#222)
- Added gitlab time tracking features
- get/set/remove estimated time per issue
- get/set/remove time spent per issue
- Added documentation for time tracking functions ([92151b2](#))
- 0.19 release ([cd69624](#))
- {Project,Group}Member: support expires\_at attribute

Fixes #224 ([a273a17](#))

- Handle settings.domain\_whitelist, partly

The API doesn't like receiving lists, although documentation says it's what's expected. To be investigated.

This fixes the tests. ([41ca449](#))

- Merge pull request #216 from ExodusIntelligence/hotfix-issue\_due\_date-215

added due\_date attribute to ProjectIssue ([1e0ae59](#))

- Merge pull request #220 from alexwidener/master

Added pipeline\_events to ProjectHook attrs ([19c7784](#))

- Added pipeline\_events to ProjectHook attrs

Ran tests, all passed. ([3f98e03](#))

- fixes gpocentek/python-gitlab#215 ([58708b1](#))
- document the dynamic aspect of objects ([2f274bc](#))
- Deploy keys: rework enable/disable

The method have been moved to the keys manager class as they don't make sens at all on the project keys themselves.

Update doc and add tests.

Fixes #196 ([492a751](#))



- Merge pull request #210 from comel/services-1

Add builds-email and pipelines-email services ([dad1345](#))

- Add builds-email and pipelines-email services ([5cfa6fc](#))
- deploy keys doc: fix inclusion ([1d827bd](#))
- Merge branch 'nutztherookie-patch-1' ([5352c36](#))
- Fix install doc

it's just confusing that it would say "pip" again :) ([4fba82e](#))

- Add support for commit creation

Fixes #206 ([ee666fd](#))

- Add support for project runners

This API allows to enable/disable specific runners for a project, and to list the project associated runners.

Fix #205 ([04435e1](#))

- Support the scope attribute in runners.list() ([de0536b](#))
- Merge pull request #203 from vilhelmen/master

Update project.archive() docs ([c538de7](#))

- Update project.archive() docs ([e7560a9](#))
- Some objects need getRequires to be set to False ([05b3abf](#))
- Forbid empty id for get()

Unless the class explicitly defines it's OK (getRequiresId set to True). ([18415fe](#))

- prepare the 0.18 release ([8028ec7](#))
- sudo: always use strings

The behavior seems to have changed on recent gitlab releases and providing an ID as int doesn't work anymore. Using a string seems to make things work again.

Fixes #193 ([d6c87d9](#))

- Update known attributes for projects

Fixes #181 ([3804661](#))

- Fix duplicated data in API docs

Fixes #190 ([73990b4](#))

- Add functional tests for Snippet ([d3d8baf](#))
- Snippet: content() -> raw()

Using the content() method causes conflicts with the API content attribute. ([064e2b4](#))

- SnippetManager: all() -> public()

Rename the method to make what it does more explicit. ([7453895](#))

- [docs] Add doc for snippets ([bd7d2f6](#))
- Merge branch 'guyzmo-features/personal\_snippets' ([0a4d40e](#))
- Merge branch 'features/personal\_snippets' of <https://github.com/guyzmo/python-gitlab> into guyzmo-features/personal\_snippets ([26c8a0f](#))

- [CLI] Fix wrong use of arguments

The previous change removed undefined arguments from the args dict, don't try to use possibly missing arguments without a fallback value. ([b05c0b6](#))

- [CLI] ignore empty arguments

Gitlab 8.15 doesn't appreciate arguments with None as value. This breaks the python-gitlab CLI.

Fixes #199 ([f4fcf45](#))

- [docs] artifacts example: open file in wb mode

Fixes #194 ([35c6bbb](#))

- [docs] update pagination section

First page is page 1.

Fixes #197 ([d86ca59](#))

- Added support for Snippets (new API in Gitlab 8.15)

cf [Gitlab-CE MR !6373](#)

Signed-off-by: Guyzmo <guyzmo+github@m0g.net> ([6022dfe](#))

- Merge pull request #192 from galet/gitlab-8.14-jira

Fix JIRA service editing for GitLab 8.14+ ([15d3362](#))

- Add jira\_issue\_transition\_id to the JIRA service optional fields ([f7e6482](#))
- Fix JIRA service editing for GitLab 8.14+

GitLab simplified the configuration for JIRA service and renamed most of the fields. To maintain backward compatibility all mandatory fields were moved to optional section. ([343c131](#))

- prepare 0.17 release ([932ccd2](#))
- Merge pull request #186 from localmed/fix-should-remove-source-branch

Fix `should_remove_source_branch` ([39288c8](#))

- Rework requests arguments
- Factorize the code
- Don't send empty auth information to requests (Fixes #188) ([6e5734b](#))
- Fix `should_remove_source_branch` ([ac2bf24](#))
- Add support for triggering a new build

Fixes #184 ([de05dae](#))

- CLI: add support for project all -all

Rework the extra opts definition to allow setting typed arguments.

Fixes #153 ([f5f734e](#))

- Merge pull request #183 from GregoryEAllen/master

Please add these missing attrs ([840cb89](#))

- Merge pull request #2 from GregoryEAllen/GregoryEAllen-patch-2

Add attr 'updated\_at' to ProjectIssue ([f290b2b](#))

- Merge pull request #1 from GregoryEAllen/GregoryEAllen-patch-1

Add attr 'created\_at' to ProjectIssueNote ([14e7ccd](#))

- Add attr 'updated\_at' to ProjectIssue ([a25fef5](#))
- Add attr 'created\_at' to ProjectIssueNote ([5b24122](#))
- Add support for templates API

Add gitlab CI and gitignores APIs

Rework the templates/license API docs ([570e75d](#))

- Restore the Gitlab.user\_projects manager ([463893f](#))
- Make the manager objects create mor dynamic

For the gitlab.Gitlab object make the detection of “submanagers” more dynamic. This will avoid duplication of definitions.

Update the sphinx extension to add these managers in the list of attributes. ([81e1c13](#))

- Implement merge requests diff support ([92180e4](#))
- Remove deprecated methods

Also deprecate {un,}archive\_() in favor of {un,}archive().

Fix #115 ([c970a22](#))

- Add a 'report a bug' link on doc ([258aab4](#))
- Implement **repr** for gitlab objects

Fix #114 ([1833117](#))

- Sphinx ext: factorize the build methods ([7cfbe87](#))
- Fix tuples definition ([6c0a3d9](#))
- pep8 fix ([68c09b7](#))
- API docs: add managers doc in GitlabObject's ([20143f5](#))
- Build managers on demand on GitlabObject's ([e57a86b](#))
- Fix docstring for http\_{username,password} ([11c1425](#))
- Rework the API documentation

Update the sphinx extension to add method definition in the docs. This makes the documentation a bit more usable.

Hide attributes that should not have been exposed. They still exist in the code but their documentation doesn't make much sense. ([be83ff9](#))

- fix line too long ([9f7f45f](#))
- Move deploy key enable/disable to the object

To keep things consistent with other objects, action methods are available on the object itself, not the manager. ([c17ecc0](#))

- Merge branch 'master-project-deploy-keys' of <https://github.com/Asher256/python-gitlab> into Asher256-master-project-deploy-keys ([0c1817f](#))
- add missing files in MANIFEST.in ([12fca84](#))
- ProjectHook: support the token attribute

Fix #170 ([cd5f849](#))

- Project deploy key response code = 201 ([6bedfc3](#))
- Fixing the response and project\_id argument ([72d982b](#))
- Documentation for enable/disable deploy key functions ([f9cb718](#))
- New exception for ProjectKey.enable\_deploy\_key and disable\_deploy\_key ([61d4cac](#))
- enable/disable deploy key methods moved to the class ProjectKey() ([fc5c52c](#))
- Delete is used for '/projects/%s/deploy\_keys/%s/disable' ([67aa795](#))
- Feature: enable / disable the deploy key in a project ([6310d71](#))
- Merge pull request #178 from cgumpert/master

fix bug when retrieving changes for merge request ([4689e73](#))

- fix bug when retrieving changes for merge request

Erroneously a merge request would return its commit when being queried for its changes. ([34d6050](#))

- Merge pull request #172 from xiaopeng163/master

edit doc badge url in README.rst ([ff32514](#))

- edit doc badge url

Signed-off-by: Peng Xiao <xiaoquwl@gmail.com> ([376acda](#))

- Don't overwrite attributes returned by the server

Fixes #171 ([2c7a999](#))

- README typo ([62058f6](#))
- Add support for the notification settings API ([b15f17b](#))
- Add support for broadcast messages API ([6d3450c](#))
- Add support for Gitlab.version() ([c185fe2](#))
- Add support for boards API

This is not fully usable because the gitlab API has some limitations:

- not possible to create boards programmatically
- not possible to get labels ID (<https://gitlab.com/gitlab-org/gitlab-ce/issues/23448>) ([f332907](#))
- Merge pull request #169 from hakkeroid/allow-iid-parameter-to-request-distinct-objects

Convert response list to single data source for iid requests ([20fdbe8](#))

- Convert response list to single data source for iid requests ([23b5b6e](#))
- Merge pull request #168 from hakkeroid/pass\_kwargs\_to\_object\_factory

Pass kwargs to object factory ([9da5d69](#))

- Add .tox to ignore to respect default tox settings ([cc151f3](#))
- Pass kwargs to the object factory ([ef69808](#))
- Merge pull request #165 from hakkeroid/patch-1

Fix ProjectBuild.play raising error even on success ([5f444e4](#))

- Fix ProjectBuild.play raises error on success

Running `play` on `ProjectBuild` raises a `GitlabBuildPlayError` although the request was successful. It looks like gitlab returns a 200-OK instead of 201-CREATED response and as such always raises an exception. (0793271)

- README: add badges for pypi and RTD (945cc66)
- Set version to 0.16 (46ea44a)
- prepare the 0.16 release (117f7f5)
- Implement `ProjectBuild.play()` (673dc36)
- Merge pull request #159 from JonathonReinhart/158-erase-build

Add `ProjectBuild.erase()` (d4a24a5)

- Update docs to use `ProjectBuild.erase()` (3b3930b)
- Add `ProjectBuild.erase()`

We can't use the existing `delete()` functionality, because GitLab uses `POST /projects/:id/builds/:build_id/erase` to erase a build. Instead of overriding `delete()`, we add a separate `erase()` method to keep the naming consistent, and allow potentially more fine-grained operations in the future.

- <https://docs.gitlab.com/ce/api/builds.html#erase-a-build> (c2f45e9)
- Workaround gitlab setup failure in tests

While running the functional tests in a venv, the token download sometimes fail. Try to get it multiple times before failing. (7d424ae)

- rework travis and tox setup (3371e0e)
- Use the plural `merge_requests` URL everywhere

This breaks compatibility with older gitlab versions but maintaining support for changed APIs is just too complex and time consuming. See issue #139 if you need a workaround.

Fixes #157 (cb30dd1)

- Fix examples for file modification

Fixes #156 (d09eaa0)

- Merge branch 'master' of github.com:gpocentek/python-gitlab (6f7e499)
- Add support for `-all` in CLI

Fixes #153 (5860421)

- Merge pull request #155 from rafaeleyng/add-only\_allow\_merge\_if\_build\_succeeds

add `only_allow_merge_if_build_succeeds` option to project objects (26d97a7)

- break lines too long (608ebbd)
- add `only_allow_merge_if_build_succeeds` option to project objects (94932a0)
- Merge pull request #154 from derek-austin/patch-1

Create a project in a group (4390afb)

- Create a project in a group

Just a sketch, feel free to toss it and do it in the right way. (b057a94)

- Merge pull request #150 from vilhelmen/master

Add branch protection notes (1e1d467)

- Brief branch protection notes

You can pass `developers_can_push` and `developers_can_merge` to the `protect` function. Handy! ([8a560c6](#))

- Merge pull request #147 from derek-austin/derek-austin-patch-1

Missing coma concatenates array values ([ddba752](#))

- Missing coma concatenates array values

`'enabled_git_access_protocolgravatar_enabled'` were two distinct values in `ApplicationSettings.optionalUpdateAttrs`. ([02a8bf4](#))

- Merge branch 'master' of github.com:gpoctek/python-gitlab ([53f9322](#))
- Merge pull request #146 from galet/add-api-url-to-jira-params

JIRA service - add `api_url` to optional attributes ([d83b838](#))

- JIRA service - add `api_url` to optional attributes

The `api_url` attribute is mandatory at least since GitLab 8.11. Otherwise server returns `gitlab.exceptions.GitlabUpdateError: 400: 400 (Bad request) "api_url" not given`.

Keep it as optional to maintain backward compatibility with older GitLab versions. ([c8c43ee](#))

- Merge branch 'master' of github.com:gpoctek/python-gitlab ([2ac42d6](#))
- Merge pull request #144 from koyaan/master

Add the ability to fork to a specific namespace ([449830f](#))

- fix doc ([0c1c894](#))
- add doc ([648dc86](#))
- fix pep8 ([a692d59](#))
- Add the ability to fork to a specific namespace ([4852462](#))
- 0.15.1 release ([90ad2de](#))
- Properly fix `_raw_list` ([dc3dcd1](#))
- 'path' is an existing gitlab attr, don't use it

Use `path_` instead of `path` as parameter name for methods using it. Otherwise it might conflict with `GitlabObject` attributes. ([b815f3a](#))

- Fix and test pagination

Fixes #140 ([c08c913](#))

- minor RST fix ([47cb278](#))
- bump version and update changelog ([79f46c2](#))
- Add support for project deployments ([8d7faf4](#))
- Add support for access requests ([40db4cd](#))
- Add support for project pipelines ([8257400](#))
- Add support for project services API ([ef2dbf7](#))
- Remove unused `ProjectTagReleaseManager` class ([ded5258](#))
- Fix `canGet` attribute (typo) ([0178f3d](#))
- Remove `_get_list_or_object()` and its tests ([451c174](#))
- Let `_data_for_gitlab` return python data ([a8f6fdd](#))

- Refactor the Gitlab class

Make use of the *raw\** methods in the CRUD methods. ([fe96edf](#))

- fix pep8 test ([e0d226b](#))
- Remove method marked as deprecated 7 months ago ([438dc2f](#))
- Add copyright header to utils.py ([83cb8c0](#))
- Move the constants at the gitlab root level ([2ced9d0](#))
- Add sidekiq metrics support ([23b2a30](#))
- implement the todo API ([131739f](#))
- Update the ApplicationSettings attributes ([1c53ecb](#))
- Fix fork creation documentation

Fixes #136 ([baa09fe](#))

- Run more tests in travis

Travis has some limitations so this patch sets up some tricks to run the functional tests. ([f82f962](#))

- ignore pep8 error ([fa75571](#))
- add a basic HTTP debug method ([c9915a4](#))
- Update changelog/authors/version for 0.14 ([e4624c9](#))
- MR merge(): update the object ([799b593](#))
- MR (un)subscribe: don't fail if state doesn't change ([d7967c6](#))
- Handle empty messages from server in exceptions ([178bfb7](#))
- MR: fix updates ([4a73b85](#))
- fix labels deletion example ([71edeeb](#))
- Fix the listing of some resources

The parent ID wasn't available in the generated objects, leading to exceptions when trying to use specific methods for these objects.

Fixes #132 ([922041d](#))

- MR: get list of changes and commits ([92edb99](#))
- remove debug print statement ([4fd00f8](#))
- Add support for project environments ([5b08d2a](#))
- add support for global deploy key listing ([9bd2cb7](#))
- Merge branch 'master' of github.com:gpocentek/python-gitlab ([e3ac32f](#))
- add a contributing section in README ([e6ffd69](#))
- doc: replace incorrect archive call() ([e1f5e15](#))
- document namespaces API ([1f52cd2](#))
- Merge pull request #131 from chrwen-omicron/enable\_container\_registry

Added a new project attribute to enable the container registry. ([e0f2290](#))

- Added a new project attribute to enable the container registry. ([d61510e](#))

- Add support from listing group issues ([580f21e](#))
- Docs: drop the FAQ

The only question is now documented in the API examples. ([261f947](#))

- Improve commit statuses and comments

Fixes #92 ([f0fbefe](#))

- tests: don't use deprecated Content method ([741896d](#))
- CLI: refactor `_die()` ([99d0177](#))
- doc: fix doubled parameter ([58ddf1d](#))
- Replace `Snippet.Content()` with a new `content()` method

This new method use the standard lowercase name and implements data streaming. ([832ed9f](#))

- fix unit tests ([3198ead](#))
- Groups can be updated ([048b1cf](#))
- Add missing args in docstrings ([075345a](#))
- Allow to stream the downloads when appropriate

Some API calls will download possibly large data, resulting in a high memory usage and out-of-memory errors. For these API calls use the requests streaming capabilities and download chunked data. The caller is responsible of providing a callable to actually store the data.

The default callable just prints the data on stdout. ([94aea52](#))

- Implement `ProjectBuild.keep_artifacts` ([e0cf1c2](#))
- Gitlab: add managers for build-related resources ([b339ed9](#))
- Implement runners global API ([6eb11fd](#))
- Add docstring for settings manager in Gitlab class ([52c8825](#))
- Fix pep8 test ([63f0c4d](#))
- implement CLI for project archive/unarchive/share ([fc68e9b](#))
- Implement sharing project with a group ([e7c4125](#))
- Fix `ProjectMember` update ([2df4c9e](#))
- Update `ProjectSnippet` attributes

'visibility\_level' has been added as an optional attribute to keep compatibility with older releases of gitlab.

Fixes #129 ([3ad612d](#))

- Implement archive/unarchive for a projet

The methods are called `archive_` and `unarchive_` to workaround a conflict with the deprecated `archive` method. Method will be renamed when the `archive` method is removed. ([7ed34ed](#))

- Fix the `Project.archive` call ([565c35e](#))
- Project: add `VISIBILITY_*` constants

They should be there instead of having them in the `Group` class. Variables in `Group` are kepted for compatibility. ([2457823](#))

- Implement user emails support ([0be4761](#))



- document users API (9ba2d89)
- Add branches API documentation (cdd801e)
- Merge pull request #128 from rafaeleyng/feat/add-note-events-to-project-hook

add `note_events` to project hooks attributes (c88c638)

- add `note_events` to project hooks attributes (ca662e2)
- MR: add (un)subscribe support (867b7ab)
- Merge branch 'label-subscribe' (d340d31)
- Add support for label (un)subscribe (6f29ff1)
- add support for namespaces (79feb87)
- milestone: optional listing attrs (dbad3bd)
- update ProjectLabel attributes (c55fd4b)
- Add support for project-issue move (73627a2)
- project issue: proper update attributes (ca68f6d)
- issues: add missing optional listing parameters (60c9910)
- Merge branch 'master' of github.com:gpocentek/python-gitlab (2e0ac3f)
- issues: add optional listing parameters (c85276a)
- Merge pull request #125 from gpocentek/issue-122

Add support for build artifacts and trace (80a1908)

- Add support for commit comments

<https://github.com/gitlabhq/gitlabhq/blob/master/doc/api/commits.md> (412e2bc)

- commit status: optional get attrs (547f385)
- commit status: add optional context url (0b8ed5a)
- Make GroupProject more "python-gitlabish" (68d15fd)
- Merge branch 'master' of <https://github.com/missionrulz/python-gitlab> into missionrulz-master (69e64a3)
- typo in doc block (d9b9f92)
- add to `init.py` & move manager after class declaration (8f707ac)
- move into own class & create manager class (eb6c26f)
- Merge pull request #124 from Condla/master

Fix: `-title` is required argument, when in reality optional (18de4ef)

- Fix `-title` is not a required argument anymore (899490b)
- Fix `-title` is not a required argument anymore (c24f0d9)
- Fix that `-title` is a required argument, when trying to update a ProjectMilestone (bea8ea9)
- Add support for build artifacts and trace

Fixes #122 (b3e0974)

- Merge pull request #121 from PeterMosmans/httpauthextended

Added HTTPAuth support for even more methods :) (422b163)

- pylint fix ([40d7969](#))
- Added HTTPauth support for even more methods :) ([59ed4fc](#))
- add HTTP auth options to doc ([6220308](#))
- Merge pull request #120 from PeterMosmans/basicauth

Added support for HTTP basic authentication ([11f1e2d](#))

- Added support for HTTP basic authentication ([e9e48b9](#))
- project issue: doc and CLI for (un)subscribe ([1b14f5c](#))
- Merge pull request #118 from IvicaArsov/issues\_subscribe\_unsubscribe

Add support for subscribe and unsubscribe in issues ([7bbbfbf](#))

- Merge branch 'master' of github.com:gpocentek/python-gitlab ([b8f19ca](#))
- Add support for subscribe and unsubscribe in issues ([d42687d](#))
- version bump ([0535808](#))
- update changelog and authors ([57936af](#))
- Merge pull request #110 from chrwen-omicron/remove\_next\_url\_from\_cls\_kwargs

Remove 'next\_url' from kwargs before passing it to the cls constructor. ([05dd8dc](#))

- Manage optional parameters for list() and get()
- List these elements in the API doc
- Implement for License objects ([fd45397](#))
- fix pep8 tests ([417d27c](#))
- implement list/get licenses ([62e4fb9](#))
- Merge branch 'master' of github.com:gpocentek/python-gitlab ([ee1620b](#))
- implement star/unstar for projects ([1de6b7e](#))
- Deprecate Project.archive() ([24c283f](#))
- Merge pull request #113 from adamreid/master

Enable updates on ProjectIssueNotes ([f5c75cb](#))

- Rename some methods to better match the API URLs

Also deprecate the file\_\* methods in favor of the files manager. ([45adb6e](#))

- ProjectFile: file\_path is required for deletion ([8ce8218](#))
- Rework the Gitlab.delete method

Fixes #107 ([0a1bb94](#))

- Revert "enable python 3.5 tests for travis"

This reverts commit 1915519f449f9b751aa9cd6bc584472602b58f36. ([c3f5b3a](#))

- Rework merge requests update

Fixes #76 ([aff99b1](#))

- enable python 3.5 tests for travis ([1915519](#))
- Enable deprecation warnings for gitlab only

Fixes #96 ([d204e66](#))

- Add new optional attributes for projects

Fixes #116 ([f12c732](#))

- Merge branch ‘master’ of <https://github.com/gpocentek/python-gitlab> ([8edd7f7](#))
- Remove unnecessary canUpdate property from ProjectIssuesNote ([111b7d9](#))
- Drop the next\_url attribute when listing

Fixes #106 ([64af398](#))

- Implement project contributors ([1600770](#))
- Implement project compare

Fixes #112 ([250f348](#))

- Add support for Project raw\_blob ([7a8f81b](#))
- Merge pull request #108 from guyzmo/using\_requests\_session

Adding a Session instance for all HTTP requests ([23e8146](#))

- Enable updates on ProjectIssueNotes ([5fe7e27](#))
- update docblock ([cd13aff](#))
- list projects under group ([d4e2cd6](#))
- Remove ‘next\_url’ from kwargs before passing it to the cls constructor.

The ‘next\_url’ argument causes problems in the \_construct\_url method if it doesn’t belong there. E.g. if you list all projects, change an attribute of a project and then try to save it, the \_construct\_url will use the ‘next\_url’ from the list method and the save will fail. ([c261875](#))

- Adding a Session instance for all HTTP requests

The session instance will make it easier for setting up once headers, including the authentication payload, and it’s also making it easier to override HTTP queries handling, when using [betamax](#) for recording and replaying the test suites. ([858352b](#))

- Add missing group creation parameters

description and visibility\_level are optional parameters for group creation. ([61bc24f](#))

- Add deletion support for issues and MR

This is supported in gitlabhq master branch for admin users (soft deletion). ([9a9a4c4](#))

- add “external” parameter for users ([349f66e](#))
- MR: add support for closes\_issues ([571a382](#))
- MR: add support for cancel\_merge\_when\_build\_succeeds ([f6a51d6](#))
- minor docs fixes ([ccbea3f](#))
- Add support for MergeRequest validation

Both API and CLI support this feature.

fixes #105 ([43e8a2a](#))

- Update changelog and authors ([bb463ae](#))
- version bump ([754d5e5](#))

- Gitlab.update(): use the proper attributes if defined ([f8528cc](#))
- add a note about project search API ([e48e4ac](#))
- Merge branch 'master' of github.com:gpocentek/python-gitlab ([e46c188](#))
- pep8 ignore H803 errors (git messages) ([86ade4a](#))
- Merge pull request #98 from Asher256/fix-unicode-syntax-py3

Fix the 'invalid syntax' on Python 3.2, because of u'password' ([aea678b](#))

- Fix the 'invalid syntax' on Python 3.2, because of u'password'

More informations regarding this issue:

Operating system: Debian Wheezy, with Python 3.2 and the last version of python-gitlab.

The gitlab module raised this exception, because of the 'u' (Unicode):

Traceback (most recent call last): File "push\_settings.py", line 14, in <module> from helper import ROOT\_EMAIL, ADMINS, git, old\_git File "/opt/scripts/gitlab/helpers/helper.py", line 25, in <module> from gitlab import Gitlab File "/opt/scripts/gitlab/helpers/gitlab/init.py", line 32, in <module> from gitlab.objects import \* # noqa File "/opt/scripts/gitlab/helpers/gitlab/objects.py", line 546 selfdict.pop(u'password', None) ^ SyntaxError: invalid syntax It is a recent change: 01802c0 (Richard Hansen 2016-02-11 22:43:25 -0500 546) selfdict.pop(u'password', None) 01802c0 (Richard Hansen 2016-02-11 22:43:25 -0500 547) otherdict.pop(u'password', None)

To solve the issue, 'u' was removed. ([7ed84a7](#))

- Re-implement \_custom\_list in the Gitlab class

Rename the method \_raw\_list. This adds support for the all=True option to enable automatic recursion and avoid pagination if requested by the user.

Fixes #93 ([453224a](#))

- remove unused \_returnClass attribute ([44d0dc5](#))
- CI: implement user get-by-username

fixes #95 ([58433d2](#))

- CLI: fix discovery of method to execute ([7260684](#))
- Improve the doc for UserManager

Describe parameters, return values and exceptions for search() and get\_by\_username(). ([b79af1d](#))

- Implement "user search" CLI ([073d8d5](#))
- Merge branch 'rhansen-get-specific-user' ([6975ac6](#))
- define UserManager.get\_by\_username() to get a user by username ([ac2e534](#))
- define UserManager.search() to search for users ([8f59516](#))
- define GitlabObject.eq() and ne() equivalence methods ([01802c0](#))
- define GitlabObject.as\_dict() to dump object as a dict ([f15a7cf](#))
- Merge pull request #89 from ExodusIntelligence/master

Adding new ProjectHook attributes: ([81be3cf](#))

- Add a coverage tox env ([2e1f84e](#))
- Add some unit tests for CLI

Reorganize the cli.py code to ease the testing. ([d2e30da](#))

- Rework the CLI code

Add support for more subcommands. (8aa8d8c)

- Merge pull request #90 from ms-boom/fix\_custom\_list

fix GitlabObject creation in \_custom\_list (f5ca0eb)

- fix GitlabObject creation in \_custom\_list (293a9dc)
- Add support for user block/unblock (e387de5)
- Added missing comma (1f81c2d)
- Adding new ProjectHook attributes:
- build\_events
- enable\_ssl\_verification

See the two links below:

- <https://github.com/gitlabhq/gitlabhq/blob/master/doc/api/projects.md#add-project-hook>
- <https://github.com/pyapi-gitlab/pyapi-gitlab/pull/173> (db9bbf6)
- bump version to cleanup my mess on pypi (942468d)
- Update ChangeLog and AUTHORS (74d82d4)
- MANIFEST: add .j2 files in docs/ (a495eec)
- CLI: fix {all,owned,search} project listing (522cfd1)
- Partially revert 00ab7d00 (9ae26fe)
- version bump (8642e9e)
- Merge branch 'test-script-cleanups' of <https://github.com/rhansen/python-gitlab> into rhansen-test-script-cleanups (01e7c06)
- don't suppress docker's standard error

While docker is quite noisy, suppressing stderr makes it difficult to troubleshoot problems. (0024552)

- wait for the docker container to stop before removing it (52e4377)
- use 'docker stop' instead of 'docker kill'

The 'stop' command first tries SIGTERM before resorting to SIGKILL, which is a gentler way to stop processes. (SIGTERM gives processes an opportunity to clean up before exiting; SIGKILL can't be caught so it is very abrupt.) (c56fc47)

- add more log messages (2609cbb)
- define a testcase() function; use it for tests (37f6d42)
- use \${CONFIG} instead of repeating the filename (033881e)
- fix usage error message (bc7332f)
- Add docstrings to some methods (1b5c8f1)
- Fix the RTD requirements (2e5476e)
- improve error handling

Break up pipelines and check the exit status of non-basic commands to ensure that any problems cause the scripts/testcases to fail right away. (5dcceb8)

- convert scripts to POSIX shell by eliminating bashisms ([57f1ad5](#))
- quote underquoted variable expansions

This protects against word splitting if the variable contains IFS characters, and it ensures that an empty variable doesn't become an elided argument. ([09ef253](#))

- convert \$GITLAB to a function

This makes it possible to quote the \$CONFIG variable expansion. ([c100a04](#))

- convert \$OK to a function

This makes it possible to quote the variable expansions. ([8198e3f](#))

- only run deactivate if it exists

The deactivate command only exists if activate is run, but cleanup() might be called before activate is run if there is an error. ([6b298c6](#))

- ensure that cleanup() runs if terminated by the user ([17914a3](#))
- check if docker container is up when waiting for gitlab

There's no point in waiting for GitLab to come up if the docker container died. ([58106a0](#))

- error out if required utilities aren't installed ([b21fdda](#))
- use the log functions for errors and status messages

This causes the error messages to go to standard error, and it makes it easy to prefix all log messages if desired. ([867fe2f](#))

- add logging and error handling helper functions ([7c0e443](#))
- compact some case statements ([dfc6c70](#))
- move common code to build\_test\_env.sh

Note that build\_test\_env.sh now creates and prepares the Python virtualenv (it didn't before). ([26999bf](#))

- Automatic doc generation for BaseManager classes

Provide a sphinx extension that parses the required/optional attributes and add info to the class docstring. ([a2eca72](#))

- wrap long lines

Use line continuations to keep lines shorter than 80 columns. ([6df844a](#))

- travis lacks py35 support without tricks ([770dd4b](#))
- add python 3.5 test env ([920d248](#))
- add support for project builds ([ebf36b8](#))
- Fix Project.tree()

Add API tests for tree(), blob() and archive(). ([fc8affd](#))

- Fix project update ([141f21a](#))
- Rework the **version** import

This simplifies the setup.py script

Also provide a --version option for CLI ([00ab7d0](#))

- fix inclusion of api/\*.rst ([9f256a7](#))
- Add sudo support ([3711f19](#))

- Fix the ‘password’ requirement for User creation ([c579c80](#))
- Add support for application settings ([16d50cd](#))
- Implement project variables support ([e5438c6](#))
- implement project triggers support ([c11bebd](#))
- Implement setting release info on a tag

Add the `set_release_description()` method to `ProjectTag`. Add python API test for this method. ([7981987](#))

- API tests for tags ([0814d86](#))
- `ProjectTag` supports deletion (gitlab 8.4.0) ([339d329](#))
- Implement `ProjectMilestone.issues()`

This lists the issues related to the milestone.

Add python API tests for issues. ([db1fb89](#))

- fix `ProjectLabel` get and delete ([1ecb739](#))
- wait a little before running the python tests ([9709d79](#))
- fix the API test for `decode()` ([c22a19e](#))
- increase the timeout value for tests ([4e21343](#))
- make connection exceptions more explicit ([08d3ccf](#))
- add a `decode` method for `ProjectFile` ([2eac071](#))
- README is an RST file... ([d3a5701](#))
- Update README with travis status ([a4918a3](#))
- fix the `test_create_unknown_path` test ([bf985b3](#))
- add a travis configuration file ([e40d9ac](#))
- Fix the `json()` method for python 3

Also add unit tests and fix pep8 test ([d7271b1](#))

- Merge branch ‘rhansen-fix-json’ ([982f54f](#))
- Merge branch ‘fix-json’ of <https://github.com/rhansen/python-gitlab> into rhansen-fix-json ([26d73e2](#))
- use a custom docker image for tests ([3f38689](#))
- skip `BaseManager` attributes when encoding to JSON

This fixes the following exception when calling `User.json()`:

```
TypeError: <gitlab.objects.UserKeyManager object at 0x7f0477391ed0> is not JSON
↪serializable ([`ca6da62`](https://github.com/python-gitlab/python-gitlab/commit/
↪ca6da62010ee88e1b03f7a5abbf69479103aa1e1))
```

- add a missing import statement

Add the import inside the function rather than at the top of the file because otherwise it would introduce a circular dependency. ([c95b3c3](#))

- Add an initial set of API tests ([7e4e1a3](#))
- include the docs in the tarball ([bbcccaa](#))
- 0.11.1 release update ([5c4f77f](#))

- add some CLI tests ([097171d](#))
- add unit tests for managers ([6baea2f](#))
- remove debugging print instruction ([7e54a39](#))
- Fix discovery of parents object attrs for managers ([0e0c81d](#))
- remove “=” in examples for consistency ([a4e29f8](#))
- (re)add CLI examples in the doc ([1b64a47](#))
- Merge pull request #82 from cdbennett/commitstatus

Support setting commit status ([02c5398](#))

- Support setting commit status

Support commit status updates. Commit status can be set by a POST to the appropriate commit URL. The status can be updated by a subsequent POST to the same URL with the same name and ref, but different values for state, description, etc.

Note: Listing the commit statuses is not yet supported. This is done through a different path on the server, under the repository path.

Example of use from the CLI:

```
# add a build status to a commit
gitlab project-commit-status create --project-id 2 \
  --commit-id a43290c --state success --name ci/jenkins \
  --target-url http://server/build/123 \
  --description &#34;Jenkins build succeeded&#34;; ([`3371008`])(https://github.com/
python-gitlab/python-gitlab/commit/33710088913c96db8eb22289e693682b41054e39))
```

- Support deletion without getting the object first

Use this feature in the CLI to avoid an extra API call to the server. ([1d7ebea](#))

- cli.py: make internal functions private ([e5821e6](#))
- Add a script to build a test env

functional\_tests.sh has been split in 2 scripts to make easier the run of gitlab container. ([572cfa9](#))

- Rework gitlab.\_sanitize

Make it a recursive function and eliminate \_sanitize\_dict.

Add unit tests. ([03d8041](#))

- Improve the API documentation. ([4781fd7](#))
- Rewrite the README

And link to the docs on RTD. ([3e8cf4e](#))

- Bump version

And update copyright years. ([ca44878](#))

- Update AUTHORS ([0163499](#))
- Update ChangeLog for 0.11 ([3fd64df](#))
- add missing import ([9c74442](#))
- document the API migration from 0.10 ([610bde8](#))



- implement group search in CLI ([5ea6d0a](#))
- Add support for groups search

Factorize the code to avoid duplication with the ProjectManager class. Implement unit tests for the group search.

Original patchh from Daniel Serodio (PR #55). ([5513d0f](#))

- unit tests for config parser ([0ae315a](#))
- Implement ProjectManager search/list methods

The existing Gitlab methods are deprecated.

Unit tests have been added. ([689ecae](#))

- CLI: fix the discovery of possible actions ([e48e149](#))
- Create a manager for ProjectFork objects ([e8631c1](#))
- Merge branch 'fgouteroux-add\_fork\_support' ([37912c1](#))
- Merge branch 'add\_fork\_support' of <https://github.com/fgouteroux/python-gitlab> into fgouteroux-add\_fork\_support ([77d34b3](#))
- Deprecate the “old” Gitlab methods

Update the associated unit tests. ([2bf9794](#))

- add fork project support ([cedf080](#))
- README update ([dc0099d](#))
- Provide a getting started doc for the API ([2237d85](#))
- Remove extra dep on sphinx-argparse ([64d6356](#))
- Rework the requirements for RTD ([4a53627](#))
- Document installation using pip and git ([7523a61](#))
- Document the CLI ([0ee53e0](#))
- add unit tests for BaseManager ([2a93c62](#))
- GitLab -> Gitlab (class names) ([fdf295f](#))
- fix pretty\_print with managers ([bef97fe](#))
- README update ([d0da618](#))
- Implement managers to get access to resources

This changes the ‘default’ API, using managers is the recommended way to get/list/create objects. Additional operations will be implemented in followup patches.

Old methods are deprecated and will disappear in a while. ([46f74e8](#))

- update the docs copyright years ([e5246bf](#))
- add missing copyright header ([b66672e](#))
- Split code in multiple files ([8fa4455](#))
- remove deprecated methods ([118b298](#))
- python3: fix CLI error when arguments are missing ([7c38ef6](#))
- fix the tests ([1db3cc1](#))
- Rename the \_created attribute \_from\_api ([2a76b74](#))

- Provide a create method for GitlabObject's

Instead of using the constructor to do everything (get, list and create), we now provide a class method for each action. This should make code easier to read. ([a636d5a](#))

- Add the CLI -g short option for --gitlab ([7e61a28](#))
- Add a get method for GitlabObject

This change provides a way to implement GET for objects that don't support it, but support LIST.

It is also a first step to a cleaner API. ([74dc2ac](#))

- functional\_tests.sh: support python 2 and 3 ([c580b1e](#))
- Move request return\_code tests in \_raise\_error\_from\_response ([81b0a06](#))
- Version bump ([38f17c1](#))
- update AUTHORS and ChangeLog ([e673dab](#))
- Add support for group members update

Closes #73 ([99c4710](#))

- Merge branch 'master' of github.com:gpocentek/python-gitlab ([45becb9](#))
- Sanitize the id used to construct URLs

Closes #28 ([5d88f68](#))

- Merge pull request #78 from cdbennett/fix\_python3\_sort\_types

Use name as sort key to fix Python 3 TypeError ([6f1fd7e](#))

- Use name as sort key to fix Python 3 TypeError

Sort types explicitly by name to fix unorderable types TypeError in Python 3.

The call to sort() on cli.py line 259 produced the error:

```
TypeError: unorderable types: type() &lt; type() ([`363b75e`](https://github.com/python-  
gitlab/python-gitlab/commit/363b75e73c2b66ab625811accdb9d639fb068675))
```

- try to fix the RTD build ([acc1511](#))
- Merge pull request #72 from pa4373/newuser-confirm-fix

Can bypassing confirm when creating new user now ([d069381](#))

- Can bypassing confirm when creating new user ([a0fe68b](#))
- Test branch creation et deletion ([f07de94](#))
- Fix GET/POST for project files ([9c58013](#))
- hide the action attribute ([3270865](#))
- Fix deletion of object not using 'id' as ID

Closes #68 ([e5aa69b](#))

- README: add missing import in sample ([d8fdb4c](#))
- setup.py: require requests>=1

Closes #69 ([21fdf1b](#))

- Provide a Gitlab.from\_config method

It provides the Gitlab object creation from the `~/.python-gitlab.cfg`, just like the CLI does. ([fef8c7f](#))

- update README for `list(all=True)` ([6cc8126](#))
- don't list everything by default ([a9e8da9](#))
- fix pep8 test ([e93188e](#))
- Merge pull request #64 from [jantman/issues/63](#)

python-gitlab Issue #63 - implement pagination for `list()` ([24d5035](#))

- Merge pull request #66 from [stefanklug/master](#)

Fix error when fetching single MergeRequests ([adbe0a4](#))

- add support to update MergeRequestNotes ([79d452d](#))
- fix url when fetching a single MergeRequest ([227f71c](#))
- issue #63 add unit tests for 'next' link handling in `list()` ([719526d](#))
- issue #63 - revert logging additions ([f9654cd](#))
- python-gitlab Issue #63 - implement pagination for `list()` ([33ceed6](#))
- Fix the update/delete CLI subcommands

Also update the testing tool to test these features.

Closes #62 ([802c144](#))

- fix delete and update CLI methods ([e57b779](#))
- more README updates ([0922ed5](#))
- Improve the README a bit

Fix typos Detail which options are required in the [global] section (closes #61) ([f6abd4d](#))

- 0.9.1 release ([1f48e65](#))
- setup.py: restore the version discovery hack ([9966616](#))
- functional\_test.sh: use a venv ([e12abf1](#))
- fix setuptools sdist ([73c68db](#))
- add tools/ to MANIFEST.in ([82ff055](#))
- add test files to MANIFEST.in ([32daf2a](#))
- add test-requirements.txt in MANIFEST.in ([5c20201](#))
- remove executable flag on cli.py ([b8ee8f8](#))
- get ready for a 0.9 release ([dce3193](#))
- Provide a basic functional test script

This can be used to quickly test the correct behavior of the CLI. The script is simple and doesn't test much for now, but it's a start. ([1bc412e](#))

- update copyright date ([aae8e2d](#))
- CLI: remove wrong attributes from the verbose output

These attributes come from the command line arguments. ([d254e64](#))

- CLI: provide a `--config-file` option ([711c5be](#))

- Rename a few more private methods ([bed3adf](#))
- Move the CLI in the gitlab package

Setup an console\_script entry point to create the executable script. ([d7bea07](#))

- Fix the tests when the host runs a web server ([4c6c778](#))
- Projects can be updated

Also fix the projects special listing (all, owned, ...)

Closes #54 ([7bdb1be](#))

- improve handling of id attributes in CLI

closes #31 ([8b42559](#))

- rework (and fix) the CLI parsing ([ede1224](#))
- use more pythonic names for some methods ([c2d1f8e](#))
- make the tests pass ([0032d46](#))
- setup tox for py27 and py34 tests ([8634a4d](#))
- move the tests inside the package ([03bbfa0](#))
- Merge branch 'tests' of <https://github.com/mjmaenpaa/python-gitlab> into mjmaenpaa-tests

Conflicts: setup.py ([0443256](#))

- Deprecate some Gitlab object methods

raw\* methods should never have been exposed; replace them with *raw\** methods

setCredentials and setToken are replaced with set\_credentials and set\_token ([b7d04b3](#))

- sphinx: don't hardcode the version in conf.py ([59173ce](#))
- gitlab is now a package ([d2e5700](#))
- add a tox target to build docs ([105896f](#))
- Add a tox configuration file

Run pep8 tests only for now, and fix pep8 errors. ([82a88a7](#))

- Merge pull request #58 from [massimone88/master](#)

Used argparse library ([99cc43a](#))

- change changelog, change, add my name on collaborators, change version ([0d5b988](#))
- implemented argparse object for parsing the argument of the command line ([d44b48d](#))
- Merge branch 'feature/impl\_argparse' into develop ([fd473cd](#))
- \*clean import package +add folder .idea to gitignore ([23fe3c9](#))
- bug fixed on requiredArguments ([d099b11](#))
- remove "gitlab" of arguments because conflicts with "gitlab" attribute with GitlabObject class ([bdc6f73](#))
- remove forgotten argument ([e6c85b5](#))
- improvement argument required for each action add try/catch for error of parsing of not gitlabObject ([2792091](#))
- implement argparse library for parsing the arguments create constans for action name clean the code ([9439ce4](#))
- "timeout" option is an int, not a bool ([2d48e71](#))

- require sphinxcontrib-napoleon to build the docs ([990eeca](#))
- Make sphinx-configuration work with python2 ([f22bfbd](#))
- Updated few gitlab.py docstrings as an example about how to document api ([3005b3d](#))
- Simple sphinx-project that automatically creates api documentation. ([e822b0b](#))
- ignore egg-info dirs ([d0884b6](#))
- add a requirements.txt file ([13cd78a](#))
- Little documentation about sudo-usage ([137ec47](#))
- Forgot to add sudo-support to update ([7d31e48](#))
- Support labels in issues correctly ([5d25344](#))
- Send proper json with correct content-type and support sudo-argument

Use json-encoder to create proper gitlab-compatible json Send only attributes specified with requiredCreateAttrs and optionalCreateAttrs Send correct content-type header with json Sudo, page & per\_page is supported for all methods by using \*\*kwargs to pass them Changed rawPut to have same parameters as rawPost ([96a44ef](#))

- Update example about how to close issue in README ([78c4b72](#))
- Added missing optionalCreateAttrs for ProjectIssue. Fixed typo in ProjectTag. ([72eb744](#))
- Improved error reporting
- Try to parse error response from gitlab.
- Use one function (\_raiseErrorFromResponse) to parse and raise exceptions related to errors reported by gitlab. ([8e13487](#))
- Improved exception-classes.
- Added http status-code and gitlab error message to GitlabError-class.
- Added new GitlabOperationError-class to help separate connection and authentication errors from other gitlab errors. ([8351b2d](#))
- Raise NotImplementedError on all cases, where can\*-boolean is False ([555cc45](#))
- No reason to have separate \_getListOrObject-method in Gitlab-class.

Changed GitlabObject-class version of \_getListOrObject to classmethod. Removed \_getListOrObject-method from Gitlab-class. Changed Gitlab-class to use GitlabObject-class version of \_getListOrObject ([90ebbeb](#))

- bump version to 0.8 ([296a72f](#))
- “Document” the timeout option ([39aa998](#))
- Update the Changelog ([8be5365](#))
- CLI: support a timeout option ([1672529](#))
- make sure to not display both id and idAttr ([9744475](#))
- ProjectLabel: use name as id attribute ([f042d2f](#))
- pretty\_print: don't display private attributes ([fa92155](#))
- Add Mika Mäenpää in the authors list ([526f1be](#))
- Merge pull request #45 from mjmaenpaa/labels\_files

Classes for ProjectLabels and ProjectFiles ([928b9f0](#))

- Merge pull request #44 from mjmaenpaa/noid\_objects

Support api-objects which don't have id in api response. ([afe0ab4](#))

- Merge pull request #43 from mjmaenpaa/url\_delete\_attrs

Moved url attributes to separate list. Added list for delete attributes. ([f7dfad3](#))

- Merge pull request #40 from mjmaenpaa/py3

Python3 compatibility ([1eccc3b](#))

- Fixed object creation in list ([134fc7a](#))
- Classes for ProjectLabels and ProjectFiles ([ad63e17](#))
- Support api-objects which don't have id in api response. ([c3ab869](#))
- Moved url attributes to separate list. Added list for delete attributes. ([ea4c099](#))
- Merge pull request #42 from mjmaenpaa/constructUrl

Moved url-construction to separate function ([221f418](#))

- Merge pull request #41 from mjmaenpaa/gitlab\_get\_exception

Gitlab.get() raised GitlabListError instead of GitlabGetError ([9736e0b](#))

- Moved url-construction to separate function ([e14e3bf](#))
- Gitlab.get() raised GitlabListError instead of GitlabGetError ([ee54b3e](#))
- Py3 compatibility with six ([431e4bd](#))
- Merge pull request #39 from mjmaenpaa/timeout

Timeout support ([9f134fc](#))

- Python 3 compatibility for cli-program ([d714c4d](#))
- Timeout support ([d2e591e](#))
- Python3 compatibility ([15c0da5](#))
- Merge pull request #38 from mjmaenpaa/currentuser\_key

Changed CurrentUser.Key to use \_getListOrObject-method like all other functions ([4664ebd](#))

- Merge pull request #37 from mjmaenpaa/list\_kwargs

No reason to add kwargs to object in Gitlab.list()-method ([2c86085](#))

- Merge pull request #36 from mjmaenpaa/setFromDict

\_setFromDict thinks False is None ([4c5c39d](#))

- CurrentUser.Key uses \_getListOrObject-method ([afc1c2](#))
- No reason to add kwargs to object in Gitlab.list()-method because GitlabObject constructor can handle them. ([40ce81e](#))
- \_setFromDict thinks False is None ([3cf35ce](#))
- Added tests.

Uses httmock library to abstract away requests-library. Uses nose to actually run tests. ([f458522](#))

- Merge pull request #34 from tekacs/master

Update .sort to use key for Python 3.x. ([ff2d84c](#))

- Update .sort to use key for Python 3.x.

Rather than really dubious cmp function. ([b483319](#))

- Merge pull request #32 from patgmiller/master

refactor “\_sanitize” for Python < 2.7 ([6c4fc34](#))

- refactor “\_sanitize” for Python < 2.7 ([89d3fa0](#))
- changelog update ([8846bf7](#))
- bump version ([3f0ac43](#))
- update copyright years ([c6f0a8d](#))
- flake8 fixes ([5d5e0f7](#))
- Fix handling of boolean values

Gitlab expects an int (1 or 0) as value for boolean attributes. Transform python bool’s into int’s when creating or updating objects.

Closes #22 ([d4803f9](#))

- Support namespace/name for project id

Closes #28 ([34d6952](#))

- update AUTHORS ([d38f219](#))
- Merge pull request #27 from cdleonard/master

Fix encoding errors on display and update with redirected output ([2b5ea46](#))

- Support state\_event in ProjectMilestone

Closes #30 ([2281283](#))

- add support for branches creation and deletion ([97e2689](#))
- add support for UserKey listing and deletion ([09e4a64](#))
- drop the module shebang ([01335f3](#))
- Fix encoding error when updating with redirected output

When output is redirected sys.stdout.encoding is None. Fix this by always encoding to utf-8, assuming gitlab handles that. The encoding used by the local system is irrelevant here. ([ec185cf](#))

- Fix encoding error when printing to redirected output

When redirecting output to a file sys.stdout.encoding is None, so use sys.getdefaultencoding() instead. ([e236fd9](#))

- Merge pull request #16 from locke105/master

Fix license classifier in setup.py ([8ce3e30](#))

- Fix license classifier in setup.py ([994e464](#))
- version bump

Update Changelog and AUTHORS ([1fe783d](#))

- projects listing: explicitly define arguments for pagination ([4fcef67](#))
- Merge pull request #13 from dpasqualin/master

Add support for extra parameters when listing all projects (Refs #12) ([4b882b7](#))

- ProjectMember: constructor should not create a User object ([1c21423](#))
- ids can be unicode

Fixes #15 ([c6e371e](#))

- Add support for extra parameters when listing all projects (Refs #12)

Signed-off-by: Diego Giovane Pasqualin <[dpasqualin@c3sl.ufpr.br](mailto:dpasqualin@c3sl.ufpr.br)> ([1b6c595](#))

- version bump ([04574f3](#))
- support creation of projects for users ([dc2bf5e](#))
- Merge branch 'ProjectFile' ([0ee6ca5](#))
- Project: add methods for create/update/delete files ([ba39e88](#))
- support projects listing: search, all, owned ([bd6b4ac](#))
- system hooks can't be updated ([2b4924e](#))
- Project.archive(): download tarball of the project ([debe41a](#))
- define new optional attributes for user creation ([1cc7b17](#))
- provide constants for access permissions in groups ([7afd232](#))
- update AUTHORS and Changelog ([962e806](#))
- Merge remote-tracking branch 'github-mrts/master' ([7fb7c47](#))
- add a Key() method for User objects ([1969abb](#))
- Merge pull request #10 from ksmets/master

Add SSH key for user ([e31bb9e](#))

- Add SSH key for user ([909c10e](#))
- Add support for project events. ([32d4224](#))
- Fix comments. ([6705928](#))
- Merge pull request #8 from Itxaka/master

fixed the requirements auto install from setup.py ([37e6648](#))

- fixed the requirements auto install from setup.py ([01ade72](#))
- version bump ([5f0136c](#))
- Add support for Gitlab 6.1 group members ([71c8750](#))
- minor syntax/pep8 updates ([09ef68f](#))
- drop leftovers from local tests ([4f001b4](#))
- Implement Gitlab 6.1 new methods
- Project: tree, blob
- ProjectCommit: diff, blob ([c9aedf2](#))
- ProjectMergeRequest: fix Note() method ([4006ab2](#))
- Allow to get a project commit (GitLab 6.1) ([6b0c678](#))
- Fix strings encoding (Closes #6) ([64cead6](#))
- version bump ([ceda87a](#))
- doc updates ([b73c92d](#))



- Merge branch 'header-private-token' of <https://github.com/dekimsey/python-gitlab> into token\_in\_header ([9adb4fa](#))
- provide a ChangeLog ([2d5342b](#))
- provide a AUTHORS file ([7c85fb7](#))
- cli: support ssl\_verify config option ([147a569](#))
- Merge pull request #5 from marbindrakon/ssl\_verify\_option

Add ssl\_verify option to Gitlab object. ([bb7ba7a](#))

- Add ssl\_verify option to Gitlab object. Defaults to True ([309f1fe](#))
- Merge pull request #4 from erikjwaxx/master

Correct url for merge requests API. ([7431d91](#))

- Correct url for merge requests API. ([5090ef4](#))
- version bump ([d1cd3dc](#))
- provide a pip requirements.txt ([c6fd8e8](#))
- drop some debug statements ([2807006](#))
- include COPYING in distribution ([d65b684](#))
- Merge pull request #1 from dekimsey/team-api

Added API for team access. ([8f65cf8](#))

- Merge remote-tracking branch 'samcdays/teams' into team-api

Conflicts: gitlab.py ([cb5b754](#))

- Added API for team access. ([8a22958](#))
- Use PRIVATE-TOKEN header for passing the auth token ([d39c471](#))
- improve pretty\_print() ([05ab473](#))
- manage project branch protection with the cmd line ([53562b3](#))
- rework the script code organization ([33c771d](#))
- rework the cmd line options ([02bd7cd](#))
- make -verbose behave like -fancy ([a9b5bf4](#))
- fix parsing of options ([2321631](#))
- gitlab: make the current-user option work ([4c998ea](#))
- README: document -page and -per-page ([a7f2065](#))
- listing: list the -page and -per-page options ([079c107](#))
- ProjectBranch: commit is an other object ([7e7b29c](#))
- README: use - instead of \_ in examples ([b4bc9df](#))
- drop the debian/ dir from master ([7be3d54](#))
- drop the tests dir, this is useless ([7c358d3](#))
- gitlab: be less verbose by default

Provide a -fancy option to output more data on list/get/create queries. ([3b15c6d](#))

- pretty\_print: use - instead of \_ ([41b6dba](#))
- id attr might not be available ([7175772](#))
- allow to use dash (-) instead of underscore (\_) in attribute names ([5a20efb](#))
- Merge branch 'master' into debian ([93d5147](#))
- provide a manifest for distribution ([1c1702d](#))
- update version in changelog ([d9d6e0c](#))
- Merge branch 'master' into debian ([23753df](#))
- gitlab: autogenerate some doc ([39a4a20](#))
- gitlab: update the object syntax ([9ca47aa](#))
- provide debian packaging ([ef44b84](#))
- python 3 support ([e7ba350](#))
- object creation: print the created object ([c6174e5](#))
- return explicit error message on 404 ([a04a5a5](#))
- gitlab: warn the user if an action cannot be performed ([34e4304](#))
- minor syntax change: canGetList => canList ([49eab91](#))
- setup a list of mandatory attributes for list and get methods ([5dda6e6](#))
- Manually parse the arguments

We can use a more common syntax (– prefix for options) this way. ([a8072d9](#))

- install the gitlab script ([dd22ce1](#))
- describe the gitlab script in README ([204f681](#))
- provide a basic CLI ([a205914](#))
- provide a ProjectSnippet.Content() method ([72e097d](#))
- add a GitlabObject.pretty\_print method ([abf1b0d](#))
- deal with ids as strings ([bc9d440](#))
- raise an exception if deletion fails ([4ee9c8c](#))
- Allow creation of objects using the “hidden API” ([1d55e67](#))
- Basic team support. ([5388d19](#))
- Check the needed attributes to create objects

Provide a required and optional arguments lists for each object that can be created using the API ([123a01e](#))

- add a json() method to GitlabObject's ([8d65870](#))
- implement project transfer support ([1625e55](#))
- add support for project deploy keys ([1e3061c](#))
- support for system hooks ([42bef0a](#))
- fix use of json() method from requests ([af84700](#))
- unittest for SSH keys ([7631e5e](#))
- some unit tests! ([967ea88](#))

- restore Gitlab.Issue() prototype ([561f349](#))
- move documentation and todo to README.md ([c10b01e](#))
- implement protect/unprotect for branches ([a02180d](#))
- typo ([dd210be](#))
- never add page and per\_page attributes to the objects ([96bf2b1](#))
- ids for single items might be str ([928d4fa](#))
- add a pagination example ([74ec951](#))
- Allow to pass additional args to list constructors

This is needed mostly for pagination support. ([0149020](#))

- spacing ([571ab0e](#))
- fix the token authentication ([938b1e6](#))
- add a \_ from private attributes ([523d764](#))
- docstrings for the Gitlab class ([046baf2](#))
- cosmetics: make pep8 happy ([9b64650](#))
- add missing raise keywork ([4d11843](#))
- add a setup.py script ([0dd1ede](#))
- fix LGPL header and provide module informations ([96e341e](#))
- rework authentication API ([eb4cd36](#))
- Rework the API

objects can be created using the usual syntax, with one optional argument (int to get one object, dict to create an object, nothing to get a list of items). A save() method creates/updates the object on the server. A delete() method removes it from the server. ([653843d](#))

- link GitLab and User classes to their possible children ([d1f80da](#))
- raise an exception on 401 return code ([01152da](#))
- drop Session() and add a Gitlab.authenticate() method ([c4920ee](#))
- create Note classes linked to parent objects ([bf25928](#))
- add support for notes ([0172900](#))
- implement the Session() method ([f188dcb](#))
- add methods to Project objects to access related objects ([ab82226](#))
- store a mirror of the gitlab instance in created objects ([02afd17](#))
- Rework object creation from json ([1fddcd2](#))
- initial import ([1d7e85d](#))

## 7.10 Release notes

Prior to version 2.0.0 and GitHub Releases, a summary of changes was maintained in release notes. They are available below for historical purposes. For the list of current releases, including breaking changes, please see the changelog.

### 7.10.1 Changes from 1.8 to 1.9

- `ProjectMemberManager.all()` and `GroupMemberManager.all()` now return a list of `ProjectMember` and `GroupMember` objects respectively, instead of a list of dicts.

### 7.10.2 Changes from 1.7 to 1.8

- You can now use the `query_parameters` argument in method calls to define arguments to send to the GitLab server. This allows to avoid conflicts between python-gitlab and GitLab server variables, and allows to use the python reserved keywords as GitLab arguments.

The following examples make the same GitLab request with the 2 syntaxes:

```
projects = gl.projects.list(owned=True, starred=True)
projects = gl.projects.list(query_parameters={'owned': True, 'starred': True})
```

The following example only works with the new parameter:

```
activities = gl.user_activities.list(
    query_parameters={'from': '2019-01-01'},
    all=True)
```

- Additionally the `all` parameter is not sent to the GitLab anymore.

### 7.10.3 Changes from 1.5 to 1.6

- When python-gitlab detects HTTP redirections from http to https it will raise a `RedirectionError` instead of a cryptic error.

Make sure to use an `https://` protocol in your GitLab URL parameter if the server requires it.

### 7.10.4 Changes from 1.4 to 1.5

- APIv3 support has been removed. Use the 1.4 release/branch if you need v3 support.
- GitLab EE features are now supported: Geo nodes, issue links, LDAP groups, project/group boards, project mirror pulling, project push rules, EE license configuration, epics.
- The `GetFromListMixin` class has been removed. The `get()` method is not available anymore for the following managers:
  - `UserKeyManager`
  - `DeployKeyManager`
  - `GroupAccessRequestManager`
  - `GroupIssueManager`
  - `GroupProjectManager`

- GroupSubgroupManager
  - IssueManager
  - ProjectCommitStatusManager
  - ProjectEnvironmentManager
  - ProjectLabelManager
  - ProjectPipelineJobManager
  - ProjectAccessRequestManager
  - TodoManager
- ProjectPipelineJob do not heritate from ProjectJob anymore and thus can only be listed.

### 7.10.5 Changes from 1.3 to 1.4

- 1.4 is the last release supporting the v3 API, and the related code will be removed in the 1.5 version.

If you are using a Gitlab server version that does not support the v4 API you can:

- upgrade the server (recommended)
- make sure to use version 1.4 of python-gitlab (`pip install python-gitlab==1.4`)

See also the [Switching to GitLab API v4 documentation](#).

- python-gitlab now handles the server rate limiting feature. It will pause for the required time when reaching the limit ([documentation](#))
- The `GetFromListMixin.get()` method is deprecated and will be removed in the next python-gitlab version. The goal of this mixin/method is to provide a way to get an object by looping through a list for GitLab objects that don't support the GET method. The method [is broken](#) and conflicts with the GET method now supported by some GitLab objects.

You can implement your own method with something like:

```
def get_from_list(self, id):
    for obj in self.list(as_list=False):
        if obj.get_id() == id:
            return obj
```

- The `GroupMemberManager`, `NamespaceManager` and `ProjectBoardManager` managers now use the GET API from GitLab instead of the `GetFromListMixin.get()` method.

### 7.10.6 Changes from 1.2 to 1.3

- `gitlab.Gitlab` objects can be used as context managers in a `with` block.

### 7.10.7 Changes from 1.1 to 1.2

- python-gitlab now respects the `*_proxy`, `REQUESTS_CA_BUNDLE` and `CURL_CA_BUNDLE` environment variables (#352)
- The following deprecated methods and objects have been removed:
  - `gitlab.v3.object Key` and `KeyManager` objects: use `DeployKey` and `DeployKeyManager` instead
  - `gitlab.v3.objects.Project.archive_` and `unarchive_` methods
  - `gitlab.Gitlab.credentials_auth`, `token_auth`, `set_url`, `set_token` and `set_credentials` methods. Once a Gitlab object has been created its URL and authentication information cannot be updated: create a new Gitlab object if you need to use new information
- The `todo()` method raises a `GitlabTodoError` exception on error

### 7.10.8 Changes from 1.0.2 to 1.1

- The `ProjectUser` class doesn't inherit from `User` anymore, and the `GroupProject` class doesn't inherit from `Project` anymore. The Gitlab API doesn't provide the same set of features for these objects, so python-gitlab objects shouldn't try to workaround that.

You can create `User` or `Project` objects from `ProjectUser` and `GroupProject` objects using the `id` attribute:

```
for gr_project in group.projects.list():
    # lazy object creation avoids a Gitlab API request
    project = gl.projects.get(gr_project.id, lazy=True)
    project.default_branch = 'develop'
    project.save()
```

### 7.10.9 Changes from 0.21 to 1.0.0

1.0.0 brings a stable python-gitlab API for the v4 Gitlab API. v3 is still used by default.

v4 is mostly compatible with the v3, but some important changes have been introduced. Make sure to read [Switching to GitLab API v4](#).

The development focus will be v4 from now on. v3 has been deprecated by GitLab and will disappear from python-gitlab at some point.

### 7.10.10 Changes from 0.20 to 0.21

- Initial support for the v4 API (experimental)

The support for v4 is stable enough to be tested, but some features might be broken. Please report issues to <https://github.com/python-gitlab/python-gitlab/issues/>

Be aware that the python-gitlab API for v4 objects might change in the next releases.

**Warning:** Consider defining explicitly which API version you want to use in the configuration files or in your `gitlab.Gitlab` instances. The default will change from v3 to v4 soon.

- Several methods have been deprecated in the `gitlab.Gitlab` class:

- `credentials_auth()` is deprecated and will be removed. Call `auth()`.
- `token_auth()` is deprecated and will be removed. Call `auth()`.
- `set_url()` is deprecated, create a new Gitlab instance if you need an updated URL.
- `set_token()` is deprecated, use the `private_token` argument of the Gitlab constructor.
- `set_credentials()` is deprecated, use the `email` and `password` arguments of the Gitlab constructor.
- The service listing method (`ProjectServiceManager.list()`) now returns a python list instead of a JSON string.

### 7.10.11 Changes from 0.19 to 0.20

- The `projects` attribute of `Group` objects is not a list of `Project` objects anymore. It is a `Manager` object giving access to `GroupProject` objects. To get the list of projects use:

```
group.projects.list()
```

Documentation: [http://python-gitlab.readthedocs.io/en/stable/gl\\_objects/groups.html#examples](http://python-gitlab.readthedocs.io/en/stable/gl_objects/groups.html#examples)

Related issue: <https://github.com/python-gitlab/python-gitlab/issues/209>

- The `Key` objects are deprecated in favor of the new `DeployKey` objects. They are exactly the same but the name makes more sense.

Documentation: [http://python-gitlab.readthedocs.io/en/stable/gl\\_objects/deploy\\_keys.html](http://python-gitlab.readthedocs.io/en/stable/gl_objects/deploy_keys.html)

Related issue: <https://github.com/python-gitlab/python-gitlab/issues/212>

## 7.11 FAQ

### 7.11.1 General

#### I cannot edit the merge request / issue I've just retrieved.

It is likely that you used a `MergeRequest`, `GroupMergeRequest`, `Issue` or `GroupIssue` object. These objects cannot be edited. But you can create a new `ProjectMergeRequest` or `ProjectIssue` object to apply changes. For example:

```
issue = gl.issues.list()[0]
project = gl.projects.get(issue.project_id, lazy=True)
editable_issue = project.issues.get(issue.iid, lazy=True)
# you can now edit the object
```

See the *merge requests example* and the *issues examples*.

## How can I clone the repository of a project?

python-gitlab does not provide an API to clone a project. You have to use a git library or call the `git` command.

The git URI is exposed in the `ssh_url_to_repo` attribute of `Project` objects.

Example:

```
import subprocess

project = gl.projects.create(data) # or gl.projects.get(project_id)
print(project.attributes) # displays all the attributes
git_url = project.ssh_url_to_repo
subprocess.call(['git', 'clone', git_url])
```

## Not all items are returned from the API

If you've passed `all=True` to the API and still cannot see all items returned, use `get_all=True` (or `--get-all` via the CLI) instead. See [Pagination](#) for more details.

## 7.11.2 Common errors

### AttributeError when accessing object attributes retrieved via `list()`

Fetching a list of objects does not always include all attributes in the objects. To retrieve an object with all attributes, use a `get()` call.

Example with projects:

```
for projects in gl.projects.list():
    # Retrieve project object with all attributes
    project = gl.projects.get(project.id)
```

### AttributeError when accessing attributes after `save()` or `refresh()`

You are most likely trying to access an attribute that was not returned by the server on the second request. Please look at the documentation in [Attributes in updated objects](#) to see how to avoid this.

### TypeError when accessing object attributes

When you encounter errors such as `object is not iterable` or `object is not subscriptable` when trying to access object attributes returned from the server, you are most likely trying to access an attribute that is shadowed by python-gitlab's own methods or managers.

You can use the object's `attributes` dictionary to access it directly instead. See the [Gitlab Objects](#) section for more details on how attributes are exposed.



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### g

- `gitlab.base`, 443
- `gitlab.cli`, 445
- `gitlab.config`, 446
- `gitlab.const`, 447
- `gitlab.exceptions`, 448
- `gitlab.mixins`, 465
- `gitlab.utils`, 474
- `gitlab.v4`, 443
- `gitlab.v4.objects`, 198



## Symbols

- access-level
  - gitlab-current-user-runner-create command line option, [499](#)
  - gitlab-group-access-request-approve command line option, [513](#)
  - gitlab-group-access-token-create command line option, [515](#)
  - gitlab-group-invitation-create command line option, [547](#)
  - gitlab-group-invitation-update command line option, [548](#)
  - gitlab-group-member-create command line option, [555](#)
  - gitlab-group-member-update command line option, [556](#)
  - gitlab-group-saml-group-link-create command line option, [568](#)
  - gitlab-project-access-request-approve command line option, [603](#)
  - gitlab-project-access-token-create command line option, [604](#)
  - gitlab-project-invitation-create command line option, [651](#)
  - gitlab-project-invitation-update command line option, [652](#)
  - gitlab-project-member-create command line option, [686](#)
  - gitlab-project-member-update command line option, [687](#)
  - gitlab-runner-create command line option, [786](#)
  - gitlab-runner-update command line option, [787](#)
- access-raw-diffs
  - gitlab-project-merge-request-changes command line option, [690](#)
- action
  - gitlab-event-list command line option, [501](#)
  - gitlab-project-event-list command line option, [640](#)
- gitlab-todo-list command line option, [795](#)
- gitlab-user-event-list command line option, [807](#)
- actions
  - gitlab-project-commit-create command line option, [622](#)
- active
  - gitlab-project-pipeline-schedule-create command line option, [735](#)
  - gitlab-project-pipeline-schedule-update command line option, [737](#)
  - gitlab-runner-create command line option, [786](#)
  - gitlab-runner-update command line option, [787](#)
  - gitlab-user-list command line option, [801](#)
- admin
  - gitlab-user-create command line option, [800](#)
  - gitlab-user-update command line option, [803](#)
- after
  - gitlab-event-list command line option, [501](#)
  - gitlab-project-event-list command line option, [640](#)
  - gitlab-user-event-list command line option, [807](#)
- after-sign-out-path
  - gitlab-application-settings-update command line option, [486](#)
- all
  - gitlab-project-commit-list command line option, [623](#)
- all-available
  - gitlab-group-descendant-group-list command line option, [530](#)
  - gitlab-group-list command line option, [510](#)
  - gitlab-group-subgroup-list command line option, [570](#)
- allow-collaboration

gitlab-project-merge-request-create command line option, <a href="#">691</a>	gitlab-project-merge-request-create command line option, <a href="#">691</a>
--allow-force-push gitlab-project-protected-branch-create command line option, <a href="#">742</a>	gitlab-project-update command line option, <a href="#">599</a>
--allow-local-requests-from-hooks-and-services gitlab-application-settings-update command line option, <a href="#">487</a>	--approvals-required gitlab-project-approval-rule-create command line option, <a href="#">607</a>
--allow-local-requests-from-system-hooks gitlab-application-settings-update command line option, <a href="#">487</a>	gitlab-project-merge-request-approval-rule-create command line option, <a href="#">701</a>
--allow-local-requests-from-web-hooks-and-services gitlab-application-settings-update command line option, <a href="#">487</a>	gitlab-project-merge-request-approval-rule-update command line option, <a href="#">703</a>
--allow-maintainer-to-push gitlab-project-merge-request-create command line option, <a href="#">691</a>	gitlab-project-merge-request-approval-update command line option, <a href="#">700</a>
gitlab-project-merge-request-update command line option, <a href="#">699</a>	--approved-by-ids gitlab-group-merge-request-list command line option, <a href="#">558</a>
--allow-merge-on-skipped-pipeline gitlab-project-create command line option, <a href="#">587</a>	gitlab-merge-request-list command line option, <a href="#">580</a>
gitlab-project-update command line option, <a href="#">599</a>	gitlab-project-deployment-merge-request-list command line option, <a href="#">637</a>
--allowed-to-merge gitlab-project-protected-branch-create command line option, <a href="#">742</a>	gitlab-project-merge-request-list command line option, <a href="#">694</a>
--allowed-to-push gitlab-project-protected-branch-create command line option, <a href="#">742</a>	--approver-ids gitlab-group-merge-request-list command line option, <a href="#">558</a>
--allowed-to-unprotect gitlab-project-protected-branch-create command line option, <a href="#">742</a>	gitlab-merge-request-list command line option, <a href="#">580</a>
--analytics-access-level gitlab-project-create command line option, <a href="#">587</a>	gitlab-project-deployment-merge-request-list command line option, <a href="#">637</a>
gitlab-project-update command line option, <a href="#">599</a>	gitlab-project-merge-request-list command line option, <a href="#">693</a>
--api-version gitlab command line option, <a href="#">481</a>	--archived gitlab-group-project-list command line option, <a href="#">564</a>
--approval-project-rule-id gitlab-project-merge-request-approval-rule-create command line option, <a href="#">701</a>	gitlab-project-fork-list command line option, <a href="#">645</a>
--approval-rule-id gitlab-project-merge-request-approval-rule-update command line option, <a href="#">702</a>	gitlab-project-list command line option, <a href="#">591</a>
--approval-rules gitlab-project-protected-environment-create-asset-proxy-allowlist command line option, <a href="#">744</a>	gitlab-shared-project-list command line option, <a href="#">790</a>
--approvals-before-merge gitlab-project-approval-update command line option, <a href="#">606</a>	gitlab-starred-project-list command line option, <a href="#">815</a>
gitlab-project-create command line option, <a href="#">587</a>	gitlab-user-project-list command line option, <a href="#">815</a>
	--artifact-path gitlab-project-artifact-raw command line option, <a href="#">609</a>
	gitlab-application-settings-update command line option, <a href="#">487</a>
	--asset-proxy-enabled gitlab-application-settings-update command line option, <a href="#">487</a>
	--asset-proxy-url

```

    gitlab-application-settings-update
        command line option, 487
--asset-proxy-whitelist
    gitlab-application-settings-update
        command line option, 487
--assets
    gitlab-project-release-create command
        line option, 752
--assignee-id
    gitlab-group-board-list-create command
        line option, 523
    gitlab-group-issue-list command line
        option, 549
    gitlab-group-merge-request-list command
        line option, 558
    gitlab-issue-list command line option,
        576
    gitlab-merge-request-list command line
        option, 580
    gitlab-project-board-list-create
        command line option, 615
    gitlab-project-deployment-merge-request-list
        command line option, 637
    gitlab-project-issue-create command
        line option, 654
    gitlab-project-issue-list command line
        option, 656
    gitlab-project-issue-update command
        line option, 660
    gitlab-project-merge-request-create
        command line option, 691
    gitlab-project-merge-request-list
        command line option, 693
    gitlab-project-merge-request-update
        command line option, 699
--assignee-ids
    gitlab-project-issue-create command
        line option, 654
    gitlab-project-issue-update command
        line option, 660
    gitlab-project-merge-request-create
        command line option, 691
--author-email
    gitlab-project-commit-create command
        line option, 622
    gitlab-project-file-create command line
        option, 642
    gitlab-project-file-update command line
        option, 644
--author-email-regex
    gitlab-group-push-rules-create command
        line option, 566
    gitlab-group-push-rules-update command
        line option, 567
    gitlab-project-push-rules-create
        command line option, 747
    gitlab-project-push-rules-update
        command line option, 749
--author-id
    gitlab-group-epic-list command line
        option, 532
    gitlab-group-issue-list command line
        option, 549
    gitlab-group-merge-request-list command
        line option, 558
    gitlab-issue-list command line option,
        576
    gitlab-merge-request-list command line
        option, 580
    gitlab-project-deployment-merge-request-list
        command line option, 637
    gitlab-project-issue-list command line
        option, 656
    gitlab-project-merge-request-list
        command line option, 693
    gitlab-todo-list command line option, 795
--author-name
    gitlab-project-commit-create command
        line option, 622
    gitlab-project-file-create command line
        option, 642
    gitlab-project-file-update command line
        option, 644
--author-username
    gitlab-merge-request-list command line
        option, 580
    gitlab-project-deployment-merge-request-list
        command line option, 637
--auto-cancel-pending-pipelines
    gitlab-project-create command line
        option, 587
    gitlab-project-update command line
        option, 599
--auto-devops-deploy-strategy
    gitlab-project-create command line
        option, 587
    gitlab-project-update command line
        option, 599
--auto-devops-enabled
    gitlab-group-create command line option,
        508
    gitlab-group-update command line option,
        512
    gitlab-project-create command line
        option, 587
    gitlab-project-update command line
        option, 599
--autoclose-referenced-issues

```

- gitlab-project-create command line option, [587](#)
- gitlab-project-update command line option, [599](#)
- avatar
  - gitlab-group-create command line option, [508](#)
  - gitlab-group-update command line option, [513](#)
  - gitlab-project-create command line option, [587](#)
  - gitlab-project-update command line option, [599](#)
  - gitlab-topic-create command line option, [796](#)
  - gitlab-topic-update command line option, [797](#)
  - gitlab-user-create command line option, [800](#)
  - gitlab-user-update command line option, [804](#)
- before
  - gitlab-event-list command line option, [501](#)
  - gitlab-project-event-list command line option, [640](#)
  - gitlab-user-event-list command line option, [807](#)
- bio
  - gitlab-user-create command line option, [799](#)
  - gitlab-user-update command line option, [803](#)
- blocked
  - gitlab-user-list command line option, [801](#)
- board-id
  - gitlab-group-board-list-create command line option, [523](#)
  - gitlab-group-board-list-delete command line option, [523](#)
  - gitlab-group-board-list-get command line option, [524](#)
  - gitlab-group-board-list-list command line option, [524](#)
  - gitlab-group-board-list-update command line option, [524](#)
  - gitlab-project-board-list-create command line option, [615](#)
  - gitlab-project-board-list-delete command line option, [615](#)
  - gitlab-project-board-list-get command line option, [615](#)
  - gitlab-project-board-list-list command line option, [616](#)
  - gitlab-project-board-list-update command line option, [616](#)
- body
  - gitlab-group-epic-discussion-note-create command line option, [534](#)
  - gitlab-group-epic-discussion-note-update command line option, [536](#)
  - gitlab-group-epic-note-create command line option, [538](#)
  - gitlab-group-epic-note-update command line option, [539](#)
  - gitlab-project-commit-discussion-create command line option, [626](#)
  - gitlab-project-commit-discussion-note-create command line option, [628](#)
  - gitlab-project-commit-discussion-note-update command line option, [629](#)
  - gitlab-project-issue-discussion-create command line option, [663](#)
  - gitlab-project-issue-discussion-note-create command line option, [664](#)
  - gitlab-project-issue-discussion-note-update command line option, [665](#)
  - gitlab-project-issue-note-create command line option, [667](#)
  - gitlab-project-issue-note-update command line option, [669](#)
  - gitlab-project-merge-request-discussion-create command line option, [706](#)
  - gitlab-project-merge-request-discussion-note-create command line option, [708](#)
  - gitlab-project-merge-request-discussion-note-update command line option, [710](#)
  - gitlab-project-merge-request-note-create command line option, [713](#)
  - gitlab-project-merge-request-note-update command line option, [714](#)
  - gitlab-project-snippet-discussion-create command line option, [769](#)
  - gitlab-project-snippet-discussion-note-create command line option, [771](#)
  - gitlab-project-snippet-discussion-note-update command line option, [772](#)
  - gitlab-project-snippet-note-create command line option, [773](#)
  - gitlab-project-snippet-note-update command line option, [774](#)
- branch
  - gitlab-project-branch-create command line option, [616](#)
  - gitlab-project-commit-cherry-pick command line option, [622](#)
  - gitlab-project-commit-create command line option, [622](#)



gitlab-project-commit-revert command line option, [624](#)  
 gitlab-project-file-create command line option, [642](#)  
 gitlab-project-file-delete command line option, [643](#)  
 gitlab-project-file-update command line option, [644](#)  
 gitlab-project-update-submodule command line option, [602](#)  
 --branch-name-regex  
   gitlab-group-push-rules-create command line option, [566](#)  
   gitlab-group-push-rules-update command line option, [567](#)  
   gitlab-project-push-rules-create command line option, [748](#)  
   gitlab-project-push-rules-update command line option, [749](#)  
 --build-coverage-regex  
   gitlab-project-create command line option, [588](#)  
   gitlab-project-update command line option, [599](#)  
 --build-git-strategy  
   gitlab-project-create command line option, [588](#)  
   gitlab-project-update command line option, [599](#)  
 --build-timeout  
   gitlab-project-create command line option, [588](#)  
   gitlab-project-update command line option, [599](#)  
 --builds-access-level  
   gitlab-project-create command line option, [588](#)  
   gitlab-project-update command line option, [599](#)  
 --builds-enabled  
   gitlab-user-project-create command line option, [814](#)  
 --bulk-import-id  
   gitlab-bulk-import-entity-get command line option, [492](#)  
   gitlab-bulk-import-entity-list command line option, [493](#)  
 --can-create-group  
   gitlab-user-create command line option, [800](#)  
   gitlab-user-update command line option, [803](#)  
 --can-push  
   gitlab-project-key-create command line option, [681](#)  
   gitlab-project-key-update command line option, [682](#)  
 --certificate  
   gitlab-project-pages-domain-create command line option, [730](#)  
   gitlab-project-pages-domain-update command line option, [731](#)  
 --ci-allow-fork-pipelines-to-run-in-parent-project  
   gitlab-project-update command line option, [599](#)  
 --ci-config-path  
   gitlab-project-create command line option, [588](#)  
   gitlab-project-update command line option, [599](#)  
 --ci-default-git-depth  
   gitlab-project-update command line option, [599](#)  
 --ci-forward-deployment-enabled  
   gitlab-project-update command line option, [599](#)  
 --ci-separated-caches  
   gitlab-project-update command line option, [600](#)  
 --close-issue  
   gitlab-group-notification-settings-update command line option, [563](#)  
   gitlab-notification-settings-update command line option, [583](#)  
   gitlab-project-notification-settings-update command line option, [726](#)  
 --close-merge-request  
   gitlab-group-notification-settings-update command line option, [563](#)  
   gitlab-notification-settings-update command line option, [583](#)  
   gitlab-project-notification-settings-update command line option, [726](#)  
 --cn  
   gitlab-group-ldap-group-link-create command line option, [554](#)  
 --code-owner-approval-required  
   gitlab-project-protected-branch-create command line option, [742](#)  
 --color  
   gitlab-broadcast-message-create command line option, [489](#)  
   gitlab-broadcast-message-update command line option, [490](#)  
   gitlab-group-label-create command line option, [551](#)  
   gitlab-group-label-update command line option, [553](#)

- gitlab-project-label-create command
  - line option, [683](#)
- gitlab-project-label-update command
  - line option, [685](#)
- color-scheme-id
  - gitlab-user-create command line option, [800](#)
  - gitlab-user-update command line option, [804](#)
- comment
  - gitlab-project-deployment-approval command line option, [633](#)
- commit-committer-check
  - gitlab-group-push-rules-create command line option, [566](#)
  - gitlab-group-push-rules-update command line option, [567](#)
  - gitlab-project-push-rules-create command line option, [748](#)
  - gitlab-project-push-rules-update command line option, [749](#)
- commit-id
  - gitlab-project-commit-comment-create command line option, [625](#)
  - gitlab-project-commit-comment-list command line option, [626](#)
  - gitlab-project-commit-discussion-create command line option, [626](#)
  - gitlab-project-commit-discussion-get command line option, [627](#)
  - gitlab-project-commit-discussion-list command line option, [627](#)
  - gitlab-project-commit-discussion-note-create command line option, [628](#)
  - gitlab-project-commit-discussion-note-delete command line option, [628](#)
  - gitlab-project-commit-discussion-note-get command line option, [629](#)
  - gitlab-project-commit-discussion-note-update command line option, [629](#)
  - gitlab-project-commit-status-create command line option, [630](#)
  - gitlab-project-commit-status-list command line option, [630](#)
  - gitlab-project-merge-request-draft-note-create command line option, [711](#)
- commit-message
  - gitlab-project-commit-create command line option, [622](#)
  - gitlab-project-file-create command line option, [642](#)
  - gitlab-project-file-delete command line option, [643](#)
  - gitlab-project-file-update command line option, [644](#)
- commit-message-negative-regex
  - gitlab-group-push-rules-create command line option, [566](#)
  - gitlab-group-push-rules-update command line option, [567](#)
  - gitlab-project-push-rules-create command line option, [748](#)
  - gitlab-project-push-rules-update command line option, [749](#)
- commit-message-regex
  - gitlab-group-push-rules-create command line option, [566](#)
  - gitlab-group-push-rules-update command line option, [567](#)
  - gitlab-project-push-rules-create command line option, [748](#)
  - gitlab-project-push-rules-update command line option, [749](#)
- commit-sha
  - gitlab-project-update-submodule command line option, [602](#)
- confidential
  - gitlab-application-create command line option, [482](#)
  - gitlab-project-issue-create command line option, [654](#)
  - gitlab-project-issue-update command line option, [660](#)
- confidential-issues-events
  - gitlab-group-hook-create command line option, [543](#)
  - gitlab-group-hook-update command line option, [545](#)
  - gitlab-project-hook-create command line option, [646](#)
  - gitlab-project-hook-update command line option, [648](#)
- confidential-note-events
  - gitlab-group-hook-create command line option, [544](#)
  - gitlab-group-hook-update command line option, [546](#)
- config-file
  - gitlab command line option, [480](#)
- configuration
  - gitlab-bulk-import-create command line option, [491](#)
- container-expiration-policy-attributes
  - gitlab-project-create command line option, [588](#)
  - gitlab-project-update command line option, [600](#)
- container-registry-access-level

gitlab-project-create command line option, 588  
 gitlab-project-update command line option, 600  
 --container-registry-enabled  
   gitlab-project-create command line option, 588  
   gitlab-project-update command line option, 600  
 --container-registry-token-expire-delay  
   gitlab-application-settings-update command line option, 486  
 --content  
   gitlab-ci-lint-create command line option, 493  
   gitlab-ci-lint-validate command line option, 494  
   gitlab-group-wiki-create command line option, 573  
   gitlab-group-wiki-update command line option, 574  
   gitlab-project-ci-lint-create command line option, 618  
   gitlab-project-ci-lint-validate command line option, 619  
   gitlab-project-file-create command line option, 642  
   gitlab-project-file-update command line option, 644  
   gitlab-project-snippet-create command line option, 765  
   gitlab-project-snippet-update command line option, 767  
   gitlab-project-wiki-create command line option, 783  
   gitlab-project-wiki-update command line option, 784  
   gitlab-snippet-create command line option, 791  
   gitlab-snippet-update command line option, 793  
 --context  
   gitlab-project-commit-status-create command line option, 630  
 --coverage  
   gitlab-project-commit-status-create command line option, 630  
 --create-access-level  
   gitlab-project-protected-tag-create command line option, 746  
 --created-after  
   gitlab-audit-event-list command line option, 488  
   gitlab-group-audit-event-list command line option, 517  
   gitlab-group-issue-list command line option, 549  
   gitlab-group-merge-request-list command line option, 558  
   gitlab-issue-list command line option, 576  
   gitlab-merge-request-list command line option, 580  
   gitlab-project-audit-event-list command line option, 610  
   gitlab-project-deployment-merge-request-list command line option, 636  
   gitlab-project-issue-list command line option, 656  
   gitlab-project-merge-request-list command line option, 693  
 --created-at  
   gitlab-application-settings-update command line option, 486  
   gitlab-group-epic-discussion-note-create command line option, 535  
   gitlab-group-epic-note-create command line option, 538  
   gitlab-project-commit-discussion-create command line option, 626  
   gitlab-project-commit-discussion-note-create command line option, 628  
   gitlab-project-issue-create command line option, 654  
   gitlab-project-issue-discussion-create command line option, 663  
   gitlab-project-issue-discussion-note-create command line option, 664  
   gitlab-project-issue-note-create command line option, 667  
   gitlab-project-merge-request-discussion-create command line option, 707  
   gitlab-project-merge-request-discussion-note-create command line option, 709  
   gitlab-project-snippet-discussion-create command line option, 769  
   gitlab-project-snippet-discussion-note-create command line option, 771  
 --created-before  
   gitlab-audit-event-list command line option, 488  
   gitlab-group-audit-event-list command line option, 517  
   gitlab-group-issue-list command line option, 549  
   gitlab-group-merge-request-list command line option, 558  
   gitlab-issue-list command line option, 576

576  
gitlab-merge-request-list command line option, 580  
gitlab-project-audit-event-list command line option, 610  
gitlab-project-deployment-merge-request-list command line option, 637  
gitlab-project-issue-list command line option, 656  
gitlab-project-merge-request-list command line option, 693  
--cron  
    gitlab-project-pipeline-schedule-create command line option, 735  
    gitlab-project-pipeline-schedule-update command line option, 737  
--cron-timezone  
    gitlab-project-pipeline-schedule-create command line option, 735  
    gitlab-project-pipeline-schedule-update command line option, 737  
--custom-attributes  
    gitlab-user-list command line option, 801  
--debug  
    gitlab command line option, 480  
--default-branch  
    gitlab-project-create command line option, 588  
    gitlab-project-update command line option, 600  
    gitlab-user-project-create command line option, 814  
--default-branch-protection  
    gitlab-application-settings-update command line option, 486  
    gitlab-group-create command line option, 509  
    gitlab-group-update command line option, 513  
--default-group-visibility  
    gitlab-application-settings-update command line option, 486  
--default-project-visibility  
    gitlab-application-settings-update command line option, 486  
--default-projects-limit  
    gitlab-application-settings-update command line option, 485  
--default-snippet-visibility  
    gitlab-application-settings-update command line option, 486  
--deny-delete-tag  
    gitlab-group-push-rules-create command line option, 565  
    gitlab-group-push-rules-update command line option, 567  
    gitlab-project-push-rules-create command line option, 748  
    gitlab-project-push-rules-update command line option, 749  
--deploy-access-levels  
    gitlab-project-protected-environment-create command line option, 744  
--deployed-after  
    gitlab-merge-request-list command line option, 581  
    gitlab-project-deployment-merge-request-list command line option, 637  
--deployed-before  
    gitlab-merge-request-list command line option, 581  
    gitlab-project-deployment-merge-request-list command line option, 637  
--deployment-events  
    gitlab-group-hook-create command line option, 544  
    gitlab-group-hook-update command line option, 546  
--deployment-id  
    gitlab-project-deployment-merge-request-list command line option, 636  
--description  
    gitlab-application-appearance-update command line option, 483  
    gitlab-current-user-runner-create command line option, 498  
    gitlab-group-create command line option, 508  
    gitlab-group-epic-create command line option, 531  
    gitlab-group-epic-update command line option, 532  
    gitlab-group-label-create command line option, 551  
    gitlab-group-label-update command line option, 553  
    gitlab-group-milestone-create command line option, 559  
    gitlab-group-milestone-update command line option, 561  
    gitlab-group-update command line option, 512  
    gitlab-project-commit-status-create command line option, 630  
    gitlab-project-create command line option, 588  
    gitlab-project-export-create command line option, 641

```

gitlab-project-issue-create command
  line option, 654
gitlab-project-issue-update command
  line option, 660
gitlab-project-label-create command
  line option, 683
gitlab-project-label-update command
  line option, 685
gitlab-project-merge-request-create
  command line option, 692
gitlab-project-merge-request-update
  command line option, 699
gitlab-project-milestone-create command
  line option, 722
gitlab-project-milestone-update command
  line option, 724
gitlab-project-pipeline-schedule-create
  command line option, 735
gitlab-project-pipeline-schedule-update
  command line option, 737
gitlab-project-release-create command
  line option, 752
gitlab-project-release-update command
  line option, 754
gitlab-project-snippet-create command
  line option, 765
gitlab-project-snippet-update command
  line option, 767
gitlab-project-trigger-create command
  line option, 778
gitlab-project-trigger-update command
  line option, 779
gitlab-project-update command line
  option, 600
gitlab-runner-create command line
  option, 786
gitlab-runner-update command line
  option, 787
gitlab-snippet-create command line
  option, 791
gitlab-snippet-update command line
  option, 793
gitlab-topic-create command line option,
  796
gitlab-topic-update command line option,
  797
gitlab-user-project-create command line
  option, 814
--direct-asset-path
  gitlab-project-release-link-create
    command line option, 755
  gitlab-project-release-link-update
    command line option, 756
--disable-overriding-approvers-per-merge-request
  gitlab-project-approval-update command
    line option, 606
--disabled-oauth-sign-in-sources
  gitlab-application-settings-update
    command line option, 486
--discussion-id
  gitlab-group-epic-discussion-note-create
    command line option, 534
  gitlab-group-epic-discussion-note-delete
    command line option, 535
  gitlab-group-epic-discussion-note-get
    command line option, 535
  gitlab-group-epic-discussion-note-update
    command line option, 536
  gitlab-project-commit-discussion-note-create
    command line option, 628
  gitlab-project-commit-discussion-note-delete
    command line option, 628
  gitlab-project-commit-discussion-note-get
    command line option, 629
  gitlab-project-commit-discussion-note-update
    command line option, 629
  gitlab-project-issue-discussion-note-create
    command line option, 664
  gitlab-project-issue-discussion-note-delete
    command line option, 665
  gitlab-project-issue-discussion-note-get
    command line option, 665
  gitlab-project-issue-discussion-note-update
    command line option, 665
  gitlab-project-merge-request-discussion-note-create
    command line option, 708
  gitlab-project-merge-request-discussion-note-delete
    command line option, 709
  gitlab-project-merge-request-discussion-note-get
    command line option, 709
  gitlab-project-merge-request-discussion-note-update
    command line option, 710
  gitlab-project-snippet-discussion-note-create
    command line option, 771
  gitlab-project-snippet-discussion-note-delete
    command line option, 771
  gitlab-project-snippet-discussion-note-get
    command line option, 772
  gitlab-project-snippet-discussion-note-update
    command line option, 772
--discussion-locked
  gitlab-project-issue-update command
    line option, 660
  gitlab-project-merge-request-update
    command line option, 699
--discussion-to-resolve
  gitlab-project-issue-create command
    line option, 654

```

--domain  
    gitlab-group-cluster-create command line option, [525](#)  
    gitlab-group-cluster-update command line option, [526](#)  
    gitlab-project-cluster-create command line option, [619](#)  
    gitlab-project-cluster-update command line option, [621](#)  
    gitlab-project-pages-domain-create command line option, [729](#)  
    gitlab-project-pages-domain-delete command line option, [730](#)  
    gitlab-project-pages-domain-get command line option, [730](#)  
    gitlab-project-pages-domain-update command line option, [731](#)  
--domain-allowlist  
    gitlab-application-settings-update command line option, [486](#)  
--domain-blacklist  
    gitlab-application-settings-update command line option, [486](#)  
--domain-blacklist-enabled  
    gitlab-application-settings-update command line option, [486](#)  
--domain-denylist  
    gitlab-application-settings-update command line option, [486](#)  
--domain-denylist-enabled  
    gitlab-application-settings-update command line option, [486](#)  
--domain-whitelist  
    gitlab-application-settings-update command line option, [486](#)  
--dry-run  
    gitlab-project-ci-lint-create command line option, [618](#)  
    gitlab-project-ci-lint-get command line option, [618](#)  
    gitlab-project-ci-lint-validate command line option, [619](#)  
--dsa-key-restriction  
    gitlab-application-settings-update command line option, [487](#)  
--due-date  
    gitlab-group-milestone-create command line option, [559](#)  
    gitlab-group-milestone-update command line option, [561](#)  
    gitlab-project-issue-create command line option, [654](#)  
    gitlab-project-issue-update command line option, [660](#)  
    gitlab-project-milestone-create command line option, [722](#)  
    gitlab-project-milestone-update command line option, [724](#)  
--duration  
    gitlab-project-issue-add-spent-time command line option, [653](#)  
    gitlab-project-issue-time-estimate command line option, [659](#)  
    gitlab-project-merge-request-add-spent-time command line option, [689](#)  
    gitlab-project-merge-request-time-estimate command line option, [697](#)  
--ecdsa-key-restriction  
    gitlab-application-settings-update command line option, [487](#)  
--ed25519-key-restriction  
    gitlab-application-settings-update command line option, [487](#)  
--email  
    gitlab-current-user-email-create command line option, [494](#)  
    gitlab-group-invitation-create command line option, [547](#)  
    gitlab-group-invitation-delete command line option, [547](#)  
    gitlab-group-invitation-get command line option, [548](#)  
    gitlab-group-invitation-update command line option, [548](#)  
    gitlab-project-invitation-create command line option, [651](#)  
    gitlab-project-invitation-delete command line option, [651](#)  
    gitlab-project-invitation-get command line option, [652](#)  
    gitlab-project-invitation-update command line option, [652](#)  
    gitlab-user-create command line option, [799](#)  
    gitlab-user-email-create command line option, [806](#)  
    gitlab-user-update command line option, [803](#)  
--email-header-and-footer-enabled  
    gitlab-application-appearance-update command line option, [483](#)  
--emails-disabled  
    gitlab-group-create command line option, [508](#)  
    gitlab-group-update command line option, [512](#)  
    gitlab-project-create command line option, [588](#)



---

```

    gitlab-project-update command line
        option, 600
--emoji
    gitlab-current-user-status-update
        command line option, 499
--enable-ssl-verification
    gitlab-group-hook-create command line
        option, 544
    gitlab-group-hook-update command line
        option, 546
    gitlab-project-hook-create command line
        option, 646
    gitlab-project-hook-update command line
        option, 648
--enabled
    gitlab-geo-node-update command line
        option, 505
    gitlab-group-cluster-create command
        line option, 525
    gitlab-project-cluster-create command
        line option, 619
    gitlab-project-remote-mirror-create
        command line option, 757
    gitlab-project-remote-mirror-update
        command line option, 758
--encoding
    gitlab-project-file-create command line
        option, 642
    gitlab-project-file-update command line
        option, 644
--end-date
    gitlab-group-epic-create command line
        option, 531
    gitlab-group-epic-update command line
        option, 532
--ends-at
    gitlab-broadcast-message-create command
        line option, 489
    gitlab-broadcast-message-update command
        line option, 490
--enforce-auth-checks-on-uploads
    gitlab-project-update command line
        option, 600
--enforce-terms
    gitlab-application-settings-update
        command line option, 487
--entities
    gitlab-bulk-import-create command line
        option, 491
--entity-id
    gitlab-audit-event-list command line
        option, 488
--entity-type
    gitlab-audit-event-list command line
        option, 488
--environment
    gitlab-merge-request-list command line
        option, 581
    gitlab-project-deployment-create
        command line option, 634
    gitlab-project-deployment-list command
        line option, 635
    gitlab-project-deployment-merge-request-list
        command line option, 637
--environment-scope
    gitlab-group-cluster-create command
        line option, 525
    gitlab-group-cluster-update command
        line option, 527
    gitlab-project-cluster-create command
        line option, 620
    gitlab-project-cluster-update command
        line option, 621
    gitlab-project-variable-create command
        line option, 781
    gitlab-project-variable-update command
        line option, 782
--environments-access-level
    gitlab-project-create command line
        option, 589
    gitlab-project-update command line
        option, 601
--epic-id
    gitlab-group-epic-discussion-note-create
        command line option, 534
    gitlab-group-epic-discussion-note-delete
        command line option, 535
    gitlab-group-epic-discussion-note-get
        command line option, 535
    gitlab-group-epic-discussion-note-update
        command line option, 536
    gitlab-group-epic-note-create command
        line option, 538
    gitlab-group-epic-note-delete command
        line option, 538
    gitlab-group-epic-note-get command line
        option, 538
    gitlab-group-epic-note-list command
        line option, 539
    gitlab-group-epic-note-update command
        line option, 539
    gitlab-group-epic-resource-label-event-get
        command line option, 541
    gitlab-group-epic-resource-label-event-list
        command line option, 542
--epic-iid
    gitlab-group-epic-award-emoji-create
        command line option, 533

```

- gitlab-group-epic-award-emoji-delete
  - command line option, [533](#)
- gitlab-group-epic-award-emoji-get
  - command line option, [533](#)
- gitlab-group-epic-award-emoji-list
  - command line option, [534](#)
- gitlab-group-epic-issue-create command line option, [536](#)
- gitlab-group-epic-issue-delete command line option, [536](#)
- gitlab-group-epic-issue-list command line option, [537](#)
- gitlab-group-epic-issue-update command line option, [537](#)
- gitlab-group-epic-note-award-emoji-create command line option, [540](#)
- gitlab-group-epic-note-award-emoji-delete command line option, [540](#)
- gitlab-group-epic-note-award-emoji-get command line option, [540](#)
- gitlab-group-epic-note-award-emoji-list command line option, [541](#)
- epic-issue-id
  - gitlab-group-epic-issue-delete command line option, [536](#)
  - gitlab-group-epic-issue-update command line option, [537](#)
- exclude-subgroups
  - gitlab-group-package-list command line option, [563](#)
- expires-at
  - gitlab-group-access-token-create command line option, [515](#)
  - gitlab-group-deploy-token-create command line option, [528](#)
  - gitlab-group-invitation-create command line option, [547](#)
  - gitlab-group-invitation-update command line option, [548](#)
  - gitlab-group-member-create command line option, [555](#)
  - gitlab-group-member-update command line option, [556](#)
  - gitlab-group-share command line option, [511](#)
  - gitlab-project-access-token-create command line option, [604](#)
  - gitlab-project-deploy-token-create command line option, [632](#)
  - gitlab-project-invitation-create command line option, [651](#)
  - gitlab-project-invitation-update command line option, [652](#)
  - gitlab-project-member-create command line option, [686](#)
  - gitlab-project-member-update command line option, [687](#)
  - gitlab-project-share command line option, [595](#)
  - gitlab-user-impersonation-token-create command line option, [809](#)
  - gitlab-user-personal-access-token-create command line option, [813](#)
- extern-uid
  - gitlab-user-create command line option, [799](#)
  - gitlab-user-list command line option, [801](#)
  - gitlab-user-update command line option, [803](#)
- external
  - gitlab-user-create command line option, [800](#)
  - gitlab-user-list command line option, [801](#)
  - gitlab-user-update command line option, [803](#)
- external-authorization-classification-label
  - gitlab-project-create command line option, [588](#)
  - gitlab-project-update command line option, [600](#)
- external-authorization-service-default-label
  - gitlab-application-settings-update command line option, [486](#)
- external-authorization-service-enabled
  - gitlab-application-settings-update command line option, [486](#)
- external-authorization-service-timeout
  - gitlab-application-settings-update command line option, [486](#)
- external-authorization-service-url
  - gitlab-application-settings-update command line option, [486](#)
- external-url
  - gitlab-project-environment-create command line option, [638](#)
  - gitlab-project-environment-update command line option, [640](#)
- extra-shared-runners-minutes-limit
  - gitlab-group-create command line option, [509](#)
  - gitlab-group-update command line option, [513](#)
- fancy
  - gitlab command line option, [480](#)
- favicon
  - gitlab-application-appearance-update command line option, [483](#)
- feature-flags-access-level



```

gitlab-project-create command line
  option, 589
gitlab-project-update command line
  option, 601
--fields
  gitlab command line option, 480
--file
  gitlab-project-secure-file-create
    command line option, 761
--file-name
  gitlab-generic-package-download command
    line option, 503
  gitlab-generic-package-upload command
    line option, 503
  gitlab-project-snippet-create command
    line option, 765
  gitlab-project-snippet-update command
    line option, 767
  gitlab-snippet-create command line
    option, 791
  gitlab-snippet-update command line
    option, 793
--file-name-regex
  gitlab-group-push-rules-create command
    line option, 566
  gitlab-group-push-rules-update command
    line option, 567
  gitlab-project-push-rules-create
    command line option, 748
  gitlab-project-push-rules-update
    command line option, 749
--file-path
  gitlab-project-file-blame command line
    option, 642
  gitlab-project-file-create command line
    option, 642
  gitlab-project-file-delete command line
    option, 643
  gitlab-project-file-get command line
    option, 643
  gitlab-project-file-raw command line
    option, 643
  gitlab-project-file-update command line
    option, 644
--file-template-project-id
  gitlab-application-settings-update
    command line option, 487
  gitlab-group-update command line option,
    513
--filename
  gitlab-project-upload command line
    option, 602
  gitlab-project-wiki-upload command line
    option, 785
--filepath
  gitlab-project-release-link-create
    command line option, 754
  gitlab-project-release-link-update
    command line option, 756
  gitlab-project-upload command line
    option, 602
  gitlab-project-wiki-upload command line
    option, 785
--files
  gitlab-project-snippet-create command
    line option, 765
  gitlab-project-snippet-update command
    line option, 767
  gitlab-snippet-create command line
    option, 791
  gitlab-snippet-update command line
    option, 792
--files-max-capacity
  gitlab-geo-node-update command line
    option, 505
--filter
  gitlab-group-ldap-group-link-create
    command line option, 554
--first-day-of-week
  gitlab-application-settings-update
    command line option, 487
--first-parent
  gitlab-project-commit-list command line
    option, 623
--font
  gitlab-broadcast-message-create command
    line option, 489
  gitlab-broadcast-message-update command
    line option, 490
--footer-message
  gitlab-application-appearance-update
    command line option, 483
--forked-from-id
  gitlab-project-create-fork-relation
    command line option, 590
--forking-access-level
  gitlab-project-create command line
    option, 588
  gitlab-project-update command line
    option, 600
--format
  gitlab-group-wiki-create command line
    option, 573
  gitlab-group-wiki-update command line
    option, 574
  gitlab-project-repository-archive
    command line option, 593
  gitlab-project-wiki-create command line

```

- option, [783](#)
- gitlab-project-wiki-update command line option, [784](#)
- from-
  - gitlab-project-repository-compare command line option, [593](#)
- fullname
  - gitlab-license-get command line option, [578](#)
- geo-node-allowed-ips
  - gitlab-application-settings-update command line option, [487](#)
- get-all
  - gitlab-application-list command line option, [482](#)
  - gitlab-audit-event-list command line option, [488](#)
  - gitlab-broadcast-message-list command line option, [490](#)
  - gitlab-bulk-import-all-entity-list command line option, [492](#)
  - gitlab-bulk-import-entity-list command line option, [493](#)
  - gitlab-bulk-import-list command line option, [491](#)
  - gitlab-current-user-email-list command line option, [495](#)
  - gitlab-current-user-gpg-key-list command line option, [496](#)
  - gitlab-current-user-key-list command line option, [498](#)
  - gitlab-deploy-key-list command line option, [500](#)
  - gitlab-deploy-token-list command line option, [500](#)
  - gitlab-dockerfile-list command line option, [501](#)
  - gitlab-event-list command line option, [502](#)
  - gitlab-feature-list command line option, [502](#)
  - gitlab-geo-node-list command line option, [505](#)
  - gitlab-gitignore-list command line option, [506](#)
  - gitlab-gitlabciyaml-list command line option, [507](#)
  - gitlab-group-access-request-list command line option, [514](#)
  - gitlab-group-access-token-list command line option, [516](#)
  - gitlab-group-audit-event-list command line option, [517](#)
  - gitlab-group-badge-list command line option, [518](#)
  - gitlab-group-billable-member-list command line option, [520](#)
  - gitlab-group-billable-member-membership-list command line option, [521](#)
  - gitlab-group-board-list command line option, [522](#)
  - gitlab-group-board-list-list command line option, [524](#)
  - gitlab-group-cluster-list command line option, [526](#)
  - gitlab-group-custom-attribute-list command line option, [528](#)
  - gitlab-group-deploy-token-list command line option, [529](#)
  - gitlab-group-descendant-group-list command line option, [530](#)
  - gitlab-group-epic-award-emoji-list command line option, [534](#)
  - gitlab-group-epic-issue-list command line option, [537](#)
  - gitlab-group-epic-list command line option, [532](#)
  - gitlab-group-epic-note-award-emoji-list command line option, [541](#)
  - gitlab-group-epic-note-list command line option, [539](#)
  - gitlab-group-epic-resource-label-event-list command line option, [542](#)
  - gitlab-group-hook-list command line option, [545](#)
  - gitlab-group-invitation-list command line option, [548](#)
  - gitlab-group-issue-list command line option, [550](#)
  - gitlab-group-iteration-list command line option, [551](#)
  - gitlab-group-label-list command line option, [552](#)
  - gitlab-group-ldap-group-link-list command line option, [555](#)
  - gitlab-group-list command line option, [510](#)
  - gitlab-group-member-all-list command line option, [557](#)
  - gitlab-group-member-list command line option, [556](#)
  - gitlab-group-merge-request-list command line option, [559](#)
  - gitlab-group-milestone-list command line option, [561](#)
  - gitlab-group-package-list command line option, [564](#)
  - gitlab-group-project-list command line

- option, [565](#)
- gitlab-group-runner-list command line option, [568](#)
- gitlab-group-saml-group-link-list command line option, [569](#)
- gitlab-group-subgroup-list command line option, [570](#)
- gitlab-group-variable-list command line option, [572](#)
- gitlab-group-wiki-list command line option, [573](#)
- gitlab-hook-list command line option, [575](#)
- gitlab-issue-list command line option, [576](#)
- gitlab-ldap-group-list command line option, [578](#)
- gitlab-license-list command line option, [579](#)
- gitlab-merge-request-list command line option, [581](#)
- gitlab-namespace-list command line option, [582](#)
- gitlab-pages-domain-list command line option, [583](#)
- gitlab-personal-access-token-list command line option, [584](#)
- gitlab-project-access-request-list command line option, [604](#)
- gitlab-project-access-token-list command line option, [605](#)
- gitlab-project-approval-rule-list command line option, [608](#)
- gitlab-project-audit-event-list command line option, [610](#)
- gitlab-project-badge-list command line option, [612](#)
- gitlab-project-board-list command line option, [614](#)
- gitlab-project-board-list-list command line option, [616](#)
- gitlab-project-branch-list command line option, [617](#)
- gitlab-project-cluster-list command line option, [620](#)
- gitlab-project-commit-comment-list command line option, [626](#)
- gitlab-project-commit-discussion-list command line option, [627](#)
- gitlab-project-commit-list command line option, [623](#)
- gitlab-project-commit-status-list command line option, [630](#)
- gitlab-project-custom-attribute-list command line option, [631](#)
- gitlab-project-deploy-token-list command line option, [633](#)
- gitlab-project-deployment-list command line option, [635](#)
- gitlab-project-deployment-merge-request-list command line option, [637](#)
- gitlab-project-environment-list command line option, [639](#)
- gitlab-project-event-list command line option, [640](#)
- gitlab-project-fork-list command line option, [645](#)
- gitlab-project-hook-list command line option, [647](#)
- gitlab-project-integration-list command line option, [650](#)
- gitlab-project-invitation-list command line option, [652](#)
- gitlab-project-issue-award-emoji-list command line option, [662](#)
- gitlab-project-issue-discussion-list command line option, [664](#)
- gitlab-project-issue-link-list command line option, [667](#)
- gitlab-project-issue-list command line option, [656](#)
- gitlab-project-issue-note-award-emoji-list command line option, [671](#)
- gitlab-project-issue-note-list command line option, [668](#)
- gitlab-project-issue-resource-iteration-event-list command line option, [672](#)
- gitlab-project-issue-resource-label-event-list command line option, [673](#)
- gitlab-project-issue-resource-milestone-event-list command line option, [674](#)
- gitlab-project-issue-resource-state-event-list command line option, [675](#)
- gitlab-project-issue-resource-weight-event-list command line option, [676](#)
- gitlab-project-job-list command line option, [679](#)
- gitlab-project-key-list command line option, [682](#)
- gitlab-project-label-list command line option, [684](#)
- gitlab-project-list command line option, [592](#)
- gitlab-project-member-all-list command line option, [688](#)
- gitlab-project-member-list command line option, [687](#)
- gitlab-project-merge-request-approval-rule-list command line option, [702](#)

gitlab-project-merge-request-award-emoji-list command line option, 705	gitlab-project-protected-tag-list command line option, 747
gitlab-project-merge-request-diff-list command line option, 706	gitlab-project-registry-repository-list command line option, 750
gitlab-project-merge-request-discussion-list command line option, 707	gitlab-project-registry-tag-list command line option, 752
gitlab-project-merge-request-draft-note-list command line option, 712	gitlab-project-release-link-list command line option, 756
gitlab-project-merge-request-list command line option, 694	gitlab-project-release-list command line option, 753
gitlab-project-merge-request-note-award-emoji-list command line option, 716	gitlab-project-remote-mirror-list command line option, 757
gitlab-project-merge-request-note-list command line option, 714	gitlab-project-resource-group-list command line option, 759
gitlab-project-merge-request-pipeline-list command line option, 717	gitlab-project-resource-group-upcoming-job-list command line option, 760
gitlab-project-merge-request-resource-label-event-list command line option, 718	gitlab-project-runner-list command line option, 761
gitlab-project-merge-request-resource-milestone-event-list command line option, 719	gitlab-project-secure-file-list command line option, 763
gitlab-project-merge-request-resource-state-event-list command line option, 720	gitlab-project-service-list command line option, 764
gitlab-project-merge-request-reviewer-detail-list command line option, 720	gitlab-project-snippet-award-emoji-list command line option, 769
gitlab-project-merge-train-list command line option, 721	gitlab-project-snippet-discussion-list command line option, 770
gitlab-project-milestone-list command line option, 723	gitlab-project-snippet-list command line option, 766
gitlab-project-note-list command line option, 725	gitlab-project-snippet-note-award-emoji-list command line option, 776
gitlab-project-package-file-list command line option, 728	gitlab-project-snippet-note-list command line option, 774
gitlab-project-package-list command line option, 727	gitlab-project-tag-list command line option, 778
gitlab-project-package-pipeline-list command line option, 729	gitlab-project-trigger-list command line option, 779
gitlab-project-pages-domain-list command line option, 730	gitlab-project-user-list command line option, 780
gitlab-project-pipeline-bridge-list command line option, 734	gitlab-project-variable-list command line option, 782
gitlab-project-pipeline-job-list command line option, 734	gitlab-project-wiki-list command line option, 784
gitlab-project-pipeline-list command line option, 733	gitlab-runner-all-list command line option, 788
gitlab-project-pipeline-schedule-list command line option, 736	gitlab-runner-job-list command line option, 789
gitlab-project-pipeline-schedule-pipeline-list command line option, 738	gitlab-runner-list command line option, 787
gitlab-project-pipeline-variable-list command line option, 741	gitlab-shared-project-list command line option, 790
gitlab-project-protected-branch-list command line option, 743	gitlab-snippet-list command line option, 792
gitlab-project-protected-environment-list command line option, 745	gitlab-starred-project-list command line option, 794

- gitlab-todo-list command line option, [795](#)
- gitlab-topic-list command line option, [797](#)
- gitlab-user-activities-list command line option, [804](#)
- gitlab-user-custom-attribute-list command line option, [805](#)
- gitlab-user-email-list command line option, [807](#)
- gitlab-user-event-list command line option, [808](#)
- gitlab-user-gpg-key-list command line option, [809](#)
- gitlab-user-impersonation-token-list command line option, [810](#)
- gitlab-user-key-list command line option, [812](#)
- gitlab-user-list command line option, [802](#)
- gitlab-user-membership-list command line option, [813](#)
- gitlab-user-project-list command line option, [816](#)
- gitlab-variable-list command line option, [817](#)
- gitlab
  - gitlab command line option, [480](#)
- gravatar-enabled
  - gitlab-application-settings-update command line option, [486](#)
- group-access
  - gitlab-group-ldap-group-link-create command line option, [554](#)
  - gitlab-group-share command line option, [511](#)
  - gitlab-project-share command line option, [595](#)
- group-id
  - gitlab-current-user-runner-create command line option, [498](#)
  - gitlab-group-access-request-approve command line option, [513](#)
  - gitlab-group-access-request-create command line option, [514](#)
  - gitlab-group-access-request-delete command line option, [514](#)
  - gitlab-group-access-request-list command line option, [514](#)
  - gitlab-group-access-token-create command line option, [515](#)
  - gitlab-group-access-token-delete command line option, [515](#)
  - gitlab-group-access-token-get command line option, [515](#)
  - gitlab-group-access-token-list command line option, [516](#)
  - gitlab-group-audit-event-get command line option, [516](#)
  - gitlab-group-audit-event-list command line option, [517](#)
  - gitlab-group-badge-create command line option, [517](#)
  - gitlab-group-badge-delete command line option, [518](#)
  - gitlab-group-badge-get command line option, [518](#)
  - gitlab-group-badge-list command line option, [518](#)
  - gitlab-group-badge-render command line option, [519](#)
  - gitlab-group-badge-update command line option, [519](#)
  - gitlab-group-billable-member-delete command line option, [520](#)
  - gitlab-group-billable-member-list command line option, [520](#)
  - gitlab-group-billable-member-membership-list command line option, [520](#)
  - gitlab-group-board-create command line option, [521](#)
  - gitlab-group-board-delete command line option, [521](#)
  - gitlab-group-board-get command line option, [522](#)
  - gitlab-group-board-list command line option, [522](#)
  - gitlab-group-board-list-create command line option, [523](#)
  - gitlab-group-board-list-delete command line option, [523](#)
  - gitlab-group-board-list-get command line option, [524](#)
  - gitlab-group-board-list-list command line option, [524](#)
  - gitlab-group-board-list-update command line option, [524](#)
  - gitlab-group-board-update command line option, [522](#)
  - gitlab-group-cluster-create command line option, [525](#)
  - gitlab-group-cluster-delete command line option, [525](#)
  - gitlab-group-cluster-get command line option, [526](#)
  - gitlab-group-cluster-list command line option, [526](#)
  - gitlab-group-cluster-update command line option, [526](#)
  - gitlab-group-custom-attribute-delete

command line option, [527](#)  
gitlab-group-custom-attribute-get  
command line option, [527](#)  
gitlab-group-custom-attribute-list  
command line option, [528](#)  
gitlab-group-deploy-token-create  
command line option, [528](#)  
gitlab-group-deploy-token-delete  
command line option, [528](#)  
gitlab-group-deploy-token-get command  
line option, [529](#)  
gitlab-group-deploy-token-list command  
line option, [529](#)  
gitlab-group-descendant-group-list  
command line option, [530](#)  
gitlab-group-epic-award-emoji-create  
command line option, [533](#)  
gitlab-group-epic-award-emoji-delete  
command line option, [533](#)  
gitlab-group-epic-award-emoji-get  
command line option, [533](#)  
gitlab-group-epic-award-emoji-list  
command line option, [534](#)  
gitlab-group-epic-create command line  
option, [531](#)  
gitlab-group-epic-delete command line  
option, [531](#)  
gitlab-group-epic-discussion-note-create  
command line option, [534](#)  
gitlab-group-epic-discussion-note-delete  
command line option, [535](#)  
gitlab-group-epic-discussion-note-get  
command line option, [535](#)  
gitlab-group-epic-discussion-note-update  
command line option, [536](#)  
gitlab-group-epic-get command line  
option, [531](#)  
gitlab-group-epic-issue-create command  
line option, [536](#)  
gitlab-group-epic-issue-delete command  
line option, [536](#)  
gitlab-group-epic-issue-list command  
line option, [537](#)  
gitlab-group-epic-issue-update command  
line option, [537](#)  
gitlab-group-epic-list command line  
option, [532](#)  
gitlab-group-epic-note-award-emoji-create  
command line option, [540](#)  
gitlab-group-epic-note-award-emoji-delete  
command line option, [540](#)  
gitlab-group-epic-note-award-emoji-get  
command line option, [540](#)  
gitlab-group-epic-note-award-emoji-list  
command line option, [541](#)  
gitlab-group-epic-note-create command  
line option, [538](#)  
gitlab-group-epic-note-delete command  
line option, [538](#)  
gitlab-group-epic-note-get command line  
option, [538](#)  
gitlab-group-epic-note-list command  
line option, [539](#)  
gitlab-group-epic-note-update command  
line option, [539](#)  
gitlab-group-epic-resource-label-event-get  
command line option, [541](#)  
gitlab-group-epic-resource-label-event-list  
command line option, [542](#)  
gitlab-group-epic-update command line  
option, [532](#)  
gitlab-group-export-create command line  
option, [542](#)  
gitlab-group-export-download command  
line option, [542](#)  
gitlab-group-export-get command line  
option, [543](#)  
gitlab-group-hook-create command line  
option, [543](#)  
gitlab-group-hook-delete command line  
option, [544](#)  
gitlab-group-hook-get command line  
option, [544](#)  
gitlab-group-hook-list command line  
option, [545](#)  
gitlab-group-hook-update command line  
option, [545](#)  
gitlab-group-import-get command line  
option, [546](#)  
gitlab-group-invitation-create command  
line option, [547](#)  
gitlab-group-invitation-delete command  
line option, [547](#)  
gitlab-group-invitation-get command  
line option, [547](#)  
gitlab-group-invitation-list command  
line option, [548](#)  
gitlab-group-invitation-update command  
line option, [548](#)  
gitlab-group-issue-list command line  
option, [549](#)  
gitlab-group-issues-statistics-get  
command line option, [550](#)  
gitlab-group-iteration-list command  
line option, [551](#)  
gitlab-group-label-create command line  
option, [551](#)  
gitlab-group-label-delete command line

option, [552](#)  
 gitlab-group-label-get command line option, [552](#)  
 gitlab-group-label-list command line option, [552](#)  
 gitlab-group-label-subscribe command line option, [553](#)  
 gitlab-group-label-unsubscribe command line option, [553](#)  
 gitlab-group-label-update command line option, [553](#)  
 gitlab-group-ldap-group-link-create command line option, [554](#)  
 gitlab-group-ldap-group-link-delete command line option, [554](#)  
 gitlab-group-ldap-group-link-list command line option, [555](#)  
 gitlab-group-member-all-get command line option, [557](#)  
 gitlab-group-member-all-list command line option, [557](#)  
 gitlab-group-member-create command line option, [555](#)  
 gitlab-group-member-delete command line option, [555](#)  
 gitlab-group-member-get command line option, [556](#)  
 gitlab-group-member-list command line option, [556](#)  
 gitlab-group-member-update command line option, [556](#)  
 gitlab-group-merge-request-list command line option, [558](#)  
 gitlab-group-milestone-create command line option, [559](#)  
 gitlab-group-milestone-delete command line option, [560](#)  
 gitlab-group-milestone-get command line option, [560](#)  
 gitlab-group-milestone-issues command line option, [560](#)  
 gitlab-group-milestone-list command line option, [561](#)  
 gitlab-group-milestone-merge-requests command line option, [561](#)  
 gitlab-group-milestone-update command line option, [561](#)  
 gitlab-group-notification-settings-get command line option, [562](#)  
 gitlab-group-notification-settings-update command line option, [562](#)  
 gitlab-group-package-list command line option, [563](#)  
 gitlab-group-project-list command line option, [564](#)  
 gitlab-group-push-rules-create command line option, [565](#)  
 gitlab-group-push-rules-delete command line option, [566](#)  
 gitlab-group-push-rules-get command line option, [566](#)  
 gitlab-group-push-rules-update command line option, [567](#)  
 gitlab-group-runner-list command line option, [568](#)  
 gitlab-group-saml-group-link-create command line option, [568](#)  
 gitlab-group-saml-group-link-delete command line option, [568](#)  
 gitlab-group-saml-group-link-get command line option, [569](#)  
 gitlab-group-saml-group-link-list command line option, [569](#)  
 gitlab-group-share command line option, [511](#)  
 gitlab-group-subgroup-list command line option, [570](#)  
 gitlab-group-transfer command line option, [511](#)  
 gitlab-group-unshare command line option, [511](#)  
 gitlab-group-variable-create command line option, [571](#)  
 gitlab-group-variable-delete command line option, [571](#)  
 gitlab-group-variable-get command line option, [571](#)  
 gitlab-group-variable-list command line option, [571](#)  
 gitlab-group-variable-update command line option, [572](#)  
 gitlab-group-wiki-create command line option, [572](#)  
 gitlab-group-wiki-delete command line option, [573](#)  
 gitlab-group-wiki-get command line option, [573](#)  
 gitlab-group-wiki-list command line option, [573](#)  
 gitlab-group-wiki-update command line option, [574](#)  
 gitlab-project-share command line option, [595](#)  
 gitlab-project-unshare command line option, [597](#)  
 gitlab-shared-project-list command line option, [790](#)  
 --group-ids



- gitlab-project-approval-rule-create
  - command line option, [607](#)
- gitlab-project-merge-request-approval-rule-create
  - command line option, [701](#)
- gitlab-project-merge-request-approval-rule-update
  - command line option, [703](#)
- group-with-project-templates-id
  - gitlab-project-create command line option, [588](#)
- header-logo
  - gitlab-application-appearance-update command line option, [483](#)
- header-message
  - gitlab-application-appearance-update command line option, [483](#)
- help
  - gitlab command line option, [480](#)
  - gitlab-application command line option, [481](#)
  - gitlab-application-appearance command line option, [482](#)
  - gitlab-application-appearance-get command line option, [482](#)
  - gitlab-application-appearance-update command line option, [483](#)
  - gitlab-application-create command line option, [481](#)
  - gitlab-application-delete command line option, [482](#)
  - gitlab-application-list command line option, [482](#)
  - gitlab-application-settings command line option, [483](#)
  - gitlab-application-settings-get command line option, [484](#)
  - gitlab-application-settings-update command line option, [485](#)
  - gitlab-application-statistics command line option, [487](#)
  - gitlab-application-statistics-get command line option, [488](#)
  - gitlab-audit-event command line option, [488](#)
  - gitlab-audit-event-get command line option, [488](#)
  - gitlab-audit-event-list command line option, [488](#)
  - gitlab-broadcast-message command line option, [489](#)
  - gitlab-broadcast-message-create command line option, [489](#)
  - gitlab-broadcast-message-delete command line option, [489](#)
  - gitlab-broadcast-message-get command line option, [490](#)
  - gitlab-broadcast-message-list command line option, [490](#)
  - gitlab-broadcast-message-update command line option, [490](#)
  - gitlab-bulk-import command line option, [491](#)
  - gitlab-bulk-import-all-entity command line option, [492](#)
  - gitlab-bulk-import-all-entity-list command line option, [492](#)
  - gitlab-bulk-import-create command line option, [491](#)
  - gitlab-bulk-import-entity command line option, [492](#)
  - gitlab-bulk-import-entity-get command line option, [492](#)
  - gitlab-bulk-import-entity-list command line option, [493](#)
  - gitlab-bulk-import-get command line option, [491](#)
  - gitlab-bulk-import-list command line option, [491](#)
  - gitlab-ci-lint command line option, [493](#)
  - gitlab-ci-lint-create command line option, [493](#)
  - gitlab-ci-lint-validate command line option, [494](#)
  - gitlab-current-user command line option, [494](#)
  - gitlab-current-user-email command line option, [494](#)
  - gitlab-current-user-email-create command line option, [494](#)
  - gitlab-current-user-email-delete command line option, [495](#)
  - gitlab-current-user-email-get command line option, [495](#)
  - gitlab-current-user-email-list command line option, [495](#)
  - gitlab-current-user-get command line option, [494](#)
  - gitlab-current-user-gpg-key command line option, [495](#)
  - gitlab-current-user-gpg-key-create command line option, [496](#)
  - gitlab-current-user-gpg-key-delete command line option, [496](#)
  - gitlab-current-user-gpg-key-get command line option, [496](#)
  - gitlab-current-user-gpg-key-list command line option, [496](#)
  - gitlab-current-user-key command line option, [497](#)



[gitlab-current-user-key-create command line option](#), 497  
[gitlab-current-user-key-delete command line option](#), 497  
[gitlab-current-user-key-get command line option](#), 497  
[gitlab-current-user-key-list command line option](#), 498  
[gitlab-current-user-runner command line option](#), 498  
[gitlab-current-user-runner-create command line option](#), 498  
[gitlab-current-user-status command line option](#), 499  
[gitlab-current-user-status-get command line option](#), 499  
[gitlab-current-user-status-update command line option](#), 499  
[gitlab-deploy-key command line option](#), 499  
[gitlab-deploy-key-list command line option](#), 500  
[gitlab-deploy-token command line option](#), 500  
[gitlab-deploy-token-list command line option](#), 500  
[gitlab-dockerfile command line option](#), 500  
[gitlab-dockerfile-get command line option](#), 501  
[gitlab-dockerfile-list command line option](#), 501  
[gitlab-event command line option](#), 501  
[gitlab-event-list command line option](#), 501  
[gitlab-feature command line option](#), 502  
[gitlab-feature-delete command line option](#), 502  
[gitlab-feature-list command line option](#), 502  
[gitlab-generic-package command line option](#), 503  
[gitlab-generic-package-download command line option](#), 503  
[gitlab-generic-package-upload command line option](#), 503  
[gitlab-geo-node command line option](#), 504  
[gitlab-geo-node-current-failures command line option](#), 504  
[gitlab-geo-node-delete command line option](#), 504  
[gitlab-geo-node-get command line option](#), 504  
[gitlab-geo-node-list command line option](#), 505  
[gitlab-geo-node-repair command line option](#), 505  
[gitlab-geo-node-status command line option](#), 505  
[gitlab-geo-node-update command line option](#), 505  
[gitlab-gitignore command line option](#), 506  
[gitlab-gitignore-get command line option](#), 506  
[gitlab-gitignore-list command line option](#), 506  
[gitlab-gitlabciyaml command line option](#), 506  
[gitlab-gitlabciyaml-get command line option](#), 507  
[gitlab-gitlabciyaml-list command line option](#), 507  
[gitlab-group command line option](#), 507  
[gitlab-group-access-request command line option](#), 513  
[gitlab-group-access-request-approve command line option](#), 513  
[gitlab-group-access-request-create command line option](#), 514  
[gitlab-group-access-request-delete command line option](#), 514  
[gitlab-group-access-request-list command line option](#), 514  
[gitlab-group-access-token command line option](#), 515  
[gitlab-group-access-token-create command line option](#), 515  
[gitlab-group-access-token-delete command line option](#), 515  
[gitlab-group-access-token-get command line option](#), 515  
[gitlab-group-access-token-list command line option](#), 516  
[gitlab-group-audit-event command line option](#), 516  
[gitlab-group-audit-event-get command line option](#), 516  
[gitlab-group-audit-event-list command line option](#), 517  
[gitlab-group-badge command line option](#), 517  
[gitlab-group-badge-create command line option](#), 517  
[gitlab-group-badge-delete command line option](#), 518  
[gitlab-group-badge-get command line option](#), 518  
[gitlab-group-badge-list command line option](#), 518

- option, [518](#)
- gitlab-group-badge-render command line option, [519](#)
- gitlab-group-badge-update command line option, [519](#)
- gitlab-group-billable-member command line option, [519](#)
- gitlab-group-billable-member-delete command line option, [519](#)
- gitlab-group-billable-member-list command line option, [520](#)
- gitlab-group-billable-member-membership command line option, [520](#)
- gitlab-group-billable-member-membership-list command line option, [520](#)
- gitlab-group-board command line option, [521](#)
- gitlab-group-board-create command line option, [521](#)
- gitlab-group-board-delete command line option, [521](#)
- gitlab-group-board-get command line option, [522](#)
- gitlab-group-board-list command line option, [522](#), [523](#)
- gitlab-group-board-list-create command line option, [523](#)
- gitlab-group-board-list-delete command line option, [523](#)
- gitlab-group-board-list-get command line option, [524](#)
- gitlab-group-board-list-list command line option, [524](#)
- gitlab-group-board-list-update command line option, [524](#)
- gitlab-group-board-update command line option, [522](#)
- gitlab-group-cluster command line option, [525](#)
- gitlab-group-cluster-create command line option, [525](#)
- gitlab-group-cluster-delete command line option, [525](#)
- gitlab-group-cluster-get command line option, [526](#)
- gitlab-group-cluster-list command line option, [526](#)
- gitlab-group-cluster-update command line option, [526](#)
- gitlab-group-create command line option, [508](#)
- gitlab-group-custom-attribute command line option, [527](#)
- gitlab-group-custom-attribute-delete command line option, [527](#)
- gitlab-group-custom-attribute-get command line option, [527](#)
- gitlab-group-custom-attribute-list command line option, [527](#)
- gitlab-group-delete command line option, [509](#)
- gitlab-group-deploy-token command line option, [528](#)
- gitlab-group-deploy-token-create command line option, [528](#)
- gitlab-group-deploy-token-delete command line option, [528](#)
- gitlab-group-deploy-token-get command line option, [529](#)
- gitlab-group-deploy-token-list command line option, [529](#)
- gitlab-group-descendant-group command line option, [529](#)
- gitlab-group-descendant-group-list command line option, [530](#)
- gitlab-group-epic command line option, [530](#)
- gitlab-group-epic-award-emoji command line option, [533](#)
- gitlab-group-epic-award-emoji-create command line option, [533](#)
- gitlab-group-epic-award-emoji-delete command line option, [533](#)
- gitlab-group-epic-award-emoji-get command line option, [533](#)
- gitlab-group-epic-award-emoji-list command line option, [534](#)
- gitlab-group-epic-create command line option, [531](#)
- gitlab-group-epic-delete command line option, [531](#)
- gitlab-group-epic-discussion-note command line option, [534](#)
- gitlab-group-epic-discussion-note-create command line option, [534](#)
- gitlab-group-epic-discussion-note-delete command line option, [535](#)
- gitlab-group-epic-discussion-note-get command line option, [535](#)
- gitlab-group-epic-discussion-note-update command line option, [535](#)
- gitlab-group-epic-get command line option, [531](#)
- gitlab-group-epic-issue command line option, [536](#)
- gitlab-group-epic-issue-create command line option, [536](#)
- gitlab-group-epic-issue-delete command

- line option, [536](#)
- gitlab-group-epic-issue-list command line option, [537](#)
- gitlab-group-epic-issue-update command line option, [537](#)
- gitlab-group-epic-list command line option, [532](#)
- gitlab-group-epic-note command line option, [537](#)
- gitlab-group-epic-note-award-emoji command line option, [539](#)
- gitlab-group-epic-note-award-emoji-create command line option, [540](#)
- gitlab-group-epic-note-award-emoji-delete command line option, [540](#)
- gitlab-group-epic-note-award-emoji-get command line option, [540](#)
- gitlab-group-epic-note-award-emoji-list command line option, [541](#)
- gitlab-group-epic-note-create command line option, [538](#)
- gitlab-group-epic-note-delete command line option, [538](#)
- gitlab-group-epic-note-get command line option, [538](#)
- gitlab-group-epic-note-list command line option, [539](#)
- gitlab-group-epic-note-update command line option, [539](#)
- gitlab-group-epic-resource-label-event command line option, [541](#)
- gitlab-group-epic-resource-label-event-get command line option, [541](#)
- gitlab-group-epic-resource-label-event-list command line option, [542](#)
- gitlab-group-epic-update command line option, [532](#)
- gitlab-group-export command line option, [542](#)
- gitlab-group-export-create command line option, [542](#)
- gitlab-group-export-download command line option, [542](#)
- gitlab-group-export-get command line option, [543](#)
- gitlab-group-get command line option, [509](#)
- gitlab-group-hook command line option, [543](#)
- gitlab-group-hook-create command line option, [543](#)
- gitlab-group-hook-delete command line option, [544](#)
- gitlab-group-hook-get command line option, [544](#)
- gitlab-group-hook-list command line option, [545](#)
- gitlab-group-hook-update command line option, [545](#)
- gitlab-group-import command line option, [546](#)
- gitlab-group-import-get command line option, [546](#)
- gitlab-group-invitation command line option, [546](#)
- gitlab-group-invitation-create command line option, [547](#)
- gitlab-group-invitation-delete command line option, [547](#)
- gitlab-group-invitation-get command line option, [547](#)
- gitlab-group-invitation-list command line option, [548](#)
- gitlab-group-invitation-update command line option, [548](#)
- gitlab-group-issue command line option, [549](#)
- gitlab-group-issue-list command line option, [549](#)
- gitlab-group-issues-statistics command line option, [550](#)
- gitlab-group-issues-statistics-get command line option, [550](#)
- gitlab-group-iteration command line option, [550](#)
- gitlab-group-iteration-list command line option, [550](#)
- gitlab-group-label command line option, [551](#)
- gitlab-group-label-create command line option, [551](#)
- gitlab-group-label-delete command line option, [552](#)
- gitlab-group-label-get command line option, [552](#)
- gitlab-group-label-list command line option, [552](#)
- gitlab-group-label-subscribe command line option, [553](#)
- gitlab-group-label-unsubscribe command line option, [553](#)
- gitlab-group-label-update command line option, [553](#)
- gitlab-group-ldap-group-link command line option, [554](#)
- gitlab-group-ldap-group-link-create command line option, [554](#)
- gitlab-group-ldap-group-link-delete command line option, [554](#)

`gitlab-group-ldap-group-link-list`  
command line option, [554](#)

`gitlab-group-ldap-sync` command line  
option, [509](#)

`gitlab-group-list` command line option,  
[510](#)

`gitlab-group-member` command line option,  
[555](#)

`gitlab-group-member-all` command line  
option, [557](#)

`gitlab-group-member-all-get` command  
line option, [557](#)

`gitlab-group-member-all-list` command  
line option, [557](#)

`gitlab-group-member-create` command line  
option, [555](#)

`gitlab-group-member-delete` command line  
option, [555](#)

`gitlab-group-member-get` command line  
option, [556](#)

`gitlab-group-member-list` command line  
option, [556](#)

`gitlab-group-member-update` command line  
option, [556](#)

`gitlab-group-merge-request` command line  
option, [557](#)

`gitlab-group-merge-request-list` command  
line option, [558](#)

`gitlab-group-milestone` command line  
option, [559](#)

`gitlab-group-milestone-create` command  
line option, [559](#)

`gitlab-group-milestone-delete` command  
line option, [560](#)

`gitlab-group-milestone-get` command line  
option, [560](#)

`gitlab-group-milestone-issues` command  
line option, [560](#)

`gitlab-group-milestone-list` command  
line option, [560](#)

`gitlab-group-milestone-merge-requests`  
command line option, [561](#)

`gitlab-group-milestone-update` command  
line option, [561](#)

`gitlab-group-notification-settings`  
command line option, [562](#)

`gitlab-group-notification-settings-get`  
command line option, [562](#)

`gitlab-group-notification-settings-update`  
command line option, [562](#)

`gitlab-group-package` command line  
option, [563](#)

`gitlab-group-package-list` command line  
option, [563](#)

`gitlab-group-project` command line  
option, [564](#)

`gitlab-group-project-list` command line  
option, [564](#)

`gitlab-group-push-rules` command line  
option, [565](#)

`gitlab-group-push-rules-create` command  
line option, [565](#)

`gitlab-group-push-rules-delete` command  
line option, [566](#)

`gitlab-group-push-rules-get` command  
line option, [566](#)

`gitlab-group-push-rules-update` command  
line option, [567](#)

`gitlab-group-restore` command line  
option, [510](#)

`gitlab-group-runner` command line option,  
[567](#)

`gitlab-group-runner-list` command line  
option, [567](#)

`gitlab-group-saml-group-link` command  
line option, [568](#)

`gitlab-group-saml-group-link-create`  
command line option, [568](#)

`gitlab-group-saml-group-link-delete`  
command line option, [568](#)

`gitlab-group-saml-group-link-get`  
command line option, [569](#)

`gitlab-group-saml-group-link-list`  
command line option, [569](#)

`gitlab-group-search` command line option,  
[510](#)

`gitlab-group-share` command line option,  
[511](#)

`gitlab-group-subgroup` command line  
option, [569](#)

`gitlab-group-subgroup-list` command line  
option, [570](#)

`gitlab-group-transfer` command line  
option, [511](#)

`gitlab-group-transfer-project` command  
line option, [511](#)

`gitlab-group-unshare` command line  
option, [511](#)

`gitlab-group-update` command line option,  
[512](#)

`gitlab-group-variable` command line  
option, [570](#)

`gitlab-group-variable-create` command  
line option, [570](#)

`gitlab-group-variable-delete` command  
line option, [571](#)

`gitlab-group-variable-get` command line  
option, [571](#)

gitlab-group-variable-list command line option, [571](#)  
 gitlab-group-variable-update command line option, [572](#)  
 gitlab-group-wiki command line option, [572](#)  
 gitlab-group-wiki-create command line option, [572](#)  
 gitlab-group-wiki-delete command line option, [573](#)  
 gitlab-group-wiki-get command line option, [573](#)  
 gitlab-group-wiki-list command line option, [573](#)  
 gitlab-group-wiki-update command line option, [574](#)  
 gitlab-hook command line option, [574](#)  
 gitlab-hook-create command line option, [574](#)  
 gitlab-hook-delete command line option, [574](#)  
 gitlab-hook-get command line option, [575](#)  
 gitlab-hook-list command line option, [575](#)  
 gitlab-issue command line option, [575](#)  
 gitlab-issue-get command line option, [575](#)  
 gitlab-issue-list command line option, [576](#)  
 gitlab-issues-statistics command line option, [577](#)  
 gitlab-issues-statistics-get command line option, [577](#)  
 gitlab-key command line option, [577](#)  
 gitlab-key-get command line option, [577](#)  
 gitlab-ldap-group command line option, [577](#)  
 gitlab-ldap-group-list command line option, [578](#)  
 gitlab-license command line option, [578](#)  
 gitlab-license-get command line option, [578](#)  
 gitlab-license-list command line option, [579](#)  
 gitlab-merge-request command line option, [579](#)  
 gitlab-merge-request-list command line option, [580](#)  
 gitlab-namespace command line option, [581](#)  
 gitlab-namespace-exists command line option, [581](#)  
 gitlab-namespace-get command line option, [581](#)  
 gitlab-namespace-list command line option, [582](#)  
 gitlab-notification-settings command line option, [582](#)  
 gitlab-notification-settings-get command line option, [582](#)  
 gitlab-notification-settings-update command line option, [582](#)  
 gitlab-pages-domain command line option, [583](#)  
 gitlab-pages-domain-list command line option, [583](#)  
 gitlab-personal-access-token command line option, [584](#)  
 gitlab-personal-access-token-delete command line option, [584](#)  
 gitlab-personal-access-token-get command line option, [584](#)  
 gitlab-personal-access-token-list command line option, [584](#)  
 gitlab-project command line option, [585](#)  
 gitlab-project-access-request command line option, [602](#)  
 gitlab-project-access-request-approve command line option, [603](#)  
 gitlab-project-access-request-create command line option, [603](#)  
 gitlab-project-access-request-delete command line option, [603](#)  
 gitlab-project-access-request-list command line option, [603](#)  
 gitlab-project-access-token command line option, [604](#)  
 gitlab-project-access-token-create command line option, [604](#)  
 gitlab-project-access-token-delete command line option, [604](#)  
 gitlab-project-access-token-get command line option, [605](#)  
 gitlab-project-access-token-list command line option, [605](#)  
 gitlab-project-additional-statistics command line option, [605](#)  
 gitlab-project-additional-statistics-get command line option, [605](#)  
 gitlab-project-approval command line option, [606](#)  
 gitlab-project-approval-get command line option, [606](#)  
 gitlab-project-approval-rule command line option, [607](#)  
 gitlab-project-approval-rule-create command line option, [607](#)  
 gitlab-project-approval-rule-delete command line option, [607](#)  
 gitlab-project-approval-rule-list command line option, [608](#)

`gitlab-project-approval-rule-update` command line option, [608](#)  
`gitlab-project-approval-update` command line option, [606](#)  
`gitlab-project-archive` command line option, [585](#)  
`gitlab-project-artifact` command line option, [608](#)  
`gitlab-project-artifact-delete` command line option, [608](#)  
`gitlab-project-artifact-download` command line option, [609](#)  
`gitlab-project-artifact-raw` command line option, [609](#)  
`gitlab-project-audit-event` command line option, [609](#)  
`gitlab-project-audit-event-get` command line option, [610](#)  
`gitlab-project-audit-event-list` command line option, [610](#)  
`gitlab-project-badge` command line option, [610](#)  
`gitlab-project-badge-create` command line option, [611](#)  
`gitlab-project-badge-delete` command line option, [611](#)  
`gitlab-project-badge-get` command line option, [611](#)  
`gitlab-project-badge-list` command line option, [612](#)  
`gitlab-project-badge-render` command line option, [612](#)  
`gitlab-project-badge-update` command line option, [612](#)  
`gitlab-project-board` command line option, [613](#)  
`gitlab-project-board-create` command line option, [613](#)  
`gitlab-project-board-delete` command line option, [613](#)  
`gitlab-project-board-get` command line option, [613](#)  
`gitlab-project-board-list` command line option, [614](#)  
`gitlab-project-board-list-create` command line option, [614](#)  
`gitlab-project-board-list-delete` command line option, [615](#)  
`gitlab-project-board-list-get` command line option, [615](#)  
`gitlab-project-board-list-list` command line option, [615](#)  
`gitlab-project-board-list-update` command line option, [616](#)  
`gitlab-project-board-update` command line option, [614](#)  
`gitlab-project-branch` command line option, [616](#)  
`gitlab-project-branch-create` command line option, [616](#)  
`gitlab-project-branch-delete` command line option, [617](#)  
`gitlab-project-branch-get` command line option, [617](#)  
`gitlab-project-branch-list` command line option, [617](#)  
`gitlab-project-ci-lint` command line option, [618](#)  
`gitlab-project-ci-lint-create` command line option, [618](#)  
`gitlab-project-ci-lint-get` command line option, [618](#)  
`gitlab-project-ci-lint-validate` command line option, [619](#)  
`gitlab-project-cluster` command line option, [619](#)  
`gitlab-project-cluster-create` command line option, [619](#)  
`gitlab-project-cluster-delete` command line option, [620](#)  
`gitlab-project-cluster-get` command line option, [620](#)  
`gitlab-project-cluster-list` command line option, [620](#)  
`gitlab-project-cluster-update` command line option, [621](#)  
`gitlab-project-commit` command line option, [621](#)  
`gitlab-project-commit-cherry-pick` command line option, [622](#)  
`gitlab-project-commit-comment` command line option, [625](#)  
`gitlab-project-commit-comment-create` command line option, [625](#)  
`gitlab-project-commit-comment-list` command line option, [626](#)  
`gitlab-project-commit-create` command line option, [622](#)  
`gitlab-project-commit-diff` command line option, [622](#)  
`gitlab-project-commit-discussion` command line option, [626](#)  
`gitlab-project-commit-discussion-create` command line option, [626](#)  
`gitlab-project-commit-discussion-get` command line option, [627](#)  
`gitlab-project-commit-discussion-list` command line option, [627](#)



gitlab-project-commit-discussion-note  
command line option, 627

gitlab-project-commit-discussion-note-create  
command line option, 628

gitlab-project-commit-discussion-note-delete  
command line option, 628

gitlab-project-commit-discussion-note-get  
command line option, 629

gitlab-project-commit-discussion-note-update  
command line option, 629

gitlab-project-commit-get command line  
option, 623

gitlab-project-commit-list command line  
option, 623

gitlab-project-commit-merge-requests  
command line option, 624

gitlab-project-commit-refs command line  
option, 624

gitlab-project-commit-revert command  
line option, 624

gitlab-project-commit-signature command  
line option, 625

gitlab-project-commit-status command  
line option, 629

gitlab-project-commit-status-create  
command line option, 630

gitlab-project-commit-status-list  
command line option, 630

gitlab-project-create command line  
option, 587

gitlab-project-create-fork-relation  
command line option, 590

gitlab-project-custom-attribute command  
line option, 631

gitlab-project-custom-attribute-delete  
command line option, 631

gitlab-project-custom-attribute-get  
command line option, 631

gitlab-project-custom-attribute-list  
command line option, 631

gitlab-project-delete command line  
option, 590

gitlab-project-delete-fork-relation  
command line option, 590

gitlab-project-delete-merged-branches  
command line option, 590

gitlab-project-deploy-token command  
line option, 632

gitlab-project-deploy-token-create  
command line option, 632

gitlab-project-deploy-token-delete  
command line option, 632

gitlab-project-deploy-token-get command  
line option, 632

gitlab-project-deploy-token-list  
command line option, 633

gitlab-project-deployment command line  
option, 633

gitlab-project-deployment-approval  
command line option, 633

gitlab-project-deployment-create  
command line option, 634

gitlab-project-deployment-get command  
line option, 634

gitlab-project-deployment-list command  
line option, 634

gitlab-project-deployment-merge-request  
command line option, 635

gitlab-project-deployment-merge-request-list  
command line option, 636

gitlab-project-deployment-update  
command line option, 635

gitlab-project-environment command line  
option, 638

gitlab-project-environment-create  
command line option, 638

gitlab-project-environment-delete  
command line option, 638

gitlab-project-environment-get command  
line option, 638

gitlab-project-environment-list command  
line option, 639

gitlab-project-environment-stop command  
line option, 639

gitlab-project-environment-update  
command line option, 639

gitlab-project-event command line  
option, 640

gitlab-project-event-list command line  
option, 640

gitlab-project-export command line  
option, 641

gitlab-project-export-create command  
line option, 641

gitlab-project-export-download command  
line option, 641

gitlab-project-export-get command line  
option, 641

gitlab-project-file command line option,  
642

gitlab-project-file-blame command line  
option, 642

gitlab-project-file-create command line  
option, 642

gitlab-project-file-delete command line  
option, 643

gitlab-project-file-get command line  
option, 643

gitlab-project-file-raw command line option, [643](#)  
gitlab-project-file-update command line option, [644](#)  
gitlab-project-fork command line option, [644](#)  
gitlab-project-fork-create command line option, [644](#)  
gitlab-project-fork-list command line option, [645](#)  
gitlab-project-get command line option, [590](#)  
gitlab-project-hook command line option, [646](#)  
gitlab-project-hook-create command line option, [646](#)  
gitlab-project-hook-delete command line option, [647](#)  
gitlab-project-hook-get command line option, [647](#)  
gitlab-project-hook-list command line option, [647](#)  
gitlab-project-hook-update command line option, [648](#)  
gitlab-project-housekeeping command line option, [591](#)  
gitlab-project-import command line option, [649](#)  
gitlab-project-import-get command line option, [649](#)  
gitlab-project-integration command line option, [649](#)  
gitlab-project-integration-available command line option, [649](#)  
gitlab-project-integration-delete command line option, [649](#)  
gitlab-project-integration-get command line option, [650](#)  
gitlab-project-integration-list command line option, [650](#)  
gitlab-project-integration-update command line option, [650](#)  
gitlab-project-invitation command line option, [651](#)  
gitlab-project-invitation-create command line option, [651](#)  
gitlab-project-invitation-delete command line option, [651](#)  
gitlab-project-invitation-get command line option, [652](#)  
gitlab-project-invitation-list command line option, [652](#)  
gitlab-project-invitation-update command line option, [652](#)  
gitlab-project-issue command line option, [653](#)  
gitlab-project-issue-add-spent-time command line option, [653](#)  
gitlab-project-issue-award-emoji command line option, [661](#)  
gitlab-project-issue-award-emoji-create command line option, [661](#)  
gitlab-project-issue-award-emoji-delete command line option, [661](#)  
gitlab-project-issue-award-emoji-get command line option, [662](#)  
gitlab-project-issue-award-emoji-list command line option, [662](#)  
gitlab-project-issue-closed-by command line option, [654](#)  
gitlab-project-issue-create command line option, [654](#)  
gitlab-project-issue-delete command line option, [655](#)  
gitlab-project-issue-discussion command line option, [662](#)  
gitlab-project-issue-discussion-create command line option, [663](#)  
gitlab-project-issue-discussion-get command line option, [663](#)  
gitlab-project-issue-discussion-list command line option, [663](#)  
gitlab-project-issue-discussion-note command line option, [664](#)  
gitlab-project-issue-discussion-note-create command line option, [664](#)  
gitlab-project-issue-discussion-note-delete command line option, [664](#)  
gitlab-project-issue-discussion-note-get command line option, [665](#)  
gitlab-project-issue-discussion-note-update command line option, [665](#)  
gitlab-project-issue-get command line option, [655](#)  
gitlab-project-issue-link command line option, [666](#)  
gitlab-project-issue-link-create command line option, [666](#)  
gitlab-project-issue-link-delete command line option, [666](#)  
gitlab-project-issue-link-list command line option, [667](#)  
gitlab-project-issue-list command line option, [655](#)  
gitlab-project-issue-move command line option, [656](#)  
gitlab-project-issue-note command line option, [667](#)



gitlab-project-issue-note-award-emoji command line option, <a href="#">669</a>	gitlab-project-issue-resource-weight-event command line option, <a href="#">675</a>
gitlab-project-issue-note-award-emoji-create command line option, <a href="#">669</a>	gitlab-project-issue-resource-weight-event-get command line option, <a href="#">675</a>
gitlab-project-issue-note-award-emoji-delete command line option, <a href="#">670</a>	gitlab-project-issue-resource-weight-event-list command line option, <a href="#">675</a>
gitlab-project-issue-note-award-emoji-get command line option, <a href="#">670</a>	gitlab-project-issue-subscribe command line option, <a href="#">658</a>
gitlab-project-issue-note-award-emoji-list command line option, <a href="#">670</a>	gitlab-project-issue-time-estimate command line option, <a href="#">658</a>
gitlab-project-issue-note-create command line option, <a href="#">667</a>	gitlab-project-issue-time-stats command line option, <a href="#">659</a>
gitlab-project-issue-note-delete command line option, <a href="#">668</a>	gitlab-project-issue-todo command line option, <a href="#">659</a>
gitlab-project-issue-note-get command line option, <a href="#">668</a>	gitlab-project-issue-unsubscribe command line option, <a href="#">659</a>
gitlab-project-issue-note-list command line option, <a href="#">668</a>	gitlab-project-issue-update command line option, <a href="#">660</a>
gitlab-project-issue-note-update command line option, <a href="#">669</a>	gitlab-project-issue-user-agent-detail command line option, <a href="#">660</a>
gitlab-project-issue-participants command line option, <a href="#">657</a>	gitlab-project-issues-statistics command line option, <a href="#">676</a>
gitlab-project-issue-related-merge-requests command line option, <a href="#">657</a>	gitlab-project-issues-statistics-get command line option, <a href="#">676</a>
gitlab-project-issue-reorder command line option, <a href="#">657</a>	gitlab-project-job command line option, <a href="#">676</a>
gitlab-project-issue-reset-spent-time command line option, <a href="#">658</a>	gitlab-project-job-artifact command line option, <a href="#">677</a>
gitlab-project-issue-reset-time-estimate command line option, <a href="#">658</a>	gitlab-project-job-artifacts command line option, <a href="#">677</a>
gitlab-project-issue-resource-iteration-event command line option, <a href="#">671</a>	gitlab-project-job-cancel command line option, <a href="#">677</a>
gitlab-project-issue-resource-iteration-event-get command line option, <a href="#">671</a>	gitlab-project-job-delete-artifacts command line option, <a href="#">677</a>
gitlab-project-issue-resource-iteration-event-list command line option, <a href="#">671</a>	gitlab-project-job-erase command line option, <a href="#">678</a>
gitlab-project-issue-resource-label-event command line option, <a href="#">672</a>	gitlab-project-job-get command line option, <a href="#">678</a>
gitlab-project-issue-resource-label-event-get command line option, <a href="#">672</a>	gitlab-project-job-keep-artifacts command line option, <a href="#">678</a>
gitlab-project-issue-resource-label-event-list command line option, <a href="#">672</a>	gitlab-project-job-list command line option, <a href="#">679</a>
gitlab-project-issue-resource-milestone-event command line option, <a href="#">673</a>	gitlab-project-job-play command line option, <a href="#">679</a>
gitlab-project-issue-resource-milestone-event-get command line option, <a href="#">673</a>	gitlab-project-job-retry command line option, <a href="#">679</a>
gitlab-project-issue-resource-milestone-event-list command line option, <a href="#">673</a>	gitlab-project-job-token-scope command line option, <a href="#">680</a>
gitlab-project-issue-resource-state-event command line option, <a href="#">674</a>	gitlab-project-job-token-scope-get command line option, <a href="#">680</a>
gitlab-project-issue-resource-state-event-get command line option, <a href="#">674</a>	gitlab-project-job-token-scope-update command line option, <a href="#">680</a>
gitlab-project-issue-resource-state-event-list command line option, <a href="#">674</a>	gitlab-project-job-trace command line option, <a href="#">680</a>

- gitlab-project-key command line option, [681](#)
- gitlab-project-key-create command line option, [681](#)
- gitlab-project-key-delete command line option, [681](#)
- gitlab-project-key-enable command line option, [681](#)
- gitlab-project-key-get command line option, [682](#)
- gitlab-project-key-list command line option, [682](#)
- gitlab-project-key-update command line option, [682](#)
- gitlab-project-label command line option, [683](#)
- gitlab-project-label-create command line option, [683](#)
- gitlab-project-label-delete command line option, [683](#)
- gitlab-project-label-get command line option, [684](#)
- gitlab-project-label-list command line option, [684](#)
- gitlab-project-label-subscribe command line option, [684](#)
- gitlab-project-label-unsubscribe command line option, [685](#)
- gitlab-project-label-update command line option, [685](#)
- gitlab-project-languages command line option, [591](#)
- gitlab-project-list command line option, [591](#)
- gitlab-project-member command line option, [685](#)
- gitlab-project-member-all command line option, [687](#)
- gitlab-project-member-all-get command line option, [688](#)
- gitlab-project-member-all-list command line option, [688](#)
- gitlab-project-member-create command line option, [686](#)
- gitlab-project-member-delete command line option, [686](#)
- gitlab-project-member-get command line option, [686](#)
- gitlab-project-member-list command line option, [687](#)
- gitlab-project-member-update command line option, [687](#)
- gitlab-project-merge-request command line option, [689](#)

- gitlab-project-merge-request-add-spent-time command line option, [689](#)
- gitlab-project-merge-request-approval command line option, [699](#)
- gitlab-project-merge-request-approval-get command line option, [699](#)
- gitlab-project-merge-request-approval-rule command line option, [700](#)
- gitlab-project-merge-request-approval-rule-create command line option, [700](#)
- gitlab-project-merge-request-approval-rule-delete command line option, [701](#)
- gitlab-project-merge-request-approval-rule-get command line option, [701](#)
- gitlab-project-merge-request-approval-rule-list command line option, [702](#)
- gitlab-project-merge-request-approval-rule-update command line option, [702](#)
- gitlab-project-merge-request-approval-state command line option, [703](#)
- gitlab-project-merge-request-approval-state-get command line option, [703](#)
- gitlab-project-merge-request-approval-update command line option, [700](#)
- gitlab-project-merge-request-approve command line option, [689](#)
- gitlab-project-merge-request-award-emoji command line option, [703](#)
- gitlab-project-merge-request-award-emoji-create command line option, [704](#)
- gitlab-project-merge-request-award-emoji-delete command line option, [704](#)
- gitlab-project-merge-request-award-emoji-get command line option, [704](#)
- gitlab-project-merge-request-award-emoji-list command line option, [705](#)
- gitlab-project-merge-request-cancel-merge-when-pipeline command line option, [690](#)
- gitlab-project-merge-request-changes command line option, [690](#)
- gitlab-project-merge-request-closes-issues command line option, [690](#)
- gitlab-project-merge-request-commits command line option, [691](#)
- gitlab-project-merge-request-create command line option, [691](#)
- gitlab-project-merge-request-delete command line option, [692](#)
- gitlab-project-merge-request-diff command line option, [705](#)
- gitlab-project-merge-request-diff-get command line option, [705](#)
- gitlab-project-merge-request-diff-list command line option, [706](#)

gitlab-project-merge-request-discussion command line option, 706	gitlab-project-merge-request-note-delete command line option, 713
gitlab-project-merge-request-discussion-create command line option, 706	gitlab-project-merge-request-note-get command line option, 713
gitlab-project-merge-request-discussion-get command line option, 707	gitlab-project-merge-request-note-list command line option, 714
gitlab-project-merge-request-discussion-list command line option, 707	gitlab-project-merge-request-note-update command line option, 714
gitlab-project-merge-request-discussion-note command line option, 708	gitlab-project-merge-request-participants command line option, 695
gitlab-project-merge-request-discussion-note-create command line option, 708	gitlab-project-merge-request-pipeline command line option, 716
gitlab-project-merge-request-discussion-note-delete command line option, 709	gitlab-project-merge-request-pipeline-create command line option, 716
gitlab-project-merge-request-discussion-note-get command line option, 709	gitlab-project-merge-request-pipeline-list command line option, 717
gitlab-project-merge-request-discussion-note-update command line option, 710	gitlab-project-merge-request-rebase command line option, 695
gitlab-project-merge-request-discussion-update command line option, 708	gitlab-project-merge-request-reset-approvals command line option, 695
gitlab-project-merge-request-draft-note command line option, 710	gitlab-project-merge-request-reset-spent-time command line option, 696
gitlab-project-merge-request-draft-note-create command line option, 710	gitlab-project-merge-request-reset-time-estimate command line option, 696
gitlab-project-merge-request-draft-note-delete command line option, 711	gitlab-project-merge-request-resource-label-event command line option, 717
gitlab-project-merge-request-draft-note-get command line option, 711	gitlab-project-merge-request-resource-label-event-get command line option, 717
gitlab-project-merge-request-draft-note-list command line option, 712	gitlab-project-merge-request-resource-label-event-list command line option, 718
gitlab-project-merge-request-draft-note-update command line option, 712	gitlab-project-merge-request-resource-milestone-event command line option, 718
gitlab-project-merge-request-get command line option, 692	gitlab-project-merge-request-resource-milestone-event- command line option, 718
gitlab-project-merge-request-list command line option, 693	gitlab-project-merge-request-resource-milestone-event- command line option, 719
gitlab-project-merge-request-merge command line option, 694	gitlab-project-merge-request-resource-state-event command line option, 719
gitlab-project-merge-request-merge-ref command line option, 694	gitlab-project-merge-request-resource-state-event-get command line option, 719
gitlab-project-merge-request-note command line option, 712	gitlab-project-merge-request-resource-state-event-list command line option, 720
gitlab-project-merge-request-note-award-emoji command line option, 714	gitlab-project-merge-request-reviewer-detail command line option, 720
gitlab-project-merge-request-note-award-emoji-create command line option, 715	gitlab-project-merge-request-reviewer-detail-list command line option, 720
gitlab-project-merge-request-note-award-emoji-delete command line option, 715	gitlab-project-merge-request-subscribe command line option, 696
gitlab-project-merge-request-note-award-emoji-get command line option, 715	gitlab-project-merge-request-time-estimate command line option, 697
gitlab-project-merge-request-note-award-emoji-list command line option, 716	gitlab-project-merge-request-time-stats command line option, 697
gitlab-project-merge-request-note-create command line option, 713	gitlab-project-merge-request-todo command line option, 697

`gitlab-project-merge-request-unapprove`  
command line option, [698](#)

`gitlab-project-merge-request-unsubscribe`  
command line option, [698](#)

`gitlab-project-merge-request-update`  
command line option, [698](#)

`gitlab-project-merge-train` command line  
option, [721](#)

`gitlab-project-merge-train-list` command  
line option, [721](#)

`gitlab-project-milestone` command line  
option, [721](#)

`gitlab-project-milestone-create` command  
line option, [722](#)

`gitlab-project-milestone-delete` command  
line option, [722](#)

`gitlab-project-milestone-get` command  
line option, [722](#)

`gitlab-project-milestone-issues` command  
line option, [723](#)

`gitlab-project-milestone-list` command  
line option, [723](#)

`gitlab-project-milestone-merge-requests`  
command line option, [723](#)

`gitlab-project-milestone-update` command  
line option, [724](#)

`gitlab-project-mirror-pull` command line  
option, [592](#)

`gitlab-project-mirror-pull-details`  
command line option, [593](#)

`gitlab-project-note` command line option,  
[724](#)

`gitlab-project-note-get` command line  
option, [724](#)

`gitlab-project-note-list` command line  
option, [725](#)

`gitlab-project-notification-settings`  
command line option, [725](#)

`gitlab-project-notification-settings-get`  
command line option, [725](#)

`gitlab-project-notification-settings-update`  
command line option, [726](#)

`gitlab-project-package` command line  
option, [726](#)

`gitlab-project-package-delete` command  
line option, [727](#)

`gitlab-project-package-file` command  
line option, [728](#)

`gitlab-project-package-file-delete`  
command line option, [728](#)

`gitlab-project-package-file-list`  
command line option, [728](#)

`gitlab-project-package-get` command line  
option, [727](#)

`gitlab-project-package-list` command  
line option, [727](#)

`gitlab-project-package-pipeline` command  
line option, [729](#)

`gitlab-project-package-pipeline-list`  
command line option, [729](#)

`gitlab-project-pages-domain` command  
line option, [729](#)

`gitlab-project-pages-domain-create`  
command line option, [729](#)

`gitlab-project-pages-domain-delete`  
command line option, [730](#)

`gitlab-project-pages-domain-get` command  
line option, [730](#)

`gitlab-project-pages-domain-list`  
command line option, [730](#)

`gitlab-project-pages-domain-update`  
command line option, [731](#)

`gitlab-project-pipeline` command line  
option, [731](#)

`gitlab-project-pipeline-bridge` command  
line option, [733](#)

`gitlab-project-pipeline-bridge-list`  
command line option, [733](#)

`gitlab-project-pipeline-cancel` command  
line option, [731](#)

`gitlab-project-pipeline-create` command  
line option, [731](#)

`gitlab-project-pipeline-delete` command  
line option, [732](#)

`gitlab-project-pipeline-get` command  
line option, [732](#)

`gitlab-project-pipeline-job` command  
line option, [734](#)

`gitlab-project-pipeline-job-list`  
command line option, [734](#)

`gitlab-project-pipeline-list` command  
line option, [732](#)

`gitlab-project-pipeline-retry` command  
line option, [733](#)

`gitlab-project-pipeline-schedule`  
command line option, [735](#)

`gitlab-project-pipeline-schedule-create`  
command line option, [735](#)

`gitlab-project-pipeline-schedule-delete`  
command line option, [735](#)

`gitlab-project-pipeline-schedule-get`  
command line option, [736](#)

`gitlab-project-pipeline-schedule-list`  
command line option, [736](#)

`gitlab-project-pipeline-schedule-pipeline`  
command line option, [737](#)

`gitlab-project-pipeline-schedule-pipeline-list`  
command line option, [738](#)

- gitlab-project-pipeline-schedule-play  
command line option, [736](#)
- gitlab-project-pipeline-schedule-take-ownership  
command line option, [737](#)
- gitlab-project-pipeline-schedule-update  
command line option, [737](#)
- gitlab-project-pipeline-schedule-variable  
command line option, [738](#)
- gitlab-project-pipeline-schedule-variable-create  
command line option, [738](#)
- gitlab-project-pipeline-schedule-variable-delete  
command line option, [739](#)
- gitlab-project-pipeline-schedule-variable-update  
command line option, [739](#)
- gitlab-project-pipeline-test-report  
command line option, [740](#)
- gitlab-project-pipeline-test-report-get  
command line option, [740](#)
- gitlab-project-pipeline-test-report-summary  
command line option, [740](#)
- gitlab-project-pipeline-test-report-summary-get  
command line option, [740](#)
- gitlab-project-pipeline-variable  
command line option, [741](#)
- gitlab-project-pipeline-variable-list  
command line option, [741](#)
- gitlab-project-protected-branch command  
line option, [741](#)
- gitlab-project-protected-branch-create  
command line option, [742](#)
- gitlab-project-protected-branch-delete  
command line option, [742](#)
- gitlab-project-protected-branch-get  
command line option, [743](#)
- gitlab-project-protected-branch-list  
command line option, [743](#)
- gitlab-project-protected-branch-update  
command line option, [743](#)
- gitlab-project-protected-environment  
command line option, [744](#)
- gitlab-project-protected-environment-create  
command line option, [744](#)
- gitlab-project-protected-environment-delete  
command line option, [744](#)
- gitlab-project-protected-environment-get  
command line option, [745](#)
- gitlab-project-protected-environment-list  
command line option, [745](#)
- gitlab-project-protected-tag command  
line option, [745](#)
- gitlab-project-protected-tag-create  
command line option, [746](#)
- gitlab-project-protected-tag-delete  
command line option, [746](#)
- gitlab-project-protected-tag-get  
command line option, [746](#)
- gitlab-project-protected-tag-list  
command line option, [747](#)
- gitlab-project-push-rules command line  
option, [747](#)
- gitlab-project-push-rules-create  
command line option, [747](#)
- gitlab-project-push-rules-delete  
command line option, [748](#)
- gitlab-project-push-rules-get command  
line option, [748](#)
- gitlab-project-push-rules-update  
command line option, [749](#)
- gitlab-project-registry-repository  
command line option, [749](#)
- gitlab-project-registry-repository-delete  
command line option, [750](#)
- gitlab-project-registry-repository-list  
command line option, [750](#)
- gitlab-project-registry-tag command  
line option, [750](#)
- gitlab-project-registry-tag-delete  
command line option, [750](#)
- gitlab-project-registry-tag-delete-in-bulk  
command line option, [751](#)
- gitlab-project-registry-tag-get command  
line option, [751](#)
- gitlab-project-registry-tag-list  
command line option, [752](#)
- gitlab-project-release command line  
option, [752](#)
- gitlab-project-release-create command  
line option, [752](#)
- gitlab-project-release-delete command  
line option, [753](#)
- gitlab-project-release-get command line  
option, [753](#)
- gitlab-project-release-link command  
line option, [754](#)
- gitlab-project-release-link-create  
command line option, [754](#)
- gitlab-project-release-link-delete  
command line option, [755](#)
- gitlab-project-release-link-get command  
line option, [755](#)
- gitlab-project-release-link-list  
command line option, [755](#)
- gitlab-project-release-link-update  
command line option, [756](#)
- gitlab-project-release-list command  
line option, [753](#)
- gitlab-project-release-update command  
line option, [754](#)

`gitlab-project-remote-mirror` command line option, [756](#)  
`gitlab-project-remote-mirror-create` command line option, [757](#)  
`gitlab-project-remote-mirror-delete` command line option, [757](#)  
`gitlab-project-remote-mirror-list` command line option, [757](#)  
`gitlab-project-remote-mirror-update` command line option, [758](#)  
`gitlab-project-repository-archive` command line option, [593](#)  
`gitlab-project-repository-blob` command line option, [593](#)  
`gitlab-project-repository-compare` command line option, [593](#)  
`gitlab-project-repository-contributors` command line option, [594](#)  
`gitlab-project-repository-merge-base` command line option, [594](#)  
`gitlab-project-repository-raw-blob` command line option, [594](#)  
`gitlab-project-repository-tree` command line option, [594](#)  
`gitlab-project-resource-group` command line option, [758](#)  
`gitlab-project-resource-group-get` command line option, [758](#)  
`gitlab-project-resource-group-list` command line option, [758](#)  
`gitlab-project-resource-group-upcoming-job` command line option, [759](#)  
`gitlab-project-resource-group-upcoming-job-list` command line option, [759](#)  
`gitlab-project-resource-group-update` command line option, [759](#)  
`gitlab-project-restore` command line option, [595](#)  
`gitlab-project-runner` command line option, [760](#)  
`gitlab-project-runner-create` command line option, [760](#)  
`gitlab-project-runner-delete` command line option, [760](#)  
`gitlab-project-runner-list` command line option, [761](#)  
`gitlab-project-search` command line option, [595](#)  
`gitlab-project-secure-file` command line option, [761](#)  
`gitlab-project-secure-file-create` command line option, [761](#)  
`gitlab-project-secure-file-delete` command line option, [762](#)  
`gitlab-project-secure-file-download` command line option, [762](#)  
`gitlab-project-secure-file-get` command line option, [762](#)  
`gitlab-project-secure-file-list` command line option, [762](#)  
`gitlab-project-service` command line option, [763](#)  
`gitlab-project-service-available` command line option, [763](#)  
`gitlab-project-service-delete` command line option, [763](#)  
`gitlab-project-service-get` command line option, [764](#)  
`gitlab-project-service-list` command line option, [764](#)  
`gitlab-project-service-update` command line option, [764](#)  
`gitlab-project-share` command line option, [595](#)  
`gitlab-project-snapshot` command line option, [595](#)  
`gitlab-project-snippet` command line option, [765](#)  
`gitlab-project-snippet-award-emoji` command line option, [767](#)  
`gitlab-project-snippet-award-emoji-create` command line option, [768](#)  
`gitlab-project-snippet-award-emoji-delete` command line option, [768](#)  
`gitlab-project-snippet-award-emoji-get` command line option, [768](#)  
`gitlab-project-snippet-award-emoji-list` command line option, [769](#)  
`gitlab-project-snippet-content` command line option, [765](#)  
`gitlab-project-snippet-create` command line option, [765](#)  
`gitlab-project-snippet-delete` command line option, [766](#)  
`gitlab-project-snippet-discussion` command line option, [769](#)  
`gitlab-project-snippet-discussion-create` command line option, [769](#)  
`gitlab-project-snippet-discussion-get` command line option, [770](#)  
`gitlab-project-snippet-discussion-list` command line option, [770](#)  
`gitlab-project-snippet-discussion-note` command line option, [770](#)  
`gitlab-project-snippet-discussion-note-create` command line option, [771](#)  
`gitlab-project-snippet-discussion-note-delete` command line option, [771](#)



gitlab-project-snippet-discussion-note-get command line option, <a href="#">771</a>	gitlab-project-trigger-create command line option, <a href="#">778</a>
gitlab-project-snippet-discussion-note-update command line option, <a href="#">772</a>	gitlab-project-trigger-delete command line option, <a href="#">778</a>
gitlab-project-snippet-get command line option, <a href="#">766</a>	gitlab-project-trigger-get command line option, <a href="#">779</a>
gitlab-project-snippet-list command line option, <a href="#">766</a>	gitlab-project-trigger-list command line option, <a href="#">779</a>
gitlab-project-snippet-note command line option, <a href="#">772</a>	gitlab-project-trigger-pipeline command line option, <a href="#">596</a>
gitlab-project-snippet-note-award-emoji command line option, <a href="#">774</a>	gitlab-project-trigger-update command line option, <a href="#">779</a>
gitlab-project-snippet-note-award-emoji-create command line option, <a href="#">774</a>	gitlab-project-unarchive command line option, <a href="#">596</a>
gitlab-project-snippet-note-award-emoji-delete command line option, <a href="#">775</a>	gitlab-project-unshare command line option, <a href="#">597</a>
gitlab-project-snippet-note-award-emoji-get command line option, <a href="#">775</a>	gitlab-project-unstar command line option, <a href="#">597</a>
gitlab-project-snippet-note-award-emoji-list command line option, <a href="#">776</a>	gitlab-project-update command line option, <a href="#">599</a>
gitlab-project-snippet-note-create command line option, <a href="#">772</a>	gitlab-project-update-submodule command line option, <a href="#">602</a>
gitlab-project-snippet-note-delete command line option, <a href="#">773</a>	gitlab-project-upload command line option, <a href="#">602</a>
gitlab-project-snippet-note-get command line option, <a href="#">773</a>	gitlab-project-user command line option, <a href="#">780</a>
gitlab-project-snippet-note-list command line option, <a href="#">773</a>	gitlab-project-user-list command line option, <a href="#">780</a>
gitlab-project-snippet-note-update command line option, <a href="#">774</a>	gitlab-project-variable command line option, <a href="#">780</a>
gitlab-project-snippet-update command line option, <a href="#">767</a>	gitlab-project-variable-create command line option, <a href="#">780</a>
gitlab-project-snippet-user-agent-detail command line option, <a href="#">767</a>	gitlab-project-variable-delete command line option, <a href="#">781</a>
gitlab-project-star command line option, <a href="#">596</a>	gitlab-project-variable-get command line option, <a href="#">781</a>
gitlab-project-storage command line option, <a href="#">776</a>	gitlab-project-variable-list command line option, <a href="#">781</a>
gitlab-project-storage-get command line option, <a href="#">776</a>	gitlab-project-variable-update command line option, <a href="#">782</a>
gitlab-project-tag command line option, <a href="#">777</a>	gitlab-project-wiki command line option, <a href="#">782</a>
gitlab-project-tag-create command line option, <a href="#">777</a>	gitlab-project-wiki-create command line option, <a href="#">783</a>
gitlab-project-tag-delete command line option, <a href="#">777</a>	gitlab-project-wiki-delete command line option, <a href="#">783</a>
gitlab-project-tag-get command line option, <a href="#">777</a>	gitlab-project-wiki-get command line option, <a href="#">783</a>
gitlab-project-tag-list command line option, <a href="#">778</a>	gitlab-project-wiki-list command line option, <a href="#">784</a>
gitlab-project-transfer command line option, <a href="#">596</a>	gitlab-project-wiki-update command line option, <a href="#">784</a>
gitlab-project-trigger command line option, <a href="#">778</a>	gitlab-project-wiki-upload command line option, <a href="#">784</a>

gitlab-registry-repository command line option, [785](#)  
gitlab-registry-repository-get command line option, [785](#)  
gitlab-runner command line option, [785](#)  
gitlab-runner-all command line option, [785](#), [788](#)  
gitlab-runner-all-list command line option, [788](#)  
gitlab-runner-create command line option, [786](#)  
gitlab-runner-delete command line option, [786](#)  
gitlab-runner-get command line option, [786](#)  
gitlab-runner-job command line option, [789](#)  
gitlab-runner-job-list command line option, [789](#)  
gitlab-runner-list command line option, [787](#)  
gitlab-runner-update command line option, [787](#)  
gitlab-runner-verify command line option, [788](#)  
gitlab-shared-project command line option, [789](#)  
gitlab-shared-project-list command line option, [789](#)  
gitlab-snippet command line option, [790](#)  
gitlab-snippet-content command line option, [791](#)  
gitlab-snippet-create command line option, [791](#)  
gitlab-snippet-delete command line option, [791](#)  
gitlab-snippet-get command line option, [792](#)  
gitlab-snippet-list command line option, [792](#)  
gitlab-snippet-public command line option, [792](#)  
gitlab-snippet-update command line option, [792](#)  
gitlab-snippet-user-agent-detail command line option, [793](#)  
gitlab-starred-project command line option, [793](#)  
gitlab-starred-project-list command line option, [793](#)  
gitlab-todo command line option, [794](#)  
gitlab-todo-delete command line option, [794](#)  
gitlab-todo-list command line option, [795](#)  
gitlab-todo-mark-all-as-done command line option, [795](#)  
gitlab-todo-mark-as-done command line option, [795](#)  
gitlab-topic command line option, [796](#)  
gitlab-topic-create command line option, [796](#)  
gitlab-topic-delete command line option, [796](#)  
gitlab-topic-get command line option, [796](#)  
gitlab-topic-list command line option, [797](#)  
gitlab-topic-merge command line option, [797](#)  
gitlab-topic-update command line option, [797](#)  
gitlab-user command line option, [798](#)  
gitlab-user-activate command line option, [798](#)  
gitlab-user-activities command line option, [804](#)  
gitlab-user-activities-list command line option, [804](#)  
gitlab-user-approve command line option, [798](#)  
gitlab-user-ban command line option, [798](#)  
gitlab-user-block command line option, [799](#)  
gitlab-user-create command line option, [799](#)  
gitlab-user-custom-attribute command line option, [804](#)  
gitlab-user-custom-attribute-delete command line option, [805](#)  
gitlab-user-custom-attribute-get command line option, [805](#)  
gitlab-user-custom-attribute-list command line option, [805](#)  
gitlab-user-deactivate command line option, [800](#)  
gitlab-user-delete command line option, [800](#)  
gitlab-user-email command line option, [806](#)  
gitlab-user-email-create command line option, [806](#)  
gitlab-user-email-delete command line option, [806](#)  
gitlab-user-email-get command line option, [806](#)  
gitlab-user-email-list command line option, [807](#)  
gitlab-user-event command line option, [807](#)



gitlab-user-event-list command line option, [807](#)  
 gitlab-user-follow command line option, [801](#)  
 gitlab-user-get command line option, [801](#)  
 gitlab-user-gpg-key command line option, [808](#)  
 gitlab-user-gpg-key-create command line option, [808](#)  
 gitlab-user-gpg-key-delete command line option, [808](#)  
 gitlab-user-gpg-key-get command line option, [808](#)  
 gitlab-user-gpg-key-list command line option, [809](#)  
 gitlab-user-impersonation-token command line option, [809](#)  
 gitlab-user-impersonation-token-create command line option, [809](#)  
 gitlab-user-impersonation-token-delete command line option, [810](#)  
 gitlab-user-impersonation-token-get command line option, [810](#)  
 gitlab-user-impersonation-token-list command line option, [810](#)  
 gitlab-user-key command line option, [811](#)  
 gitlab-user-key-create command line option, [811](#)  
 gitlab-user-key-delete command line option, [811](#)  
 gitlab-user-key-get command line option, [811](#)  
 gitlab-user-key-list command line option, [812](#)  
 gitlab-user-list command line option, [801](#)  
 gitlab-user-membership command line option, [812](#)  
 gitlab-user-membership-get command line option, [812](#)  
 gitlab-user-membership-list command line option, [812](#)  
 gitlab-user-personal-access-token command line option, [813](#)  
 gitlab-user-personal-access-token-create command line option, [813](#)  
 gitlab-user-project command line option, [813](#)  
 gitlab-user-project-create command line option, [814](#)  
 gitlab-user-project-list command line option, [815](#)  
 gitlab-user-reject command line option, [802](#)  
 gitlab-user-status command line option, [816](#)  
 gitlab-user-status-get command line option, [816](#)  
 gitlab-user-unban command line option, [802](#)  
 gitlab-user-unblock command line option, [802](#)  
 gitlab-user-unfollow command line option, [802](#)  
 gitlab-user-update command line option, [803](#)  
 gitlab-variable command line option, [816](#)  
 gitlab-variable-create command line option, [816](#)  
 gitlab-variable-delete command line option, [817](#)  
 gitlab-variable-get command line option, [817](#)  
 gitlab-variable-list command line option, [817](#)  
 gitlab-variable-update command line option, [818](#)  
 --home-page-url  
     gitlab-application-settings-update command line option, [486](#)  
 --id  
     gitlab-application-delete command line option, [482](#)  
     gitlab-application-settings-update command line option, [485](#)  
     gitlab-audit-event-get command line option, [488](#)  
     gitlab-broadcast-message-delete command line option, [489](#)  
     gitlab-broadcast-message-get command line option, [490](#)  
     gitlab-broadcast-message-update command line option, [490](#)  
     gitlab-bulk-import-entity-get command line option, [492](#)  
     gitlab-bulk-import-get command line option, [491](#)  
     gitlab-current-user-email-delete command line option, [495](#)  
     gitlab-current-user-email-get command line option, [495](#)  
     gitlab-current-user-gpg-key-delete command line option, [496](#)  
     gitlab-current-user-gpg-key-get command line option, [496](#)  
     gitlab-current-user-key-delete command line option, [497](#)  
     gitlab-current-user-key-get command line option, [497](#)

`gitlab-geo-node-current-failures`  
command line option, [504](#)

`gitlab-geo-node-delete` command line  
option, [504](#)

`gitlab-geo-node-get` command line option,  
[504](#)

`gitlab-geo-node-repair` command line  
option, [505](#)

`gitlab-geo-node-status` command line  
option, [505](#)

`gitlab-geo-node-update` command line  
option, [505](#)

`gitlab-group-access-request-approve`  
command line option, [513](#)

`gitlab-group-access-request-delete`  
command line option, [514](#)

`gitlab-group-access-token-delete`  
command line option, [515](#)

`gitlab-group-access-token-get` command  
line option, [516](#)

`gitlab-group-audit-event-get` command  
line option, [516](#)

`gitlab-group-badge-delete` command line  
option, [518](#)

`gitlab-group-badge-get` command line  
option, [518](#)

`gitlab-group-badge-render` command line  
option, [519](#)

`gitlab-group-badge-update` command line  
option, [519](#)

`gitlab-group-billable-member-delete`  
command line option, [520](#)

`gitlab-group-board-delete` command line  
option, [521](#)

`gitlab-group-board-get` command line  
option, [522](#)

`gitlab-group-board-list-delete` command  
line option, [523](#)

`gitlab-group-board-list-get` command  
line option, [524](#)

`gitlab-group-board-list-update` command  
line option, [524](#)

`gitlab-group-board-update` command line  
option, [522](#)

`gitlab-group-cluster-delete` command  
line option, [525](#)

`gitlab-group-cluster-get` command line  
option, [526](#)

`gitlab-group-cluster-update` command  
line option, [526](#)

`gitlab-group-delete` command line option,  
[509](#)

`gitlab-group-deploy-token-delete`  
command line option, [528](#)

`gitlab-group-deploy-token-get` command  
line option, [529](#)

`gitlab-group-epic-award-emoji-delete`  
command line option, [533](#)

`gitlab-group-epic-award-emoji-get`  
command line option, [534](#)

`gitlab-group-epic-discussion-note-delete`  
command line option, [535](#)

`gitlab-group-epic-discussion-note-get`  
command line option, [535](#)

`gitlab-group-epic-discussion-note-update`  
command line option, [536](#)

`gitlab-group-epic-note-award-emoji-delete`  
command line option, [540](#)

`gitlab-group-epic-note-award-emoji-get`  
command line option, [540](#)

`gitlab-group-epic-note-delete` command  
line option, [538](#)

`gitlab-group-epic-note-get` command line  
option, [538](#)

`gitlab-group-epic-note-update` command  
line option, [539](#)

`gitlab-group-epic-resource-label-event-get`  
command line option, [541](#)

`gitlab-group-get` command line option, [509](#)

`gitlab-group-hook-delete` command line  
option, [544](#)

`gitlab-group-hook-get` command line  
option, [544](#)

`gitlab-group-hook-update` command line  
option, [545](#)

`gitlab-group-ldap-group-link-delete`  
command line option, [554](#)

`gitlab-group-ldap-sync` command line  
option, [509](#)

`gitlab-group-member-all-get` command  
line option, [557](#)

`gitlab-group-member-delete` command line  
option, [555](#)

`gitlab-group-member-get` command line  
option, [556](#)

`gitlab-group-member-update` command line  
option, [556](#)

`gitlab-group-milestone-delete` command  
line option, [560](#)

`gitlab-group-milestone-get` command line  
option, [560](#)

`gitlab-group-milestone-issues` command  
line option, [560](#)

`gitlab-group-milestone-merge-requests`  
command line option, [561](#)

`gitlab-group-milestone-update` command  
line option, [561](#)

`gitlab-group-restore` command line

- option, [510](#)
- gitlab-group-search command line option, [510](#)
- gitlab-group-share command line option, [511](#)
- gitlab-group-transfer command line option, [511](#)
- gitlab-group-transfer-project command line option, [511](#)
- gitlab-group-unshare command line option, [511](#)
- gitlab-group-update command line option, [512](#)
- gitlab-hook-delete command line option, [574](#)
- gitlab-hook-get command line option, [575](#)
- gitlab-issue-get command line option, [575](#)
- gitlab-key-get command line option, [577](#)
- gitlab-namespace-exists command line option, [581](#)
- gitlab-namespace-get command line option, [581](#)
- gitlab-personal-access-token-delete command line option, [584](#)
- gitlab-personal-access-token-get command line option, [584](#)
- gitlab-project-access-request-approve command line option, [603](#)
- gitlab-project-access-request-delete command line option, [603](#)
- gitlab-project-access-token-delete command line option, [604](#)
- gitlab-project-access-token-get command line option, [605](#)
- gitlab-project-approval-rule-delete command line option, [607](#)
- gitlab-project-approval-rule-update command line option, [608](#)
- gitlab-project-archive command line option, [585](#)
- gitlab-project-audit-event-get command line option, [610](#)
- gitlab-project-badge-delete command line option, [611](#)
- gitlab-project-badge-get command line option, [611](#)
- gitlab-project-badge-render command line option, [612](#)
- gitlab-project-badge-update command line option, [612](#)
- gitlab-project-board-delete command line option, [613](#)
- gitlab-project-board-get command line option, [613](#)
- gitlab-project-board-list-delete command line option, [615](#)
- gitlab-project-board-list-get command line option, [615](#)
- gitlab-project-board-list-update command line option, [616](#)
- gitlab-project-board-update command line option, [614](#)
- gitlab-project-cluster-delete command line option, [620](#)
- gitlab-project-cluster-get command line option, [620](#)
- gitlab-project-cluster-update command line option, [621](#)
- gitlab-project-commit-cherry-pick command line option, [622](#)
- gitlab-project-commit-diff command line option, [622](#)
- gitlab-project-commit-discussion-get command line option, [627](#)
- gitlab-project-commit-discussion-note-delete command line option, [628](#)
- gitlab-project-commit-discussion-note-get command line option, [629](#)
- gitlab-project-commit-discussion-note-update command line option, [629](#)
- gitlab-project-commit-get command line option, [623](#)
- gitlab-project-commit-merge-requests command line option, [624](#)
- gitlab-project-commit-refs command line option, [624](#)
- gitlab-project-commit-revert command line option, [624](#)
- gitlab-project-commit-signature command line option, [625](#)
- gitlab-project-create-fork-relation command line option, [590](#)
- gitlab-project-delete command line option, [590](#)
- gitlab-project-delete-fork-relation command line option, [590](#)
- gitlab-project-delete-merged-branches command line option, [590](#)
- gitlab-project-deploy-token-delete command line option, [632](#)
- gitlab-project-deploy-token-get command line option, [633](#)
- gitlab-project-deployment-approval command line option, [633](#)
- gitlab-project-deployment-get command line option, [634](#)
- gitlab-project-deployment-update command line option, [635](#)

gitlab-project-environment-delete command line option, <a href="#">638</a>	gitlab-project-job-cancel command line option, <a href="#">677</a>
gitlab-project-environment-get command line option, <a href="#">638</a>	gitlab-project-job-delete-artifacts command line option, <a href="#">678</a>
gitlab-project-environment-stop command line option, <a href="#">639</a>	gitlab-project-job-erase command line option, <a href="#">678</a>
gitlab-project-environment-update command line option, <a href="#">639</a>	gitlab-project-job-get command line option, <a href="#">678</a>
gitlab-project-get command line option, <a href="#">590</a>	gitlab-project-job-keep-artifacts command line option, <a href="#">678</a>
gitlab-project-hook-delete command line option, <a href="#">647</a>	gitlab-project-job-play command line option, <a href="#">679</a>
gitlab-project-hook-get command line option, <a href="#">647</a>	gitlab-project-job-retry command line option, <a href="#">679</a>
gitlab-project-hook-update command line option, <a href="#">648</a>	gitlab-project-job-trace command line option, <a href="#">680</a>
gitlab-project-housekeeping command line option, <a href="#">591</a>	gitlab-project-key-delete command line option, <a href="#">681</a>
gitlab-project-issue-award-emoji-delete command line option, <a href="#">661</a>	gitlab-project-key-enable command line option, <a href="#">681</a>
gitlab-project-issue-award-emoji-get command line option, <a href="#">662</a>	gitlab-project-key-get command line option, <a href="#">682</a>
gitlab-project-issue-discussion-get command line option, <a href="#">663</a>	gitlab-project-key-update command line option, <a href="#">682</a>
gitlab-project-issue-discussion-note-delete command line option, <a href="#">665</a>	gitlab-project-languages command line option, <a href="#">591</a>
gitlab-project-issue-discussion-note-get command line option, <a href="#">665</a>	gitlab-project-member-all-get command line option, <a href="#">688</a>
gitlab-project-issue-discussion-note-update command line option, <a href="#">665</a>	gitlab-project-member-delete command line option, <a href="#">686</a>
gitlab-project-issue-note-award-emoji-delete command line option, <a href="#">670</a>	gitlab-project-member-get command line option, <a href="#">686</a>
gitlab-project-issue-note-award-emoji-get command line option, <a href="#">670</a>	gitlab-project-member-update command line option, <a href="#">687</a>
gitlab-project-issue-note-delete command line option, <a href="#">668</a>	gitlab-project-merge-request-approval-rule-create command line option, <a href="#">701</a>
gitlab-project-issue-note-get command line option, <a href="#">668</a>	gitlab-project-merge-request-approval-rule-delete command line option, <a href="#">701</a>
gitlab-project-issue-note-update command line option, <a href="#">669</a>	gitlab-project-merge-request-approval-rule-get command line option, <a href="#">701</a>
gitlab-project-issue-resource-iteration-event-get command line option, <a href="#">671</a>	gitlab-project-merge-request-approval-rule-update command line option, <a href="#">702</a>
gitlab-project-issue-resource-label-event-get command line option, <a href="#">672</a>	gitlab-project-merge-request-approval-state-get command line option, <a href="#">703</a>
gitlab-project-issue-resource-milestone-event-get command line option, <a href="#">673</a>	gitlab-project-merge-request-award-emoji-delete command line option, <a href="#">704</a>
gitlab-project-issue-resource-state-event-get command line option, <a href="#">674</a>	gitlab-project-merge-request-award-emoji-get command line option, <a href="#">704</a>
gitlab-project-issue-resource-weight-event-get command line option, <a href="#">675</a>	gitlab-project-merge-request-diff-get command line option, <a href="#">705</a>
gitlab-project-job-artifact command line option, <a href="#">677</a>	gitlab-project-merge-request-discussion-get command line option, <a href="#">707</a>
gitlab-project-job-artifacts command line option, <a href="#">677</a>	gitlab-project-merge-request-discussion-note-delete command line option, <a href="#">709</a>

[gitlab-project-merge-request-discussion-note-get command line option](#), 709  
[gitlab-project-merge-request-discussion-note-update command line option](#), 710  
[gitlab-project-merge-request-discussion-update command line option](#), 708  
[gitlab-project-merge-request-draft-note-delete command line option](#), 711  
[gitlab-project-merge-request-draft-note-get command line option](#), 711  
[gitlab-project-merge-request-draft-note-update command line option](#), 712  
[gitlab-project-merge-request-note-award-emoji-delete command line option](#), 715  
[gitlab-project-merge-request-note-award-emoji-get command line option](#), 716  
[gitlab-project-merge-request-note-delete command line option](#), 713  
[gitlab-project-merge-request-note-get command line option](#), 713  
[gitlab-project-merge-request-note-update command line option](#), 714  
[gitlab-project-merge-request-resource-label-event-get command line option](#), 717  
[gitlab-project-merge-request-resource-milestone-get command line option](#), 718  
[gitlab-project-merge-request-resource-state-event-get command line option](#), 719  
[gitlab-project-milestone-delete command line option](#), 722  
[gitlab-project-milestone-get command line option](#), 722  
[gitlab-project-milestone-issues command line option](#), 723  
[gitlab-project-milestone-merge-requests command line option](#), 723  
[gitlab-project-milestone-update command line option](#), 724  
[gitlab-project-mirror-pull command line option](#), 592  
[gitlab-project-mirror-pull-details command line option](#), 593  
[gitlab-project-note-get command line option](#), 724  
[gitlab-project-package-delete command line option](#), 727  
[gitlab-project-package-file-delete command line option](#), 728  
[gitlab-project-package-get command line option](#), 727  
[gitlab-project-pipeline-cancel command line option](#), 731  
[gitlab-project-pipeline-delete command line option](#), 732  
[gitlab-project-pipeline-get command line option](#), 732  
[gitlab-project-pipeline-retry command line option](#), 733  
[gitlab-project-pipeline-schedule-delete command line option](#), 735  
[gitlab-project-pipeline-schedule-get command line option](#), 736  
[gitlab-project-pipeline-schedule-play command line option](#), 736  
[gitlab-project-pipeline-schedule-take-ownership command line option](#), 737  
[gitlab-project-pipeline-schedule-update command line option](#), 737  
[gitlab-project-registry-repository-delete command line option](#), 750  
[gitlab-project-release-link-delete command line option](#), 755  
[gitlab-project-release-link-get command line option](#), 755  
[gitlab-project-release-link-update command line option](#), 756  
[gitlab-project-remote-mirror-delete command line option](#), 757  
[gitlab-project-remote-mirror-update command line option](#), 758  
[gitlab-project-repository-archive command line option](#), 593  
[gitlab-project-repository-blob command line option](#), 593  
[gitlab-project-repository-compare command line option](#), 593  
[gitlab-project-repository-contributors command line option](#), 594  
[gitlab-project-repository-merge-base command line option](#), 594  
[gitlab-project-repository-raw-blob command line option](#), 594  
[gitlab-project-repository-tree command line option](#), 594  
[gitlab-project-restore command line option](#), 595  
[gitlab-project-runner-delete command line option](#), 760  
[gitlab-project-search command line option](#), 595  
[gitlab-project-secure-file-delete command line option](#), 762  
[gitlab-project-secure-file-download command line option](#), 762  
[gitlab-project-secure-file-get command line option](#), 762  
[gitlab-project-share command line option](#), 595

gitlab-project-snapshot command line option, [595](#)  
gitlab-project-snippet-award-emoji-delete command line option, [768](#)  
gitlab-project-snippet-award-emoji-get command line option, [768](#)  
gitlab-project-snippet-content command line option, [765](#)  
gitlab-project-snippet-delete command line option, [766](#)  
gitlab-project-snippet-discussion-get command line option, [770](#)  
gitlab-project-snippet-discussion-note-delete command line option, [771](#)  
gitlab-project-snippet-discussion-note-get command line option, [772](#)  
gitlab-project-snippet-discussion-note-update command line option, [772](#)  
gitlab-project-snippet-get command line option, [766](#)  
gitlab-project-snippet-note-award-emoji-delete command line option, [775](#)  
gitlab-project-snippet-note-award-emoji-get command line option, [775](#)  
gitlab-project-snippet-note-delete command line option, [773](#)  
gitlab-project-snippet-note-get command line option, [773](#)  
gitlab-project-snippet-note-update command line option, [774](#)  
gitlab-project-snippet-update command line option, [767](#)  
gitlab-project-snippet-user-agent-detail command line option, [767](#)  
gitlab-project-star command line option, [596](#)  
gitlab-project-storage-get command line option, [776](#)  
gitlab-project-transfer command line option, [596](#)  
gitlab-project-trigger-delete command line option, [778](#)  
gitlab-project-trigger-get command line option, [779](#)  
gitlab-project-trigger-pipeline command line option, [596](#)  
gitlab-project-trigger-update command line option, [779](#)  
gitlab-project-unarchive command line option, [596](#)  
gitlab-project-unshare command line option, [597](#)  
gitlab-project-unstar command line option, [597](#)  
gitlab-project-update command line option, [599](#)  
gitlab-project-update-submodule command line option, [602](#)  
gitlab-project-upload command line option, [602](#)  
gitlab-registry-repository-get command line option, [785](#)  
gitlab-runner-all command line option, [785](#)  
gitlab-runner-delete command line option, [786](#)  
gitlab-runner-get command line option, [786](#)  
gitlab-runner-update command line option, [787](#)  
gitlab-runner-verify command line option, [788](#)  
gitlab-snippet-content command line option, [791](#)  
gitlab-snippet-delete command line option, [791](#)  
gitlab-snippet-get command line option, [792](#)  
gitlab-snippet-public command line option, [792](#)  
gitlab-snippet-update command line option, [792](#)  
gitlab-snippet-user-agent-detail command line option, [793](#)  
gitlab-todo-delete command line option, [794](#)  
gitlab-todo-mark-all-as-done command line option, [795](#)  
gitlab-todo-mark-as-done command line option, [795](#)  
gitlab-topic-delete command line option, [796](#)  
gitlab-topic-get command line option, [796](#)  
gitlab-topic-merge command line option, [797](#)  
gitlab-topic-update command line option, [797](#)  
gitlab-user-activate command line option, [798](#)  
gitlab-user-approve command line option, [798](#)  
gitlab-user-ban command line option, [798](#)  
gitlab-user-block command line option, [799](#)  
gitlab-user-deactivate command line option, [800](#)  
gitlab-user-delete command line option, [800](#)



- gitlab-user-email-delete command line option, [806](#)
- gitlab-user-email-get command line option, [806](#)
- gitlab-user-follow command line option, [801](#)
- gitlab-user-get command line option, [801](#)
- gitlab-user-gpg-key-delete command line option, [808](#)
- gitlab-user-gpg-key-get command line option, [808](#)
- gitlab-user-impersonation-token-delete command line option, [810](#)
- gitlab-user-impersonation-token-get command line option, [810](#)
- gitlab-user-key-delete command line option, [811](#)
- gitlab-user-key-get command line option, [811](#)
- gitlab-user-reject command line option, [802](#)
- gitlab-user-unban command line option, [802](#)
- gitlab-user-unblock command line option, [802](#)
- gitlab-user-unfollow command line option, [802](#)
- gitlab-user-update command line option, [803](#)
- id-after
  - gitlab-project-list command line option, [591](#)
  - gitlab-user-project-list command line option, [816](#)
- id-before
  - gitlab-project-list command line option, [591](#)
  - gitlab-user-project-list command line option, [816](#)
- iid
  - gitlab-group-epic-delete command line option, [531](#)
  - gitlab-group-epic-get command line option, [531](#)
  - gitlab-group-epic-update command line option, [532](#)
  - gitlab-project-issue-add-spent-time command line option, [653](#)
  - gitlab-project-issue-closed-by command line option, [654](#)
  - gitlab-project-issue-delete command line option, [655](#)
  - gitlab-project-issue-get command line option, [655](#)
  - gitlab-project-issue-move command line option, [656](#)
  - gitlab-project-issue-participants command line option, [657](#)
  - gitlab-project-issue-related-merge-requests command line option, [657](#)
  - gitlab-project-issue-reorder command line option, [657](#)
  - gitlab-project-issue-reset-spent-time command line option, [658](#)
  - gitlab-project-issue-reset-time-estimate command line option, [658](#)
  - gitlab-project-issue-subscribe command line option, [658](#)
  - gitlab-project-issue-time-estimate command line option, [659](#)
  - gitlab-project-issue-time-stats command line option, [659](#)
  - gitlab-project-issue-todo command line option, [659](#)
  - gitlab-project-issue-unsubscribe command line option, [659](#)
  - gitlab-project-issue-update command line option, [660](#)
  - gitlab-project-issue-user-agent-detail command line option, [661](#)
  - gitlab-project-merge-request-add-spent-time command line option, [689](#)
  - gitlab-project-merge-request-approve command line option, [689](#)
  - gitlab-project-merge-request-cancel-merge-when-pipeline-fails command line option, [690](#)
  - gitlab-project-merge-request-changes command line option, [690](#)
  - gitlab-project-merge-request-closes-issues command line option, [690](#)
  - gitlab-project-merge-request-commits command line option, [691](#)
  - gitlab-project-merge-request-delete command line option, [692](#)
  - gitlab-project-merge-request-get command line option, [692](#)
  - gitlab-project-merge-request-merge command line option, [694](#)
  - gitlab-project-merge-request-merge-ref command line option, [695](#)
  - gitlab-project-merge-request-participants command line option, [695](#)
  - gitlab-project-merge-request-rebase command line option, [695](#)
  - gitlab-project-merge-request-reset-approvals command line option, [695](#)
  - gitlab-project-merge-request-reset-spent-time command line option, [696](#)

gitlab-project-merge-request-reset-time-estimate option, [580](#)  
command line option, [696](#)  
gitlab-project-merge-request-subscribe  
command line option, [696](#)  
gitlab-project-merge-request-time-estimate  
command line option, [697](#)  
gitlab-project-merge-request-time-stats  
command line option, [697](#)  
gitlab-project-merge-request-todo  
command line option, [697](#)  
gitlab-project-merge-request-unapprove  
command line option, [698](#)  
gitlab-project-merge-request-unsubscribe  
command line option, [698](#)  
gitlab-project-merge-request-update  
command line option, [698](#)  
--iids  
gitlab-group-issue-list command line  
option, [549](#)  
gitlab-group-milestone-list command  
line option, [561](#)  
gitlab-issue-list command line option,  
[576](#)  
gitlab-project-issue-list command line  
option, [655](#)  
gitlab-project-merge-request-list  
command line option, [693](#)  
gitlab-project-milestone-list command  
line option, [723](#)  
--image-url  
gitlab-group-badge-create command line  
option, [517](#)  
gitlab-group-badge-render command line  
option, [519](#)  
gitlab-group-badge-update command line  
option, [519](#)  
gitlab-project-badge-create command  
line option, [611](#)  
gitlab-project-badge-render command  
line option, [612](#)  
gitlab-project-badge-update command  
line option, [613](#)  
--import-sources  
gitlab-application-settings-update  
command line option, [486](#)  
--import-url  
gitlab-project-create command line  
option, [588](#)  
gitlab-project-update command line  
option, [600](#)  
gitlab-user-project-create command line  
option, [814](#)  
--in  
gitlab-merge-request-list command line  
gitlab-project-deployment-merge-request-list  
command line option, [637](#)  
--in-reply-to-discussion-id  
gitlab-project-merge-request-draft-note-create  
command line option, [711](#)  
--include-ancestors  
gitlab-group-iteration-list command  
line option, [551](#)  
--include-diverged-commits-count  
gitlab-project-merge-request-get  
command line option, [692](#)  
--include-html-description  
gitlab-project-release-list command  
line option, [753](#)  
gitlab-project-resource-group-list  
command line option, [759](#)  
--include-jobs  
gitlab-ci-lint-create command line  
option, [493](#)  
gitlab-ci-lint-validate command line  
option, [494](#)  
gitlab-project-ci-lint-create command  
line option, [618](#)  
gitlab-project-ci-lint-get command line  
option, [618](#)  
gitlab-project-ci-lint-validate command  
line option, [619](#)  
--include-merged-yaml  
gitlab-ci-lint-create command line  
option, [493](#)  
gitlab-ci-lint-validate command line  
option, [494](#)  
--include-rebase-in-progress  
gitlab-project-merge-request-get  
command line option, [692](#)  
--include-retried  
gitlab-project-pipeline-job-list  
command line option, [734](#)  
--include-subgroups  
gitlab-group-project-list command line  
option, [565](#)  
--info  
gitlab-runner-create command line  
option, [786](#)  
--infrastructure-access-level  
gitlab-project-create command line  
option, [589](#)  
gitlab-project-update command line  
option, [601](#)  
--initialize-with-readme  
gitlab-project-create command line  
option, [588](#)  
--instance-statistics-visibility-private



---

```

gitlab-application-settings-update
    command line option, 487
--invite-source
    gitlab-group-invitation-create command
        line option, 547
    gitlab-project-invitation-create
        command line option, 651
--issue-branch-template
    gitlab-project-update command line
        option, 601
--issue-id
    gitlab-group-epic-issue-create command
        line option, 536
--issue-iid
    gitlab-project-issue-award-emoji-create
        command line option, 661
    gitlab-project-issue-award-emoji-delete
        command line option, 661
    gitlab-project-issue-award-emoji-get
        command line option, 662
    gitlab-project-issue-award-emoji-list
        command line option, 662
    gitlab-project-issue-discussion-create
        command line option, 663
    gitlab-project-issue-discussion-get
        command line option, 663
    gitlab-project-issue-discussion-list
        command line option, 663
    gitlab-project-issue-discussion-note-create
        command line option, 664
    gitlab-project-issue-discussion-note-delete
        command line option, 665
    gitlab-project-issue-discussion-note-get
        command line option, 665
    gitlab-project-issue-discussion-note-update-issues-enabled
        command line option, 665
    gitlab-project-issue-link-create
        command line option, 666
    gitlab-project-issue-link-delete
        command line option, 666
    gitlab-project-issue-link-list command
        line option, 667
    gitlab-project-issue-note-award-emoji-create
        command line option, 669
    gitlab-project-issue-note-award-emoji-delete
        command line option, 670
    gitlab-project-issue-note-award-emoji-get
        command line option, 670
    gitlab-project-issue-note-award-emoji-list
        command line option, 671
    gitlab-project-issue-note-create
        command line option, 667
    gitlab-project-issue-note-delete
        command line option, 668
    gitlab-project-issue-note-get command
        line option, 668
    gitlab-project-issue-note-list command
        line option, 668
    gitlab-project-issue-note-update
        command line option, 669
    gitlab-project-issue-resource-iteration-event-get
        command line option, 671
    gitlab-project-issue-resource-iteration-event-list
        command line option, 672
    gitlab-project-issue-resource-label-event-get
        command line option, 672
    gitlab-project-issue-resource-label-event-list
        command line option, 673
    gitlab-project-issue-resource-milestone-event-get
        command line option, 673
    gitlab-project-issue-resource-milestone-event-list
        command line option, 674
    gitlab-project-issue-resource-state-event-get
        command line option, 674
    gitlab-project-issue-resource-state-event-list
        command line option, 675
    gitlab-project-issue-resource-weight-event-get
        command line option, 675
    gitlab-project-issue-resource-weight-event-list
        command line option, 675
--issue-link-id
    gitlab-project-issue-link-delete
        command line option, 666
--issues-access-level
    gitlab-project-create command line
        option, 588
    gitlab-project-update command line
        option, 600
    gitlab-project-create command line
        option, 588
    gitlab-project-update command line
        option, 600
    gitlab-user-project-create command line
        option, 814
--issues-events
    gitlab-group-hook-create command line
        option, 543
    gitlab-group-hook-update command line
        option, 545
    gitlab-project-hook-create command line
        option, 646
    gitlab-project-hook-update command line
        option, 648
--issues-template
    gitlab-project-update command line
        option, 600
--iteration-id

```

- gitlab-group-issue-list command line option, [549](#)
- gitlab-issue-list command line option, [576](#)
- gitlab-project-issue-list command line option, [656](#)
- job
  - gitlab-project-artifact-download command line option, [609](#)
  - gitlab-project-artifact-raw command line option, [609](#)
- job-events
  - gitlab-group-hook-create command line option, [544](#)
  - gitlab-group-hook-update command line option, [546](#)
  - gitlab-project-hook-create command line option, [646](#)
  - gitlab-project-hook-update command line option, [648](#)
- job-token
  - gitlab command line option, [481](#)
  - gitlab-project-artifact-download command line option, [609](#)
- jobs-enabled
  - gitlab-project-create command line option, [588](#)
  - gitlab-project-update command line option, [600](#)
- keep-latest-artifact
  - gitlab-project-update command line option, [600](#)
- keep-n
  - gitlab-project-registry-tag-delete-in-bulk command line option, [751](#)
- key
  - gitlab-current-user-gpg-key-create command line option, [496](#)
  - gitlab-current-user-key-create command line option, [497](#)
  - gitlab-group-custom-attribute-delete command line option, [527](#)
  - gitlab-group-custom-attribute-get command line option, [527](#)
  - gitlab-group-variable-create command line option, [571](#)
  - gitlab-group-variable-delete command line option, [571](#)
  - gitlab-group-variable-get command line option, [571](#)
  - gitlab-group-variable-update command line option, [572](#)
  - gitlab-license-get command line option, [578](#)
  - gitlab-project-custom-attribute-delete command line option, [631](#)
  - gitlab-project-custom-attribute-get command line option, [631](#)
  - gitlab-project-key-create command line option, [681](#)
  - gitlab-project-pages-domain-create command line option, [730](#)
  - gitlab-project-pages-domain-update command line option, [731](#)
  - gitlab-project-pipeline-schedule-variable-create command line option, [739](#)
  - gitlab-project-pipeline-schedule-variable-delete command line option, [739](#)
  - gitlab-project-pipeline-schedule-variable-update command line option, [739](#)
  - gitlab-project-resource-group-get command line option, [758](#)
  - gitlab-project-resource-group-update command line option, [759](#)
  - gitlab-project-variable-create command line option, [781](#)
  - gitlab-project-variable-delete command line option, [781](#)
  - gitlab-project-variable-get command line option, [781](#)
  - gitlab-project-variable-update command line option, [782](#)
  - gitlab-user-custom-attribute-delete command line option, [805](#)
  - gitlab-user-custom-attribute-get command line option, [805](#)
  - gitlab-user-gpg-key-create command line option, [808](#)
  - gitlab-user-key-create command line option, [811](#)
  - gitlab-variable-create command line option, [817](#)
  - gitlab-variable-delete command line option, [817](#)
  - gitlab-variable-get command line option, [817](#)
  - gitlab-variable-update command line option, [818](#)
- key-id
  - gitlab-project-key-enable command line option, [681](#)
- label-id
  - gitlab-group-board-list-create command line option, [523](#)
  - gitlab-project-board-list-create command line option, [615](#)
- labels
  - gitlab-group-epic-create command line

- option, [531](#)
- gitlab-group-epic-list command line option, [532](#)
- gitlab-group-epic-update command line option, [532](#)
- gitlab-group-issue-list command line option, [549](#)
- gitlab-group-merge-request-list command line option, [558](#)
- gitlab-issue-list command line option, [576](#)
- gitlab-merge-request-list command line option, [580](#)
- gitlab-project-deployment-merge-request-list command line option, [636](#)
- gitlab-project-issue-create command line option, [654](#)
- gitlab-project-issue-list command line option, [656](#)
- gitlab-project-issue-update command line option, [660](#)
- gitlab-project-merge-request-create command line option, [692](#)
- gitlab-project-merge-request-list command line option, [693](#)
- gitlab-project-merge-request-update command line option, [699](#)
- last-activity-after
  - gitlab-project-list command line option, [591](#)
- last-activity-before
  - gitlab-project-list command line option, [591](#)
- level
  - gitlab-group-notification-settings-update command line option, [562](#)
  - gitlab-notification-settings-update command line option, [583](#)
  - gitlab-project-notification-settings-update command line option, [726](#)
- lfs-enabled
  - gitlab-group-create command line option, [508](#)
  - gitlab-group-update command line option, [513](#)
  - gitlab-project-create command line option, [588](#)
  - gitlab-project-update command line option, [600](#)
- line
  - gitlab-project-commit-comment-create command line option, [625](#)
- line-type
  - gitlab-project-commit-comment-create command line option, [625](#)
- link-type
  - gitlab-project-release-link-create command line option, [755](#)
  - gitlab-project-release-link-update command line option, [756](#)
- link-url
  - gitlab-group-badge-create command line option, [517](#)
  - gitlab-group-badge-render command line option, [519](#)
  - gitlab-group-badge-update command line option, [519](#)
  - gitlab-project-badge-create command line option, [611](#)
  - gitlab-project-badge-render command line option, [612](#)
  - gitlab-project-badge-update command line option, [612](#)
- linkedin
  - gitlab-user-create command line option, [799](#)
  - gitlab-user-update command line option, [803](#)
- local-markdown-version
  - gitlab-application-settings-update command line option, [487](#)
- location
  - gitlab-user-create command line option, [800](#)
  - gitlab-user-update command line option, [804](#)
- locked
  - gitlab-current-user-runner-create command line option, [498](#)
  - gitlab-runner-create command line option, [786](#)
  - gitlab-runner-update command line option, [787](#)
- logo
  - gitlab-application-appearance-update command line option, [483](#)
- maintenance-note
  - gitlab-current-user-runner-create command line option, [499](#)
- managed
  - gitlab-group-cluster-create command line option, [525](#)
  - gitlab-project-cluster-create command line option, [620](#)
- management-project-id
  - gitlab-group-cluster-update command line option, [526](#)
  - gitlab-project-cluster-update command

- line option, [621](#)
- masked
  - gitlab-group-variable-create command line option, [571](#)
  - gitlab-group-variable-update command line option, [572](#)
  - gitlab-project-variable-create command line option, [781](#)
  - gitlab-project-variable-update command line option, [782](#)
  - gitlab-variable-create command line option, [817](#)
  - gitlab-variable-update command line option, [818](#)
- max-attachment-size
  - gitlab-application-settings-update command line option, [486](#)
- max-file-size
  - gitlab-group-push-rules-create command line option, [566](#)
  - gitlab-group-push-rules-update command line option, [567](#)
  - gitlab-project-push-rules-create command line option, [748](#)
  - gitlab-project-push-rules-update command line option, [749](#)
- maximum-timeout
  - gitlab-current-user-runner-create command line option, [499](#)
  - gitlab-runner-create command line option, [786](#)
  - gitlab-runner-update command line option, [787](#)
- member-check
  - gitlab-group-push-rules-create command line option, [566](#)
  - gitlab-group-push-rules-update command line option, [567](#)
  - gitlab-project-push-rules-create command line option, [748](#)
  - gitlab-project-push-rules-update command line option, [749](#)
- membership
  - gitlab-project-fork-list command line option, [645](#)
  - gitlab-project-list command line option, [592](#)
  - gitlab-starred-project-list command line option, [794](#)
  - gitlab-user-project-list command line option, [815](#)
- membership-lock
  - gitlab-group-create command line option, [508](#)
  - gitlab-group-update command line option, [512](#)
- mentions-disabled
  - gitlab-group-create command line option, [508](#)
  - gitlab-group-update command line option, [513](#)
- merge-access-level
  - gitlab-project-protected-branch-create command line option, [742](#)
- merge-commit-message
  - gitlab-project-merge-request-merge command line option, [694](#)
- merge-commit-template
  - gitlab-project-update command line option, [600](#)
- merge-merge-request
  - gitlab-group-notification-settings-update command line option, [563](#)
  - gitlab-notification-settings-update command line option, [583](#)
  - gitlab-project-notification-settings-update command line option, [726](#)
- merge-method
  - gitlab-project-create command line option, [588](#)
  - gitlab-project-update command line option, [600](#)
- merge-pipelines-enabled
  - gitlab-project-create command line option, [588](#)
  - gitlab-project-update command line option, [600](#)
- merge-request-iid
  - gitlab-project-merge-request-approval-rule-create command line option, [701](#)
  - gitlab-project-merge-request-approval-rule-update command line option, [702](#)
- merge-request-to-resolve-discussions-of
  - gitlab-project-issue-create command line option, [654](#)
- merge-requests-access-level
  - gitlab-project-create command line option, [588](#)
  - gitlab-project-update command line option, [600](#)
- merge-requests-author-approval
  - gitlab-project-approval-update command line option, [606](#)
- merge-requests-disable-committers-approval
  - gitlab-project-approval-update command line option, [606](#)
- merge-requests-enabled
  - gitlab-project-create command line

- option, [588](#)
- gitlab-project-update command line option, [600](#)
- gitlab-user-project-create command line option, [814](#)
- merge-requests-events
  - gitlab-group-hook-create command line option, [543](#)
  - gitlab-group-hook-update command line option, [545](#)
  - gitlab-project-hook-create command line option, [646](#)
  - gitlab-project-hook-update command line option, [648](#)
- merge-requests-template
  - gitlab-project-update command line option, [600](#)
- merge-trains-enabled
  - gitlab-project-update command line option, [600](#)
- merge-when-pipeline-succeeds
  - gitlab-project-merge-request-merge command line option, [694](#)
- message
  - gitlab-broadcast-message-create command line option, [489](#)
  - gitlab-broadcast-message-update command line option, [490](#)
  - gitlab-current-user-status-update command line option, [499](#)
  - gitlab-project-tag-create command line option, [777](#)
- message-background-color
  - gitlab-application-appearance-update command line option, [483](#)
- message-font-color
  - gitlab-application-appearance-update command line option, [483](#)
- milestone
  - gitlab-group-issue-list command line option, [549](#)
  - gitlab-group-merge-request-list command line option, [558](#)
  - gitlab-issue-list command line option, [576](#)
  - gitlab-merge-request-list command line option, [580](#)
  - gitlab-project-deployment-merge-request-list command line option, [636](#)
  - gitlab-project-issue-list command line option, [656](#)
  - gitlab-project-merge-request-list command line option, [693](#)
- milestone-id
  - gitlab-group-board-list-create command line option, [523](#)
  - gitlab-project-board-list-create command line option, [615](#)
  - gitlab-project-issue-create command line option, [654](#)
  - gitlab-project-issue-update command line option, [660](#)
  - gitlab-project-merge-request-create command line option, [692](#)
  - gitlab-project-merge-request-update command line option, [699](#)
- milestones
  - gitlab-project-release-update command line option, [754](#)
- min-access-level
  - gitlab-group-descendant-group-list command line option, [530](#)
  - gitlab-group-list command line option, [510](#)
  - gitlab-group-project-list command line option, [565](#)
  - gitlab-group-subgroup-list command line option, [570](#)
  - gitlab-project-list command line option, [592](#)
  - gitlab-shared-project-list command line option, [790](#)
  - gitlab-starred-project-list command line option, [794](#)
  - gitlab-user-project-list command line option, [816](#)
- mirror
  - gitlab-project-create command line option, [588](#)
  - gitlab-project-update command line option, [600](#)
- mirror-overwrites-diverged-branches
  - gitlab-project-update command line option, [600](#)
- mirror-trigger-builds
  - gitlab-project-create command line option, [588](#)
  - gitlab-project-update command line option, [600](#)
- mirror-user-id
  - gitlab-project-update command line option, [600](#)
- monitor-access-level
  - gitlab-project-create command line option, [589](#)
  - gitlab-project-update command line option, [601](#)
- move-after-id

gitlab-group-epic-issue-update command  
line option, 537

gitlab-project-issue-reorder command  
line option, 657

--move-before-id  
gitlab-group-epic-issue-update command  
line option, 537

gitlab-project-issue-reorder command  
line option, 657

--mr-default-target-self  
gitlab-project-update command line  
option, 600

--mr-iid  
gitlab-project-merge-request-approval-get  
command line option, 699

gitlab-project-merge-request-approval-rule-create  
command line option, 701

gitlab-project-merge-request-approval-rule-delete  
command line option, 701

gitlab-project-merge-request-approval-rule-get  
command line option, 701

gitlab-project-merge-request-approval-rule-list  
command line option, 702

gitlab-project-merge-request-approval-rule-update  
command line option, 702

gitlab-project-merge-request-approval-state-get  
command line option, 703

gitlab-project-merge-request-approval-update  
command line option, 700

gitlab-project-merge-request-award-emoji-create  
command line option, 704

gitlab-project-merge-request-award-emoji-delete  
command line option, 704

gitlab-project-merge-request-award-emoji-get  
command line option, 704

gitlab-project-merge-request-award-emoji-list  
command line option, 705

gitlab-project-merge-request-diff-get  
command line option, 705

gitlab-project-merge-request-diff-list  
command line option, 706

gitlab-project-merge-request-discussion-create  
command line option, 706

gitlab-project-merge-request-discussion-get  
command line option, 707

gitlab-project-merge-request-discussion-list  
command line option, 707

gitlab-project-merge-request-discussion-note-create  
command line option, 708

gitlab-project-merge-request-discussion-note-delete  
command line option, 709

gitlab-project-merge-request-discussion-note-get  
command line option, 709

gitlab-project-merge-request-discussion-note-update  
command line option, 710

gitlab-project-merge-request-discussion-update  
command line option, 708

gitlab-project-merge-request-draft-note-create  
command line option, 711

gitlab-project-merge-request-draft-note-delete  
command line option, 711

gitlab-project-merge-request-draft-note-get  
command line option, 711

gitlab-project-merge-request-draft-note-list  
command line option, 712

gitlab-project-merge-request-draft-note-update  
command line option, 712

gitlab-project-merge-request-note-award-emoji-create  
command line option, 715

gitlab-project-merge-request-note-award-emoji-delete  
command line option, 715

gitlab-project-merge-request-note-award-emoji-get  
command line option, 715

gitlab-project-merge-request-note-award-emoji-list  
command line option, 716

gitlab-project-merge-request-note-create  
command line option, 713

gitlab-project-merge-request-note-delete  
command line option, 713

gitlab-project-merge-request-note-get  
command line option, 713

gitlab-project-merge-request-note-list  
command line option, 714

gitlab-project-merge-request-note-update  
command line option, 714

gitlab-project-merge-request-pipeline-create  
command line option, 716

gitlab-project-merge-request-pipeline-list  
command line option, 717

gitlab-project-merge-request-resource-label-event-get  
command line option, 717

gitlab-project-merge-request-resource-label-event-list  
command line option, 718

gitlab-project-merge-request-resource-milestone-event-  
command line option, 718

gitlab-project-merge-request-resource-milestone-event-  
command line option, 719

gitlab-project-merge-request-resource-state-event-get  
command line option, 719

gitlab-project-merge-request-resource-state-event-list  
command line option, 720

gitlab-project-merge-request-reviewer-detail-list  
command line option, 720

gitlab-project-merge-request-reviewer-detail-list  
command line option, 720

gitlab-group-issue-list command line  
option, 549

gitlab-group-merge-request-list command  
line option, 558



- gitlab-issue-list command line option, [576](#)
- gitlab-merge-request-list command line option, [580](#)
- gitlab-project-deployment-merge-request-list command line option, [637](#)
- gitlab-project-issue-list command line option, [656](#)
- gitlab-project-merge-request-list command line option, [694](#)
- name
  - gitlab-application-create command line option, [481](#)
  - gitlab-dockerfile-get command line option, [501](#)
  - gitlab-feature-delete command line option, [502](#)
  - gitlab-gitignore-get command line option, [506](#)
  - gitlab-gitlabciyaml-get command line option, [507](#)
  - gitlab-group-access-token-create command line option, [515](#)
  - gitlab-group-board-create command line option, [521](#)
  - gitlab-group-cluster-create command line option, [525](#)
  - gitlab-group-cluster-update command line option, [526](#)
  - gitlab-group-create command line option, [508](#)
  - gitlab-group-deploy-token-create command line option, [528](#)
  - gitlab-group-epic-award-emoji-create command line option, [533](#)
  - gitlab-group-epic-note-award-emoji-create command line option, [540](#)
  - gitlab-group-label-create command line option, [551](#)
  - gitlab-group-label-delete command line option, [552](#)
  - gitlab-group-label-get command line option, [552](#)
  - gitlab-group-label-subscribe command line option, [553](#)
  - gitlab-group-label-unsubscribe command line option, [553](#)
  - gitlab-group-label-update command line option, [553](#)
  - gitlab-group-saml-group-link-delete command line option, [568](#)
  - gitlab-group-saml-group-link-get command line option, [569](#)
  - gitlab-group-update command line option, [512](#)
  - gitlab-project-access-token-create command line option, [604](#)
  - gitlab-project-approval-rule-create command line option, [607](#)
  - gitlab-project-board-create command line option, [613](#)
  - gitlab-project-branch-delete command line option, [617](#)
  - gitlab-project-branch-get command line option, [617](#)
  - gitlab-project-cluster-create command line option, [619](#)
  - gitlab-project-cluster-update command line option, [621](#)
  - gitlab-project-commit-status-create command line option, [630](#)
  - gitlab-project-create command line option, [587](#)
  - gitlab-project-deploy-token-create command line option, [632](#)
  - gitlab-project-environment-create command line option, [638](#)
  - gitlab-project-environment-list command line option, [639](#)
  - gitlab-project-environment-update command line option, [640](#)
  - gitlab-project-issue-award-emoji-create command line option, [661](#)
  - gitlab-project-issue-note-award-emoji-create command line option, [669](#)
  - gitlab-project-label-create command line option, [683](#)
  - gitlab-project-label-delete command line option, [683](#)
  - gitlab-project-label-get command line option, [684](#)
  - gitlab-project-label-subscribe command line option, [684](#)
  - gitlab-project-label-unsubscribe command line option, [685](#)
  - gitlab-project-label-update command line option, [685](#)
  - gitlab-project-merge-request-approval-rule-create command line option, [701](#)
  - gitlab-project-merge-request-approval-rule-update command line option, [702](#)
  - gitlab-project-merge-request-award-emoji-create command line option, [704](#)
  - gitlab-project-merge-request-note-award-emoji-create command line option, [715](#)
  - gitlab-project-pipeline-list command line option, [733](#)
  - gitlab-project-protected-branch-create

command line option, [742](#)  
gitlab-project-protected-branch-delete  
command line option, [742](#)  
gitlab-project-protected-branch-get  
command line option, [743](#)  
gitlab-project-protected-branch-update  
command line option, [743](#)  
gitlab-project-protected-environment-create  
command line option, [744](#)  
gitlab-project-protected-environment-delete  
command line option, [744](#)  
gitlab-project-protected-environment-get  
command line option, [745](#)  
gitlab-project-protected-tag-create  
command line option, [746](#)  
gitlab-project-protected-tag-delete  
command line option, [746](#)  
gitlab-project-protected-tag-get  
command line option, [746](#)  
gitlab-project-registry-tag-delete  
command line option, [751](#)  
gitlab-project-registry-tag-delete-in-bulk  
command line option, [751](#)  
gitlab-project-registry-tag-get command  
line option, [751](#)  
gitlab-project-release-create command  
line option, [752](#)  
gitlab-project-release-link-create  
command line option, [754](#)  
gitlab-project-release-link-update  
command line option, [756](#)  
gitlab-project-release-update command  
line option, [754](#)  
gitlab-project-secure-file-create  
command line option, [761](#)  
gitlab-project-snippet-award-emoji-create  
command line option, [768](#)  
gitlab-project-snippet-note-award-emoji-create  
command line option, [775](#)  
gitlab-project-tag-delete command line  
option, [777](#)  
gitlab-project-tag-get command line  
option, [777](#)  
gitlab-project-update command line  
option, [600](#)  
gitlab-topic-create command line option,  
[796](#)  
gitlab-topic-update command line option,  
[797](#)  
gitlab-user-create command line option,  
[799](#)  
gitlab-user-impersonation-token-create  
command line option, [809](#)  
gitlab-user-personal-access-token-create  
command line option, [813](#)  
gitlab-user-project-create command line  
option, [814](#)  
gitlab-user-update command line option,  
[803](#)  
--name-regex-delete  
gitlab-project-registry-tag-delete-in-bulk  
command line option, [751](#)  
--name-regex-keep  
gitlab-project-registry-tag-delete-in-bulk  
command line option, [751](#)  
--namespace  
gitlab-namespace-exists command line  
option, [581](#)  
gitlab-project-fork-create command line  
option, [644](#)  
--namespace-id  
gitlab-project-create command line  
option, [588](#)  
--new-issue  
gitlab-group-notification-settings-update  
command line option, [563](#)  
gitlab-notification-settings-update  
command line option, [583](#)  
gitlab-project-notification-settings-update  
command line option, [726](#)  
--new-merge-request  
gitlab-group-notification-settings-update  
command line option, [563](#)  
gitlab-notification-settings-update  
command line option, [583](#)  
gitlab-project-notification-settings-update  
command line option, [726](#)  
--new-name  
gitlab-group-label-update command line  
option, [553](#)  
gitlab-project-label-update command  
line option, [685](#)  
--new-note  
gitlab-group-notification-settings-update  
command line option, [563](#)  
gitlab-notification-settings-update  
command line option, [583](#)  
gitlab-project-notification-settings-update  
command line option, [726](#)  
--new-project-guidelines  
gitlab-application-appearance-update  
command line option, [483](#)  
--no-ssl-verify  
gitlab command line option, [480](#)  
--not  
gitlab-merge-request-list command line  
option, [580](#)  
gitlab-project-deployment-merge-request-list



- command line option, [637](#)
- note
  - gitlab-project-commit-comment-create
    - command line option, [625](#)
  - gitlab-project-merge-request-draft-note-create
    - command line option, [711](#)
- note-events
  - gitlab-group-hook-create
    - command line option, [544](#)
  - gitlab-group-hook-update
    - command line option, [546](#)
  - gitlab-project-hook-create
    - command line option, [646](#)
  - gitlab-project-hook-update
    - command line option, [648](#)
- note-id
  - gitlab-group-epic-note-award-emoji-create
    - command line option, [540](#)
  - gitlab-group-epic-note-award-emoji-delete
    - command line option, [540](#)
  - gitlab-group-epic-note-award-emoji-get
    - command line option, [540](#)
  - gitlab-group-epic-note-award-emoji-list
    - command line option, [541](#)
  - gitlab-project-issue-note-award-emoji-create
    - command line option, [669](#)
  - gitlab-project-issue-note-award-emoji-delete
    - command line option, [670](#)
  - gitlab-project-issue-note-award-emoji-get
    - command line option, [670](#)
  - gitlab-project-issue-note-award-emoji-list
    - command line option, [671](#)
  - gitlab-project-merge-request-note-award-emoji-create
    - command line option, [715](#)
  - gitlab-project-merge-request-note-award-emoji-delete
    - command line option, [715](#)
  - gitlab-project-merge-request-note-award-emoji-get
    - command line option, [715](#)
  - gitlab-project-merge-request-note-award-emoji-list
    - command line option, [716](#)
  - gitlab-project-snippet-note-award-emoji-create
    - command line option, [775](#)
  - gitlab-project-snippet-note-award-emoji-delete
    - command line option, [775](#)
  - gitlab-project-snippet-note-award-emoji-get
    - command line option, [775](#)
  - gitlab-project-snippet-note-award-emoji-list
    - command line option, [776](#)
- notification-email
  - gitlab-group-notification-settings-update
    - command line option, [563](#)
  - gitlab-notification-settings-update
    - command line option, [583](#)
  - gitlab-project-notification-settings-update
    - command line option, [726](#)
- oauth-token
  - gitlab command line option, [481](#)
- older-than
  - gitlab-project-registry-tag-delete-in-bulk
    - command line option, [751](#)
- only-allow-merge-if-all-discussions-are-resolved
  - gitlab-project-create
    - command line option, [588](#)
  - gitlab-project-update
    - command line option, [601](#)
- only-allow-merge-if-all-status-checks-passed
  - gitlab-project-create
    - command line option, [587](#)
  - gitlab-project-update
    - command line option, [599](#)
- only-allow-merge-if-build-succeeds
  - gitlab-user-project-create
    - command line option, [814](#)
- only-allow-merge-if-pipeline-succeeds
  - gitlab-project-create
    - command line option, [588](#)
  - gitlab-project-update
    - command line option, [601](#)
- only-mirror-protected-branches
  - gitlab-project-update
    - command line option, [601](#)
- only-protected-branches
  - gitlab-project-remote-mirror-create
    - command line option, [757](#)
  - gitlab-project-remote-mirror-update
    - command line option, [758](#)
- operations-access-level
  - gitlab-project-create
    - command line option, [588](#)
  - gitlab-project-update
    - command line option, [600](#)
- order
  - gitlab-project-commit-list
    - command line option, [623](#)
- order-by
  - gitlab command line option, [481](#)
- gitlab-group-descendant-group-list
  - command line option, [530](#)
- gitlab-group-epic-list
  - command line option, [532](#)
- gitlab-group-issue-list
  - command line option, [549](#)
- gitlab-group-list
  - command line option, [510](#)
- gitlab-group-merge-request-list
  - command line option, [558](#)
- gitlab-group-package-list
  - command line option, [563](#)

- gitlab-group-project-list command line option, [564](#)
- gitlab-group-subgroup-list command line option, [570](#)
- gitlab-issue-list command line option, [576](#)
- gitlab-merge-request-list command line option, [580](#)
- gitlab-project-deployment-list command line option, [635](#)
- gitlab-project-deployment-merge-request-list command line option, [636](#)
- gitlab-project-fork-list command line option, [645](#)
- gitlab-project-issue-list command line option, [656](#)
- gitlab-project-list command line option, [592](#)
- gitlab-project-merge-request-list command line option, [693](#)
- gitlab-project-package-list command line option, [727](#)
- gitlab-project-pipeline-list command line option, [733](#)
- gitlab-project-release-list command line option, [753](#)
- gitlab-project-resource-group-list command line option, [759](#)
- gitlab-shared-project-list command line option, [790](#)
- gitlab-starred-project-list command line option, [794](#)
- gitlab-user-project-list command line option, [815](#)
- organization
  - gitlab-user-create command line option, [800](#)
  - gitlab-user-update command line option, [803](#)
- outbound-local-requests-whitelist
  - gitlab-application-settings-update command line option, [486](#)
- output
  - gitlab command line option, [480](#)
- owned
  - gitlab-group-descendant-group-list command line option, [530](#)
  - gitlab-group-list command line option, [510](#)
  - gitlab-group-project-list command line option, [564](#)
  - gitlab-group-subgroup-list command line option, [570](#)
  - gitlab-project-fork-list command line option, [645](#)
  - gitlab-project-list command line option, [592](#)
  - gitlab-starred-project-list command line option, [794](#)
  - gitlab-user-project-list command line option, [815](#)
- package-id
  - gitlab-project-package-file-delete command line option, [728](#)
  - gitlab-project-package-file-list command line option, [728](#)
  - gitlab-project-package-pipeline-list command line option, [729](#)
- package-name
  - gitlab-generic-package-download command line option, [503](#)
  - gitlab-generic-package-upload command line option, [503](#)
  - gitlab-group-package-list command line option, [563](#)
  - gitlab-project-package-list command line option, [727](#)
- package-type
  - gitlab-group-package-list command line option, [563](#)
  - gitlab-project-package-list command line option, [727](#)
- package-version
  - gitlab-generic-package-download command line option, [503](#)
  - gitlab-generic-package-upload command line option, [503](#)
- packages-enabled
  - gitlab-project-create command line option, [588](#)
  - gitlab-project-update command line option, [601](#)
- page
  - gitlab-application-list command line option, [482](#)
  - gitlab-audit-event-list command line option, [488](#)
  - gitlab-broadcast-message-list command line option, [490](#)
  - gitlab-bulk-import-all-entity-list command line option, [492](#)
  - gitlab-bulk-import-entity-list command line option, [493](#)
  - gitlab-bulk-import-list command line option, [491](#)
  - gitlab-current-user-email-list command line option, [495](#)
  - gitlab-current-user-gpg-key-list

- command line option, [496](#)
- gitlab-current-user-key-list command line option, [498](#)
- gitlab-deploy-key-list command line option, [500](#)
- gitlab-deploy-token-list command line option, [500](#)
- gitlab-dockerfile-list command line option, [501](#)
- gitlab-event-list command line option, [502](#)
- gitlab-feature-list command line option, [502](#)
- gitlab-geo-node-list command line option, [505](#)
- gitlab-gitignore-list command line option, [506](#)
- gitlab-gitlabciyaml-list command line option, [507](#)
- gitlab-group-access-request-list command line option, [514](#)
- gitlab-group-access-token-list command line option, [516](#)
- gitlab-group-audit-event-list command line option, [517](#)
- gitlab-group-badge-list command line option, [518](#)
- gitlab-group-billable-member-list command line option, [520](#)
- gitlab-group-billable-member-membership-list command line option, [521](#)
- gitlab-group-board-list command line option, [522](#)
- gitlab-group-board-list-list command line option, [524](#)
- gitlab-group-cluster-list command line option, [526](#)
- gitlab-group-custom-attribute-list command line option, [528](#)
- gitlab-group-deploy-token-list command line option, [529](#)
- gitlab-group-descendant-group-list command line option, [530](#)
- gitlab-group-epic-award-emoji-list command line option, [534](#)
- gitlab-group-epic-issue-list command line option, [537](#)
- gitlab-group-epic-list command line option, [532](#)
- gitlab-group-epic-note-award-emoji-list command line option, [541](#)
- gitlab-group-epic-note-list command line option, [539](#)
- gitlab-group-epic-resource-label-event-list command line option, [542](#)
- gitlab-group-hook-list command line option, [545](#)
- gitlab-group-invitation-list command line option, [548](#)
- gitlab-group-issue-list command line option, [550](#)
- gitlab-group-iteration-list command line option, [551](#)
- gitlab-group-label-list command line option, [552](#)
- gitlab-group-ldap-group-link-list command line option, [555](#)
- gitlab-group-list command line option, [510](#)
- gitlab-group-member-all-list command line option, [557](#)
- gitlab-group-member-list command line option, [556](#)
- gitlab-group-merge-request-list command line option, [559](#)
- gitlab-group-milestone-list command line option, [561](#)
- gitlab-group-package-list command line option, [564](#)
- gitlab-group-project-list command line option, [565](#)
- gitlab-group-runner-list command line option, [568](#)
- gitlab-group-saml-group-link-list command line option, [569](#)
- gitlab-group-subgroup-list command line option, [570](#)
- gitlab-group-variable-list command line option, [572](#)
- gitlab-group-wiki-list command line option, [573](#)
- gitlab-hook-list command line option, [575](#)
- gitlab-issue-list command line option, [576](#)
- gitlab-ldap-group-list command line option, [578](#)
- gitlab-license-list command line option, [579](#)
- gitlab-merge-request-list command line option, [581](#)
- gitlab-namespace-list command line option, [582](#)
- gitlab-pages-domain-list command line option, [583](#)
- gitlab-personal-access-token-list command line option, [584](#)
- gitlab-project-access-request-list command line option, [604](#)

`gitlab-project-access-token-list`  
command line option, [605](#)

`gitlab-project-approval-rule-list`  
command line option, [608](#)

`gitlab-project-audit-event-list` command  
line option, [610](#)

`gitlab-project-badge-list` command line  
option, [612](#)

`gitlab-project-board-list` command line  
option, [614](#)

`gitlab-project-board-list-list` command  
line option, [616](#)

`gitlab-project-branch-list` command line  
option, [617](#)

`gitlab-project-cluster-list` command  
line option, [620](#)

`gitlab-project-commit-comment-list`  
command line option, [626](#)

`gitlab-project-commit-discussion-list`  
command line option, [627](#)

`gitlab-project-commit-list` command line  
option, [623](#)

`gitlab-project-commit-status-list`  
command line option, [630](#)

`gitlab-project-custom-attribute-list`  
command line option, [631](#)

`gitlab-project-deploy-token-list`  
command line option, [633](#)

`gitlab-project-deployment-list` command  
line option, [635](#)

`gitlab-project-deployment-merge-request-list`  
command line option, [637](#)

`gitlab-project-environment-list` command  
line option, [639](#)

`gitlab-project-event-list` command line  
option, [640](#)

`gitlab-project-fork-list` command line  
option, [645](#)

`gitlab-project-hook-list` command line  
option, [647](#)

`gitlab-project-integration-list` command  
line option, [650](#)

`gitlab-project-invitation-list` command  
line option, [652](#)

`gitlab-project-issue-award-emoji-list`  
command line option, [662](#)

`gitlab-project-issue-discussion-list`  
command line option, [663](#)

`gitlab-project-issue-link-list` command  
line option, [667](#)

`gitlab-project-issue-list` command line  
option, [656](#)

`gitlab-project-issue-note-award-emoji-list`  
command line option, [671](#)

`gitlab-project-issue-note-list` command  
line option, [668](#)

`gitlab-project-issue-resource-iteration-event-list`  
command line option, [672](#)

`gitlab-project-issue-resource-label-event-list`  
command line option, [673](#)

`gitlab-project-issue-resource-milestone-event-list`  
command line option, [674](#)

`gitlab-project-issue-resource-state-event-list`  
command line option, [675](#)

`gitlab-project-issue-resource-weight-event-list`  
command line option, [676](#)

`gitlab-project-job-list` command line  
option, [679](#)

`gitlab-project-key-list` command line  
option, [682](#)

`gitlab-project-label-list` command line  
option, [684](#)

`gitlab-project-list` command line option,  
[592](#)

`gitlab-project-member-all-list` command  
line option, [688](#)

`gitlab-project-member-list` command line  
option, [687](#)

`gitlab-project-merge-request-approval-rule-list`  
command line option, [702](#)

`gitlab-project-merge-request-award-emoji-list`  
command line option, [705](#)

`gitlab-project-merge-request-diff-list`  
command line option, [706](#)

`gitlab-project-merge-request-discussion-list`  
command line option, [707](#)

`gitlab-project-merge-request-draft-note-list`  
command line option, [712](#)

`gitlab-project-merge-request-list`  
command line option, [694](#)

`gitlab-project-merge-request-note-award-emoji-list`  
command line option, [716](#)

`gitlab-project-merge-request-note-list`  
command line option, [714](#)

`gitlab-project-merge-request-pipeline-list`  
command line option, [717](#)

`gitlab-project-merge-request-resource-label-event-list`  
command line option, [718](#)

`gitlab-project-merge-request-resource-milestone-event-`  
command line option, [719](#)

`gitlab-project-merge-request-resource-state-event-list`  
command line option, [720](#)

`gitlab-project-merge-request-reviewer-detail-list`  
command line option, [720](#)

`gitlab-project-merge-train-list` command  
line option, [721](#)

`gitlab-project-milestone-list` command  
line option, [723](#)

- gitlab-project-note-list command line option, [725](#)
- gitlab-project-package-file-list command line option, [728](#)
- gitlab-project-package-list command line option, [727](#)
- gitlab-project-package-pipeline-list command line option, [729](#)
- gitlab-project-pages-domain-list command line option, [730](#)
- gitlab-project-pipeline-bridge-list command line option, [734](#)
- gitlab-project-pipeline-job-list command line option, [734](#)
- gitlab-project-pipeline-list command line option, [733](#)
- gitlab-project-pipeline-schedule-list command line option, [736](#)
- gitlab-project-pipeline-schedule-pipeline-list command line option, [738](#)
- gitlab-project-pipeline-variable-list command line option, [741](#)
- gitlab-project-protected-branch-list command line option, [743](#)
- gitlab-project-protected-environment-list command line option, [745](#)
- gitlab-project-protected-tag-list command line option, [747](#)
- gitlab-project-registry-repository-list command line option, [750](#)
- gitlab-project-registry-tag-list command line option, [752](#)
- gitlab-project-release-link-list command line option, [756](#)
- gitlab-project-release-list command line option, [753](#)
- gitlab-project-remote-mirror-list command line option, [757](#)
- gitlab-project-resource-group-list command line option, [759](#)
- gitlab-project-resource-group-upcoming-job-list command line option, [760](#)
- gitlab-project-runner-list command line option, [761](#)
- gitlab-project-secure-file-list command line option, [763](#)
- gitlab-project-service-list command line option, [764](#)
- gitlab-project-snippet-award-emoji-list command line option, [769](#)
- gitlab-project-snippet-discussion-list command line option, [770](#)
- gitlab-project-snippet-list command line option, [766](#)
- gitlab-project-snippet-note-award-emoji-list command line option, [776](#)
- gitlab-project-snippet-note-list command line option, [774](#)
- gitlab-project-tag-list command line option, [778](#)
- gitlab-project-trigger-list command line option, [779](#)
- gitlab-project-user-list command line option, [780](#)
- gitlab-project-variable-list command line option, [782](#)
- gitlab-project-wiki-list command line option, [784](#)
- gitlab-runner-all-list command line option, [788](#)
- gitlab-runner-job-list command line option, [789](#)
- gitlab-runner-list command line option, [787](#)
- gitlab-shared-project-list command line option, [790](#)
- gitlab-snippet-list command line option, [792](#)
- gitlab-starred-project-list command line option, [794](#)
- gitlab-todo-list command line option, [795](#)
- gitlab-topic-list command line option, [797](#)
- gitlab-user-activities-list command line option, [804](#)
- gitlab-user-custom-attribute-list command line option, [805](#)
- gitlab-user-email-list command line option, [807](#)
- gitlab-user-event-list command line option, [807](#)
- gitlab-user-gpg-key-list command line option, [809](#)
- gitlab-user-impersonation-token-list command line option, [810](#)
- gitlab-user-key-list command line option, [812](#)
- gitlab-user-list command line option, [801](#)
- gitlab-user-membership-list command line option, [813](#)
- gitlab-user-project-list command line option, [816](#)
- gitlab-variable-list command line option, [817](#)
- pages-access-level
- gitlab-project-create command line option, [589](#)
- gitlab-project-update command line

- option, [601](#)
- pagination
  - gitlab command line option, [481](#)
- parent-id
  - gitlab-group-create command line option, [509](#)
  - gitlab-namespace-exists command line option, [581](#)
- password
  - gitlab-user-create command line option, [799](#)
  - gitlab-user-update command line option, [803](#)
- password-authentication-enabled-for-web
  - gitlab-application-settings-update command line option, [486](#)
- path
  - gitlab-generic-package-upload command line option, [503](#)
  - gitlab-group-create command line option, [508](#)
  - gitlab-group-update command line option, [512](#)
  - gitlab-project-commit-comment-create command line option, [625](#)
  - gitlab-project-commit-list command line option, [623](#)
  - gitlab-project-create command line option, [587](#)
  - gitlab-project-repository-tree command line option, [594](#)
  - gitlab-project-update command line option, [601](#)
- paused
  - gitlab-current-user-runner-create command line option, [498](#)
  - gitlab-runner-all-list command line option, [788](#)
  - gitlab-runner-list command line option, [787](#)
- per-page
  - gitlab command line option, [481](#)
  - gitlab-application-list command line option, [482](#)
  - gitlab-audit-event-list command line option, [488](#)
  - gitlab-broadcast-message-list command line option, [490](#)
  - gitlab-bulk-import-all-entity-list command line option, [492](#)
  - gitlab-bulk-import-entity-list command line option, [493](#)
  - gitlab-bulk-import-list command line option, [491](#)
  - gitlab-current-user-email-list command line option, [495](#)
  - gitlab-current-user-gpg-key-list command line option, [496](#)
  - gitlab-current-user-key-list command line option, [498](#)
  - gitlab-deploy-key-list command line option, [500](#)
  - gitlab-deploy-token-list command line option, [500](#)
  - gitlab-dockerfile-list command line option, [501](#)
  - gitlab-event-list command line option, [502](#)
  - gitlab-feature-list command line option, [502](#)
  - gitlab-geo-node-list command line option, [505](#)
  - gitlab-gitignore-list command line option, [506](#)
  - gitlab-gitlabciyaml-list command line option, [507](#)
  - gitlab-group-access-request-list command line option, [514](#)
  - gitlab-group-access-token-list command line option, [516](#)
  - gitlab-group-audit-event-list command line option, [517](#)
  - gitlab-group-badge-list command line option, [518](#)
  - gitlab-group-billable-member-list command line option, [520](#)
  - gitlab-group-billable-member-membership-list command line option, [521](#)
  - gitlab-group-board-list command line option, [522](#)
  - gitlab-group-board-list-list command line option, [524](#)
  - gitlab-group-cluster-list command line option, [526](#)
  - gitlab-group-custom-attribute-list command line option, [528](#)
  - gitlab-group-deploy-token-list command line option, [529](#)
  - gitlab-group-descendant-group-list command line option, [530](#)
  - gitlab-group-epic-award-emoji-list command line option, [534](#)
  - gitlab-group-epic-issue-list command line option, [537](#)
  - gitlab-group-epic-list command line option, [532](#)
  - gitlab-group-epic-note-award-emoji-list command line option, [541](#)



gitlab-group-epic-note-list command line option, 539

gitlab-group-epic-resource-label-event-list command line option, 542

gitlab-group-hook-list command line option, 545

gitlab-group-invitation-list command line option, 548

gitlab-group-issue-list command line option, 550

gitlab-group-iteration-list command line option, 551

gitlab-group-label-list command line option, 552

gitlab-group-ldap-group-link-list command line option, 555

gitlab-group-list command line option, 510

gitlab-group-member-all-list command line option, 557

gitlab-group-member-list command line option, 556

gitlab-group-merge-request-list command line option, 559

gitlab-group-milestone-list command line option, 561

gitlab-group-package-list command line option, 564

gitlab-group-project-list command line option, 565

gitlab-group-runner-list command line option, 568

gitlab-group-saml-group-link-list command line option, 569

gitlab-group-subgroup-list command line option, 570

gitlab-group-variable-list command line option, 572

gitlab-group-wiki-list command line option, 573

gitlab-hook-list command line option, 575

gitlab-issue-list command line option, 576

gitlab-ldap-group-list command line option, 578

gitlab-license-list command line option, 579

gitlab-merge-request-list command line option, 581

gitlab-namespace-list command line option, 582

gitlab-pages-domain-list command line option, 583

gitlab-personal-access-token-list command line option, 584

gitlab-project-access-request-list command line option, 604

gitlab-project-access-token-list command line option, 605

gitlab-project-approval-rule-list command line option, 608

gitlab-project-audit-event-list command line option, 610

gitlab-project-badge-list command line option, 612

gitlab-project-board-list command line option, 614

gitlab-project-board-list-list command line option, 616

gitlab-project-branch-list command line option, 617

gitlab-project-cluster-list command line option, 620

gitlab-project-commit-comment-list command line option, 626

gitlab-project-commit-discussion-list command line option, 627

gitlab-project-commit-list command line option, 623

gitlab-project-commit-status-list command line option, 630

gitlab-project-custom-attribute-list command line option, 631

gitlab-project-deploy-token-list command line option, 633

gitlab-project-deployment-list command line option, 635

gitlab-project-deployment-merge-request-list command line option, 637

gitlab-project-environment-list command line option, 639

gitlab-project-event-list command line option, 640

gitlab-project-fork-list command line option, 645

gitlab-project-hook-list command line option, 647

gitlab-project-integration-list command line option, 650

gitlab-project-invitation-list command line option, 652

gitlab-project-issue-award-emoji-list command line option, 662

gitlab-project-issue-discussion-list command line option, 663

gitlab-project-issue-link-list command line option, 667

gitlab-project-issue-list command line

- option, [656](#)
- gitlab-project-issue-note-award-emoji-list
  - command line option, [671](#)
- gitlab-project-issue-note-list command line option, [668](#)
- gitlab-project-issue-resource-iteration-event-list
  - command line option, [672](#)
- gitlab-project-issue-resource-label-event-list
  - command line option, [673](#)
- gitlab-project-issue-resource-milestone-event-list
  - command line option, [674](#)
- gitlab-project-issue-resource-state-event-list
  - command line option, [675](#)
- gitlab-project-issue-resource-weight-event-list
  - command line option, [676](#)
- gitlab-project-job-list command line option, [679](#)
- gitlab-project-key-list command line option, [682](#)
- gitlab-project-label-list command line option, [684](#)
- gitlab-project-list command line option, [592](#)
- gitlab-project-member-all-list command line option, [688](#)
- gitlab-project-member-list command line option, [687](#)
- gitlab-project-merge-request-approval-rule-list
  - command line option, [702](#)
- gitlab-project-merge-request-award-emoji-list
  - command line option, [705](#)
- gitlab-project-merge-request-diff-list
  - command line option, [706](#)
- gitlab-project-merge-request-discussion-list
  - command line option, [707](#)
- gitlab-project-merge-request-draft-note-list
  - command line option, [712](#)
- gitlab-project-merge-request-list
  - command line option, [694](#)
- gitlab-project-merge-request-note-award-emoji-list
  - command line option, [716](#)
- gitlab-project-merge-request-note-list
  - command line option, [714](#)
- gitlab-project-merge-request-pipeline-list
  - command line option, [717](#)
- gitlab-project-merge-request-resource-label-event-list
  - command line option, [718](#)
- gitlab-project-merge-request-resource-milestone-event-list
  - command line option, [719](#)
- gitlab-project-merge-request-resource-state-event-list
  - command line option, [720](#)
- gitlab-project-merge-request-reviewer-detail-list
  - command line option, [720](#)
- gitlab-project-merge-train-list command line option, [721](#)
- gitlab-project-milestone-list command line option, [723](#)
- gitlab-project-note-list command line option, [725](#)
- gitlab-project-package-file-list
  - command line option, [728](#)
- gitlab-project-package-list command line option, [727](#)
- gitlab-project-package-pipeline-list
  - command line option, [729](#)
- gitlab-project-pages-domain-list
  - command line option, [730](#)
- gitlab-project-pipeline-bridge-list
  - command line option, [734](#)
- gitlab-project-pipeline-job-list
  - command line option, [734](#)
- gitlab-project-pipeline-list command line option, [733](#)
- gitlab-project-pipeline-schedule-list
  - command line option, [736](#)
- gitlab-project-pipeline-schedule-pipeline-list
  - command line option, [738](#)
- gitlab-project-pipeline-variable-list
  - command line option, [741](#)
- gitlab-project-protected-branch-list
  - command line option, [743](#)
- gitlab-project-protected-environment-list
  - command line option, [745](#)
- gitlab-project-protected-tag-list
  - command line option, [747](#)
- gitlab-project-registry-repository-list
  - command line option, [750](#)
- gitlab-project-registry-tag-list
  - command line option, [752](#)
- gitlab-project-release-link-list
  - command line option, [756](#)
- gitlab-project-release-list command line option, [753](#)
- gitlab-project-remote-mirror-list
  - command line option, [757](#)
- gitlab-project-resource-group-list
  - command line option, [759](#)
- gitlab-project-resource-group-upcoming-job-list
  - command line option, [760](#)
- gitlab-project-runner-list command line option, [761](#)
- gitlab-project-secure-file-list command line option, [763](#)
- gitlab-project-service-list command line option, [764](#)
- gitlab-project-snippet-award-emoji-list
  - command line option, [769](#)
- gitlab-project-snippet-discussion-list



command line option, 770  
 gitlab-project-snippet-list command line option, 766  
 gitlab-project-snippet-note-award-emoji-list command line option, 776  
 gitlab-project-snippet-note-list command line option, 774  
 gitlab-project-tag-list command line option, 778  
 gitlab-project-trigger-list command line option, 779  
 gitlab-project-user-list command line option, 780  
 gitlab-project-variable-list command line option, 782  
 gitlab-project-wiki-list command line option, 784  
 gitlab-runner-all-list command line option, 788  
 gitlab-runner-job-list command line option, 789  
 gitlab-runner-list command line option, 787  
 gitlab-shared-project-list command line option, 790  
 gitlab-snippet-list command line option, 792  
 gitlab-starred-project-list command line option, 794  
 gitlab-todo-list command line option, 795  
 gitlab-topic-list command line option, 797  
 gitlab-user-activities-list command line option, 804  
 gitlab-user-custom-attribute-list command line option, 805  
 gitlab-user-email-list command line option, 807  
 gitlab-user-event-list command line option, 807  
 gitlab-user-gpg-key-list command line option, 809  
 gitlab-user-impersonation-token-list command line option, 810  
 gitlab-user-key-list command line option, 812  
 gitlab-user-list command line option, 802  
 gitlab-user-membership-list command line option, 813  
 gitlab-user-project-list command line option, 816  
 gitlab-variable-list command line option, 817  
 --performance-bar-allowed-group-id  
 gitlab-application-settings-update command line option, 487  
 --pipeline-events  
 gitlab-group-hook-create command line option, 544  
 gitlab-group-hook-update command line option, 546  
 gitlab-project-hook-create command line option, 646  
 gitlab-project-hook-update command line option, 648  
 --pipeline-id  
 gitlab-project-pipeline-bridge-list command line option, 734  
 gitlab-project-pipeline-job-list command line option, 734  
 gitlab-project-pipeline-test-report-get command line option, 740  
 gitlab-project-pipeline-test-report-summary-get command line option, 740  
 gitlab-project-pipeline-variable-list command line option, 741  
 --pipeline-schedule-id  
 gitlab-project-pipeline-schedule-pipeline-list command line option, 738  
 gitlab-project-pipeline-schedule-variable-create command line option, 739  
 gitlab-project-pipeline-schedule-variable-delete command line option, 739  
 gitlab-project-pipeline-schedule-variable-update command line option, 739  
 --plantuml-enabled  
 gitlab-application-settings-update command line option, 486  
 --plantuml-url  
 gitlab-application-settings-update command line option, 487  
 --platform-kubernetes-attributes  
 gitlab-group-cluster-create command line option, 525  
 gitlab-group-cluster-update command line option, 527  
 gitlab-project-cluster-create command line option, 619  
 gitlab-project-cluster-update command line option, 621  
 --polling-interval-multiplier  
 gitlab-application-settings-update command line option, 487  
 --popular  
 gitlab-license-list command line option, 579  
 --position  
 gitlab-group-board-list-update command

- line option, [524](#)
- gitlab-project-board-list-update
  - command line option, [616](#)
- gitlab-project-commit-discussion-note-create
  - command line option, [628](#)
- gitlab-project-merge-request-discussion-create
  - command line option, [707](#)
- gitlab-project-merge-request-draft-note-create
  - command line option, [711](#)
- gitlab-project-merge-request-draft-note-update
  - command line option, [712](#)
- prevent-forking-outside-group
  - gitlab-group-update command line option, [513](#)
- prevent-secrets
  - gitlab-group-push-rules-create command line option, [566](#)
  - gitlab-group-push-rules-update command line option, [567](#)
  - gitlab-project-push-rules-create
    - command line option, [748](#)
  - gitlab-project-push-rules-update
    - command line option, [749](#)
- printing-merge-request-link-enabled
  - gitlab-project-create command line option, [589](#)
- priority
  - gitlab-group-label-create command line option, [551](#)
  - gitlab-group-label-update command line option, [553](#)
  - gitlab-project-label-create command line option, [683](#)
  - gitlab-project-label-update command line option, [685](#)
- private-profile
  - gitlab-user-create command line option, [800](#)
  - gitlab-user-update command line option, [804](#)
- private-token
  - gitlab command line option, [481](#)
- process-mode
  - gitlab-project-resource-group-update
    - command line option, [759](#)
- project
  - gitlab-license-get command line option, [578](#)
- project-creation-level
  - gitlab-group-create command line option, [508](#)
  - gitlab-group-update command line option, [512](#)
- project-id
  - gitlab-current-user-runner-create
    - command line option, [498](#)
  - gitlab-generic-package-download command line option, [503](#)
  - gitlab-generic-package-upload command line option, [503](#)
  - gitlab-group-transfer-project command line option, [511](#)
  - gitlab-project-access-request-approve
    - command line option, [603](#)
  - gitlab-project-access-request-create
    - command line option, [603](#)
  - gitlab-project-access-request-delete
    - command line option, [603](#)
  - gitlab-project-access-request-list
    - command line option, [604](#)
  - gitlab-project-access-token-create
    - command line option, [604](#)
  - gitlab-project-access-token-delete
    - command line option, [604](#)
  - gitlab-project-access-token-get command line option, [605](#)
  - gitlab-project-access-token-list
    - command line option, [605](#)
  - gitlab-project-additional-statistics-get
    - command line option, [605](#)
  - gitlab-project-approval-get command line option, [606](#)
  - gitlab-project-approval-rule-create
    - command line option, [607](#)
  - gitlab-project-approval-rule-delete
    - command line option, [607](#)
  - gitlab-project-approval-rule-list
    - command line option, [608](#)
  - gitlab-project-approval-rule-update
    - command line option, [608](#)
  - gitlab-project-approval-update command line option, [606](#)
  - gitlab-project-artifact-delete command line option, [608](#)
  - gitlab-project-artifact-download
    - command line option, [609](#)
  - gitlab-project-artifact-raw command line option, [609](#)
  - gitlab-project-audit-event-get command line option, [610](#)
  - gitlab-project-audit-event-list command line option, [610](#)
  - gitlab-project-badge-create command line option, [611](#)
  - gitlab-project-badge-delete command line option, [611](#)
  - gitlab-project-badge-get command line option, [611](#)

gitlab-project-badge-list command line option, [612](#)  
gitlab-project-badge-render command line option, [612](#)  
gitlab-project-badge-update command line option, [612](#)  
gitlab-project-board-create command line option, [613](#)  
gitlab-project-board-delete command line option, [613](#)  
gitlab-project-board-get command line option, [613](#)  
gitlab-project-board-list command line option, [614](#)  
gitlab-project-board-list-create command line option, [615](#)  
gitlab-project-board-list-delete command line option, [615](#)  
gitlab-project-board-list-get command line option, [615](#)  
gitlab-project-board-list-list command line option, [616](#)  
gitlab-project-board-list-update command line option, [616](#)  
gitlab-project-board-update command line option, [614](#)  
gitlab-project-branch-create command line option, [616](#)  
gitlab-project-branch-delete command line option, [617](#)  
gitlab-project-branch-get command line option, [617](#)  
gitlab-project-branch-list command line option, [617](#)  
gitlab-project-ci-lint-create command line option, [618](#)  
gitlab-project-ci-lint-get command line option, [618](#)  
gitlab-project-ci-lint-validate command line option, [619](#)  
gitlab-project-cluster-create command line option, [619](#)  
gitlab-project-cluster-delete command line option, [620](#)  
gitlab-project-cluster-get command line option, [620](#)  
gitlab-project-cluster-list command line option, [620](#)  
gitlab-project-cluster-update command line option, [621](#)  
gitlab-project-commit-cherry-pick command line option, [622](#)  
gitlab-project-commit-comment-create command line option, [625](#)  
gitlab-project-commit-comment-list command line option, [626](#)  
gitlab-project-commit-create command line option, [622](#)  
gitlab-project-commit-diff command line option, [622](#)  
gitlab-project-commit-discussion-create command line option, [626](#)  
gitlab-project-commit-discussion-get command line option, [627](#)  
gitlab-project-commit-discussion-list command line option, [627](#)  
gitlab-project-commit-discussion-note-create command line option, [628](#)  
gitlab-project-commit-discussion-note-delete command line option, [628](#)  
gitlab-project-commit-discussion-note-get command line option, [629](#)  
gitlab-project-commit-discussion-note-update command line option, [629](#)  
gitlab-project-commit-get command line option, [623](#)  
gitlab-project-commit-list command line option, [623](#)  
gitlab-project-commit-merge-requests command line option, [624](#)  
gitlab-project-commit-refs command line option, [624](#)  
gitlab-project-commit-revert command line option, [624](#)  
gitlab-project-commit-signature command line option, [625](#)  
gitlab-project-commit-status-create command line option, [630](#)  
gitlab-project-commit-status-list command line option, [630](#)  
gitlab-project-custom-attribute-delete command line option, [631](#)  
gitlab-project-custom-attribute-get command line option, [631](#)  
gitlab-project-custom-attribute-list command line option, [631](#)  
gitlab-project-deploy-token-create command line option, [632](#)  
gitlab-project-deploy-token-delete command line option, [632](#)  
gitlab-project-deploy-token-get command line option, [632](#)  
gitlab-project-deploy-token-list command line option, [633](#)  
gitlab-project-deployment-approval command line option, [633](#)  
gitlab-project-deployment-create command line option, [634](#)

gitlab-project-deployment-get command line option, [634](#)  
gitlab-project-deployment-list command line option, [634](#)  
gitlab-project-deployment-merge-request-list command line option, [636](#)  
gitlab-project-deployment-update command line option, [635](#)  
gitlab-project-environment-create command line option, [638](#)  
gitlab-project-environment-delete command line option, [638](#)  
gitlab-project-environment-get command line option, [638](#)  
gitlab-project-environment-list command line option, [639](#)  
gitlab-project-environment-stop command line option, [639](#)  
gitlab-project-environment-update command line option, [639](#)  
gitlab-project-event-list command line option, [640](#)  
gitlab-project-export-create command line option, [641](#)  
gitlab-project-export-download command line option, [641](#)  
gitlab-project-export-get command line option, [641](#)  
gitlab-project-file-blame command line option, [642](#)  
gitlab-project-file-create command line option, [642](#)  
gitlab-project-file-delete command line option, [643](#)  
gitlab-project-file-get command line option, [643](#)  
gitlab-project-file-raw command line option, [643](#)  
gitlab-project-file-update command line option, [644](#)  
gitlab-project-fork-create command line option, [644](#)  
gitlab-project-fork-list command line option, [645](#)  
gitlab-project-hook-create command line option, [646](#)  
gitlab-project-hook-delete command line option, [647](#)  
gitlab-project-hook-get command line option, [647](#)  
gitlab-project-hook-list command line option, [647](#)  
gitlab-project-hook-update command line option, [648](#)  
gitlab-project-import-get command line option, [649](#)  
gitlab-project-integration-available command line option, [649](#)  
gitlab-project-integration-delete command line option, [649](#)  
gitlab-project-integration-get command line option, [650](#)  
gitlab-project-integration-list command line option, [650](#)  
gitlab-project-integration-update command line option, [650](#)  
gitlab-project-invitation-create command line option, [651](#)  
gitlab-project-invitation-delete command line option, [651](#)  
gitlab-project-invitation-get command line option, [652](#)  
gitlab-project-invitation-list command line option, [652](#)  
gitlab-project-invitation-update command line option, [652](#)  
gitlab-project-issue-add-spent-time command line option, [653](#)  
gitlab-project-issue-award-emoji-create command line option, [661](#)  
gitlab-project-issue-award-emoji-delete command line option, [661](#)  
gitlab-project-issue-award-emoji-get command line option, [662](#)  
gitlab-project-issue-award-emoji-list command line option, [662](#)  
gitlab-project-issue-closed-by command line option, [654](#)  
gitlab-project-issue-create command line option, [654](#)  
gitlab-project-issue-delete command line option, [655](#)  
gitlab-project-issue-discussion-create command line option, [663](#)  
gitlab-project-issue-discussion-get command line option, [663](#)  
gitlab-project-issue-discussion-list command line option, [663](#)  
gitlab-project-issue-discussion-note-create command line option, [664](#)  
gitlab-project-issue-discussion-note-delete command line option, [665](#)  
gitlab-project-issue-discussion-note-get command line option, [665](#)  
gitlab-project-issue-discussion-note-update command line option, [665](#)  
gitlab-project-issue-get command line option, [655](#)

- gitlab-project-issue-link-create  
command line option, 666
- gitlab-project-issue-link-delete  
command line option, 666
- gitlab-project-issue-link-list command  
line option, 667
- gitlab-project-issue-list command line  
option, 655
- gitlab-project-issue-move command line  
option, 656
- gitlab-project-issue-note-award-emoji-create  
command line option, 669
- gitlab-project-issue-note-award-emoji-delete  
command line option, 670
- gitlab-project-issue-note-award-emoji-get  
command line option, 670
- gitlab-project-issue-note-award-emoji-list  
command line option, 671
- gitlab-project-issue-note-create  
command line option, 667
- gitlab-project-issue-note-delete  
command line option, 668
- gitlab-project-issue-note-get command  
line option, 668
- gitlab-project-issue-note-list command  
line option, 668
- gitlab-project-issue-note-update  
command line option, 669
- gitlab-project-issue-participants  
command line option, 657
- gitlab-project-issue-related-merge-requests  
command line option, 657
- gitlab-project-issue-reorder command  
line option, 657
- gitlab-project-issue-reset-spent-time  
command line option, 658
- gitlab-project-issue-reset-time-estimate  
command line option, 658
- gitlab-project-issue-resource-iteration-event-get  
command line option, 671
- gitlab-project-issue-resource-iteration-event-list  
command line option, 672
- gitlab-project-issue-resource-label-event-get  
command line option, 672
- gitlab-project-issue-resource-label-event-list  
command line option, 673
- gitlab-project-issue-resource-milestone-event-get  
command line option, 673
- gitlab-project-issue-resource-milestone-event-list  
command line option, 674
- gitlab-project-issue-resource-state-event-get  
command line option, 674
- gitlab-project-issue-resource-state-event-list  
command line option, 674
- gitlab-project-issue-resource-weight-event-get  
command line option, 675
- gitlab-project-issue-resource-weight-event-list  
command line option, 675
- gitlab-project-issue-subscribe command  
line option, 658
- gitlab-project-issue-time-estimate  
command line option, 658
- gitlab-project-issue-time-stats command  
line option, 659
- gitlab-project-issue-todo command line  
option, 659
- gitlab-project-issue-unsubscribe  
command line option, 659
- gitlab-project-issue-update command  
line option, 660
- gitlab-project-issue-user-agent-detail  
command line option, 660
- gitlab-project-issues-statistics-get  
command line option, 676
- gitlab-project-job-artifact command  
line option, 677
- gitlab-project-job-artifacts command  
line option, 677
- gitlab-project-job-cancel command line  
option, 677
- gitlab-project-job-delete-artifacts  
command line option, 677
- gitlab-project-job-erase command line  
option, 678
- gitlab-project-job-get command line  
option, 678
- gitlab-project-job-keep-artifacts  
command line option, 678
- gitlab-project-job-list command line  
option, 679
- gitlab-project-job-play command line  
option, 679
- gitlab-project-job-retry command line  
option, 679
- gitlab-project-job-token-scope-get  
command line option, 680
- gitlab-project-job-token-scope-update  
command line option, 680
- gitlab-project-job-trace command line  
option, 680
- gitlab-project-key-create command line  
option, 681
- gitlab-project-key-delete command line  
option, 681
- gitlab-project-key-enable command line  
option, 681
- gitlab-project-key-get command line  
option, 682

`gitlab-project-key-list` command line option, [682](#)

`gitlab-project-key-update` command line option, [682](#)

`gitlab-project-label-create` command line option, [683](#)

`gitlab-project-label-delete` command line option, [683](#)

`gitlab-project-label-get` command line option, [684](#)

`gitlab-project-label-list` command line option, [684](#)

`gitlab-project-label-subscribe` command line option, [684](#)

`gitlab-project-label-unsubscribe` command line option, [685](#)

`gitlab-project-label-update` command line option, [685](#)

`gitlab-project-member-all-get` command line option, [688](#)

`gitlab-project-member-all-list` command line option, [688](#)

`gitlab-project-member-create` command line option, [686](#)

`gitlab-project-member-delete` command line option, [686](#)

`gitlab-project-member-get` command line option, [686](#)

`gitlab-project-member-list` command line option, [687](#)

`gitlab-project-member-update` command line option, [687](#)

`gitlab-project-merge-request-add-spent-time` command line option, [689](#)

`gitlab-project-merge-request-approval-get` command line option, [699](#)

`gitlab-project-merge-request-approval-rule-create` command line option, [700](#)

`gitlab-project-merge-request-approval-rule-delete` command line option, [701](#)

`gitlab-project-merge-request-approval-rule-get` command line option, [701](#)

`gitlab-project-merge-request-approval-rule-list` command line option, [702](#)

`gitlab-project-merge-request-approval-rule-update` command line option, [702](#)

`gitlab-project-merge-request-approval-state-get` command line option, [703](#)

`gitlab-project-merge-request-approval-update` command line option, [700](#)

`gitlab-project-merge-request-approve` command line option, [689](#)

`gitlab-project-merge-request-award-emoji-create` command line option, [704](#)

`gitlab-project-merge-request-award-emoji-delete` command line option, [704](#)

`gitlab-project-merge-request-award-emoji-get` command line option, [704](#)

`gitlab-project-merge-request-award-emoji-list` command line option, [705](#)

`gitlab-project-merge-request-cancel-merge-when-pipeline` command line option, [690](#)

`gitlab-project-merge-request-changes` command line option, [690](#)

`gitlab-project-merge-request-closes-issues` command line option, [690](#)

`gitlab-project-merge-request-commits` command line option, [691](#)

`gitlab-project-merge-request-create` command line option, [691](#)

`gitlab-project-merge-request-delete` command line option, [692](#)

`gitlab-project-merge-request-diff-get` command line option, [705](#)

`gitlab-project-merge-request-diff-list` command line option, [706](#)

`gitlab-project-merge-request-discussion-create` command line option, [706](#)

`gitlab-project-merge-request-discussion-get` command line option, [707](#)

`gitlab-project-merge-request-discussion-list` command line option, [707](#)

`gitlab-project-merge-request-discussion-note-create` command line option, [708](#)

`gitlab-project-merge-request-discussion-note-delete` command line option, [709](#)

`gitlab-project-merge-request-discussion-note-get` command line option, [709](#)

`gitlab-project-merge-request-discussion-note-update` command line option, [710](#)

`gitlab-project-merge-request-discussion-update` command line option, [708](#)

`gitlab-project-merge-request-draft-note-create` command line option, [711](#)

`gitlab-project-merge-request-draft-note-delete` command line option, [711](#)

`gitlab-project-merge-request-draft-note-get` command line option, [711](#)

`gitlab-project-merge-request-draft-note-list` command line option, [712](#)

`gitlab-project-merge-request-draft-note-update` command line option, [712](#)

`gitlab-project-merge-request-get` command line option, [692](#)

`gitlab-project-merge-request-list` command line option, [693](#)

`gitlab-project-merge-request-merge` command line option, [694](#)



- gitlab-project-merge-request-merge-ref  
command line option, 694
- gitlab-project-merge-request-note-award-emoji-create  
command line option, 715
- gitlab-project-merge-request-note-award-emoji-delete  
command line option, 715
- gitlab-project-merge-request-note-award-emoji-get  
command line option, 715
- gitlab-project-merge-request-note-award-emoji-list  
command line option, 716
- gitlab-project-merge-request-note-create  
command line option, 713
- gitlab-project-merge-request-note-delete  
command line option, 713
- gitlab-project-merge-request-note-get  
command line option, 713
- gitlab-project-merge-request-note-list  
command line option, 714
- gitlab-project-merge-request-note-update  
command line option, 714
- gitlab-project-merge-request-participants  
command line option, 695
- gitlab-project-merge-request-pipeline-create  
command line option, 716
- gitlab-project-merge-request-pipeline-list  
command line option, 717
- gitlab-project-merge-request-rebase  
command line option, 695
- gitlab-project-merge-request-reset-approvals  
command line option, 695
- gitlab-project-merge-request-reset-spent-time  
command line option, 696
- gitlab-project-merge-request-reset-time-estimate  
command line option, 696
- gitlab-project-merge-request-resource-label-event-get  
command line option, 717
- gitlab-project-merge-request-resource-label-event-list  
command line option, 718
- gitlab-project-merge-request-resource-milestone-get  
command line option, 718
- gitlab-project-merge-request-resource-milestone-list  
command line option, 719
- gitlab-project-merge-request-resource-state-event-get  
command line option, 719
- gitlab-project-merge-request-resource-state-event-list  
command line option, 720
- gitlab-project-merge-request-reviewer-detail-list  
command line option, 720
- gitlab-project-merge-request-subscribe  
command line option, 696
- gitlab-project-merge-request-time-estimate  
command line option, 697
- gitlab-project-merge-request-time-stats  
command line option, 697
- gitlab-project-merge-request-todo  
command line option, 697
- gitlab-project-merge-request-unapprove  
command line option, 698
- gitlab-project-merge-request-unsubscribe  
command line option, 698
- gitlab-project-merge-request-update  
command line option, 698
- gitlab-project-merge-train-list  
command line option, 721
- gitlab-project-milestone-create  
command line option, 722
- gitlab-project-milestone-delete  
command line option, 722
- gitlab-project-milestone-get  
command line option, 722
- gitlab-project-milestone-issues  
command line option, 723
- gitlab-project-milestone-list  
command line option, 723
- gitlab-project-milestone-merge-requests  
command line option, 723
- gitlab-project-milestone-update  
command line option, 724
- gitlab-project-note-get  
command line option, 724
- gitlab-project-note-list  
command line option, 725
- gitlab-project-notification-settings-get  
command line option, 725
- gitlab-project-notification-settings-update  
command line option, 726
- gitlab-project-package-delete  
command line option, 727
- gitlab-project-package-file-delete  
command line option, 728
- gitlab-project-package-file-list  
command line option, 728
- gitlab-project-package-get  
command line option, 727
- gitlab-project-package-list  
command line option, 727
- gitlab-project-package-pipeline-list  
command line option, 729
- gitlab-project-pages-domain-create  
command line option, 729
- gitlab-project-pages-domain-delete  
command line option, 730
- gitlab-project-pages-domain-get  
command line option, 730
- gitlab-project-pages-domain-list  
command line option, 730
- gitlab-project-pages-domain-update  
command line option, 731

<code>gitlab-project-pipeline-bridge-list</code> command line option, <a href="#">733</a>	<code>gitlab-project-protected-environment-create</code> command line option, <a href="#">744</a>
<code>gitlab-project-pipeline-cancel</code> command line option, <a href="#">731</a>	<code>gitlab-project-protected-environment-delete</code> command line option, <a href="#">744</a>
<code>gitlab-project-pipeline-create</code> command line option, <a href="#">731</a>	<code>gitlab-project-protected-environment-get</code> command line option, <a href="#">745</a>
<code>gitlab-project-pipeline-delete</code> command line option, <a href="#">732</a>	<code>gitlab-project-protected-environment-list</code> command line option, <a href="#">745</a>
<code>gitlab-project-pipeline-get</code> command line option, <a href="#">732</a>	<code>gitlab-project-protected-tag-create</code> command line option, <a href="#">746</a>
<code>gitlab-project-pipeline-job-list</code> command line option, <a href="#">734</a>	<code>gitlab-project-protected-tag-delete</code> command line option, <a href="#">746</a>
<code>gitlab-project-pipeline-list</code> command line option, <a href="#">732</a>	<code>gitlab-project-protected-tag-get</code> command line option, <a href="#">746</a>
<code>gitlab-project-pipeline-retry</code> command line option, <a href="#">733</a>	<code>gitlab-project-protected-tag-list</code> command line option, <a href="#">747</a>
<code>gitlab-project-pipeline-schedule-create</code> command line option, <a href="#">735</a>	<code>gitlab-project-push-rules-create</code> command line option, <a href="#">747</a>
<code>gitlab-project-pipeline-schedule-delete</code> command line option, <a href="#">735</a>	<code>gitlab-project-push-rules-delete</code> command line option, <a href="#">748</a>
<code>gitlab-project-pipeline-schedule-get</code> command line option, <a href="#">736</a>	<code>gitlab-project-push-rules-get</code> command line option, <a href="#">748</a>
<code>gitlab-project-pipeline-schedule-list</code> command line option, <a href="#">736</a>	<code>gitlab-project-push-rules-update</code> command line option, <a href="#">749</a>
<code>gitlab-project-pipeline-schedule-pipeline-list</code> command line option, <a href="#">738</a>	<code>gitlab-project-registry-repository-delete</code> command line option, <a href="#">750</a>
<code>gitlab-project-pipeline-schedule-play</code> command line option, <a href="#">736</a>	<code>gitlab-project-registry-repository-list</code> command line option, <a href="#">750</a>
<code>gitlab-project-pipeline-schedule-take-ownership</code> command line option, <a href="#">737</a>	<code>gitlab-project-registry-tag-delete</code> command line option, <a href="#">751</a>
<code>gitlab-project-pipeline-schedule-update</code> command line option, <a href="#">737</a>	<code>gitlab-project-registry-tag-delete-in-bulk</code> command line option, <a href="#">751</a>
<code>gitlab-project-pipeline-schedule-variable-create</code> command line option, <a href="#">738</a>	<code>gitlab-project-registry-tag-get</code> command line option, <a href="#">751</a>
<code>gitlab-project-pipeline-schedule-variable-delete</code> command line option, <a href="#">739</a>	<code>gitlab-project-registry-tag-list</code> command line option, <a href="#">752</a>
<code>gitlab-project-pipeline-schedule-variable-update</code> command line option, <a href="#">739</a>	<code>gitlab-project-release-create</code> command line option, <a href="#">752</a>
<code>gitlab-project-pipeline-test-report-get</code> command line option, <a href="#">740</a>	<code>gitlab-project-release-delete</code> command line option, <a href="#">753</a>
<code>gitlab-project-pipeline-test-report-summary-get</code> command line option, <a href="#">740</a>	<code>gitlab-project-release-get</code> command line option, <a href="#">753</a>
<code>gitlab-project-pipeline-variable-list</code> command line option, <a href="#">741</a>	<code>gitlab-project-release-link-create</code> command line option, <a href="#">754</a>
<code>gitlab-project-protected-branch-create</code> command line option, <a href="#">742</a>	<code>gitlab-project-release-link-delete</code> command line option, <a href="#">755</a>
<code>gitlab-project-protected-branch-delete</code> command line option, <a href="#">742</a>	<code>gitlab-project-release-link-get</code> command line option, <a href="#">755</a>
<code>gitlab-project-protected-branch-get</code> command line option, <a href="#">743</a>	<code>gitlab-project-release-link-list</code> command line option, <a href="#">755</a>
<code>gitlab-project-protected-branch-list</code> command line option, <a href="#">743</a>	<code>gitlab-project-release-link-update</code> command line option, <a href="#">756</a>
<code>gitlab-project-protected-branch-update</code> command line option, <a href="#">743</a>	<code>gitlab-project-release-list</code> command line option, <a href="#">753</a>



gitlab-project-release-update command line option, [754](#)  
gitlab-project-remote-mirror-create command line option, [757](#)  
gitlab-project-remote-mirror-delete command line option, [757](#)  
gitlab-project-remote-mirror-list command line option, [757](#)  
gitlab-project-remote-mirror-update command line option, [758](#)  
gitlab-project-resource-group-get command line option, [758](#)  
gitlab-project-resource-group-list command line option, [759](#)  
gitlab-project-resource-group-upcoming-job-list command line option, [760](#)  
gitlab-project-resource-group-update command line option, [759](#)  
gitlab-project-runner-create command line option, [760](#)  
gitlab-project-runner-delete command line option, [760](#)  
gitlab-project-runner-list command line option, [761](#)  
gitlab-project-secure-file-create command line option, [761](#)  
gitlab-project-secure-file-delete command line option, [762](#)  
gitlab-project-secure-file-download command line option, [762](#)  
gitlab-project-secure-file-get command line option, [762](#)  
gitlab-project-secure-file-list command line option, [762](#)  
gitlab-project-service-available command line option, [763](#)  
gitlab-project-service-delete command line option, [763](#)  
gitlab-project-service-get command line option, [764](#)  
gitlab-project-service-list command line option, [764](#)  
gitlab-project-service-update command line option, [764](#)  
gitlab-project-snippet-award-emoji-create command line option, [768](#)  
gitlab-project-snippet-award-emoji-delete command line option, [768](#)  
gitlab-project-snippet-award-emoji-get command line option, [768](#)  
gitlab-project-snippet-award-emoji-list command line option, [769](#)  
gitlab-project-snippet-content command line option, [765](#)  
gitlab-project-snippet-create command line option, [765](#)  
gitlab-project-snippet-delete command line option, [766](#)  
gitlab-project-snippet-discussion-create command line option, [769](#)  
gitlab-project-snippet-discussion-get command line option, [770](#)  
gitlab-project-snippet-discussion-list command line option, [770](#)  
gitlab-project-snippet-discussion-note-create command line option, [771](#)  
gitlab-project-snippet-discussion-note-delete command line option, [771](#)  
gitlab-project-snippet-discussion-note-get command line option, [772](#)  
gitlab-project-snippet-discussion-note-update command line option, [772](#)  
gitlab-project-snippet-get command line option, [766](#)  
gitlab-project-snippet-list command line option, [766](#)  
gitlab-project-snippet-note-award-emoji-create command line option, [775](#)  
gitlab-project-snippet-note-award-emoji-delete command line option, [775](#)  
gitlab-project-snippet-note-award-emoji-get command line option, [775](#)  
gitlab-project-snippet-note-award-emoji-list command line option, [776](#)  
gitlab-project-snippet-note-create command line option, [773](#)  
gitlab-project-snippet-note-delete command line option, [773](#)  
gitlab-project-snippet-note-get command line option, [773](#)  
gitlab-project-snippet-note-list command line option, [774](#)  
gitlab-project-snippet-note-update command line option, [774](#)  
gitlab-project-snippet-update command line option, [767](#)  
gitlab-project-snippet-user-agent-detail command line option, [767](#)  
gitlab-project-storage-get command line option, [776](#)  
gitlab-project-tag-create command line option, [777](#)  
gitlab-project-tag-delete command line option, [777](#)  
gitlab-project-tag-get command line option, [777](#)  
gitlab-project-tag-list command line option, [778](#)

- gitlab-project-trigger-create command line option, [778](#)
- gitlab-project-trigger-delete command line option, [778](#)
- gitlab-project-trigger-get command line option, [779](#)
- gitlab-project-trigger-list command line option, [779](#)
- gitlab-project-trigger-update command line option, [779](#)
- gitlab-project-user-list command line option, [780](#)
- gitlab-project-variable-create command line option, [781](#)
- gitlab-project-variable-delete command line option, [781](#)
- gitlab-project-variable-get command line option, [781](#)
- gitlab-project-variable-list command line option, [782](#)
- gitlab-project-variable-update command line option, [782](#)
- gitlab-project-wiki-create command line option, [783](#)
- gitlab-project-wiki-delete command line option, [783](#)
- gitlab-project-wiki-get command line option, [783](#)
- gitlab-project-wiki-list command line option, [784](#)
- gitlab-project-wiki-update command line option, [784](#)
- gitlab-project-wiki-upload command line option, [784](#)
- gitlab-todo-list command line option, [795](#)
- projects-limit
  - gitlab-user-create command line option, [799](#)
  - gitlab-user-update command line option, [803](#)
- protected
  - gitlab-group-variable-create command line option, [571](#)
  - gitlab-group-variable-update command line option, [572](#)
  - gitlab-project-variable-create command line option, [781](#)
  - gitlab-project-variable-update command line option, [782](#)
  - gitlab-variable-create command line option, [817](#)
  - gitlab-variable-update command line option, [818](#)
- protected-branch-ids
  - gitlab-project-approval-rule-create command line option, [607](#)
- provider
  - gitlab-group-ldap-group-link-create command line option, [554](#)
  - gitlab-ldap-group-list command line option, [578](#)
  - gitlab-user-create command line option, [799](#)
  - gitlab-user-list command line option, [801](#)
  - gitlab-user-update command line option, [803](#)
- public
  - gitlab-user-project-create command line option, [814](#)
- public-builds
  - gitlab-project-create command line option, [589](#)
  - gitlab-project-update command line option, [601](#)
  - gitlab-user-project-create command line option, [814](#)
- public-email
  - gitlab-user-create command line option, [800](#)
  - gitlab-user-update command line option, [804](#)
- push-access-level
  - gitlab-project-protected-branch-create command line option, [742](#)
- push-events
  - gitlab-group-hook-create command line option, [543](#)
  - gitlab-group-hook-update command line option, [545](#)
  - gitlab-project-hook-create command line option, [646](#)
  - gitlab-project-hook-update command line option, [648](#)
- query
  - gitlab-group-invitation-list command line option, [548](#)
  - gitlab-project-invitation-list command line option, [652](#)
- reassign-issue
  - gitlab-group-notification-settings-update command line option, [563](#)
  - gitlab-notification-settings-update command line option, [583](#)
  - gitlab-project-notification-settings-update command line option, [726](#)
- reassign-merge-request
  - gitlab-group-notification-settings-update command line option, [563](#)

```

gitlab-notification-settings-update
    command line option, 583
gitlab-project-notification-settings-update
    command line option, 726
--recursive
    gitlab-project-repository-tree command
        line option, 594
--redirect-uri
    gitlab-application-create command line
        option, 481
--ref
    gitlab-project-branch-create command
        line option, 616
    gitlab-project-ci-lint-create command
        line option, 618
    gitlab-project-ci-lint-get command line
        option, 618
    gitlab-project-ci-lint-validate command
        line option, 619
    gitlab-project-commit-status-create
        command line option, 630
    gitlab-project-deployment-create
        command line option, 634
    gitlab-project-file-blame command line
        option, 642
    gitlab-project-file-get command line
        option, 643
    gitlab-project-file-raw command line
        option, 643
    gitlab-project-pipeline-create command
        line option, 731
    gitlab-project-pipeline-list command
        line option, 732
    gitlab-project-pipeline-schedule-create
        command line option, 735
    gitlab-project-pipeline-schedule-update
        command line option, 737
    gitlab-project-release-create command
        line option, 752
    gitlab-project-repository-tree command
        line option, 594
    gitlab-project-tag-create command line
        option, 777
    gitlab-project-trigger-pipeline command
        line option, 596
--ref-name
    gitlab-project-artifact-delete command
        line option, 608
    gitlab-project-artifact-download
        command line option, 609
    gitlab-project-artifact-raw command
        line option, 609
    gitlab-project-commit-list command line
        option, 623
--refs
    gitlab-project-repository-merge-base
        command line option, 594
--reject-unsigned-commits
    gitlab-group-push-rules-create command
        line option, 566
    gitlab-group-push-rules-update command
        line option, 567
    gitlab-project-push-rules-create
        command line option, 748
    gitlab-project-push-rules-update
        command line option, 749
--released-at
    gitlab-project-release-update command
        line option, 754
--releases-access-level
    gitlab-project-create command line
        option, 589
    gitlab-project-update command line
        option, 601
--releases-events
    gitlab-group-hook-create command line
        option, 544
    gitlab-group-hook-update command line
        option, 546
--remove-source-branch
    gitlab-project-merge-request-create
        command line option, 692
    gitlab-project-merge-request-update
        command line option, 699
--remove-source-branch-after-merge
    gitlab-project-create command line
        option, 589
    gitlab-project-update command line
        option, 601
--render-html
    gitlab-project-merge-request-get
        command line option, 692
--reopen-issue
    gitlab-group-notification-settings-update
        command line option, 563
    gitlab-notification-settings-update
        command line option, 583
    gitlab-project-notification-settings-update
        command line option, 726
--reopen-merge-request
    gitlab-group-notification-settings-update
        command line option, 563
    gitlab-notification-settings-update
        command line option, 583
    gitlab-project-notification-settings-update
        command line option, 726
--repos-max-capacity
    gitlab-geo-node-update command line

```

- option, [506](#)
- repository-access-level
  - gitlab-project-create command line option, [589](#)
  - gitlab-project-update command line option, [601](#)
- repository-checksum-failed
  - gitlab-project-list command line option, [592](#)
  - gitlab-user-project-list command line option, [815](#)
- repository-id
  - gitlab-project-registry-tag-delete command line option, [751](#)
  - gitlab-project-registry-tag-delete-in-bulk command line option, [751](#)
  - gitlab-project-registry-tag-get command line option, [751](#)
  - gitlab-project-registry-tag-list command line option, [752](#)
- repository-storage
  - gitlab-project-create command line option, [589](#)
  - gitlab-project-list command line option, [592](#)
  - gitlab-project-update command line option, [601](#)
- repository-storages
  - gitlab-application-settings-update command line option, [486](#)
- represented-as
  - gitlab-project-deployment-approval command line option, [633](#)
- request-access-enabled
  - gitlab-group-create command line option, [509](#)
  - gitlab-group-update command line option, [513](#)
  - gitlab-project-create command line option, [589](#)
  - gitlab-project-update command line option, [601](#)
- require-two-factor-authentication
  - gitlab-group-create command line option, [508](#)
  - gitlab-group-update command line option, [512](#)
- required-approval-count
  - gitlab-project-protected-environment-create command line option, [744](#)
- requirements-access-level
  - gitlab-project-create command line option, [589](#)
  - gitlab-project-update command line option, [601](#)
- reset-approvals-on-push
  - gitlab-project-approval-update command line option, [606](#)
- reset-password
  - gitlab-user-create command line option, [799](#)
- resolve-discussion
  - gitlab-project-merge-request-draft-note-create command line option, [711](#)
- resolve-outdated-diff-discussions
  - gitlab-project-create command line option, [589](#)
  - gitlab-project-update command line option, [601](#)
- resolved
  - gitlab-project-merge-request-discussion-update command line option, [708](#)
- resource-group-key
  - gitlab-project-resource-group-upcoming-job-list command line option, [760](#)
- restrict-user-defined-variables
  - gitlab-project-update command line option, [601](#)
- restricted-visibility-levels
  - gitlab-application-settings-update command line option, [486](#)
- reviewer-id
  - gitlab-merge-request-list command line option, [580](#)
  - gitlab-project-deployment-merge-request-list command line option, [637](#)
- reviewer-ids
  - gitlab-project-merge-request-create command line option, [692](#)
  - gitlab-project-merge-request-update command line option, [699](#)
- reviewer-username
  - gitlab-merge-request-list command line option, [580](#)
  - gitlab-project-deployment-merge-request-list command line option, [637](#)
- rsa-key-restriction
  - gitlab-application-settings-update command line option, [487](#)
- run-untagged
  - gitlab-current-user-runner-create command line option, [498](#)
  - gitlab-runner-create command line option, [786](#)
  - gitlab-runner-update command line option, [787](#)
- runner-id
  - gitlab-project-runner-create command line option, [601](#)

line option, [760](#)  
 gitlab-runner-job-list command line option, [789](#)  
 --runner-type  
   gitlab-current-user-runner-create command line option, [498](#)  
 --saml-group-name  
   gitlab-group-saml-group-link-create command line option, [568](#)  
 --scope  
   gitlab-event-list command line option, [502](#)  
   gitlab-group-merge-request-list command line option, [558](#)  
   gitlab-group-runner-list command line option, [568](#)  
   gitlab-group-search command line option, [510](#)  
   gitlab-issue-list command line option, [576](#)  
   gitlab-merge-request-list command line option, [580](#)  
   gitlab-project-deployment-merge-request-list command line option, [637](#)  
   gitlab-project-event-list command line option, [640](#)  
   gitlab-project-issue-list command line option, [656](#)  
   gitlab-project-job-list command line option, [679](#)  
   gitlab-project-merge-request-list command line option, [693](#)  
   gitlab-project-merge-train-list command line option, [721](#)  
   gitlab-project-pipeline-bridge-list command line option, [734](#)  
   gitlab-project-pipeline-job-list command line option, [734](#)  
   gitlab-project-pipeline-list command line option, [732](#)  
   gitlab-project-runner-list command line option, [761](#)  
   gitlab-project-search command line option, [595](#)  
   gitlab-runner-all command line option, [785](#)  
   gitlab-runner-all-list command line option, [788](#)  
   gitlab-runner-list command line option, [787](#)  
   gitlab-user-event-list command line option, [807](#)  
 --scopes  
   gitlab-application-create command line option, [482](#)  
 gitlab-group-access-token-create command line option, [515](#)  
 gitlab-group-deploy-token-create command line option, [528](#)  
 gitlab-group-deploy-token-list command line option, [529](#)  
 gitlab-project-access-token-create command line option, [604](#)  
 gitlab-project-deploy-token-create command line option, [632](#)  
 gitlab-project-deploy-token-list command line option, [633](#)  
 gitlab-user-impersonation-token-create command line option, [809](#)  
 gitlab-user-personal-access-token-create command line option, [813](#)  
 --search  
   gitlab-group-billable-member-list command line option, [520](#)  
   gitlab-group-descendant-group-list command line option, [530](#)  
   gitlab-group-epic-list command line option, [532](#)  
   gitlab-group-issue-list command line option, [549](#)  
   gitlab-group-iteration-list command line option, [551](#)  
   gitlab-group-list command line option, [510](#)  
   gitlab-group-merge-request-list command line option, [559](#)  
   gitlab-group-milestone-list command line option, [561](#)  
   gitlab-group-project-list command line option, [564](#)  
   gitlab-group-search command line option, [510](#)  
   gitlab-group-subgroup-list command line option, [570](#)  
   gitlab-issue-list command line option, [576](#)  
   gitlab-ldap-group-list command line option, [578](#)  
   gitlab-merge-request-list command line option, [580](#)  
   gitlab-namespace-list command line option, [582](#)  
   gitlab-project-deployment-merge-request-list command line option, [637](#)  
   gitlab-project-environment-list command line option, [639](#)  
   gitlab-project-fork-list command line option, [645](#)

- gitlab-project-issue-list command line option, [656](#)
- gitlab-project-list command line option, [592](#)
- gitlab-project-merge-request-list command line option, [694](#)
- gitlab-project-milestone-list command line option, [723](#)
- gitlab-project-search command line option, [595](#)
- gitlab-project-user-list command line option, [780](#)
- gitlab-shared-project-list command line option, [790](#)
- gitlab-starred-project-list command line option, [794](#)
- gitlab-user-list command line option, [801](#)
- gitlab-user-project-list command line option, [815](#)
- search-namespaces
  - gitlab-project-list command line option, [592](#)
- security-and-compliance-access-level
  - gitlab-project-create command line option, [589](#)
  - gitlab-project-update command line option, [601](#)
- server-url
  - gitlab command line option, [480](#)
- service-desk-enabled
  - gitlab-project-update command line option, [601](#)
- session-expire-delay
  - gitlab-application-settings-update command line option, [486](#)
- sha
  - gitlab-project-deployment-create command line option, [634](#)
  - gitlab-project-merge-request-approve command line option, [689](#)
  - gitlab-project-pipeline-list command line option, [732](#)
  - gitlab-project-repository-archive command line option, [593](#)
  - gitlab-project-repository-blob command line option, [593](#)
  - gitlab-project-repository-raw-blob command line option, [594](#)
- share-with-group-lock
  - gitlab-group-create command line option, [508](#)
  - gitlab-group-update command line option, [512](#)
- shared-runners-enabled
  - gitlab-project-create command line option, [589](#)
  - gitlab-project-update command line option, [601](#)
- shared-runners-minutes-limit
  - gitlab-group-create command line option, [509](#)
  - gitlab-group-update command line option, [513](#)
- shared-runners-setting
  - gitlab-group-update command line option, [513](#)
- should-remove-source-branch
  - gitlab-project-merge-request-merge command line option, [694](#)
- show-default-award-emojis
  - gitlab-project-create command line option, [589](#)
  - gitlab-project-update command line option, [601](#)
- sign-in-text
  - gitlab-application-settings-update command line option, [486](#)
- signup-enabled
  - gitlab-application-settings-update command line option, [485](#)
- simple
  - gitlab-group-project-list command line option, [564](#)
  - gitlab-project-fork-list command line option, [645](#)
  - gitlab-project-list command line option, [592](#)
  - gitlab-shared-project-list command line option, [790](#)
  - gitlab-starred-project-list command line option, [794](#)
  - gitlab-user-project-list command line option, [815](#)
- since
  - gitlab-project-commit-list command line option, [623](#)
- skip-confirmation
  - gitlab-user-create command line option, [800](#)
- skip-groups
  - gitlab-group-descendant-group-list command line option, [530](#)
  - gitlab-group-list command line option, [510](#)
  - gitlab-group-subgroup-list command line option, [570](#)
- skip-login
  - gitlab command line option, [481](#)

---

```

--skip-reconfirmation
    gitlab-user-update command line option,
        803
--skip-users
    gitlab-project-user-list command line
        option, 780
--skype
    gitlab-user-create command line option,
        799
    gitlab-user-update command line option,
        803
--slug
    gitlab-group-wiki-delete command line
        option, 573
    gitlab-group-wiki-get command line
        option, 573
    gitlab-group-wiki-update command line
        option, 574
    gitlab-project-integration-available
        command line option, 649
    gitlab-project-integration-delete
        command line option, 649
    gitlab-project-integration-get command
        line option, 650
    gitlab-project-integration-update
        command line option, 650
    gitlab-project-service-available
        command line option, 763
    gitlab-project-service-delete command
        line option, 763
    gitlab-project-service-get command line
        option, 764
    gitlab-project-service-update command
        line option, 764
    gitlab-project-wiki-delete command line
        option, 783
    gitlab-project-wiki-get command line
        option, 783
    gitlab-project-wiki-update command line
        option, 784
    gitlab-project-wiki-upload command line
        option, 785
--snippet-id
    gitlab-project-snippet-award-emoji-create
        command line option, 768
    gitlab-project-snippet-award-emoji-delete
        command line option, 768
    gitlab-project-snippet-award-emoji-get
        command line option, 768
    gitlab-project-snippet-award-emoji-list
        command line option, 769
    gitlab-project-snippet-discussion-create
        command line option, 769
    gitlab-project-snippet-discussion-get
        command line option, 770
    gitlab-project-snippet-discussion-list
        command line option, 770
    gitlab-project-snippet-discussion-note-create
        command line option, 771
    gitlab-project-snippet-discussion-note-delete
        command line option, 771
    gitlab-project-snippet-discussion-note-get
        command line option, 772
    gitlab-project-snippet-discussion-note-update
        command line option, 772
    gitlab-project-snippet-note-award-emoji-create
        command line option, 775
    gitlab-project-snippet-note-award-emoji-delete
        command line option, 775
    gitlab-project-snippet-note-award-emoji-get
        command line option, 775
    gitlab-project-snippet-note-award-emoji-list
        command line option, 776
    gitlab-project-snippet-note-create
        command line option, 773
    gitlab-project-snippet-note-delete
        command line option, 773
    gitlab-project-snippet-note-get command
        line option, 773
    gitlab-project-snippet-note-list
        command line option, 774
    gitlab-project-snippet-note-update
        command line option, 774
--snippets-access-level
    gitlab-project-create command line
        option, 589
    gitlab-project-update command line
        option, 601
--snippets-enabled
    gitlab-project-create command line
        option, 589
    gitlab-project-update command line
        option, 601
    gitlab-user-project-create command line
        option, 814
--sort
    gitlab-bulk-import-all-entity-list
        command line option, 492
    gitlab-bulk-import-entity-list command
        line option, 493
    gitlab-bulk-import-list command line
        option, 491
    gitlab-event-list command line option,
        502
    gitlab-group-billable-member-list
        command line option, 520
    gitlab-group-descendant-group-list
        command line option, 530

```



- gitlab-group-epic-list command line option, [532](#)
- gitlab-group-issue-list command line option, [549](#)
- gitlab-group-list command line option, [510](#)
- gitlab-group-merge-request-list command line option, [558](#)
- gitlab-group-package-list command line option, [563](#)
- gitlab-group-project-list command line option, [564](#)
- gitlab-group-subgroup-list command line option, [570](#)
- gitlab-issue-list command line option, [576](#)
- gitlab-merge-request-list command line option, [580](#)
- gitlab-project-deployment-list command line option, [635](#)
- gitlab-project-deployment-merge-request-list command line option, [636](#)
- gitlab-project-event-list command line option, [640](#)
- gitlab-project-fork-list command line option, [645](#)
- gitlab-project-issue-list command line option, [656](#)
- gitlab-project-list command line option, [592](#)
- gitlab-project-merge-request-list command line option, [693](#)
- gitlab-project-package-list command line option, [727](#)
- gitlab-project-pipeline-list command line option, [733](#)
- gitlab-project-release-list command line option, [753](#)
- gitlab-project-resource-group-list command line option, [759](#)
- gitlab-shared-project-list command line option, [790](#)
- gitlab-starred-project-list command line option, [794](#)
- gitlab-user-event-list command line option, [807](#)
- gitlab-user-project-list command line option, [815](#)
- source
  - gitlab-project-pipeline-list command line option, [732](#)
- source-branch
  - gitlab-group-merge-request-list command line option, [559](#)
  - gitlab-merge-request-list command line option, [580](#)
  - gitlab-project-deployment-merge-request-list command line option, [637](#)
  - gitlab-project-merge-request-create command line option, [691](#)
  - gitlab-project-merge-request-list command line option, [694](#)
- source-id
  - gitlab-user-membership-get command line option, [812](#)
- source-topic-id
  - gitlab-topic-merge command line option, [797](#)
- squash
  - gitlab-project-merge-request-create command line option, [692](#)
  - gitlab-project-merge-request-update command line option, [699](#)
- squash-commit-template
  - gitlab-project-update command line option, [601](#)
- squash-option
  - gitlab-project-create command line option, [589](#)
  - gitlab-project-update command line option, [601](#)
  - gitlab-user-project-create command line option, [814](#)
- ssl-verify
  - gitlab command line option, [480](#)
- starred
  - gitlab-group-project-list command line option, [565](#)
  - gitlab-project-fork-list command line option, [645](#)
  - gitlab-project-list command line option, [592](#)
  - gitlab-shared-project-list command line option, [790](#)
  - gitlab-starred-project-list command line option, [794](#)
  - gitlab-user-project-list command line option, [815](#)
- start-date
  - gitlab-group-epic-create command line option, [531](#)
  - gitlab-group-epic-update command line option, [532](#)
  - gitlab-group-milestone-create command line option, [559](#)
  - gitlab-group-milestone-update command line option, [561](#)
  - gitlab-project-milestone-create command



- line option, [722](#)
  - gitlab-project-milestone-update command line option, [724](#)
- starts-at
  - gitlab-broadcast-message-create command line option, [489](#)
  - gitlab-broadcast-message-update command line option, [490](#)
- state
  - gitlab-group-issue-list command line option, [549](#)
  - gitlab-group-iteration-list command line option, [551](#)
  - gitlab-group-merge-request-list command line option, [558](#)
  - gitlab-group-milestone-list command line option, [561](#)
  - gitlab-issue-list command line option, [576](#)
  - gitlab-merge-request-list command line option, [580](#)
  - gitlab-project-commit-status-create command line option, [630](#)
  - gitlab-project-deployment-merge-request-list command line option, [636](#)
  - gitlab-project-issue-list command line option, [656](#)
  - gitlab-project-merge-request-list command line option, [693](#)
  - gitlab-project-milestone-list command line option, [723](#)
  - gitlab-todo-list command line option, [795](#)
  - gitlab-user-impersonation-token-list command line option, [810](#)
- state-event
  - gitlab-group-milestone-update command line option, [562](#)
  - gitlab-project-issue-update command line option, [660](#)
  - gitlab-project-merge-request-update command line option, [699](#)
  - gitlab-project-milestone-create command line option, [722](#)
  - gitlab-project-milestone-update command line option, [724](#)
- states
  - gitlab-project-environment-list command line option, [639](#)
- statistics
  - gitlab-group-descendant-group-list command line option, [530](#)
  - gitlab-group-list command line option, [510](#)
  - gitlab-group-subgroup-list command line option, [570](#)
  - gitlab-project-fork-list command line option, [645](#)
  - gitlab-project-list command line option, [592](#)
  - gitlab-starred-project-list command line option, [794](#)
  - gitlab-user-project-list command line option, [815](#)
- status
  - gitlab-bulk-import-all-entity-list command line option, [492](#)
  - gitlab-bulk-import-entity-list command line option, [493](#)
  - gitlab-bulk-import-list command line option, [491](#)
  - gitlab-project-deployment-approval command line option, [633](#)
  - gitlab-project-deployment-create command line option, [634](#)
  - gitlab-project-deployment-list command line option, [635](#)
  - gitlab-project-pipeline-list command line option, [732](#)
  - gitlab-runner-all-list command line option, [788](#)
  - gitlab-runner-job-list command line option, [789](#)
  - gitlab-runner-list command line option, [787](#)
  - gitlab-user-list command line option, [801](#)
- subgroup-creation-level
  - gitlab-group-create command line option, [508](#)
  - gitlab-group-update command line option, [512](#)
- subgroup-events
  - gitlab-group-hook-create command line option, [544](#)
  - gitlab-group-hook-update command line option, [546](#)
- submodule
  - gitlab-project-update-submodule command line option, [602](#)
- sudo
  - gitlab-application-appearance-get command line option, [482](#)
  - gitlab-application-appearance-update command line option, [483](#)
  - gitlab-application-create command line option, [481](#)
  - gitlab-application-delete command line option, [482](#)
  - gitlab-application-list command line

option, [482](#)  
gitlab-application-settings-get command  
line option, [484](#)  
gitlab-application-settings-update  
command line option, [485](#)  
gitlab-application-statistics-get  
command line option, [488](#)  
gitlab-audit-event-get command line  
option, [488](#)  
gitlab-audit-event-list command line  
option, [488](#)  
gitlab-broadcast-message-create command  
line option, [489](#)  
gitlab-broadcast-message-delete command  
line option, [489](#)  
gitlab-broadcast-message-get command  
line option, [490](#)  
gitlab-broadcast-message-list command  
line option, [490](#)  
gitlab-broadcast-message-update command  
line option, [490](#)  
gitlab-bulk-import-all-entity-list  
command line option, [492](#)  
gitlab-bulk-import-create command line  
option, [491](#)  
gitlab-bulk-import-entity-get command  
line option, [492](#)  
gitlab-bulk-import-entity-list command  
line option, [493](#)  
gitlab-bulk-import-get command line  
option, [491](#)  
gitlab-bulk-import-list command line  
option, [491](#)  
gitlab-ci-lint-create command line  
option, [493](#)  
gitlab-current-user-email-create  
command line option, [494](#)  
gitlab-current-user-email-delete  
command line option, [495](#)  
gitlab-current-user-email-get command  
line option, [495](#)  
gitlab-current-user-email-list command  
line option, [495](#)  
gitlab-current-user-get command line  
option, [494](#)  
gitlab-current-user-gpg-key-create  
command line option, [496](#)  
gitlab-current-user-gpg-key-delete  
command line option, [496](#)  
gitlab-current-user-gpg-key-get command  
line option, [496](#)  
gitlab-current-user-gpg-key-list  
command line option, [496](#)  
gitlab-current-user-key-create command  
line option, [497](#)  
gitlab-current-user-key-delete command  
line option, [497](#)  
gitlab-current-user-key-get command  
line option, [497](#)  
gitlab-current-user-key-list command  
line option, [498](#)  
gitlab-current-user-runner-create  
command line option, [498](#)  
gitlab-current-user-status-get command  
line option, [499](#)  
gitlab-current-user-status-update  
command line option, [499](#)  
gitlab-deploy-key-list command line  
option, [500](#)  
gitlab-deploy-token-list command line  
option, [500](#)  
gitlab-dockerfile-get command line  
option, [501](#)  
gitlab-dockerfile-list command line  
option, [501](#)  
gitlab-event-list command line option,  
[501](#)  
gitlab-feature-delete command line  
option, [502](#)  
gitlab-feature-list command line option,  
[502](#)  
gitlab-generic-package-download command  
line option, [503](#)  
gitlab-generic-package-upload command  
line option, [503](#)  
gitlab-geo-node-delete command line  
option, [504](#)  
gitlab-geo-node-get command line option,  
[504](#)  
gitlab-geo-node-list command line  
option, [505](#)  
gitlab-geo-node-update command line  
option, [505](#)  
gitlab-gitignore-get command line  
option, [506](#)  
gitlab-gitignore-list command line  
option, [506](#)  
gitlab-gitlabciyaml-get command line  
option, [507](#)  
gitlab-gitlabciyaml-list command line  
option, [507](#)  
gitlab-group-access-request-approve  
command line option, [513](#)  
gitlab-group-access-request-create  
command line option, [514](#)  
gitlab-group-access-request-delete  
command line option, [514](#)  
gitlab-group-access-request-list

- command line option, [514](#)
- gitlab-group-access-token-create
  - command line option, [515](#)
- gitlab-group-access-token-delete
  - command line option, [515](#)
- gitlab-group-access-token-get command line option, [515](#)
- gitlab-group-access-token-list command line option, [516](#)
- gitlab-group-audit-event-get command line option, [516](#)
- gitlab-group-audit-event-list command line option, [517](#)
- gitlab-group-badge-create command line option, [517](#)
- gitlab-group-badge-delete command line option, [518](#)
- gitlab-group-badge-get command line option, [518](#)
- gitlab-group-badge-list command line option, [518](#)
- gitlab-group-badge-render command line option, [519](#)
- gitlab-group-badge-update command line option, [519](#)
- gitlab-group-billable-member-delete command line option, [519](#)
- gitlab-group-billable-member-list command line option, [520](#)
- gitlab-group-billable-member-membership-list command line option, [520](#)
- gitlab-group-board-create command line option, [521](#)
- gitlab-group-board-delete command line option, [521](#)
- gitlab-group-board-get command line option, [522](#)
- gitlab-group-board-list command line option, [522](#)
- gitlab-group-board-list-create command line option, [523](#)
- gitlab-group-board-list-delete command line option, [523](#)
- gitlab-group-board-list-get command line option, [524](#)
- gitlab-group-board-list-list command line option, [524](#)
- gitlab-group-board-list-update command line option, [524](#)
- gitlab-group-board-update command line option, [522](#)
- gitlab-group-cluster-create command line option, [525](#)
- gitlab-group-cluster-delete command line option, [525](#)
- gitlab-group-cluster-get command line option, [526](#)
- gitlab-group-cluster-list command line option, [526](#)
- gitlab-group-cluster-update command line option, [526](#)
- gitlab-group-create command line option, [508](#)
- gitlab-group-custom-attribute-delete command line option, [527](#)
- gitlab-group-custom-attribute-get command line option, [527](#)
- gitlab-group-custom-attribute-list command line option, [527](#)
- gitlab-group-delete command line option, [509](#)
- gitlab-group-deploy-token-create command line option, [528](#)
- gitlab-group-deploy-token-delete command line option, [528](#)
- gitlab-group-deploy-token-get command line option, [529](#)
- gitlab-group-deploy-token-list command line option, [529](#)
- gitlab-group-descendant-group-list command line option, [530](#)
- gitlab-group-epic-award-emoji-create command line option, [533](#)
- gitlab-group-epic-award-emoji-delete command line option, [533](#)
- gitlab-group-epic-award-emoji-get command line option, [533](#)
- gitlab-group-epic-award-emoji-list command line option, [534](#)
- gitlab-group-epic-create command line option, [531](#)
- gitlab-group-epic-delete command line option, [531](#)
- gitlab-group-epic-discussion-note-create command line option, [534](#)
- gitlab-group-epic-discussion-note-delete command line option, [535](#)
- gitlab-group-epic-discussion-note-get command line option, [535](#)
- gitlab-group-epic-discussion-note-update command line option, [535](#)
- gitlab-group-epic-get command line option, [531](#)
- gitlab-group-epic-issue-create command line option, [536](#)
- gitlab-group-epic-issue-delete command line option, [536](#)
- gitlab-group-epic-issue-list command

- line option, [537](#)
- gitlab-group-epic-issue-update command
  - line option, [537](#)
- gitlab-group-epic-list command line option, [532](#)
- gitlab-group-epic-note-award-emoji-create command line option, [540](#)
- gitlab-group-epic-note-award-emoji-delete command line option, [540](#)
- gitlab-group-epic-note-award-emoji-get command line option, [540](#)
- gitlab-group-epic-note-award-emoji-list command line option, [541](#)
- gitlab-group-epic-note-create command line option, [538](#)
- gitlab-group-epic-note-delete command line option, [538](#)
- gitlab-group-epic-note-get command line option, [538](#)
- gitlab-group-epic-note-list command line option, [539](#)
- gitlab-group-epic-note-update command line option, [539](#)
- gitlab-group-epic-resource-label-event-get command line option, [541](#)
- gitlab-group-epic-resource-label-event-list command line option, [542](#)
- gitlab-group-epic-update command line option, [532](#)
- gitlab-group-export-create command line option, [542](#)
- gitlab-group-export-download command line option, [542](#)
- gitlab-group-export-get command line option, [543](#)
- gitlab-group-get command line option, [509](#)
- gitlab-group-hook-create command line option, [543](#)
- gitlab-group-hook-delete command line option, [544](#)
- gitlab-group-hook-get command line option, [544](#)
- gitlab-group-hook-list command line option, [545](#)
- gitlab-group-hook-update command line option, [545](#)
- gitlab-group-import-get command line option, [546](#)
- gitlab-group-invitation-create command line option, [547](#)
- gitlab-group-invitation-delete command line option, [547](#)
- gitlab-group-invitation-get command line option, [547](#)
- gitlab-group-invitation-list command line option, [548](#)
- gitlab-group-invitation-update command line option, [548](#)
- gitlab-group-issue-list command line option, [549](#)
- gitlab-group-issues-statistics-get command line option, [550](#)
- gitlab-group-iteration-list command line option, [550](#)
- gitlab-group-label-create command line option, [551](#)
- gitlab-group-label-delete command line option, [552](#)
- gitlab-group-label-get command line option, [552](#)
- gitlab-group-label-list command line option, [552](#)
- gitlab-group-label-subscribe command line option, [553](#)
- gitlab-group-label-unsubscribe command line option, [553](#)
- gitlab-group-label-update command line option, [553](#)
- gitlab-group-ldap-group-link-create command line option, [554](#)
- gitlab-group-ldap-group-link-delete command line option, [554](#)
- gitlab-group-ldap-group-link-list command line option, [554](#)
- gitlab-group-list command line option, [510](#)
- gitlab-group-member-all-get command line option, [557](#)
- gitlab-group-member-all-list command line option, [557](#)
- gitlab-group-member-create command line option, [555](#)
- gitlab-group-member-delete command line option, [555](#)
- gitlab-group-member-get command line option, [556](#)
- gitlab-group-member-list command line option, [556](#)
- gitlab-group-member-update command line option, [556](#)
- gitlab-group-merge-request-list command line option, [558](#)
- gitlab-group-milestone-create command line option, [559](#)
- gitlab-group-milestone-delete command line option, [560](#)
- gitlab-group-milestone-get command line option, [560](#)

- gitlab-group-milestone-issues command line option, [560](#)
- gitlab-group-milestone-list command line option, [560](#)
- gitlab-group-milestone-merge-requests command line option, [561](#)
- gitlab-group-milestone-update command line option, [561](#)
- gitlab-group-notification-settings-get command line option, [562](#)
- gitlab-group-notification-settings-update command line option, [562](#)
- gitlab-group-package-list command line option, [563](#)
- gitlab-group-project-list command line option, [564](#)
- gitlab-group-push-rules-create command line option, [565](#)
- gitlab-group-push-rules-delete command line option, [566](#)
- gitlab-group-push-rules-get command line option, [566](#)
- gitlab-group-push-rules-update command line option, [567](#)
- gitlab-group-runner-list command line option, [567](#)
- gitlab-group-saml-group-link-create command line option, [568](#)
- gitlab-group-saml-group-link-delete command line option, [568](#)
- gitlab-group-saml-group-link-get command line option, [569](#)
- gitlab-group-saml-group-link-list command line option, [569](#)
- gitlab-group-subgroup-list command line option, [570](#)
- gitlab-group-update command line option, [512](#)
- gitlab-group-variable-create command line option, [570](#)
- gitlab-group-variable-delete command line option, [571](#)
- gitlab-group-variable-get command line option, [571](#)
- gitlab-group-variable-list command line option, [571](#)
- gitlab-group-variable-update command line option, [572](#)
- gitlab-group-wiki-create command line option, [572](#)
- gitlab-group-wiki-delete command line option, [573](#)
- gitlab-group-wiki-get command line option, [573](#)
- gitlab-group-wiki-list command line option, [573](#)
- gitlab-group-wiki-update command line option, [574](#)
- gitlab-hook-create command line option, [574](#)
- gitlab-hook-delete command line option, [574](#)
- gitlab-hook-get command line option, [575](#)
- gitlab-hook-list command line option, [575](#)
- gitlab-issue-get command line option, [575](#)
- gitlab-issue-list command line option, [576](#)
- gitlab-issues-statistics-get command line option, [577](#)
- gitlab-key-get command line option, [577](#)
- gitlab-ldap-group-list command line option, [578](#)
- gitlab-license-get command line option, [578](#)
- gitlab-license-list command line option, [579](#)
- gitlab-merge-request-list command line option, [580](#)
- gitlab-namespace-get command line option, [581](#)
- gitlab-namespace-list command line option, [582](#)
- gitlab-notification-settings-get command line option, [582](#)
- gitlab-notification-settings-update command line option, [583](#)
- gitlab-pages-domain-list command line option, [583](#)
- gitlab-personal-access-token-delete command line option, [584](#)
- gitlab-personal-access-token-get command line option, [584](#)
- gitlab-personal-access-token-list command line option, [584](#)
- gitlab-project-access-request-approve command line option, [603](#)
- gitlab-project-access-request-create command line option, [603](#)
- gitlab-project-access-request-delete command line option, [603](#)
- gitlab-project-access-request-list command line option, [603](#)
- gitlab-project-access-token-create command line option, [604](#)
- gitlab-project-access-token-delete command line option, [604](#)
- gitlab-project-access-token-get command line option, [605](#)

`gitlab-project-access-token-list`  
command line option, [605](#)

`gitlab-project-additional-statistics-get`  
command line option, [605](#)

`gitlab-project-approval-get` command  
line option, [606](#)

`gitlab-project-approval-rule-create`  
command line option, [607](#)

`gitlab-project-approval-rule-delete`  
command line option, [607](#)

`gitlab-project-approval-rule-list`  
command line option, [608](#)

`gitlab-project-approval-rule-update`  
command line option, [608](#)

`gitlab-project-approval-update` command  
line option, [606](#)

`gitlab-project-artifact-delete` command  
line option, [608](#)

`gitlab-project-artifact-download`  
command line option, [609](#)

`gitlab-project-artifact-raw` command  
line option, [609](#)

`gitlab-project-audit-event-get` command  
line option, [610](#)

`gitlab-project-audit-event-list` command  
line option, [610](#)

`gitlab-project-badge-create` command  
line option, [611](#)

`gitlab-project-badge-delete` command  
line option, [611](#)

`gitlab-project-badge-get` command line  
option, [611](#)

`gitlab-project-badge-list` command line  
option, [612](#)

`gitlab-project-badge-render` command  
line option, [612](#)

`gitlab-project-badge-update` command  
line option, [612](#)

`gitlab-project-board-create` command  
line option, [613](#)

`gitlab-project-board-delete` command  
line option, [613](#)

`gitlab-project-board-get` command line  
option, [613](#)

`gitlab-project-board-list` command line  
option, [614](#)

`gitlab-project-board-list-create`  
command line option, [614](#)

`gitlab-project-board-list-delete`  
command line option, [615](#)

`gitlab-project-board-list-get` command  
line option, [615](#)

`gitlab-project-board-list-list` command  
line option, [615](#)

`gitlab-project-board-list-update`  
command line option, [616](#)

`gitlab-project-board-update` command  
line option, [614](#)

`gitlab-project-branch-create` command  
line option, [616](#)

`gitlab-project-branch-delete` command  
line option, [617](#)

`gitlab-project-branch-get` command line  
option, [617](#)

`gitlab-project-branch-list` command line  
option, [617](#)

`gitlab-project-ci-lint-create` command  
line option, [618](#)

`gitlab-project-ci-lint-get` command line  
option, [618](#)

`gitlab-project-ci-lint-validate` command  
line option, [619](#)

`gitlab-project-cluster-create` command  
line option, [619](#)

`gitlab-project-cluster-delete` command  
line option, [620](#)

`gitlab-project-cluster-get` command line  
option, [620](#)

`gitlab-project-cluster-list` command  
line option, [620](#)

`gitlab-project-cluster-update` command  
line option, [621](#)

`gitlab-project-commit-cherry-pick`  
command line option, [622](#)

`gitlab-project-commit-comment-create`  
command line option, [625](#)

`gitlab-project-commit-comment-list`  
command line option, [626](#)

`gitlab-project-commit-create` command  
line option, [622](#)

`gitlab-project-commit-diff` command line  
option, [622](#)

`gitlab-project-commit-discussion-create`  
command line option, [626](#)

`gitlab-project-commit-discussion-get`  
command line option, [627](#)

`gitlab-project-commit-discussion-list`  
command line option, [627](#)

`gitlab-project-commit-discussion-note-create`  
command line option, [628](#)

`gitlab-project-commit-discussion-note-delete`  
command line option, [628](#)

`gitlab-project-commit-discussion-note-get`  
command line option, [629](#)

`gitlab-project-commit-discussion-note-update`  
command line option, [629](#)

`gitlab-project-commit-get` command line  
option, [623](#)



gitlab-project-commit-list command line option, [623](#)  
gitlab-project-commit-merge-requests command line option, [624](#)  
gitlab-project-commit-refs command line option, [624](#)  
gitlab-project-commit-revert command line option, [624](#)  
gitlab-project-commit-signature command line option, [625](#)  
gitlab-project-commit-status-create command line option, [630](#)  
gitlab-project-commit-status-list command line option, [630](#)  
gitlab-project-create command line option, [587](#)  
gitlab-project-custom-attribute-delete command line option, [631](#)  
gitlab-project-custom-attribute-get command line option, [631](#)  
gitlab-project-custom-attribute-list command line option, [631](#)  
gitlab-project-delete command line option, [590](#)  
gitlab-project-deploy-token-create command line option, [632](#)  
gitlab-project-deploy-token-delete command line option, [632](#)  
gitlab-project-deploy-token-get command line option, [632](#)  
gitlab-project-deploy-token-list command line option, [633](#)  
gitlab-project-deployment-approval command line option, [633](#)  
gitlab-project-deployment-create command line option, [634](#)  
gitlab-project-deployment-get command line option, [634](#)  
gitlab-project-deployment-list command line option, [634](#)  
gitlab-project-deployment-merge-request-list command line option, [636](#)  
gitlab-project-deployment-update command line option, [635](#)  
gitlab-project-environment-create command line option, [638](#)  
gitlab-project-environment-delete command line option, [638](#)  
gitlab-project-environment-get command line option, [638](#)  
gitlab-project-environment-list command line option, [639](#)  
gitlab-project-environment-stop command line option, [639](#)  
gitlab-project-environment-update command line option, [639](#)  
gitlab-project-event-list command line option, [640](#)  
gitlab-project-export-create command line option, [641](#)  
gitlab-project-export-download command line option, [641](#)  
gitlab-project-export-get command line option, [641](#)  
gitlab-project-file-blame command line option, [642](#)  
gitlab-project-file-create command line option, [642](#)  
gitlab-project-file-delete command line option, [643](#)  
gitlab-project-file-get command line option, [643](#)  
gitlab-project-file-raw command line option, [643](#)  
gitlab-project-file-update command line option, [644](#)  
gitlab-project-fork-create command line option, [644](#)  
gitlab-project-fork-list command line option, [645](#)  
gitlab-project-get command line option, [590](#)  
gitlab-project-hook-create command line option, [646](#)  
gitlab-project-hook-delete command line option, [647](#)  
gitlab-project-hook-get command line option, [647](#)  
gitlab-project-hook-list command line option, [647](#)  
gitlab-project-hook-update command line option, [648](#)  
gitlab-project-import-get command line option, [649](#)  
gitlab-project-integration-available command line option, [649](#)  
gitlab-project-integration-delete command line option, [649](#)  
gitlab-project-integration-get command line option, [650](#)  
gitlab-project-integration-list command line option, [650](#)  
gitlab-project-integration-update command line option, [650](#)  
gitlab-project-invitation-create command line option, [651](#)  
gitlab-project-invitation-delete command line option, [651](#)

`gitlab-project-invitation-get` command line option, [652](#)  
`gitlab-project-invitation-list` command line option, [652](#)  
`gitlab-project-invitation-update` command line option, [652](#)  
`gitlab-project-issue-add-spent-time` command line option, [653](#)  
`gitlab-project-issue-award-emoji-create` command line option, [661](#)  
`gitlab-project-issue-award-emoji-delete` command line option, [661](#)  
`gitlab-project-issue-award-emoji-get` command line option, [662](#)  
`gitlab-project-issue-award-emoji-list` command line option, [662](#)  
`gitlab-project-issue-closed-by` command line option, [654](#)  
`gitlab-project-issue-create` command line option, [654](#)  
`gitlab-project-issue-delete` command line option, [655](#)  
`gitlab-project-issue-discussion-create` command line option, [663](#)  
`gitlab-project-issue-discussion-get` command line option, [663](#)  
`gitlab-project-issue-discussion-list` command line option, [663](#)  
`gitlab-project-issue-discussion-note-create` command line option, [664](#)  
`gitlab-project-issue-discussion-note-delete` command line option, [664](#)  
`gitlab-project-issue-discussion-note-get` command line option, [665](#)  
`gitlab-project-issue-discussion-note-update` command line option, [665](#)  
`gitlab-project-issue-get` command line option, [655](#)  
`gitlab-project-issue-link-create` command line option, [666](#)  
`gitlab-project-issue-link-delete` command line option, [666](#)  
`gitlab-project-issue-link-list` command line option, [667](#)  
`gitlab-project-issue-list` command line option, [655](#)  
`gitlab-project-issue-move` command line option, [656](#)  
`gitlab-project-issue-note-award-emoji-create` command line option, [669](#)  
`gitlab-project-issue-note-award-emoji-delete` command line option, [670](#)  
`gitlab-project-issue-note-award-emoji-get` command line option, [670](#)  
`gitlab-project-issue-note-award-emoji-list` command line option, [670](#)  
`gitlab-project-issue-note-create` command line option, [667](#)  
`gitlab-project-issue-note-delete` command line option, [668](#)  
`gitlab-project-issue-note-get` command line option, [668](#)  
`gitlab-project-issue-note-list` command line option, [668](#)  
`gitlab-project-issue-note-update` command line option, [669](#)  
`gitlab-project-issue-participants` command line option, [657](#)  
`gitlab-project-issue-related-merge-requests` command line option, [657](#)  
`gitlab-project-issue-reorder` command line option, [657](#)  
`gitlab-project-issue-reset-spent-time` command line option, [658](#)  
`gitlab-project-issue-reset-time-estimate` command line option, [658](#)  
`gitlab-project-issue-resource-iteration-event-get` command line option, [671](#)  
`gitlab-project-issue-resource-iteration-event-list` command line option, [672](#)  
`gitlab-project-issue-resource-label-event-get` command line option, [672](#)  
`gitlab-project-issue-resource-label-event-list` command line option, [672](#)  
`gitlab-project-issue-resource-milestone-event-get` command line option, [673](#)  
`gitlab-project-issue-resource-milestone-event-list` command line option, [673](#)  
`gitlab-project-issue-resource-state-event-get` command line option, [674](#)  
`gitlab-project-issue-resource-state-event-list` command line option, [674](#)  
`gitlab-project-issue-resource-weight-event-get` command line option, [675](#)  
`gitlab-project-issue-resource-weight-event-list` command line option, [675](#)  
`gitlab-project-issue-subscribe` command line option, [658](#)  
`gitlab-project-issue-time-estimate` command line option, [658](#)  
`gitlab-project-issue-time-stats` command line option, [659](#)  
`gitlab-project-issue-todo` command line option, [659](#)  
`gitlab-project-issue-unsubscribe` command line option, [659](#)  
`gitlab-project-issue-update` command line option, [660](#)



gitlab-project-issue-user-agent-detail  
command line option, 660

gitlab-project-issues-statistics-get  
command line option, 676

gitlab-project-job-artifact command  
line option, 677

gitlab-project-job-artifacts command  
line option, 677

gitlab-project-job-cancel command line  
option, 677

gitlab-project-job-delete-artifacts  
command line option, 677

gitlab-project-job-erase command line  
option, 678

gitlab-project-job-get command line  
option, 678

gitlab-project-job-keep-artifacts  
command line option, 678

gitlab-project-job-list command line  
option, 679

gitlab-project-job-play command line  
option, 679

gitlab-project-job-retry command line  
option, 679

gitlab-project-job-token-scope-get  
command line option, 680

gitlab-project-job-token-scope-update  
command line option, 680

gitlab-project-job-trace command line  
option, 680

gitlab-project-key-create command line  
option, 681

gitlab-project-key-delete command line  
option, 681

gitlab-project-key-enable command line  
option, 681

gitlab-project-key-get command line  
option, 682

gitlab-project-key-list command line  
option, 682

gitlab-project-key-update command line  
option, 682

gitlab-project-label-create command  
line option, 683

gitlab-project-label-delete command  
line option, 683

gitlab-project-label-get command line  
option, 684

gitlab-project-label-list command line  
option, 684

gitlab-project-label-subscribe command  
line option, 684

gitlab-project-label-unsubscribe  
command line option, 685

gitlab-project-label-update command  
line option, 685

gitlab-project-list command line option,  
591

gitlab-project-member-all-get command  
line option, 688

gitlab-project-member-all-list command  
line option, 688

gitlab-project-member-create command  
line option, 686

gitlab-project-member-delete command  
line option, 686

gitlab-project-member-get command line  
option, 686

gitlab-project-member-list command line  
option, 687

gitlab-project-member-update command  
line option, 687

gitlab-project-merge-request-add-spent-time  
command line option, 689

gitlab-project-merge-request-approval-get  
command line option, 699

gitlab-project-merge-request-approval-rule-create  
command line option, 700

gitlab-project-merge-request-approval-rule-delete  
command line option, 701

gitlab-project-merge-request-approval-rule-get  
command line option, 701

gitlab-project-merge-request-approval-rule-list  
command line option, 702

gitlab-project-merge-request-approval-rule-update  
command line option, 702

gitlab-project-merge-request-approval-state-get  
command line option, 703

gitlab-project-merge-request-approval-update  
command line option, 700

gitlab-project-merge-request-approve  
command line option, 689

gitlab-project-merge-request-award-emoji-create  
command line option, 704

gitlab-project-merge-request-award-emoji-delete  
command line option, 704

gitlab-project-merge-request-award-emoji-get  
command line option, 704

gitlab-project-merge-request-award-emoji-list  
command line option, 705

gitlab-project-merge-request-cancel-merge-when-pipeline-fails  
command line option, 690

gitlab-project-merge-request-changes  
command line option, 690

gitlab-project-merge-request-closes-issues  
command line option, 690

gitlab-project-merge-request-commits  
command line option, 691

<code>gitlab-project-merge-request-create</code> command line option, <a href="#">691</a>	<code>gitlab-project-merge-request-note-get</code> command line option, <a href="#">713</a>
<code>gitlab-project-merge-request-delete</code> command line option, <a href="#">692</a>	<code>gitlab-project-merge-request-note-list</code> command line option, <a href="#">714</a>
<code>gitlab-project-merge-request-diff-get</code> command line option, <a href="#">705</a>	<code>gitlab-project-merge-request-note-update</code> command line option, <a href="#">714</a>
<code>gitlab-project-merge-request-diff-list</code> command line option, <a href="#">706</a>	<code>gitlab-project-merge-request-participants</code> command line option, <a href="#">695</a>
<code>gitlab-project-merge-request-discussion-create</code> command line option, <a href="#">706</a>	<code>gitlab-project-merge-request-pipeline-create</code> command line option, <a href="#">716</a>
<code>gitlab-project-merge-request-discussion-get</code> command line option, <a href="#">707</a>	<code>gitlab-project-merge-request-pipeline-list</code> command line option, <a href="#">717</a>
<code>gitlab-project-merge-request-discussion-list</code> command line option, <a href="#">707</a>	<code>gitlab-project-merge-request-rebase</code> command line option, <a href="#">695</a>
<code>gitlab-project-merge-request-discussion-note-create</code> command line option, <a href="#">708</a>	<code>gitlab-project-merge-request-reset-approvals</code> command line option, <a href="#">695</a>
<code>gitlab-project-merge-request-discussion-note-delete</code> command line option, <a href="#">709</a>	<code>gitlab-project-merge-request-reset-spent-time</code> command line option, <a href="#">696</a>
<code>gitlab-project-merge-request-discussion-note-get</code> command line option, <a href="#">709</a>	<code>gitlab-project-merge-request-reset-time-estimate</code> command line option, <a href="#">696</a>
<code>gitlab-project-merge-request-discussion-note-update</code> command line option, <a href="#">710</a>	<code>gitlab-project-merge-request-resource-label-event-get</code> command line option, <a href="#">717</a>
<code>gitlab-project-merge-request-discussion-update</code> command line option, <a href="#">708</a>	<code>gitlab-project-merge-request-resource-label-event-list</code> command line option, <a href="#">718</a>
<code>gitlab-project-merge-request-draft-note-create</code> command line option, <a href="#">711</a>	<code>gitlab-project-merge-request-resource-milestone-event-</code> command line option, <a href="#">718</a>
<code>gitlab-project-merge-request-draft-note-delete</code> command line option, <a href="#">711</a>	<code>gitlab-project-merge-request-resource-milestone-event-</code> command line option, <a href="#">719</a>
<code>gitlab-project-merge-request-draft-note-get</code> command line option, <a href="#">711</a>	<code>gitlab-project-merge-request-resource-state-event-get</code> command line option, <a href="#">719</a>
<code>gitlab-project-merge-request-draft-note-list</code> command line option, <a href="#">712</a>	<code>gitlab-project-merge-request-resource-state-event-list</code> command line option, <a href="#">720</a>
<code>gitlab-project-merge-request-draft-note-update</code> command line option, <a href="#">712</a>	<code>gitlab-project-merge-request-reviewer-detail-list</code> command line option, <a href="#">720</a>
<code>gitlab-project-merge-request-get</code> command line option, <a href="#">692</a>	<code>gitlab-project-merge-request-subscribe</code> command line option, <a href="#">696</a>
<code>gitlab-project-merge-request-list</code> command line option, <a href="#">693</a>	<code>gitlab-project-merge-request-time-estimate</code> command line option, <a href="#">697</a>
<code>gitlab-project-merge-request-merge</code> command line option, <a href="#">694</a>	<code>gitlab-project-merge-request-time-stats</code> command line option, <a href="#">697</a>
<code>gitlab-project-merge-request-merge-ref</code> command line option, <a href="#">694</a>	<code>gitlab-project-merge-request-todo</code> command line option, <a href="#">697</a>
<code>gitlab-project-merge-request-note-award-emoji-create</code> command line option, <a href="#">715</a>	<code>gitlab-project-merge-request-unapprove</code> command line option, <a href="#">698</a>
<code>gitlab-project-merge-request-note-award-emoji-delete</code> command line option, <a href="#">715</a>	<code>gitlab-project-merge-request-unsubscribe</code> command line option, <a href="#">698</a>
<code>gitlab-project-merge-request-note-award-emoji-get</code> command line option, <a href="#">715</a>	<code>gitlab-project-merge-request-update</code> command line option, <a href="#">698</a>
<code>gitlab-project-merge-request-note-award-emoji-list</code> command line option, <a href="#">716</a>	<code>gitlab-project-merge-train-list</code> command line option, <a href="#">721</a>
<code>gitlab-project-merge-request-note-create</code> command line option, <a href="#">713</a>	<code>gitlab-project-milestone-create</code> command line option, <a href="#">722</a>
<code>gitlab-project-merge-request-note-delete</code> command line option, <a href="#">713</a>	<code>gitlab-project-milestone-delete</code> command line option, <a href="#">722</a>

gitlab-project-milestone-get command line option, [722](#)  
gitlab-project-milestone-issues command line option, [723](#)  
gitlab-project-milestone-list command line option, [723](#)  
gitlab-project-milestone-merge-requests command line option, [723](#)  
gitlab-project-milestone-update command line option, [724](#)  
gitlab-project-note-get command line option, [724](#)  
gitlab-project-note-list command line option, [725](#)  
gitlab-project-notification-settings-get command line option, [725](#)  
gitlab-project-notification-settings-update command line option, [726](#)  
gitlab-project-package-delete command line option, [727](#)  
gitlab-project-package-file-delete command line option, [728](#)  
gitlab-project-package-file-list command line option, [728](#)  
gitlab-project-package-get command line option, [727](#)  
gitlab-project-package-list command line option, [727](#)  
gitlab-project-package-pipeline-list command line option, [729](#)  
gitlab-project-pages-domain-create command line option, [729](#)  
gitlab-project-pages-domain-delete command line option, [730](#)  
gitlab-project-pages-domain-get command line option, [730](#)  
gitlab-project-pages-domain-list command line option, [730](#)  
gitlab-project-pages-domain-update command line option, [731](#)  
gitlab-project-pipeline-bridge-list command line option, [733](#)  
gitlab-project-pipeline-cancel command line option, [731](#)  
gitlab-project-pipeline-create command line option, [731](#)  
gitlab-project-pipeline-delete command line option, [732](#)  
gitlab-project-pipeline-get command line option, [732](#)  
gitlab-project-pipeline-job-list command line option, [734](#)  
gitlab-project-pipeline-list command line option, [732](#)  
gitlab-project-pipeline-retry command line option, [733](#)  
gitlab-project-pipeline-schedule-create command line option, [735](#)  
gitlab-project-pipeline-schedule-delete command line option, [735](#)  
gitlab-project-pipeline-schedule-get command line option, [736](#)  
gitlab-project-pipeline-schedule-list command line option, [736](#)  
gitlab-project-pipeline-schedule-pipeline-list command line option, [738](#)  
gitlab-project-pipeline-schedule-play command line option, [736](#)  
gitlab-project-pipeline-schedule-take-ownership command line option, [737](#)  
gitlab-project-pipeline-schedule-update command line option, [737](#)  
gitlab-project-pipeline-schedule-variable-create command line option, [738](#)  
gitlab-project-pipeline-schedule-variable-delete command line option, [739](#)  
gitlab-project-pipeline-schedule-variable-update command line option, [739](#)  
gitlab-project-pipeline-test-report-get command line option, [740](#)  
gitlab-project-pipeline-test-report-summary-get command line option, [740](#)  
gitlab-project-pipeline-variable-list command line option, [741](#)  
gitlab-project-protected-branch-create command line option, [742](#)  
gitlab-project-protected-branch-delete command line option, [742](#)  
gitlab-project-protected-branch-get command line option, [743](#)  
gitlab-project-protected-branch-list command line option, [743](#)  
gitlab-project-protected-branch-update command line option, [743](#)  
gitlab-project-protected-environment-create command line option, [744](#)  
gitlab-project-protected-environment-delete command line option, [744](#)  
gitlab-project-protected-environment-get command line option, [745](#)  
gitlab-project-protected-environment-list command line option, [745](#)  
gitlab-project-protected-tag-create command line option, [746](#)  
gitlab-project-protected-tag-delete command line option, [746](#)  
gitlab-project-protected-tag-get command line option, [746](#)

`gitlab-project-protected-tag-list`  
command line option, [747](#)

`gitlab-project-push-rules-create`  
command line option, [747](#)

`gitlab-project-push-rules-delete`  
command line option, [748](#)

`gitlab-project-push-rules-get` command  
line option, [748](#)

`gitlab-project-push-rules-update`  
command line option, [749](#)

`gitlab-project-registry-repository-delete`  
command line option, [750](#)

`gitlab-project-registry-repository-list`  
command line option, [750](#)

`gitlab-project-registry-tag-delete`  
command line option, [750](#)

`gitlab-project-registry-tag-delete-in-bulk`  
command line option, [751](#)

`gitlab-project-registry-tag-get` command  
line option, [751](#)

`gitlab-project-registry-tag-list`  
command line option, [752](#)

`gitlab-project-release-create` command  
line option, [752](#)

`gitlab-project-release-delete` command  
line option, [753](#)

`gitlab-project-release-get` command line  
option, [753](#)

`gitlab-project-release-link-create`  
command line option, [754](#)

`gitlab-project-release-link-delete`  
command line option, [755](#)

`gitlab-project-release-link-get` command  
line option, [755](#)

`gitlab-project-release-link-list`  
command line option, [755](#)

`gitlab-project-release-link-update`  
command line option, [756](#)

`gitlab-project-release-list` command  
line option, [753](#)

`gitlab-project-release-update` command  
line option, [754](#)

`gitlab-project-remote-mirror-create`  
command line option, [757](#)

`gitlab-project-remote-mirror-delete`  
command line option, [757](#)

`gitlab-project-remote-mirror-list`  
command line option, [757](#)

`gitlab-project-remote-mirror-update`  
command line option, [758](#)

`gitlab-project-resource-group-get`  
command line option, [758](#)

`gitlab-project-resource-group-list`  
command line option, [759](#)

`gitlab-project-resource-group-upcoming-job-list`  
command line option, [760](#)

`gitlab-project-resource-group-update`  
command line option, [759](#)

`gitlab-project-runner-create` command  
line option, [760](#)

`gitlab-project-runner-delete` command  
line option, [760](#)

`gitlab-project-runner-list` command line  
option, [761](#)

`gitlab-project-secure-file-create`  
command line option, [761](#)

`gitlab-project-secure-file-delete`  
command line option, [762](#)

`gitlab-project-secure-file-download`  
command line option, [762](#)

`gitlab-project-secure-file-get` command  
line option, [762](#)

`gitlab-project-secure-file-list` command  
line option, [762](#)

`gitlab-project-service-available`  
command line option, [763](#)

`gitlab-project-service-delete` command  
line option, [763](#)

`gitlab-project-service-get` command line  
option, [764](#)

`gitlab-project-service-list` command  
line option, [764](#)

`gitlab-project-service-update` command  
line option, [764](#)

`gitlab-project-snippet-award-emoji-create`  
command line option, [768](#)

`gitlab-project-snippet-award-emoji-delete`  
command line option, [768](#)

`gitlab-project-snippet-award-emoji-get`  
command line option, [768](#)

`gitlab-project-snippet-award-emoji-list`  
command line option, [769](#)

`gitlab-project-snippet-content` command  
line option, [765](#)

`gitlab-project-snippet-create` command  
line option, [765](#)

`gitlab-project-snippet-delete` command  
line option, [766](#)

`gitlab-project-snippet-discussion-create`  
command line option, [769](#)

`gitlab-project-snippet-discussion-get`  
command line option, [770](#)

`gitlab-project-snippet-discussion-list`  
command line option, [770](#)

`gitlab-project-snippet-discussion-note-create`  
command line option, [771](#)

`gitlab-project-snippet-discussion-note-delete`  
command line option, [771](#)

gitlab-project-snippet-discussion-note-get command line option, <a href="#">772</a>	gitlab-project-variable-create command line option, <a href="#">780</a>
gitlab-project-snippet-discussion-note-update command line option, <a href="#">772</a>	gitlab-project-variable-delete command line option, <a href="#">781</a>
gitlab-project-snippet-get command line option, <a href="#">766</a>	gitlab-project-variable-get command line option, <a href="#">781</a>
gitlab-project-snippet-list command line option, <a href="#">766</a>	gitlab-project-variable-list command line option, <a href="#">781</a>
gitlab-project-snippet-note-award-emoji-create command line option, <a href="#">775</a>	gitlab-project-variable-update command line option, <a href="#">782</a>
gitlab-project-snippet-note-award-emoji-delete command line option, <a href="#">775</a>	gitlab-project-wiki-create command line option, <a href="#">783</a>
gitlab-project-snippet-note-award-emoji-get command line option, <a href="#">775</a>	gitlab-project-wiki-delete command line option, <a href="#">783</a>
gitlab-project-snippet-note-award-emoji-list command line option, <a href="#">776</a>	gitlab-project-wiki-get command line option, <a href="#">783</a>
gitlab-project-snippet-note-create command line option, <a href="#">772</a>	gitlab-project-wiki-list command line option, <a href="#">784</a>
gitlab-project-snippet-note-delete command line option, <a href="#">773</a>	gitlab-project-wiki-update command line option, <a href="#">784</a>
gitlab-project-snippet-note-get command line option, <a href="#">773</a>	gitlab-project-wiki-upload command line option, <a href="#">784</a>
gitlab-project-snippet-note-list command line option, <a href="#">773</a>	gitlab-registry-repository-get command line option, <a href="#">785</a>
gitlab-project-snippet-note-update command line option, <a href="#">774</a>	gitlab-runner-all-list command line option, <a href="#">788</a>
gitlab-project-snippet-update command line option, <a href="#">767</a>	gitlab-runner-create command line option, <a href="#">786</a>
gitlab-project-snippet-user-agent-detail command line option, <a href="#">767</a>	gitlab-runner-delete command line option, <a href="#">786</a>
gitlab-project-storage-get command line option, <a href="#">776</a>	gitlab-runner-get command line option, <a href="#">786</a>
gitlab-project-tag-create command line option, <a href="#">777</a>	gitlab-runner-job-list command line option, <a href="#">789</a>
gitlab-project-tag-delete command line option, <a href="#">777</a>	gitlab-runner-list command line option, <a href="#">787</a>
gitlab-project-tag-get command line option, <a href="#">777</a>	gitlab-runner-update command line option, <a href="#">787</a>
gitlab-project-tag-list command line option, <a href="#">778</a>	gitlab-shared-project-list command line option, <a href="#">790</a>
gitlab-project-trigger-create command line option, <a href="#">778</a>	gitlab-snippet-create command line option, <a href="#">791</a>
gitlab-project-trigger-delete command line option, <a href="#">778</a>	gitlab-snippet-delete command line option, <a href="#">791</a>
gitlab-project-trigger-get command line option, <a href="#">779</a>	gitlab-snippet-get command line option, <a href="#">792</a>
gitlab-project-trigger-list command line option, <a href="#">779</a>	gitlab-snippet-list command line option, <a href="#">792</a>
gitlab-project-trigger-update command line option, <a href="#">779</a>	gitlab-snippet-update command line option, <a href="#">792</a>
gitlab-project-update command line option, <a href="#">599</a>	gitlab-starred-project-list command line option, <a href="#">793</a>
gitlab-project-user-list command line option, <a href="#">780</a>	gitlab-todo-delete command line option, <a href="#">794</a>

gitlab-todo-list command line option, [795](#)  
gitlab-topic-create command line option, [796](#)  
gitlab-topic-delete command line option, [796](#)  
gitlab-topic-get command line option, [796](#)  
gitlab-topic-list command line option, [797](#)  
gitlab-topic-update command line option, [797](#)  
gitlab-user-activities-list command line option, [804](#)  
gitlab-user-create command line option, [799](#)  
gitlab-user-custom-attribute-delete command line option, [805](#)  
gitlab-user-custom-attribute-get command line option, [805](#)  
gitlab-user-custom-attribute-list command line option, [805](#)  
gitlab-user-delete command line option, [800](#)  
gitlab-user-email-create command line option, [806](#)  
gitlab-user-email-delete command line option, [806](#)  
gitlab-user-email-get command line option, [806](#)  
gitlab-user-email-list command line option, [807](#)  
gitlab-user-event-list command line option, [807](#)  
gitlab-user-get command line option, [801](#)  
gitlab-user-gpg-key-create command line option, [808](#)  
gitlab-user-gpg-key-delete command line option, [808](#)  
gitlab-user-gpg-key-get command line option, [808](#)  
gitlab-user-gpg-key-list command line option, [809](#)  
gitlab-user-impersonation-token-create command line option, [809](#)  
gitlab-user-impersonation-token-delete command line option, [810](#)  
gitlab-user-impersonation-token-get command line option, [810](#)  
gitlab-user-impersonation-token-list command line option, [810](#)  
gitlab-user-key-create command line option, [811](#)  
gitlab-user-key-delete command line option, [811](#)  
gitlab-user-key-get command line option, [811](#)  
gitlab-user-key-list command line option, [812](#)  
gitlab-user-list command line option, [801](#)  
gitlab-user-membership-get command line option, [812](#)  
gitlab-user-membership-list command line option, [812](#)  
gitlab-user-personal-access-token-create command line option, [813](#)  
gitlab-user-project-create command line option, [814](#)  
gitlab-user-project-list command line option, [815](#)  
gitlab-user-status-get command line option, [816](#)  
gitlab-user-update command line option, [803](#)  
gitlab-variable-create command line option, [816](#)  
gitlab-variable-delete command line option, [817](#)  
gitlab-variable-get command line option, [817](#)  
gitlab-variable-list command line option, [817](#)  
gitlab-variable-update command line option, [818](#)  
--suggestion-commit-message  
    gitlab-project-update command line option, [601](#)  
--tag  
    gitlab-project-deployment-create command line option, [634](#)  
--tag-list  
    gitlab-current-user-runner-create command line option, [499](#)  
    gitlab-group-runner-list command line option, [568](#)  
    gitlab-project-create command line option, [589](#)  
    gitlab-project-runner-list command line option, [761](#)  
    gitlab-project-update command line option, [601](#)  
    gitlab-runner-all-list command line option, [788](#)  
    gitlab-runner-create command line option, [786](#)  
    gitlab-runner-list command line option, [787](#)  
    gitlab-runner-update command line option, [787](#)  
--tag-name



- gitlab-project-release-create command line option, [752](#)
- gitlab-project-release-delete command line option, [753](#)
- gitlab-project-release-get command line option, [753](#)
- gitlab-project-release-link-create command line option, [754](#)
- gitlab-project-release-link-delete command line option, [755](#)
- gitlab-project-release-link-get command line option, [755](#)
- gitlab-project-release-link-list command line option, [755](#)
- gitlab-project-release-link-update command line option, [756](#)
- gitlab-project-release-update command line option, [754](#)
- gitlab-project-tag-create command line option, [777](#)
- tag-push-events
  - gitlab-group-hook-create command line option, [543](#)
  - gitlab-group-hook-update command line option, [546](#)
  - gitlab-project-hook-create command line option, [646](#)
  - gitlab-project-hook-update command line option, [648](#)
- target-access-levels
  - gitlab-broadcast-message-create command line option, [489](#)
  - gitlab-broadcast-message-update command line option, [490](#)
- target-branch
  - gitlab-group-merge-request-list command line option, [559](#)
  - gitlab-merge-request-list command line option, [580](#)
  - gitlab-project-deployment-merge-request-list command line option, [637](#)
  - gitlab-project-merge-request-create command line option, [691](#)
  - gitlab-project-merge-request-list command line option, [694](#)
  - gitlab-project-merge-request-update command line option, [699](#)
- target-issue-iid
  - gitlab-project-issue-link-create command line option, [666](#)
- target-project-id
  - gitlab-project-issue-link-create command line option, [666](#)
  - gitlab-project-merge-request-create command line option, [692](#)
- target-topic-id
  - gitlab-topic-merge command line option, [797](#)
- target-type
  - gitlab-event-list command line option, [501](#)
  - gitlab-project-event-list command line option, [640](#)
  - gitlab-user-event-list command line option, [807](#)
- target-url
  - gitlab-project-commit-status-create command line option, [630](#)
- tasks-project-id
  - gitlab-group-invitation-create command line option, [547](#)
  - gitlab-project-invitation-create command line option, [651](#)
- tasks-to-be-done
  - gitlab-group-invitation-create command line option, [547](#)
  - gitlab-group-member-create command line option, [555](#)
  - gitlab-project-invitation-create command line option, [651](#)
  - gitlab-project-member-create command line option, [686](#)
- template-name
  - gitlab-project-create command line option, [589](#)
- template-project-id
  - gitlab-project-create command line option, [589](#)
- terminal-max-session-time
  - gitlab-application-settings-update command line option, [487](#)
- terms
  - gitlab-application-settings-update command line option, [487](#)
- theme-id
  - gitlab-user-create command line option, [800](#)
  - gitlab-user-update command line option, [804](#)
- timeout
  - gitlab command line option, [481](#)
- title
  - gitlab-application-appearance-update command line option, [483](#)
  - gitlab-current-user-key-create command line option, [497](#)
  - gitlab-group-epic-create command line option, [531](#)

- gitlab-group-epic-update command line option, [532](#)
- gitlab-group-milestone-create command line option, [559](#)
- gitlab-group-milestone-update command line option, [561](#)
- gitlab-group-wiki-create command line option, [572](#)
- gitlab-group-wiki-update command line option, [574](#)
- gitlab-project-issue-create command line option, [654](#)
- gitlab-project-issue-update command line option, [660](#)
- gitlab-project-key-create command line option, [681](#)
- gitlab-project-key-update command line option, [682](#)
- gitlab-project-merge-request-create command line option, [691](#)
- gitlab-project-merge-request-update command line option, [699](#)
- gitlab-project-milestone-create command line option, [722](#)
- gitlab-project-milestone-update command line option, [724](#)
- gitlab-project-snippet-create command line option, [765](#)
- gitlab-project-snippet-update command line option, [767](#)
- gitlab-project-wiki-create command line option, [783](#)
- gitlab-project-wiki-update command line option, [784](#)
- gitlab-snippet-create command line option, [791](#)
- gitlab-snippet-update command line option, [792](#)
- gitlab-topic-create command line option, [796](#)
- gitlab-user-key-create command line option, [811](#)
- to
  - gitlab-project-repository-compare command line option, [593](#)
- to-namespace
  - gitlab-project-transfer command line option, [596](#)
- to-project-id
  - gitlab-project-issue-move command line option, [656](#)
- token
  - gitlab-group-hook-create command line option, [544](#)
  - gitlab-group-hook-update command line option, [546](#)
  - gitlab-project-hook-create command line option, [646](#)
  - gitlab-project-hook-update command line option, [648](#)
  - gitlab-project-trigger-pipeline command line option, [596](#)
  - gitlab-runner-create command line option, [786](#)
  - gitlab-runner-verify command line option, [788](#)
- top-level-only
  - gitlab-group-list command line option, [510](#)
- topic
  - gitlab-project-list command line option, [592](#)
- topics
  - gitlab-project-create command line option, [589](#)
  - gitlab-project-update command line option, [601](#)
- trailers
  - gitlab-project-commit-list command line option, [623](#)
- twitter
  - gitlab-user-create command line option, [799](#)
  - gitlab-user-update command line option, [803](#)
- two-factor
  - gitlab-user-list command line option, [801](#)
- two-factor-grace-period
  - gitlab-group-create command line option, [508](#)
  - gitlab-group-update command line option, [512](#)
- type
  - gitlab-project-commit-refs command line option, [624](#)
  - gitlab-runner-all-list command line option, [788](#)
  - gitlab-runner-list command line option, [787](#)
  - gitlab-todo-list command line option, [795](#)
  - gitlab-user-membership-list command line option, [812](#)
- unprotect-access-level
  - gitlab-project-protected-branch-create command line option, [742](#)
- until
  - gitlab-project-commit-list command line option, [623](#)



- updated-after
  - gitlab-group-issue-list command line option, [550](#)
  - gitlab-group-merge-request-list command line option, [558](#)
  - gitlab-issue-list command line option, [576](#)
  - gitlab-merge-request-list command line option, [580](#)
  - gitlab-project-deployment-list command line option, [635](#)
  - gitlab-project-deployment-merge-request-list command line option, [637](#)
  - gitlab-project-issue-list command line option, [656](#)
  - gitlab-project-merge-request-list command line option, [693](#)
- updated-at
  - gitlab-application-settings-update command line option, [486](#)
  - gitlab-project-issue-update command line option, [660](#)
- updated-before
  - gitlab-group-issue-list command line option, [550](#)
  - gitlab-group-merge-request-list command line option, [558](#)
  - gitlab-issue-list command line option, [576](#)
  - gitlab-merge-request-list command line option, [580](#)
  - gitlab-project-deployment-list command line option, [635](#)
  - gitlab-project-deployment-merge-request-list command line option, [637](#)
  - gitlab-project-issue-list command line option, [656](#)
  - gitlab-project-merge-request-list command line option, [693](#)
- url
  - gitlab-geo-node-update command line option, [505](#)
  - gitlab-group-hook-create command line option, [543](#)
  - gitlab-group-hook-update command line option, [545](#)
  - gitlab-hook-create command line option, [574](#)
  - gitlab-project-hook-create command line option, [646](#)
  - gitlab-project-hook-update command line option, [648](#)
  - gitlab-project-release-link-create command line option, [754](#)
  - gitlab-project-release-link-update command line option, [756](#)
  - gitlab-project-remote-mirror-create command line option, [757](#)
- use-custom-template
  - gitlab-project-create command line option, [589](#)
- user-agent
  - gitlab command line option, [481](#)
- user-id
  - gitlab-group-billable-member-membership-list command line option, [521](#)
  - gitlab-group-invitation-create command line option, [547](#)
  - gitlab-group-member-create command line option, [555](#)
  - gitlab-personal-access-token-list command line option, [584](#)
  - gitlab-project-invitation-create command line option, [651](#)
  - gitlab-project-member-create command line option, [686](#)
  - gitlab-starred-project-list command line option, [793](#)
  - gitlab-user-custom-attribute-delete command line option, [805](#)
  - gitlab-user-custom-attribute-get command line option, [805](#)
  - gitlab-user-custom-attribute-list command line option, [805](#)
  - gitlab-user-email-create command line option, [806](#)
  - gitlab-user-email-delete command line option, [806](#)
  - gitlab-user-email-get command line option, [806](#)
  - gitlab-user-email-list command line option, [807](#)
  - gitlab-user-event-list command line option, [807](#)
  - gitlab-user-gpg-key-create command line option, [808](#)
  - gitlab-user-gpg-key-delete command line option, [808](#)
  - gitlab-user-gpg-key-get command line option, [808](#)
  - gitlab-user-gpg-key-list command line option, [809](#)
  - gitlab-user-impersonation-token-create command line option, [809](#)
  - gitlab-user-impersonation-token-delete command line option, [810](#)
  - gitlab-user-impersonation-token-get command line option, [810](#)

- gitlab-user-impersonation-token-list
  - command line option, [810](#)
- gitlab-user-key-create command line option, [811](#)
- gitlab-user-key-delete command line option, [811](#)
- gitlab-user-key-get command line option, [811](#)
- gitlab-user-key-list command line option, [812](#)
- gitlab-user-membership-get command line option, [812](#)
- gitlab-user-membership-list command line option, [812](#)
- gitlab-user-personal-access-token-create command line option, [813](#)
- gitlab-user-project-create command line option, [814](#)
- gitlab-user-project-list command line option, [815](#)
- gitlab-user-status-get command line option, [816](#)
- user-ids
  - gitlab-project-approval-rule-create command line option, [607](#)
  - gitlab-project-merge-request-approval-rule-create command line option, [701](#)
  - gitlab-project-merge-request-approval-rule-update command line option, [703](#)
- user-oauth-applications
  - gitlab-application-settings-update command line option, [486](#)
- user-show-add-ssh-key-message
  - gitlab-application-settings-update command line option, [487](#)
- username
  - gitlab-group-deploy-token-create command line option, [528](#)
  - gitlab-project-deploy-token-create command line option, [632](#)
  - gitlab-project-pipeline-list command line option, [733](#)
  - gitlab-user-create command line option, [799](#)
  - gitlab-user-list command line option, [801](#)
  - gitlab-user-update command line option, [803](#)
- usernames
  - gitlab-project-approval-rule-create command line option, [607](#)
- value
  - gitlab-group-variable-create command line option, [571](#)
  - gitlab-group-variable-update command line option, [572](#)
  - gitlab-project-pipeline-schedule-variable-create command line option, [739](#)
  - gitlab-project-pipeline-schedule-variable-update command line option, [739](#)
  - gitlab-project-variable-create command line option, [781](#)
  - gitlab-project-variable-update command line option, [782](#)
  - gitlab-variable-create command line option, [817](#)
  - gitlab-variable-update command line option, [818](#)
- variable-type
  - gitlab-group-variable-create command line option, [571](#)
  - gitlab-group-variable-update command line option, [572](#)
  - gitlab-project-variable-create command line option, [781](#)
  - gitlab-project-variable-update command line option, [782](#)
  - gitlab-variable-create command line option, [817](#)
  - gitlab-variable-update command line option, [818](#)
- verbose
  - gitlab command line option, [480](#)
- version
  - gitlab command line option, [480](#)
- view
  - gitlab-group-merge-request-list command line option, [558](#)
  - gitlab-merge-request-list command line option, [580](#)
  - gitlab-project-deployment-merge-request-list command line option, [636](#)
  - gitlab-project-merge-request-list command line option, [693](#)
- visibility
  - gitlab-group-create command line option, [508](#)
  - gitlab-group-project-list command line option, [564](#)
  - gitlab-group-update command line option, [512](#)
  - gitlab-project-create command line option, [589](#)
  - gitlab-project-fork-list command line option, [645](#)
  - gitlab-project-list command line option, [592](#)
  - gitlab-project-snippet-create command line option, [765](#)

```

gitlab-project-snippet-update command
  line option, 767
gitlab-project-update command line
  option, 602
gitlab-shared-project-list command line
  option, 790
gitlab-snippet-create command line
  option, 791
gitlab-snippet-update command line
  option, 793
gitlab-starred-project-list command
  line option, 794
gitlab-user-project-create command line
  option, 814
gitlab-user-project-list command line
  option, 815
--wall-enabled
  gitlab-user-project-create command line
    option, 814
--website-url
  gitlab-user-create command line option,
    800
  gitlab-user-update command line option,
    803
--wiki
  gitlab-project-snapshot command line
    option, 595
--wiki-access-level
  gitlab-project-create command line
    option, 589
  gitlab-project-update command line
    option, 602
--wiki-checksum-failed
  gitlab-project-list command line option,
    592
  gitlab-user-project-list command line
    option, 815
--wiki-enabled
  gitlab-project-create command line
    option, 589
  gitlab-project-update command line
    option, 602
  gitlab-user-project-create command line
    option, 814
--wiki-events
  gitlab-project-hook-update command line
    option, 648
--wiki-page-events
  gitlab-group-hook-create command line
    option, 544
  gitlab-group-hook-update command line
    option, 546
  gitlab-project-hook-create command line
    option, 646
--wip
  gitlab-group-merge-request-list command
    line option, 559
  gitlab-merge-request-list command line
    option, 580
  gitlab-project-deployment-merge-request-list
    command line option, 637
  gitlab-project-merge-request-list
    command line option, 694
--with-content
  gitlab-group-wiki-list command line
    option, 573
  gitlab-project-wiki-list command line
    option, 784
--with-custom-attributes
  gitlab-group-descendant-group-list
    command line option, 530
  gitlab-group-list command line option,
    510
  gitlab-group-project-list command line
    option, 565
  gitlab-group-subgroup-list command line
    option, 570
  gitlab-project-fork-list command line
    option, 645
  gitlab-project-list command line option,
    592
  gitlab-shared-project-list command line
    option, 790
  gitlab-starred-project-list command
    line option, 794
  gitlab-user-project-list command line
    option, 815
--with-issues-enabled
  gitlab-group-project-list command line
    option, 565
  gitlab-project-fork-list command line
    option, 645
  gitlab-project-list command line option,
    592
  gitlab-shared-project-list command line
    option, 790
  gitlab-starred-project-list command
    line option, 794
  gitlab-user-project-list command line
    option, 815
--with-labels-details
  gitlab-merge-request-list command line
    option, 580
  gitlab-project-deployment-merge-request-list
    command line option, 636
--with-merge-requests-enabled
  gitlab-group-project-list command line
    option, 565

```

- gitlab-project-fork-list command line option, [645](#)
- gitlab-project-list command line option, [592](#)
- gitlab-shared-project-list command line option, [790](#)
- gitlab-starred-project-list command line option, [794](#)
- gitlab-user-project-list command line option, [815](#)
- with-merge-status-recheck
  - gitlab-merge-request-list command line option, [580](#)
  - gitlab-project-deployment-merge-request-list command line option, [636](#)
- with-programming-language
  - gitlab-project-list command line option, [592](#)
  - gitlab-user-project-list command line option, [815](#)
- with-security-reports
  - gitlab-group-project-list command line option, [565](#)
- with-shared
  - gitlab-group-project-list command line option, [565](#)
- with-stats
  - gitlab-project-commit-list command line option, [623](#)
- yaml-errors
  - gitlab-project-pipeline-list command line option, [733](#)
- c
  - gitlab command line option, [480](#)
- d
  - gitlab command line option, [480](#)
- f
  - gitlab command line option, [480](#)
- g
  - gitlab command line option, [480](#)
- h
  - gitlab command line option, [480](#)
  - gitlab-application command line option, [481](#)
  - gitlab-application-appearance command line option, [482](#)
  - gitlab-application-appearance-get command line option, [482](#)
  - gitlab-application-appearance-update command line option, [483](#)
  - gitlab-application-create command line option, [481](#)
  - gitlab-application-delete command line option, [482](#)
  - gitlab-application-list command line option, [482](#)
  - gitlab-application-settings command line option, [483](#)
  - gitlab-application-settings-get command line option, [484](#)
  - gitlab-application-settings-update command line option, [485](#)
  - gitlab-application-statistics command line option, [487](#)
  - gitlab-application-statistics-get command line option, [488](#)
  - gitlab-audit-event command line option, [488](#)
  - gitlab-audit-event-get command line option, [488](#)
  - gitlab-audit-event-list command line option, [488](#)
  - gitlab-broadcast-message command line option, [489](#)
  - gitlab-broadcast-message-create command line option, [489](#)
  - gitlab-broadcast-message-delete command line option, [489](#)
  - gitlab-broadcast-message-get command line option, [490](#)
  - gitlab-broadcast-message-list command line option, [490](#)
  - gitlab-broadcast-message-update command line option, [490](#)
  - gitlab-bulk-import command line option, [491](#)
  - gitlab-bulk-import-all-entity command line option, [492](#)
  - gitlab-bulk-import-all-entity-list command line option, [492](#)
  - gitlab-bulk-import-create command line option, [491](#)
  - gitlab-bulk-import-entity command line option, [492](#)
  - gitlab-bulk-import-entity-get command line option, [492](#)
  - gitlab-bulk-import-entity-list command line option, [493](#)
  - gitlab-bulk-import-get command line option, [491](#)
  - gitlab-bulk-import-list command line option, [491](#)
  - gitlab-ci-lint command line option, [493](#)
  - gitlab-ci-lint-create command line option, [493](#)
  - gitlab-ci-lint-validate command line option, [494](#)
  - gitlab-current-user command line option,

- 494
- gitlab-current-user-email command line option, 494
- gitlab-current-user-email-create command line option, 494
- gitlab-current-user-email-delete command line option, 495
- gitlab-current-user-email-get command line option, 495
- gitlab-current-user-email-list command line option, 495
- gitlab-current-user-get command line option, 494
- gitlab-current-user-gpg-key command line option, 495
- gitlab-current-user-gpg-key-create command line option, 496
- gitlab-current-user-gpg-key-delete command line option, 496
- gitlab-current-user-gpg-key-get command line option, 496
- gitlab-current-user-gpg-key-list command line option, 496
- gitlab-current-user-key command line option, 497
- gitlab-current-user-key-create command line option, 497
- gitlab-current-user-key-delete command line option, 497
- gitlab-current-user-key-get command line option, 497
- gitlab-current-user-key-list command line option, 498
- gitlab-current-user-runner command line option, 498
- gitlab-current-user-runner-create command line option, 498
- gitlab-current-user-status command line option, 499
- gitlab-current-user-status-get command line option, 499
- gitlab-current-user-status-update command line option, 499
- gitlab-deploy-key command line option, 499
- gitlab-deploy-key-list command line option, 500
- gitlab-deploy-token command line option, 500
- gitlab-deploy-token-list command line option, 500
- gitlab-dockerfile command line option, 500
- gitlab-dockerfile-get command line option, 501
- gitlab-dockerfile-list command line option, 501
- gitlab-event command line option, 501
- gitlab-event-list command line option, 501
- gitlab-feature command line option, 502
- gitlab-feature-delete command line option, 502
- gitlab-feature-list command line option, 502
- gitlab-generic-package command line option, 503
- gitlab-generic-package-download command line option, 503
- gitlab-generic-package-upload command line option, 503
- gitlab-geo-node command line option, 504
- gitlab-geo-node-current-failures command line option, 504
- gitlab-geo-node-delete command line option, 504
- gitlab-geo-node-get command line option, 504
- gitlab-geo-node-list command line option, 505
- gitlab-geo-node-repair command line option, 505
- gitlab-geo-node-status command line option, 505
- gitlab-geo-node-update command line option, 505
- gitlab-gitignore command line option, 506
- gitlab-gitignore-get command line option, 506
- gitlab-gitignore-list command line option, 506
- gitlab-gitlabciyaml command line option, 506
- gitlab-gitlabciyaml-get command line option, 507
- gitlab-gitlabciyaml-list command line option, 507
- gitlab-group command line option, 507
- gitlab-group-access-request command line option, 513
- gitlab-group-access-request-approve command line option, 513
- gitlab-group-access-request-create command line option, 514
- gitlab-group-access-request-delete command line option, 514
- gitlab-group-access-request-list command line option, 514

`gitlab-group-access-token` command line option, [515](#)  
`gitlab-group-access-token-create` command line option, [515](#)  
`gitlab-group-access-token-delete` command line option, [515](#)  
`gitlab-group-access-token-get` command line option, [515](#)  
`gitlab-group-access-token-list` command line option, [516](#)  
`gitlab-group-audit-event` command line option, [516](#)  
`gitlab-group-audit-event-get` command line option, [516](#)  
`gitlab-group-audit-event-list` command line option, [517](#)  
`gitlab-group-badge` command line option, [517](#)  
`gitlab-group-badge-create` command line option, [517](#)  
`gitlab-group-badge-delete` command line option, [518](#)  
`gitlab-group-badge-get` command line option, [518](#)  
`gitlab-group-badge-list` command line option, [518](#)  
`gitlab-group-badge-render` command line option, [519](#)  
`gitlab-group-badge-update` command line option, [519](#)  
`gitlab-group-billable-member` command line option, [519](#)  
`gitlab-group-billable-member-delete` command line option, [519](#)  
`gitlab-group-billable-member-list` command line option, [520](#)  
`gitlab-group-billable-member-membership` command line option, [520](#)  
`gitlab-group-billable-member-membership-list` command line option, [520](#)  
`gitlab-group-board` command line option, [521](#)  
`gitlab-group-board-create` command line option, [521](#)  
`gitlab-group-board-delete` command line option, [521](#)  
`gitlab-group-board-get` command line option, [522](#)  
`gitlab-group-board-list` command line option, [522](#), [523](#)  
`gitlab-group-board-list-create` command line option, [523](#)  
`gitlab-group-board-list-delete` command line option, [523](#)  
`gitlab-group-board-list-get` command line option, [524](#)  
`gitlab-group-board-list-list` command line option, [524](#)  
`gitlab-group-board-list-update` command line option, [524](#)  
`gitlab-group-board-update` command line option, [522](#)  
`gitlab-group-cluster` command line option, [525](#)  
`gitlab-group-cluster-create` command line option, [525](#)  
`gitlab-group-cluster-delete` command line option, [525](#)  
`gitlab-group-cluster-get` command line option, [526](#)  
`gitlab-group-cluster-list` command line option, [526](#)  
`gitlab-group-cluster-update` command line option, [526](#)  
`gitlab-group-create` command line option, [508](#)  
`gitlab-group-custom-attribute` command line option, [527](#)  
`gitlab-group-custom-attribute-delete` command line option, [527](#)  
`gitlab-group-custom-attribute-get` command line option, [527](#)  
`gitlab-group-custom-attribute-list` command line option, [527](#)  
`gitlab-group-delete` command line option, [509](#)  
`gitlab-group-deploy-token` command line option, [528](#)  
`gitlab-group-deploy-token-create` command line option, [528](#)  
`gitlab-group-deploy-token-delete` command line option, [528](#)  
`gitlab-group-deploy-token-get` command line option, [529](#)  
`gitlab-group-deploy-token-list` command line option, [529](#)  
`gitlab-group-descendant-group` command line option, [529](#)  
`gitlab-group-descendant-group-list` command line option, [530](#)  
`gitlab-group-epic` command line option, [530](#)  
`gitlab-group-epic-award-emoji` command line option, [533](#)  
`gitlab-group-epic-award-emoji-create` command line option, [533](#)  
`gitlab-group-epic-award-emoji-delete` command line option, [533](#)

- gitlab-group-epic-award-emoji-get  
command line option, [533](#)
- gitlab-group-epic-award-emoji-list  
command line option, [534](#)
- gitlab-group-epic-create command line  
option, [531](#)
- gitlab-group-epic-delete command line  
option, [531](#)
- gitlab-group-epic-discussion-note  
command line option, [534](#)
- gitlab-group-epic-discussion-note-create  
command line option, [534](#)
- gitlab-group-epic-discussion-note-delete  
command line option, [535](#)
- gitlab-group-epic-discussion-note-get  
command line option, [535](#)
- gitlab-group-epic-discussion-note-update  
command line option, [535](#)
- gitlab-group-epic-get command line  
option, [531](#)
- gitlab-group-epic-issue command line  
option, [536](#)
- gitlab-group-epic-issue-create command  
line option, [536](#)
- gitlab-group-epic-issue-delete command  
line option, [536](#)
- gitlab-group-epic-issue-list command  
line option, [537](#)
- gitlab-group-epic-issue-update command  
line option, [537](#)
- gitlab-group-epic-list command line  
option, [532](#)
- gitlab-group-epic-note command line  
option, [537](#)
- gitlab-group-epic-note-award-emoji  
command line option, [539](#)
- gitlab-group-epic-note-award-emoji-create  
command line option, [540](#)
- gitlab-group-epic-note-award-emoji-delete  
command line option, [540](#)
- gitlab-group-epic-note-award-emoji-get  
command line option, [540](#)
- gitlab-group-epic-note-award-emoji-list  
command line option, [541](#)
- gitlab-group-epic-note-create command  
line option, [538](#)
- gitlab-group-epic-note-delete command  
line option, [538](#)
- gitlab-group-epic-note-get command line  
option, [538](#)
- gitlab-group-epic-note-list command  
line option, [539](#)
- gitlab-group-epic-note-update command  
line option, [539](#)
- gitlab-group-epic-resource-label-event  
command line option, [541](#)
- gitlab-group-epic-resource-label-event-get  
command line option, [541](#)
- gitlab-group-epic-resource-label-event-list  
command line option, [542](#)
- gitlab-group-epic-update command line  
option, [532](#)
- gitlab-group-export command line option,  
[542](#)
- gitlab-group-export-create command line  
option, [542](#)
- gitlab-group-export-download command  
line option, [542](#)
- gitlab-group-export-get command line  
option, [543](#)
- gitlab-group-get command line option, [509](#)
- gitlab-group-hook command line option,  
[543](#)
- gitlab-group-hook-create command line  
option, [543](#)
- gitlab-group-hook-delete command line  
option, [544](#)
- gitlab-group-hook-get command line  
option, [544](#)
- gitlab-group-hook-list command line  
option, [545](#)
- gitlab-group-hook-update command line  
option, [545](#)
- gitlab-group-import command line option,  
[546](#)
- gitlab-group-import-get command line  
option, [546](#)
- gitlab-group-invitation command line  
option, [546](#)
- gitlab-group-invitation-create command  
line option, [547](#)
- gitlab-group-invitation-delete command  
line option, [547](#)
- gitlab-group-invitation-get command  
line option, [547](#)
- gitlab-group-invitation-list command  
line option, [548](#)
- gitlab-group-invitation-update command  
line option, [548](#)
- gitlab-group-issue command line option,  
[549](#)
- gitlab-group-issue-list command line  
option, [549](#)
- gitlab-group-issues-statistics command  
line option, [550](#)
- gitlab-group-issues-statistics-get  
command line option, [550](#)
- gitlab-group-iteration command line



- option, [550](#)
- gitlab-group-iteration-list command
  - line option, [550](#)
- gitlab-group-label command line option, [551](#)
- gitlab-group-label-create command line option, [551](#)
- gitlab-group-label-delete command line option, [552](#)
- gitlab-group-label-get command line option, [552](#)
- gitlab-group-label-list command line option, [552](#)
- gitlab-group-label-subscribe command line option, [553](#)
- gitlab-group-label-unsubscribe command line option, [553](#)
- gitlab-group-label-update command line option, [553](#)
- gitlab-group-ldap-group-link command line option, [554](#)
- gitlab-group-ldap-group-link-create command line option, [554](#)
- gitlab-group-ldap-group-link-delete command line option, [554](#)
- gitlab-group-ldap-group-link-list command line option, [554](#)
- gitlab-group-ldap-sync command line option, [509](#)
- gitlab-group-list command line option, [510](#)
- gitlab-group-member command line option, [555](#)
- gitlab-group-member-all command line option, [557](#)
- gitlab-group-member-all-get command line option, [557](#)
- gitlab-group-member-all-list command line option, [557](#)
- gitlab-group-member-create command line option, [555](#)
- gitlab-group-member-delete command line option, [555](#)
- gitlab-group-member-get command line option, [556](#)
- gitlab-group-member-list command line option, [556](#)
- gitlab-group-member-update command line option, [556](#)
- gitlab-group-merge-request command line option, [557](#)
- gitlab-group-merge-request-list command line option, [558](#)
- gitlab-group-milestone command line

- option, [559](#)
- gitlab-group-milestone-create command line option, [559](#)
- gitlab-group-milestone-delete command line option, [560](#)
- gitlab-group-milestone-get command line option, [560](#)
- gitlab-group-milestone-issues command line option, [560](#)
- gitlab-group-milestone-list command line option, [560](#)
- gitlab-group-milestone-merge-requests command line option, [561](#)
- gitlab-group-milestone-update command line option, [561](#)
- gitlab-group-notification-settings command line option, [562](#)
- gitlab-group-notification-settings-get command line option, [562](#)
- gitlab-group-notification-settings-update command line option, [562](#)
- gitlab-group-package command line option, [563](#)
- gitlab-group-package-list command line option, [563](#)
- gitlab-group-project command line option, [564](#)
- gitlab-group-project-list command line option, [564](#)
- gitlab-group-push-rules command line option, [565](#)
- gitlab-group-push-rules-create command line option, [565](#)
- gitlab-group-push-rules-delete command line option, [566](#)
- gitlab-group-push-rules-get command line option, [566](#)
- gitlab-group-push-rules-update command line option, [567](#)
- gitlab-group-restore command line option, [510](#)
- gitlab-group-runner command line option, [567](#)
- gitlab-group-runner-list command line option, [567](#)
- gitlab-group-saml-group-link command line option, [568](#)
- gitlab-group-saml-group-link-create command line option, [568](#)
- gitlab-group-saml-group-link-delete command line option, [568](#)
- gitlab-group-saml-group-link-get command line option, [569](#)
- gitlab-group-saml-group-link-list



- command line option, [569](#)
- gitlab-group-search command line option, [510](#)
- gitlab-group-share command line option, [511](#)
- gitlab-group-subgroup command line option, [569](#)
- gitlab-group-subgroup-list command line option, [570](#)
- gitlab-group-transfer command line option, [511](#)
- gitlab-group-transfer-project command line option, [511](#)
- gitlab-group-unshare command line option, [511](#)
- gitlab-group-update command line option, [512](#)
- gitlab-group-variable command line option, [570](#)
- gitlab-group-variable-create command line option, [570](#)
- gitlab-group-variable-delete command line option, [571](#)
- gitlab-group-variable-get command line option, [571](#)
- gitlab-group-variable-list command line option, [571](#)
- gitlab-group-variable-update command line option, [572](#)
- gitlab-group-wiki command line option, [572](#)
- gitlab-group-wiki-create command line option, [572](#)
- gitlab-group-wiki-delete command line option, [573](#)
- gitlab-group-wiki-get command line option, [573](#)
- gitlab-group-wiki-list command line option, [573](#)
- gitlab-group-wiki-update command line option, [574](#)
- gitlab-hook command line option, [574](#)
- gitlab-hook-create command line option, [574](#)
- gitlab-hook-delete command line option, [574](#)
- gitlab-hook-get command line option, [575](#)
- gitlab-hook-list command line option, [575](#)
- gitlab-issue command line option, [575](#)
- gitlab-issue-get command line option, [575](#)
- gitlab-issue-list command line option, [576](#)
- gitlab-issues-statistics command line option, [577](#)
- gitlab-issues-statistics-get command line option, [577](#)
- gitlab-key command line option, [577](#)
- gitlab-key-get command line option, [577](#)
- gitlab-ldap-group command line option, [577](#)
- gitlab-ldap-group-list command line option, [578](#)
- gitlab-license command line option, [578](#)
- gitlab-license-get command line option, [578](#)
- gitlab-license-list command line option, [579](#)
- gitlab-merge-request command line option, [579](#)
- gitlab-merge-request-list command line option, [580](#)
- gitlab-namespace command line option, [581](#)
- gitlab-namespace-exists command line option, [581](#)
- gitlab-namespace-get command line option, [581](#)
- gitlab-namespace-list command line option, [582](#)
- gitlab-notification-settings command line option, [582](#)
- gitlab-notification-settings-get command line option, [582](#)
- gitlab-notification-settings-update command line option, [582](#)
- gitlab-pages-domain command line option, [583](#)
- gitlab-pages-domain-list command line option, [583](#)
- gitlab-personal-access-token command line option, [584](#)
- gitlab-personal-access-token-delete command line option, [584](#)
- gitlab-personal-access-token-get command line option, [584](#)
- gitlab-personal-access-token-list command line option, [584](#)
- gitlab-project command line option, [585](#)
- gitlab-project-access-request command line option, [602](#)
- gitlab-project-access-request-approve command line option, [603](#)
- gitlab-project-access-request-create command line option, [603](#)
- gitlab-project-access-request-delete command line option, [603](#)
- gitlab-project-access-request-list command line option, [603](#)
- gitlab-project-access-token command

line option, [604](#)  
gitlab-project-access-token-create  
command line option, [604](#)  
gitlab-project-access-token-delete  
command line option, [604](#)  
gitlab-project-access-token-get command  
line option, [605](#)  
gitlab-project-access-token-list  
command line option, [605](#)  
gitlab-project-additional-statistics  
command line option, [605](#)  
gitlab-project-additional-statistics-get  
command line option, [605](#)  
gitlab-project-approval command line  
option, [606](#)  
gitlab-project-approval-get command  
line option, [606](#)  
gitlab-project-approval-rule command  
line option, [607](#)  
gitlab-project-approval-rule-create  
command line option, [607](#)  
gitlab-project-approval-rule-delete  
command line option, [607](#)  
gitlab-project-approval-rule-list  
command line option, [608](#)  
gitlab-project-approval-rule-update  
command line option, [608](#)  
gitlab-project-approval-update command  
line option, [606](#)  
gitlab-project-archive command line  
option, [585](#)  
gitlab-project-artifact command line  
option, [608](#)  
gitlab-project-artifact-delete command  
line option, [608](#)  
gitlab-project-artifact-download  
command line option, [609](#)  
gitlab-project-artifact-raw command  
line option, [609](#)  
gitlab-project-audit-event command line  
option, [609](#)  
gitlab-project-audit-event-get command  
line option, [610](#)  
gitlab-project-audit-event-list command  
line option, [610](#)  
gitlab-project-badge command line  
option, [610](#)  
gitlab-project-badge-create command  
line option, [611](#)  
gitlab-project-badge-delete command  
line option, [611](#)  
gitlab-project-badge-get command line  
option, [611](#)  
gitlab-project-badge-list command line  
option, [612](#)  
gitlab-project-badge-render command  
line option, [612](#)  
gitlab-project-badge-update command  
line option, [612](#)  
gitlab-project-board command line  
option, [613](#)  
gitlab-project-board-create command  
line option, [613](#)  
gitlab-project-board-delete command  
line option, [613](#)  
gitlab-project-board-get command line  
option, [613](#)  
gitlab-project-board-list command line  
option, [614](#)  
gitlab-project-board-list-create  
command line option, [614](#)  
gitlab-project-board-list-delete  
command line option, [615](#)  
gitlab-project-board-list-get command  
line option, [615](#)  
gitlab-project-board-list-list command  
line option, [615](#)  
gitlab-project-board-list-update  
command line option, [616](#)  
gitlab-project-board-update command  
line option, [614](#)  
gitlab-project-branch command line  
option, [616](#)  
gitlab-project-branch-create command  
line option, [616](#)  
gitlab-project-branch-delete command  
line option, [617](#)  
gitlab-project-branch-get command line  
option, [617](#)  
gitlab-project-branch-list command line  
option, [617](#)  
gitlab-project-ci-lint command line  
option, [618](#)  
gitlab-project-ci-lint-create command  
line option, [618](#)  
gitlab-project-ci-lint-get command line  
option, [618](#)  
gitlab-project-ci-lint-validate command  
line option, [619](#)  
gitlab-project-cluster command line  
option, [619](#)  
gitlab-project-cluster-create command  
line option, [619](#)  
gitlab-project-cluster-delete command  
line option, [620](#)  
gitlab-project-cluster-get command line  
option, [620](#)  
gitlab-project-cluster-list command

- line option, [620](#)
- gitlab-project-cluster-update command
  - line option, [621](#)
- gitlab-project-commit command line
  - option, [621](#)
- gitlab-project-commit-cherry-pick
  - command line option, [622](#)
- gitlab-project-commit-comment command
  - line option, [625](#)
- gitlab-project-commit-comment-create
  - command line option, [625](#)
- gitlab-project-commit-comment-list
  - command line option, [626](#)
- gitlab-project-commit-create command
  - line option, [622](#)
- gitlab-project-commit-diff command line
  - option, [622](#)
- gitlab-project-commit-discussion
  - command line option, [626](#)
- gitlab-project-commit-discussion-create
  - command line option, [626](#)
- gitlab-project-commit-discussion-get
  - command line option, [627](#)
- gitlab-project-commit-discussion-list
  - command line option, [627](#)
- gitlab-project-commit-discussion-note
  - command line option, [627](#)
- gitlab-project-commit-discussion-note-create
  - command line option, [628](#)
- gitlab-project-commit-discussion-note-delete
  - command line option, [628](#)
- gitlab-project-commit-discussion-note-get
  - command line option, [629](#)
- gitlab-project-commit-discussion-note-update
  - command line option, [629](#)
- gitlab-project-commit-get command line
  - option, [623](#)
- gitlab-project-commit-list command line
  - option, [623](#)
- gitlab-project-commit-merge-requests
  - command line option, [624](#)
- gitlab-project-commit-refs command line
  - option, [624](#)
- gitlab-project-commit-revert command
  - line option, [624](#)
- gitlab-project-commit-signature command
  - line option, [625](#)
- gitlab-project-commit-status command
  - line option, [629](#)
- gitlab-project-commit-status-create
  - command line option, [630](#)
- gitlab-project-commit-status-list
  - command line option, [630](#)
- gitlab-project-create command line
  - option, [587](#)
- gitlab-project-create-fork-relation
  - command line option, [590](#)
- gitlab-project-custom-attribute command
  - line option, [631](#)
- gitlab-project-custom-attribute-delete
  - command line option, [631](#)
- gitlab-project-custom-attribute-get
  - command line option, [631](#)
- gitlab-project-custom-attribute-list
  - command line option, [631](#)
- gitlab-project-delete command line
  - option, [590](#)
- gitlab-project-delete-fork-relation
  - command line option, [590](#)
- gitlab-project-delete-merged-branches
  - command line option, [590](#)
- gitlab-project-deploy-token command
  - line option, [632](#)
- gitlab-project-deploy-token-create
  - command line option, [632](#)
- gitlab-project-deploy-token-delete
  - command line option, [632](#)
- gitlab-project-deploy-token-get command
  - line option, [632](#)
- gitlab-project-deploy-token-list
  - command line option, [633](#)
- gitlab-project-deployment command line
  - option, [633](#)
- gitlab-project-deployment-approval
  - command line option, [633](#)
- gitlab-project-deployment-create
  - command line option, [634](#)
- gitlab-project-deployment-get command
  - line option, [634](#)
- gitlab-project-deployment-list command
  - line option, [634](#)
- gitlab-project-deployment-merge-request
  - command line option, [635](#)
- gitlab-project-deployment-merge-request-list
  - command line option, [636](#)
- gitlab-project-deployment-update
  - command line option, [635](#)
- gitlab-project-environment command line
  - option, [638](#)
- gitlab-project-environment-create
  - command line option, [638](#)
- gitlab-project-environment-delete
  - command line option, [638](#)
- gitlab-project-environment-get command
  - line option, [638](#)
- gitlab-project-environment-list command
  - line option, [639](#)
- gitlab-project-environment-stop command

- line option, [639](#)
- gitlab-project-environment-update
  - command line option, [639](#)
- gitlab-project-event command line option, [640](#)
- gitlab-project-event-list command line option, [640](#)
- gitlab-project-export command line option, [641](#)
- gitlab-project-export-create command line option, [641](#)
- gitlab-project-export-download command line option, [641](#)
- gitlab-project-export-get command line option, [641](#)
- gitlab-project-file command line option, [642](#)
- gitlab-project-file-blame command line option, [642](#)
- gitlab-project-file-create command line option, [642](#)
- gitlab-project-file-delete command line option, [643](#)
- gitlab-project-file-get command line option, [643](#)
- gitlab-project-file-raw command line option, [643](#)
- gitlab-project-file-update command line option, [644](#)
- gitlab-project-fork command line option, [644](#)
- gitlab-project-fork-create command line option, [644](#)
- gitlab-project-fork-list command line option, [645](#)
- gitlab-project-get command line option, [590](#)
- gitlab-project-hook command line option, [646](#)
- gitlab-project-hook-create command line option, [646](#)
- gitlab-project-hook-delete command line option, [647](#)
- gitlab-project-hook-get command line option, [647](#)
- gitlab-project-hook-list command line option, [647](#)
- gitlab-project-hook-update command line option, [648](#)
- gitlab-project-housekeeping command line option, [591](#)
- gitlab-project-import command line option, [649](#)
- gitlab-project-import-get command line

- option, [649](#)
- gitlab-project-integration command line option, [649](#)
- gitlab-project-integration-available
  - command line option, [649](#)
- gitlab-project-integration-delete
  - command line option, [649](#)
- gitlab-project-integration-get command line option, [650](#)
- gitlab-project-integration-list command line option, [650](#)
- gitlab-project-integration-update
  - command line option, [650](#)
- gitlab-project-invitation command line option, [651](#)
- gitlab-project-invitation-create
  - command line option, [651](#)
- gitlab-project-invitation-delete
  - command line option, [651](#)
- gitlab-project-invitation-get command line option, [652](#)
- gitlab-project-invitation-list command line option, [652](#)
- gitlab-project-invitation-update
  - command line option, [652](#)
- gitlab-project-issue command line option, [653](#)
- gitlab-project-issue-add-spent-time
  - command line option, [653](#)
- gitlab-project-issue-award-emoji
  - command line option, [661](#)
- gitlab-project-issue-award-emoji-create
  - command line option, [661](#)
- gitlab-project-issue-award-emoji-delete
  - command line option, [661](#)
- gitlab-project-issue-award-emoji-get
  - command line option, [662](#)
- gitlab-project-issue-award-emoji-list
  - command line option, [662](#)
- gitlab-project-issue-closed-by command line option, [654](#)
- gitlab-project-issue-create command line option, [654](#)
- gitlab-project-issue-delete command line option, [655](#)
- gitlab-project-issue-discussion command line option, [662](#)
- gitlab-project-issue-discussion-create
  - command line option, [663](#)
- gitlab-project-issue-discussion-get
  - command line option, [663](#)
- gitlab-project-issue-discussion-list
  - command line option, [663](#)
- gitlab-project-issue-discussion-note

- command line option, 664
- gitlab-project-issue-discussion-note-create
  - command line option, 664
- gitlab-project-issue-discussion-note-delete
  - command line option, 664
- gitlab-project-issue-discussion-note-get
  - command line option, 665
- gitlab-project-issue-discussion-note-update
  - command line option, 665
- gitlab-project-issue-get command line option, 655
- gitlab-project-issue-link command line option, 666
- gitlab-project-issue-link-create
  - command line option, 666
- gitlab-project-issue-link-delete
  - command line option, 666
- gitlab-project-issue-link-list command line option, 667
- gitlab-project-issue-list command line option, 655
- gitlab-project-issue-move command line option, 656
- gitlab-project-issue-note command line option, 667
- gitlab-project-issue-note-award-emoji
  - command line option, 669
- gitlab-project-issue-note-award-emoji-create
  - command line option, 669
- gitlab-project-issue-note-award-emoji-delete
  - command line option, 670
- gitlab-project-issue-note-award-emoji-get
  - command line option, 670
- gitlab-project-issue-note-award-emoji-list
  - command line option, 670
- gitlab-project-issue-note-create
  - command line option, 667
- gitlab-project-issue-note-delete
  - command line option, 668
- gitlab-project-issue-note-get command line option, 668
- gitlab-project-issue-note-list command line option, 668
- gitlab-project-issue-note-update
  - command line option, 669
- gitlab-project-issue-participants
  - command line option, 657
- gitlab-project-issue-related-merge-requests
  - command line option, 657
- gitlab-project-issue-reorder command line option, 657
- gitlab-project-issue-reset-spent-time
  - command line option, 658
- gitlab-project-issue-reset-time-estimate
  - command line option, 658
- gitlab-project-issue-resource-iteration-event
  - command line option, 671
- gitlab-project-issue-resource-iteration-event-get
  - command line option, 671
- gitlab-project-issue-resource-iteration-event-list
  - command line option, 671
- gitlab-project-issue-resource-label-event
  - command line option, 672
- gitlab-project-issue-resource-label-event-get
  - command line option, 672
- gitlab-project-issue-resource-label-event-list
  - command line option, 672
- gitlab-project-issue-resource-milestone-event
  - command line option, 673
- gitlab-project-issue-resource-milestone-event-get
  - command line option, 673
- gitlab-project-issue-resource-milestone-event-list
  - command line option, 673
- gitlab-project-issue-resource-state-event
  - command line option, 674
- gitlab-project-issue-resource-state-event-get
  - command line option, 674
- gitlab-project-issue-resource-state-event-list
  - command line option, 674
- gitlab-project-issue-resource-weight-event
  - command line option, 675
- gitlab-project-issue-resource-weight-event-get
  - command line option, 675
- gitlab-project-issue-resource-weight-event-list
  - command line option, 675
- gitlab-project-issue-subscribe command line option, 658
- gitlab-project-issue-time-estimate
  - command line option, 658
- gitlab-project-issue-time-stats command line option, 659
- gitlab-project-issue-todo command line option, 659
- gitlab-project-issue-unsubscribe
  - command line option, 659
- gitlab-project-issue-update command line option, 660
- gitlab-project-issue-user-agent-detail
  - command line option, 660
- gitlab-project-issues-statistics
  - command line option, 676
- gitlab-project-issues-statistics-get
  - command line option, 676
- gitlab-project-job command line option, 676
- gitlab-project-job-artifact command line option, 677
- gitlab-project-job-artifacts command

- line option, [677](#)
- gitlab-project-job-cancel command line option, [677](#)
- gitlab-project-job-delete-artifacts command line option, [677](#)
- gitlab-project-job-erase command line option, [678](#)
- gitlab-project-job-get command line option, [678](#)
- gitlab-project-job-keep-artifacts command line option, [678](#)
- gitlab-project-job-list command line option, [679](#)
- gitlab-project-job-play command line option, [679](#)
- gitlab-project-job-retry command line option, [679](#)
- gitlab-project-job-token-scope command line option, [680](#)
- gitlab-project-job-token-scope-get command line option, [680](#)
- gitlab-project-job-token-scope-update command line option, [680](#)
- gitlab-project-job-trace command line option, [680](#)
- gitlab-project-key command line option, [681](#)
- gitlab-project-key-create command line option, [681](#)
- gitlab-project-key-delete command line option, [681](#)
- gitlab-project-key-enable command line option, [681](#)
- gitlab-project-key-get command line option, [682](#)
- gitlab-project-key-list command line option, [682](#)
- gitlab-project-key-update command line option, [682](#)
- gitlab-project-label command line option, [683](#)
- gitlab-project-label-create command line option, [683](#)
- gitlab-project-label-delete command line option, [683](#)
- gitlab-project-label-get command line option, [684](#)
- gitlab-project-label-list command line option, [684](#)
- gitlab-project-label-subscribe command line option, [684](#)
- gitlab-project-label-unsubscribe command line option, [685](#)
- gitlab-project-label-update command

- line option, [685](#)
- gitlab-project-languages command line option, [591](#)
- gitlab-project-list command line option, [591](#)
- gitlab-project-member command line option, [685](#)
- gitlab-project-member-all command line option, [687](#)
- gitlab-project-member-all-get command line option, [688](#)
- gitlab-project-member-all-list command line option, [688](#)
- gitlab-project-member-create command line option, [686](#)
- gitlab-project-member-delete command line option, [686](#)
- gitlab-project-member-get command line option, [686](#)
- gitlab-project-member-list command line option, [687](#)
- gitlab-project-member-update command line option, [687](#)
- gitlab-project-merge-request command line option, [689](#)
- gitlab-project-merge-request-add-spent-time command line option, [689](#)
- gitlab-project-merge-request-approval command line option, [699](#)
- gitlab-project-merge-request-approval-get command line option, [699](#)
- gitlab-project-merge-request-approval-rule command line option, [700](#)
- gitlab-project-merge-request-approval-rule-create command line option, [700](#)
- gitlab-project-merge-request-approval-rule-delete command line option, [701](#)
- gitlab-project-merge-request-approval-rule-get command line option, [701](#)
- gitlab-project-merge-request-approval-rule-list command line option, [702](#)
- gitlab-project-merge-request-approval-rule-update command line option, [702](#)
- gitlab-project-merge-request-approval-state command line option, [703](#)
- gitlab-project-merge-request-approval-state-get command line option, [703](#)
- gitlab-project-merge-request-approval-update command line option, [700](#)
- gitlab-project-merge-request-approve command line option, [689](#)
- gitlab-project-merge-request-award-emoji command line option, [703](#)
- gitlab-project-merge-request-award-emoji-create



command line option, 704	command line option, 712
gitlab-project-merge-request-award-emoji-delete	gitlab-project-merge-request-draft-note-update
command line option, 704	command line option, 712
gitlab-project-merge-request-award-emoji-get	gitlab-project-merge-request-get
command line option, 704	command line option, 692
gitlab-project-merge-request-award-emoji-list	gitlab-project-merge-request-list
command line option, 705	command line option, 693
gitlab-project-merge-request-cancel-merge-when-pipeline-fails	gitlab-project-merge-request-merge
command line option, 690	command line option, 694
gitlab-project-merge-request-changes	gitlab-project-merge-request-merge-ref
command line option, 690	command line option, 694
gitlab-project-merge-request-closes-issues	gitlab-project-merge-request-note
command line option, 690	command line option, 712
gitlab-project-merge-request-commits	gitlab-project-merge-request-note-award-emoji
command line option, 691	command line option, 714
gitlab-project-merge-request-create	gitlab-project-merge-request-note-award-emoji-create
command line option, 691	command line option, 715
gitlab-project-merge-request-delete	gitlab-project-merge-request-note-award-emoji-delete
command line option, 692	command line option, 715
gitlab-project-merge-request-diff	gitlab-project-merge-request-note-award-emoji-get
command line option, 705	command line option, 715
gitlab-project-merge-request-diff-get	gitlab-project-merge-request-note-award-emoji-list
command line option, 705	command line option, 716
gitlab-project-merge-request-diff-list	gitlab-project-merge-request-note-create
command line option, 706	command line option, 713
gitlab-project-merge-request-discussion	gitlab-project-merge-request-note-delete
command line option, 706	command line option, 713
gitlab-project-merge-request-discussion-create	gitlab-project-merge-request-note-get
command line option, 706	command line option, 713
gitlab-project-merge-request-discussion-get	gitlab-project-merge-request-note-list
command line option, 707	command line option, 714
gitlab-project-merge-request-discussion-list	gitlab-project-merge-request-note-update
command line option, 707	command line option, 714
gitlab-project-merge-request-discussion-note	gitlab-project-merge-request-participants
command line option, 708	command line option, 695
gitlab-project-merge-request-discussion-note-create	gitlab-project-merge-request-pipeline
command line option, 708	command line option, 716
gitlab-project-merge-request-discussion-note-delete	gitlab-project-merge-request-pipeline-create
command line option, 709	command line option, 716
gitlab-project-merge-request-discussion-note-get	gitlab-project-merge-request-pipeline-list
command line option, 709	command line option, 717
gitlab-project-merge-request-discussion-note-update	gitlab-project-merge-request-rebase
command line option, 710	command line option, 695
gitlab-project-merge-request-discussion-update	gitlab-project-merge-request-reset-approvals
command line option, 708	command line option, 695
gitlab-project-merge-request-draft-note	gitlab-project-merge-request-reset-spent-time
command line option, 710	command line option, 696
gitlab-project-merge-request-draft-note-create	gitlab-project-merge-request-reset-time-estimate
command line option, 710	command line option, 696
gitlab-project-merge-request-draft-note-delete	gitlab-project-merge-request-resource-label-event
command line option, 711	command line option, 717
gitlab-project-merge-request-draft-note-get	gitlab-project-merge-request-resource-label-event-get
command line option, 711	command line option, 717
gitlab-project-merge-request-draft-note-list	gitlab-project-merge-request-resource-label-event-list

- command line option, [718](#)
- gitlab-project-merge-request-resource-milestones
  - command line option, [718](#)
- gitlab-project-merge-request-resource-milestones-get
  - command line option, [718](#)
- gitlab-project-merge-request-resource-milestones-list
  - command line option, [719](#)
- gitlab-project-merge-request-resource-state-events
  - command line option, [719](#)
- gitlab-project-merge-request-resource-state-events-get
  - command line option, [719](#)
- gitlab-project-merge-request-resource-state-events-list
  - command line option, [720](#)
- gitlab-project-merge-request-reviewer-detail
  - command line option, [720](#)
- gitlab-project-merge-request-reviewer-detail-list
  - command line option, [720](#)
- gitlab-project-merge-request-subscribe
  - command line option, [696](#)
- gitlab-project-merge-request-time-estimate
  - command line option, [697](#)
- gitlab-project-merge-request-time-stats
  - command line option, [697](#)
- gitlab-project-merge-request-todo
  - command line option, [697](#)
- gitlab-project-merge-request-unapprove
  - command line option, [698](#)
- gitlab-project-merge-request-unsubscribe
  - command line option, [698](#)
- gitlab-project-merge-request-update
  - command line option, [698](#)
- gitlab-project-merge-train
  - command line option, [721](#)
- gitlab-project-merge-train-list
  - command line option, [721](#)
- gitlab-project-milestone
  - command line option, [721](#)
- gitlab-project-milestone-create
  - command line option, [722](#)
- gitlab-project-milestone-delete
  - command line option, [722](#)
- gitlab-project-milestone-get
  - command line option, [722](#)
- gitlab-project-milestone-issues
  - command line option, [723](#)
- gitlab-project-milestone-list
  - command line option, [723](#)
- gitlab-project-milestone-merge-requests
  - command line option, [723](#)
- gitlab-project-milestone-update
  - command line option, [724](#)
- gitlab-project-mirror-pull
  - command line option, [592](#)
- gitlab-project-mirror-pull-details
  - command line option, [593](#)
- gitlab-project-note
  - command line option, [724](#)
- gitlab-project-note-get
  - command line option, [724](#)
- gitlab-project-note-list
  - command line option, [725](#)
- gitlab-project-notification-settings
  - command line option, [725](#)
- gitlab-project-notification-settings-get
  - command line option, [725](#)
- gitlab-project-notification-settings-update
  - command line option, [726](#)
- gitlab-project-package
  - command line option, [726](#)
- gitlab-project-package-delete
  - command line option, [727](#)
- gitlab-project-package-file
  - command line option, [728](#)
- gitlab-project-package-file-delete
  - command line option, [728](#)
- gitlab-project-package-file-list
  - command line option, [728](#)
- gitlab-project-package-get
  - command line option, [727](#)
- gitlab-project-package-list
  - command line option, [727](#)
- gitlab-project-package-pipeline
  - command line option, [729](#)
- gitlab-project-package-pipeline-list
  - command line option, [729](#)
- gitlab-project-pages-domain
  - command line option, [729](#)
- gitlab-project-pages-domain-create
  - command line option, [729](#)
- gitlab-project-pages-domain-delete
  - command line option, [730](#)
- gitlab-project-pages-domain-get
  - command line option, [730](#)
- gitlab-project-pages-domain-list
  - command line option, [730](#)
- gitlab-project-pages-domain-update
  - command line option, [731](#)
- gitlab-project-pipeline
  - command line option, [731](#)
- gitlab-project-pipeline-bridge
  - command line option, [733](#)
- gitlab-project-pipeline-bridge-list
  - command line option, [733](#)
- gitlab-project-pipeline-cancel
  - command line option, [731](#)
- gitlab-project-pipeline-create
  - command line option, [731](#)
- gitlab-project-pipeline-delete
  - command line option, [731](#)



- line option, [732](#)
- gitlab-project-pipeline-get command line option, [732](#)
- gitlab-project-pipeline-job command line option, [734](#)
- gitlab-project-pipeline-job-list command line option, [734](#)
- gitlab-project-pipeline-list command line option, [732](#)
- gitlab-project-pipeline-retry command line option, [733](#)
- gitlab-project-pipeline-schedule command line option, [735](#)
- gitlab-project-pipeline-schedule-create command line option, [735](#)
- gitlab-project-pipeline-schedule-delete command line option, [735](#)
- gitlab-project-pipeline-schedule-get command line option, [736](#)
- gitlab-project-pipeline-schedule-list command line option, [736](#)
- gitlab-project-pipeline-schedule-pipeline command line option, [737](#)
- gitlab-project-pipeline-schedule-pipeline-list command line option, [738](#)
- gitlab-project-pipeline-schedule-play command line option, [736](#)
- gitlab-project-pipeline-schedule-take-ownership command line option, [737](#)
- gitlab-project-pipeline-schedule-update command line option, [737](#)
- gitlab-project-pipeline-schedule-variable command line option, [738](#)
- gitlab-project-pipeline-schedule-variable-create command line option, [738](#)
- gitlab-project-pipeline-schedule-variable-delete command line option, [739](#)
- gitlab-project-pipeline-schedule-variable-update command line option, [739](#)
- gitlab-project-pipeline-test-report command line option, [740](#)
- gitlab-project-pipeline-test-report-get command line option, [740](#)
- gitlab-project-pipeline-test-report-summary command line option, [740](#)
- gitlab-project-pipeline-test-report-summary-get command line option, [740](#)
- gitlab-project-pipeline-variable command line option, [741](#)
- gitlab-project-pipeline-variable-list command line option, [741](#)
- gitlab-project-protected-branch command line option, [741](#)
- gitlab-project-protected-branch-create command line option, [742](#)
- gitlab-project-protected-branch-delete command line option, [742](#)
- gitlab-project-protected-branch-get command line option, [743](#)
- gitlab-project-protected-branch-list command line option, [743](#)
- gitlab-project-protected-branch-update command line option, [743](#)
- gitlab-project-protected-environment command line option, [744](#)
- gitlab-project-protected-environment-create command line option, [744](#)
- gitlab-project-protected-environment-delete command line option, [744](#)
- gitlab-project-protected-environment-get command line option, [745](#)
- gitlab-project-protected-environment-list command line option, [745](#)
- gitlab-project-protected-tag command line option, [745](#)
- gitlab-project-protected-tag-create command line option, [746](#)
- gitlab-project-protected-tag-delete command line option, [746](#)
- gitlab-project-protected-tag-get command line option, [746](#)
- gitlab-project-protected-tag-list command line option, [747](#)
- gitlab-project-push-rules command line option, [747](#)
- gitlab-project-push-rules-create command line option, [747](#)
- gitlab-project-push-rules-delete command line option, [748](#)
- gitlab-project-push-rules-get command line option, [748](#)
- gitlab-project-push-rules-update command line option, [749](#)
- gitlab-project-registry-repository command line option, [749](#)
- gitlab-project-registry-repository-delete command line option, [750](#)
- gitlab-project-registry-repository-list command line option, [750](#)
- gitlab-project-registry-tag command line option, [750](#)
- gitlab-project-registry-tag-delete command line option, [750](#)
- gitlab-project-registry-tag-delete-in-bulk command line option, [751](#)
- gitlab-project-registry-tag-get command line option, [751](#)
- gitlab-project-registry-tag-list

- command line option, [752](#)
- gitlab-project-release command line option, [752](#)
- gitlab-project-release-create command line option, [752](#)
- gitlab-project-release-delete command line option, [753](#)
- gitlab-project-release-get command line option, [753](#)
- gitlab-project-release-link command line option, [754](#)
- gitlab-project-release-link-create command line option, [754](#)
- gitlab-project-release-link-delete command line option, [755](#)
- gitlab-project-release-link-get command line option, [755](#)
- gitlab-project-release-link-list command line option, [755](#)
- gitlab-project-release-link-update command line option, [756](#)
- gitlab-project-release-list command line option, [753](#)
- gitlab-project-release-update command line option, [754](#)
- gitlab-project-remote-mirror command line option, [756](#)
- gitlab-project-remote-mirror-create command line option, [757](#)
- gitlab-project-remote-mirror-delete command line option, [757](#)
- gitlab-project-remote-mirror-list command line option, [757](#)
- gitlab-project-remote-mirror-update command line option, [758](#)
- gitlab-project-repository-archive command line option, [593](#)
- gitlab-project-repository-blob command line option, [593](#)
- gitlab-project-repository-compare command line option, [593](#)
- gitlab-project-repository-contributors command line option, [594](#)
- gitlab-project-repository-merge-base command line option, [594](#)
- gitlab-project-repository-raw-blob command line option, [594](#)
- gitlab-project-repository-tree command line option, [594](#)
- gitlab-project-resource-group command line option, [758](#)
- gitlab-project-resource-group-get command line option, [758](#)
- gitlab-project-resource-group-list

- command line option, [758](#)
- gitlab-project-resource-group-upcoming-job command line option, [759](#)
- gitlab-project-resource-group-upcoming-job-list command line option, [759](#)
- gitlab-project-resource-group-update command line option, [759](#)
- gitlab-project-restore command line option, [595](#)
- gitlab-project-runner command line option, [760](#)
- gitlab-project-runner-create command line option, [760](#)
- gitlab-project-runner-delete command line option, [760](#)
- gitlab-project-runner-list command line option, [761](#)
- gitlab-project-search command line option, [595](#)
- gitlab-project-secure-file command line option, [761](#)
- gitlab-project-secure-file-create command line option, [761](#)
- gitlab-project-secure-file-delete command line option, [762](#)
- gitlab-project-secure-file-download command line option, [762](#)
- gitlab-project-secure-file-get command line option, [762](#)
- gitlab-project-secure-file-list command line option, [762](#)
- gitlab-project-service command line option, [763](#)
- gitlab-project-service-available command line option, [763](#)
- gitlab-project-service-delete command line option, [763](#)
- gitlab-project-service-get command line option, [764](#)
- gitlab-project-service-list command line option, [764](#)
- gitlab-project-service-update command line option, [764](#)
- gitlab-project-share command line option, [595](#)
- gitlab-project-snapshot command line option, [595](#)
- gitlab-project-snippet command line option, [765](#)
- gitlab-project-snippet-award-emoji command line option, [767](#)
- gitlab-project-snippet-award-emoji-create command line option, [768](#)
- gitlab-project-snippet-award-emoji-delete

- command line option, [768](#)
- gitlab-project-snippet-award-emoji-get
  - command line option, [768](#)
- gitlab-project-snippet-award-emoji-list
  - command line option, [769](#)
- gitlab-project-snippet-content command line option, [765](#)
- gitlab-project-snippet-create command line option, [765](#)
- gitlab-project-snippet-delete command line option, [766](#)
- gitlab-project-snippet-discussion
  - command line option, [769](#)
- gitlab-project-snippet-discussion-create
  - command line option, [769](#)
- gitlab-project-snippet-discussion-get
  - command line option, [770](#)
- gitlab-project-snippet-discussion-list
  - command line option, [770](#)
- gitlab-project-snippet-discussion-note
  - command line option, [770](#)
- gitlab-project-snippet-discussion-note-create
  - command line option, [771](#)
- gitlab-project-snippet-discussion-note-delete
  - command line option, [771](#)
- gitlab-project-snippet-discussion-note-get
  - command line option, [771](#)
- gitlab-project-snippet-discussion-note-update
  - command line option, [772](#)
- gitlab-project-snippet-get command line option, [766](#)
- gitlab-project-snippet-list command line option, [766](#)
- gitlab-project-snippet-note command line option, [772](#)
- gitlab-project-snippet-note-award-emoji
  - command line option, [774](#)
- gitlab-project-snippet-note-award-emoji-create
  - command line option, [774](#)
- gitlab-project-snippet-note-award-emoji-delete
  - command line option, [775](#)
- gitlab-project-snippet-note-award-emoji-get
  - command line option, [775](#)
- gitlab-project-snippet-note-award-emoji-list
  - command line option, [776](#)
- gitlab-project-snippet-note-create
  - command line option, [772](#)
- gitlab-project-snippet-note-delete
  - command line option, [773](#)
- gitlab-project-snippet-note-get command line option, [773](#)
- gitlab-project-snippet-note-list
  - command line option, [773](#)
- gitlab-project-snippet-note-update
  - command line option, [774](#)
- gitlab-project-snippet-update command line option, [767](#)
- gitlab-project-snippet-user-agent-detail
  - command line option, [767](#)
- gitlab-project-star command line option, [596](#)
- gitlab-project-storage command line option, [776](#)
- gitlab-project-storage-get command line option, [776](#)
- gitlab-project-tag command line option, [777](#)
- gitlab-project-tag-create command line option, [777](#)
- gitlab-project-tag-delete command line option, [777](#)
- gitlab-project-tag-get command line option, [777](#)
- gitlab-project-tag-list command line option, [778](#)
- gitlab-project-transfer command line option, [596](#)
- gitlab-project-trigger command line option, [778](#)
- gitlab-project-trigger-create command line option, [778](#)
- gitlab-project-trigger-delete command line option, [778](#)
- gitlab-project-trigger-get command line option, [779](#)
- gitlab-project-trigger-list command line option, [779](#)
- gitlab-project-trigger-pipeline command line option, [596](#)
- gitlab-project-trigger-update command line option, [779](#)
- gitlab-project-unarchive command line option, [596](#)
- gitlab-project-unshare command line option, [597](#)
- gitlab-project-unstar command line option, [597](#)
- gitlab-project-update command line option, [599](#)
- gitlab-project-update-submodule command line option, [602](#)
- gitlab-project-upload command line option, [602](#)
- gitlab-project-user command line option, [780](#)
- gitlab-project-user-list command line option, [780](#)
- gitlab-project-variable command line

- option, [780](#)
- gitlab-project-variable-create command line option, [780](#)
- gitlab-project-variable-delete command line option, [781](#)
- gitlab-project-variable-get command line option, [781](#)
- gitlab-project-variable-list command line option, [781](#)
- gitlab-project-variable-update command line option, [782](#)
- gitlab-project-wiki command line option, [782](#)
- gitlab-project-wiki-create command line option, [783](#)
- gitlab-project-wiki-delete command line option, [783](#)
- gitlab-project-wiki-get command line option, [783](#)
- gitlab-project-wiki-list command line option, [784](#)
- gitlab-project-wiki-update command line option, [784](#)
- gitlab-project-wiki-upload command line option, [784](#)
- gitlab-registry-repository command line option, [785](#)
- gitlab-registry-repository-get command line option, [785](#)
- gitlab-runner command line option, [785](#)
- gitlab-runner-all command line option, [785](#), [788](#)
- gitlab-runner-all-list command line option, [788](#)
- gitlab-runner-create command line option, [786](#)
- gitlab-runner-delete command line option, [786](#)
- gitlab-runner-get command line option, [786](#)
- gitlab-runner-job command line option, [789](#)
- gitlab-runner-job-list command line option, [789](#)
- gitlab-runner-list command line option, [787](#)
- gitlab-runner-update command line option, [787](#)
- gitlab-runner-verify command line option, [788](#)
- gitlab-shared-project command line option, [789](#)
- gitlab-shared-project-list command line option, [789](#)
- gitlab-snippet command line option, [790](#)
- gitlab-snippet-content command line option, [791](#)
- gitlab-snippet-create command line option, [791](#)
- gitlab-snippet-delete command line option, [791](#)
- gitlab-snippet-get command line option, [792](#)
- gitlab-snippet-list command line option, [792](#)
- gitlab-snippet-public command line option, [792](#)
- gitlab-snippet-update command line option, [792](#)
- gitlab-snippet-user-agent-detail command line option, [793](#)
- gitlab-starred-project command line option, [793](#)
- gitlab-starred-project-list command line option, [793](#)
- gitlab-todo command line option, [794](#)
- gitlab-todo-delete command line option, [794](#)
- gitlab-todo-list command line option, [795](#)
- gitlab-todo-mark-all-as-done command line option, [795](#)
- gitlab-todo-mark-as-done command line option, [795](#)
- gitlab-topic command line option, [796](#)
- gitlab-topic-create command line option, [796](#)
- gitlab-topic-delete command line option, [796](#)
- gitlab-topic-get command line option, [796](#)
- gitlab-topic-list command line option, [797](#)
- gitlab-topic-merge command line option, [797](#)
- gitlab-topic-update command line option, [797](#)
- gitlab-user command line option, [798](#)
- gitlab-user-activate command line option, [798](#)
- gitlab-user-activities command line option, [804](#)
- gitlab-user-activities-list command line option, [804](#)
- gitlab-user-approve command line option, [798](#)
- gitlab-user-ban command line option, [798](#)
- gitlab-user-block command line option, [799](#)
- gitlab-user-create command line option,

- 799
- gitlab-user-custom-attribute command line option, 804
  - gitlab-user-custom-attribute-delete command line option, 805
  - gitlab-user-custom-attribute-get command line option, 805
  - gitlab-user-custom-attribute-list command line option, 805
  - gitlab-user-deactivate command line option, 800
  - gitlab-user-delete command line option, 800
  - gitlab-user-email command line option, 806
  - gitlab-user-email-create command line option, 806
  - gitlab-user-email-delete command line option, 806
  - gitlab-user-email-get command line option, 806
  - gitlab-user-email-list command line option, 807
  - gitlab-user-event command line option, 807
  - gitlab-user-event-list command line option, 807
  - gitlab-user-follow command line option, 801
  - gitlab-user-get command line option, 801
  - gitlab-user-gpg-key command line option, 808
  - gitlab-user-gpg-key-create command line option, 808
  - gitlab-user-gpg-key-delete command line option, 808
  - gitlab-user-gpg-key-get command line option, 808
  - gitlab-user-gpg-key-list command line option, 809
  - gitlab-user-impersonation-token command line option, 809
  - gitlab-user-impersonation-token-create command line option, 809
  - gitlab-user-impersonation-token-delete command line option, 810
  - gitlab-user-impersonation-token-get command line option, 810
  - gitlab-user-impersonation-token-list command line option, 810
  - gitlab-user-key command line option, 811
  - gitlab-user-key-create command line option, 811
  - gitlab-user-key-delete command line option, 811
  - gitlab-user-key-get command line option, 811
  - gitlab-user-key-list command line option, 812
  - gitlab-user-list command line option, 801
  - gitlab-user-membership command line option, 812
  - gitlab-user-membership-get command line option, 812
  - gitlab-user-membership-list command line option, 812
  - gitlab-user-personal-access-token command line option, 813
  - gitlab-user-personal-access-token-create command line option, 813
  - gitlab-user-project command line option, 813
  - gitlab-user-project-create command line option, 814
  - gitlab-user-project-list command line option, 815
  - gitlab-user-reject command line option, 802
  - gitlab-user-status command line option, 816
  - gitlab-user-status-get command line option, 816
  - gitlab-user-unban command line option, 802
  - gitlab-user-unblock command line option, 802
  - gitlab-user-unfollow command line option, 802
  - gitlab-user-update command line option, 803
  - gitlab-variable command line option, 816
  - gitlab-variable-create command line option, 816
  - gitlab-variable-delete command line option, 817
  - gitlab-variable-get command line option, 817
  - gitlab-variable-list command line option, 817
  - gitlab-variable-update command line option, 818
  - o gitlab command line option, 480
  - v gitlab command line option, 480
- ## A
- access\_tokens (*gitlab.v4.objects.Group* attribute), 222

- `access_tokens` (*gitlab.v4.objects.Project* attribute), 277
  - `AccessLevel` (class in *gitlab.const*), 447
  - `AccessRequestMixin` (class in *gitlab.mixins*), 465
  - `accessrequests` (*gitlab.v4.objects.Group* attribute), 222
  - `accessrequests` (*gitlab.v4.objects.Project* attribute), 277
  - `activate()` (*gitlab.v4.objects.User* method), 427
  - `add_spent_time()` (*gitlab.mixins.TimeTrackingMixin* method), 471
  - `additionalstatistics` (*gitlab.v4.objects.Project* attribute), 277
  - `ADMIN` (*gitlab.const.AccessLevel* attribute), 447
  - `all()` (*gitlab.v4.objects.RunnerManager* method), 418
  - `Any` (class in *gitlab.v4.objects*), 198
  - `api_url` (*gitlab.Gitlab* property), 188
  - `api_version` (*gitlab.Gitlab* property), 188
  - `appearance` (*gitlab.Gitlab* attribute), 188
  - `Application` (class in *gitlab.v4.objects*), 198
  - `ApplicationAppearance` (class in *gitlab.v4.objects*), 198
  - `ApplicationAppearanceManager` (class in *gitlab.v4.objects*), 198
  - `ApplicationManager` (class in *gitlab.v4.objects*), 200
  - `applications` (*gitlab.Gitlab* attribute), 188
  - `ApplicationSettings` (class in *gitlab.v4.objects*), 200
  - `ApplicationSettingsManager` (class in *gitlab.v4.objects*), 200
  - `ApplicationStatistics` (class in *gitlab.v4.objects*), 203
  - `ApplicationStatisticsManager` (class in *gitlab.v4.objects*), 203
  - `approval()` (*gitlab.v4.objects.ProjectDeployment* method), 305
  - `approval_rule_id` (*gitlab.v4.objects.ProjectMergeRequestApprovalRule* attribute), 357
  - `approval_rules` (*gitlab.v4.objects.ProjectMergeRequest* attribute), 352
  - `approval_state` (*gitlab.v4.objects.ProjectMergeRequest* attribute), 352
  - `approvalrules` (*gitlab.v4.objects.Project* attribute), 277
  - `approvals` (*gitlab.v4.objects.Project* attribute), 277
  - `approvals` (*gitlab.v4.objects.ProjectMergeRequest* attribute), 352
  - `approve()` (*gitlab.mixins.AccessRequestMixin* method), 465
  - `approve()` (*gitlab.v4.objects.ProjectMergeRequest* method), 352
  - `approve()` (*gitlab.v4.objects.User* method), 427
  - `archive()` (*gitlab.v4.objects.Project* method), 277
  - `artifact()` (*gitlab.v4.objects.ProjectJob* method), 334
  - `artifacts` (*gitlab.v4.objects.Project* attribute), 277
  - `artifacts()` (*gitlab.v4.objects.ProjectJob* method), 334
  - `asdict()` (*gitlab.base.RESTObject* method), 444
  - `attributes` (*gitlab.base.RESTObject* property), 444
  - `audit_events` (*gitlab.Gitlab* attribute), 188
  - `audit_events` (*gitlab.v4.objects.Group* attribute), 222
  - `audit_events` (*gitlab.v4.objects.Project* attribute), 277
  - `AuditEvent` (class in *gitlab.v4.objects*), 203
  - `AuditEventManager` (class in *gitlab.v4.objects*), 204
  - `auth()` (*gitlab.Gitlab* method), 188
  - `available()` (*gitlab.v4.objects.ProjectIntegrationManager* method), 318
  - `awardemojis` (*gitlab.v4.objects.GroupEpicNote* attribute), 240
  - `awardemojis` (*gitlab.v4.objects.ProjectIssue* attribute), 320
  - `awardemojis` (*gitlab.v4.objects.ProjectIssueNote* attribute), 327
  - `awardemojis` (*gitlab.v4.objects.ProjectMergeRequest* attribute), 353
  - `awardemojis` (*gitlab.v4.objects.ProjectMergeRequestNote* attribute), 367
  - `awardemojis` (*gitlab.v4.objects.ProjectSnippet* attribute), 400
  - `awardemojis` (*gitlab.v4.objects.ProjectSnippetNote* attribute), 405
- ## B
- `BadgeRenderMixin` (class in *gitlab.mixins*), 465
  - `badges` (*gitlab.v4.objects.Group* attribute), 222
  - `badges` (*gitlab.v4.objects.Project* attribute), 277
  - `ban()` (*gitlab.v4.objects.User* method), 427
  - `billable_members` (*gitlab.v4.objects.Group* attribute), 222
  - `blame()` (*gitlab.v4.objects.ProjectFileManager* method), 312
  - `BLOBS` (*gitlab.const.SearchScope* attribute), 447
  - `block()` (*gitlab.v4.objects.User* method), 428
  - `boards` (*gitlab.v4.objects.Group* attribute), 222
  - `boards` (*gitlab.v4.objects.Project* attribute), 277
  - `branch` (*gitlab.v4.objects.ProjectFile* attribute), 310
  - `branches` (*gitlab.v4.objects.Project* attribute), 277
  - `bridges` (*gitlab.v4.objects.ProjectPipeline* attribute), 379
  - `BroadcastMessage` (class in *gitlab.v4.objects*), 204
  - `BroadcastMessageManager` (class in *gitlab.v4.objects*), 204
  - `broadcastmessages` (*gitlab.Gitlab* attribute), 188
  - `bulk_import_entities` (*gitlab.Gitlab* attribute), 188
  - `bulk_imports` (*gitlab.Gitlab* attribute), 188
  - `bulk_publish()` (*gitlab.v4.objects.ProjectMergeRequestDraftNoteManager* method), 364
  - `BulkImport` (class in *gitlab.v4.objects*), 205
  - `BulkImportAllEntity` (class in *gitlab.v4.objects*), 206
  - `BulkImportAllEntityManager` (class in *gitlab.v4.objects*), 206



BulkImportEntity (class in *gitlab.v4.objects*), 206  
 BulkImportEntityManager (class in *gitlab.v4.objects*), 206  
 BulkImportManager (class in *gitlab.v4.objects*), 207

## C

cancel() (*gitlab.v4.objects.ProjectJob* method), 335  
 cancel() (*gitlab.v4.objects.ProjectPipeline* method), 380  
 cancel\_merge\_when\_pipeline\_succeeds() (*gitlab.v4.objects.ProjectMergeRequest* method), 353  
 changes() (*gitlab.v4.objects.ProjectMergeRequest* method), 353  
 cherry\_pick() (*gitlab.v4.objects.ProjectCommit* method), 297  
 ci\_lint (*gitlab.Gitlab* attribute), 188  
 ci\_lint (*gitlab.v4.objects.Project* attribute), 277  
 CiLint (class in *gitlab.v4.objects*), 208  
 CiLintManager (class in *gitlab.v4.objects*), 208  
 closed\_by() (*gitlab.v4.objects.ProjectIssue* method), 320  
 closes\_issues() (*gitlab.v4.objects.ProjectMergeRequest* method), 353  
 cls\_to\_gitlab\_resource() (in module *gitlab.cli*), 445  
 clusters (*gitlab.v4.objects.Group* attribute), 222  
 clusters (*gitlab.v4.objects.Project* attribute), 277  
 comments (*gitlab.v4.objects.ProjectCommit* attribute), 297  
 commit\_message (*gitlab.v4.objects.ProjectFile* attribute), 310  
 COMMITS (*gitlab.const.SearchScope* attribute), 447  
 commits (*gitlab.v4.objects.Project* attribute), 277  
 commits() (*gitlab.v4.objects.ProjectMergeRequest* method), 354  
 compound\_metrics() (*gitlab.v4.objects.SidekiqManager* method), 420  
 ConfigError, 446  
 content() (*gitlab.v4.objects.ProjectSnippet* method), 400  
 content() (*gitlab.v4.objects.Snippet* method), 421  
 copy\_dict() (in module *gitlab.utils*), 475  
 create() (*gitlab.mixins.CreateMixin* method), 466  
 create() (*gitlab.v4.objects.GroupClusterManager* method), 232  
 create() (*gitlab.v4.objects.GroupEpicIssueManager* method), 238  
 create() (*gitlab.v4.objects.ProjectClusterManager* method), 296  
 create() (*gitlab.v4.objects.ProjectCommitStatusManager* method), 303  
 create() (*gitlab.v4.objects.ProjectFileManager* method), 312  
 create() (*gitlab.v4.objects.ProjectForkManager* method), 315  
 create() (*gitlab.v4.objects.ProjectIssueLinkManager* method), 325  
 create() (*gitlab.v4.objects.ProjectMergeRequestApprovalRuleManager* method), 358  
 create() (*gitlab.v4.objects.ProjectPipelineManager* method), 382  
 create\_fork\_relation() (*gitlab.v4.objects.Project* method), 277  
 CreateMixin (class in *gitlab.mixins*), 466  
 CRUdMixin (class in *gitlab.mixins*), 466  
 current\_failures() (*gitlab.v4.objects.GeoNodeManager* method), 219  
 current\_page (*gitlab.base.RESTObjectList* property), 444  
 current\_page (*gitlab.GitlabList* property), 197  
 CurrentUser (class in *gitlab.v4.objects*), 209  
 CurrentUserEmail (class in *gitlab.v4.objects*), 209  
 CurrentUserEmailManager (class in *gitlab.v4.objects*), 209  
 CurrentUserGPGKey (class in *gitlab.v4.objects*), 210  
 CurrentUserGPGKeyManager (class in *gitlab.v4.objects*), 210  
 CurrentUserKey (class in *gitlab.v4.objects*), 211  
 CurrentUserKeyManager (class in *gitlab.v4.objects*), 211  
 CurrentUserManager (class in *gitlab.v4.objects*), 212  
 CurrentUserRunner (class in *gitlab.v4.objects*), 212  
 CurrentUserRunnerManager (class in *gitlab.v4.objects*), 212  
 CurrentUserStatus (class in *gitlab.v4.objects*), 213  
 CurrentUserStatusManager (class in *gitlab.v4.objects*), 213  
 CUSTOM (*gitlab.const.NotificationLevel* attribute), 447  
 customattributes (*gitlab.v4.objects.Group* attribute), 222  
 customattributes (*gitlab.v4.objects.Project* attribute), 278  
 customattributes (*gitlab.v4.objects.User* attribute), 428

## D

deactivate() (*gitlab.v4.objects.User* method), 428  
 decode() (*gitlab.v4.objects.ProjectFile* method), 310  
 delete() (*gitlab.mixins.DeleteMixin* method), 466  
 delete() (*gitlab.mixins.ObjectDeleteMixin* method), 469  
 delete() (*gitlab.v4.objects.GroupLDAPGroupLink* method), 249

- `delete()` (*gitlab.v4.objects.ProjectArtifactManager method*), 288
  - `delete()` (*gitlab.v4.objects.ProjectFile method*), 311
  - `delete()` (*gitlab.v4.objects.ProjectFileManager method*), 313
  - `delete_artifacts()` (*gitlab.v4.objects.ProjectJob method*), 335
  - `delete_fork_relation()` (*gitlab.v4.objects.Project method*), 278
  - `delete_in_bulk()` (*gitlab.v4.objects.ProjectRegistryTagManager method*), 392
  - `delete_merged_branches()` (*gitlab.v4.objects.RepositoryMixin method*), 412
  - `DeleteMixin` (*class in gitlab.mixins*), 466
  - `DeployKey` (*class in gitlab.v4.objects*), 213
  - `DeployKeyManager` (*class in gitlab.v4.objects*), 214
  - `deploykeys` (*gitlab.Gitlab attribute*), 189
  - `deployments` (*gitlab.v4.objects.Project attribute*), 278
  - `DeployToken` (*class in gitlab.v4.objects*), 214
  - `DeployTokenManager` (*class in gitlab.v4.objects*), 214
  - `deploytokens` (*gitlab.Gitlab attribute*), 189
  - `deploytokens` (*gitlab.v4.objects.Group attribute*), 222
  - `deploytokens` (*gitlab.v4.objects.Project attribute*), 278
  - `descendant_groups` (*gitlab.v4.objects.Group attribute*), 222
  - `DEVELOPER` (*gitlab.const.AccessLevel attribute*), 447
  - `die()` (*in module gitlab.cli*), 445
  - `diff()` (*gitlab.v4.objects.ProjectCommit method*), 297
  - `diffs` (*gitlab.v4.objects.ProjectMergeRequest attribute*), 354
  - `DISABLED` (*gitlab.const.NotificationLevel attribute*), 447
  - `discussions` (*gitlab.v4.objects.ProjectCommit attribute*), 297
  - `discussions` (*gitlab.v4.objects.ProjectIssue attribute*), 320
  - `discussions` (*gitlab.v4.objects.ProjectMergeRequest attribute*), 354
  - `discussions` (*gitlab.v4.objects.ProjectSnippet attribute*), 401
  - `Dockerfile` (*class in gitlab.v4.objects*), 214
  - `DockerfileManager` (*class in gitlab.v4.objects*), 214
  - `dockerfiles` (*gitlab.Gitlab attribute*), 189
  - `docs()` (*in module gitlab.cli*), 445
  - `download()` (*gitlab.mixins.DownloadMixin method*), 467
  - `download()` (*gitlab.v4.objects.GenericPackageManager method*), 217
  - `download()` (*gitlab.v4.objects.ProjectArtifactManager method*), 288
  - `download()` (*gitlab.v4.objects.ProjectSecureFile method*), 398
  - `DownloadMixin` (*class in gitlab.mixins*), 467
  - `draft_notes` (*gitlab.v4.objects.ProjectMergeRequest attribute*), 354
- ## E
- `emails` (*gitlab.v4.objects.CurrentUser attribute*), 209
  - `emails` (*gitlab.v4.objects.User attribute*), 428
  - `enable()` (*gitlab.v4.objects.ProjectKeyManager method*), 338
  - `enable_debug()` (*gitlab.Gitlab method*), 189
  - `encoded_id` (*gitlab.base.RESTObject property*), 444
  - `EncodedId` (*class in gitlab.utils*), 474
  - `entities` (*gitlab.v4.objects.BulkImport attribute*), 206
  - `environments` (*gitlab.v4.objects.Project attribute*), 278
  - `epics` (*gitlab.v4.objects.Group attribute*), 222
  - `erase()` (*gitlab.v4.objects.ProjectJob method*), 335
  - `Event` (*class in gitlab.v4.objects*), 215
  - `EventManager` (*class in gitlab.v4.objects*), 215
  - `events` (*gitlab.Gitlab attribute*), 189
  - `events` (*gitlab.v4.objects.Project attribute*), 278
  - `events` (*gitlab.v4.objects.User attribute*), 428
  - `exists()` (*gitlab.v4.objects.NamespaceManager method*), 273
  - `exports` (*gitlab.v4.objects.Group attribute*), 222
  - `exports` (*gitlab.v4.objects.Project attribute*), 278
- ## F
- `Feature` (*class in gitlab.v4.objects*), 216
  - `FeatureManager` (*class in gitlab.v4.objects*), 216
  - `features` (*gitlab.Gitlab attribute*), 189
  - `file_path` (*gitlab.v4.objects.ProjectFile attribute*), 311
  - `files` (*gitlab.v4.objects.Project attribute*), 278
  - `follow()` (*gitlab.v4.objects.User method*), 428
  - `followers_users` (*gitlab.v4.objects.User attribute*), 428
  - `following_users` (*gitlab.v4.objects.User attribute*), 428
  - `forks` (*gitlab.v4.objects.Project attribute*), 278
  - `format()` (*gitlab.utils.MaskingFormatter method*), 474
  - `format_help()` (*gitlab.cli.VerticalHelpFormatter method*), 445
  - `from_config()` (*gitlab.Gitlab class method*), 189
- ## G
- `generic_packages` (*gitlab.v4.objects.Project attribute*), 278
  - `GenericPackage` (*class in gitlab.v4.objects*), 216
  - `GenericPackageManager` (*class in gitlab.v4.objects*), 217
  - `GeoNode` (*class in gitlab.v4.objects*), 218
  - `GeoNodeManager` (*class in gitlab.v4.objects*), 219
  - `geonodes` (*gitlab.Gitlab attribute*), 189
  - `get()` (*gitlab.mixins.GetMixin method*), 467
  - `get()` (*gitlab.mixins.GetWithoutIdMixin method*), 468



- get() (*gitlab.v4.objects.ApplicationAppearanceManager method*), 199
- get() (*gitlab.v4.objects.ApplicationSettingsManager method*), 202
- get() (*gitlab.v4.objects.ApplicationStatisticsManager method*), 203
- get() (*gitlab.v4.objects.AuditEventManager method*), 204
- get() (*gitlab.v4.objects.BroadcastMessageManager method*), 205
- get() (*gitlab.v4.objects.BulkImportEntityManager method*), 207
- get() (*gitlab.v4.objects.BulkImportManager method*), 207
- get() (*gitlab.v4.objects.CurrentUserEmailManager method*), 209
- get() (*gitlab.v4.objects.CurrentUserGPGKeyManager method*), 210
- get() (*gitlab.v4.objects.CurrentUserKeyManager method*), 211
- get() (*gitlab.v4.objects.CurrentUserManager method*), 212
- get() (*gitlab.v4.objects.CurrentUserStatusManager method*), 213
- get() (*gitlab.v4.objects.DockerfileManager method*), 215
- get() (*gitlab.v4.objects.GeoNodeManager method*), 219
- get() (*gitlab.v4.objects.GitignoreManager method*), 221
- get() (*gitlab.v4.objects.GitlabciyamlManager method*), 221
- get() (*gitlab.v4.objects.GroupAccessTokenManager method*), 226
- get() (*gitlab.v4.objects.GroupAuditEventManager method*), 227
- get() (*gitlab.v4.objects.GroupBadgeManager method*), 228
- get() (*gitlab.v4.objects.GroupBoardListManager method*), 230
- get() (*gitlab.v4.objects.GroupBoardManager method*), 231
- get() (*gitlab.v4.objects.GroupClusterManager method*), 232
- get() (*gitlab.v4.objects.GroupCustomAttributeManager method*), 233
- get() (*gitlab.v4.objects.GroupDeployTokenManager method*), 234
- get() (*gitlab.v4.objects.GroupEpicAwardEmojiManager method*), 236
- get() (*gitlab.v4.objects.GroupEpicDiscussionNoteManager method*), 237
- get() (*gitlab.v4.objects.GroupEpicManager method*), 239
- get() (*gitlab.v4.objects.GroupEpicNoteAwardEmojiManager method*), 240
- get() (*gitlab.v4.objects.GroupEpicNoteManager method*), 241
- get() (*gitlab.v4.objects.GroupEpicResourceLabelEventManager method*), 242
- get() (*gitlab.v4.objects.GroupExportManager method*), 242
- get() (*gitlab.v4.objects.GroupHookManager method*), 244
- get() (*gitlab.v4.objects.GroupImportManager method*), 245
- get() (*gitlab.v4.objects.GroupInvitationManager method*), 246
- get() (*gitlab.v4.objects.GroupIssuesStatisticsManager method*), 248
- get() (*gitlab.v4.objects.GroupLabelManager method*), 250
- get() (*gitlab.v4.objects.GroupManager method*), 253
- get() (*gitlab.v4.objects.GroupMemberAllManager method*), 254
- get() (*gitlab.v4.objects.GroupMemberManager method*), 255
- get() (*gitlab.v4.objects.GroupMilestoneManager method*), 258
- get() (*gitlab.v4.objects.GroupNotificationSettingsManager method*), 259
- get() (*gitlab.v4.objects.GroupPushRulesManager method*), 262
- get() (*gitlab.v4.objects.GroupSAMLGroupLinkManager method*), 263
- get() (*gitlab.v4.objects.GroupVariableManager method*), 265
- get() (*gitlab.v4.objects.GroupWikiManager method*), 266
- get() (*gitlab.v4.objects.HookManager method*), 267
- get() (*gitlab.v4.objects.IssueManager method*), 268
- get() (*gitlab.v4.objects.IssuesStatisticsManager method*), 269
- get() (*gitlab.v4.objects.KeyManager method*), 270
- get() (*gitlab.v4.objects.LicenseManager method*), 271
- get() (*gitlab.v4.objects.NamespaceManager method*), 274
- get() (*gitlab.v4.objects.NotificationSettingsManager method*), 275
- get() (*gitlab.v4.objects.PersonalAccessTokenManager method*), 276
- get() (*gitlab.v4.objects.ProjectAccessTokenManager method*), 284
- get() (*gitlab.v4.objects.ProjectAdditionalStatisticsManager method*), 285
- get() (*gitlab.v4.objects.ProjectApprovalManager method*), 286
- get() (*gitlab.v4.objects.ProjectAuditEventManager method*), 290
- get() (*gitlab.v4.objects.ProjectBadgeManager method*),

291  
get() (gitlab.v4.objects.ProjectBoardListManager method), 292  
get() (gitlab.v4.objects.ProjectBoardManager method), 293  
get() (gitlab.v4.objects.ProjectBranchManager method), 293  
get() (gitlab.v4.objects.ProjectCiLintManager method), 294  
get() (gitlab.v4.objects.ProjectClusterManager method), 296  
get() (gitlab.v4.objects.ProjectCommitDiscussionManager method), 299  
get() (gitlab.v4.objects.ProjectCommitDiscussionNoteManager method), 300  
get() (gitlab.v4.objects.ProjectCommitManager method), 302  
get() (gitlab.v4.objects.ProjectCustomAttributeManager method), 303  
get() (gitlab.v4.objects.ProjectDeploymentManager method), 306  
get() (gitlab.v4.objects.ProjectDeployTokenManager method), 304  
get() (gitlab.v4.objects.ProjectEnvironmentManager method), 309  
get() (gitlab.v4.objects.ProjectExportManager method), 310  
get() (gitlab.v4.objects.ProjectFileManager method), 313  
get() (gitlab.v4.objects.ProjectHookManager method), 317  
get() (gitlab.v4.objects.ProjectImportManager method), 317  
get() (gitlab.v4.objects.ProjectIntegrationManager method), 318  
get() (gitlab.v4.objects.ProjectInvitationManager method), 319  
get() (gitlab.v4.objects.ProjectIssueAwardEmojiManager method), 322  
get() (gitlab.v4.objects.ProjectIssueDiscussionManager method), 323  
get() (gitlab.v4.objects.ProjectIssueDiscussionNoteManager method), 324  
get() (gitlab.v4.objects.ProjectIssueManager method), 326  
get() (gitlab.v4.objects.ProjectIssueNoteAwardEmojiManager method), 327  
get() (gitlab.v4.objects.ProjectIssueNoteManager method), 328  
get() (gitlab.v4.objects.ProjectIssueResourceIterationEventManager method), 329  
get() (gitlab.v4.objects.ProjectIssueResourceLabelEventManager method), 330  
get() (gitlab.v4.objects.ProjectIssueResourceMilestoneEventManager method), 331  
get() (gitlab.v4.objects.ProjectIssueResourceStateEventManager method), 331  
get() (gitlab.v4.objects.ProjectIssueResourceWeightEventManager method), 332  
get() (gitlab.v4.objects.ProjectIssuesStatisticsManager method), 333  
get() (gitlab.v4.objects.ProjectJobManager method), 337  
get() (gitlab.v4.objects.ProjectJobTokenScopeManager method), 338  
get() (gitlab.v4.objects.ProjectKeyManager method), 339  
get() (gitlab.v4.objects.ProjectLabelManager method), 340  
get() (gitlab.v4.objects.ProjectManager method), 346  
get() (gitlab.v4.objects.ProjectMemberAllManager method), 351  
get() (gitlab.v4.objects.ProjectMemberManager method), 352  
get() (gitlab.v4.objects.ProjectMergeRequestApprovalManager method), 356  
get() (gitlab.v4.objects.ProjectMergeRequestApprovalRuleManager method), 359  
get() (gitlab.v4.objects.ProjectMergeRequestApprovalStateManager method), 359  
get() (gitlab.v4.objects.ProjectMergeRequestAwardEmojiManager method), 360  
get() (gitlab.v4.objects.ProjectMergeRequestDiffManager method), 361  
get() (gitlab.v4.objects.ProjectMergeRequestDiscussionManager method), 362  
get() (gitlab.v4.objects.ProjectMergeRequestDiscussionNoteManager method), 363  
get() (gitlab.v4.objects.ProjectMergeRequestDraftNoteManager method), 364  
get() (gitlab.v4.objects.ProjectMergeRequestManager method), 367  
get() (gitlab.v4.objects.ProjectMergeRequestNoteAwardEmojiManager method), 368  
get() (gitlab.v4.objects.ProjectMergeRequestNoteManager method), 368  
get() (gitlab.v4.objects.ProjectMergeRequestResourceLabelEventManager method), 370  
get() (gitlab.v4.objects.ProjectMergeRequestResourceMilestoneEventManager method), 370  
get() (gitlab.v4.objects.ProjectMergeRequestResourceStateEventManager method), 371  
get() (gitlab.v4.objects.ProjectMilestoneManager method), 374  
get() (gitlab.v4.objects.ProjectNoteManager method), 375  
get() (gitlab.v4.objects.ProjectNotificationSettingsManager method), 376

- `get()` (*gitlab.v4.objects.ProjectPackageManager method*), 377
- `get()` (*gitlab.v4.objects.ProjectPagesDomainManager method*), 379
- `get()` (*gitlab.v4.objects.ProjectPipelineManager method*), 382
- `get()` (*gitlab.v4.objects.ProjectPipelineScheduleManager method*), 384
- `get()` (*gitlab.v4.objects.ProjectPipelineTestReportManager method*), 386
- `get()` (*gitlab.v4.objects.ProjectPipelineTestReportSummaryManager method*), 386
- `get()` (*gitlab.v4.objects.ProjectProtectedBranchManager method*), 388
- `get()` (*gitlab.v4.objects.ProjectProtectedEnvironmentManager method*), 389
- `get()` (*gitlab.v4.objects.ProjectProtectedTagManager method*), 389
- `get()` (*gitlab.v4.objects.ProjectPushRulesManager method*), 391
- `get()` (*gitlab.v4.objects.ProjectRegistryTagManager method*), 392
- `get()` (*gitlab.v4.objects.ProjectReleaseLinkManager method*), 394
- `get()` (*gitlab.v4.objects.ProjectReleaseManager method*), 395
- `get()` (*gitlab.v4.objects.ProjectResourceGroupManager method*), 397
- `get()` (*gitlab.v4.objects.ProjectSecureFileManager method*), 399
- `get()` (*gitlab.v4.objects.ProjectServiceManager method*), 400
- `get()` (*gitlab.v4.objects.ProjectSnippetAwardEmojiManager method*), 401
- `get()` (*gitlab.v4.objects.ProjectSnippetDiscussionManager method*), 402
- `get()` (*gitlab.v4.objects.ProjectSnippetDiscussionNoteManager method*), 403
- `get()` (*gitlab.v4.objects.ProjectSnippetManager method*), 404
- `get()` (*gitlab.v4.objects.ProjectSnippetNoteAwardEmojiManager method*), 405
- `get()` (*gitlab.v4.objects.ProjectSnippetNoteManager method*), 406
- `get()` (*gitlab.v4.objects.ProjectStorageManager method*), 407
- `get()` (*gitlab.v4.objects.ProjectTagManager method*), 407
- `get()` (*gitlab.v4.objects.ProjectTriggerManager method*), 408
- `get()` (*gitlab.v4.objects.ProjectVariableManager method*), 410
- `get()` (*gitlab.v4.objects.ProjectWikiManager method*), 411
- `get()` (*gitlab.v4.objects.RegistryRepositoryManager method*), 412
- `get()` (*gitlab.v4.objects.RunnerManager method*), 418
- `get()` (*gitlab.v4.objects.SnippetManager method*), 422
- `get()` (*gitlab.v4.objects.TopicManager method*), 426
- `get()` (*gitlab.v4.objects.UserCustomAttributesManager method*), 431
- `get()` (*gitlab.v4.objects.UserEmailManager method*), 432
- `get()` (*gitlab.v4.objects.UserGPGKeyManager method*), 433
- `get()` (*gitlab.v4.objects.UserImpersonationTokenManager method*), 434
- `get()` (*gitlab.v4.objects.UserKeyManager method*), 435
- `get()` (*gitlab.v4.objects.UserManager method*), 437
- `get()` (*gitlab.v4.objects.UserMembershipManager method*), 438
- `get()` (*gitlab.v4.objects.UserStatusManager method*), 441
- `get()` (*gitlab.v4.objects.VariableManager method*), 442
- `get_content_type()` (in module *gitlab.utils*), 475
- `get_id()` (*gitlab.base.RESTObject method*), 444
- `get_license()` (*gitlab.Gitlab method*), 190
- `GetMixin` (class in *gitlab.mixins*), 467
- `GetWithoutIdMixin` (class in *gitlab.mixins*), 468
- `Gitignore` (class in *gitlab.v4.objects*), 220
- `GitignoreManager` (class in *gitlab.v4.objects*), 220
- `gitignores` (*gitlab.Gitlab attribute*), 190
- `Gitlab` (class in *gitlab*), 187
- `gitlab` (*gitlab.base.RESTManager attribute*), 443
- `gitlab` (*gitlab.mixins.CreateMixin attribute*), 466
- `gitlab` (*gitlab.mixins.CRUDMixin attribute*), 466
- `gitlab` (*gitlab.mixins.DeleteMixin attribute*), 467
- `gitlab` (*gitlab.mixins.GetMixin attribute*), 468
- `gitlab` (*gitlab.mixins.GetWithoutIdMixin attribute*), 468
- `gitlab` (*gitlab.mixins.ListMixin attribute*), 468
- `gitlab` (*gitlab.mixins.NoUpdateMixin attribute*), 468
- `gitlab` (*gitlab.mixins.RetrieveMixin attribute*), 470
- `gitlab` (*gitlab.mixins.SetMixin attribute*), 470
- `gitlab` (*gitlab.mixins.UpdateMixin attribute*), 473
- `gitlab` command line option
  - `--api-version`, 481
  - `--config-file`, 480
  - `--debug`, 480
  - `--fancy`, 480
  - `--fields`, 480
  - `--gitlab`, 480
  - `--help`, 480
  - `--job-token`, 481
  - `--no-ssl-verify`, 480
  - `--oauth-token`, 481
  - `--order-by`, 481
  - `--output`, 480
  - `--pagination`, 481

- `--per-page`, 481
  - `--private-token`, 481
  - `--server-url`, 480
  - `--skip-login`, 481
  - `--ssl-verify`, 480
  - `--timeout`, 481
  - `--user-agent`, 481
  - `--verbose`, 480
  - `--version`, 480
  - `-c`, 480
  - `-d`, 480
  - `-f`, 480
  - `-g`, 480
  - `-h`, 480
  - `-o`, 480
  - `-v`, 480
- `gitlab.base`
  - module, 443
- `gitlab.cli`
  - module, 445
- `gitlab.config`
  - module, 446
- `gitlab.const`
  - module, 447
- `gitlab.exceptions`
  - module, 448
- `gitlab.mixins`
  - module, 465
- `gitlab.utils`
  - module, 474
- `gitlab.v4`
  - module, 443
- `gitlab.v4.objects`
  - module, 198
- `gitlab_resource_to_cls()` (in module `gitlab.cli`), 446
- `gitlab-application` command line option
  - `--help`, 481
  - `-h`, 481
- `gitlab-application-appearance` command line option
  - `--help`, 482
  - `-h`, 482
- `gitlab-application-appearance-get` command line option
  - `--help`, 482
  - `--sudo`, 482
  - `-h`, 482
- `gitlab-application-appearance-update` command line option
  - `--description`, 483
  - `--email-header-and-footer-enabled`, 483
  - `--favicon`, 483
  - `--footer-message`, 483
  - `--header-logo`, 483
  - `--header-message`, 483
  - `--help`, 483
  - `--logo`, 483
  - `--message-background-color`, 483
  - `--message-font-color`, 483
  - `--new-project-guidelines`, 483
  - `--sudo`, 483
  - `--title`, 483
  - `-h`, 483
- `gitlab-application-create` command line option
  - `--confidential`, 482
  - `--help`, 481
  - `--name`, 481
  - `--redirect-uri`, 481
  - `--scopes`, 482
  - `--sudo`, 481
  - `-h`, 481
- `gitlab-application-delete` command line option
  - `--help`, 482
  - `--id`, 482
  - `--sudo`, 482
  - `-h`, 482
- `gitlab-application-list` command line option
  - `--get-all`, 482
  - `--help`, 482
  - `--page`, 482
  - `--per-page`, 482
  - `--sudo`, 482
  - `-h`, 482
- `gitlab-application-settings` command line option
  - `--help`, 483
  - `-h`, 483
- `gitlab-application-settings-get` command line option
  - `--help`, 484
  - `--sudo`, 484
  - `-h`, 484
- `gitlab-application-settings-update` command line option
  - `--after-sign-out-path`, 486
  - `--allow-local-requests-from-hooks-and-services`, 487
  - `--allow-local-requests-from-system-hooks`, 487
  - `--allow-local-requests-from-web-hooks-and-services`, 487
  - `--asset-proxy-allowlist`, 487
  - `--asset-proxy-enabled`, 487
  - `--asset-proxy-url`, 487
  - `--asset-proxy-whitelist`, 487

```

--container-registry-token-expire-delay,
    486
--created-at, 486
--default-branch-protection, 486
--default-group-visibility, 486
--default-project-visibility, 486
--default-projects-limit, 485
--default-snippet-visibility, 486
--disabled-oauth-sign-in-sources, 486
--domain-allowlist, 486
--domain-blacklist, 486
--domain-blacklist-enabled, 486
--domain-denylist, 486
--domain-denylist-enabled, 486
--domain-whitelist, 486
--dsa-key-restriction, 487
--ecdsa-key-restriction, 487
--ed25519-key-restriction, 487
--enforce-terms, 487
--external-authorization-service-default-label,
    486
--external-authorization-service-enabled,
    486
--external-authorization-service-timeout,
    486
--external-authorization-service-url, 486
--file-template-project-id, 487
--first-day-of-week, 487
--geo-node-allowed-ips, 487
--gravatar-enabled, 486
--help, 485
--home-page-url, 486
--id, 485
--import-sources, 486
--instance-statistics-visibility-private,
    487
--local-markdown-version, 487
--max-attachment-size, 486
--outbound-local-requests-whitelist, 486
--password-authentication-enabled-for-web,
    486
--performance-bar-allowed-group-id, 487
--plantuml-enabled, 486
--plantuml-url, 487
--polling-interval-multiplier, 487
--repository-storages, 486
--restricted-visibility-levels, 486
--rsa-key-restriction, 487
--session-expire-delay, 486
--sign-in-text, 486
--signup-enabled, 485
--sudo, 485
--terminal-max-session-time, 487
--terms, 487
--updated-at, 486
--user-oauth-applications, 486
--user-show-add-ssh-key-message, 487
    -h, 485
gitlab-application-statistics command line
    option
    --help, 487
    -h, 487
gitlab-application-statistics-get command
    line option
    --help, 488
    --sudo, 488
    -h, 488
gitlab-audit-event command line option
    --help, 488
    -h, 488
gitlab-audit-event-get command line option
    --help, 488
    --id, 488
    --sudo, 488
    -h, 488
gitlab-audit-event-list command line option
    --created-after, 488
    --created-before, 488
    --entity-id, 488
    --entity-type, 488
    --get-all, 488
    --help, 488
    --page, 488
    --per-page, 488
    --sudo, 488
    -h, 488
gitlab-broadcast-message command line
    option
    --help, 489
    -h, 489
gitlab-broadcast-message-create command
    line option
    --color, 489
    --ends-at, 489
    --font, 489
    --help, 489
    --message, 489
    --starts-at, 489
    --sudo, 489
    --target-access-levels, 489
    -h, 489
gitlab-broadcast-message-delete command
    line option
    --help, 489
    --id, 489
    --sudo, 489
    -h, 489

```



gitlab-broadcast-message-get command line  
option  
--help, 490  
--id, 490  
--sudo, 490  
-h, 490

gitlab-broadcast-message-list command line  
option  
--get-all, 490  
--help, 490  
--page, 490  
--per-page, 490  
--sudo, 490  
-h, 490

gitlab-broadcast-message-update command  
line option  
--color, 490  
--ends-at, 490  
--font, 490  
--help, 490  
--id, 490  
--message, 490  
--starts-at, 490  
--sudo, 490  
--target-access-levels, 490  
-h, 490

gitlab-bulk-import command line option  
--help, 491  
-h, 491

gitlab-bulk-import-all-entity command line  
option  
--help, 492  
-h, 492

gitlab-bulk-import-all-entity-list command  
line option  
--get-all, 492  
--help, 492  
--page, 492  
--per-page, 492  
--sort, 492  
--status, 492  
--sudo, 492  
-h, 492

gitlab-bulk-import-create command line  
option  
--configuration, 491  
--entities, 491  
--help, 491  
--sudo, 491  
-h, 491

gitlab-bulk-import-entity command line  
option  
--help, 492  
-h, 492

gitlab-bulk-import-entity-get command line  
option  
--bulk-import-id, 492  
--help, 492  
--id, 492  
--sudo, 492  
-h, 492

gitlab-bulk-import-entity-list command line  
option  
--bulk-import-id, 493  
--get-all, 493  
--help, 493  
--page, 493  
--per-page, 493  
--sort, 493  
--status, 493  
--sudo, 493  
-h, 493

gitlab-bulk-import-get command line option  
--help, 491  
--id, 491  
--sudo, 491  
-h, 491

gitlab-bulk-import-list command line option  
--get-all, 491  
--help, 491  
--page, 491  
--per-page, 491  
--sort, 491  
--status, 491  
--sudo, 491  
-h, 491

gitlab-ci-lint command line option  
--help, 493  
-h, 493

gitlab-ci-lint-create command line option  
--content, 493  
--help, 493  
--include-jobs, 493  
--include-merged-yaml, 493  
--sudo, 493  
-h, 493

gitlab-ci-lint-validate command line option  
--content, 494  
--help, 494  
--include-jobs, 494  
--include-merged-yaml, 494  
-h, 494

gitlab-current-user command line option  
--help, 494  
-h, 494

gitlab-current-user-email command line  
option  
--help, 494

---

```

    -h, 494
gitlab-current-user-email-create command
    line option
    --email, 494
    --help, 494
    --sudo, 494
    -h, 494
gitlab-current-user-email-delete command
    line option
    --help, 495
    --id, 495
    --sudo, 495
    -h, 495
gitlab-current-user-email-get command line
    option
    --help, 495
    --id, 495
    --sudo, 495
    -h, 495
gitlab-current-user-email-list command line
    option
    --get-all, 495
    --help, 495
    --page, 495
    --per-page, 495
    --sudo, 495
    -h, 495
gitlab-current-user-get command line option
    --help, 494
    --sudo, 494
    -h, 494
gitlab-current-user-gpg-key command line
    option
    --help, 495
    -h, 495
gitlab-current-user-gpg-key-create command
    line option
    --help, 496
    --key, 496
    --sudo, 496
    -h, 496
gitlab-current-user-gpg-key-delete command
    line option
    --help, 496
    --id, 496
    --sudo, 496
    -h, 496
gitlab-current-user-gpg-key-get command
    line option
    --help, 496
    --id, 496
    --sudo, 496
    -h, 496
gitlab-current-user-gpg-key-list command
    line option
    --get-all, 496
    --help, 496
    --page, 496
    --per-page, 496
    --sudo, 496
    -h, 496
gitlab-current-user-key command line option
    --help, 497
    -h, 497
gitlab-current-user-key-create command line
    option
    --help, 497
    --key, 497
    --sudo, 497
    --title, 497
    -h, 497
gitlab-current-user-key-delete command line
    option
    --help, 497
    --id, 497
    --sudo, 497
    -h, 497
gitlab-current-user-key-get command line
    option
    --help, 497
    --id, 497
    --sudo, 497
    -h, 497
gitlab-current-user-key-list command line
    option
    --get-all, 498
    --help, 498
    --page, 498
    --per-page, 498
    --sudo, 498
    -h, 498
gitlab-current-user-runner command line
    option
    --help, 498
    -h, 498
gitlab-current-user-runner-create command
    line option
    --access-level, 499
    --description, 498
    --group-id, 498
    --help, 498
    --locked, 498
    --maintenance-note, 499
    --maximum-timeout, 499
    --paused, 498
    --project-id, 498
    --run-untagged, 498

```

- [--runner-type, 498](#)
  - [--sudo, 498](#)
  - [--tag-list, 499](#)
  - [-h, 498](#)
- gitlab-current-user-status command line option
  - [--help, 499](#)
  - [-h, 499](#)
- gitlab-current-user-status-get command line option
  - [--help, 499](#)
  - [--sudo, 499](#)
  - [-h, 499](#)
- gitlab-current-user-status-update command line option
  - [--emoji, 499](#)
  - [--help, 499](#)
  - [--message, 499](#)
  - [--sudo, 499](#)
  - [-h, 499](#)
- gitlab-deploy-key command line option
  - [--help, 499](#)
  - [-h, 499](#)
- gitlab-deploy-key-list command line option
  - [--get-all, 500](#)
  - [--help, 500](#)
  - [--page, 500](#)
  - [--per-page, 500](#)
  - [--sudo, 500](#)
  - [-h, 500](#)
- gitlab-deploy-token command line option
  - [--help, 500](#)
  - [-h, 500](#)
- gitlab-deploy-token-list command line option
  - [--get-all, 500](#)
  - [--help, 500](#)
  - [--page, 500](#)
  - [--per-page, 500](#)
  - [--sudo, 500](#)
  - [-h, 500](#)
- gitlab-dockerfile command line option
  - [--help, 500](#)
  - [-h, 500](#)
- gitlab-dockerfile-get command line option
  - [--help, 501](#)
  - [--name, 501](#)
  - [--sudo, 501](#)
  - [-h, 501](#)
- gitlab-dockerfile-list command line option
  - [--get-all, 501](#)
  - [--help, 501](#)
  - [--page, 501](#)
  - [--per-page, 501](#)
- [--sudo, 501](#)
  - [-h, 501](#)
- gitlab-event command line option
  - [--help, 501](#)
  - [-h, 501](#)
- gitlab-event-list command line option
  - [--action, 501](#)
  - [--after, 501](#)
  - [--before, 501](#)
  - [--get-all, 502](#)
  - [--help, 501](#)
  - [--page, 502](#)
  - [--per-page, 502](#)
  - [--scope, 502](#)
  - [--sort, 502](#)
  - [--sudo, 501](#)
  - [--target-type, 501](#)
  - [-h, 501](#)
- gitlab-feature command line option
  - [--help, 502](#)
  - [-h, 502](#)
- gitlab-feature-delete command line option
  - [--help, 502](#)
  - [--name, 502](#)
  - [--sudo, 502](#)
  - [-h, 502](#)
- gitlab-feature-list command line option
  - [--get-all, 502](#)
  - [--help, 502](#)
  - [--page, 502](#)
  - [--per-page, 502](#)
  - [--sudo, 502](#)
  - [-h, 502](#)
- gitlab-generic-package command line option
  - [--help, 503](#)
  - [-h, 503](#)
- gitlab-generic-package-download command line option
  - [--file-name, 503](#)
  - [--help, 503](#)
  - [--package-name, 503](#)
  - [--package-version, 503](#)
  - [--project-id, 503](#)
  - [--sudo, 503](#)
  - [-h, 503](#)
- gitlab-generic-package-upload command line option
  - [--file-name, 503](#)
  - [--help, 503](#)
  - [--package-name, 503](#)
  - [--package-version, 503](#)
  - [--path, 503](#)
  - [--project-id, 503](#)
  - [--sudo, 503](#)



---

-h, 503  
 gitlab-geo-node command line option  
 --help, 504  
 -h, 504  
 gitlab-geo-node-current-failures command line option  
 --help, 504  
 --id, 504  
 -h, 504  
 gitlab-geo-node-delete command line option  
 --help, 504  
 --id, 504  
 --sudo, 504  
 -h, 504  
 gitlab-geo-node-get command line option  
 --help, 504  
 --id, 504  
 --sudo, 504  
 -h, 504  
 gitlab-geo-node-list command line option  
 --get-all, 505  
 --help, 505  
 --page, 505  
 --per-page, 505  
 --sudo, 505  
 -h, 505  
 gitlab-geo-node-repair command line option  
 --help, 505  
 --id, 505  
 -h, 505  
 gitlab-geo-node-status command line option  
 --help, 505  
 --id, 505  
 -h, 505  
 gitlab-geo-node-update command line option  
 --enabled, 505  
 --files-max-capacity, 505  
 --help, 505  
 --id, 505  
 --repos-max-capacity, 506  
 --sudo, 505  
 --url, 505  
 -h, 505  
 gitlab-gitignore command line option  
 --help, 506  
 -h, 506  
 gitlab-gitignore-get command line option  
 --help, 506  
 --name, 506  
 --sudo, 506  
 -h, 506  
 gitlab-gitignore-list command line option  
 --get-all, 506  
 --help, 506  
 --page, 506  
 --per-page, 506  
 --sudo, 506  
 -h, 506  
 gitlab-gitlabciyaml command line option  
 --help, 506  
 -h, 506  
 gitlab-gitlabciyaml-get command line option  
 --help, 507  
 --name, 507  
 --sudo, 507  
 -h, 507  
 gitlab-gitlabciyaml-list command line option  
 --get-all, 507  
 --help, 507  
 --page, 507  
 --per-page, 507  
 --sudo, 507  
 -h, 507  
 gitlab-group command line option  
 --help, 507  
 -h, 507  
 gitlab-group-access-request command line option  
 --help, 513  
 -h, 513  
 gitlab-group-access-request-approve command line option  
 --access-level, 513  
 --group-id, 513  
 --help, 513  
 --id, 513  
 --sudo, 513  
 -h, 513  
 gitlab-group-access-request-create command line option  
 --group-id, 514  
 --help, 514  
 --sudo, 514  
 -h, 514  
 gitlab-group-access-request-delete command line option  
 --group-id, 514  
 --help, 514  
 --id, 514  
 --sudo, 514  
 -h, 514  
 gitlab-group-access-request-list command line option  
 --get-all, 514  
 --group-id, 514  
 --help, 514  
 --page, 514  
 --per-page, 514

```
--sudo, 514
-h, 514
gitlab-group-access-token command line
    option
--help, 515
-h, 515
gitlab-group-access-token-create command
    line option
--access-level, 515
--expires-at, 515
--group-id, 515
--help, 515
--name, 515
--scopes, 515
--sudo, 515
-h, 515
gitlab-group-access-token-delete command
    line option
--group-id, 515
--help, 515
--id, 515
--sudo, 515
-h, 515
gitlab-group-access-token-get command line
    option
--group-id, 515
--help, 515
--id, 516
--sudo, 515
-h, 515
gitlab-group-access-token-list command line
    option
--get-all, 516
--group-id, 516
--help, 516
--page, 516
--per-page, 516
--sudo, 516
-h, 516
gitlab-group-audit-event command line
    option
--help, 516
-h, 516
gitlab-group-audit-event-get command line
    option
--group-id, 516
--help, 516
--id, 516
--sudo, 516
-h, 516
gitlab-group-audit-event-list command line
    option
--created-after, 517
--created-before, 517
--get-all, 517
--group-id, 517
--help, 517
--page, 517
--per-page, 517
--sudo, 517
-h, 517
gitlab-group-badge command line option
--help, 517
-h, 517
gitlab-group-badge-create command line
    option
--group-id, 517
--help, 517
--image-url, 517
--link-url, 517
--sudo, 517
-h, 517
gitlab-group-badge-delete command line
    option
--group-id, 518
--help, 518
--id, 518
--sudo, 518
-h, 518
gitlab-group-badge-get command line option
--group-id, 518
--help, 518
--id, 518
--sudo, 518
-h, 518
gitlab-group-badge-list command line option
--get-all, 518
--group-id, 518
--help, 518
--page, 518
--per-page, 518
--sudo, 518
-h, 518
gitlab-group-badge-render command line
    option
--group-id, 519
--help, 519
--id, 519
--image-url, 519
--link-url, 519
--sudo, 519
-h, 519
gitlab-group-badge-update command line
    option
--group-id, 519
--help, 519
--id, 519
--image-url, 519
```

```

--link-url, 519
--sudo, 519
-h, 519
gitlab-group-billable-member command line
    option
--help, 519
-h, 519
gitlab-group-billable-member-delete command
    line option
--group-id, 520
--help, 519
--id, 520
--sudo, 519
-h, 519
gitlab-group-billable-member-list command
    line option
--get-all, 520
--group-id, 520
--help, 520
--page, 520
--per-page, 520
--search, 520
--sort, 520
--sudo, 520
-h, 520
gitlab-group-billable-member-membership
    command line option
--help, 520
-h, 520
gitlab-group-billable-member-membership-list
    command line option
--get-all, 521
--group-id, 520
--help, 520
--page, 521
--per-page, 521
--sudo, 520
--user-id, 521
-h, 520
gitlab-group-board command line option
--help, 521
-h, 521
gitlab-group-board-create command line
    option
--group-id, 521
--help, 521
--name, 521
--sudo, 521
-h, 521
gitlab-group-board-delete command line
    option
--group-id, 521
--help, 521
--id, 521
--sudo, 521
-h, 521
gitlab-group-board-get command line option
--group-id, 522
--help, 522
--id, 522
--sudo, 522
-h, 522
gitlab-group-board-list command line option
--get-all, 522
--group-id, 522
--help, 522, 523
--page, 522
--per-page, 522
--sudo, 522
-h, 522, 523
gitlab-group-board-list-create command line
    option
--assignee-id, 523
--board-id, 523
--group-id, 523
--help, 523
--label-id, 523
--milestone-id, 523
--sudo, 523
-h, 523
gitlab-group-board-list-delete command line
    option
--board-id, 523
--group-id, 523
--help, 523
--id, 523
--sudo, 523
-h, 523
gitlab-group-board-list-get command line
    option
--board-id, 524
--group-id, 524
--help, 524
--id, 524
--sudo, 524
-h, 524
gitlab-group-board-list-list command line
    option
--board-id, 524
--get-all, 524
--group-id, 524
--help, 524
--page, 524
--per-page, 524
--sudo, 524
-h, 524
gitlab-group-board-list-update command line
    option

```

```
--board-id, 524
--group-id, 524
--help, 524
--id, 524
--position, 524
--sudo, 524
-h, 524
gitlab-group-board-update command line
    option
--group-id, 522
--help, 522
--id, 522
--sudo, 522
-h, 522
gitlab-group-cluster command line option
--help, 525
-h, 525
gitlab-group-cluster-create command line
    option
--domain, 525
--enabled, 525
--environment-scope, 525
--group-id, 525
--help, 525
--managed, 525
--name, 525
--platform-kubernetes-attributes, 525
--sudo, 525
-h, 525
gitlab-group-cluster-delete command line
    option
--group-id, 525
--help, 525
--id, 525
--sudo, 525
-h, 525
gitlab-group-cluster-get command line
    option
--group-id, 526
--help, 526
--id, 526
--sudo, 526
-h, 526
gitlab-group-cluster-list command line
    option
--get-all, 526
--group-id, 526
--help, 526
--page, 526
--per-page, 526
--sudo, 526
-h, 526
gitlab-group-cluster-update command line
    option
--domain, 526
--environment-scope, 527
--group-id, 526
--help, 526
--id, 526
--management-project-id, 526
--name, 526
--platform-kubernetes-attributes, 527
--sudo, 526
-h, 526
gitlab-group-create command line option
--auto-devops-enabled, 508
--avatar, 508
--default-branch-protection, 509
--description, 508
--emails-disabled, 508
--extra-shared-runners-minutes-limit, 509
--help, 508
--lfs-enabled, 508
--membership-lock, 508
--mentions-disabled, 508
--name, 508
--parent-id, 509
--path, 508
--project-creation-level, 508
--request-access-enabled, 509
--require-two-factor-authentication, 508
--share-with-group-lock, 508
--shared-runners-minutes-limit, 509
--subgroup-creation-level, 508
--sudo, 508
--two-factor-grace-period, 508
--visibility, 508
-h, 508
gitlab-group-custom-attribute command line
    option
--help, 527
-h, 527
gitlab-group-custom-attribute-delete
    command line option
--group-id, 527
--help, 527
--key, 527
--sudo, 527
-h, 527
gitlab-group-custom-attribute-get command
    line option
--group-id, 527
--help, 527
--key, 527
--sudo, 527
-h, 527
gitlab-group-custom-attribute-list command
    line option
```

```

--get-all, 528
--group-id, 528
--help, 527
--page, 528
--per-page, 528
--sudo, 527
-h, 527
gitlab-group-delete command line option
--help, 509
--id, 509
--sudo, 509
-h, 509
gitlab-group-deploy-token command line
option
--help, 528
-h, 528
gitlab-group-deploy-token-create command
line option
--expires-at, 528
--group-id, 528
--help, 528
--name, 528
--scopes, 528
--sudo, 528
--username, 528
-h, 528
gitlab-group-deploy-token-delete command
line option
--group-id, 528
--help, 528
--id, 528
--sudo, 528
-h, 528
gitlab-group-deploy-token-get command line
option
--group-id, 529
--help, 529
--id, 529
--sudo, 529
-h, 529
gitlab-group-deploy-token-list command line
option
--get-all, 529
--group-id, 529
--help, 529
--page, 529
--per-page, 529
--scopes, 529
--sudo, 529
-h, 529
gitlab-group-descendant-group command line
option
--help, 529
-h, 529
gitlab-group-descendant-group-list command
line option
--all-available, 530
--get-all, 530
--group-id, 530
--help, 530
--min-access-level, 530
--order-by, 530
--owned, 530
--page, 530
--per-page, 530
--search, 530
--skip-groups, 530
--sort, 530
--statistics, 530
--sudo, 530
--with-custom-attributes, 530
-h, 530
gitlab-group-epic command line option
--help, 530
-h, 530
gitlab-group-epic-award-emoji command line
option
--help, 533
-h, 533
gitlab-group-epic-award-emoji-create
command line option
--epic-iid, 533
--group-id, 533
--help, 533
--name, 533
--sudo, 533
-h, 533
gitlab-group-epic-award-emoji-delete
command line option
--epic-iid, 533
--group-id, 533
--help, 533
--id, 533
--sudo, 533
-h, 533
gitlab-group-epic-award-emoji-get command
line option
--epic-iid, 533
--group-id, 533
--help, 533
--id, 534
--sudo, 533
-h, 533
gitlab-group-epic-award-emoji-list command
line option
--epic-iid, 534
--get-all, 534
--group-id, 534

```

```
--help, 534
--page, 534
--per-page, 534
--sudo, 534
-h, 534
gitlab-group-epic-create command line
    option
--description, 531
--end-date, 531
--group-id, 531
--help, 531
--labels, 531
--start-date, 531
--sudo, 531
--title, 531
-h, 531
gitlab-group-epic-delete command line
    option
--group-id, 531
--help, 531
--iid, 531
--sudo, 531
-h, 531
gitlab-group-epic-discussion-note command
line option
--help, 534
-h, 534
gitlab-group-epic-discussion-note-create
command line option
--body, 534
--created-at, 535
--discussion-id, 534
--epic-id, 534
--group-id, 534
--help, 534
--sudo, 534
-h, 534
gitlab-group-epic-discussion-note-delete
command line option
--discussion-id, 535
--epic-id, 535
--group-id, 535
--help, 535
--id, 535
--sudo, 535
-h, 535
gitlab-group-epic-discussion-note-get
command line option
--discussion-id, 535
--epic-id, 535
--group-id, 535
--help, 535
--id, 535
--sudo, 535
-h, 535
gitlab-group-epic-discussion-note-update
command line option
--body, 536
--discussion-id, 536
--epic-id, 536
--group-id, 536
--help, 535
--id, 536
--sudo, 535
-h, 535
gitlab-group-epic-get command line option
--group-id, 531
--help, 531
--iid, 531
--sudo, 531
-h, 531
gitlab-group-epic-issue command line option
--help, 536
-h, 536
gitlab-group-epic-issue-create command line
    option
--epic-iid, 536
--group-id, 536
--help, 536
--issue-id, 536
--sudo, 536
-h, 536
gitlab-group-epic-issue-delete command line
    option
--epic-iid, 536
--epic-issue-id, 536
--group-id, 536
--help, 536
--sudo, 536
-h, 536
gitlab-group-epic-issue-list command line
    option
--epic-iid, 537
--get-all, 537
--group-id, 537
--help, 537
--page, 537
--per-page, 537
--sudo, 537
-h, 537
gitlab-group-epic-issue-update command line
    option
--epic-iid, 537
--epic-issue-id, 537
--group-id, 537
--help, 537
--move-after-id, 537
--move-before-id, 537
```

```

--sudo, 537
-h, 537
gitlab-group-epic-list command line option
--author-id, 532
--get-all, 532
--group-id, 532
--help, 532
--labels, 532
--order-by, 532
--page, 532
--per-page, 532
--search, 532
--sort, 532
--sudo, 532
-h, 532
gitlab-group-epic-note command line option
--help, 537
-h, 537
gitlab-group-epic-note-award-emoji command
line option
--help, 539
-h, 539
gitlab-group-epic-note-award-emoji-create
command line option
--epic-iid, 540
--group-id, 540
--help, 540
--name, 540
--note-id, 540
--sudo, 540
-h, 540
gitlab-group-epic-note-award-emoji-delete
command line option
--epic-iid, 540
--group-id, 540
--help, 540
--id, 540
--note-id, 540
--sudo, 540
-h, 540
gitlab-group-epic-note-award-emoji-get
command line option
--epic-iid, 540
--group-id, 540
--help, 540
--id, 540
--note-id, 540
--sudo, 540
-h, 540
gitlab-group-epic-note-award-emoji-list
command line option
--epic-iid, 541
--get-all, 541
--group-id, 541
--help, 541
--note-id, 541
--page, 541
--per-page, 541
--sudo, 541
-h, 541
gitlab-group-epic-note-create command line
option
--body, 538
--created-at, 538
--epic-id, 538
--group-id, 538
--help, 538
--sudo, 538
-h, 538
gitlab-group-epic-note-delete command line
option
--epic-id, 538
--group-id, 538
--help, 538
--id, 538
--sudo, 538
-h, 538
gitlab-group-epic-note-get command line
option
--epic-id, 538
--group-id, 538
--help, 538
--id, 538
--sudo, 538
-h, 538
gitlab-group-epic-note-list command line
option
--epic-id, 539
--get-all, 539
--group-id, 539
--help, 539
--page, 539
--per-page, 539
--sudo, 539
-h, 539
gitlab-group-epic-note-update command line
option
--body, 539
--epic-id, 539
--group-id, 539
--help, 539
--id, 539
--sudo, 539
-h, 539
gitlab-group-epic-resource-label-event
command line option
--help, 541
-h, 541

```

`gitlab-group-epic-resource-label-event-get`  
command line option  
--epic-id, 541  
--group-id, 541  
--help, 541  
--id, 541  
--sudo, 541  
-h, 541

`gitlab-group-epic-resource-label-event-list`  
command line option  
--epic-id, 542  
--get-all, 542  
--group-id, 542  
--help, 542  
--page, 542  
--per-page, 542  
--sudo, 542  
-h, 542

`gitlab-group-epic-update` command line option  
--description, 532  
--end-date, 532  
--group-id, 532  
--help, 532  
--iid, 532  
--labels, 532  
--start-date, 532  
--sudo, 532  
--title, 532  
-h, 532

`gitlab-group-export` command line option  
--help, 542  
-h, 542

`gitlab-group-export-create` command line option  
--group-id, 542  
--help, 542  
--sudo, 542  
-h, 542

`gitlab-group-export-download` command line option  
--group-id, 542  
--help, 542  
--sudo, 542  
-h, 542

`gitlab-group-export-get` command line option  
--group-id, 543  
--help, 543  
--sudo, 543  
-h, 543

`gitlab-group-get` command line option  
--help, 509  
--id, 509  
--sudo, 509  
-h, 509

`gitlab-group-hook` command line option  
--help, 543  
-h, 543

`gitlab-group-hook-create` command line option  
--confidential-issues-events, 543  
--confidential-note-events, 544  
--deployment-events, 544  
--enable-ssl-verification, 544  
--group-id, 543  
--help, 543  
--issues-events, 543  
--job-events, 544  
--merge-requests-events, 543  
--note-events, 544  
--pipeline-events, 544  
--push-events, 543  
--releases-events, 544  
--subgroup-events, 544  
--sudo, 543  
--tag-push-events, 543  
--token, 544  
--url, 543  
--wiki-page-events, 544  
-h, 543

`gitlab-group-hook-delete` command line option  
--group-id, 544  
--help, 544  
--id, 544  
--sudo, 544  
-h, 544

`gitlab-group-hook-get` command line option  
--group-id, 544  
--help, 544  
--id, 544  
--sudo, 544  
-h, 544

`gitlab-group-hook-list` command line option  
--get-all, 545  
--group-id, 545  
--help, 545  
--page, 545  
--per-page, 545  
--sudo, 545  
-h, 545

`gitlab-group-hook-update` command line option  
--confidential-issues-events, 545  
--confidential-note-events, 546  
--deployment-events, 546  
--enable-ssl-verification, 546  
--group-id, 545



```

--help, 545
--id, 545
--issues-events, 545
--job-events, 546
--merge-requests-events, 545
--note-events, 546
--pipeline-events, 546
--push-events, 545
--releases-events, 546
--subgroup-events, 546
--sudo, 545
--tag-push-events, 546
--token, 546
--url, 545
--wiki-page-events, 546
-h, 545
gitlab-group-import command line option
--help, 546
-h, 546
gitlab-group-import-get command line option
--group-id, 546
--help, 546
--sudo, 546
-h, 546
gitlab-group-invitation command line option
--help, 546
-h, 546
gitlab-group-invitation-create command line
    option
--access-level, 547
--email, 547
--expires-at, 547
--group-id, 547
--help, 547
--invite-source, 547
--sudo, 547
--tasks-project-id, 547
--tasks-to-be-done, 547
--user-id, 547
-h, 547
gitlab-group-invitation-delete command line
    option
--email, 547
--group-id, 547
--help, 547
--sudo, 547
-h, 547
gitlab-group-invitation-get command line
    option
--email, 548
--group-id, 547
--help, 547
--sudo, 547
-h, 547
gitlab-group-invitation-list command line
    option
--get-all, 548
--group-id, 548
--help, 548
--page, 548
--per-page, 548
--query, 548
--sudo, 548
-h, 548
gitlab-group-invitation-update command line
    option
--access-level, 548
--email, 548
--expires-at, 548
--group-id, 548
--help, 548
--sudo, 548
-h, 548
gitlab-group-issue command line option
--help, 549
-h, 549
gitlab-group-issue-list command line option
--assignee-id, 549
--author-id, 549
--created-after, 549
--created-before, 549
--get-all, 550
--group-id, 549
--help, 549
--iids, 549
--iteration-id, 549
--labels, 549
--milestone, 549
--my-reaction-emoji, 549
--order-by, 549
--page, 550
--per-page, 550
--search, 549
--sort, 549
--state, 549
--sudo, 549
--updated-after, 550
--updated-before, 550
-h, 549
gitlab-group-issues-statistics command line
    option
--help, 550
-h, 550
gitlab-group-issues-statistics-get command
    line option
--group-id, 550
--help, 550
--sudo, 550

```

-h, 550  
gitlab-group-iteration command line option  
--help, 550  
-h, 550  
gitlab-group-iteration-list command line option  
--get-all, 551  
--group-id, 551  
--help, 550  
--include-ancestors, 551  
--page, 551  
--per-page, 551  
--search, 551  
--state, 551  
--sudo, 550  
-h, 550  
gitlab-group-label command line option  
--help, 551  
-h, 551  
gitlab-group-label-create command line option  
--color, 551  
--description, 551  
--group-id, 551  
--help, 551  
--name, 551  
--priority, 551  
--sudo, 551  
-h, 551  
gitlab-group-label-delete command line option  
--group-id, 552  
--help, 552  
--name, 552  
--sudo, 552  
-h, 552  
gitlab-group-label-get command line option  
--group-id, 552  
--help, 552  
--name, 552  
--sudo, 552  
-h, 552  
gitlab-group-label-list command line option  
--get-all, 552  
--group-id, 552  
--help, 552  
--page, 552  
--per-page, 552  
--sudo, 552  
-h, 552  
gitlab-group-label-subscribe command line option  
--group-id, 553  
--help, 553  
--name, 553  
--sudo, 553  
-h, 553  
gitlab-group-label-unsubscribe command line option  
--group-id, 553  
--help, 553  
--name, 553  
--sudo, 553  
-h, 553  
gitlab-group-label-update command line option  
--color, 553  
--description, 553  
--group-id, 553  
--help, 553  
--name, 553  
--new-name, 553  
--priority, 553  
--sudo, 553  
-h, 553  
gitlab-group-ldap-group-link command line option  
--help, 554  
-h, 554  
gitlab-group-ldap-group-link-create command line option  
--cn, 554  
--filter, 554  
--group-access, 554  
--group-id, 554  
--help, 554  
--provider, 554  
--sudo, 554  
-h, 554  
gitlab-group-ldap-group-link-delete command line option  
--group-id, 554  
--help, 554  
--id, 554  
--sudo, 554  
-h, 554  
gitlab-group-ldap-group-link-list command line option  
--get-all, 555  
--group-id, 555  
--help, 554  
--page, 555  
--per-page, 555  
--sudo, 554  
-h, 554  
gitlab-group-ldap-sync command line option  
--help, 509  
--id, 509

---

```

    -h, 509
gitlab-group-list command line option
    --all-available, 510
    --get-all, 510
    --help, 510
    --min-access-level, 510
    --order-by, 510
    --owned, 510
    --page, 510
    --per-page, 510
    --search, 510
    --skip-groups, 510
    --sort, 510
    --statistics, 510
    --sudo, 510
    --top-level-only, 510
    --with-custom-attributes, 510
    -h, 510
gitlab-group-member command line option
    --help, 555
    -h, 555
gitlab-group-member-all command line option
    --help, 557
    -h, 557
gitlab-group-member-all-get command line
    option
    --group-id, 557
    --help, 557
    --id, 557
    --sudo, 557
    -h, 557
gitlab-group-member-all-list command line
    option
    --get-all, 557
    --group-id, 557
    --help, 557
    --page, 557
    --per-page, 557
    --sudo, 557
    -h, 557
gitlab-group-member-create command line
    option
    --access-level, 555
    --expires-at, 555
    --group-id, 555
    --help, 555
    --sudo, 555
    --tasks-to-be-done, 555
    --user-id, 555
    -h, 555
gitlab-group-member-delete command line
    option
    --group-id, 555
    --help, 555
    --id, 555
    --sudo, 555
    -h, 555
gitlab-group-member-get command line option
    --group-id, 556
    --help, 556
    --id, 556
    --sudo, 556
    -h, 556
gitlab-group-member-list command line
    option
    --get-all, 556
    --group-id, 556
    --help, 556
    --page, 556
    --per-page, 556
    --sudo, 556
    -h, 556
gitlab-group-member-update command line
    option
    --access-level, 556
    --expires-at, 556
    --group-id, 556
    --help, 556
    --id, 556
    --sudo, 556
    -h, 556
gitlab-group-merge-request command line
    option
    --help, 557
    -h, 557
gitlab-group-merge-request-list command
    line option
    --approved-by-ids, 558
    --approver-ids, 558
    --assignee-id, 558
    --author-id, 558
    --created-after, 558
    --created-before, 558
    --get-all, 559
    --group-id, 558
    --help, 558
    --labels, 558
    --milestone, 558
    --my-reaction-emoji, 558
    --order-by, 558
    --page, 559
    --per-page, 559
    --scope, 558
    --search, 559
    --sort, 558
    --source-branch, 559
    --state, 558
    --sudo, 558

```

```
--target-branch, 559
--updated-after, 558
--updated-before, 558
--view, 558
--wip, 559
-h, 558
gitlab-group-milestone command line option
--help, 559
-h, 559
gitlab-group-milestone-create command line
    option
--description, 559
--due-date, 559
--group-id, 559
--help, 559
--start-date, 559
--sudo, 559
--title, 559
-h, 559
gitlab-group-milestone-delete command line
    option
--group-id, 560
--help, 560
--id, 560
--sudo, 560
-h, 560
gitlab-group-milestone-get command line
    option
--group-id, 560
--help, 560
--id, 560
--sudo, 560
-h, 560
gitlab-group-milestone-issues command line
    option
--group-id, 560
--help, 560
--id, 560
--sudo, 560
-h, 560
gitlab-group-milestone-list command line
    option
--get-all, 561
--group-id, 561
--help, 560
--iids, 561
--page, 561
--per-page, 561
--search, 561
--state, 561
--sudo, 560
-h, 560
gitlab-group-milestone-merge-requests
    command line option
--group-id, 561
--help, 561
--id, 561
--sudo, 561
-h, 561
gitlab-group-milestone-update command line
    option
--description, 561
--due-date, 561
--group-id, 561
--help, 561
--id, 561
--start-date, 561
--state-event, 562
--sudo, 561
--title, 561
-h, 561
gitlab-group-notification-settings command
    line option
--help, 562
-h, 562
gitlab-group-notification-settings-get
    command line option
--group-id, 562
--help, 562
--sudo, 562
-h, 562
gitlab-group-notification-settings-update
    command line option
--close-issue, 563
--close-merge-request, 563
--group-id, 562
--help, 562
--level, 562
--merge-merge-request, 563
--new-issue, 563
--new-merge-request, 563
--new-note, 563
--notification-email, 563
--reassign-issue, 563
--reassign-merge-request, 563
--reopen-issue, 563
--reopen-merge-request, 563
--sudo, 562
-h, 562
gitlab-group-package command line option
--help, 563
-h, 563
gitlab-group-package-list command line
    option
--exclude-subgroups, 563
--get-all, 564
--group-id, 563
--help, 563
```

```

--order-by, 563
--package-name, 563
--package-type, 563
--page, 564
--per-page, 564
--sort, 563
--sudo, 563
-h, 563
gitlab-group-project command line option
--help, 564
-h, 564
gitlab-group-project-list command line
    option
--archived, 564
--get-all, 565
--group-id, 564
--help, 564
--include-subgroups, 565
--min-access-level, 565
--order-by, 564
--owned, 564
--page, 565
--per-page, 565
--search, 564
--simple, 564
--sort, 564
--starred, 565
--sudo, 564
--visibility, 564
--with-custom-attributes, 565
--with-issues-enabled, 565
--with-merge-requests-enabled, 565
--with-security-reports, 565
--with-shared, 565
-h, 564
gitlab-group-push-rules command line option
--help, 565
-h, 565
gitlab-group-push-rules-create command line
    option
--author-email-regex, 566
--branch-name-regex, 566
--commit-committer-check, 566
--commit-message-negative-regex, 566
--commit-message-regex, 566
--deny-delete-tag, 565
--file-name-regex, 566
--group-id, 565
--help, 565
--max-file-size, 566
--member-check, 566
--prevent-secrets, 566
--reject-unsigned-commits, 566
--sudo, 565
-h, 565
gitlab-group-push-rules-delete command line
    option
--group-id, 566
--help, 566
--sudo, 566
-h, 566
gitlab-group-push-rules-get command line
    option
--group-id, 566
--help, 566
--sudo, 566
-h, 566
gitlab-group-push-rules-update command line
    option
--author-email-regex, 567
--branch-name-regex, 567
--commit-committer-check, 567
--commit-message-negative-regex, 567
--commit-message-regex, 567
--deny-delete-tag, 567
--file-name-regex, 567
--group-id, 567
--help, 567
--max-file-size, 567
--member-check, 567
--prevent-secrets, 567
--reject-unsigned-commits, 567
--sudo, 567
-h, 567
gitlab-group-restore command line option
--help, 510
--id, 510
-h, 510
gitlab-group-runner command line option
--help, 567
-h, 567
gitlab-group-runner-list command line
    option
--get-all, 568
--group-id, 568
--help, 567
--page, 568
--per-page, 568
--scope, 568
--sudo, 567
--tag-list, 568
-h, 567
gitlab-group-saml-group-link command line
    option
--help, 568
-h, 568
gitlab-group-saml-group-link-create command
    line option

```

```
--access-level, 568
--group-id, 568
--help, 568
--saml-group-name, 568
--sudo, 568
-h, 568
gitlab-group-saml-group-link-delete command
    line option
--group-id, 568
--help, 568
--name, 568
--sudo, 568
-h, 568
gitlab-group-saml-group-link-get command
    line option
--group-id, 569
--help, 569
--name, 569
--sudo, 569
-h, 569
gitlab-group-saml-group-link-list command
    line option
--get-all, 569
--group-id, 569
--help, 569
--page, 569
--per-page, 569
--sudo, 569
-h, 569
gitlab-group-search command line option
--help, 510
--id, 510
--scope, 510
--search, 510
-h, 510
gitlab-group-share command line option
--expires-at, 511
--group-access, 511
--group-id, 511
--help, 511
--id, 511
-h, 511
gitlab-group-subgroup command line option
--help, 569
-h, 569
gitlab-group-subgroup-list command line
    option
--all-available, 570
--get-all, 570
--group-id, 570
--help, 570
--min-access-level, 570
--order-by, 570
--owned, 570
--page, 570
--per-page, 570
--search, 570
--skip-groups, 570
--sort, 570
--statistics, 570
--sudo, 570
--with-custom-attributes, 570
-h, 570
gitlab-group-transfer command line option
--group-id, 511
--help, 511
--id, 511
-h, 511
gitlab-group-transfer-project command line
    option
--help, 511
--id, 511
--project-id, 511
-h, 511
gitlab-group-unshare command line option
--group-id, 511
--help, 511
--id, 511
-h, 511
gitlab-group-update command line option
--auto-devops-enabled, 512
--avatar, 513
--default-branch-protection, 513
--description, 512
--emails-disabled, 512
--extra-shared-runners-minutes-limit, 513
--file-template-project-id, 513
--help, 512
--id, 512
--lfs-enabled, 513
--membership-lock, 512
--mentions-disabled, 513
--name, 512
--path, 512
--prevent-forking-outside-group, 513
--project-creation-level, 512
--request-access-enabled, 513
--require-two-factor-authentication, 512
--share-with-group-lock, 512
--shared-runners-minutes-limit, 513
--shared-runners-setting, 513
--subgroup-creation-level, 512
--sudo, 512
--two-factor-grace-period, 512
--visibility, 512
-h, 512
gitlab-group-variable command line option
--help, 570
```

---

```

    -h, 570
gitlab-group-variable-create command line
    option
    --group-id, 571
    --help, 570
    --key, 571
    --masked, 571
    --protected, 571
    --sudo, 570
    --value, 571
    --variable-type, 571
    -h, 570
gitlab-group-variable-delete command line
    option
    --group-id, 571
    --help, 571
    --key, 571
    --sudo, 571
    -h, 571
gitlab-group-variable-get command line
    option
    --group-id, 571
    --help, 571
    --key, 571
    --sudo, 571
    -h, 571
gitlab-group-variable-list command line
    option
    --get-all, 572
    --group-id, 571
    --help, 571
    --page, 572
    --per-page, 572
    --sudo, 571
    -h, 571
gitlab-group-variable-update command line
    option
    --group-id, 572
    --help, 572
    --key, 572
    --masked, 572
    --protected, 572
    --sudo, 572
    --value, 572
    --variable-type, 572
    -h, 572
gitlab-group-wiki command line option
    --help, 572
    -h, 572
gitlab-group-wiki-create command line
    option
    --content, 573
    --format, 573
    --group-id, 572
    --help, 572
    --sudo, 572
    -h, 572
gitlab-group-wiki-delete command line
    option
    --group-id, 573
    --help, 573
    --slug, 573
    --sudo, 573
    -h, 573
gitlab-group-wiki-get command line option
    --group-id, 573
    --help, 573
    --slug, 573
    --sudo, 573
    -h, 573
gitlab-group-wiki-list command line option
    --get-all, 573
    --group-id, 573
    --help, 573
    --page, 573
    --per-page, 573
    --sudo, 573
    --with-content, 573
    -h, 573
gitlab-group-wiki-update command line
    option
    --content, 574
    --format, 574
    --group-id, 574
    --help, 574
    --slug, 574
    --sudo, 574
    --title, 574
    -h, 574
gitlab-hook command line option
    --help, 574
    -h, 574
gitlab-hook-create command line option
    --help, 574
    --sudo, 574
    --url, 574
    -h, 574
gitlab-hook-delete command line option
    --help, 574
    --id, 574
    --sudo, 574
    -h, 574
gitlab-hook-get command line option
    --help, 575
    --id, 575
    --sudo, 575
    -h, 575

```

gitlab-hook-list command line option

- get-all, 575
- help, 575
- page, 575
- per-page, 575
- sudo, 575
- h, 575

gitlab-issue command line option

- help, 575
- h, 575

gitlab-issue-get command line option

- help, 575
- id, 575
- sudo, 575
- h, 575

gitlab-issue-list command line option

- assignee-id, 576
- author-id, 576
- created-after, 576
- created-before, 576
- get-all, 576
- help, 576
- iids, 576
- iteration-id, 576
- labels, 576
- milestone, 576
- my-reaction-emoji, 576
- order-by, 576
- page, 576
- per-page, 576
- scope, 576
- search, 576
- sort, 576
- state, 576
- sudo, 576
- updated-after, 576
- updated-before, 576
- h, 576

gitlab-issues-statistics command line option

- help, 577
- h, 577

gitlab-issues-statistics-get command line option

- help, 577
- sudo, 577
- h, 577

gitlab-key command line option

- help, 577
- h, 577

gitlab-key-get command line option

- help, 577
- id, 577
- sudo, 577

- h, 577

gitlab-ldap-group command line option

- help, 577
- h, 577

gitlab-ldap-group-list command line option

- get-all, 578
- help, 578
- page, 578
- per-page, 578
- provider, 578
- search, 578
- sudo, 578
- h, 578

gitlab-license command line option

- help, 578
- h, 578

gitlab-license-get command line option

- fullname, 578
- help, 578
- key, 578
- project, 578
- sudo, 578
- h, 578

gitlab-license-list command line option

- get-all, 579
- help, 579
- page, 579
- per-page, 579
- popular, 579
- sudo, 579
- h, 579

gitlab-merge-request command line option

- help, 579
- h, 579

gitlab-merge-request-list command line option

- approved-by-ids, 580
- approver-ids, 580
- assignee-id, 580
- author-id, 580
- author-username, 580
- created-after, 580
- created-before, 580
- deployed-after, 581
- deployed-before, 581
- environment, 581
- get-all, 581
- help, 580
- in, 580
- labels, 580
- milestone, 580
- my-reaction-emoji, 580
- not, 580
- order-by, 580



```

--page, 581
--per-page, 581
--reviewer-id, 580
--reviewer-username, 580
--scope, 580
--search, 580
--sort, 580
--source-branch, 580
--state, 580
--sudo, 580
--target-branch, 580
--updated-after, 580
--updated-before, 580
--view, 580
--wip, 580
--with-labels-details, 580
--with-merge-status-recheck, 580
-h, 580
gitlab-namespace command line option
--help, 581
-h, 581
gitlab-namespace-exists command line option
--help, 581
--id, 581
--namespace, 581
--parent-id, 581
-h, 581
gitlab-namespace-get command line option
--help, 581
--id, 581
--sudo, 581
-h, 581
gitlab-namespace-list command line option
--get-all, 582
--help, 582
--page, 582
--per-page, 582
--search, 582
--sudo, 582
-h, 582
gitlab-notification-settings command line
  option
--help, 582
-h, 582
gitlab-notification-settings-get command
  line option
--help, 582
--sudo, 582
-h, 582
gitlab-notification-settings-update command
  line option
--close-issue, 583
--close-merge-request, 583
--help, 582
--level, 583
--merge-merge-request, 583
--new-issue, 583
--new-merge-request, 583
--new-note, 583
--notification-email, 583
--reassign-issue, 583
--reassign-merge-request, 583
--reopen-issue, 583
--reopen-merge-request, 583
--sudo, 583
-h, 582
gitlab-pages-domain command line option
--help, 583
-h, 583
gitlab-pages-domain-list command line
  option
--get-all, 583
--help, 583
--page, 583
--per-page, 583
--sudo, 583
-h, 583
gitlab-personal-access-token command line
  option
--help, 584
-h, 584
gitlab-personal-access-token-delete command
  line option
--help, 584
--id, 584
--sudo, 584
-h, 584
gitlab-personal-access-token-get command
  line option
--help, 584
--id, 584
--sudo, 584
-h, 584
gitlab-personal-access-token-list command
  line option
--get-all, 584
--help, 584
--page, 584
--per-page, 584
--sudo, 584
--user-id, 584
-h, 584
gitlab-project command line option
--help, 585
-h, 585
gitlab-project-access-request command line
  option
--help, 602

```

```
-h, 602
gitlab-project-access-request-approve
    command line option
    --access-level, 603
    --help, 603
    --id, 603
    --project-id, 603
    --sudo, 603
    -h, 603
gitlab-project-access-request-create
    command line option
    --help, 603
    --project-id, 603
    --sudo, 603
    -h, 603
gitlab-project-access-request-delete
    command line option
    --help, 603
    --id, 603
    --project-id, 603
    --sudo, 603
    -h, 603
gitlab-project-access-request-list command
    line option
    --get-all, 604
    --help, 603
    --page, 604
    --per-page, 604
    --project-id, 604
    --sudo, 603
    -h, 603
gitlab-project-access-token command line
    option
    --help, 604
    -h, 604
gitlab-project-access-token-create command
    line option
    --access-level, 604
    --expires-at, 604
    --help, 604
    --name, 604
    --project-id, 604
    --scopes, 604
    --sudo, 604
    -h, 604
gitlab-project-access-token-delete command
    line option
    --help, 604
    --id, 604
    --project-id, 604
    --sudo, 604
    -h, 604
gitlab-project-access-token-get command
    line option
    --help, 605
    --id, 605
    --project-id, 605
    --sudo, 605
    -h, 605
gitlab-project-access-token-list command
    line option
    --get-all, 605
    --help, 605
    --page, 605
    --per-page, 605
    --project-id, 605
    --sudo, 605
    -h, 605
gitlab-project-additional-statistics
    command line option
    --help, 605
    -h, 605
gitlab-project-additional-statistics-get
    command line option
    --help, 605
    --project-id, 605
    --sudo, 605
    -h, 605
gitlab-project-approval command line option
    --help, 606
    -h, 606
gitlab-project-approval-get command line
    option
    --help, 606
    --project-id, 606
    --sudo, 606
    -h, 606
gitlab-project-approval-rule command line
    option
    --help, 607
    -h, 607
gitlab-project-approval-rule-create command
    line option
    --approvals-required, 607
    --group-ids, 607
    --help, 607
    --name, 607
    --project-id, 607
    --protected-branch-ids, 607
    --sudo, 607
    --user-ids, 607
    --usernames, 607
    -h, 607
gitlab-project-approval-rule-delete command
    line option
    --help, 607
    --id, 607
    --project-id, 607
```

```

--sudo, 607
-h, 607
gitlab-project-approval-rule-list command
    line option
--get-all, 608
--help, 608
--page, 608
--per-page, 608
--project-id, 608
--sudo, 608
-h, 608
gitlab-project-approval-rule-update command
    line option
--help, 608
--id, 608
--project-id, 608
--sudo, 608
-h, 608
gitlab-project-approval-update command line
    option
--approvals-before-merge, 606
--disable-overriding-approvers-per-merge-request, line option
    606
--help, 606
--merge-requests-author-approval, 606
--merge-requests-disable-committers-approval,
    606
--project-id, 606
--reset-approvals-on-push, 606
--sudo, 606
-h, 606
gitlab-project-archive command line option
--help, 585
--id, 585
-h, 585
gitlab-project-artifact command line option
--help, 608
-h, 608
gitlab-project-artifact-delete command line
    option
--help, 608
--project-id, 608
--ref-name, 608
--sudo, 608
-h, 608
gitlab-project-artifact-download command
    line option
--help, 609
--job, 609
--job-token, 609
--project-id, 609
--ref-name, 609
--sudo, 609
-h, 609
gitlab-project-artifact-raw command line
    option
--artifact-path, 609
--help, 609
--job, 609
--project-id, 609
--ref-name, 609
--sudo, 609
-h, 609
gitlab-project-audit-event command line
    option
--help, 609
-h, 609
gitlab-project-audit-event-get command line
    option
--help, 610
--id, 610
--project-id, 610
--sudo, 610
-h, 610
gitlab-project-audit-event-list command
--created-after, 610
--created-before, 610
--get-all, 610
--help, 610
--page, 610
--per-page, 610
--project-id, 610
--sudo, 610
-h, 610
gitlab-project-badge command line option
--help, 610
-h, 610
gitlab-project-badge-create command line
    option
--help, 611
--image-url, 611
--link-url, 611
--project-id, 611
--sudo, 611
-h, 611
gitlab-project-badge-delete command line
    option
--help, 611
--id, 611
--project-id, 611
--sudo, 611
-h, 611
gitlab-project-badge-get command line
    option
--help, 611
--id, 611
--project-id, 611

```

```
--sudo, 611
-h, 611
gitlab-project-badge-list command line
    option
--get-all, 612
--help, 612
--page, 612
--per-page, 612
--project-id, 612
--sudo, 612
-h, 612
gitlab-project-badge-render command line
    option
--help, 612
--id, 612
--image-url, 612
--link-url, 612
--project-id, 612
--sudo, 612
-h, 612
gitlab-project-badge-update command line
    option
--help, 612
--id, 612
--image-url, 613
--link-url, 612
--project-id, 612
--sudo, 612
-h, 612
gitlab-project-board command line option
--help, 613
-h, 613
gitlab-project-board-create command line
    option
--help, 613
--name, 613
--project-id, 613
--sudo, 613
-h, 613
gitlab-project-board-delete command line
    option
--help, 613
--id, 613
--project-id, 613
--sudo, 613
-h, 613
gitlab-project-board-get command line
    option
--help, 613
--id, 613
--project-id, 613
--sudo, 613
-h, 613
```

```
gitlab-project-board-list command line
    option
--get-all, 614
--help, 614
--page, 614
--per-page, 614
--project-id, 614
--sudo, 614
-h, 614
gitlab-project-board-list-create command
    line option
--assignee-id, 615
--board-id, 615
--help, 614
--label-id, 615
--milestone-id, 615
--project-id, 615
--sudo, 614
-h, 614
gitlab-project-board-list-delete command
    line option
--board-id, 615
--help, 615
--id, 615
--project-id, 615
--sudo, 615
-h, 615
gitlab-project-board-list-get command line
    option
--board-id, 615
--help, 615
--id, 615
--project-id, 615
--sudo, 615
-h, 615
gitlab-project-board-list-list command line
    option
--board-id, 616
--get-all, 616
--help, 615
--page, 616
--per-page, 616
--project-id, 616
--sudo, 615
-h, 615
gitlab-project-board-list-update command
    line option
--board-id, 616
--help, 616
--id, 616
--position, 616
--project-id, 616
--sudo, 616
-h, 616
```

gitlab-project-board-update command line option

--help, 614  
--id, 614  
--project-id, 614  
--sudo, 614  
-h, 614

gitlab-project-branch command line option

--help, 616  
-h, 616

gitlab-project-branch-create command line option

--branch, 616  
--help, 616  
--project-id, 616  
--ref, 616  
--sudo, 616  
-h, 616

gitlab-project-branch-delete command line option

--help, 617  
--name, 617  
--project-id, 617  
--sudo, 617  
-h, 617

gitlab-project-branch-get command line option

--help, 617  
--name, 617  
--project-id, 617  
--sudo, 617  
-h, 617

gitlab-project-branch-list command line option

--get-all, 617  
--help, 617  
--page, 617  
--per-page, 617  
--project-id, 617  
--sudo, 617  
-h, 617

gitlab-project-ci-lint command line option

--help, 618  
-h, 618

gitlab-project-ci-lint-create command line option

--content, 618  
--dry-run, 618  
--help, 618  
--include-jobs, 618  
--project-id, 618  
--ref, 618  
--sudo, 618  
-h, 618

gitlab-project-ci-lint-get command line option

--dry-run, 618  
--help, 618  
--include-jobs, 618  
--project-id, 618  
--ref, 618  
--sudo, 618  
-h, 618

gitlab-project-ci-lint-validate command line option

--content, 619  
--dry-run, 619  
--help, 619  
--include-jobs, 619  
--project-id, 619  
--ref, 619  
--sudo, 619  
-h, 619

gitlab-project-cluster command line option

--help, 619  
-h, 619

gitlab-project-cluster-create command line option

--domain, 619  
--enabled, 619  
--environment-scope, 620  
--help, 619  
--managed, 620  
--name, 619  
--platform-kubernetes-attributes, 619  
--project-id, 619  
--sudo, 619  
-h, 619

gitlab-project-cluster-delete command line option

--help, 620  
--id, 620  
--project-id, 620  
--sudo, 620  
-h, 620

gitlab-project-cluster-get command line option

--help, 620  
--id, 620  
--project-id, 620  
--sudo, 620  
-h, 620

gitlab-project-cluster-list command line option

--get-all, 620  
--help, 620  
--page, 620  
--per-page, 620

```
--project-id, 620
--sudo, 620
-h, 620
gitlab-project-cluster-update command line
    option
--domain, 621
--environment-scope, 621
--help, 621
--id, 621
--management-project-id, 621
--name, 621
--platform-kubernetes-attributes, 621
--project-id, 621
--sudo, 621
-h, 621
gitlab-project-commit command line option
--help, 621
-h, 621
gitlab-project-commit-cherry-pick command
    line option
--branch, 622
--help, 622
--id, 622
--project-id, 622
--sudo, 622
-h, 622
gitlab-project-commit-comment command line
    option
--help, 625
-h, 625
gitlab-project-commit-comment-create
    command line option
--commit-id, 625
--help, 625
--line, 625
--line-type, 625
--note, 625
--path, 625
--project-id, 625
--sudo, 625
-h, 625
gitlab-project-commit-comment-list command
    line option
--commit-id, 626
--get-all, 626
--help, 626
--page, 626
--per-page, 626
--project-id, 626
--sudo, 626
-h, 626
gitlab-project-commit-create command line
    option
--actions, 622
--author-email, 622
--author-name, 622
--branch, 622
--commit-message, 622
--help, 622
--project-id, 622
--sudo, 622
-h, 622
gitlab-project-commit-diff command line
    option
--help, 622
--id, 622
--project-id, 622
--sudo, 622
-h, 622
gitlab-project-commit-discussion command
    line option
--help, 626
-h, 626
gitlab-project-commit-discussion-create
    command line option
--body, 626
--commit-id, 626
--created-at, 626
--help, 626
--project-id, 626
--sudo, 626
-h, 626
gitlab-project-commit-discussion-get
    command line option
--commit-id, 627
--help, 627
--id, 627
--project-id, 627
--sudo, 627
-h, 627
gitlab-project-commit-discussion-list
    command line option
--commit-id, 627
--get-all, 627
--help, 627
--page, 627
--per-page, 627
--project-id, 627
--sudo, 627
-h, 627
gitlab-project-commit-discussion-note
    command line option
--help, 627
-h, 627
gitlab-project-commit-discussion-note-create
    command line option
--body, 628
--commit-id, 628
```

```

--created-at, 628
--discussion-id, 628
--help, 628
--position, 628
--project-id, 628
--sudo, 628
-h, 628
gitlab-project-commit-discussion-note-delete
    command line option
--commit-id, 628
--discussion-id, 628
--help, 628
--id, 628
--project-id, 628
--sudo, 628
-h, 628
gitlab-project-commit-discussion-note-get
    command line option
--commit-id, 629
--discussion-id, 629
--help, 629
--id, 629
--project-id, 629
--sudo, 629
-h, 629
gitlab-project-commit-discussion-note-update
    command line option
--body, 629
--commit-id, 629
--discussion-id, 629
--help, 629
--id, 629
--project-id, 629
--sudo, 629
-h, 629
gitlab-project-commit-get command line
    option
--help, 623
--id, 623
--project-id, 623
--sudo, 623
-h, 623
gitlab-project-commit-list command line
    option
--all, 623
--first-parent, 623
--get-all, 623
--help, 623
--order, 623
--page, 623
--path, 623
--per-page, 623
--project-id, 623
--ref-name, 623
--since, 623
--sudo, 623
--trailers, 623
--until, 623
--with-stats, 623
-h, 623
gitlab-project-commit-merge-requests
    command line option
--help, 624
--id, 624
--project-id, 624
--sudo, 624
-h, 624
gitlab-project-commit-refs command line
    option
--help, 624
--id, 624
--project-id, 624
--sudo, 624
--type, 624
-h, 624
gitlab-project-commit-revert command line
    option
--branch, 624
--help, 624
--id, 624
--project-id, 624
--sudo, 624
-h, 624
gitlab-project-commit-signature command
    line option
--help, 625
--id, 625
--project-id, 625
--sudo, 625
-h, 625
gitlab-project-commit-status command line
    option
--help, 629
-h, 629
gitlab-project-commit-status-create command
    line option
--commit-id, 630
--context, 630
--coverage, 630
--description, 630
--help, 630
--name, 630
--project-id, 630
--ref, 630
--state, 630
--sudo, 630
--target-url, 630
-h, 630

```

gitlab-project-commit-status-list command  
line option

--commit-id, 630  
--get-all, 630  
--help, 630  
--page, 630  
--per-page, 630  
--project-id, 630  
--sudo, 630  
-h, 630

gitlab-project-create command line option

--allow-merge-on-skipped-pipeline, 587  
--analytics-access-level, 587  
--approvals-before-merge, 587  
--auto-cancel-pending-pipelines, 587  
--auto-devops-deploy-strategy, 587  
--auto-devops-enabled, 587  
--autoclose-referenced-issues, 587  
--avatar, 587  
--build-coverage-regex, 588  
--build-git-strategy, 588  
--build-timeout, 588  
--builds-access-level, 588  
--ci-config-path, 588  
--container-expiration-policy-attributes, 588  
--container-registry-access-level, 588  
--container-registry-enabled, 588  
--default-branch, 588  
--description, 588  
--emails-disabled, 588  
--environments-access-level, 589  
--external-authorization-classification-label, 588  
--feature-flags-access-level, 589  
--forking-access-level, 588  
--group-with-project-templates-id, 588  
--help, 587  
--import-url, 588  
--infrastructure-access-level, 589  
--initialize-with-readme, 588  
--issues-access-level, 588  
--issues-enabled, 588  
--jobs-enabled, 588  
--lfs-enabled, 588  
--merge-method, 588  
--merge-pipelines-enabled, 588  
--merge-requests-access-level, 588  
--merge-requests-enabled, 588  
--mirror, 588  
--mirror-trigger-builds, 588  
--monitor-access-level, 589  
--name, 587  
--namespace-id, 588

--only-allow-merge-if-all-discussions-are-resolved, 588  
--only-allow-merge-if-all-status-checks-passed, 587  
--only-allow-merge-if-pipeline-succeeds, 588  
--operations-access-level, 588  
--packages-enabled, 588  
--pages-access-level, 589  
--path, 587  
--printing-merge-request-link-enabled, 589  
--public-builds, 589  
--releases-access-level, 589  
--remove-source-branch-after-merge, 589  
--repository-access-level, 589  
--repository-storage, 589  
--request-access-enabled, 589  
--requirements-access-level, 589  
--resolve-outdated-diff-discussions, 589  
--security-and-compliance-access-level, 589  
--shared-runners-enabled, 589  
--show-default-award-emojis, 589  
--snippets-access-level, 589  
--snippets-enabled, 589  
--squash-option, 589  
--sudo, 587  
--tag-list, 589  
--template-name, 589  
--template-project-id, 589  
--topics, 589  
--use-custom-template, 589  
--visibility, 589  
--wiki-access-level, 589  
--wiki-enabled, 589  
-h, 587

gitlab-project-create-fork-relation command  
line option

--forked-from-id, 590  
--help, 590  
--id, 590  
-h, 590

gitlab-project-custom-attribute command  
line option

--help, 631  
-h, 631

gitlab-project-custom-attribute-delete  
command line option

--help, 631  
--key, 631  
--project-id, 631  
--sudo, 631  
-h, 631



---

```

gitlab-project-custom-attribute-get command
    line option
    --help, 631
    --key, 631
    --project-id, 631
    --sudo, 631
    -h, 631
gitlab-project-custom-attribute-list
    command line option
    --get-all, 631
    --help, 631
    --page, 631
    --per-page, 631
    --project-id, 631
    --sudo, 631
    -h, 631
gitlab-project-delete command line option
    --help, 590
    --id, 590
    --sudo, 590
    -h, 590
gitlab-project-delete-fork-relation command
    line option
    --help, 590
    --id, 590
    -h, 590
gitlab-project-delete-merged-branches
    command line option
    --help, 590
    --id, 590
    -h, 590
gitlab-project-deploy-token command line
    option
    --help, 632
    -h, 632
gitlab-project-deploy-token-create command
    line option
    --expires-at, 632
    --help, 632
    --name, 632
    --project-id, 632
    --scopes, 632
    --sudo, 632
    --username, 632
    -h, 632
gitlab-project-deploy-token-delete command
    line option
    --help, 632
    --id, 632
    --project-id, 632
    --sudo, 632
    -h, 632
gitlab-project-deploy-token-get command
    line option
    --help, 632
    --id, 633
    --project-id, 632
    --sudo, 632
    -h, 632
gitlab-project-deploy-token-list command
    line option
    --get-all, 633
    --help, 633
    --page, 633
    --per-page, 633
    --project-id, 633
    --scopes, 633
    --sudo, 633
    -h, 633
gitlab-project-deployment command line
    option
    --help, 633
    -h, 633
gitlab-project-deployment-approval command
    line option
    --comment, 633
    --help, 633
    --id, 633
    --project-id, 633
    --represented-as, 633
    --status, 633
    --sudo, 633
    -h, 633
gitlab-project-deployment-create command
    line option
    --environment, 634
    --help, 634
    --project-id, 634
    --ref, 634
    --sha, 634
    --status, 634
    --sudo, 634
    --tag, 634
    -h, 634
gitlab-project-deployment-get command line
    option
    --help, 634
    --id, 634
    --project-id, 634
    --sudo, 634
    -h, 634
gitlab-project-deployment-list command line
    option
    --environment, 635
    --get-all, 635
    --help, 634
    --order-by, 635
    --page, 635

```

```
--per-page, 635
--project-id, 634
--sort, 635
--status, 635
--sudo, 634
--updated-after, 635
--updated-before, 635
-h, 634
gitlab-project-deployment-merge-request
    command line option
    --help, 635
    -h, 635
gitlab-project-deployment-merge-request-list
    command line option
    --approved-by-ids, 637
    --approver-ids, 637
    --assignee-id, 637
    --author-id, 637
    --author-username, 637
    --created-after, 636
    --created-before, 637
    --deployed-after, 637
    --deployed-before, 637
    --deployment-id, 636
    --environment, 637
    --get-all, 637
    --help, 636
    --in, 637
    --labels, 636
    --milestone, 636
    --my-reaction-emoji, 637
    --not, 637
    --order-by, 636
    --page, 637
    --per-page, 637
    --project-id, 636
    --reviewer-id, 637
    --reviewer-username, 637
    --scope, 637
    --search, 637
    --sort, 636
    --source-branch, 637
    --state, 636
    --sudo, 636
    --target-branch, 637
    --updated-after, 637
    --updated-before, 637
    --view, 636
    --wip, 637
    --with-labels-details, 636
    --with-merge-status-recheck, 636
    -h, 636
gitlab-project-deployment-update command
    line option
    --help, 635
    --id, 635
    --project-id, 635
    --sudo, 635
    -h, 635
gitlab-project-environment command line
    option
    --help, 638
    -h, 638
gitlab-project-environment-create command
    line option
    --external-url, 638
    --help, 638
    --name, 638
    --project-id, 638
    --sudo, 638
    -h, 638
gitlab-project-environment-delete command
    line option
    --help, 638
    --id, 638
    --project-id, 638
    --sudo, 638
    -h, 638
gitlab-project-environment-get command line
    option
    --help, 638
    --id, 638
    --project-id, 638
    --sudo, 638
    -h, 638
gitlab-project-environment-list command
    line option
    --get-all, 639
    --help, 639
    --name, 639
    --page, 639
    --per-page, 639
    --project-id, 639
    --search, 639
    --states, 639
    --sudo, 639
    -h, 639
gitlab-project-environment-stop command
    line option
    --help, 639
    --id, 639
    --project-id, 639
    --sudo, 639
    -h, 639
gitlab-project-environment-update command
    line option
    --external-url, 640
    --help, 639
```

```

--id, 639
--name, 640
--project-id, 639
--sudo, 639
-h, 639
gitlab-project-event command line option
--help, 640
-h, 640
gitlab-project-event-list command line
  option
--action, 640
--after, 640
--before, 640
--get-all, 640
--help, 640
--page, 640
--per-page, 640
--project-id, 640
--scope, 640
--sort, 640
--sudo, 640
--target-type, 640
-h, 640
gitlab-project-export command line option
--help, 641
-h, 641
gitlab-project-export-create command line
  option
--description, 641
--help, 641
--project-id, 641
--sudo, 641
-h, 641
gitlab-project-export-download command line
  option
--help, 641
--project-id, 641
--sudo, 641
-h, 641
gitlab-project-export-get command line
  option
--help, 641
--project-id, 641
--sudo, 641
-h, 641
gitlab-project-file command line option
--help, 642
-h, 642
gitlab-project-file-blame command line
  option
--file-path, 642
--help, 642
--project-id, 642
--ref, 642
--sudo, 642
-h, 642
gitlab-project-file-create command line
  option
--author-email, 642
--author-name, 642
--branch, 642
--commit-message, 642
--content, 642
--encoding, 642
--file-path, 642
--help, 642
--project-id, 642
--sudo, 642
-h, 642
gitlab-project-file-delete command line
  option
--branch, 643
--commit-message, 643
--file-path, 643
--help, 643
--project-id, 643
--sudo, 643
-h, 643
gitlab-project-file-get command line option
--file-path, 643
--help, 643
--project-id, 643
--ref, 643
--sudo, 643
-h, 643
gitlab-project-file-raw command line option
--file-path, 643
--help, 643
--project-id, 643
--ref, 643
--sudo, 643
-h, 643
gitlab-project-file-update command line
  option
--author-email, 644
--author-name, 644
--branch, 644
--commit-message, 644
--content, 644
--encoding, 644
--file-path, 644
--help, 644
--project-id, 644
--sudo, 644
-h, 644
gitlab-project-fork command line option
--help, 644
-h, 644

```

gitlab-project-fork-create command line  
option

--help, 644  
--namespace, 644  
--project-id, 644  
--sudo, 644  
-h, 644

gitlab-project-fork-list command line  
option

--archived, 645  
--get-all, 645  
--help, 645  
--membership, 645  
--order-by, 645  
--owned, 645  
--page, 645  
--per-page, 645  
--project-id, 645  
--search, 645  
--simple, 645  
--sort, 645  
--starred, 645  
--statistics, 645  
--sudo, 645  
--visibility, 645  
--with-custom-attributes, 645  
--with-issues-enabled, 645  
--with-merge-requests-enabled, 645  
-h, 645

gitlab-project-get command line option

--help, 590  
--id, 590  
--sudo, 590  
-h, 590

gitlab-project-hook command line option

--help, 646  
-h, 646

gitlab-project-hook-create command line  
option

--confidential-issues-events, 646  
--enable-ssl-verification, 646  
--help, 646  
--issues-events, 646  
--job-events, 646  
--merge-requests-events, 646  
--note-events, 646  
--pipeline-events, 646  
--project-id, 646  
--push-events, 646  
--sudo, 646  
--tag-push-events, 646  
--token, 646  
--url, 646  
--wiki-page-events, 646

-h, 646

gitlab-project-hook-delete command line  
option

--help, 647  
--id, 647  
--project-id, 647  
--sudo, 647  
-h, 647

gitlab-project-hook-get command line option

--help, 647  
--id, 647  
--project-id, 647  
--sudo, 647  
-h, 647

gitlab-project-hook-list command line  
option

--get-all, 647  
--help, 647  
--page, 647  
--per-page, 647  
--project-id, 647  
--sudo, 647  
-h, 647

gitlab-project-hook-update command line  
option

--confidential-issues-events, 648  
--enable-ssl-verification, 648  
--help, 648  
--id, 648  
--issues-events, 648  
--job-events, 648  
--merge-requests-events, 648  
--note-events, 648  
--pipeline-events, 648  
--project-id, 648  
--push-events, 648  
--sudo, 648  
--tag-push-events, 648  
--token, 648  
--url, 648  
--wiki-events, 648  
-h, 648

gitlab-project-housekeeping command line  
option

--help, 591  
--id, 591  
-h, 591

gitlab-project-import command line option

--help, 649  
-h, 649

gitlab-project-import-get command line  
option

--help, 649  
--project-id, 649

```

--sudo, 649
-h, 649
gitlab-project-integration command line
    option
--help, 649
-h, 649
gitlab-project-integration-available
    command line option
--help, 649
--project-id, 649
--slug, 649
--sudo, 649
-h, 649
gitlab-project-integration-delete command
    line option
--help, 649
--project-id, 649
--slug, 649
--sudo, 649
-h, 649
gitlab-project-integration-get command line
    option
--help, 650
--project-id, 650
--slug, 650
--sudo, 650
-h, 650
gitlab-project-integration-list command
    line option
--get-all, 650
--help, 650
--page, 650
--per-page, 650
--project-id, 650
--sudo, 650
-h, 650
gitlab-project-integration-update command
    line option
--help, 650
--project-id, 650
--slug, 650
--sudo, 650
-h, 650
gitlab-project-invitation command line
    option
--help, 651
-h, 651
gitlab-project-invitation-create command
    line option
--access-level, 651
--email, 651
--expires-at, 651
--help, 651
--invite-source, 651
--project-id, 651
--sudo, 651
--tasks-project-id, 651
--tasks-to-be-done, 651
--user-id, 651
-h, 651
gitlab-project-invitation-delete command
    line option
--email, 651
--help, 651
--project-id, 651
--sudo, 651
-h, 651
gitlab-project-invitation-get command line
    option
--email, 652
--help, 652
--project-id, 652
--sudo, 652
-h, 652
gitlab-project-invitation-list command line
    option
--get-all, 652
--help, 652
--page, 652
--per-page, 652
--project-id, 652
--query, 652
--sudo, 652
-h, 652
gitlab-project-invitation-update command
    line option
--access-level, 652
--email, 652
--expires-at, 652
--help, 652
--project-id, 652
--sudo, 652
-h, 652
gitlab-project-issue command line option
--help, 653
-h, 653
gitlab-project-issue-add-spent-time command
    line option
--duration, 653
--help, 653
--iid, 653
--project-id, 653
--sudo, 653
-h, 653
gitlab-project-issue-award-emoji command
    line option
--help, 661
-h, 661

```

```
gitlab-project-issue-award-emoji-create
    command line option
    --help, 661
    --issue-iid, 661
    --name, 661
    --project-id, 661
    --sudo, 661
    -h, 661
gitlab-project-issue-award-emoji-delete
    command line option
    --help, 661
    --id, 661
    --issue-iid, 661
    --project-id, 661
    --sudo, 661
    -h, 661
gitlab-project-issue-award-emoji-get
    command line option
    --help, 662
    --id, 662
    --issue-iid, 662
    --project-id, 662
    --sudo, 662
    -h, 662
gitlab-project-issue-award-emoji-list
    command line option
    --get-all, 662
    --help, 662
    --issue-iid, 662
    --page, 662
    --per-page, 662
    --project-id, 662
    --sudo, 662
    -h, 662
gitlab-project-issue-closed-by command line
    option
    --help, 654
    --iid, 654
    --project-id, 654
    --sudo, 654
    -h, 654
gitlab-project-issue-create command line
    option
    --assignee-id, 654
    --assignee-ids, 654
    --confidential, 654
    --created-at, 654
    --description, 654
    --discussion-to-resolve, 654
    --due-date, 654
    --help, 654
    --labels, 654
    --merge-request-to-resolve-discussions-of,
        654
    --milestone-id, 654
    --project-id, 654
    --sudo, 654
    --title, 654
    -h, 654
gitlab-project-issue-delete command line
    option
    --help, 655
    --iid, 655
    --project-id, 655
    --sudo, 655
    -h, 655
gitlab-project-issue-discussion command
    line option
    --help, 662
    -h, 662
gitlab-project-issue-discussion-create
    command line option
    --body, 663
    --created-at, 663
    --help, 663
    --issue-iid, 663
    --project-id, 663
    --sudo, 663
    -h, 663
gitlab-project-issue-discussion-get command
    line option
    --help, 663
    --id, 663
    --issue-iid, 663
    --project-id, 663
    --sudo, 663
    -h, 663
gitlab-project-issue-discussion-list
    command line option
    --get-all, 664
    --help, 663
    --issue-iid, 663
    --page, 663
    --per-page, 663
    --project-id, 663
    --sudo, 663
    -h, 663
gitlab-project-issue-discussion-note
    command line option
    --help, 664
    -h, 664
gitlab-project-issue-discussion-note-create
    command line option
    --body, 664
    --created-at, 664
    --discussion-id, 664
    --help, 664
    --issue-iid, 664
```

```

--project-id, 664
--sudo, 664
-h, 664
gitlab-project-issue-discussion-note-delete
    command line option
--discussion-id, 665
--help, 664
--id, 665
--issue-iid, 665
--project-id, 665
--sudo, 664
-h, 664
gitlab-project-issue-discussion-note-get
    command line option
--discussion-id, 665
--help, 665
--id, 665
--issue-iid, 665
--project-id, 665
--sudo, 665
-h, 665
gitlab-project-issue-discussion-note-update
    command line option
--body, 665
--discussion-id, 665
--help, 665
--id, 665
--issue-iid, 665
--project-id, 665
--sudo, 665
-h, 665
gitlab-project-issue-get command line
    option
--help, 655
--iid, 655
--project-id, 655
--sudo, 655
-h, 655
gitlab-project-issue-link command line
    option
--help, 666
-h, 666
gitlab-project-issue-link-create command
    line option
--help, 666
--issue-iid, 666
--project-id, 666
--sudo, 666
--target-issue-iid, 666
--target-project-id, 666
-h, 666
gitlab-project-issue-link-delete command
    line option
--help, 666
--issue-iid, 666
--issue-link-id, 666
--project-id, 666
--sudo, 666
-h, 666
gitlab-project-issue-link-list command line
    option
--get-all, 667
--help, 667
--issue-iid, 667
--page, 667
--per-page, 667
--project-id, 667
--sudo, 667
-h, 667
gitlab-project-issue-list command line
    option
--assignee-id, 656
--author-id, 656
--created-after, 656
--created-before, 656
--get-all, 656
--help, 655
--iids, 655
--iteration-id, 656
--labels, 656
--milestone, 656
--my-reaction-emoji, 656
--order-by, 656
--page, 656
--per-page, 656
--project-id, 655
--scope, 656
--search, 656
--sort, 656
--state, 656
--sudo, 655
--updated-after, 656
--updated-before, 656
-h, 655
gitlab-project-issue-move command line
    option
--help, 656
--iid, 656
--project-id, 656
--sudo, 656
--to-project-id, 656
-h, 656
gitlab-project-issue-note command line
    option
--help, 667
-h, 667
gitlab-project-issue-note-award-emoji
    command line option

```

```
--help, 669
-h, 669
gitlab-project-issue-note-award-emoji-create
    command line option
    --help, 669
    --issue-iid, 669
    --name, 669
    --note-id, 669
    --project-id, 669
    --sudo, 669
    -h, 669
gitlab-project-issue-note-award-emoji-delete
    command line option
    --help, 670
    --id, 670
    --issue-iid, 670
    --note-id, 670
    --project-id, 670
    --sudo, 670
    -h, 670
gitlab-project-issue-note-award-emoji-get
    command line option
    --help, 670
    --id, 670
    --issue-iid, 670
    --note-id, 670
    --project-id, 670
    --sudo, 670
    -h, 670
gitlab-project-issue-note-award-emoji-list
    command line option
    --get-all, 671
    --help, 670
    --issue-iid, 671
    --note-id, 671
    --page, 671
    --per-page, 671
    --project-id, 671
    --sudo, 670
    -h, 670
gitlab-project-issue-note-create
    command line option
    --body, 667
    --created-at, 667
    --help, 667
    --issue-iid, 667
    --project-id, 667
    --sudo, 667
    -h, 667
gitlab-project-issue-note-delete
    command line option
    --help, 668
    --id, 668
    --issue-iid, 668
    --project-id, 668
    --sudo, 668
    -h, 668
gitlab-project-issue-note-get
    command line option
    --help, 668
    --id, 668
    --issue-iid, 668
    --project-id, 668
    --sudo, 668
    -h, 668
gitlab-project-issue-note-list
    command line option
    --get-all, 668
    --help, 668
    --issue-iid, 668
    --page, 668
    --per-page, 668
    --project-id, 668
    --sudo, 668
    -h, 668
gitlab-project-issue-note-update
    command line option
    --body, 669
    --help, 669
    --id, 669
    --issue-iid, 669
    --project-id, 669
    --sudo, 669
    -h, 669
gitlab-project-issue-participants
    command line option
    --help, 657
    --iid, 657
    --project-id, 657
    --sudo, 657
    -h, 657
gitlab-project-issue-related-merge-requests
    command line option
    --help, 657
    --iid, 657
    --project-id, 657
    --sudo, 657
    -h, 657
gitlab-project-issue-reorder
    command line option
    --help, 657
    --iid, 657
    --move-after-id, 657
    --move-before-id, 657
    --project-id, 657
    --sudo, 657
    -h, 657
```



```

gitlab-project-issue-reset-spent-time
    command line option
    --help, 658
    --iid, 658
    --project-id, 658
    --sudo, 658
    -h, 658
gitlab-project-issue-reset-time-estimate
    command line option
    --help, 658
    --iid, 658
    --project-id, 658
    --sudo, 658
    -h, 658
gitlab-project-issue-resource-iteration-event
    command line option
    --help, 671
    -h, 671
gitlab-project-issue-resource-iteration-event-get
    command line option
    --help, 671
    --id, 671
    --issue-iid, 671
    --project-id, 671
    --sudo, 671
    -h, 671
gitlab-project-issue-resource-iteration-event-list
    command line option
    --get-all, 672
    --help, 671
    --issue-iid, 672
    --page, 672
    --per-page, 672
    --project-id, 672
    --sudo, 672
    -h, 671
gitlab-project-issue-resource-label-event
    command line option
    --help, 672
    -h, 672
gitlab-project-issue-resource-label-event-get
    command line option
    --help, 672
    --id, 672
    --issue-iid, 672
    --project-id, 672
    --sudo, 672
    -h, 672
gitlab-project-issue-resource-label-event-list
    command line option
    --get-all, 673
    --help, 672
    --issue-iid, 673
    --page, 673
    --per-page, 673
    --project-id, 673
    --sudo, 673
    -h, 673
gitlab-project-issue-resource-milestone-event
    command line option
    --help, 673
    -h, 673
gitlab-project-issue-resource-milestone-event-get
    command line option
    --help, 673
    --id, 673
    --issue-iid, 673
    --project-id, 673
    --sudo, 673
    -h, 673
gitlab-project-issue-resource-milestone-event-list
    command line option
    --get-all, 674
    --help, 673
    --issue-iid, 674
    --page, 674
    --per-page, 674
    --project-id, 674
    --sudo, 673
    -h, 673
gitlab-project-issue-resource-state-event
    command line option
    --help, 674
    -h, 674
gitlab-project-issue-resource-state-event-get
    command line option
    --help, 674
    --id, 674
    --issue-iid, 674
    --project-id, 674
    --sudo, 674
    -h, 674
gitlab-project-issue-resource-state-event-list
    command line option
    --get-all, 675
    --help, 674
    --issue-iid, 675
    --page, 675
    --per-page, 675
    --project-id, 674
    --sudo, 674
    -h, 674
gitlab-project-issue-resource-weight-event
    command line option
    --help, 675
    -h, 675
gitlab-project-issue-resource-weight-event-get
    command line option

```

```
--help, 675
--id, 675
--issue-iid, 675
--project-id, 675
--sudo, 675
-h, 675
gitlab-project-issue-resource-weight-event-list
    command line option
--get-all, 676
--help, 675
--issue-iid, 675
--page, 676
--per-page, 676
--project-id, 675
--sudo, 675
-h, 675
gitlab-project-issue-subscribe command line
    option
--help, 658
--iid, 658
--project-id, 658
--sudo, 658
-h, 658
gitlab-project-issue-time-estimate command
    line option
--duration, 659
--help, 658
--iid, 659
--project-id, 658
--sudo, 658
-h, 658
gitlab-project-issue-time-stats command
    line option
--help, 659
--iid, 659
--project-id, 659
--sudo, 659
-h, 659
gitlab-project-issue-todo command line
    option
--help, 659
--iid, 659
--project-id, 659
--sudo, 659
-h, 659
gitlab-project-issue-unsubscribe command
    line option
--help, 659
--iid, 659
--project-id, 659
--sudo, 659
-h, 659
gitlab-project-issue-update command line
    option
--help, 675
--id, 675
--issue-iid, 675
--project-id, 675
--sudo, 675
-h, 675
--assignee-id, 660
--assignee-ids, 660
--confidential, 660
--description, 660
--discussion-locked, 660
--due-date, 660
--help, 660
--iid, 660
--labels, 660
--milestone-id, 660
--project-id, 660
--state-event, 660
--sudo, 660
--title, 660
--updated-at, 660
-h, 660
gitlab-project-issue-user-agent-detail
    command line option
--help, 660
--iid, 661
--project-id, 660
--sudo, 660
-h, 660
gitlab-project-issues-statistics command
    line option
--help, 676
-h, 676
gitlab-project-issues-statistics-get
    command line option
--help, 676
--project-id, 676
--sudo, 676
-h, 676
gitlab-project-job command line option
--help, 676
-h, 676
gitlab-project-job-artifact command line
    option
--help, 677
--id, 677
--project-id, 677
--sudo, 677
-h, 677
gitlab-project-job-artifacts command line
    option
--help, 677
--id, 677
--project-id, 677
--sudo, 677
-h, 677
gitlab-project-job-cancel command line
    option
--help, 677
--id, 677
```

```

--project-id, 677
--sudo, 677
-h, 677
gitlab-project-job-delete-artifacts command
    line option
--help, 677
--id, 678
--project-id, 677
--sudo, 677
-h, 677
gitlab-project-job-erase command line
    option
--help, 678
--id, 678
--project-id, 678
--sudo, 678
-h, 678
gitlab-project-job-get command line option
--help, 678
--id, 678
--project-id, 678
--sudo, 678
-h, 678
gitlab-project-job-keep-artifacts command
    line option
--help, 678
--id, 678
--project-id, 678
--sudo, 678
-h, 678
gitlab-project-job-list command line option
--get-all, 679
--help, 679
--page, 679
--per-page, 679
--project-id, 679
--scope, 679
--sudo, 679
-h, 679
gitlab-project-job-play command line option
--help, 679
--id, 679
--project-id, 679
--sudo, 679
-h, 679
gitlab-project-job-retry command line
    option
--help, 679
--id, 679
--project-id, 679
--sudo, 679
-h, 679
gitlab-project-job-token-scope command line
    option
--help, 680
-h, 680
gitlab-project-job-token-scope-get command
    line option
--help, 680
--project-id, 680
--sudo, 680
-h, 680
gitlab-project-job-token-scope-update
    command line option
--help, 680
--project-id, 680
--sudo, 680
-h, 680
gitlab-project-job-trace command line
    option
--help, 680
--id, 680
--project-id, 680
--sudo, 680
-h, 680
gitlab-project-key command line option
--help, 681
-h, 681
gitlab-project-key-create command line
    option
--can-push, 681
--help, 681
--key, 681
--project-id, 681
--sudo, 681
--title, 681
-h, 681
gitlab-project-key-delete command line
    option
--help, 681
--id, 681
--project-id, 681
--sudo, 681
-h, 681
gitlab-project-key-enable command line
    option
--help, 681
--id, 681
--key-id, 681
--project-id, 681
--sudo, 681
-h, 681
gitlab-project-key-get command line option
--help, 682
--id, 682
--project-id, 682
--sudo, 682
-h, 682

```

gitlab-project-key-list command line option

- `--get-all`, 682
- `--help`, 682
- `--page`, 682
- `--per-page`, 682
- `--project-id`, 682
- `--sudo`, 682
- `-h`, 682

gitlab-project-key-update command line option

- `--can-push`, 682
- `--help`, 682
- `--id`, 682
- `--project-id`, 682
- `--sudo`, 682
- `--title`, 682
- `-h`, 682

gitlab-project-label command line option

- `--help`, 683
- `-h`, 683

gitlab-project-label-create command line option

- `--color`, 683
- `--description`, 683
- `--help`, 683
- `--name`, 683
- `--priority`, 683
- `--project-id`, 683
- `--sudo`, 683
- `-h`, 683

gitlab-project-label-delete command line option

- `--help`, 683
- `--name`, 683
- `--project-id`, 683
- `--sudo`, 683
- `-h`, 683

gitlab-project-label-get command line option

- `--help`, 684
- `--name`, 684
- `--project-id`, 684
- `--sudo`, 684
- `-h`, 684

gitlab-project-label-list command line option

- `--get-all`, 684
- `--help`, 684
- `--page`, 684
- `--per-page`, 684
- `--project-id`, 684
- `--sudo`, 684
- `-h`, 684

gitlab-project-label-subscribe command line option

- `--help`, 684
- `--name`, 684
- `--project-id`, 684
- `--sudo`, 684
- `-h`, 684

gitlab-project-label-unsubscribe command line option

- `--help`, 685
- `--name`, 685
- `--project-id`, 685
- `--sudo`, 685
- `-h`, 685

gitlab-project-label-update command line option

- `--color`, 685
- `--description`, 685
- `--help`, 685
- `--name`, 685
- `--new-name`, 685
- `--priority`, 685
- `--project-id`, 685
- `--sudo`, 685
- `-h`, 685

gitlab-project-languages command line option

- `--help`, 591
- `--id`, 591
- `-h`, 591

gitlab-project-list command line option

- `--archived`, 591
- `--get-all`, 592
- `--help`, 591
- `--id-after`, 591
- `--id-before`, 591
- `--last-activity-after`, 591
- `--last-activity-before`, 591
- `--membership`, 592
- `--min-access-level`, 592
- `--order-by`, 592
- `--owned`, 592
- `--page`, 592
- `--per-page`, 592
- `--repository-checksum-failed`, 592
- `--repository-storage`, 592
- `--search`, 592
- `--search-namespaces`, 592
- `--simple`, 592
- `--sort`, 592
- `--starred`, 592
- `--statistics`, 592
- `--sudo`, 591
- `--topic`, 592

```

--visibility, 592
--wiki-checksum-failed, 592
--with-custom-attributes, 592
--with-issues-enabled, 592
--with-merge-requests-enabled, 592
--with-programming-language, 592
-h, 591
gitlab-project-member command line option
--help, 685
-h, 685
gitlab-project-member-all command line
    option
--help, 687
-h, 687
gitlab-project-member-all-get command line
    option
--help, 688
--id, 688
--project-id, 688
--sudo, 688
-h, 688
gitlab-project-member-all-list command line
    option
--get-all, 688
--help, 688
--page, 688
--per-page, 688
--project-id, 688
--sudo, 688
-h, 688
gitlab-project-member-create command line
    option
--access-level, 686
--expires-at, 686
--help, 686
--project-id, 686
--sudo, 686
--tasks-to-be-done, 686
--user-id, 686
-h, 686
gitlab-project-member-delete command line
    option
--help, 686
--id, 686
--project-id, 686
--sudo, 686
-h, 686
gitlab-project-member-get command line
    option
--help, 686
--id, 686
--project-id, 686
--sudo, 686
-h, 686
gitlab-project-member-list command line
    option
--get-all, 687
--help, 687
--page, 687
--per-page, 687
--project-id, 687
--sudo, 687
-h, 687
gitlab-project-member-update command line
    option
--access-level, 687
--expires-at, 687
--help, 687
--id, 687
--project-id, 687
--sudo, 687
-h, 687
gitlab-project-merge-request command line
    option
--help, 689
-h, 689
gitlab-project-merge-request-add-spent-time
    command line option
--duration, 689
--help, 689
--iid, 689
--project-id, 689
--sudo, 689
-h, 689
gitlab-project-merge-request-approval
    command line option
--help, 699
-h, 699
gitlab-project-merge-request-approval-get
    command line option
--help, 699
--mr-iid, 699
--project-id, 699
--sudo, 699
-h, 699
gitlab-project-merge-request-approval-rule
    command line option
--help, 700
-h, 700
gitlab-project-merge-request-approval-rule-create
    command line option
--approval-project-rule-id, 701
--approvals-required, 701
--group-ids, 701
--help, 700
--id, 701
--merge-request-iid, 701
--mr-iid, 701

```

```
--name, 701
--project-id, 700
--sudo, 700
--user-ids, 701
-h, 700
gitlab-project-merge-request-approval-rule-delete
    command line option
--help, 701
--id, 701
--mr-iid, 701
--project-id, 701
--sudo, 701
-h, 701
gitlab-project-merge-request-approval-rule-get
    command line option
--help, 701
--id, 701
--mr-iid, 701
--project-id, 701
--sudo, 701
-h, 701
gitlab-project-merge-request-approval-rule-list
    command line option
--get-all, 702
--help, 702
--mr-iid, 702
--page, 702
--per-page, 702
--project-id, 702
--sudo, 702
-h, 702
gitlab-project-merge-request-approval-rule-update
    command line option
--approval-rule-id, 702
--approvals-required, 703
--group-ids, 703
--help, 702
--id, 702
--merge-request-iid, 702
--mr-iid, 702
--name, 702
--project-id, 702
--sudo, 702
--user-ids, 703
-h, 702
gitlab-project-merge-request-approval-state
    command line option
--help, 703
-h, 703
gitlab-project-merge-request-approval-state-get
    command line option
--help, 703
--id, 703
--mr-iid, 703
--project-id, 703
--sudo, 703
-h, 703
gitlab-project-merge-request-approval-update
    command line option
--approvals-required, 700
--help, 700
--mr-iid, 700
--project-id, 700
--sudo, 700
-h, 700
gitlab-project-merge-request-approve
    command line option
--help, 689
--iid, 689
--project-id, 689
--sha, 689
--sudo, 689
-h, 689
gitlab-project-merge-request-award-emoji
    command line option
--help, 703
-h, 703
gitlab-project-merge-request-award-emoji-create
    command line option
--help, 704
--mr-iid, 704
--name, 704
--project-id, 704
--sudo, 704
-h, 704
gitlab-project-merge-request-award-emoji-delete
    command line option
--help, 704
--id, 704
--mr-iid, 704
--project-id, 704
--sudo, 704
-h, 704
gitlab-project-merge-request-award-emoji-get
    command line option
--help, 704
--id, 704
--mr-iid, 704
--project-id, 704
--sudo, 704
-h, 704
gitlab-project-merge-request-award-emoji-list
    command line option
--get-all, 705
--help, 705
--mr-iid, 705
--page, 705
--per-page, 705
```

```

--project-id, 705
--sudo, 705
-h, 705
gitlab-project-merge-request-cancel-merge-when-pipeline-succeeds
    command line option
--help, 690
--iid, 690
--project-id, 690
--sudo, 690
-h, 690
gitlab-project-merge-request-changes
    command line option
--access-raw-diffs, 690
--help, 690
--iid, 690
--project-id, 690
--sudo, 690
-h, 690
gitlab-project-merge-request-closes-issues
    command line option
--help, 690
--iid, 690
--project-id, 690
--sudo, 690
-h, 690
gitlab-project-merge-request-commits
    command line option
--help, 691
--iid, 691
--project-id, 691
--sudo, 691
-h, 691
gitlab-project-merge-request-create command
    line option
--allow-collaboration, 691
--allow-maintainer-to-push, 691
--approvals-before-merge, 691
--assignee-id, 691
--assignee-ids, 691
--description, 692
--help, 691
--labels, 692
--milestone-id, 692
--project-id, 691
--remove-source-branch, 692
--reviewer-ids, 692
--source-branch, 691
--squash, 692
--sudo, 691
--target-branch, 691
--target-project-id, 692
--title, 691
-h, 691
gitlab-project-merge-request-delete command
    line option
--help, 692
--project-id, 692
--sudo, 692
-h, 692
gitlab-project-merge-request-diff command
    line option
--help, 705
-h, 705
gitlab-project-merge-request-diff-get
    command line option
--help, 705
--id, 705
--mr-iid, 705
--project-id, 705
--sudo, 705
-h, 705
gitlab-project-merge-request-diff-list
    command line option
--get-all, 706
--help, 706
--mr-iid, 706
--page, 706
--per-page, 706
--project-id, 706
--sudo, 706
-h, 706
gitlab-project-merge-request-discussion
    command line option
--help, 706
-h, 706
gitlab-project-merge-request-discussion-create
    command line option
--body, 706
--created-at, 707
--help, 706
--mr-iid, 706
--position, 707
--project-id, 706
--sudo, 706
-h, 706
gitlab-project-merge-request-discussion-get
    command line option
--help, 707
--id, 707
--mr-iid, 707
--project-id, 707
--sudo, 707
-h, 707
gitlab-project-merge-request-discussion-list
    command line option
--get-all, 707

```

```
--help, 707
--mr-iid, 707
--page, 707
--per-page, 707
--project-id, 707
--sudo, 707
-h, 707
gitlab-project-merge-request-discussion-note
    command line option
--help, 708
-h, 708
gitlab-project-merge-request-discussion-note-create
    command line option
--body, 708
--created-at, 709
--discussion-id, 708
--help, 708
--mr-iid, 708
--project-id, 708
--sudo, 708
-h, 708
gitlab-project-merge-request-discussion-note-delete
    command line option
    command line option
--discussion-id, 709
--help, 709
--id, 709
--mr-iid, 709
--project-id, 709
--sudo, 709
-h, 709
gitlab-project-merge-request-discussion-note-get
    command line option
--discussion-id, 709
--help, 709
--id, 709
--mr-iid, 709
--project-id, 709
--sudo, 709
-h, 709
gitlab-project-merge-request-discussion-note-update
    command line option
--body, 710
--discussion-id, 710
--help, 710
--id, 710
--mr-iid, 710
--project-id, 710
--sudo, 710
-h, 710
gitlab-project-merge-request-discussion-update
    command line option
--help, 708
--id, 708
--mr-iid, 708
--project-id, 708
--resolved, 708
--sudo, 708
-h, 708
gitlab-project-merge-request-draft-note
    command line option
--help, 710
-h, 710
gitlab-project-merge-request-draft-note-create
    command line option
--commit-id, 711
--help, 710
--in-reply-to-discussion-id, 711
--mr-iid, 711
--note, 711
--position, 711
--project-id, 711
--resolve-discussion, 711
--sudo, 711
-h, 710
gitlab-project-merge-request-draft-note-delete
    command line option
--help, 711
--id, 711
--mr-iid, 711
--project-id, 711
--sudo, 711
-h, 711
gitlab-project-merge-request-draft-note-get
    command line option
--help, 711
--id, 711
--mr-iid, 711
--project-id, 711
--sudo, 711
-h, 711
gitlab-project-merge-request-draft-note-list
    command line option
--get-all, 712
--mr-iid, 712
--page, 712
--per-page, 712
--project-id, 712
--sudo, 712
-h, 712
gitlab-project-merge-request-draft-note-update
    command line option
--help, 712
--id, 712
--mr-iid, 712
--position, 712
--project-id, 712
--sudo, 712
```



```

    -h, 712
gitlab-project-merge-request-get command
    line option
    --help, 692
    --iid, 692
    --include-diverged-commits-count, 692
    --include-rebase-in-progress, 692
    --project-id, 692
    --render-html, 692
    --sudo, 692
    -h, 692
gitlab-project-merge-request-list command
    line option
    --approved-by-ids, 694
    --approver-ids, 693
    --assignee-id, 693
    --author-id, 693
    --created-after, 693
    --created-before, 693
    --get-all, 694
    --help, 693
    --iids, 693
    --labels, 693
    --milestone, 693
    --my-reaction-emoji, 694
    --order-by, 693
    --page, 694
    --per-page, 694
    --project-id, 693
    --scope, 693
    --search, 694
    --sort, 693
    --source-branch, 694
    --state, 693
    --sudo, 693
    --target-branch, 694
    --updated-after, 693
    --updated-before, 693
    --view, 693
    --wip, 694
    -h, 693
gitlab-project-merge-request-merge command
    line option
    --help, 694
    --iid, 694
    --merge-commit-message, 694
    --merge-when-pipeline-succeeds, 694
    --project-id, 694
    --should-remove-source-branch, 694
    --sudo, 694
    -h, 694
gitlab-project-merge-request-merge-ref
    command line option
    --help, 694
    --iid, 695
    --project-id, 694
    --sudo, 694
    -h, 694
gitlab-project-merge-request-note command
    line option
    --help, 712
    -h, 712
gitlab-project-merge-request-note-award-emoji
    command line option
    --help, 714
    -h, 714
gitlab-project-merge-request-note-award-emoji-create
    command line option
    --help, 715
    --mr-iid, 715
    --name, 715
    --note-id, 715
    --project-id, 715
    --sudo, 715
    -h, 715
gitlab-project-merge-request-note-award-emoji-delete
    command line option
    --help, 715
    --id, 715
    --mr-iid, 715
    --note-id, 715
    --project-id, 715
    --sudo, 715
    -h, 715
gitlab-project-merge-request-note-award-emoji-get
    command line option
    --help, 715
    --id, 716
    --mr-iid, 715
    --note-id, 715
    --project-id, 715
    --sudo, 715
    -h, 715
gitlab-project-merge-request-note-award-emoji-list
    command line option
    --get-all, 716
    --help, 716
    --mr-iid, 716
    --note-id, 716
    --page, 716
    --per-page, 716
    --project-id, 716
    --sudo, 716
    -h, 716
gitlab-project-merge-request-note-create
    command line option
    --body, 713
    --help, 713

```

```
--mr-iid, 713
--project-id, 713
--sudo, 713
-h, 713
gitlab-project-merge-request-note-delete
    command line option
--help, 713
--id, 713
--mr-iid, 713
--project-id, 713
--sudo, 713
-h, 713
gitlab-project-merge-request-note-get
    command line option
--help, 713
--id, 713
--mr-iid, 713
--project-id, 713
--sudo, 713
-h, 713
gitlab-project-merge-request-note-list
    command line option
--get-all, 714
--help, 714
--mr-iid, 714
--page, 714
--per-page, 714
--project-id, 714
--sudo, 714
-h, 714
gitlab-project-merge-request-note-update
    command line option
--body, 714
--help, 714
--id, 714
--mr-iid, 714
--project-id, 714
--sudo, 714
-h, 714
gitlab-project-merge-request-participants
    command line option
--help, 695
--iid, 695
--project-id, 695
--sudo, 695
-h, 695
gitlab-project-merge-request-pipeline
    command line option
--help, 716
-h, 716
gitlab-project-merge-request-pipeline-create
    command line option
--help, 716
--mr-iid, 716
--project-id, 716
--sudo, 716
-h, 716
gitlab-project-merge-request-pipeline-list
    command line option
--get-all, 717
--help, 717
--mr-iid, 717
--page, 717
--per-page, 717
--project-id, 717
--sudo, 717
-h, 717
gitlab-project-merge-request-rebase
    command line option
--help, 695
--iid, 695
--project-id, 695
--sudo, 695
-h, 695
gitlab-project-merge-request-reset-approvals
    command line option
--help, 695
--iid, 695
--project-id, 695
--sudo, 695
-h, 695
gitlab-project-merge-request-reset-spent-time
    command line option
--help, 696
--iid, 696
--project-id, 696
--sudo, 696
-h, 696
gitlab-project-merge-request-reset-time-estimate
    command line option
--help, 696
--iid, 696
--project-id, 696
--sudo, 696
-h, 696
gitlab-project-merge-request-resource-label-event
    command line option
--help, 717
-h, 717
gitlab-project-merge-request-resource-label-event-get
    command line option
--help, 717
--id, 717
--mr-iid, 717
--project-id, 717
--sudo, 717
-h, 717
```

```

gitlab-project-merge-request-resource-label-event-list
    command line option
    --get-all, 718
    --help, 718
    --mr-iid, 718
    --page, 718
    --per-page, 718
    --project-id, 718
    --sudo, 718
    -h, 718
gitlab-project-merge-request-resource-milestone-event-list
    command line option
    --help, 718
    -h, 718
gitlab-project-merge-request-resource-milestone-event-get
    command line option
    --help, 718
    --id, 718
    --mr-iid, 718
    --project-id, 718
    --sudo, 718
    -h, 718
gitlab-project-merge-request-resource-milestone-event-create
    command line option
    --get-all, 719
    --help, 719
    --mr-iid, 719
    --page, 719
    --per-page, 719
    --project-id, 719
    --sudo, 719
    -h, 719
gitlab-project-merge-request-resource-state-event
    command line option
    --help, 719
    -h, 719
gitlab-project-merge-request-resource-state-event-list
    command line option
    --help, 719
    --id, 719
    --mr-iid, 719
    --project-id, 719
    --sudo, 719
    -h, 719
gitlab-project-merge-request-resource-state-event-create
    command line option
    --help, 719
    --id, 719
    --mr-iid, 719
    --project-id, 719
    --sudo, 719
    -h, 719
gitlab-project-merge-request-resource-state-event-list
    command line option
    --get-all, 720
    --help, 720
    --mr-iid, 720
    --page, 720
    --per-page, 720
    --project-id, 720
    --sudo, 720
    -h, 720
gitlab-project-merge-request-reviewer-detail
    command line option
    --help, 720
    -h, 720
gitlab-project-merge-request-reviewer-detail-list
    command line option
    --get-all, 720
    --help, 720
    --mr-iid, 720
    --page, 720
    --per-page, 720
    --project-id, 720
    --sudo, 720
    -h, 720
gitlab-project-merge-request-subscribe
    command line option
    --help, 696
    --iid, 696
    --project-id, 696
    --sudo, 696
    -h, 696
gitlab-project-merge-request-time-estimate
    command line option
    --duration, 697
    --help, 697
    --iid, 697
    --project-id, 697
    --sudo, 697
    -h, 697
gitlab-project-merge-request-time-stats
    command line option
    --help, 697
    --iid, 697
    --project-id, 697
    --sudo, 697
    -h, 697
gitlab-project-merge-request-todo
    command line option
    --help, 697
    --iid, 697
    --project-id, 697
    --sudo, 697
    -h, 697
gitlab-project-merge-request-unapprove
    command line option
    --help, 698
    --iid, 698
    --project-id, 698
    --sudo, 698
    -h, 698
gitlab-project-merge-request-unsubscribe
    command line option
    --help, 698
    --iid, 698

```

```
--project-id, 698
--sudo, 698
-h, 698
gitlab-project-merge-request-update command
    line option
    --allow-maintainer-to-push, 699
    --assignee-id, 699
    --description, 699
    --discussion-locked, 699
    --help, 698
    --iid, 698
    --labels, 699
    --milestone-id, 699
    --project-id, 698
    --remove-source-branch, 699
    --reviewer-ids, 699
    --squash, 699
    --state-event, 699
    --sudo, 698
    --target-branch, 699
    --title, 699
    -h, 698
gitlab-project-merge-train command line
    option
    --help, 721
    -h, 721
gitlab-project-merge-train-list command
    line option
    --get-all, 721
    --help, 721
    --page, 721
    --per-page, 721
    --project-id, 721
    --scope, 721
    --sudo, 721
    -h, 721
gitlab-project-milestone command line
    option
    --help, 721
    -h, 721
gitlab-project-milestone-create command
    line option
    --description, 722
    --due-date, 722
    --help, 722
    --project-id, 722
    --start-date, 722
    --state-event, 722
    --sudo, 722
    --title, 722
    -h, 722
gitlab-project-milestone-delete command
    line option
    --help, 722
    --id, 722
    --project-id, 722
    --sudo, 722
    -h, 722
gitlab-project-milestone-get command line
    option
    --help, 722
    --id, 722
    --project-id, 722
    --sudo, 722
    -h, 722
gitlab-project-milestone-issues command
    line option
    --help, 723
    --id, 723
    --project-id, 723
    --sudo, 723
    -h, 723
gitlab-project-milestone-list command line
    option
    --get-all, 723
    --help, 723
    --iids, 723
    --page, 723
    --per-page, 723
    --project-id, 723
    --search, 723
    --state, 723
    --sudo, 723
    -h, 723
gitlab-project-milestone-merge-requests
    command line option
    --help, 723
    --id, 723
    --project-id, 723
    --sudo, 723
    -h, 723
gitlab-project-milestone-update command
    line option
    --description, 724
    --due-date, 724
    --help, 724
    --id, 724
    --project-id, 724
    --start-date, 724
    --state-event, 724
    --sudo, 724
    --title, 724
    -h, 724
gitlab-project-mirror-pull command line
    option
    --help, 592
    --id, 592
    -h, 592
```

```

gitlab-project-mirror-pull-details command
    line option
    --help, 593
    --id, 593
    -h, 593
gitlab-project-note command line option
    --help, 724
    -h, 724
gitlab-project-note-get command line option
    --help, 724
    --id, 724
    --project-id, 724
    --sudo, 724
    -h, 724
gitlab-project-note-list command line
    option
    --get-all, 725
    --help, 725
    --page, 725
    --per-page, 725
    --project-id, 725
    --sudo, 725
    -h, 725
gitlab-project-notification-settings
    command line option
    --help, 725
    -h, 725
gitlab-project-notification-settings-get
    command line option
    --help, 725
    --project-id, 725
    --sudo, 725
    -h, 725
gitlab-project-notification-settings-update
    command line option
    --close-issue, 726
    --close-merge-request, 726
    --help, 726
    --level, 726
    --merge-merge-request, 726
    --new-issue, 726
    --new-merge-request, 726
    --new-note, 726
    --notification-email, 726
    --project-id, 726
    --reassign-issue, 726
    --reassign-merge-request, 726
    --reopen-issue, 726
    --reopen-merge-request, 726
    --sudo, 726
    -h, 726
gitlab-project-package command line option
    --help, 726
    -h, 726
gitlab-project-package-delete command line
    option
    --help, 727
    --id, 727
    --project-id, 727
    --sudo, 727
    -h, 727
gitlab-project-package-file command line
    option
    --help, 728
    -h, 728
gitlab-project-package-file-delete command
    line option
    --help, 728
    --id, 728
    --package-id, 728
    --project-id, 728
    --sudo, 728
    -h, 728
gitlab-project-package-file-list command
    line option
    --get-all, 728
    --help, 728
    --package-id, 728
    --page, 728
    --per-page, 728
    --project-id, 728
    --sudo, 728
    -h, 728
gitlab-project-package-get command line
    option
    --help, 727
    --id, 727
    --project-id, 727
    --sudo, 727
    -h, 727
gitlab-project-package-list command line
    option
    --get-all, 727
    --help, 727
    --order-by, 727
    --package-name, 727
    --package-type, 727
    --page, 727
    --per-page, 727
    --project-id, 727
    --sort, 727
    --sudo, 727
    -h, 727
gitlab-project-package-pipeline command
    line option
    --help, 729
    -h, 729

```

```
gitlab-project-package-pipeline-list
    command line option
    --get-all, 729
    --help, 729
    --package-id, 729
    --page, 729
    --per-page, 729
    --project-id, 729
    --sudo, 729
    -h, 729
gitlab-project-pages-domain command line
    option
    --help, 729
    -h, 729
gitlab-project-pages-domain-create command
    line option
    --certificate, 730
    --domain, 729
    --help, 729
    --key, 730
    --project-id, 729
    --sudo, 729
    -h, 729
gitlab-project-pages-domain-delete command
    line option
    --domain, 730
    --help, 730
    --project-id, 730
    --sudo, 730
    -h, 730
gitlab-project-pages-domain-get command
    line option
    --domain, 730
    --help, 730
    --project-id, 730
    --sudo, 730
    -h, 730
gitlab-project-pages-domain-list command
    line option
    --get-all, 730
    --help, 730
    --page, 730
    --per-page, 730
    --project-id, 730
    --sudo, 730
    -h, 730
gitlab-project-pages-domain-update command
    line option
    --certificate, 731
    --domain, 731
    --help, 731
    --key, 731
    --project-id, 731
    --sudo, 731
    -h, 731
gitlab-project-pipeline command line option
    --help, 731
    -h, 731
gitlab-project-pipeline-bridge command line
    option
    --help, 733
    -h, 733
gitlab-project-pipeline-bridge-list command
    line option
    --get-all, 734
    --help, 733
    --page, 734
    --per-page, 734
    --pipeline-id, 734
    --project-id, 733
    --scope, 734
    --sudo, 733
    -h, 733
gitlab-project-pipeline-cancel command line
    option
    --help, 731
    --id, 731
    --project-id, 731
    --sudo, 731
    -h, 731
gitlab-project-pipeline-create command line
    option
    --help, 731
    --project-id, 731
    --ref, 731
    --sudo, 731
    -h, 731
gitlab-project-pipeline-delete command line
    option
    --help, 732
    --id, 732
    --project-id, 732
    --sudo, 732
    -h, 732
gitlab-project-pipeline-get command line
    option
    --help, 732
    --id, 732
    --project-id, 732
    --sudo, 732
    -h, 732
gitlab-project-pipeline-job command line
    option
    --help, 734
    -h, 734
gitlab-project-pipeline-job-list command
    line option
    --get-all, 734
```

```

--help, 734
--include-retried, 734
--page, 734
--per-page, 734
--pipeline-id, 734
--project-id, 734
--scope, 734
--sudo, 734
-h, 734
gitlab-project-pipeline-list command line
    option
--get-all, 733
--help, 732
--name, 733
--order-by, 733
--page, 733
--per-page, 733
--project-id, 732
--ref, 732
--scope, 732
--sha, 732
--sort, 733
--source, 732
--status, 732
--sudo, 732
--username, 733
--yaml-errors, 733
-h, 732
gitlab-project-pipeline-retry command line
    option
--help, 733
--id, 733
--project-id, 733
--sudo, 733
-h, 733
gitlab-project-pipeline-schedule command
    line option
--help, 735
-h, 735
gitlab-project-pipeline-schedule-create
    command line option
--active, 735
--cron, 735
--cron-timezone, 735
--description, 735
--help, 735
--project-id, 735
--ref, 735
--sudo, 735
-h, 735
gitlab-project-pipeline-schedule-delete
    command line option
--help, 735
--id, 735
--project-id, 735
--sudo, 735
-h, 735
gitlab-project-pipeline-schedule-get
    command line option
--help, 736
--id, 736
--project-id, 736
--sudo, 736
-h, 736
gitlab-project-pipeline-schedule-list
    command line option
--get-all, 736
--help, 736
--page, 736
--per-page, 736
--project-id, 736
--sudo, 736
-h, 736
gitlab-project-pipeline-schedule-pipeline
    command line option
--help, 737
-h, 737
gitlab-project-pipeline-schedule-pipeline-list
    command line option
--get-all, 738
--help, 738
--page, 738
--per-page, 738
--pipeline-schedule-id, 738
--project-id, 738
--sudo, 738
-h, 738
gitlab-project-pipeline-schedule-play
    command line option
--help, 736
--id, 736
--project-id, 736
--sudo, 736
-h, 736
gitlab-project-pipeline-schedule-take-ownership
    command line option
--help, 737
--id, 737
--project-id, 737
--sudo, 737
-h, 737
gitlab-project-pipeline-schedule-update
    command line option
--active, 737
--cron, 737
--cron-timezone, 737
--description, 737
--help, 737

```

```
--id, 737
--project-id, 737
--ref, 737
--sudo, 737
-h, 737
gitlab-project-pipeline-schedule-variable
    command line option
    --help, 738
    -h, 738
gitlab-project-pipeline-schedule-variable-create
    command line option
    --help, 738
    --key, 739
    --pipeline-schedule-id, 739
    --project-id, 738
    --sudo, 738
    --value, 739
    -h, 738
gitlab-project-pipeline-schedule-variable-delete
    command line option
    --help, 739
    --key, 739
    --pipeline-schedule-id, 739
    --project-id, 739
    --sudo, 739
    -h, 739
gitlab-project-pipeline-schedule-variable-update
    command line option
    --help, 739
    --key, 739
    --pipeline-schedule-id, 739
    --project-id, 739
    --sudo, 739
    --value, 739
    -h, 739
gitlab-project-pipeline-test-report command
    line option
    --help, 740
    -h, 740
gitlab-project-pipeline-test-report-get
    command line option
    --help, 740
    --pipeline-id, 740
    --project-id, 740
    --sudo, 740
    -h, 740
gitlab-project-pipeline-test-report-summary
    command line option
    --help, 740
    -h, 740
gitlab-project-pipeline-test-report-summary-get
    command line option
    --help, 740
    --pipeline-id, 740
--project-id, 740
--sudo, 740
-h, 740
gitlab-project-pipeline-variable command
    line option
    --help, 741
    -h, 741
gitlab-project-pipeline-variable-list
    command line option
    --get-all, 741
    --help, 741
    --page, 741
    --per-page, 741
    --pipeline-id, 741
    --project-id, 741
    --sudo, 741
    -h, 741
gitlab-project-protected-branch command
    line option
    --help, 741
    -h, 741
gitlab-project-protected-branch-create
    command line option
    --allow-force-push, 742
    --allowed-to-merge, 742
    --allowed-to-push, 742
    --allowed-to-unprotect, 742
    --code-owner-approval-required, 742
    --help, 742
    --merge-access-level, 742
    --name, 742
    --project-id, 742
    --push-access-level, 742
    --sudo, 742
    --unprotect-access-level, 742
    -h, 742
gitlab-project-protected-branch-delete
    command line option
    --help, 742
    --name, 742
    --project-id, 742
    --sudo, 742
    -h, 742
gitlab-project-protected-branch-get command
    line option
    --help, 743
    --name, 743
    --project-id, 743
    --sudo, 743
    -h, 743
gitlab-project-protected-branch-list
    command line option
    --get-all, 743
    --help, 743
```



```

--page, 743
--per-page, 743
--project-id, 743
--sudo, 743
-h, 743
gitlab-project-protected-branch-update
    command line option
--help, 743
--name, 743
--project-id, 743
--sudo, 743
-h, 743
gitlab-project-protected-environment
    command line option
--help, 744
-h, 744
gitlab-project-protected-environment-create
    command line option
--approval-rules, 744
--deploy-access-levels, 744
--help, 744
--name, 744
--project-id, 744
--required-approval-count, 744
--sudo, 744
-h, 744
gitlab-project-protected-environment-delete
    command line option
--help, 744
--name, 744
--project-id, 744
--sudo, 744
-h, 744
gitlab-project-protected-environment-get
    command line option
--help, 745
--name, 745
--project-id, 745
--sudo, 745
-h, 745
gitlab-project-protected-environment-list
    command line option
--get-all, 745
--help, 745
--page, 745
--per-page, 745
--project-id, 745
--sudo, 745
-h, 745
gitlab-project-protected-tag command line
    option
--help, 745
-h, 745
gitlab-project-protected-tag-create command
    line option
--create-access-level, 746
--help, 746
--name, 746
--project-id, 746
--sudo, 746
-h, 746
gitlab-project-protected-tag-delete command
    line option
--help, 746
--name, 746
--project-id, 746
--sudo, 746
-h, 746
gitlab-project-protected-tag-get command
    line option
--help, 746
--name, 746
--project-id, 746
--sudo, 746
-h, 746
gitlab-project-protected-tag-list command
    line option
--get-all, 747
--help, 747
--page, 747
--per-page, 747
--project-id, 747
--sudo, 747
-h, 747
gitlab-project-push-rules command line
    option
--help, 747
-h, 747
gitlab-project-push-rules-create command
    line option
--author-email-regex, 747
--branch-name-regex, 748
--commit-committer-check, 748
--commit-message-negative-regex, 748
--commit-message-regex, 748
--deny-delete-tag, 748
--file-name-regex, 748
--help, 747
--max-file-size, 748
--member-check, 748
--prevent-secrets, 748
--project-id, 747
--reject-unsigned-commits, 748
--sudo, 747
-h, 747
gitlab-project-push-rules-delete command
    line option

```

--help, 748  
--project-id, 748  
--sudo, 748  
-h, 748  
gitlab-project-push-rules-get command line option  
--help, 748  
--project-id, 748  
--sudo, 748  
-h, 748  
gitlab-project-push-rules-update command line option  
--author-email-regex, 749  
--branch-name-regex, 749  
--commit-committer-check, 749  
--commit-message-negative-regex, 749  
--commit-message-regex, 749  
--deny-delete-tag, 749  
--file-name-regex, 749  
--help, 749  
--max-file-size, 749  
--member-check, 749  
--prevent-secrets, 749  
--project-id, 749  
--reject-unsigned-commits, 749  
--sudo, 749  
-h, 749  
gitlab-project-registry-repository command line option  
--help, 749  
-h, 749  
gitlab-project-registry-repository-delete command line option  
--help, 750  
--id, 750  
--project-id, 750  
--sudo, 750  
-h, 750  
gitlab-project-registry-repository-list command line option  
--get-all, 750  
--help, 750  
--page, 750  
--per-page, 750  
--project-id, 750  
--sudo, 750  
-h, 750  
gitlab-project-registry-tag command line option  
--help, 750  
-h, 750  
gitlab-project-registry-tag-delete command line option  
--help, 750  
--project-id, 751  
--repository-id, 751  
--sudo, 750  
-h, 750  
gitlab-project-registry-tag-delete-in-bulk command line option  
--help, 751  
--keep-n, 751  
--name, 751  
--name-regex-delete, 751  
--name-regex-keep, 751  
--older-than, 751  
--project-id, 751  
--repository-id, 751  
--sudo, 751  
-h, 751  
gitlab-project-registry-tag-get command line option  
--help, 751  
--name, 751  
--project-id, 751  
--repository-id, 751  
--sudo, 751  
-h, 751  
gitlab-project-registry-tag-list command line option  
--get-all, 752  
--help, 752  
--page, 752  
--per-page, 752  
--project-id, 752  
--repository-id, 752  
--sudo, 752  
-h, 752  
gitlab-project-release command line option  
--help, 752  
-h, 752  
gitlab-project-release-create command line option  
--assets, 752  
--description, 752  
--help, 752  
--name, 752  
--project-id, 752  
--ref, 752  
--sudo, 752  
--tag-name, 752  
-h, 752  
gitlab-project-release-delete command line option  
--help, 753  
--project-id, 753  
--sudo, 753

```

    --tag-name, 753
    -h, 753
gitlab-project-release-get command line
    option
    --help, 753
    --project-id, 753
    --sudo, 753
    --tag-name, 753
    -h, 753
gitlab-project-release-link command line
    option
    --help, 754
    -h, 754
gitlab-project-release-link-create command
    line option
    --direct-asset-path, 755
    --filepath, 754
    --help, 754
    --link-type, 755
    --name, 754
    --project-id, 754
    --sudo, 754
    --tag-name, 754
    --url, 754
    -h, 754
gitlab-project-release-link-delete command
    line option
    --help, 755
    --id, 755
    --project-id, 755
    --sudo, 755
    --tag-name, 755
    -h, 755
gitlab-project-release-link-get command
    line option
    --help, 755
    --id, 755
    --project-id, 755
    --sudo, 755
    --tag-name, 755
    -h, 755
gitlab-project-release-link-list command
    line option
    --get-all, 756
    --help, 755
    --page, 756
    --per-page, 756
    --project-id, 755
    --sudo, 755
    --tag-name, 755
    -h, 755
gitlab-project-release-link-update command
    line option
    --direct-asset-path, 756
    --filepath, 756
    --help, 756
    --id, 756
    --link-type, 756
    --name, 756
    --project-id, 756
    --sudo, 756
    --tag-name, 756
    --url, 756
    -h, 756
gitlab-project-release-list command line
    option
    --get-all, 753
    --help, 753
    --include-html-description, 753
    --order-by, 753
    --page, 753
    --per-page, 753
    --project-id, 753
    --sort, 753
    --sudo, 753
    -h, 753
gitlab-project-release-update command line
    option
    --description, 754
    --help, 754
    --milestones, 754
    --name, 754
    --project-id, 754
    --released-at, 754
    --sudo, 754
    --tag-name, 754
    -h, 754
gitlab-project-remote-mirror command line
    option
    --help, 756
    -h, 756
gitlab-project-remote-mirror-create command
    line option
    --enabled, 757
    --help, 757
    --only-protected-branches, 757
    --project-id, 757
    --sudo, 757
    --url, 757
    -h, 757
gitlab-project-remote-mirror-delete command
    line option
    --help, 757
    --id, 757
    --project-id, 757
    --sudo, 757
    -h, 757

```

`gitlab-project-remote-mirror-list` command line option

- `--get-all`, 757
- `--help`, 757
- `--page`, 757
- `--per-page`, 757
- `--project-id`, 757
- `--sudo`, 757
- `-h`, 757

`gitlab-project-remote-mirror-update` command line option

- `--enabled`, 758
- `--help`, 758
- `--id`, 758
- `--only-protected-branches`, 758
- `--project-id`, 758
- `--sudo`, 758
- `-h`, 758

`gitlab-project-repository-archive` command line option

- `--format`, 593
- `--help`, 593
- `--id`, 593
- `--sha`, 593
- `-h`, 593

`gitlab-project-repository-blob` command line option

- `--help`, 593
- `--id`, 593
- `--sha`, 593
- `-h`, 593

`gitlab-project-repository-compare` command line option

- `--from-`, 593
- `--help`, 593
- `--id`, 593
- `--to`, 593
- `-h`, 593

`gitlab-project-repository-contributors` command line option

- `--help`, 594
- `--id`, 594
- `-h`, 594

`gitlab-project-repository-merge-base` command line option

- `--help`, 594
- `--id`, 594
- `--refs`, 594
- `-h`, 594

`gitlab-project-repository-raw-blob` command line option

- `--help`, 594
- `--id`, 594
- `--sha`, 594
- `-h`, 594

`gitlab-project-repository-tree` command line option

- `--help`, 594
- `--id`, 594
- `--path`, 594
- `--recursive`, 594
- `--ref`, 594
- `-h`, 594

`gitlab-project-resource-group` command line option

- `--help`, 758
- `-h`, 758

`gitlab-project-resource-group-get` command line option

- `--help`, 758
- `--key`, 758
- `--project-id`, 758
- `--sudo`, 758
- `-h`, 758

`gitlab-project-resource-group-list` command line option

- `--get-all`, 759
- `--help`, 758
- `--include-html-description`, 759
- `--order-by`, 759
- `--page`, 759
- `--per-page`, 759
- `--project-id`, 759
- `--sort`, 759
- `--sudo`, 759
- `-h`, 758

`gitlab-project-resource-group-upcoming-job` command line option

- `--help`, 759
- `-h`, 759

`gitlab-project-resource-group-upcoming-job-list` command line option

- `--get-all`, 760
- `--help`, 759
- `--page`, 760
- `--per-page`, 760
- `--project-id`, 760
- `--resource-group-key`, 760
- `--sudo`, 760
- `-h`, 759

`gitlab-project-resource-group-update` command line option

- `--help`, 759
- `--key`, 759
- `--process-mode`, 759
- `--project-id`, 759
- `--sudo`, 759
- `-h`, 759

- gitlab-project-restore command line option
  - help, [595](#)
  - id, [595](#)
  - h, [595](#)
- gitlab-project-runner command line option
  - help, [760](#)
  - h, [760](#)
- gitlab-project-runner-create command line option
  - help, [760](#)
  - project-id, [760](#)
  - runner-id, [760](#)
  - sudo, [760](#)
  - h, [760](#)
- gitlab-project-runner-delete command line option
  - help, [760](#)
  - id, [760](#)
  - project-id, [760](#)
  - sudo, [760](#)
  - h, [760](#)
- gitlab-project-runner-list command line option
  - get-all, [761](#)
  - help, [761](#)
  - page, [761](#)
  - per-page, [761](#)
  - project-id, [761](#)
  - scope, [761](#)
  - sudo, [761](#)
  - tag-list, [761](#)
  - h, [761](#)
- gitlab-project-search command line option
  - help, [595](#)
  - id, [595](#)
  - scope, [595](#)
  - search, [595](#)
  - h, [595](#)
- gitlab-project-secure-file command line option
  - help, [761](#)
  - h, [761](#)
- gitlab-project-secure-file-create command line option
  - file, [761](#)
  - help, [761](#)
  - name, [761](#)
  - project-id, [761](#)
  - sudo, [761](#)
  - h, [761](#)
- gitlab-project-secure-file-delete command line option
  - help, [762](#)
  - id, [762](#)
  - project-id, [762](#)
  - sudo, [762](#)
  - h, [762](#)
- gitlab-project-secure-file-download command line option
  - help, [762](#)
  - id, [762](#)
  - project-id, [762](#)
  - sudo, [762](#)
  - h, [762](#)
- gitlab-project-secure-file-get command line option
  - help, [762](#)
  - id, [762](#)
  - project-id, [762](#)
  - sudo, [762](#)
  - h, [762](#)
- gitlab-project-secure-file-list command line option
  - get-all, [763](#)
  - help, [762](#)
  - page, [763](#)
  - per-page, [763](#)
  - project-id, [762](#)
  - sudo, [762](#)
  - h, [762](#)
- gitlab-project-service command line option
  - help, [763](#)
  - h, [763](#)
- gitlab-project-service-available command line option
  - help, [763](#)
  - project-id, [763](#)
  - slug, [763](#)
  - sudo, [763](#)
  - h, [763](#)
- gitlab-project-service-delete command line option
  - help, [763](#)
  - project-id, [763](#)
  - slug, [763](#)
  - sudo, [763](#)
  - h, [763](#)
- gitlab-project-service-get command line option
  - help, [764](#)
  - project-id, [764](#)
  - slug, [764](#)
  - sudo, [764](#)
  - h, [764](#)
- gitlab-project-service-list command line option
  - get-all, [764](#)
  - help, [764](#)

```
--page, 764
--per-page, 764
--project-id, 764
--sudo, 764
-h, 764
gitlab-project-service-update command line
    option
--help, 764
--project-id, 764
--slug, 764
--sudo, 764
-h, 764
gitlab-project-share command line option
--expires-at, 595
--group-access, 595
--group-id, 595
--help, 595
--id, 595
-h, 595
gitlab-project-snapshot command line option
--help, 595
--id, 595
--wiki, 595
-h, 595
gitlab-project-snippet command line option
--help, 765
-h, 765
gitlab-project-snippet-award-emoji command
    line option
--help, 767
-h, 767
gitlab-project-snippet-award-emoji-create
    command line option
--help, 768
--name, 768
--project-id, 768
--snippet-id, 768
--sudo, 768
-h, 768
gitlab-project-snippet-award-emoji-delete
    command line option
--help, 768
--id, 768
--project-id, 768
--snippet-id, 768
--sudo, 768
-h, 768
gitlab-project-snippet-award-emoji-get
    command line option
--help, 768
--id, 768
--project-id, 768
--snippet-id, 768
--sudo, 768
-h, 768
gitlab-project-snippet-award-emoji-list
    command line option
--get-all, 769
--help, 769
--page, 769
--per-page, 769
--project-id, 769
--snippet-id, 769
--sudo, 769
-h, 769
gitlab-project-snippet-content command line
    option
--help, 765
--id, 765
--project-id, 765
--sudo, 765
-h, 765
gitlab-project-snippet-create command line
    option
--content, 765
--description, 765
--file-name, 765
--files, 765
--help, 765
--project-id, 765
--sudo, 765
--title, 765
--visibility, 765
-h, 765
gitlab-project-snippet-delete command line
    option
--help, 766
--id, 766
--project-id, 766
--sudo, 766
-h, 766
gitlab-project-snippet-discussion command
    line option
--help, 769
-h, 769
gitlab-project-snippet-discussion-create
    command line option
--body, 769
--created-at, 769
--help, 769
--project-id, 769
--snippet-id, 769
--sudo, 769
-h, 769
gitlab-project-snippet-discussion-get
    command line option
--help, 770
--id, 770
```

```

--project-id, 770
--snippet-id, 770
--sudo, 770
-h, 770
gitlab-project-snippet-discussion-list
    command line option
--get-all, 770
--help, 770
--page, 770
--per-page, 770
--project-id, 770
--snippet-id, 770
--sudo, 770
-h, 770
gitlab-project-snippet-discussion-note
    command line option
--help, 770
-h, 770
gitlab-project-snippet-discussion-note-create
    command line option
--body, 771
--created-at, 771
--discussion-id, 771
--help, 771
--project-id, 771
--snippet-id, 771
--sudo, 771
-h, 771
gitlab-project-snippet-discussion-note-delete
    command line option
--discussion-id, 771
--help, 771
--id, 771
--project-id, 771
--snippet-id, 771
--sudo, 771
-h, 771
gitlab-project-snippet-discussion-note-get
    command line option
--discussion-id, 772
--help, 771
--id, 772
--project-id, 772
--snippet-id, 772
--sudo, 772
-h, 771
gitlab-project-snippet-discussion-note-update
    command line option
--body, 772
--discussion-id, 772
--help, 772
--id, 772
--project-id, 772
--snippet-id, 772
--sudo, 772
-h, 772
gitlab-project-snippet-get
    command line option
--help, 766
--id, 766
--project-id, 766
--sudo, 766
-h, 766
gitlab-project-snippet-list
    command line option
--get-all, 766
--help, 766
--page, 766
--per-page, 766
--project-id, 766
--sudo, 766
-h, 766
gitlab-project-snippet-note
    command line option
--help, 772
-h, 772
gitlab-project-snippet-note-award-emoji
    command line option
--help, 774
-h, 774
gitlab-project-snippet-note-award-emoji-create
    command line option
--help, 774
--name, 775
--note-id, 775
--project-id, 775
--snippet-id, 775
--sudo, 775
-h, 774
gitlab-project-snippet-note-award-emoji-delete
    command line option
--help, 775
--id, 775
--note-id, 775
--project-id, 775
--snippet-id, 775
--sudo, 775
-h, 775
gitlab-project-snippet-note-award-emoji-get
    command line option
--help, 775
--id, 775
--note-id, 775
--project-id, 775
--snippet-id, 775
--sudo, 775
-h, 775

```

```
gitlab-project-snippet-note-award-emoji-list command line option
  --get-all, 776
  --help, 776
  --note-id, 776
  --page, 776
  --per-page, 776
  --project-id, 776
  --snippet-id, 776
  --sudo, 776
  -h, 776
gitlab-project-snippet-note-create command line option
  --body, 773
  --help, 772
  --project-id, 773
  --snippet-id, 773
  --sudo, 772
  -h, 772
gitlab-project-snippet-note-delete command line option
  --help, 773
  --id, 773
  --project-id, 773
  --snippet-id, 773
  --sudo, 773
  -h, 773
gitlab-project-snippet-note-get command line option
  --help, 773
  --id, 773
  --project-id, 773
  --snippet-id, 773
  --sudo, 773
  -h, 773
gitlab-project-snippet-note-list command line option
  --get-all, 774
  --help, 773
  --page, 774
  --per-page, 774
  --project-id, 774
  --snippet-id, 774
  --sudo, 773
  -h, 773
gitlab-project-snippet-note-update command line option
  --body, 774
  --help, 774
  --id, 774
  --project-id, 774
  --snippet-id, 774
  --sudo, 774
  -h, 774
gitlab-project-snippet-update command line option
  --content, 767
  --description, 767
  --file-name, 767
  --files, 767
  --help, 767
  --id, 767
  --project-id, 767
  --sudo, 767
  --title, 767
  --visibility, 767
  -h, 767
gitlab-project-snippet-user-agent-detail command line option
  --help, 767
  --id, 767
  --project-id, 767
  --sudo, 767
  -h, 767
gitlab-project-star command line option
  --help, 596
  --id, 596
  -h, 596
gitlab-project-storage command line option
  --help, 776
  -h, 776
gitlab-project-storage-get command line option
  --help, 776
  --id, 776
  --project-id, 776
  --sudo, 776
  -h, 776
gitlab-project-tag command line option
  --help, 777
  -h, 777
gitlab-project-tag-create command line option
  --help, 777
  --message, 777
  --project-id, 777
  --ref, 777
  --sudo, 777
  --tag-name, 777
  -h, 777
gitlab-project-tag-delete command line option
  --help, 777
  --name, 777
  --project-id, 777
  --sudo, 777
  -h, 777
gitlab-project-tag-get command line option
```



```

--help, 777
--name, 777
--project-id, 777
--sudo, 777
-h, 777
gitlab-project-tag-list command line option
--get-all, 778
--help, 778
--page, 778
--per-page, 778
--project-id, 778
--sudo, 778
-h, 778
gitlab-project-transfer command line option
--help, 596
--id, 596
--to-namespace, 596
-h, 596
gitlab-project-trigger command line option
--help, 778
-h, 778
gitlab-project-trigger-create command line
    option
--description, 778
--help, 778
--project-id, 778
--sudo, 778
-h, 778
gitlab-project-trigger-delete command line
    option
--help, 778
--id, 778
--project-id, 778
--sudo, 778
-h, 778
gitlab-project-trigger-get command line
    option
--help, 779
--id, 779
--project-id, 779
--sudo, 779
-h, 779
gitlab-project-trigger-list command line
    option
--get-all, 779
--help, 779
--page, 779
--per-page, 779
--project-id, 779
--sudo, 779
-h, 779
gitlab-project-trigger-pipeline command
    line option
--help, 596
--id, 596
--ref, 596
--token, 596
-h, 596
gitlab-project-trigger-update command line
    option
--description, 779
--help, 779
--id, 779
--project-id, 779
--sudo, 779
-h, 779
gitlab-project-unarchive command line
    option
--help, 596
--id, 596
-h, 596
gitlab-project-unshare command line option
--group-id, 597
--help, 597
--id, 597
-h, 597
gitlab-project-unstar command line option
--help, 597
--id, 597
-h, 597
gitlab-project-update command line option
--allow-merge-on-skipped-pipeline, 599
--analytics-access-level, 599
--approvals-before-merge, 599
--auto-cancel-pending-pipelines, 599
--auto-devops-deploy-strategy, 599
--auto-devops-enabled, 599
--autoclose-referenced-issues, 599
--avatar, 599
--build-coverage-regex, 599
--build-git-strategy, 599
--build-timeout, 599
--builds-access-level, 599
--ci-allow-fork-pipelines-to-run-in-parent-project,
    599
--ci-config-path, 599
--ci-default-git-depth, 599
--ci-forward-deployment-enabled, 599
--ci-separated-caches, 600
--container-expiration-policy-attributes,
    600
--container-registry-access-level, 600
--container-registry-enabled, 600
--default-branch, 600
--description, 600
--emails-disabled, 600
--enforce-auth-checks-on-uploads, 600
--environments-access-level, 601

```

`--external-authorization-classification-label`, `--show-default-award-emojis`, 601  
600  
`--feature-flags-access-level`, 601  
`--forking-access-level`, 600  
`--help`, 599  
`--id`, 599  
`--import-url`, 600  
`--infrastructure-access-level`, 601  
`--issue-branch-template`, 601  
`--issues-access-level`, 600  
`--issues-enabled`, 600  
`--issues-template`, 600  
`--jobs-enabled`, 600  
`--keep-latest-artifact`, 600  
`--lfs-enabled`, 600  
`--merge-commit-template`, 600  
`--merge-method`, 600  
`--merge-pipelines-enabled`, 600  
`--merge-requests-access-level`, 600  
`--merge-requests-enabled`, 600  
`--merge-requests-template`, 600  
`--merge-trains-enabled`, 600  
`--mirror`, 600  
`--mirror-overwrites-diverged-branches`,  
600  
`--mirror-trigger-builds`, 600  
`--mirror-user-id`, 600  
`--monitor-access-level`, 601  
`--mr-default-target-self`, 600  
`--name`, 600  
`--only-allow-merge-if-all-discussions-are-enabled`, 601  
601  
`--only-allow-merge-if-all-status-checks-passed`, 599  
`--only-allow-merge-if-pipeline-succeeds`,  
601  
`--only-mirror-protected-branches`, 601  
`--operations-access-level`, 600  
`--packages-enabled`, 601  
`--pages-access-level`, 601  
`--path`, 601  
`--public-builds`, 601  
`--releases-access-level`, 601  
`--remove-source-branch-after-merge`, 601  
`--repository-access-level`, 601  
`--repository-storage`, 601  
`--request-access-enabled`, 601  
`--requirements-access-level`, 601  
`--resolve-outdated-diff-discussions`, 601  
`--restrict-user-defined-variables`, 601  
`--security-and-compliance-access-level`,  
601  
`--service-desk-enabled`, 601  
`--shared-runners-enabled`, 601  
`--snippets-access-level`, 601  
`--snippets-enabled`, 601  
`--squash-commit-template`, 601  
`--squash-option`, 601  
`--sudo`, 599  
`--suggestion-commit-message`, 601  
`--tag-list`, 601  
`--topics`, 601  
`--visibility`, 602  
`--wiki-access-level`, 602  
`--wiki-enabled`, 602  
`-h`, 599  
gitlab-project-update-submodule command  
line option  
`--branch`, 602  
`--commit-sha`, 602  
`--help`, 602  
`--id`, 602  
`--submodule`, 602  
`-h`, 602  
gitlab-project-upload command line option  
`--filename`, 602  
`--filepath`, 602  
`--help`, 602  
`--id`, 602  
`-h`, 602  
gitlab-project-user command line option  
`--help`, 780  
`-h`, 780  
gitlab-project-user-list command line  
option  
`--get-all`, 780  
`--help`, 780  
`--page`, 780  
`--per-page`, 780  
`--project-id`, 780  
`--search`, 780  
`--skip-users`, 780  
`--sudo`, 780  
`-h`, 780  
gitlab-project-variable command line option  
`--help`, 780  
`-h`, 780  
gitlab-project-variable-create command line  
option  
`--environment-scope`, 781  
`--help`, 780  
`--key`, 781  
`--masked`, 781  
`--project-id`, 781  
`--protected`, 781  
`--sudo`, 780  
`--value`, 781

```

    --variable-type, 781
    -h, 780
gitlab-project-variable-delete command line
    option
    --help, 781
    --key, 781
    --project-id, 781
    --sudo, 781
    -h, 781
gitlab-project-variable-get command line
    option
    --help, 781
    --key, 781
    --project-id, 781
    --sudo, 781
    -h, 781
gitlab-project-variable-list command line
    option
    --get-all, 782
    --help, 781
    --page, 782
    --per-page, 782
    --project-id, 782
    --sudo, 781
    -h, 781
gitlab-project-variable-update command line
    option
    --environment-scope, 782
    --help, 782
    --key, 782
    --masked, 782
    --project-id, 782
    --protected, 782
    --sudo, 782
    --value, 782
    --variable-type, 782
    -h, 782
gitlab-project-wiki command line option
    --help, 782
    -h, 782
gitlab-project-wiki-create command line
    option
    --content, 783
    --format, 783
    --help, 783
    --project-id, 783
    --sudo, 783
    --title, 783
    -h, 783
gitlab-project-wiki-delete command line
    option
    --help, 783
    --project-id, 783
    --slug, 783
    --sudo, 783
    -h, 783
gitlab-project-wiki-get command line option
    --help, 783
    --project-id, 783
    --slug, 783
    --sudo, 783
    -h, 783
gitlab-project-wiki-list command line
    option
    --get-all, 784
    --help, 784
    --page, 784
    --per-page, 784
    --project-id, 784
    --sudo, 784
    --with-content, 784
    -h, 784
gitlab-project-wiki-update command line
    option
    --content, 784
    --format, 784
    --help, 784
    --project-id, 784
    --slug, 784
    --sudo, 784
    --title, 784
    -h, 784
gitlab-project-wiki-upload command line
    option
    --filename, 785
    --filepath, 785
    --help, 784
    --project-id, 784
    --slug, 785
    --sudo, 784
    -h, 784
gitlab-registry-repository command line
    option
    --help, 785
    -h, 785
gitlab-registry-repository-get command line
    option
    --help, 785
    --id, 785
    --sudo, 785
    -h, 785
gitlab-runner command line option
    --help, 785
    -h, 785
gitlab-runner-all command line option
    --help, 785, 788
    --id, 785
    --scope, 785

```

-h, 785, 788  
gitlab-runner-all-list command line option  
--get-all, 788  
--help, 788  
--page, 788  
--paused, 788  
--per-page, 788  
--scope, 788  
--status, 788  
--sudo, 788  
--tag-list, 788  
--type, 788  
-h, 788  
gitlab-runner-create command line option  
--access-level, 786  
--active, 786  
--description, 786  
--help, 786  
--info, 786  
--locked, 786  
--maximum-timeout, 786  
--run-untagged, 786  
--sudo, 786  
--tag-list, 786  
--token, 786  
-h, 786  
gitlab-runner-delete command line option  
--help, 786  
--id, 786  
--sudo, 786  
-h, 786  
gitlab-runner-get command line option  
--help, 786  
--id, 786  
--sudo, 786  
-h, 786  
gitlab-runner-job command line option  
--help, 789  
-h, 789  
gitlab-runner-job-list command line option  
--get-all, 789  
--help, 789  
--page, 789  
--per-page, 789  
--runner-id, 789  
--status, 789  
--sudo, 789  
-h, 789  
gitlab-runner-list command line option  
--get-all, 787  
--help, 787  
--page, 787  
--paused, 787  
--per-page, 787  
--scope, 787  
--status, 787  
--sudo, 787  
--tag-list, 787  
--type, 787  
-h, 787  
gitlab-runner-update command line option  
--access-level, 787  
--active, 787  
--description, 787  
--help, 787  
--id, 787  
--locked, 787  
--maximum-timeout, 787  
--run-untagged, 787  
--sudo, 787  
--tag-list, 787  
-h, 787  
gitlab-runner-verify command line option  
--help, 788  
--id, 788  
--token, 788  
-h, 788  
gitlab-shared-project command line option  
--help, 789  
-h, 789  
gitlab-shared-project-list command line option  
--archived, 790  
--get-all, 790  
--group-id, 790  
--help, 789  
--min-access-level, 790  
--order-by, 790  
--page, 790  
--per-page, 790  
--search, 790  
--simple, 790  
--sort, 790  
--starred, 790  
--sudo, 790  
--visibility, 790  
--with-custom-attributes, 790  
--with-issues-enabled, 790  
--with-merge-requests-enabled, 790  
-h, 789  
gitlab-snippet command line option  
--help, 790  
-h, 790  
gitlab-snippet-content command line option  
--help, 791  
--id, 791  
-h, 791  
gitlab-snippet-create command line option

```

--content, 791
--description, 791
--file-name, 791
--files, 791
--help, 791
--sudo, 791
--title, 791
--visibility, 791
-h, 791
gitlab-snippet-delete command line option
--help, 791
--id, 791
--sudo, 791
-h, 791
gitlab-snippet-get command line option
--help, 792
--id, 792
--sudo, 792
-h, 792
gitlab-snippet-list command line option
--get-all, 792
--help, 792
--page, 792
--per-page, 792
--sudo, 792
-h, 792
gitlab-snippet-public command line option
--help, 792
--id, 792
-h, 792
gitlab-snippet-update command line option
--content, 793
--description, 793
--file-name, 793
--files, 792
--help, 792
--id, 792
--sudo, 792
--title, 792
--visibility, 793
-h, 792
gitlab-snippet-user-agent-detail command
line option
--help, 793
--id, 793
-h, 793
gitlab-starred-project command line option
--help, 793
-h, 793
gitlab-starred-project-list command line
option
--archived, 793
--get-all, 794
--help, 793
--membership, 794
--min-access-level, 794
--order-by, 794
--owned, 794
--page, 794
--per-page, 794
--search, 794
--simple, 794
--sort, 794
--starred, 794
--statistics, 794
--sudo, 793
--user-id, 793
--visibility, 794
--with-custom-attributes, 794
--with-issues-enabled, 794
--with-merge-requests-enabled, 794
-h, 793
gitlab-todo command line option
--help, 794
-h, 794
gitlab-todo-delete command line option
--help, 794
--id, 794
--sudo, 794
-h, 794
gitlab-todo-list command line option
--action, 795
--author-id, 795
--get-all, 795
--help, 795
--page, 795
--per-page, 795
--project-id, 795
--state, 795
--sudo, 795
--type, 795
-h, 795
gitlab-todo-mark-all-as-done command line
option
--help, 795
--id, 795
-h, 795
gitlab-todo-mark-as-done command line
option
--help, 795
--id, 795
-h, 795
gitlab-topic command line option
--help, 796
-h, 796
gitlab-topic-create command line option
--avatar, 796
--description, 796

```

```
--help, 796
--name, 796
--sudo, 796
--title, 796
-h, 796
gitlab-topic-delete command line option
--help, 796
--id, 796
--sudo, 796
-h, 796
gitlab-topic-get command line option
--help, 796
--id, 796
--sudo, 796
-h, 796
gitlab-topic-list command line option
--get-all, 797
--help, 797
--page, 797
--per-page, 797
--sudo, 797
-h, 797
gitlab-topic-merge command line option
--help, 797
--id, 797
--source-topic-id, 797
--target-topic-id, 797
-h, 797
gitlab-topic-update command line option
--avatar, 797
--description, 797
--help, 797
--id, 797
--name, 797
--sudo, 797
-h, 797
gitlab-user command line option
--help, 798
-h, 798
gitlab-user-activate command line option
--help, 798
--id, 798
-h, 798
gitlab-user-activities command line option
--help, 804
-h, 804
gitlab-user-activities-list command line
    option
--get-all, 804
--help, 804
--page, 804
--per-page, 804
--sudo, 804
-h, 804
gitlab-user-approve command line option
--help, 798
--id, 798
-h, 798
gitlab-user-ban command line option
--help, 798
--id, 798
-h, 798
gitlab-user-block command line option
--help, 799
--id, 799
-h, 799
gitlab-user-create command line option
--admin, 800
--avatar, 800
--bio, 799
--can-create-group, 800
--color-scheme-id, 800
--email, 799
--extern-uid, 799
--external, 800
--help, 799
--linkedin, 799
--location, 800
--name, 799
--organization, 800
--password, 799
--private-profile, 800
--projects-limit, 799
--provider, 799
--public-email, 800
--reset-password, 799
--skip-confirmation, 800
--skype, 799
--sudo, 799
--theme-id, 800
--twitter, 799
--username, 799
--website-url, 800
-h, 799
gitlab-user-custom-attribute command line
    option
--help, 804
-h, 804
gitlab-user-custom-attribute-delete command
    line option
--help, 805
--key, 805
--sudo, 805
--user-id, 805
-h, 805
gitlab-user-custom-attribute-get command
    line option
--help, 805
```

```

--key, 805
--sudo, 805
--user-id, 805
-h, 805
gitlab-user-custom-attribute-list command
  line option
--get-all, 805
--help, 805
--page, 805
--per-page, 805
--sudo, 805
--user-id, 805
-h, 805
gitlab-user-deactivate command line option
--help, 800
--id, 800
-h, 800
gitlab-user-delete command line option
--help, 800
--id, 800
--sudo, 800
-h, 800
gitlab-user-email command line option
--help, 806
-h, 806
gitlab-user-email-create command line
  option
--email, 806
--help, 806
--sudo, 806
--user-id, 806
-h, 806
gitlab-user-email-delete command line
  option
--help, 806
--id, 806
--sudo, 806
--user-id, 806
-h, 806
gitlab-user-email-get command line option
--help, 806
--id, 806
--sudo, 806
--user-id, 806
-h, 806
gitlab-user-email-list command line option
--get-all, 807
--help, 807
--page, 807
--per-page, 807
--sudo, 807
--user-id, 807
-h, 807
gitlab-user-event command line option
--help, 807
-h, 807
gitlab-user-event-list command line option
--action, 807
--after, 807
--before, 807
--get-all, 808
--help, 807
--page, 807
--per-page, 807
--scope, 807
--sort, 807
--sudo, 807
--target-type, 807
--user-id, 807
-h, 807
gitlab-user-follow command line option
--help, 801
--id, 801
-h, 801
gitlab-user-get command line option
--help, 801
--id, 801
--sudo, 801
-h, 801
gitlab-user-gpg-key command line option
--help, 808
-h, 808
gitlab-user-gpg-key-create command line
  option
--help, 808
--key, 808
--sudo, 808
--user-id, 808
-h, 808
gitlab-user-gpg-key-delete command line
  option
--help, 808
--id, 808
--sudo, 808
--user-id, 808
-h, 808
gitlab-user-gpg-key-get command line option
--help, 808
--id, 808
--sudo, 808
--user-id, 808
-h, 808
gitlab-user-gpg-key-list command line
  option
--get-all, 809
--help, 809
--page, 809
--per-page, 809

```

```
--sudo, 809
--user-id, 809
-h, 809
gitlab-user-impersonation-token command
  line option
  --help, 809
  -h, 809
gitlab-user-impersonation-token-create
  command line option
  --expires-at, 809
  --help, 809
  --name, 809
  --scopes, 809
  --sudo, 809
  --user-id, 809
  -h, 809
gitlab-user-impersonation-token-delete
  command line option
  --help, 810
  --id, 810
  --sudo, 810
  --user-id, 810
  -h, 810
gitlab-user-impersonation-token-get command
  line option
  --help, 810
  --id, 810
  --sudo, 810
  --user-id, 810
  -h, 810
gitlab-user-impersonation-token-list
  command line option
  --get-all, 810
  --help, 810
  --page, 810
  --per-page, 810
  --state, 810
  --sudo, 810
  --user-id, 810
  -h, 810
gitlab-user-key command line option
  --help, 811
  -h, 811
gitlab-user-key-create command line option
  --help, 811
  --key, 811
  --sudo, 811
  --title, 811
  --user-id, 811
  -h, 811
gitlab-user-key-delete command line option
  --help, 811
  --id, 811
  --sudo, 811
  --user-id, 811
  -h, 811
gitlab-user-key-get command line option
  --help, 811
  --id, 811
  --sudo, 811
  --user-id, 811
  -h, 811
gitlab-user-key-list command line option
  --get-all, 812
  --help, 812
  --page, 812
  --per-page, 812
  --sudo, 812
  --user-id, 812
  -h, 812
gitlab-user-list command line option
  --active, 801
  --blocked, 801
  --custom-attributes, 801
  --extern-uid, 801
  --external, 801
  --get-all, 802
  --help, 801
  --page, 801
  --per-page, 802
  --provider, 801
  --search, 801
  --status, 801
  --sudo, 801
  --two-factor, 801
  --username, 801
  -h, 801
gitlab-user-membership command line option
  --help, 812
  -h, 812
gitlab-user-membership-get command line
  option
  --help, 812
  --source-id, 812
  --sudo, 812
  --user-id, 812
  -h, 812
gitlab-user-membership-list command line
  option
  --get-all, 813
  --help, 812
  --page, 813
  --per-page, 813
  --sudo, 812
  --type, 812
  --user-id, 812
  -h, 812
```



```

gitlab-user-personal-access-token command
    line option
    --help, 813
    -h, 813
gitlab-user-personal-access-token-create
    command line option
    --expires-at, 813
    --help, 813
    --name, 813
    --scopes, 813
    --sudo, 813
    --user-id, 813
    -h, 813
gitlab-user-project command line option
    --help, 813
    -h, 813
gitlab-user-project-create command line
    option
    --builds-enabled, 814
    --default-branch, 814
    --description, 814
    --help, 814
    --import-url, 814
    --issues-enabled, 814
    --merge-requests-enabled, 814
    --name, 814
    --only-allow-merge-if-build-succeeds, 814
    --public, 814
    --public-builds, 814
    --snippets-enabled, 814
    --squash-option, 814
    --sudo, 814
    --user-id, 814
    --visibility, 814
    --wall-enabled, 814
    --wiki-enabled, 814
    -h, 814
gitlab-user-project-list command line
    option
    --archived, 815
    --get-all, 816
    --help, 815
    --id-after, 816
    --id-before, 816
    --membership, 815
    --min-access-level, 816
    --order-by, 815
    --owned, 815
    --page, 816
    --per-page, 816
    --repository-checksum-failed, 815
    --search, 815
    --simple, 815
    --sort, 815
    --starred, 815
    --statistics, 815
    --sudo, 815
    --user-id, 815
    --visibility, 815
    --wiki-checksum-failed, 815
    --with-custom-attributes, 815
    --with-issues-enabled, 815
    --with-merge-requests-enabled, 815
    --with-programming-language, 815
    -h, 815
gitlab-user-reject command line option
    --help, 802
    --id, 802
    -h, 802
gitlab-user-status command line option
    --help, 816
    -h, 816
gitlab-user-status-get command line option
    --help, 816
    --sudo, 816
    --user-id, 816
    -h, 816
gitlab-user-unban command line option
    --help, 802
    --id, 802
    -h, 802
gitlab-user-unblock command line option
    --help, 802
    --id, 802
    -h, 802
gitlab-user-unfollow command line option
    --help, 802
    --id, 802
    -h, 802
gitlab-user-update command line option
    --admin, 803
    --avatar, 804
    --bio, 803
    --can-create-group, 803
    --color-scheme-id, 804
    --email, 803
    --extern-uid, 803
    --external, 803
    --help, 803
    --id, 803
    --linkedin, 803
    --location, 804
    --name, 803
    --organization, 803
    --password, 803
    --private-profile, 804
    --projects-limit, 803
    --provider, 803

```

--public-email, 804  
--skip-reconfirmation, 803  
--skype, 803  
--sudo, 803  
--theme-id, 804  
--twitter, 803  
--username, 803  
--website-url, 803  
-h, 803  
gitlab-variable command line option  
--help, 816  
-h, 816  
gitlab-variable-create command line option  
--help, 816  
--key, 817  
--masked, 817  
--protected, 817  
--sudo, 816  
--value, 817  
--variable-type, 817  
-h, 816  
gitlab-variable-delete command line option  
--help, 817  
--key, 817  
--sudo, 817  
-h, 817  
gitlab-variable-get command line option  
--help, 817  
--key, 817  
--sudo, 817  
-h, 817  
gitlab-variable-list command line option  
--get-all, 817  
--help, 817  
--page, 817  
--per-page, 817  
--sudo, 817  
-h, 817  
gitlab-variable-update command line option  
--help, 818  
--key, 818  
--masked, 818  
--protected, 818  
--sudo, 818  
--value, 818  
--variable-type, 818  
-h, 818  
GitlabActivateError, 448  
GitlabAttachFileError, 448  
GitlabAuthenticationError, 448  
GitlabBanError, 448  
GitlabBlockError, 449  
GitlabBuildCancelError, 449  
GitlabBuildEraseError, 449  
GitlabBuildPlayError, 449  
GitlabBuildRetryError, 450  
GitlabCancelError, 450  
GitlabCherryPickError, 450  
GitlabCiLintError, 450  
Gitlabciyaml (*class in gitlab.v4.objects*), 221  
GitlabciyamlManager (*class in gitlab.v4.objects*), 221  
gitlabciyaml (*gitlab.Gitlab attribute*), 190  
GitlabConfigHelperError, 446  
GitlabConfigMissingError, 446  
GitlabConfigParser (*class in gitlab.config*), 446  
GitlabConnectionError, 451  
GitlabCreateError, 451  
GitlabDataError, 446  
GitlabDeactivateError, 451  
GitlabDeleteError, 451  
GitlabDeploymentApprovalError, 452  
GitlabError, 452  
GitlabFollowError, 452  
GitlabGetError, 452  
GitlabGroupTransferError, 453  
GitlabHeadError, 453  
GitlabHousekeepingError, 453  
GitlabHttpError, 453  
GitlabIDError, 446  
GitlabImportError, 454  
GitlabInvitationError, 454  
GitlabJobCancelError, 454  
GitlabJobEraseError, 454  
GitlabJobPlayError, 455  
GitlabJobRetryError, 455  
GitlabLicenseError, 455  
GitlabList (*class in gitlab*), 197  
GitlabListError, 455  
GitlabMarkdownError, 457  
GitlabMRApprovalError, 455  
GitlabMRClosedError, 456  
GitlabMRForbiddenError, 456  
GitlabMROnBuildSuccessError, 456  
GitlabMRRebaseError, 456  
GitlabMRResetApprovalError, 457  
GitlabOperationError, 457  
GitlabOwnershipError, 457  
GitlabParsingError, 458  
GitlabPipelineCancelError, 458  
GitlabPipelinePlayError, 458  
GitlabPipelineRetryError, 458  
GitlabProjectDeployKeyError, 459  
GitlabPromoteError, 459  
GitlabProtectError, 459  
GitlabRenderError, 459  
GitlabRepairError, 460  
GitlabRestoreError, 460  
GitlabRetryError, 460

- GitlabRevertError, 460
- GitlabRotateError, 460
- GitlabSearchError, 461
- GitlabSetError, 461
- GitlabStopError, 461
- GitlabSubscribeError, 461
- GitlabTimeTrackingError, 462
- GitlabTodoError, 462
- GitlabTopicMergeError, 462
- GitlabTransferProjectError, 462
- GitlabUnbanError, 463
- GitlabUnblockError, 463
- GitlabUnfollowError, 463
- GitlabUnsubscribeError, 463
- GitlabUpdateError, 464
- GitlabUploadError, 464
- GitlabUserApproveError, 464
- GitlabUserRejectError, 464
- GitlabVerifyError, 464
- GLOBAL (*gitlab.const.NotificationLevel* attribute), 447
- GLOBAL\_SNIPPET\_TITLES (*gitlab.const.SearchScope* attribute), 447
- gpgkeys (*gitlab.v4.objects.CurrentUser* attribute), 209
- gpgkeys (*gitlab.v4.objects.User* attribute), 429
- Group (*class in gitlab.v4.objects*), 222
- GroupAccessRequest (*class in gitlab.v4.objects*), 225
- GroupAccessRequestManager (*class in gitlab.v4.objects*), 225
- GroupAccessToken (*class in gitlab.v4.objects*), 225
- GroupAccessTokenManager (*class in gitlab.v4.objects*), 226
- GroupAuditEvent (*class in gitlab.v4.objects*), 226
- GroupAuditEventManager (*class in gitlab.v4.objects*), 227
- GroupBadge (*class in gitlab.v4.objects*), 227
- GroupBadgeManager (*class in gitlab.v4.objects*), 227
- GroupBillableMember (*class in gitlab.v4.objects*), 228
- GroupBillableMemberManager (*class in gitlab.v4.objects*), 228
- GroupBillableMemberMembership (*class in gitlab.v4.objects*), 229
- GroupBillableMemberMembershipManager (*class in gitlab.v4.objects*), 229
- GroupBoard (*class in gitlab.v4.objects*), 229
- GroupBoardList (*class in gitlab.v4.objects*), 229
- GroupBoardListManager (*class in gitlab.v4.objects*), 230
- GroupBoardManager (*class in gitlab.v4.objects*), 230
- GroupCluster (*class in gitlab.v4.objects*), 231
- GroupClusterManager (*class in gitlab.v4.objects*), 231
- GroupCustomAttribute (*class in gitlab.v4.objects*), 233
- GroupCustomAttributeManager (*class in gitlab.v4.objects*), 233
- GroupDeployToken (*class in gitlab.v4.objects*), 233
- GroupDeployTokenManager (*class in gitlab.v4.objects*), 234
- GroupDescendantGroup (*class in gitlab.v4.objects*), 234
- GroupDescendantGroupManager (*class in gitlab.v4.objects*), 235
- GroupEpic (*class in gitlab.v4.objects*), 235
- GroupEpicAwardEmoji (*class in gitlab.v4.objects*), 235
- GroupEpicAwardEmojiManager (*class in gitlab.v4.objects*), 236
- GroupEpicDiscussionNote (*class in gitlab.v4.objects*), 236
- GroupEpicDiscussionNoteManager (*class in gitlab.v4.objects*), 236
- GroupEpicIssue (*class in gitlab.v4.objects*), 237
- GroupEpicIssueManager (*class in gitlab.v4.objects*), 238
- GroupEpicManager (*class in gitlab.v4.objects*), 238
- GroupEpicNote (*class in gitlab.v4.objects*), 239
- GroupEpicNoteAwardEmoji (*class in gitlab.v4.objects*), 240
- GroupEpicNoteAwardEmojiManager (*class in gitlab.v4.objects*), 240
- GroupEpicNoteManager (*class in gitlab.v4.objects*), 241
- GroupEpicResourceLabelEvent (*class in gitlab.v4.objects*), 241
- GroupEpicResourceLabelEventManager (*class in gitlab.v4.objects*), 242
- GroupExport (*class in gitlab.v4.objects*), 242
- GroupExportManager (*class in gitlab.v4.objects*), 242
- GroupHook (*class in gitlab.v4.objects*), 243
- GroupHookManager (*class in gitlab.v4.objects*), 243
- GroupImport (*class in gitlab.v4.objects*), 245
- GroupImportManager (*class in gitlab.v4.objects*), 245
- GroupInvitation (*class in gitlab.v4.objects*), 245
- GroupInvitationManager (*class in gitlab.v4.objects*), 245
- GroupIssue (*class in gitlab.v4.objects*), 246
- GroupIssueManager (*class in gitlab.v4.objects*), 247
- GroupIssuesStatistics (*class in gitlab.v4.objects*), 247
- GroupIssuesStatisticsManager (*class in gitlab.v4.objects*), 247
- GroupIteration (*class in gitlab.v4.objects*), 248
- GroupIterationManager (*class in gitlab.v4.objects*), 248
- GroupLabel (*class in gitlab.v4.objects*), 249
- GroupLabelManager (*class in gitlab.v4.objects*), 250
- GroupLDAPGroupLink (*class in gitlab.v4.objects*), 249
- GroupLDAPGroupLinkManager (*class in gitlab.v4.objects*), 249
- GroupManager (*class in gitlab.v4.objects*), 251

`GroupMember` (class in `gitlab.v4.objects`), 254  
`GroupMemberAll` (class in `gitlab.v4.objects`), 254  
`GroupMemberAllManager` (class in `gitlab.v4.objects`), 254  
`GroupMemberManager` (class in `gitlab.v4.objects`), 254  
`GroupMergeRequest` (class in `gitlab.v4.objects`), 255  
`GroupMergeRequestManager` (class in `gitlab.v4.objects`), 256  
`GroupMilestone` (class in `gitlab.v4.objects`), 256  
`GroupMilestoneManager` (class in `gitlab.v4.objects`), 257  
`GroupNotificationSettings` (class in `gitlab.v4.objects`), 258  
`GroupNotificationSettingsManager` (class in `gitlab.v4.objects`), 258  
`GroupPackage` (class in `gitlab.v4.objects`), 259  
`GroupPackageManager` (class in `gitlab.v4.objects`), 260  
`GroupProject` (class in `gitlab.v4.objects`), 260  
`GroupProjectManager` (class in `gitlab.v4.objects`), 260  
`GroupPushRules` (class in `gitlab.v4.objects`), 261  
`GroupPushRulesManager` (class in `gitlab.v4.objects`), 261  
`GroupRegistryRepositoryManager` (class in `gitlab.v4.objects`), 262  
`GroupRunner` (class in `gitlab.v4.objects`), 262  
`GroupRunnerManager` (class in `gitlab.v4.objects`), 263  
`groups` (`gitlab.Gitlab` attribute), 190  
`groups` (`gitlab.v4.objects.Project` attribute), 278  
`GroupSAMLGroupLink` (class in `gitlab.v4.objects`), 263  
`GroupSAMLGroupLinkManager` (class in `gitlab.v4.objects`), 263  
`GroupSubgroup` (class in `gitlab.v4.objects`), 264  
`GroupSubgroupManager` (class in `gitlab.v4.objects`), 264  
`GroupVariable` (class in `gitlab.v4.objects`), 264  
`GroupVariableManager` (class in `gitlab.v4.objects`), 265  
`GroupWiki` (class in `gitlab.v4.objects`), 265  
`GroupWikiManager` (class in `gitlab.v4.objects`), 266  
`GUEST` (`gitlab.const.AccessLevel` attribute), 447

## H

`headers` (`gitlab.Gitlab` attribute), 190  
`Hook` (class in `gitlab.v4.objects`), 267  
`HookManager` (class in `gitlab.v4.objects`), 267  
`hooks` (`gitlab.Gitlab` attribute), 190  
`hooks` (`gitlab.v4.objects.Group` attribute), 222  
`hooks` (`gitlab.v4.objects.Project` attribute), 278  
`housekeeping()` (`gitlab.v4.objects.Project` method), 278  
`http_delete()` (`gitlab.Gitlab` method), 190  
`http_get()` (`gitlab.Gitlab` method), 190  
`http_head()` (`gitlab.Gitlab` method), 191  
`http_list()` (`gitlab.Gitlab` method), 191

`http_patch()` (`gitlab.Gitlab` method), 192  
`http_post()` (`gitlab.Gitlab` method), 192  
`http_put()` (`gitlab.Gitlab` method), 193  
`http_request()` (`gitlab.Gitlab` method), 193

## I

`id` (`gitlab.v4.objects.ProjectMergeRequestApprovalRule` attribute), 357  
`identityproviders` (`gitlab.v4.objects.User` attribute), 429  
`impersonationtokens` (`gitlab.v4.objects.User` attribute), 429  
`import_bitbucket_server()` (`gitlab.v4.objects.ProjectManager` method), 346  
`import_github()` (`gitlab.v4.objects.ProjectManager` method), 348  
`import_group()` (`gitlab.v4.objects.GroupManager` method), 253  
`import_project()` (`gitlab.v4.objects.ProjectManager` method), 349  
`imports` (`gitlab.v4.objects.Group` attribute), 222  
`imports` (`gitlab.v4.objects.Project` attribute), 278  
`integrations` (`gitlab.v4.objects.Project` attribute), 278  
`INTERNAL` (`gitlab.const.Visibility` attribute), 448  
`invitations` (`gitlab.v4.objects.Group` attribute), 222  
`invitations` (`gitlab.v4.objects.Project` attribute), 278  
`Issue` (class in `gitlab.v4.objects`), 267  
`IssueManager` (class in `gitlab.v4.objects`), 268  
`ISSUES` (`gitlab.const.SearchScope` attribute), 447  
`issues` (`gitlab.Gitlab` attribute), 194  
`issues` (`gitlab.v4.objects.Group` attribute), 222  
`issues` (`gitlab.v4.objects.GroupEpic` attribute), 235  
`issues` (`gitlab.v4.objects.Project` attribute), 279  
`issues()` (`gitlab.v4.objects.GroupMilestone` method), 257  
`issues()` (`gitlab.v4.objects.ProjectMilestone` method), 373  
`issues_statistics` (`gitlab.Gitlab` attribute), 194  
`issues_statistics` (`gitlab.v4.objects.Group` attribute), 222  
`issues_statistics` (`gitlab.v4.objects.Project` attribute), 279  
`IssuesStatistics` (class in `gitlab.v4.objects`), 269  
`IssuesStatisticsManager` (class in `gitlab.v4.objects`), 269  
`iterations` (`gitlab.v4.objects.Group` attribute), 222  
`iterations` (`gitlab.v4.objects.Project` attribute), 279

## J

`job_stats()` (`gitlab.v4.objects.SidekiqManager` method), 420  
`job_token_scope` (`gitlab.v4.objects.Project` attribute), 279



jobs (*gitlab.v4.objects.Project* attribute), 279  
 jobs (*gitlab.v4.objects.ProjectPipeline* attribute), 380  
 jobs (*gitlab.v4.objects.Runner* attribute), 416

## K

keep\_artifacts() (*gitlab.v4.objects.ProjectJob* method), 335  
 Key (class in *gitlab.v4.objects*), 269  
 KeyManager (class in *gitlab.v4.objects*), 270  
 keys (*gitlab.Gitlab* attribute), 194  
 keys (*gitlab.v4.objects.CurrentUser* attribute), 209  
 keys (*gitlab.v4.objects.Project* attribute), 279  
 keys (*gitlab.v4.objects.User* attribute), 429

## L

labels (*gitlab.v4.objects.Group* attribute), 222  
 labels (*gitlab.v4.objects.Project* attribute), 279  
 languages() (*gitlab.v4.objects.Project* method), 279  
 ldap\_group\_links (*gitlab.v4.objects.Group* attribute), 222  
 ldap\_sync() (*gitlab.v4.objects.Group* method), 222  
 LDAPGroup (class in *gitlab.v4.objects*), 270  
 LDAPGroupManager (class in *gitlab.v4.objects*), 270  
 ldapgroups (*gitlab.Gitlab* attribute), 194  
 License (class in *gitlab.v4.objects*), 271  
 LicenseManager (class in *gitlab.v4.objects*), 271  
 licenses (*gitlab.Gitlab* attribute), 194  
 links (*gitlab.v4.objects.ProjectIssue* attribute), 320  
 links (*gitlab.v4.objects.ProjectRelease* attribute), 393  
 list() (*gitlab.mixins.ListMixin* method), 468  
 list() (*gitlab.v4.objects.LDAPGroupManager* method), 271  
 list() (*gitlab.v4.objects.UserProjectManager* method), 440  
 ListMixin (class in *gitlab.mixins*), 468  
 lists (*gitlab.v4.objects.GroupBoard* attribute), 229  
 lists (*gitlab.v4.objects.ProjectBoard* attribute), 291

## M

main() (in module *gitlab.cli*), 446  
 MAINTAINER (*gitlab.const.AccessLevel* attribute), 447  
 manager (*gitlab.base.RESTObject* attribute), 444  
 manager (*gitlab.mixins.AccessRequestMixin* attribute), 465  
 manager (*gitlab.mixins.DownloadMixin* attribute), 467  
 manager (*gitlab.mixins.ObjectDeleteMixin* attribute), 469  
 manager (*gitlab.mixins.ParticipantsMixin* attribute), 469  
 manager (*gitlab.mixins.RefreshMixin* attribute), 469  
 manager (*gitlab.mixins.SaveMixin* attribute), 470  
 manager (*gitlab.mixins.SubscribableMixin* attribute), 471  
 manager (*gitlab.mixins.TimeTrackingMixin* attribute), 471

manager (*gitlab.mixins.TODOMixin* attribute), 473  
 manager (*gitlab.mixins.UserAgentDetailMixin* attribute), 473  
 manager (*gitlab.v4.objects.GroupEpicIssue* attribute), 237  
 manager (*gitlab.v4.objects.GroupLabel* attribute), 250  
 manager (*gitlab.v4.objects.ProjectFile* attribute), 311  
 manager (*gitlab.v4.objects.ProjectLabel* attribute), 339  
 mark\_all\_as\_done() (*gitlab.v4.objects.TODOManager* method), 425  
 mark\_as\_done() (*gitlab.v4.objects.TODO* method), 424  
 markdown() (*gitlab.Gitlab* method), 194  
 MaskingFormatter (class in *gitlab.utils*), 474  
 members (*gitlab.v4.objects.Group* attribute), 223  
 members (*gitlab.v4.objects.Project* attribute), 279  
 members\_all (*gitlab.v4.objects.Group* attribute), 223  
 members\_all (*gitlab.v4.objects.Project* attribute), 279  
 memberships (*gitlab.v4.objects.GroupBillableMember* attribute), 228  
 memberships (*gitlab.v4.objects.User* attribute), 429  
 MENTION (*gitlab.const.NotificationLevel* attribute), 447  
 merge() (*gitlab.v4.objects.ProjectMergeRequest* method), 354  
 merge() (*gitlab.v4.objects.TopicManager* method), 426  
 merge\_config() (*gitlab.Gitlab* class method), 194  
 merge\_ref() (*gitlab.v4.objects.ProjectMergeRequest* method), 355  
 merge\_request\_iid (*gitlab.v4.objects.ProjectMergeRequestApprovalRule* attribute), 357  
 MERGE\_REQUESTS (*gitlab.const.SearchScope* attribute), 447  
 merge\_requests() (*gitlab.v4.objects.GroupMilestone* method), 257  
 merge\_requests() (*gitlab.v4.objects.ProjectCommit* method), 297  
 merge\_requests() (*gitlab.v4.objects.ProjectMilestone* method), 373  
 merge\_trains (*gitlab.v4.objects.Project* attribute), 279  
 MergeRequest (class in *gitlab.v4.objects*), 272  
 MergeRequestManager (class in *gitlab.v4.objects*), 272  
 mergerequests (*gitlab.Gitlab* attribute), 195  
 mergerequests (*gitlab.v4.objects.Group* attribute), 223  
 mergerequests (*gitlab.v4.objects.Project* attribute), 279  
 mergerequests (*gitlab.v4.objects.ProjectDeployment* attribute), 305  
 MILESTONES (*gitlab.const.SearchScope* attribute), 447  
 milestones (*gitlab.v4.objects.Group* attribute), 223  
 milestones (*gitlab.v4.objects.Project* attribute), 279  
 MINIMAL\_ACCESS (*gitlab.const.AccessLevel* attribute), 447  
 mirror\_pull() (*gitlab.v4.objects.Project* method), 279  
 mirror\_pull\_details() (*gitlab.v4.objects.Project* method), 279

## module

gitlab.base, 443  
gitlab.cli, 445  
gitlab.config, 446  
gitlab.const, 447  
gitlab.exceptions, 448  
gitlab.mixins, 465  
gitlab.utils, 474  
gitlab.v4, 443  
gitlab.v4.objects, 198

move() (*gitlab.v4.objects.ProjectIssue* method), 320

## N

Namespace (*class in gitlab.v4.objects*), 273  
NamespaceManager (*class in gitlab.v4.objects*), 273  
namespaces (*gitlab.Gitlab* attribute), 195  
next() (*gitlab.base.RESTObjectList* method), 445  
next() (*gitlab.GitlabList* method), 197  
next\_page (*gitlab.base.RESTObjectList* property), 445  
next\_page (*gitlab.GitlabList* property), 197  
NO\_ACCESS (*gitlab.const.AccessLevel* attribute), 447  
notes (*gitlab.v4.objects.GroupEpic* attribute), 235  
notes (*gitlab.v4.objects.Project* attribute), 280  
notes (*gitlab.v4.objects.ProjectCommitDiscussion* attribute), 299  
notes (*gitlab.v4.objects.ProjectIssue* attribute), 321  
notes (*gitlab.v4.objects.ProjectIssueDiscussion* attribute), 322  
notes (*gitlab.v4.objects.ProjectMergeRequest* attribute), 355  
notes (*gitlab.v4.objects.ProjectMergeRequestDiscussion* attribute), 362  
notes (*gitlab.v4.objects.ProjectSnippet* attribute), 401  
notes (*gitlab.v4.objects.ProjectSnippetDiscussion* attribute), 402  
NotificationLevel (*class in gitlab.const*), 447  
NotificationSettings (*class in gitlab.v4.objects*), 274  
notificationsettings (*gitlab.Gitlab* attribute), 195  
notificationsettings (*gitlab.v4.objects.Group* attribute), 223  
notificationsettings (*gitlab.v4.objects.Project* attribute), 280  
NotificationSettingsManager (*class in gitlab.v4.objects*), 274  
NoUpdateMixin (*class in gitlab.mixins*), 468

## O

ObjectDeleteMixin (*class in gitlab.mixins*), 469  
OWNER (*gitlab.const.AccessLevel* attribute), 447

## P

package\_files (*gitlab.v4.objects.ProjectPackage* attribute), 377

packages (*gitlab.v4.objects.Group* attribute), 223  
packages (*gitlab.v4.objects.Project* attribute), 280  
PagesDomain (*class in gitlab.v4.objects*), 275  
PagesDomainManager (*class in gitlab.v4.objects*), 275  
pagesdomains (*gitlab.Gitlab* attribute), 195  
pagesdomains (*gitlab.v4.objects.Project* attribute), 280  
parent\_attrs (*gitlab.base.RESTManager* property), 443  
participants() (*gitlab.mixins.ParticipantsMixin* method), 469  
ParticipantsMixin (*class in gitlab.mixins*), 469  
PARTICIPATING (*gitlab.const.NotificationLevel* attribute), 447  
path (*gitlab.base.RESTManager* property), 443  
per\_page (*gitlab.base.RESTObjectList* property), 445  
per\_page (*gitlab.GitlabList* property), 197  
personal\_access\_tokens (*gitlab.Gitlab* attribute), 195  
personal\_access\_tokens (*gitlab.v4.objects.User* attribute), 429  
PersonalAccessToken (*class in gitlab.v4.objects*), 276  
PersonalAccessTokenManager (*class in gitlab.v4.objects*), 276  
pformat() (*gitlab.base.RESTObject* method), 444  
pipelines (*gitlab.v4.objects.Project* attribute), 280  
pipelines (*gitlab.v4.objects.ProjectMergeRequest* attribute), 355  
pipelines (*gitlab.v4.objects.ProjectPackage* attribute), 377  
pipelines (*gitlab.v4.objects.ProjectPipelineSchedule* attribute), 383  
pipelineschedules (*gitlab.v4.objects.Project* attribute), 280  
play() (*gitlab.v4.objects.ProjectJob* method), 336  
play() (*gitlab.v4.objects.ProjectPipelineSchedule* method), 383  
pprint() (*gitlab.base.RESTObject* method), 444  
prev\_page (*gitlab.base.RESTObjectList* property), 445  
prev\_page (*gitlab.GitlabList* property), 197  
PRIVATE (*gitlab.const.Visibility* attribute), 448  
process\_metrics() (*gitlab.v4.objects.SidekiqManager* method), 421  
Project (*class in gitlab.v4.objects*), 276  
PROJECT\_NOTES (*gitlab.const.SearchScope* attribute), 447  
ProjectAccessRequest (*class in gitlab.v4.objects*), 283  
ProjectAccessRequestManager (*class in gitlab.v4.objects*), 284  
ProjectAccessToken (*class in gitlab.v4.objects*), 284  
ProjectAccessTokenManager (*class in gitlab.v4.objects*), 284  
ProjectAdditionalStatistics (*class in gitlab.v4.objects*), 285

- ProjectAdditionalStatisticsManager (class in gitlab.v4.objects), 285
- ProjectApproval (class in gitlab.v4.objects), 286
- ProjectApprovalManager (class in gitlab.v4.objects), 286
- ProjectApprovalRule (class in gitlab.v4.objects), 286
- ProjectApprovalRuleManager (class in gitlab.v4.objects), 287
- ProjectArtifact (class in gitlab.v4.objects), 287
- ProjectArtifactManager (class in gitlab.v4.objects), 287
- ProjectAudit (class in gitlab.v4.objects), 289
- ProjectAuditEvent (class in gitlab.v4.objects), 289
- ProjectAuditEventManager (class in gitlab.v4.objects), 289
- ProjectAuditManager (class in gitlab.v4.objects), 290
- ProjectBadge (class in gitlab.v4.objects), 290
- ProjectBadgeManager (class in gitlab.v4.objects), 290
- ProjectBoard (class in gitlab.v4.objects), 291
- ProjectBoardList (class in gitlab.v4.objects), 291
- ProjectBoardListManager (class in gitlab.v4.objects), 292
- ProjectBoardManager (class in gitlab.v4.objects), 292
- ProjectBranch (class in gitlab.v4.objects), 293
- ProjectBranchManager (class in gitlab.v4.objects), 293
- ProjectCiLint (class in gitlab.v4.objects), 294
- ProjectCiLintManager (class in gitlab.v4.objects), 294
- ProjectCluster (class in gitlab.v4.objects), 295
- ProjectClusterManager (class in gitlab.v4.objects), 295
- ProjectCommit (class in gitlab.v4.objects), 296
- ProjectCommitComment (class in gitlab.v4.objects), 298
- ProjectCommitCommentManager (class in gitlab.v4.objects), 299
- ProjectCommitDiscussion (class in gitlab.v4.objects), 299
- ProjectCommitDiscussionManager (class in gitlab.v4.objects), 299
- ProjectCommitDiscussionNote (class in gitlab.v4.objects), 300
- ProjectCommitDiscussionNoteManager (class in gitlab.v4.objects), 300
- ProjectCommitManager (class in gitlab.v4.objects), 301
- ProjectCommitStatus (class in gitlab.v4.objects), 302
- ProjectCommitStatusManager (class in gitlab.v4.objects), 302
- ProjectCustomAttribute (class in gitlab.v4.objects), 303
- ProjectCustomAttributeManager (class in gitlab.v4.objects), 303
- ProjectDeployment (class in gitlab.v4.objects), 305
- ProjectDeploymentManager (class in gitlab.v4.objects), 305
- ProjectDeploymentMergeRequest (class in gitlab.v4.objects), 306
- ProjectDeploymentMergeRequestManager (class in gitlab.v4.objects), 306
- ProjectDeployToken (class in gitlab.v4.objects), 304
- ProjectDeployTokenManager (class in gitlab.v4.objects), 304
- ProjectEnvironment (class in gitlab.v4.objects), 308
- ProjectEnvironmentManager (class in gitlab.v4.objects), 308
- ProjectEvent (class in gitlab.v4.objects), 309
- ProjectEventManager (class in gitlab.v4.objects), 309
- ProjectExport (class in gitlab.v4.objects), 309
- ProjectExportManager (class in gitlab.v4.objects), 310
- ProjectFile (class in gitlab.v4.objects), 310
- ProjectFileManager (class in gitlab.v4.objects), 311
- ProjectFork (class in gitlab.v4.objects), 314
- ProjectForkManager (class in gitlab.v4.objects), 314
- ProjectHook (class in gitlab.v4.objects), 315
- ProjectHookManager (class in gitlab.v4.objects), 316
- ProjectImport (class in gitlab.v4.objects), 317
- ProjectImportManager (class in gitlab.v4.objects), 317
- ProjectIntegration (class in gitlab.v4.objects), 318
- ProjectIntegrationManager (class in gitlab.v4.objects), 318
- ProjectInvitation (class in gitlab.v4.objects), 319
- ProjectInvitationManager (class in gitlab.v4.objects), 319
- ProjectIssue (class in gitlab.v4.objects), 320
- ProjectIssueAwardEmoji (class in gitlab.v4.objects), 321
- ProjectIssueAwardEmojiManager (class in gitlab.v4.objects), 322
- ProjectIssueDiscussion (class in gitlab.v4.objects), 322
- ProjectIssueDiscussionManager (class in gitlab.v4.objects), 322
- ProjectIssueDiscussionNote (class in gitlab.v4.objects), 323
- ProjectIssueDiscussionNoteManager (class in gitlab.v4.objects), 323
- ProjectIssueLink (class in gitlab.v4.objects), 324
- ProjectIssueLinkManager (class in gitlab.v4.objects), 324
- ProjectIssueManager (class in gitlab.v4.objects), 325
- ProjectIssueNote (class in gitlab.v4.objects), 327
- ProjectIssueNoteAwardEmoji (class in gitlab.v4.objects), 327

- `ProjectIssueNoteAwardEmojiManager` (class in `gitlab.v4.objects`), 327
- `ProjectIssueNoteManager` (class in `gitlab.v4.objects`), 328
- `ProjectIssueResourceIterationEvent` (class in `gitlab.v4.objects`), 329
- `ProjectIssueResourceIterationEventManager` (class in `gitlab.v4.objects`), 329
- `ProjectIssueResourceLabelEvent` (class in `gitlab.v4.objects`), 329
- `ProjectIssueResourceLabelEventManager` (class in `gitlab.v4.objects`), 330
- `ProjectIssueResourceMilestoneEvent` (class in `gitlab.v4.objects`), 330
- `ProjectIssueResourceMilestoneEventManager` (class in `gitlab.v4.objects`), 330
- `ProjectIssueResourceStateEvent` (class in `gitlab.v4.objects`), 331
- `ProjectIssueResourceStateEventManager` (class in `gitlab.v4.objects`), 331
- `ProjectIssueResourceWeightEvent` (class in `gitlab.v4.objects`), 332
- `ProjectIssueResourceWeightEventManager` (class in `gitlab.v4.objects`), 332
- `ProjectIssuesStatistics` (class in `gitlab.v4.objects`), 333
- `ProjectIssuesStatisticsManager` (class in `gitlab.v4.objects`), 333
- `ProjectIterationManager` (class in `gitlab.v4.objects`), 333
- `ProjectJob` (class in `gitlab.v4.objects`), 334
- `ProjectJobManager` (class in `gitlab.v4.objects`), 337
- `ProjectJobTokenScope` (class in `gitlab.v4.objects`), 337
- `ProjectJobTokenScopeManager` (class in `gitlab.v4.objects`), 337
- `ProjectKey` (class in `gitlab.v4.objects`), 338
- `ProjectKeyManager` (class in `gitlab.v4.objects`), 338
- `ProjectLabel` (class in `gitlab.v4.objects`), 339
- `ProjectLabelManager` (class in `gitlab.v4.objects`), 340
- `ProjectManager` (class in `gitlab.v4.objects`), 341
- `ProjectMember` (class in `gitlab.v4.objects`), 350
- `ProjectMemberAll` (class in `gitlab.v4.objects`), 351
- `ProjectMemberAllManager` (class in `gitlab.v4.objects`), 351
- `ProjectMemberManager` (class in `gitlab.v4.objects`), 351
- `ProjectMergeRequest` (class in `gitlab.v4.objects`), 352
- `ProjectMergeRequestApproval` (class in `gitlab.v4.objects`), 356
- `ProjectMergeRequestApprovalManager` (class in `gitlab.v4.objects`), 356
- `ProjectMergeRequestApprovalRule` (class in `gitlab.v4.objects`), 357
- `ProjectMergeRequestApprovalRuleManager` (class in `gitlab.v4.objects`), 358
- `ProjectMergeRequestApprovalState` (class in `gitlab.v4.objects`), 359
- `ProjectMergeRequestApprovalStateManager` (class in `gitlab.v4.objects`), 359
- `ProjectMergeRequestAwardEmoji` (class in `gitlab.v4.objects`), 360
- `ProjectMergeRequestAwardEmojiManager` (class in `gitlab.v4.objects`), 360
- `ProjectMergeRequestDiff` (class in `gitlab.v4.objects`), 361
- `ProjectMergeRequestDiffManager` (class in `gitlab.v4.objects`), 361
- `ProjectMergeRequestDiscussion` (class in `gitlab.v4.objects`), 361
- `ProjectMergeRequestDiscussionManager` (class in `gitlab.v4.objects`), 362
- `ProjectMergeRequestDiscussionNote` (class in `gitlab.v4.objects`), 362
- `ProjectMergeRequestDiscussionNoteManager` (class in `gitlab.v4.objects`), 363
- `ProjectMergeRequestDraftNote` (class in `gitlab.v4.objects`), 363
- `ProjectMergeRequestDraftNoteManager` (class in `gitlab.v4.objects`), 364
- `ProjectMergeRequestManager` (class in `gitlab.v4.objects`), 365
- `ProjectMergeRequestNote` (class in `gitlab.v4.objects`), 367
- `ProjectMergeRequestNoteAwardEmoji` (class in `gitlab.v4.objects`), 367
- `ProjectMergeRequestNoteAwardEmojiManager` (class in `gitlab.v4.objects`), 367
- `ProjectMergeRequestNoteManager` (class in `gitlab.v4.objects`), 368
- `ProjectMergeRequestPipeline` (class in `gitlab.v4.objects`), 369
- `ProjectMergeRequestPipelineManager` (class in `gitlab.v4.objects`), 369
- `ProjectMergeRequestResourceLabelEvent` (class in `gitlab.v4.objects`), 369
- `ProjectMergeRequestResourceLabelEventManager` (class in `gitlab.v4.objects`), 369
- `ProjectMergeRequestResourceMilestoneEvent` (class in `gitlab.v4.objects`), 370
- `ProjectMergeRequestResourceMilestoneEventManager` (class in `gitlab.v4.objects`), 370
- `ProjectMergeRequestResourceStateEvent` (class in `gitlab.v4.objects`), 371
- `ProjectMergeRequestResourceStateEventManager` (class in `gitlab.v4.objects`), 371
- `ProjectMergeRequestReviewerDetail` (class in `gitlab.v4.objects`), 372



- ProjectMergeRequestReviewerDetailManager (class in *gitlab.v4.objects*), 372
- ProjectMergeTrain (class in *gitlab.v4.objects*), 372
- ProjectMergeTrainManager (class in *gitlab.v4.objects*), 372
- ProjectMilestone (class in *gitlab.v4.objects*), 372
- ProjectMilestoneManager (class in *gitlab.v4.objects*), 373
- ProjectNote (class in *gitlab.v4.objects*), 375
- ProjectNoteManager (class in *gitlab.v4.objects*), 375
- ProjectNotificationSettings (class in *gitlab.v4.objects*), 375
- ProjectNotificationSettingsManager (class in *gitlab.v4.objects*), 376
- ProjectPackage (class in *gitlab.v4.objects*), 376
- ProjectPackageFile (class in *gitlab.v4.objects*), 377
- ProjectPackageFileManager (class in *gitlab.v4.objects*), 377
- ProjectPackageManager (class in *gitlab.v4.objects*), 377
- ProjectPackagePipeline (class in *gitlab.v4.objects*), 378
- ProjectPackagePipelineManager (class in *gitlab.v4.objects*), 378
- ProjectPagesDomain (class in *gitlab.v4.objects*), 378
- ProjectPagesDomainManager (class in *gitlab.v4.objects*), 378
- ProjectPipeline (class in *gitlab.v4.objects*), 379
- ProjectPipelineBridge (class in *gitlab.v4.objects*), 380
- ProjectPipelineBridgeManager (class in *gitlab.v4.objects*), 380
- ProjectPipelineJob (class in *gitlab.v4.objects*), 381
- ProjectPipelineJobManager (class in *gitlab.v4.objects*), 381
- ProjectPipelineManager (class in *gitlab.v4.objects*), 381
- ProjectPipelineSchedule (class in *gitlab.v4.objects*), 382
- ProjectPipelineScheduleManager (class in *gitlab.v4.objects*), 383
- ProjectPipelineSchedulePipeline (class in *gitlab.v4.objects*), 384
- ProjectPipelineSchedulePipelineManager (class in *gitlab.v4.objects*), 384
- ProjectPipelineScheduleVariable (class in *gitlab.v4.objects*), 384
- ProjectPipelineScheduleVariableManager (class in *gitlab.v4.objects*), 385
- ProjectPipelineTestReport (class in *gitlab.v4.objects*), 385
- ProjectPipelineTestReportManager (class in *gitlab.v4.objects*), 385
- ProjectPipelineTestReportSummary (class in *gitlab.v4.objects*), 386
- ProjectPipelineTestReportSummaryManager (class in *gitlab.v4.objects*), 386
- ProjectPipelineVariable (class in *gitlab.v4.objects*), 387
- ProjectPipelineVariableManager (class in *gitlab.v4.objects*), 387
- ProjectProtectedBranch (class in *gitlab.v4.objects*), 387
- ProjectProtectedBranchManager (class in *gitlab.v4.objects*), 387
- ProjectProtectedEnvironment (class in *gitlab.v4.objects*), 388
- ProjectProtectedEnvironmentManager (class in *gitlab.v4.objects*), 388
- ProjectProtectedTag (class in *gitlab.v4.objects*), 389
- ProjectProtectedTagManager (class in *gitlab.v4.objects*), 389
- ProjectPushRules (class in *gitlab.v4.objects*), 390
- ProjectPushRulesManager (class in *gitlab.v4.objects*), 390
- ProjectRegistryRepository (class in *gitlab.v4.objects*), 391
- ProjectRegistryRepositoryManager (class in *gitlab.v4.objects*), 391
- ProjectRegistryTag (class in *gitlab.v4.objects*), 392
- ProjectRegistryTagManager (class in *gitlab.v4.objects*), 392
- ProjectRelease (class in *gitlab.v4.objects*), 393
- ProjectReleaseLink (class in *gitlab.v4.objects*), 393
- ProjectReleaseLinkManager (class in *gitlab.v4.objects*), 393
- ProjectReleaseManager (class in *gitlab.v4.objects*), 394
- ProjectRemoteMirror (class in *gitlab.v4.objects*), 395
- ProjectRemoteMirrorManager (class in *gitlab.v4.objects*), 395
- ProjectResourceGroup (class in *gitlab.v4.objects*), 396
- ProjectResourceGroupManager (class in *gitlab.v4.objects*), 396
- ProjectResourceGroupUpcomingJob (class in *gitlab.v4.objects*), 397
- ProjectResourceGroupUpcomingJobManager (class in *gitlab.v4.objects*), 397
- ProjectRunner (class in *gitlab.v4.objects*), 397
- ProjectRunnerManager (class in *gitlab.v4.objects*), 398
- PROJECTS (*gitlab.const.SearchScope* attribute), 447
- projects (*gitlab.Gitlab* attribute), 195
- projects (*gitlab.v4.objects.Group* attribute), 223
- projects (*gitlab.v4.objects.User* attribute), 429
- ProjectSecureFile (class in *gitlab.v4.objects*), 398
- ProjectSecureFileManager (class in *git-*

*lab.v4.objects*), 399

*ProjectService* (class in *gitlab.v4.objects*), 399

*ProjectServiceManager* (class in *gitlab.v4.objects*), 399

*ProjectSnippet* (class in *gitlab.v4.objects*), 400

*ProjectSnippetAwardEmoji* (class in *gitlab.v4.objects*), 401

*ProjectSnippetAwardEmojiManager* (class in *gitlab.v4.objects*), 401

*ProjectSnippetDiscussion* (class in *gitlab.v4.objects*), 401

*ProjectSnippetDiscussionManager* (class in *gitlab.v4.objects*), 402

*ProjectSnippetDiscussionNote* (class in *gitlab.v4.objects*), 402

*ProjectSnippetDiscussionNoteManager* (class in *gitlab.v4.objects*), 403

*ProjectSnippetManager* (class in *gitlab.v4.objects*), 403

*ProjectSnippetNote* (class in *gitlab.v4.objects*), 404

*ProjectSnippetNoteAwardEmoji* (class in *gitlab.v4.objects*), 405

*ProjectSnippetNoteAwardEmojiManager* (class in *gitlab.v4.objects*), 405

*ProjectSnippetNoteManager* (class in *gitlab.v4.objects*), 405

*ProjectStorage* (class in *gitlab.v4.objects*), 406

*ProjectStorageManager* (class in *gitlab.v4.objects*), 406

*ProjectTag* (class in *gitlab.v4.objects*), 407

*ProjectTagManager* (class in *gitlab.v4.objects*), 407

*ProjectTrigger* (class in *gitlab.v4.objects*), 408

*ProjectTriggerManager* (class in *gitlab.v4.objects*), 408

*ProjectUser* (class in *gitlab.v4.objects*), 409

*ProjectUserManager* (class in *gitlab.v4.objects*), 409

*ProjectVariable* (class in *gitlab.v4.objects*), 409

*ProjectVariableManager* (class in *gitlab.v4.objects*), 409

*ProjectWiki* (class in *gitlab.v4.objects*), 410

*ProjectWikiManager* (class in *gitlab.v4.objects*), 411

*protected\_environments* (*gitlab.v4.objects.Project* attribute), 280

*protectedbranches* (*gitlab.v4.objects.Project* attribute), 280

*protectedtags* (*gitlab.v4.objects.Project* attribute), 280

*PUBLIC* (*gitlab.const.Visibility* attribute), 448

*public()* (*gitlab.v4.objects.SnippetManager* method), 423

*publish()* (*gitlab.v4.objects.ProjectMergeRequestDraftNote* method), 364

*pushrules* (*gitlab.v4.objects.Group* attribute), 223

*pushrules* (*gitlab.v4.objects.Project* attribute), 280

## Q

*queue\_metrics()* (*gitlab.v4.objects.SidekiqManager* method), 421

## R

*raw()* (*gitlab.v4.objects.ProjectArtifactManager* method), 288

*raw()* (*gitlab.v4.objects.ProjectFileManager* method), 313

*rebase()* (*gitlab.v4.objects.ProjectMergeRequest* method), 355

*RedirectError*, 465

*refresh()* (*gitlab.mixins.RefreshMixin* method), 469

*RefreshMixin* (class in *gitlab.mixins*), 469

*refs()* (*gitlab.v4.objects.ProjectCommit* method), 297

*register\_custom\_action()* (in module *gitlab.cli*), 446

*registry\_repositories* (*gitlab.Gitlab* attribute), 195

*registry\_repositories* (*gitlab.v4.objects.Group* attribute), 223

*RegistryRepository* (class in *gitlab.v4.objects*), 411

*RegistryRepositoryManager* (class in *gitlab.v4.objects*), 412

*reject()* (*gitlab.v4.objects.User* method), 429

*related\_merge\_requests()* (*gitlab.v4.objects.ProjectIssue* method), 321

*releases* (*gitlab.v4.objects.Project* attribute), 280

*remote\_import()* (*gitlab.v4.objects.ProjectManager* method), 349

*remote\_import\_s3()* (*gitlab.v4.objects.ProjectManager* method), 350

*remote\_mirrors* (*gitlab.v4.objects.Project* attribute), 280

*remove\_none\_from\_dict()* (in module *gitlab.utils*), 475

*render()* (*gitlab.mixins.BadgeRenderMixin* method), 465

*reorder()* (*gitlab.v4.objects.ProjectIssue* method), 321

*repair()* (*gitlab.v4.objects.GeoNode* method), 218

*REPORTER* (*gitlab.const.AccessLevel* attribute), 447

*repositories* (*gitlab.v4.objects.Project* attribute), 280

*repository\_archive()* (*gitlab.v4.objects.RepositoryMixin* method), 412

*repository\_blob()* (*gitlab.v4.objects.RepositoryMixin* method), 413

*repository\_compare()* (*gitlab.v4.objects.RepositoryMixin* method), 413

*repository\_contributors()* (*gitlab.v4.objects.RepositoryMixin* method), 414

- `repository_merge_base()` (*gitlab.v4.objects.RepositoryMixin* method), 414
  - `repository_raw_blob()` (*gitlab.v4.objects.RepositoryMixin* method), 414
  - `repository_tree()` (*gitlab.v4.objects.RepositoryMixin* method), 415
  - `RepositoryMixin` (class in *gitlab.v4.objects*), 412
  - `reset_approvals()` (*gitlab.v4.objects.ProjectMergeRequest* method), 355
  - `reset_spent_time()` (*gitlab.mixins.TimeTrackingMixin* method), 472
  - `reset_time_estimate()` (*gitlab.mixins.TimeTrackingMixin* method), 472
  - `resource_groups` (*gitlab.v4.objects.Project* attribute), 280
  - `resource_iteration_events` (*gitlab.v4.objects.ProjectIssue* attribute), 321
  - `resource_weight_events` (*gitlab.v4.objects.ProjectIssue* attribute), 321
  - `resourcelabelevents` (*gitlab.v4.objects.GroupEpic* attribute), 235
  - `resourcelabelevents` (*gitlab.v4.objects.ProjectIssue* attribute), 321
  - `resourcelabelevents` (*gitlab.v4.objects.ProjectMergeRequest* attribute), 355
  - `resourcemilestoneevents` (*gitlab.v4.objects.ProjectIssue* attribute), 321
  - `resourcemilestoneevents` (*gitlab.v4.objects.ProjectMergeRequest* attribute), 355
  - `resourcestateevents` (*gitlab.v4.objects.ProjectIssue* attribute), 321
  - `resourcestateevents` (*gitlab.v4.objects.ProjectMergeRequest* attribute), 355
  - `response_content()` (in module *gitlab.utils*), 475
  - `RESTManager` (class in *gitlab.base*), 443
  - `RESTObject` (class in *gitlab.base*), 443
  - `RESTObjectList` (class in *gitlab.base*), 444
  - `restore()` (*gitlab.v4.objects.Group* method), 223
  - `restore()` (*gitlab.v4.objects.Project* method), 280
  - `RetrieveMixin` (class in *gitlab.mixins*), 470
  - `retry()` (*gitlab.v4.objects.ProjectJob* method), 336
  - `retry()` (*gitlab.v4.objects.ProjectPipeline* method), 380
  - `revert()` (*gitlab.v4.objects.ProjectCommit* method), 298
  - `reviewer_details` (*gitlab.v4.objects.ProjectMergeRequest* attribute), 355
  - `Runner` (class in *gitlab.v4.objects*), 416
  - `RunnerAll` (class in *gitlab.v4.objects*), 416
  - `RunnerAllManager` (class in *gitlab.v4.objects*), 416
  - `RunnerJob` (class in *gitlab.v4.objects*), 417
  - `RunnerJobManager` (class in *gitlab.v4.objects*), 417
  - `RunnerManager` (class in *gitlab.v4.objects*), 417
  - `runners` (*gitlab.Gitlab* attribute), 195
  - `runners` (*gitlab.v4.objects.CurrentUser* attribute), 209
  - `runners` (*gitlab.v4.objects.Group* attribute), 223
  - `runners` (*gitlab.v4.objects.Project* attribute), 280
  - `runners_all` (*gitlab.Gitlab* attribute), 195
- ## S
- `saml_group_links` (*gitlab.v4.objects.Group* attribute), 223
  - `save()` (*gitlab.mixins.SaveMixin* method), 470
  - `save()` (*gitlab.v4.objects.GroupEpicIssue* method), 237
  - `save()` (*gitlab.v4.objects.GroupLabel* method), 250
  - `save()` (*gitlab.v4.objects.ProjectFile* method), 311
  - `save()` (*gitlab.v4.objects.ProjectLabel* method), 339
  - `save()` (*gitlab.v4.objects.ProjectMergeRequestApprovalRule* method), 357
  - `SaveMixin` (class in *gitlab.mixins*), 470
  - `search()` (*gitlab.Gitlab* method), 195
  - `search()` (*gitlab.v4.objects.Group* method), 223
  - `search()` (*gitlab.v4.objects.Project* method), 280
  - `SearchScope` (class in *gitlab.const*), 447
  - `secure_files` (*gitlab.v4.objects.Project* attribute), 281
  - `services` (*gitlab.v4.objects.Project* attribute), 281
  - `set()` (*gitlab.mixins.SetMixin* method), 470
  - `set()` (*gitlab.v4.objects.FeatureManager* method), 216
  - `set_approvers()` (*gitlab.v4.objects.ProjectMergeRequestApprovalManager* method), 357
  - `set_license()` (*gitlab.Gitlab* method), 196
  - `SetMixin` (class in *gitlab.mixins*), 470
  - `settings` (*gitlab.Gitlab* attribute), 196
  - `share()` (*gitlab.v4.objects.Group* method), 224
  - `share()` (*gitlab.v4.objects.Project* method), 281
  - `shared_projects` (*gitlab.v4.objects.Group* attribute), 224
  - `SharedProject` (class in *gitlab.v4.objects*), 419
  - `SharedProjectManager` (class in *gitlab.v4.objects*), 419
  - `sidekiq` (*gitlab.Gitlab* attribute), 196
  - `SidekiqManager` (class in *gitlab.v4.objects*), 420
  - `signature()` (*gitlab.v4.objects.ProjectCommit* method), 298
  - `snapshot()` (*gitlab.v4.objects.Project* method), 281
  - `Snippet` (class in *gitlab.v4.objects*), 421
  - `SnippetManager` (class in *gitlab.v4.objects*), 422
  - `snippets` (*gitlab.Gitlab* attribute), 196
  - `snippets` (*gitlab.v4.objects.Project* attribute), 282

`ssl_verify` (*gitlab.Gitlab attribute*), 196  
`star()` (*gitlab.v4.objects.Project method*), 282  
`starred_projects` (*gitlab.v4.objects.User attribute*), 429  
`StarredProject` (*class in gitlab.v4.objects*), 423  
`StarredProjectManager` (*class in gitlab.v4.objects*), 423  
`statistics` (*gitlab.Gitlab attribute*), 196  
`status` (*gitlab.v4.objects.CurrentUser attribute*), 209  
`status` (*gitlab.v4.objects.User attribute*), 429  
`status()` (*gitlab.v4.objects.GeoNode method*), 219  
`status()` (*gitlab.v4.objects.GeoNodeManager method*), 220  
`statuses` (*gitlab.v4.objects.ProjectCommit attribute*), 298  
`stop()` (*gitlab.v4.objects.ProjectEnvironment method*), 308  
`storage` (*gitlab.v4.objects.Project attribute*), 282  
`subgroups` (*gitlab.v4.objects.Group attribute*), 224  
`SubscribableMixin` (*class in gitlab.mixins*), 471  
`subscribe()` (*gitlab.mixins.SubscribableMixin method*), 471

## T

`tags` (*gitlab.v4.objects.Project attribute*), 282  
`tags` (*gitlab.v4.objects.ProjectRegistryRepository attribute*), 391  
`take_ownership()` (*gitlab.v4.objects.ProjectPipelineSchedule method*), 383  
`test_report` (*gitlab.v4.objects.ProjectPipeline attribute*), 380  
`test_report_summary` (*gitlab.v4.objects.ProjectPipeline attribute*), 380  
`time_estimate()` (*gitlab.mixins.TimeTrackingMixin method*), 472  
`time_stats()` (*gitlab.mixins.TimeTrackingMixin method*), 472  
`timeout` (*gitlab.Gitlab attribute*), 196  
`TimeTrackingMixin` (*class in gitlab.mixins*), 471  
`to_json()` (*gitlab.base.RESTObject method*), 444  
`Todo` (*class in gitlab.v4.objects*), 424  
`todo()` (*gitlab.mixins.TODOMixin method*), 473  
`TodoManager` (*class in gitlab.v4.objects*), 424  
`TODOMixin` (*class in gitlab.mixins*), 472  
`todos` (*gitlab.Gitlab attribute*), 196  
`Topic` (*class in gitlab.v4.objects*), 425  
`TopicManager` (*class in gitlab.v4.objects*), 425  
`topics` (*gitlab.Gitlab attribute*), 196  
`total` (*gitlab.base.RESTObjectList property*), 445  
`total` (*gitlab.GitlabList property*), 198  
`total_pages` (*gitlab.base.RESTObjectList property*), 445

`total_pages` (*gitlab.GitlabList property*), 198  
`trace()` (*gitlab.v4.objects.ProjectJob method*), 336  
`transfer()` (*gitlab.v4.objects.Group method*), 224  
`transfer()` (*gitlab.v4.objects.Project method*), 282  
`transfer_project()` (*gitlab.v4.objects.Group method*), 224  
`trigger_pipeline()` (*gitlab.v4.objects.Project method*), 282  
`triggers` (*gitlab.v4.objects.Project attribute*), 283

## U

`unapprove()` (*gitlab.v4.objects.ProjectMergeRequest method*), 356  
`unarchive()` (*gitlab.v4.objects.Project method*), 283  
`unban()` (*gitlab.v4.objects.User method*), 429  
`unblock()` (*gitlab.v4.objects.User method*), 429  
`unfollow()` (*gitlab.v4.objects.User method*), 430  
`unshare()` (*gitlab.v4.objects.Group method*), 225  
`unshare()` (*gitlab.v4.objects.Project method*), 283  
`unstar()` (*gitlab.v4.objects.Project method*), 283  
`unsubscribe()` (*gitlab.mixins.SubscribableMixin method*), 471  
`upcoming_jobs` (*gitlab.v4.objects.ProjectResourceGroup attribute*), 396  
`update()` (*gitlab.mixins.UpdateMixin method*), 473  
`update()` (*gitlab.v4.objects.ApplicationAppearanceManager method*), 199  
`update()` (*gitlab.v4.objects.ApplicationSettingsManager method*), 202  
`update()` (*gitlab.v4.objects.GroupLabelManager method*), 251  
`update()` (*gitlab.v4.objects.ProjectFileManager method*), 314  
`update()` (*gitlab.v4.objects.ProjectLabelManager method*), 341  
`update_submodule()` (*gitlab.v4.objects.RepositoryMixin method*), 415  
`UpdateMixin` (*class in gitlab.mixins*), 473  
`upload()` (*gitlab.v4.objects.GenericPackageManager method*), 217  
`url` (*gitlab.Gitlab property*), 196  
`User` (*class in gitlab.v4.objects*), 426  
`user` (*gitlab.Gitlab attribute*), 197  
`user_activities` (*gitlab.Gitlab attribute*), 197  
`user_agent_detail()` (*gitlab.mixins.UserAgentDetailMixin method*), 473  
`UserActivities` (*class in gitlab.v4.objects*), 430  
`UserActivitiesManager` (*class in gitlab.v4.objects*), 430  
`UserAgentDetailMixin` (*class in gitlab.mixins*), 473  
`UserCustomAttribute` (*class in gitlab.v4.objects*), 430



UserCustomAttributeManager (class in *gitlab.v4.objects*), 431  
 UserEmail (class in *gitlab.v4.objects*), 431  
 UserEmailManager (class in *gitlab.v4.objects*), 431  
 UserEvent (class in *gitlab.v4.objects*), 432  
 UserEventManager (class in *gitlab.v4.objects*), 432  
 UserGPGKey (class in *gitlab.v4.objects*), 432  
 UserGPGKeyManager (class in *gitlab.v4.objects*), 433  
 UserIdentityProviderManager (class in *gitlab.v4.objects*), 433  
 UserImpersonationToken (class in *gitlab.v4.objects*), 434  
 UserImpersonationTokenManager (class in *gitlab.v4.objects*), 434  
 UserKey (class in *gitlab.v4.objects*), 435  
 UserKeyManager (class in *gitlab.v4.objects*), 435  
 UserManager (class in *gitlab.v4.objects*), 435  
 UserMembership (class in *gitlab.v4.objects*), 438  
 UserMembershipManager (class in *gitlab.v4.objects*), 438  
 UserPersonalAccessToken (class in *gitlab.v4.objects*), 438  
 UserPersonalAccessTokenManager (class in *gitlab.v4.objects*), 439  
 UserProject (class in *gitlab.v4.objects*), 439  
 UserProjectManager (class in *gitlab.v4.objects*), 439  
 USERS (*gitlab.const.SearchScope* attribute), 447  
 users (*gitlab.Gitlab* attribute), 197  
 users (*gitlab.v4.objects.Project* attribute), 283  
 UserStatus (class in *gitlab.v4.objects*), 441  
 UserStatusManager (class in *gitlab.v4.objects*), 441

## V

validate() (*gitlab.v4.objects.CiLintManager* method), 208  
 validate() (*gitlab.v4.objects.ProjectCiLintManager* method), 295  
 Variable (class in *gitlab.v4.objects*), 442  
 VariableManager (class in *gitlab.v4.objects*), 442  
 variables (*gitlab.Gitlab* attribute), 197  
 variables (*gitlab.v4.objects.Group* attribute), 225  
 variables (*gitlab.v4.objects.Project* attribute), 283  
 variables (*gitlab.v4.objects.ProjectPipeline* attribute), 380  
 variables (*gitlab.v4.objects.ProjectPipelineSchedule* attribute), 383  
 verify() (*gitlab.v4.objects.RunnerManager* method), 419  
 version() (*gitlab.Gitlab* method), 197  
 VerticalHelpFormatter (class in *gitlab.cli*), 445  
 Visibility (class in *gitlab.const*), 448

## W

warn() (in module *gitlab.utils*), 475