

---

# **python-gitlab Documentation**

***Release 3.9.0***

**Gauvain Pocentek, Mika Mäenpää**

**Aug 28, 2022**



## TABLE OF CONTENTS

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Using the docker images</b>	<b>5</b>
2.1	Building the image . . . . .	5
<b>3</b>	<b>Bug reports</b>	<b>7</b>
<b>4</b>	<b>Gitter Community Chat</b>	<b>9</b>
<b>5</b>	<b>Documentation</b>	<b>11</b>
5.1	Build the docs . . . . .	11
<b>6</b>	<b>Contributing</b>	<b>13</b>
6.1	Getting started with the CLI . . . . .	13
6.2	Getting started with the API . . . . .	18
6.3	Advanced usage . . . . .	25
6.4	CLI examples . . . . .	27
6.5	API examples . . . . .	32
6.6	API reference (gitlab package) . . . . .	146
6.7	CLI reference (gitlab command) . . . . .	454
6.8	Changelog . . . . .	781
6.9	Release notes . . . . .	818
6.10	FAQ . . . . .	822
<b>7</b>	<b>Indices and tables</b>	<b>823</b>
	<b>Python Module Index</b>	<b>825</b>
	<b>Index</b>	<b>827</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`).



## INSTALLATION

As of 3.0.0, `python-gitlab` is compatible with Python 3.7+.

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> ...
```

### 2.1 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](#).

### 5.1 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](#).

### 6.1 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.

#### 6.1.1 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
<code>ssl_verify</code>	<code>True</code> , <code>False</code> , or a <code>str</code>	Verify the SSL certificate. Set to <code>False</code> to disable verification, though this will create warnings. Any other value is interpreted as path to a <code>CA_BUNDLE</code> file or directory with certificates of trusted CAs.
<code>timeout</code>	Integer	Number of seconds to wait for an answer before failing.
<code>api_version</code>	Integer	The API version to use to make queries. Only 4 is available since 1.5.0.
<code>per_page</code>	Integer between 1 and 100	The number of items to return in listing queries. GitLab limits the value at 100.
<code>user_agent</code>	<code>str</code>	A string defining a custom user agent to use when <code>gitlab</code> 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
<code>url</code>	URL for the GitLab server. Do <b>NOT</b> use a URL which redirects.
<code>private_token</code>	Your user token. Login/password is not supported. Refer to <a href="#">the official documentation</a> to learn how to obtain a token.
<code>oauth_token</code>	An OAuth token for authentication. The Gitlab server must be configured to support this authentication method.
<code>job_token</code>	Your job token. See <a href="#">the official documentation</a> to learn how to obtain a token.
<code>api_version</code>	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://somewe.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://somewe.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
```

## 6.1.2 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
```

### 6.1.3 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)"
```

## 6.2 Getting started with the API

python-gitlab only supports GitLab API v4.

### 6.2.1 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)
```

(continues on next page)

(continued from previous page)

```

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()

```

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](#).

## 6.2.2 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
```

### 6.2.3 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)

# 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()
```

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.

---

**Note:** `attributes` returns the parent object attributes that are defined in `object._from_parent_attrs`. What this can mean is that for example a `ProjectIssue` object will have a `project_id` key in the dictionary returned from `attributes` but `asdict()` will not.

---

```
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
```

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))
```

## 6.2.4 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.

### 6.2.5 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
```

### 6.2.6 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"])
```

### 6.2.7 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`.

---

### 6.2.8 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')
```

## 6.2.9 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.

## 6.3 Advanced usage

### 6.3.1 Using a custom session

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

You can provide your own `Session` object with custom configuration when you create a `Gitlab` object.

### 6.3.2 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.

### 6.3.3 Proxy configuration

The following sample illustrates how to define a proxy configuration when using python-gitlab:

```
import os
import gitlab
import requests

session = requests.Session()
session.proxies = {
    'https': os.environ.get('https_proxy'),
    'http': os.environ.get('http_proxy'),
}
gl = gitlab.gitlab(url, token, api_version=4, session=session)
```

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

### 6.3.4 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>

### 6.3.5 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>

### 6.3.6 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.

### 6.3.7 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), and 504 (Gateway Timeout) are retried. 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
```

### 6.3.8 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)
```

## 6.4 CLI examples

### See also:

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

### 6.4.1 CI Lint

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
```

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
```

## 6.4.2 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
```

### 6.4.3 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
```

### 6.4.4 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
```

### 6.4.5 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"
↪"
```

## 6.4.6 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
```

## 6.4.7 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
```

### 6.4.8 Issues

Get a list of issues for this project:

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

### 6.4.9 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"
```

### 6.4.10 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
```

### 6.4.11 Artifacts

Download the artifacts zip archive of a job:

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

### 6.4.12 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
```

### 6.4.13 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
```

## 6.5 API examples

### 6.5.1 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()
```

## 6.5.2 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()
```

## 6.5.3 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()
```

## 6.5.4 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)
```

## 6.5.5 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_link = new_link  
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'])
```

## 6.5.6 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*.

## 6.5.7 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()
```

## 6.5.8 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})
```

(continues on next page)

(continued from previous page)

```
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"})
```

## 6.5.9 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').read()),
            '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',
                                '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'})
```

## 6.5.10 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)
```

### 6.5.11 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)
```

## 6.5.12 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()
```

## 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()
```

### 6.5.13 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.diff.get(diff_id)
mr.discussions.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()
```

## 6.5.14 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*.

## 6.5.15 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)
```

## 6.5.16 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()
```

## 6.5.17 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()
```

## 6.5.18 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()
```

## 6.5.19 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'})
```

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()
```

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:

```
group.add_ldap_group_link(ldap_group_cn, gitlab.const.AccessLevel.DEVELOPER, 'ldapmain')
```

Remove a link:

```
group.delete_ldap_group_link(ldap_group_cn, 'ldapmain')
```

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')
```

## 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 (returns None if there are no 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()
```

## 6.5.20 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)
```

Create group access token:

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

Revoke a group access tokens:

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

## 6.5.21 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")
```

## 6.5.22 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)
# pr
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()
```

## 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) (EE feature)



## 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')
```

## 6.5.23 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()
```

## 6.5.24 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/EyfywfxY")
```

## 6.5.25 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()
```

## 6.5.26 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)
```

## 6.5.27 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()
```



## 6.5.28 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")
```

## 6.5.29 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_id)
```

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_id)
# or
mr.delete()
```

Accept a MR:

```
mr.merge()
```

Cancel a MR when the build succeeds:

```
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()
```

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()
```

## 6.5.30 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()
```

Change project-level MR allowed approvers:

```
project.approvals.set_approvers(approver_ids=[105],
                                approver_group_ids=[653, 654])
```

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],
})
```

## 6.5.31 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)
```

## 6.5.32 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)
```

## 6.5.33 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()
```

## 6.5.34 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()
```

## 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.

---

## 6.5.35 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')
```

## 6.5.36 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)
```

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")
```

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 API first to revoke it.

As of 14.2, GitLab does not provide a GET API for single personal access tokens. You must use the list method to retrieve single tokens.

---

## 6.5.37 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 schedule

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.Project.pipelineschedules`
- 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 = projects.pipelineschedules.get(schedule_id)
```

Create a new schedule:

```
sched = project.pipelineschedules.create({
    'ref': 'main',
    'description': 'Daily test',
    'cron': '@ 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()
```

## 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')
```

**Attention:** An older method `project.artifacts()` is deprecated and will be removed in a future version.

**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')
```

**Attention:** An older method `project.artifact()` is deprecated and will be removed in a future version.

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 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()
```

## 6.5.38 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()
```

## 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'
    )
```

(continues on next page)

(continued from previous page)

```
# 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'},
    )
```

## 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',
                                   'file_name': 'foo.py',
                                   'code': '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 (returns None if there are no 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
```

## 6.5.39 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)
```

Create project access token:

```
access_token = gl.projects.get(1).access_tokens.create({"name": "test", "scopes": ["api", "read_repository"]})
```

Revoke a project access tokens:

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

## 6.5.40 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')
```

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()
```

## 6.5.41 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()
```

## 6.5.42 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()
```

### 6.5.43 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 `list()` and `all()` methods to list runners.

Both methods accept a `scope` parameter to filter the list. Allowed values for this parameter are:

- `active`
- `paused`
- `online`
- `specific` (`all()` only)
- `shared` (`all()` 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})
```

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')
```

## 6.5.44 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()
```

## 6.5.45 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()
```

## 6.5.46 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/ce/api/container\\_registry.html#delete-repository-tags-in-bulk](https://docs.gitlab.com/ce/api/container_registry.html#delete-repository-tags-in-bulk)

---



## 6.5.47 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()
```

## 6.5.48 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()
```

## 6.5.49 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',
                              'file_name': 'snippet1.py',
                              'content': open('snippet1.py').read()})
```

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()
```

## 6.5.50 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()
```

## 6.5.51 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:

```
gitlabciymls = gl.gitlabciymls.list()
```

Get a GitLab CI template:

```
gitlabciyaml = gl.gitlabciymls.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)
```

## 6.5.52 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()
```

## 6.5.53 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)
```

## 6.5.54 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/ce/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,
)
```



### 6.5.55 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()
```

### 6.5.56 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()
```

### 6.5.57 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()
```

## 6.5.58 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()
```

## 6.6 API reference (gitlab package)

### 6.6.1 Module contents

Wrapper for the GitLab API.

```
class gitlab.Gitlab(url: Optional[str] = None, private_token: Optional[str] = None, oauth_token:
    Optional[str] = None, job_token: Optional[str] = None, ssl_verify: Union[bool, str] =
    True, http_username: Optional[str] = None, http_password: Optional[str] = None,
    timeout: Optional[float] = None, api_version: str = '4', session: Optional[Session] =
    None, per_page: Optional[int] = None, pagination: Optional[str] = None, order_by:
    Optional[str] = None, user_agent: str = 'python-gitlab/3.9.0', retry_transient_errors: bool
    = False, keep_base_url: bool = False)
```

Bases: object

Represents a GitLab server connection.

#### Parameters

- **url** (*Optional[str]*) – The URL of the GitLab server (defaults to <https://gitlab.com>).
- **private\_token** (*Optional[str]*) – The user private token
- **oauth\_token** (*Optional[str]*) – An oauth token
- **job\_token** (*Optional[str]*) – A CI job token
- **ssl\_verify** (*Union[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** (*Optional[float]*) – Timeout to use for requests to the GitLab server.
- **http\_username** (*Optional[str]*) – Username for HTTP authentication
- **http\_password** (*Optional[str]*) – Password for HTTP authentication
- **api\_version** (*str*) – Gitlab API version to use (support for 4 only)
- **pagination** (*Optional[str]*) – Can be set to 'keyset' to use keyset pagination
- **order\_by** (*Optional[str]*) – 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** (*Optional[Session]*) –
- **per\_page** (*Optional[int]*) –

**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](#)

**ci\_lint**

See [CiLintManager](#)

**deploykeys**

See [DeployKeyManager](#)

**deploytokens**

See [DeployTokenManager](#)

**dockerfiles**

See [DockerfileManager](#)

**static enable\_debug()** → None

**Return type**

None

**events**

See [EventManager](#)

**features**

See [FeatureManager](#)

**classmethod from\_config**(gitlab\_id: Optional[str] = None, config\_files: Optional[List[str]] = None) → [Gitlab](#)

Create a Gitlab connection from configuration files.

**Parameters**

- **gitlab\_id** (Optional[str]) – ID of the configuration section.
- **list[str]** (*config\_files*) – List of paths to configuration files.
- **config\_files** (Optional[List[str]]) –

**Returns**

A Gitlab connection.

**Raises**

[`gitlab.config.GitlabDataError`](#) – If the configuration is not correct.

**Return type**

[`Gitlab`](#)

**geonodes**

See [`GeoNodeManager`](#)

**get\_license**(*\*\*kwargs: Any*) → Dict[str, Any]

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, Any]*

**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*: Optional[Dict[str, Any]] = None, *streamed*: bool = False, *raw*: bool = False, *\*\*kwargs*: Any) → Union[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** (Optional[Dict[str, Any]]) – 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

Union[Dict[str, Any], Response]

**http\_head**(*path*: str, *query\_data*: Optional[Dict[str, Any]] = None, *\*\*kwargs*: Any) → requests.structures.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** (Optional[Dict[str, Any]]) – 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

requests.structures.CaseInsensitiveDict[Any]

**http\_list**(*path*: str, *query\_data*: Optional[Dict[str, Any]] = None, *\**, *as\_list*: Optional[bool] = None, *iterator*: Optional[bool] = None, *\*\*kwargs*: Any) → Union[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** (Optional[Dict[str, Any]]) – Data to send as query parameters
- **iterator** (Optional[bool]) – Indicate if should return a generator (True)
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo, page, per\_page)
- **as\_list** (Optional[bool]) –

**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**

`Union[GitlabList, List[Dict[str, Any]]]`

**http\_post**(*path*: str, *query\_data*: Optional[Dict[str, Any]] = None, *post\_data*: Optional[Dict[str, Any]] = None, *raw*: bool = False, *files*: Optional[Dict[str, Any]] = None, *\*\*kwargs*: Any) → Union[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** (Optional[Dict[str, Any]]) – Data to send as query parameters
- **post\_data** (Optional[Dict[str, Any]]) – 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** (Optional[Dict[str, Any]]) – 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**

`Union[Dict[str, Any], Response]`

**http\_put**(*path*: str, *query\_data*: Optional[Dict[str, Any]] = None, *post\_data*: Optional[Union[Dict[str, Any], bytes]] = None, *raw*: bool = False, *files*: Optional[Dict[str, Any]] = None, *\*\*kwargs*: Any) → Union[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** (Optional[Dict[str, Any]]) – Data to send as query parameters
- **post\_data** (Optional[Union[Dict[str, Any], bytes]]) – 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** (Optional[Dict[str, Any]]) – 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**

*Union[Dict[str, Any], Response]*

**http\_request**(*verb: str, path: str, query\_data: Optional[Dict[str, Any]] = None, post\_data: Optional[Union[Dict[str, Any], bytes]] = None, raw: bool = False, streamed: bool = False, files: Optional[Dict[str, Any]] = None, timeout: Optional[float] = None, obey\_rate\_limit: bool = True, retry\_transient\_errors: Optional[bool] = 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** (*Optional[Dict[str, Any]]*) – Data to send as query parameters
- **post\_data** (*Optional[Union[Dict[str, Any], bytes]]*) – 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** (*Optional[Dict[str, Any]]*) – The files to send to the server
- **timeout** (*Optional[float]*) – 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** (*Optional[bool]*) – 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](#)

**lint**(*content: str, \*\*kwargs: Any*) → Tuple[bool, List[str]]

Validate a gitlab CI configuration.

**Parameters**

- **content** (*str*) – The .gitlab-ci.yml content
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabVerifyError** – If the validation could not be done

**Returns**

(True, []) if the file is valid, (False, errors(list)) otherwise

**Return type**

Tuple[bool, List[str]]

**markdown**(*text: str, gfm: bool = False, project: Optional[str] = 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** (*Optional[str]*) – 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: Optional[str] = None, config\_files: Optional[List[str]] = 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** (*Optional[str]*) – ID of the configuration section.
- **config\_files** (*Optional[List[str]]*) – 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*) → Union[*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**

*Union[GitlabList, List[Dict[str, Any]]]*

**session**

Create a session object for requests

**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

**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: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

**exception** gitlab.GitlabAttachFileError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

**exception** gitlab.GitlabAuthenticationError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

**exception** `gitlab.GitlabBanError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = *None*, *response\_body*: *Optional[bytes]* = *None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

*None*

**exception** `gitlab.GitlabBlockError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = *None*, *response\_body*: *Optional[bytes]* = *None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

*None*

**exception** `gitlab.GitlabBuildCancelError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = *None*, *response\_body*: *Optional[bytes]* = *None*)

Bases: [`GitlabCancelError`](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

*None*

**exception** `gitlab.GitlabBuildEraseError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = *None*, *response\_body*: *Optional[bytes]* = *None*)

Bases: [`GitlabRetryError`](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

*None*

**exception** `gitlab.GitlabBuildPlayError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = *None*, *response\_body*: *Optional[bytes]* = *None*)

Bases: [`GitlabRetryError`](#)

**Parameters**



- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabBuildRetryError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabRetryError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabCancelError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabCherryPickError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabCiLintError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabConnectionError`(*error\_message*: `Union[str, bytes]` = "", *response\_code*: `Optional[int]` = None, *response\_body*: `Optional[bytes]` = None)

Bases: [`GitlabError`](#)**Parameters**

- **error\_message** (`Union[str, bytes]`) –
- **response\_code** (`Optional[int]`) –
- **response\_body** (`Optional[bytes]`) –

**Return type**

None

**exception** `gitlab.GitlabCreateError`(*error\_message*: `Union[str, bytes]` = "", *response\_code*: `Optional[int]` = None, *response\_body*: `Optional[bytes]` = None)

Bases: [`GitlabOperationError`](#)**Parameters**

- **error\_message** (`Union[str, bytes]`) –
- **response\_code** (`Optional[int]`) –
- **response\_body** (`Optional[bytes]`) –

**Return type**

None

**exception** `gitlab.GitlabDeactivateError`(*error\_message*: `Union[str, bytes]` = "", *response\_code*: `Optional[int]` = None, *response\_body*: `Optional[bytes]` = None)

Bases: [`GitlabOperationError`](#)**Parameters**

- **error\_message** (`Union[str, bytes]`) –
- **response\_code** (`Optional[int]`) –
- **response\_body** (`Optional[bytes]`) –

**Return type**

None

**exception** `gitlab.GitlabDeleteError`(*error\_message*: `Union[str, bytes]` = "", *response\_code*: `Optional[int]` = None, *response\_body*: `Optional[bytes]` = None)

Bases: [`GitlabOperationError`](#)**Parameters**

- **error\_message** (`Union[str, bytes]`) –
- **response\_code** (`Optional[int]`) –
- **response\_body** (`Optional[bytes]`) –

**Return type**

None

**exception** `gitlab.GitlabError`(*error\_message*: `Union[str, bytes]` = "", *response\_code*: `Optional[int]` = `None`, *response\_body*: `Optional[bytes]` = `None`)

Bases: `Exception`

#### Parameters

- **error\_message** (`Union[str, bytes]`) –
- **response\_code** (`Optional[int]`) –
- **response\_body** (`Optional[bytes]`) –

#### Return type

`None`

**exception** `gitlab.GitlabFollowError`(*error\_message*: `Union[str, bytes]` = "", *response\_code*: `Optional[int]` = `None`, *response\_body*: `Optional[bytes]` = `None`)

Bases: `GitlabOperationError`

#### Parameters

- **error\_message** (`Union[str, bytes]`) –
- **response\_code** (`Optional[int]`) –
- **response\_body** (`Optional[bytes]`) –

#### Return type

`None`

**exception** `gitlab.GitlabGetError`(*error\_message*: `Union[str, bytes]` = "", *response\_code*: `Optional[int]` = `None`, *response\_body*: `Optional[bytes]` = `None`)

Bases: `GitlabOperationError`

#### Parameters

- **error\_message** (`Union[str, bytes]`) –
- **response\_code** (`Optional[int]`) –
- **response\_body** (`Optional[bytes]`) –

#### Return type

`None`

**exception** `gitlab.GitlabGroupTransferError`(*error\_message*: `Union[str, bytes]` = "", *response\_code*: `Optional[int]` = `None`, *response\_body*: `Optional[bytes]` = `None`)

Bases: `GitlabOperationError`

#### Parameters

- **error\_message** (`Union[str, bytes]`) –
- **response\_code** (`Optional[int]`) –
- **response\_body** (`Optional[bytes]`) –

#### Return type

`None`

**exception** `gitlab.GitlabHeadError`(*error\_message*: `Union[str, bytes]` = "", *response\_code*: `Optional[int]` = `None`, *response\_body*: `Optional[bytes]` = `None`)

Bases: `GitlabOperationError`

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabHousekeepingError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabHttpError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabError](#)**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabImportError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabInvitationError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –

- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabJobCancelError`(*error\_message: Union[str, bytes] = "*, *response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [`GitlabCancelError`](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabJobEraseError`(*error\_message: Union[str, bytes] = "*, *response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [`GitlabRetryError`](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabJobPlayError`(*error\_message: Union[str, bytes] = "*, *response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [`GitlabRetryError`](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabJobRetryError`(*error\_message: Union[str, bytes] = "*, *response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [`GitlabRetryError`](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabLicenseError`(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: `GitlabOperationError`

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**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`: Optional[int]

The next page number.

If None, the current page is the last.

**property** `per_page`: Optional[int]

The number of items per page.

**property** `prev_page`: Optional[int]

The previous page number.

If None, the current page is the first.

**property** `total`: Optional[int]

The total number of items.

**property** `total_pages`: Optional[int]

The total number of pages.

**exception** `gitlab.GitlabListError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = *None*, *response\_body*: *Optional[bytes]* = *None*)

Bases: [`GitlabOperationError`](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** `gitlab.GitlabMRApprovalError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = *None*, *response\_body*: *Optional[bytes]* = *None*)

Bases: [`GitlabOperationError`](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** `gitlab.GitlabMRClosedError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = *None*, *response\_body*: *Optional[bytes]* = *None*)

Bases: [`GitlabOperationError`](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** `gitlab.GitlabMRForbiddenError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = *None*, *response\_body*: *Optional[bytes]* = *None*)

Bases: [`GitlabOperationError`](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** `gitlab.GitlabMROnBuildSuccessError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = *None*, *response\_body*: *Optional[bytes]* = *None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabMRRebaseError`(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabMarkdownError`(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabOperationError`(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [`GitlabError`](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabOwnershipError`(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –



- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabParsingError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabPipelineCancelError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabCancelError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabPipelinePlayError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabRetryError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabPipelineRetryError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabRetryError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabProjectDeployKeyError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = None, *response\_body*: *Optional[bytes]* = None)

Bases: [\*GitlabOperationError\*](#)**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabPromoteError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = None, *response\_body*: *Optional[bytes]* = None)

Bases: [\*GitlabOperationError\*](#)**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabProtectError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = None, *response\_body*: *Optional[bytes]* = None)

Bases: [\*GitlabOperationError\*](#)**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabRenderError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = None, *response\_body*: *Optional[bytes]* = None)

Bases: [\*GitlabOperationError\*](#)**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabRepairError(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

#### Return type

None

**exception** gitlab.GitlabRestoreError(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

#### Return type

None

**exception** gitlab.GitlabRetryError(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

#### Return type

None

**exception** gitlab.GitlabRevertError(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

#### Return type

None

**exception** gitlab.GitlabSearchError(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabSetError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabStopError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabSubscribeError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabTimeTrackingError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabTodoError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = None, *response\_body*: *Optional[bytes]* = None)

Bases: [`GitlabOperationError`](#)**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabTransferProjectError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = None, *response\_body*: *Optional[bytes]* = None)

Bases: [`GitlabOperationError`](#)**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabUnbanError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = None, *response\_body*: *Optional[bytes]* = None)

Bases: [`GitlabOperationError`](#)**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.GitlabUnblockError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = None, *response\_body*: *Optional[bytes]* = None)

Bases: [`GitlabOperationError`](#)**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabUnfollowError(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

**exception** gitlab.GitlabUnsubscribeError(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

**exception** gitlab.GitlabUpdateError(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

**exception** gitlab.GitlabUploadError(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

**exception** gitlab.GitlabUserApproveError(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabUserRejectError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.GitlabVerifyError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.RedirectError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**class** gitlab.Gitlab(*url: Optional[str] = None, private\_token: Optional[str] = None, oauth\_token: Optional[str] = None, job\_token: Optional[str] = None, ssl\_verify: Union[bool, str] = True, http\_username: Optional[str] = None, http\_password: Optional[str] = None, timeout: Optional[float] = None, api\_version: str = '4', session: Optional[Session] = None, per\_page: Optional[int] = None, pagination: Optional[str] = None, order\_by: Optional[str] = None, user\_agent: str = 'python-gitlab/3.9.0', retry\_transient\_errors: bool = False, keep\_base\_url: bool = False*)

Bases: `object`

Represents a GitLab server connection.

**Parameters**

- **url** (*Optional[str]*) – The URL of the GitLab server (defaults to <https://gitlab.com>).
- **private\_token** (*Optional[str]*) – The user private token
- **oauth\_token** (*Optional[str]*) – An oauth token
- **job\_token** (*Optional[str]*) – A CI job token
- **ssl\_verify** (*Union[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** (*Optional[float]*) – Timeout to use for requests to the GitLab server.
- **http\_username** (*Optional[str]*) – Username for HTTP authentication
- **http\_password** (*Optional[str]*) – Password for HTTP authentication
- **api\_version** (*str*) – Gitlab API version to use (support for 4 only)
- **pagination** (*Optional[str]*) – Can be set to ‘keyset’ to use keyset pagination
- **order\_by** (*Optional[str]*) – 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** (*Optional[Session]*) –
- **per\_page** (*Optional[int]*) –

**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](#)



**ci\_lint**

See [\*CiLintManager\*](#)

**deploykeys**

See [\*DeployKeyManager\*](#)

**deploytokens**

See [\*DeployTokenManager\*](#)

**dockerfiles**

See [\*DockerfileManager\*](#)

**static** `enable_debug()` → None

**Return type**

None

**events**

See [\*EventManager\*](#)

**features**

See [\*FeatureManager\*](#)

**classmethod** `from_config(gitlab_id: Optional[str] = None, config_files: Optional[List[str]] = None) → Gitlab`

Create a Gitlab connection from configuration files.

**Parameters**

- **gitlab\_id** (*Optional[str]*) – ID of the configuration section.
- **list[str]** (*config\_files*) – List of paths to configuration files.
- **config\_files** (*Optional[List[str]]*) –

**Returns**

A Gitlab connection.

**Raises**

[\*gitlab.config.GitlabDataError\*](#) – If the configuration is not correct.

**Return type**

[\*Gitlab\*](#)

**geonodes**

See [\*GeoNodeManager\*](#)

**get\_license(\*\*kwargs: Any) → Dict[str, Any]**

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, Any]***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: Optional[Dict[str, Any]] = None, streamed: bool = False, raw: bool = False, \*\*kwargs: Any*) → Union[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** (*Optional[Dict[str, Any]]*) – 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***Union[Dict[str, Any], Response]*

**http\_head**(*path: str, query\_data: Optional[Dict[str, Any]] = None, \*\*kwargs: Any*) → *requests.structures.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** (*Optional[Dict[str, Any]]*) – 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***requests.structures.CaseInsensitiveDict[Any]*

**http\_list**(*path: str, query\_data: Optional[Dict[str, Any]] = None, \*, as\_list: Optional[bool] = None, iterator: Optional[bool] = None, \*\*kwargs: Any*) → *Union[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** (*Optional[Dict[str, Any]]*) – Data to send as query parameters
- **iterator** (*Optional[bool]*) – Indicate if should return a generator (True)
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo, page, per\_page)
- **as\_list** (*Optional[bool]*) –

**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***Union[GitlabList, List[Dict[str, Any]]]*

**http\_post**(*path: str, query\_data: Optional[Dict[str, Any]] = None, post\_data: Optional[Dict[str, Any]] = None, raw: bool = False, files: Optional[Dict[str, Any]] = None, \*\*kwargs: Any*) → *Union[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** (*Optional[Dict[str, Any]]*) – Data to send as query parameters
- **post\_data** (*Optional[Dict[str, Any]]*) – 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** (*Optional[Dict[str, Any]]*) – 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**

*Union[Dict[str, Any], Response]*

**http\_put** (*path: str, query\_data: Optional[Dict[str, Any]] = None, post\_data: Optional[Union[Dict[str, Any], bytes]] = None, raw: bool = False, files: Optional[Dict[str, Any]] = None, \*\*kwargs: Any*) → *Union[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** (*Optional[Dict[str, Any]]*) – Data to send as query parameters
- **post\_data** (*Optional[Union[Dict[str, Any], bytes]]*) – 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** (*Optional[Dict[str, Any]]*) – 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**

*Union[Dict[str, Any], Response]*

**http\_request** (*verb: str, path: str, query\_data: Optional[Dict[str, Any]] = None, post\_data: Optional[Union[Dict[str, Any], bytes]] = None, raw: bool = False, streamed: bool = False, files: Optional[Dict[str, Any]] = None, timeout: Optional[float] = None, obey\_rate\_limit: bool = True, retry\_transient\_errors: Optional[bool] = 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** (*Optional[Dict[str, Any]]*) – Data to send as query parameters
- **post\_data** (*Optional[Union[Dict[str, Any], bytes]]*) – 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** (*Optional[Dict[str, Any]]*) – The files to send to the server
- **timeout** (*Optional[float]*) – 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** (*Optional[bool]*) – 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*

**lint** (*content: str, \*\*kwargs: Any*) → *Tuple[bool, List[str]]*

Validate a gitlab CI configuration.

**Parameters**

- **content** (*str*) – The .gitlab-ci.yml content
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabVerifyError** – If the validation could not be done

**Returns**

(True, []) if the file is valid, (False, errors(list)) otherwise

**Return type**

*Tuple[bool, List[str]]*

**markdown**(*text: str, gfm: bool = False, project: Optional[str] = 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** (*Optional[str]*) – 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: Optional[str] = None, config\_files: Optional[List[str]] = 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** (*Optional[str]*) – ID of the configuration section.
- **config\_files** (*Optional[List[str]]*) – 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*) → Union[*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***Union[GitlabList, List[Dict[str, Any]]]***session**

Create a session object for requests

**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

**timeout**

Timeout to use for requests to gitlab server

**todos**

See [\*TodoManager\*](#)

**topics**

See [\*TopicManager\*](#)

**property url: str**

The user-provided server URL.

**user: Optional[objects.CurrentUser]**

**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:** **Optional[int]**

The next page number.

If None, the current page is the last.

**property per\_page:** **Optional[int]**

The number of items per page.

**property prev\_page:** **Optional[int]**

The previous page number.

If None, the current page is the first.

**property total:** **Optional[int]**

The total number of items.

**property total\_pages:** **Optional[int]**

The total number of pages.

## 6.6.2 Subpackages

### gitlab.v4 package

#### Submodules

#### gitlab.v4.objects module

```
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) –

**manager:** [RESTManager](#)

```
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) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ApplicationAppearanceManager(gl: Gitlab, parent: Optional[RESTObject] =
                                                    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` (*Optional* [[RESTObject](#)]) –

`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](#)

**gitlab:** [Gitlab](#)

`update(id: Optional[Union[int, str]] = None, new_data: Optional[Dict[str, Any]] = None, **kwargs: Any) → Dict[str, Any]`

Update an object on the server.

#### Parameters

- **id** (*Optional* [*Union* [*int*, *str*]]) – ID of the object to update (can be `None` if not required)
- **new\_data** (*Optional* [*Dict* [*str*, *Any*]]) – 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*: *Optional* [[RESTObject](#)] = `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** (*Optional* [[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ApplicationSettingsManager(gl: Gitlab, parent: Optional [RESTObject] =
                                                    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` (*Optional* [[RESTObject](#)]) –

**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](#)

**gitlab:** [Gitlab](#)

**update**(*id: Optional[Union[int, str]] = None, new\_data: Optional[Dict[str, Any]] = None, \*\*kwargs: Any*) → [Dict](#)[[str](#), [Any](#)]

Update an object on the server.

#### Parameters

- **id** (*Optional[Union[int, str]]*) – ID of the object to update (can be `None` if not required)
- **new\_data** (*Optional[Dict[str, Any]]*) – 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.**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*) –

**manager:** [RESTManager](#)

**class** gitlab.v4.objects.**AuditEventManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: [RetrieveMixin](#), [RESTManager](#)

**Object listing filters**

- **created\_after**
- **created\_before**
- **entity\_type**
- **entity\_id**

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[RESTObject]*) –

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → [AuditEvent](#)

Retrieve a single object.

**Parameters**

- **id** (*Union[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](#)

gitlab: [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.BroadcastMessageManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional[[RESTObject](#)]*) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → BroadcastMessage
```

Retrieve a single object.

#### Parameters

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.CiLintManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional[[RESTObject](#)]*) –

**gitlab:** [Gitlab](#)

```
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](#)

**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*) –

**manager**: [RESTManager](#)

**class** gitlab.v4.objects.**CurrentUserEmailManager**(*gl*: [Gitlab](#), *parent*: *Optional[RESTObject] = None*)

Bases: [RetrieveMixin](#), [CreateMixin](#), [DeleteMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- **email**

**Object update**

**Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[RESTObject]*) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → *CurrentUserEmail*

Retrieve a single object.

#### Parameters

- **id** (Union[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*

**gitlab:** *Gitlab*

**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) –

**manager:** *RESTManager*

**class** gitlab.v4.objects.**CurrentUserGPGKeyManager**(*gl*: *Gitlab*, *parent*: Optional[*RESTObject*] = None)

Bases: *RetrieveMixin*, *CreateMixin*, *DeleteMixin*, *RESTManager*

#### Object Creation

Required attributes for object create:

- **key**

#### Object update

##### Parameters

- **gl** (*Gitlab*) –
- **parent** (Optional[*RESTObject*]) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → *CurrentUserGPGKey*

Retrieve a single object.

#### Parameters

- **id** (Union[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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.CurrentUserKeyManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [RetrieveMixin](#), [CreateMixin](#), [DeleteMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- title
- key

**Object update****Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[RESTObject]*) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → CurrentUserKey
```

Retrieve a single object.

**Parameters**

- **id** (*Union[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`

**gitlab:** `Gitlab`

**class** gitlab.v4.objects.**CurrentUserManager**(gl: `Gitlab`, parent: *Optional[RESTObject]* = None)

Bases: `GetWithoutIdMixin`, `RESTManager`

**Object Creation****Object update****Parameters**

- **gl** (`Gitlab`) –
- **parent** (*Optional[RESTObject]*) –

**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`

**gitlab:** `Gitlab`

**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) –

**manager:** `RESTManager`

```
class gitlab.v4.objects.CurrentUserStatusManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [GetWithoutIdMixin](#), [UpdateMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

Optional attributes for object update:

- emoji
- message

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** ([Optional\[RESTObject\]](#)) –

```
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](#)

**gitlab:** [Gitlab](#)

```
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](#)) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.DeployKeyManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [ListMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –

- **parent** (*Optional*[[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.DeployTokenManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [ListMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional*[[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.DockerfileManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional*[[RESTObject](#)]) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → *Dockerfile*

Retrieve a single object.

#### Parameters

- **id** (Union[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*

**gitlab:** *Gitlab*

**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) –

**manager:** *RESTManager*

**class** gitlab.v4.objects.**EventManager**(*gl*: *Gitlab*, *parent*: Optional[*RESTObject*] = None)

Bases: *ListMixin*, *RESTManager*

#### Object listing filters

- action
- target\_type
- before
- after
- sort

#### Object Creation

#### Object update

#### Parameters

- **gl** (*Gitlab*) –
- **parent** (Optional[*RESTObject*]) –



gitlab: [Gitlab](#)

```
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](#)) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.FeatureManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [ListMixin](#), [DeleteMixin](#), [RESTManager](#)

#### Object Creation

##### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** ([Optional](#) [[RESTObject](#)]) –

gitlab: [Gitlab](#)

```
set(name: str, value: Union[bool, int], feature_group: Optional[str] = None, user: Optional[str] = None,
    group: Optional[str] = None, project: Optional[str] = None, **kwargs: Any) → Feature
```

Create or update the object.

#### Parameters

- **name** ([str](#)) – The value to set for the object
- **value** ([Union](#)[[bool](#), [int](#)]) – The value to set for the object
- **feature\_group** ([Optional](#) [[str](#)]) – A feature group name
- **user** ([Optional](#) [[str](#)]) – A GitLab username
- **group** ([Optional](#) [[str](#)]) – A GitLab group
- **project** ([Optional](#) [[str](#)]) – 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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GenericPackageManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[RESTObject]*) –

```
download(package_name: str, package_version: str, file_name: str, streamed: bool = False, action:
Optional[Callable[[bytes], None]] = None, chunk_size: int = 1024, *, iterator: bool = False,
**kwargs: Any) → Optional[Union[bytes, Iterator[Any]]]
```

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** (*Optional[Callable[[bytes], 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

*Optional[Union[bytes, Iterator[Any]]]*

**gitlab:** *Gitlab*

**upload**(*package\_name: str, package\_version: str, file\_name: str, path: Union[str, Path], \*\*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** (*Union[str, Path]*) – The path to a local file to upload
- **kwargs** (*Any*) –

#### Raises

- **GitlabConnectionError** – If the server cannot be reached
- **GitlabUploadError** – If the file upload fails
- **GitlabUploadError** – If filepath cannot be read

#### 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*) –

**manager:** *RESTManager*

**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: Optional[RESTObject] = 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** (Optional[[RESTObject](#)]) –

**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: Union[str, int], lazy: bool = False, *\*\*kwargs: Any*) → [GeoNode](#)

Retrieve a single object.

**Parameters**

- **id** (Union[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*

**gitlab:** *Gitlab*

**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*) –

**manager:** *RESTManager*

**class** gitlab.v4.objects.**GitignoreManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: *RetrieveMixin, RESTManager*

**Object Creation****Object update****Parameters**

- **gl** (*Gitlab*) –
- **parent** (*Optional[RESTObject]*) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → *Gitignore*

Retrieve a single object.

#### Parameters

- **id** (Union[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*

**gitlab:** *Gitlab*

**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) –

**manager:** *RESTManager*

**class** gitlab.v4.objects.**GitlabciyamlManager**(*gl*: *Gitlab*, *parent*: Optional[*RESTObject*] = None)

Bases: *RetrieveMixin*, *RESTManager*

#### Object Creation

#### Object update

#### Parameters

- **gl** (*Gitlab*) –
- **parent** (Optional[*RESTObject*]) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → *Gitlabciyaml*

Retrieve a single object.

#### Parameters

- **id** (Union[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

gitlab: [Gitlab](#)

```
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](#)

**add\_ldap\_group\_link**(*cn: str, group\_access: int, provider: str, \*\*kwargs: Any*) → None

Add an LDAP group link.

**Parameters**

- **cn** ([str](#)) – CN of the LDAP group
- **group\_access** ([int](#)) – Minimum access level for members of the LDAP group
- **provider** ([str](#)) – LDAP provider for the LDAP group
- **\*\*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

**audit\_events:** [GroupAuditEventManager](#)

**badges:** [GroupBadgeManager](#)

**billable\_members:** [GroupBillableMemberManager](#)

**boards:** [GroupBoardManager](#)

**clusters:** [GroupClusterManager](#)

**customattributes:** [GroupCustomAttributeManager](#)

**delete\_ldap\_group\_link**(*cn: str, provider: Optional[str] = None, \*\*kwargs: Any*) → None

Delete an LDAP group link.

**Parameters**

- **cn** (*str*) – CN of the LDAP group
- **provider** (*Optional[str]*) – LDAP provider for the LDAP group
- **\*\*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

**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\_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\*](#)

**runners:** [\*GroupRunnerManager\*](#)

**search**(*scope: str, search: str, \*\*kwargs: Any*) → Union[[\*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

Union[[\*GitlabList\*](#), List[Dict[str, Any]]]

**share**(*group\_id: int, group\_access: int, expires\_at: Optional[str] = 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** (*Optional[str]*) –
- **\*\*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: Optional[int] = 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** (*Optional[int]*) – 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](#)) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupAccessRequestManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [ListMixin](#), [CreateMixin](#), [DeleteMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** ([Optional](#)[[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
class gitlab.v4.objects.GroupAccessToken(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](#)) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupAccessTokenManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [ListMixin](#), [CreateMixin](#), [DeleteMixin](#), [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** (*Optional* [[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupAuditEventManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

#### Object listing filters

- **created\_after**
- **created\_before**

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → GroupAuditEvent
```

Retrieve a single object.

#### Parameters

- **id** (*Union*[*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](#)

**gitlab:** [Gitlab](#)

```
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](#)) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupBadgeManager(gl: Gitlab, parent: Optional[RESTObject] = 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** ([Optional](#)[[RESTObject](#)]) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → GroupBadge
```

Retrieve a single object.

#### Parameters

- **id** ([Union](#)[[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](#)

gitlab: [Gitlab](#)

```
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: Optional[RESTObject] =
                                                    None)
```

Bases: [ListMixin](#), [DeleteMixin](#), [RESTManager](#)

#### Object listing filters

- search
- sort

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (Optional[[RESTObject](#)]) –

gitlab: [Gitlab](#)

```
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) –

manager: [RESTManager](#)

```
class gitlab.v4.objects.GroupBillableMemberMembershipManager(gl: Gitlab, parent:
                                                                Optional[RESTObject] = None)
```

Bases: [ListMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

**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*) –

**manager:** [RESTManager](#)

**class** gitlab.v4.objects.**GroupBoardListManager**(*gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional* [[RESTObject](#)]) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → *GroupBoardList*

Retrieve a single object.

#### Parameters

- **id** (Union[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*

**gitlab:** *Gitlab*

**class** gitlab.v4.objects.**GroupBoardManager**(*gl*: *Gitlab*, *parent*: Optional[RESTObject] = None)

Bases: *CRUDMixin*, *RESTManager*

#### Object Creation

Required attributes for object create:

- name

#### Object update

##### Parameters

- **gl** (*Gitlab*) –
- **parent** (Optional[RESTObject]) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → *GroupBoard*

Retrieve a single object.

#### Parameters

- **id** (Union[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*



gitlab: [Gitlab](#)

**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*) –

**manager:** [RESTManager](#)

**class** gitlab.v4.objects.**GroupClusterManager**(gl: [Gitlab](#), parent: *Optional[RESTObject] = 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** (*Optional[RESTObject]*) –

**create**(data: *Optional[Dict[str, Any]] = None*, \*\*kwargs: *Any*) → [GroupCluster](#)

Create a new object.

#### Parameters

- **data** (*Optional[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` 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*: `Union[str, int]`, *lazy*: `bool = False`, *\*\*kwargs*: *Any*) → `GroupCluster`

Retrieve a single object.

**Parameters**

- **id** (`Union[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`

**gitlab:** `Gitlab`

```
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`) –

**manager:** `RESTManager`

```
class gitlab.v4.objects.GroupCustomAttributeManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: `RetrieveMixin`, `SetMixin`, `DeleteMixin`, `RESTManager`

**Object Creation**

**Object update**

**Parameters**

- **gl** (*Gitlab*) –
- **parent** (*Optional[RESTObject]*) –

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → *GroupCustomAttribute*

Retrieve a single object.

**Parameters**

- **id** (*Union[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*

**gitlab:** *Gitlab*

```
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*) –

**manager:** *RESTManager*

```
class gitlab.v4.objects.GroupDeployTokenManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [GroupDeployToken](#)

Retrieve a single object.

### Parameters

- **id** (*Union*[*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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupDescendantGroupManager(gl: Gitlab, parent: Optional[RESTObject] =
                                                    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` (*Optional* [`RESTObject`]) –

```
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*) –

`manager:` *RESTManager*

```
class gitlab.v4.objects.GroupEpicAwardEmojiManager(gl: Gitlab, parent: Optional[RESTObject] =
    None)
```

Bases: *NoUpdateMixin*, *RESTManager*

**Object Creation**

Required attributes for object create:

- `name`

**Object update**

**Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → [GroupEpicAwardEmoji](#)

Retrieve a single object.

**Parameters**

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupEpicDiscussionNoteManager(gl: Gitlab, parent: Optional [RESTObject] =
                                                         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** (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [GroupEpicDiscussionNote](#)

Retrieve a single object.

**Parameters**

- **id** (*Union*[*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](#)

**gitlab:** [Gitlab](#)

```
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: *Optional*[[RESTObject](#)] = 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** (*Optional*[[RESTObject](#)]) –

**create**(data: *Optional*[[Dict](#)[*str*, *Any*]] = None, **\*\*kwargs**: *Any*) → [GroupEpicIssue](#)

Create a new object.

#### Parameters

- **data** (*Optional*[[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`)

#### 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](#)

**gitlab:** [Gitlab](#)

**class** gitlab.v4.objects.**GroupEpicManager**(gl: [Gitlab](#), parent: *Optional*[[RESTObject](#)] = 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` (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *GroupEpic*

Retrieve a single object.

#### Parameters

- `id` (*Union*[*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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupEpicNoteAwardEmojiManager(gl: Gitlab, parent: Optional[RESTObject] =
                                                         None)
```

Bases: [NoUpdateMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **name**

#### Object update

##### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[[RESTObject](#)]*) –

**get** (*id: Union[str, int]*, *lazy: bool = False*, *\*\*kwargs: Any*) → [GroupEpicNoteAwardEmoji](#)

Retrieve a single object.

##### Parameters

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

**class** gitlab.v4.objects.GroupEpicNoteManager(*gl*: [Gitlab](#), *parent*: *Optional*[[RESTObject](#)] = *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* (*Optional*[[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [GroupEpicNote](#)

Retrieve a single object.

#### Parameters

- *id* (*Union*[*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](#)

**gitlab:** [Gitlab](#)

**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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupEpicResourceLabelEventManager(gl: Gitlab, parent:
Optional[RESTObject] = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

**Object Creation**

**Object update**

**Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**get**(*id: Union[str, int]*, *lazy: bool = False*, *\*\*kwargs: Any*) → [GroupEpicResourceLabelEvent](#)

Retrieve a single object.

**Parameters**

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupExportManager(gl: Gitlab, parent: Optional [RESTObject] = None)
```

Bases: [GetWithoutIdMixin](#), [CreateMixin](#), [RESTManager](#)

**Object Creation**

**Object update**

**Parameters**

- **gl** (*Gitlab*) –
- **parent** (*Optional* [*RESTObject*]) –

**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*

**gitlab:** *Gitlab*

```
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*) –

**manager:** *RESTManager*

```
class gitlab.v4.objects.GroupHookManager(gl: Gitlab, parent: Optional[RESTObject] = 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` (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [GroupHook](#)

Retrieve a single object.

### Parameters

- `id` (*Union*[*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](#)

gitlab: [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupImportManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [GetWithoutIdMixin](#), [RESTManager](#)

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[[RESTObject](#)]*) –

```
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](#)

gitlab: [Gitlab](#)

```
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](#)) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupInvitationManager(gl: Gitlab, parent: Optional[RESTObject] = 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** ([Optional](#)[[RESTObject](#)]) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → GroupInvitation
```

Retrieve a single object.

#### Parameters

- **id** ([Union](#)[[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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupIssueManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [ListMixin](#), [RESTManager](#)

**Object listing filters**

- state
- labels
- milestone
- order\_by
- sort
- iids
- author\_id
- assignee\_id
- my\_reaction\_emoji
- search
- created\_after
- created\_before
- updated\_after
- updated\_before

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupIssuesStatisticsManager(gl: Gitlab, parent: Optional [RESTObject] =
                                                    None)
```

Bases: [GetWithoutIdMixin](#), [RESTManager](#)**Object listing filters**

- **iids**

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**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](#)**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupIterationManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [ListMixin](#), [RESTManager](#)

#### Object listing filters

- state
- search
- include\_ancestors

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[RESTObject]*) –

**gitlab:** [Gitlab](#)

```
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*: *Optional*[[RESTObject](#)] = 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** (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [GroupLabel](#)

Retrieve a single object.

**Parameters**

- **id** (*Union* [*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](#)

**gitlab:** [Gitlab](#)

**update**(*name: Optional[str], new\_data: Optional[Dict[str, Any]] = None, \*\*kwargs: Any*) → Dict[str, Any]

Update a Label on the server.

#### Parameters

- **name** (*Optional[str]*) – The name of the label
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)
- **new\_data** (*Optional[Dict[str, Any]]*) –
- **\*\*kwargs** –

#### Return type

Dict[str, Any]

**class** gitlab.v4.objects.**GroupManager**(*gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional* [[RESTObject](#)]) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → *Group*

Retrieve a single object.

#### Parameters

- **id** (Union[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*

**gitlab:** *Gitlab*

**import\_group**(*file*: BinaryIO, *path*: str, *name*: str, *parent\_id*: Optional[str] = None, *\*\*kwargs*: Any) → Union[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** (Optional[str]) – 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

Union[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*) –

**manager:** [RESTManager](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupMemberAllManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[[RESTObject](#)]*) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → GroupMemberAll
```

Retrieve a single object.

#### Parameters

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

```
class gitlab.v4.objects.GroupMemberManager(gl: Gitlab, parent: Optional[RESTObject] = 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` (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *GroupMember*

Retrieve a single object.

### Parameters

- `id` (*Union*[*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](#)

**gitlab:** [Gitlab](#)

**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*) –

**manager:** [RESTManager](#)

**class** gitlab.v4.objects.**GroupMergeRequestManager**(gl: [Gitlab](#), parent: *Optional[RESTObject]* = 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** (*Optional[RESTObject]*) –

**gitlab:** [Gitlab](#)

**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*

**manager:** *RESTManager*

**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: *Optional[RESTObject]* = 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` (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [GroupMilestone](#)

Retrieve a single object.

#### Parameters

- `id` (*Union*[*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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** *RESTManager*

**class** gitlab.v4.objects.**GroupNotificationSettingsManager**(*gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional[RESTObject]*) –

**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*

**gitlab:** *Gitlab*

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupPackageManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [ListMixin](#), [RESTManager](#)

#### Object listing filters

- **exclude\_subgroups**
- **order\_by**
- **sort**
- **package\_type**
- **package\_name**

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[RESTObject]*) –

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupProjectManager(gl: Gitlab, parent: Optional[RESTObject] = 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` (*Optional* [[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupPushRulesManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional* [[RESTObject](#)]) –

**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](#)

**gitlab:** [Gitlab](#)



```
class gitlab.v4.objects.GroupRegistryRepositoryManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [ListMixin](#), [RESTManager](#)

**Object Creation**

**Object update**

**Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional*[[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.GroupRunnerManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional*[[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [\*RESTManager\*](#)

**class** gitlab.v4.objects.**GroupSubgroupManager**(*gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional[RESTObject]*) –

**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*) –

**manager:** [\*RESTManager\*](#)

**class** gitlab.v4.objects.**GroupVariableManager**(*gl: Gitlab, parent: Optional[RESTObject] = 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` (*Optional* [`RESTObject`]) –

**get**(*id*: `Union[str, int]`, *lazy*: `bool = False`, *\*\*kwargs*: `Any`) → `GroupVariable`

Retrieve a single object.

### Parameters

- `id` (`Union[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`

**gitlab:** `Gitlab`

```
class gitlab.v4.objects.GroupWiki(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`) –

**manager:** [\*RESTManager\*](#)

**class** gitlab.v4.objects.**GroupWikiManager**(*gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional* [[\*RESTObject\*](#)]) –

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → [\*GroupWiki\*](#)

Retrieve a single object.

#### Parameters

- **id** (*Union[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\*](#)

**gitlab:** [\*Gitlab\*](#)

```
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](#)) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.HookManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [NoUpdateMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **url**

#### Object update

##### Parameters

- **gl** ([Gitlab](#)) –
- **parent** ([Optional](#)[[RESTObject](#)]) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → Hook
```

Retrieve a single object.

##### Parameters

- **id** ([Union](#)[[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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [\*RESTManager\*](#)

**class** gitlab.v4.objects.**IssueManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: [\*RetrieveMixin\*](#), [\*RESTManager\*](#)

#### Object listing filters

- **state**
- **labels**
- **milestone**
- **scope**
- **author\_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** (*Optional[[\*RESTObject\*](#)]*) –

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → [\*Issue\*](#)

Retrieve a single object.

##### Parameters

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.IssuesStatisticsManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [GetWithoutIdMixin](#), [RESTManager](#)

**Object listing filters**

- **iids**

**Object Creation**

**Object update**

**Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[[RESTObject](#)]*) –

```
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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

**class** gitlab.v4.objects.**KeyManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: [GetMixin](#), [RESTManager](#)

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[RESTObject]*) –

**get**(*id: Optional[Union[int, str]] = None, lazy: bool = False, \*\*kwargs: Any*) → [Key](#)

Retrieve a single object.

**Parameters**

- **id** (*Optional[Union[int, str]]*) – 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](#)

**gitlab:** [Gitlab](#)

**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*) –

**manager:** [RESTManager](#)



**class** gitlab.v4.objects.LDAPGroupManager(*gl*: Gitlab, *parent*: Optional[RESTObject] = None)

Bases: [RESTManager](#)

Object listing filters

- search
- provider

Object Creation

Object update

Parameters

- *gl* (Gitlab) –
- *parent* (Optional[RESTObject]) –

gitlab: [Gitlab](#)

**list**(\*\*kwargs: Any) → Union[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

Union[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) –

manager: [RESTManager](#)

```
class gitlab.v4.objects.LicenseManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

Object listing filters

- popular

Object Creation

Object update

Parameters

- **gl** ([Gitlab](#)) –
- **parent** ([Optional\[RESTObject\]](#)) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → License
```

Retrieve a single object.

Parameters

- **id** ([Union\[str, int\]](#)) – 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](#)

gitlab: [Gitlab](#)

```
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](#)) –

manager: [RESTManager](#)

```
class gitlab.v4.objects.MergeRequestManager(gl: Gitlab, parent: Optional[RESTObject] = 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` (*Optional* [[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.NamespaceManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

#### Object listing filters

- search

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[RESTObject]*) –

```
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: Union[str, int], lazy: bool = False, **kwargs: Any) → Namespace
```

Retrieve a single object.

#### Parameters

- **id** (*Union[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

**gitlab:** *Gitlab*

```
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*) –

**manager:** *RESTManager*

```
class gitlab.v4.objects.NotificationSettingsManager(gl: Gitlab, parent: Optional[RESTObject] =
                                                    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** (*Optional* [[RESTObject](#)]) –

**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](#)

**gitlab:** [Gitlab](#)

```
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) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.PagesDomainManager(gl: Gitlab, parent: Optional [RESTObject] = None)
```

Bases: [ListMixin](#), [RESTManager](#)

**Object Creation**

**Object update**

**Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
class gitlab.v4.objects.PersonalAccessToken(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) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.PersonalAccessTokenManager(gl: Gitlab, parent: Optional[RESTObject] =
None)
```

Bases: [DeleteMixin](#), [ListMixin](#), [RESTManager](#)

Object listing filters

- `user_id`

Object Creation

Object update

Parameters

- `gl` ([Gitlab](#)) –
- `parent` ([Optional](#) [[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
class gitlab.v4.objects.Project(manager: RESTManager, attrs: Dict[str, Any], *, created_from_list: bool
= False, lazy: bool = False)
```

Bases: [RefreshMixin](#), [SaveMixin](#), [ObjectDeleteMixin](#), [RepositoryMixin](#), [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](#)

**artifact**(*\*args*: [Any](#), *\*\*kwargs*: [Any](#)) → [Optional](#) [[bytes](#)]

Parameters

- ***args*** ([Any](#)) –

- **kwargs** (*Any*) –

**Return type**

*Optional*[bytes]

**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:** [\*GroupIterationManager\*](#)

**jobs:** [\*ProjectJobManager\*](#)

**keys:** [\*ProjectKeyManager\*](#)

**labels:** [\*ProjectLabelManager\*](#)

**languages**(\*\*kwargs: Any) → Union[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

Union[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

**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\*](#)

**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*) → Union[[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

Union[[GitlabList](#), List[Dict[str, Any]]]

**services:** [ProjectServiceManager](#)

**share**(*group\_id: int, group\_access: int, expires\_at: Optional[str] = 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** (*Optional[str]*) –
- **\*\*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: Optional[Callable[[bytes], None]] = None, chunk\_size: int = 1024, \*, iterator: bool = False, \*\*kwargs: Any*) → Optional[Union[bytes, Iterator[Any]]]

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** (*Optional[Callable[[bytes], 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**

*Optional[Union[bytes, Iterator[Any]]]*

**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: Union[int, str]*, *\*\*kwargs: Any*) → None

Transfer a project to the given namespace ID

**Parameters**

- **to\_namespace** (*Union[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

**transfer\_project**(*\*args: Any*, *\*\*kwargs: Any*) → None

**Parameters**

- **args** (*Any*) –
- **kwargs** (*Any*) –

**Return type**

None

**trigger\_pipeline**(*ref: str, token: str, variables: Optional[Dict[str, Any]] = 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** (*Optional[Dict[str, Any]]*) – 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

**upload**(*filename: str, filedata: Optional[bytes] = None, filepath: Optional[str] = None, \*\*kwargs: Any*) → Dict[str, Any]

Upload the specified file into the project.

---

**Note:** Either *filedata* or *filepath* *MUST* be specified.

---

**Parameters**

- **filename** (*str*) – The name of the file being uploaded
- **filedata** (*Optional[bytes]*) – The raw data of the file being uploaded
- **filepath** (*Optional[str]*) – The path to a local file to upload (optional)
- **kwargs** (*Any*) –

**Raises**

- **GitlabConnectionError** – If the server cannot be reached
- **GitlabUploadError** – If the file upload fails
- **GitlabUploadError** – If *filedata* and *filepath* are not specified
- **GitlabUploadError** – If both *filedata* and *filepath* are specified

**Returns**

- *alt* - The alternate text for the upload
- *url* - The direct url to the uploaded file
- *markdown* - Markdown for the uploaded file

**Return type**

A dict with the keys

**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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectAccessRequestManager(gl: Gitlab, parent: Optional[RESTObject] =
                                                    None)
```

Bases: [ListMixin](#), [CreateMixin](#), [DeleteMixin](#), [RESTManager](#)

**Object Creation**

**Object update**

**Parameters**

- **gl** ([Gitlab](#)) –
- **parent** ([Optional\[RESTObject\]](#)) –

**gitlab:** [Gitlab](#)

```
class gitlab.v4.objects.ProjectAccessToken(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](#)) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectAccessTokenManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [ListMixin](#), [CreateMixin](#), [DeleteMixin](#), [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** ([Optional\[RESTObject\]](#)) –

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectAdditionalStatisticsManager(gl: Gitlab, parent:
Optional[RESTObject] = None)
```

Bases: [GetWithoutIdMixin](#), [RESTManager](#)

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

```
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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)



**class** gitlab.v4.objects.**ProjectApprovalManager**(gl: [Gitlab](#), parent: *Optional*[[RESTObject](#)] = 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** (*Optional*[[RESTObject](#)]) –

**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](#)

**gitlab:** [Gitlab](#)

**set\_approvers**(*approver\_ids: Optional*[*List*[int]] = None, *approver\_group\_ids: Optional*[*List*[int]] = None, \*\*kwargs: Any) → Dict[str, Any]

Change project-level allowed approvers and approver groups.

#### Parameters

- **approver\_ids** (*Optional*[*List*[int]]) – User IDs that can approve MRs
- **approver\_group\_ids** (*Optional*[*List*[int]]) – Group IDs whose members can approve MRs
- **kwargs** (Any) –

#### Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUpdateError** – If the server failed to perform the request

#### Returns

A dict value of the result

**Return type***Dict[str, Any]*

```
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*) –

**manager:** *RESTManager*

```
class gitlab.v4.objects.ProjectApprovalRuleManager(gl: Gitlab, parent: Optional[RESTObject] =
                                                    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**

**Object update****Parameters**

- **gl** (*Gitlab*) –
- **parent** (*Optional[RESTObject]*) –

**gitlab:** *Gitlab*

```
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*) –

**manager:** *RESTManager*

**class** gitlab.v4.objects.**ProjectArtifactManager**(gl: [Gitlab](#), parent: *Optional*[[RESTObject](#)] = None)

Bases: [RESTManager](#)

**Object Creation**

**Object update**

**Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional*[[RESTObject](#)]) –

**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: *Optional*[Callable[[bytes], None]] = None, chunk\_size: int = 1024, \*, iterator: bool = False, \*\*kwargs: Any) → *Optional*[Union[bytes, Iterator[Any]]]

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
- **supported.** (are not) –
- **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** (*Optional*[Callable[[bytes], 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**

*Optional*[Union[bytes, Iterator[Any]]]

**gitlab:** [Gitlab](#)

**raw**(*ref\_name: str, artifact\_path: str, job: str, streamed: bool = False, action: Optional[Callable[[bytes], None]] = None, chunk\_size: int = 1024, \*, iterator: bool = False, \*\*kwargs: Any*) → *Optional[Union[bytes, Iterator[Any]]]*

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** (*Optional[Callable[[bytes], 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

*Optional[Union[bytes, Iterator[Any]]]*

**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*) –

**manager:** [RESTManager](#)

**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*) –

**manager:** [RESTManager](#)

**class** gitlab.v4.objects.**ProjectAuditEventManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: [RetrieveMixin](#), [RESTManager](#)

#### Object listing filters

- **created\_after**
- **created\_before**

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → [ProjectAuditEvent](#)

Retrieve a single object.

##### Parameters

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

**class** gitlab.v4.objects.**ProjectAuditManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: [ProjectAuditEventManager](#)

#### Object listing filters

- **created\_after**
- **created\_before**

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#)) –

- **parent** (*Optional*[[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectBadgeManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional*[[RESTObject](#)]) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → ProjectBadge
```

Retrieve a single object.

#### Parameters

- **id** (*Union*[*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](#)

gitlab: [Gitlab](#)

```
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](#)) –

manager: [RESTManager](#)

```
class gitlab.v4.objects.ProjectBoardListManager(gl: Gitlab, parent: Optional[RESTObject] = 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** ([Optional](#) [[RESTObject](#)]) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → ProjectBoardList

Retrieve a single object.

#### Parameters

- **id** (Union[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

**gitlab:** *Gitlab*

**class** gitlab.v4.objects.**ProjectBoardManager**(*gl*: *Gitlab*, *parent*: Optional[RESTObject] = None)

Bases: *CRUDMixin*, *RESTManager*

#### Object Creation

Required attributes for object create:

- name

#### Object update

##### Parameters

- **gl** (*Gitlab*) –
- **parent** (Optional[RESTObject]) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → ProjectBoard

Retrieve a single object.

#### Parameters

- **id** (Union[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



gitlab: [Gitlab](#)

```
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](#)) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectBranchManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [NoUpdateMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **branch**
- **ref**

#### Object update

##### Parameters

- **gl** ([Gitlab](#)) –
- **parent** ([Optional](#)[[RESTObject](#)]) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → ProjectBranch
```

Retrieve a single object.

##### Parameters

- **id** ([Union](#)[[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](#)

gitlab: [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectCiLintManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional[RESTObject]*) –

```
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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager**: [RESTManager](#)

**class** gitlab.v4.objects.**ProjectClusterManager**(*gl*: [Gitlab](#), *parent*: *Optional[RESTObject]* = *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** (*Optional[RESTObject]*) –

**create**(*data: Optional[Dict[str, Any]] = None, \*\*kwargs: Any*) → *ProjectCluster*

Create a new object.

**Parameters**

- **data** (*Optional[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` 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: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → *ProjectCluster*

Retrieve a single object.

**Parameters**

- **id** (*Union[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*

**gitlab:** *Gitlab*

**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) → Union[[\*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**

Union[[\*GitlabList\*](#), List[Dict[str, Any]]]

**discussions:** [\*ProjectCommitDiscussionManager\*](#)

**merge\_requests**(*\*\*kwargs*: Any) → Union[[\*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**

Union[[\*GitlabList\*](#), List[Dict[str, Any]]]

**refs**(*type*: str = 'all', *\*\*kwargs*: Any) → Union[[\*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**

*Union[GitlabList, List[Dict[str, Any]]]*

**revert**(*branch: str, \*\*kwargs: Any*) → Union[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**

*Union[Dict[str, Any], Response]*

**signature**(*\*\*kwargs: Any*) → Union[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**

*Union[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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectCommitCommentManager(gl: Gitlab, parent: Optional[RESTObject] =
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** (*Optional* [[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
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: Optional[RESTObject] =
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** (*Optional* [[RESTObject](#)]) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → *ProjectCommitDiscussion*

Retrieve a single object.

#### Parameters

- **id** (Union[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*

**gitlab:** *Gitlab*

```
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) –

**manager:** *RESTManager*

```
class gitlab.v4.objects.ProjectCommitDiscussionNoteManager(gl: Gitlab, parent:  
                                                           Optional[RESTObject] = 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** (*Optional* [[RESTObject](#)]) –

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → *ProjectCommitDiscussionNote*

Retrieve a single object.

#### Parameters

- **id** (*Union[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*

**gitlab:** [Gitlab](#)

**class** gitlab.v4.objects.**ProjectCommitManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: [RetrieveMixin](#), [CreateMixin](#), [RESTManager](#)

#### Object listing filters

- **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** (*Optional[RESTObject]*) –

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → *ProjectCommit*

Retrieve a single object.

**Parameters**

- **id** (*Union[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*

**gitlab:** *Gitlab*

```
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*) –

**manager:** *RESTManager*

```
class gitlab.v4.objects.ProjectCommitStatusManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional* [[RESTObject](#)]) –

**create**(*data: Optional[Dict[str, Any]] = None, \*\*kwargs: Any*) → *ProjectCommitStatus*

Create a new object.

##### Parameters

- **data** (*Optional[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` 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*

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectCustomAttributeManager(gl: Gitlab, parent: Optional[RESTObject] =
                                                    None)
```

Bases: *RetrieveMixin*, *SetMixin*, *DeleteMixin*, *RESTManager*

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#)) –

- **parent** (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [ProjectCustomAttribute](#)

Retrieve a single object.

#### Parameters

- **id** (*Union*[*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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectDeployTokenManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectDeployToken*

Retrieve a single object.

#### Parameters

- **id** (*Union*[*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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**mergerequests:** [ProjectDeploymentMergeRequestManager](#)

```
class gitlab.v4.objects.ProjectDeploymentManager(gl: Gitlab, parent: Optional[RESTObject] = 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`** (*Optional* [[RESTObject](#)]) –

**`get`**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [ProjectDeployment](#)

Retrieve a single object.

##### Parameters

- **`id`** (*Union*[*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](#)

**`gitlab`**: [Gitlab](#)

```
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*) –

**`manager`**: [RESTManager](#)

```
class gitlab.v4.objects.ProjectDeploymentMergeRequestManager(gl: Gitlab, parent:  
Optional[RESTObject] = 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` (*Optional* [[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

**stop**(\*\*kwargs: Any) → Union[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

*Union[Dict[str, Any], Response]*

```
class gitlab.v4.objects.ProjectEnvironmentManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional*[[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [ProjectEnvironment](#)

Retrieve a single object.

#### Parameters

- **id** (*Union*[*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](#)

**gitlab:** [Gitlab](#)

**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*) –

**manager:** [RESTManager](#)

**class** gitlab.v4.objects.**ProjectEventManager**(*gl*: [Gitlab](#), *parent*: *Optional*[[RESTObject](#)] = *None*)

Bases: [EventManager](#)

#### Object listing filters

- action
- target\_type
- before
- after
- sort

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –

- **parent** (*Optional*[[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectExportManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [GetWithoutIdMixin](#), [CreateMixin](#), [RESTManager](#)

#### Object Creation

Optional attributes for object create:

- **description**

#### Object update

##### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional*[[RESTObject](#)]) –

```
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](#)

**gitlab:** [Gitlab](#)

```
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: *Optional*[[RESTObject](#)] = 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** (*Optional* [[RESTObject](#)]) –

**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: Optional[Dict[str, Any]] = None, \*\*kwargs: Any*) → *ProjectFile*

Create a new object.

**Parameters**

- **data** (*Optional[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**

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`**gitlab:** `Gitlab`

**raw**(*file\_path*: *str*, *ref*: *str*, *streamed*: *bool* = *False*, *action*: *Optional*[*Callable*[*...*, *Any*]] = *None*, *chunk\_size*: *int* = 1024, \*, *iterator*: *bool* = *False*, *\*\*kwargs*: *Any*) → *Optional*[*Union*[*bytes*, *Iterator*[*Any*]]]

Return the content of a file for a commit.

**Parameters**

- **ref** (*str*) – ID of the commit
- **filepath** – 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** (*Optional*[*Callable*[*...*, *Any*]]) – 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)
- **file\_path** (*str*) –

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the file could not be retrieved

**Returns**

The file content

**Return type**`Optional[Union[bytes, Iterator[Any]]]`

**update**(*file\_path*: *str*, *new\_data*: *Optional*[*Dict*[*str*, *Any*]] = *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** (*Optional*[*Dict*[*str*, *Any*]]) – 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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectForkManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional[[RESTObject](#)]*) –

**create**(data: *Optional[Dict[str, Any]]* = None, \*\*kwargs: Any) → [ProjectFork](#)

Creates a new object.

##### Parameters

- **data** (*Optional[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)

**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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectHookManager(gl: Gitlab, parent: Optional[RESTObject] = 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` (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [ProjectHook](#)

Retrieve a single object.

#### Parameters

- `id` (*Union*[*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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

**class** gitlab.v4.objects.**ProjectImportManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: [GetWithoutIdMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[RESTObject]*) –

**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](#)

**gitlab:** [Gitlab](#)

**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*) –

**manager:** [RESTManager](#)

**class** gitlab.v4.objects.**ProjectIntegrationManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: [GetMixin](#), [UpdateMixin](#), [DeleteMixin](#), [ListMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**available()** → [List\[str\]](#)

List the services known by python-gitlab.

#### Returns

The list of service code names.

#### Return type

[List\[str\]](#)

**get**(*id*: [Union\[str, int\]](#), *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [ProjectIntegration](#)

Retrieve a single object.

#### Parameters

- **id** ([Union\[str, int\]](#)) – 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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectInvitationManager(gl: Gitlab, parent: Optional [RESTObject] = 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` (Optional [[RESTObject](#)]) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → [ProjectInvitation](#)

Retrieve a single object.

#### Parameters

- `id` (Union[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](#)

**gitlab:** [Gitlab](#)

**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: Optional[int] = None, move\_before\_id: Optional[int] = None, \*\*kwargs: Any*) → None

Reorder an issue on a board.

**Parameters**

- **move\_after\_id** (*Optional[int]*) – ID of an issue that should be placed after this issue
- **move\_before\_id** (*Optional[int]*) – 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

**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*) –

**manager:** [RESTManager](#)

**class** gitlab.v4.objects.**ProjectIssueAwardEmojiManager**(*gl: [Gitlab](#), parent: Optional[[RESTObject](#)] = None*)

Bases: [NoUpdateMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- **name**

**Object update****Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[[RESTObject](#)]*) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → *ProjectIssueAwardEmoji*

Retrieve a single object.

#### Parameters

- **id** (Union[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*

**gitlab:** *Gitlab*

**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*: Optional[*RESTObject*] = 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** (Optional[*RESTObject*]) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → *ProjectIssueDiscussion*

Retrieve a single object.

#### Parameters

- **id** (Union[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*

**gitlab:** *Gitlab*

```
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) –

**manager:** *RESTManager*

```
class gitlab.v4.objects.ProjectIssueDiscussionNoteManager(gl: Gitlab, parent:
                                                         Optional[RESTObject] = 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** (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectIssueDiscussionNote*

Retrieve a single object.

#### Parameters

- **id** (*Union*[*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*

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectIssueLinkManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional* [*RESTObject*]) –

**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]*

**gitlab:** *Gitlab*

**class** gitlab.v4.objects.**ProjectIssueManager**(*gl*: *Gitlab*, *parent*: *Optional[RESTObject]* = *None*)

Bases: *CRUDMixin, RESTManager*

**Object listing filters**

- *iids*
- *state*
- *labels*
- *milestone*
- *scope*
- *author\_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` (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectIssue*

Retrieve a single object.

### Parameters

- `id` (*Union*[*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`

**gitlab:** `Gitlab`

```
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`) –

**manager:** `RESTManager`

```
class gitlab.v4.objects.ProjectIssueNoteAwardEmojiManager(gl: Gitlab, parent:
                                                         Optional[RESTObject] = None)
```

Bases: `NoUpdateMixin`, `RESTManager`

#### Object Creation

Required attributes for object create:

- **name**

#### Object update

##### Parameters

- **gl** (`Gitlab`) –
- **parent** (`Optional[RESTObject]`) –

**get** (*id*: `Union[str, int]`, *lazy*: `bool` = `False`, *\*\*kwargs*: `Any`) → `ProjectIssueNoteAwardEmoji`

Retrieve a single object.

##### Parameters

- **id** (`Union[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](#)

gitlab: [Gitlab](#)

**class** gitlab.v4.objects.**ProjectIssueNoteManager**(gl: [Gitlab](#), parent: *Optional*[[RESTObject](#)] = 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** (*Optional* [[RESTObject](#)]) –

**get**(id: *Union*[*str*, *int*], lazy: *bool* = False, \*\*kwargs: *Any*) → [ProjectIssueNote](#)

Retrieve a single object.

**Parameters**

- **id** (*Union*[*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](#)

gitlab: [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectIssueResourceLabelEventManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[RESTObject]*) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → ProjectIssueResourceLabelEvent
```

Retrieve a single object.

##### Parameters

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectIssueResourceMilestoneEventManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

**Object Creation**

**Object update**

**Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[[RESTObject](#)]*) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → ProjectIssueResourceMilestoneEvent
```

Retrieve a single object.

**Parameters**

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectIssueResourceStateEventManager(gl: Gitlab, parent:
                                                                Optional[RESTObject] = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

**Object Creation**

**Object update**

**Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional*[[RESTObject](#)]) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → ProjectIssueResourceStateEvent
```

Retrieve a single object.

**Parameters**

- **id** (*Union*[*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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectIssuesStatisticsManager(gl: Gitlab, parent: Optional[RESTObject] =
                                                         None)
```

Bases: [GetWithoutIdMixin](#), [RESTManager](#)

**Object listing filters**

- **iids**



**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**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](#)

**gitlab:** [Gitlab](#)

**class** gitlab.v4.objects.**ProjectIterationManager**(*gl: [Gitlab](#), parent: Optional[[RESTObject](#)] = None*)

Bases: [ListMixin](#), [RESTManager](#)

**Object listing filters**

- state
- search
- include\_ancestors

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

**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: Optional[Callable[[...], Any]] = None, chunk\_size: int = 1024, \*, iterator: bool = False, \*\*kwargs: Any*) → *Optional[Union[bytes, Iterator[Any]]]*

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** (*Optional[Callable[[...], Any]]*) – 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

*Optional[Union[bytes, Iterator[Any]]]*

**artifacts**(*streamed: bool = False, action: Optional[Callable[[...], Any]] = None, chunk\_size: int = 1024, \*, iterator: bool = False, \*\*kwargs: Any*) → *Optional[Union[bytes, Iterator[Any]]]*

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** (*Optional[Callable[[...], Any]]*) – 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

*Optional[Union[bytes, Iterator[Any]]]*

**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

**manager:** [\*RESTManager\*](#)**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: Optional[Callable[[...], Any]] = 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** (Optional[Callable[[...], Any]]) – 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: Optional[RESTObject] = None)

Bases: [RetrieveMixin](#), [RESTManager](#)

**Object listing filters**

- scope

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#)) –

- **parent** (*Optional*[[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [ProjectJob](#)

Retrieve a single object.

#### Parameters

- **id** (*Union*[*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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectKeyManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional* [[RESTObject](#)]) –

**enable**(*key\_id: int, \*\*kwargs: Any*) → Union[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**

*Union[Dict[str, Any], Response]*

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → [ProjectKey](#)

Retrieve a single object.

**Parameters**

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

```
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: *Optional[RESTObject]* = 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** (*Optional[RESTObject]*) –

**get**(id: *Union[str, int]*, lazy: *bool* = False, \*\*kwargs: Any) → [\*ProjectLabel\*](#)

Retrieve a single object.

#### Parameters

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

**update**(*name: Optional[str], new\_data: Optional[Dict[str, Any]] = None, \*\*kwargs: Any*) → Dict[str, Any]

Update a Label on the server.

**Parameters**

- **name** (*Optional[str]*) – The name of the label
- **\*\*kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)
- **new\_data** (*Optional[Dict[str, Any]]*) –
- **\*\*kwargs** –

**Return type**

*Dict[str, Any]*

**class** gitlab.v4.objects.**ProjectManager**(*gl: [Gitlab](#), parent: Optional[[RESTObject](#)] = 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
- 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\_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`
- `remove_source_branch_after_merge`
- `repository_access_level`
- `repository_storage`
- `request_access_enabled`
- `resolve_outdated_diff_discussions`
- `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`
- `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`
- `container_expiration_policy_attributes`
- `container_registry_enabled`
- `default_branch`
- `description`
- `emails_disabled`
- `external_authorization_classification_label`
- `forking_access_level`
- `import_url`
- `issues_access_level`
- `issues_enabled`
- `jobs_enabled`
- `lfs_enabled`
- `merge_method`
- `merge_pipelines_enabled`
- `merge_requests_access_level`
- `merge_requests_enabled`
- `mirror_overwrites_diverged_branches`
- `mirror_trigger_builds`
- `mirror_user_id`

- mirror
- 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
- remove\_source\_branch\_after\_merge
- repository\_access\_level
- repository\_storage
- request\_access\_enabled
- resolve\_outdated\_diff\_discussions
- service\_desk\_enabled
- shared\_runners\_enabled
- show\_default\_award\_emojis
- snippets\_access\_level
- snippets\_enabled
- squash\_option
- suggestion\_commit\_message
- tag\_list
- topics
- visibility
- wiki\_access\_level
- wiki\_enabled
- issues\_template
- merge\_requests\_template

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → *Project*

Retrieve a single object.

#### Parameters

- **id** (Union[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*

**gitlab:** *Gitlab*

**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*: Optional[str] = None, *target\_namespace*: Optional[str] = None, *\*\*kwargs*: Any) → Union[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** (Optional[str]) – New repository name (Optional)
- **target\_namespace** (Optional[str]) – 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**

*Union[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)
    project = gl.projects.get(project.id)
print("BitBucket import complete")
```

**import\_github**(*personal\_access\_token: str, repo\_id: int, target\_namespace: str, new\_name: Optional[str]*  
*= None, \*\*kwargs: Any*) → *Union[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** (*Optional[str]*) – New repo name (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**

*Union[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)
    project = gl.projects.get(project.id)
print("Github import complete")
```

**import\_project**(*file*: str, *path*: str, *name*: Optional[str] = None, *namespace*: Optional[str] = None, *overwrite*: bool = False, *override\_params*: Optional[Dict[str, Any]] = None, *\*\*kwargs*: Any) → Union[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
- **namespace** (Optional[str]) – 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** (Optional[Dict[str, Any]]) – Set the specific settings for the project
- **\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)
- **name** (Optional[str]) –

**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**

*Union[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*) –

**manager:** [RESTManager](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectMemberAllManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[[RESTObject](#)]*) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → ProjectMemberAll
```

Retrieve a single object.

##### Parameters

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)



**class** gitlab.v4.objects.**ProjectMemberManager**(gl: [Gitlab](#), parent: *Optional*[[RESTObject](#)] = 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** (*Optional*[[RESTObject](#)]) –

**get**(id: *Union*[*str*, *int*], lazy: *bool* = False, \*\*kwargs: *Any*) → [ProjectMember](#)

Retrieve a single object.

#### Parameters

- **id** (*Union*[*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](#)

**gitlab:** [Gitlab](#)

**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: Optional[str] = None, \*\*kwargs: Any*) → *Dict[str, Any]*

Approve the merge request.

**Parameters**

- **sha** (*Optional[str]*) – 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*) → [ProjectMergeRequest](#)

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**

[ProjectMergeRequest](#)

**Return type**

[ProjectMergeRequest](#)

**changes**(*\*\*kwargs: Any*) → *Union[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**

*Union[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*

**merge**(merge\_commit\_message: Optional[str] = None, should\_remove\_source\_branch: Optional[bool] = None, merge\_when\_pipeline\_succeeds: Optional[bool] = None, \*\*kwargs: Any) → Dict[str, Any]

Accept the merge request.

**Parameters**

- **merge\_commit\_message** (*Optional[str]*) – Commit message
- **should\_remove\_source\_branch** (*Optional[bool]*) – If True, removes the source branch
- **merge\_when\_pipeline\_succeeds** (*Optional[bool]*) – 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*) → Union[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***Union[Dict[str, Any], Response]***notes:** *ProjectMergeRequestNoteManager***pipelines:** *ProjectMergeRequestPipelineManager***rebase**(*\*\*kwargs: Any*) → Union[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***Union[Dict[str, Any], Response]***resourcelabelevents:** *ProjectMergeRequestResourceLabelEventManager***resourcemilestoneevents:** *ProjectMergeRequestResourceMilestoneEventManager***resourcestateevents:** *ProjectMergeRequestResourceStateEventManager***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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectMergeRequestApprovalManager(gl: Gitlab, parent:
Optional[RESTObject] = None)
```

Bases: [GetWithoutIdMixin](#), [UpdateMixin](#), [RESTManager](#)

**Object Creation****Object update**

Required attributes for object update:

- **approvals\_required**

**Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[RESTObject]*) –

```
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](#)

gitlab: [Gitlab](#)

**set\_approvers**(*approvals\_required: int, approver\_ids: Optional[List[int]] = None, approver\_group\_ids: Optional[List[int]] = 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** (*Optional[List[int]]*) – User IDs that can approve MRs
- **approver\_group\_ids** (*Optional[List[int]]*) – 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*: *Optional*[[RESTObject](#)] = *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** (*Optional*[[RESTObject](#)]) –

**create**(*data*: *Optional*[*Dict*[*str*, *Any*]] = *None*, *\*\*kwargs*: *Any*) → [RESTObject](#)

Create a new object.

### Parameters

- **data** (*Optional*[*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` 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*: `Union[str, int]`, *lazy*: `bool = False`, *\*\*kwargs*: `Any`) → `ProjectMergeRequestApprovalRule`

Retrieve a single object.

**Parameters**

- **id** (`Union[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`

**gitlab:** `Gitlab`

**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`) –

**manager:** `RESTManager`

**class** `gitlab.v4.objects.ProjectMergeRequestApprovalStateManager`(*gl*: `Gitlab`, *parent*: `Optional[RESTObject] = None`)

Bases: `GetWithoutIdMixin`, `RESTManager`

**Object Creation****Object update****Parameters**

- **gl** (`Gitlab`) –
- **parent** (`Optional[RESTObject]`) –



**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*

**gitlab:** *Gitlab*

```
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*) –

**manager:** *RESTManager*

```
class gitlab.v4.objects.ProjectMergeRequestAwardEmojiManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: *NoUpdateMixin, RESTManager*

**Object Creation**

Required attributes for object create:

- **name**

**Object update**

**Parameters**

- **gl** (*Gitlab*) –
- **parent** (*Optional[RESTObject]*) –

**get**(id: Union[str, int], lazy: bool = False, \*\*kwargs: Any) → *ProjectMergeRequestAwardEmoji*

Retrieve a single object.

**Parameters**

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectMergeRequestDiffManager(gl: Gitlab, parent: Optional[RESTObject] =
                                                         None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[[RESTObject](#)]*) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → ProjectMergeRequestDiff
```

Retrieve a single object.

**Parameters**

- **id** (*Union[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](#)

gitlab: [Gitlab](#)

```
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: Optional[RESTObject] = 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** ([Optional](#) [[RESTObject](#)]) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → ProjectMergeRequestDiscussion
```

Retrieve a single object.

#### Parameters

- **id** ([Union](#) [[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](#)

**gitlab:** [Gitlab](#)

```
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](#)) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectMergeRequestDiscussionNoteManager(gl: Gitlab, parent: Optional[RESTObject] = 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** ([Optional](#) [[RESTObject](#)]) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → ProjectMergeRequestDiscussionNote
```

Retrieve a single object.

**Parameters**

- **id** ([Union](#)[[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](#)

**gitlab:** [Gitlab](#)

```
class gitlab.v4.objects.ProjectMergeRequestManager(gl: Gitlab, parent: Optional[RESTObject] =  
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:

- `assignee_id`
- `description`
- `target_project_id`
- `labels`
- `milestone_id`
- `remove_source_branch`
- `allow_maintainer_to_push`
- `squash`
- `reviewer_ids`

### 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** (*Optional* [[RESTObject](#)]) –

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → [ProjectMergeRequest](#)

Retrieve a single object.

### Parameters

- **id** (*Union[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`

**gitlab:** `Gitlab`

```
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`) –

**manager:** `RESTManager`

```
class gitlab.v4.objects.ProjectMergeRequestNoteAwardEmojiManager(gl: Gitlab, parent:
                                                                Optional[RESTObject] =
                                                                None)
```

Bases: `NoUpdateMixin`, `RESTManager`

**Object Creation**

Required attributes for object create:

- **name**

**Object update**

Parameters

- **gl** (`Gitlab`) –
- **parent** (`Optional[RESTObject]`) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → ProjectMergeRequestNoteAwardEmoji
```

Retrieve a single object.

Parameters

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

```
class gitlab.v4.objects.ProjectMergeRequestNoteManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [CRUDMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- body

**Object update**

Required attributes for object update:

- body

**Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[RESTObject]*) –

**get**(*id: Union[str, int]*, *lazy: bool = False*, *\*\*kwargs: Any*) → [ProjectMergeRequestNote](#)

Retrieve a single object.

**Parameters**

- **id** (*Union[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](#)



gitlab: [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectMergeRequestPipelineManager(gl: Gitlab, parent:
Optional[RESTObject] = None)
```

Bases: [CreateMixin](#), [ListMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[[RESTObject](#)]*) –

gitlab: [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectMergeRequestResourceLabelEventManager(gl: Gitlab, parent:
Optional[RESTObject] =
None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –

- **parent** (*Optional*[[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → *ProjectMergeRequestResourceLabelEvent*

Retrieve a single object.

#### Parameters

- **id** (*Union*[*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*

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectMergeRequestResourceMilestoneEventManager(gl: Gitlab, parent:  
                                                                Op-  
                                                                tional[RESTObject]  
                                                                = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional*[[RESTObject](#)]) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → *ProjectMergeRequestResourceMilestoneEvent*

Retrieve a single object.

#### Parameters

- **id** (Union[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*

**gitlab:** *Gitlab*

```
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) –

**manager:** *RESTManager*

```
class gitlab.v4.objects.ProjectMergeRequestResourceStateEventManager(gl: Gitlab, parent:
    Optional[RESTObject] =
    None)
```

Bases: *RetrieveMixin*, *RESTManager*

#### Object Creation

#### Object update

#### Parameters

- **gl** (*Gitlab*) –
- **parent** (Optional[*RESTObject*]) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → *ProjectMergeRequestResourceStateEvent*

Retrieve a single object.

#### Parameters

- **id** (Union[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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectMergeTrainManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [ListMixin](#), [RESTManager](#)

**Object listing filters**

- scope

**Object Creation****Object update****Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[[RESTObject](#)]*) –

**gitlab:** [Gitlab](#)

```
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*

**manager:** *RESTManager*

**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: *Optional[RESTObject]* = 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` (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [ProjectMilestone](#)

Retrieve a single object.

### Parameters

- `id` (*Union*[*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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** *RESTManager*

**class** gitlab.v4.objects.**ProjectNoteManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: *RetrieveMixin, RESTManager*

### Object Creation

Required attributes for object create:

- **body**

### Object update

#### Parameters

- **gl** (*Gitlab*) –
- **parent** (*Optional[RESTObject]*) –

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → *ProjectNote*

Retrieve a single object.

#### Parameters

- **id** (*Union[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*

**gitlab:** *Gitlab*

**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*) –

**manager:** *RESTManager*

```
class gitlab.v4.objects.ProjectNotificationSettingsManager(gl: Gitlab, parent:
                                                         Optional[RESTObject] = 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** (*Optional[RESTObject]*) –

```
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*

**gitlab:** *Gitlab*

```
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](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectPackageFileManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [DeleteMixin](#), [ListMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[RESTObject]*) –

**gitlab:** [Gitlab](#)

```
class gitlab.v4.objects.ProjectPackageManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [ListMixin](#), [GetMixin](#), [DeleteMixin](#), [RESTManager](#)

#### Object listing filters

- **order\_by**
- **sort**
- **package\_type**
- **package\_name**

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[RESTObject]*) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → ProjectPackage
```

Retrieve a single object.

## Parameters

- **id** (*Union[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

gitlab: *Gitlab*

```
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`) –

```
manager: RESTManager
```

```
class gitlab.v4.objects.ProjectPagesDomainManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [ProjectPagesDomain](#)

Retrieve a single object.

#### Parameters

- **id** (*Union*[*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](#)

**gitlab:** [Gitlab](#)

```
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*) → *Union*[*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

*Union*[*Dict*[*str*, *Any*], *Response*]

**jobs:** [ProjectPipelineJobManager](#)

**retry**(\*\*kwargs: Any) → Union[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**

Union[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) –

**manager:** [RESTManager](#)

**class** gitlab.v4.objects.**ProjectPipelineBridgeManager**(gl: [Gitlab](#), parent: Optional[[RESTObject](#)] =  
None)

Bases: [ListMixin](#), [RESTManager](#)

**Object listing filters**

- **scope**

**Object Creation**

**Object update**

**Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (Optional[[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

**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*) –

**manager:** [\*RESTManager\*](#)

**class** gitlab.v4.objects.**ProjectPipelineJobManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: [\*ListMixin\*](#), [\*RESTManager\*](#)

#### Object listing filters

- `scope`
- `include_retried`

#### Object Creation

#### Object update

##### Parameters

- `gl` ([\*Gitlab\*](#)) –
- `parent` (*Optional[RESTObject]*) –

**gitlab:** [\*Gitlab\*](#)

**class** gitlab.v4.objects.**ProjectPipelineManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: [\*RetrieveMixin\*](#), [\*CreateMixin\*](#), [\*DeleteMixin\*](#), [\*RESTManager\*](#)

#### Object listing filters

- `scope`
- `status`
- `ref`
- `sha`
- `yaml_errors`
- `name`
- `username`
- `order_by`
- `sort`

#### Object Creation

Required attributes for object create:

- `ref`

#### Object update

##### Parameters

- `gl` ([\*Gitlab\*](#)) –
- `parent` (*Optional[RESTObject]*) –

**create**(*data: Optional[Dict[str, Any]] = None, \*\*kwargs: Any*) → [\*ProjectPipeline\*](#)

Creates a new object.

##### Parameters

- **data** (*Optional*[*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)

**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*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, **\*\*kwargs**: *Any*) → [ProjectPipeline](#)

Retrieve a single object.

**Parameters**

- **id** (*Union*[*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](#)

**gitlab:** [Gitlab](#)

**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*) –

**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: Optional[RESTObject] = 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** (*Optional* [[RESTObject](#)]) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → *ProjectPipelineSchedule*

Retrieve a single object.

#### Parameters

- **id** (Union[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*

**gitlab:** *Gitlab*

```
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) –

**manager:** *RESTManager*

```
class gitlab.v4.objects.ProjectPipelineScheduleVariableManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional*[[RESTObject](#)]) –

gitlab: [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectPipelineTestReportManager(gl: Gitlab, parent: Optional[RESTObject]
                                                         = None)
```

Bases: [GetWithoutIdMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional*[[RESTObject](#)]) –

**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](#)

gitlab: [Gitlab](#)

```
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*) –

**manager:** [\*RESTManager\*](#)

**class** gitlab.v4.objects.**ProjectPipelineTestReportSummaryManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: [\*GetWithoutIdMixin\*](#), [\*RESTManager\*](#)

**Object Creation**

**Object update**

**Parameters**

- **gl** (*Gitlab*) –
- **parent** (*Optional[RESTObject]*) –

**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\*](#)

**gitlab:** [\*Gitlab\*](#)

**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*) –

**manager:** [\*RESTManager\*](#)

**class** gitlab.v4.objects.**ProjectPipelineVariableManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: [\*ListMixin\*](#), [\*RESTManager\*](#)

**Object Creation**

**Object update**

**Parameters**

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
class gitlab.v4.objects.ProjectProtectedBranch(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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectProtectedBranchManager(gl: Gitlab, parent: Optional [RESTObject] =
                                                         None)
```

Bases: [NoUpdateMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **name**

Optional attributes for object create:

- **push\_access\_level**
- **merge\_access\_level**
- **unprotect\_access\_level**
- **allowed\_to\_push**
- **allowed\_to\_merge**
- **allowed\_to\_unprotect**
- **code\_owner\_approval\_required**

#### Object update

##### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [ProjectProtectedBranch](#)

Retrieve a single object.

##### Parameters

- **id** (*Union*[*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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectProtectedEnvironmentManager(gl: Gitlab, parent:
Optional[RESTObject] = 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** (*Optional[[RESTObject](#)]*) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → ProjectProtectedEnvironment
```

Retrieve a single object.

**Parameters**

- **id** (*Union[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](#)

gitlab: [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectProtectedTagManager(gl: Gitlab, parent: Optional[RESTObject] =
                                                    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** (*Optional* [[RESTObject](#)]) –

**get**(*id: Union[str, int]*, *lazy: bool = False*, *\*\*kwargs: Any*) → [ProjectProtectedTag](#)

Retrieve a single object.

**Parameters**

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectPushRulesManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional* [[RESTObject](#)]) –

**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](#)

**gitlab:** [Gitlab](#)

```
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: Optional[RESTObject]
                                                         = None)
```

Bases: [DeleteMixin](#), [ListMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

**Parameters**

- **gl** (*Gitlab*) –
- **parent** (*Optional[RESTObject]*) –

**gitlab:** *Gitlab*

```
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*) –

**manager:** *RESTManager*

```
class gitlab.v4.objects.ProjectRegistryTagManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: *DeleteMixin, RetrieveMixin, RESTManager*

**Object Creation****Object update****Parameters**

- **gl** (*Gitlab*) –
- **parent** (*Optional[RESTObject]*) –

```
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*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → *ProjectRegistryTag*

Retrieve a single object.

#### Parameters

- **id** (Union[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*

**gitlab:** *Gitlab*

**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) –

**manager:** *RESTManager*

**class** gitlab.v4.objects.**ProjectReleaseLinkManager**(*gl*: *Gitlab*, *parent*: Optional[*RESTObject*] = None)

Bases: *CRUDMixin*, *RESTManager*

#### Object Creation

Required attributes for object create:

- name

- url

Optional attributes for object create:

- filepath
- link\_type

### Object update

Optional attributes for object update:

- name
- url
- filepath
- link\_type

### Parameters

- **gl** (*Gitlab*) –
- **parent** (*Optional[RESTObject]*) –

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → *ProjectReleaseLink*

Retrieve a single object.

### Parameters

- **id** (*Union[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*

**gitlab:** *Gitlab*

**class** gitlab.v4.objects.**ProjectReleaseManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: *CRUDMixin, RESTManager*

### Object listing filters

- order\_by
- sort
- include\_html\_description

### Object Creation

Required attributes for object create:

- tag\_name

- `description`

Optional attributes for object create:

- `name`
- `ref`
- `assets`

### Object update

Optional attributes for object update:

- `name`
- `description`
- `milestones`
- `released_at`

#### Parameters

- `gl` ([Gitlab](#)) –
- `parent` (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [ProjectRelease](#)

Retrieve a single object.

#### Parameters

- `id` (*Union*[*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](#)

**gitlab:** [Gitlab](#)

**class** `gitlab.v4.objects.ProjectRemoteMirror`(*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*) –

**manager:** [\*RESTManager\*](#)

```
class gitlab.v4.objects.ProjectRemoteMirrorManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [\*ListMixin\*](#), [\*CreateMixin\*](#), [\*UpdateMixin\*](#), [\*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` (*Optional* [[\*RESTObject\*](#)]) –

**gitlab:** [\*Gitlab\*](#)

```
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*) –

**manager:** [\*RESTManager\*](#)

```
class gitlab.v4.objects.ProjectRunnerManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional[RESTObject]*) –

gitlab: *Gitlab*

```
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*) –

manager: *RESTManager*

```
class gitlab.v4.objects.ProjectServiceManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: *ProjectIntegrationManager*

**Object Creation****Object update****Parameters**

- **gl** (*Gitlab*) –
- **parent** (*Optional[RESTObject]*) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → ProjectService
```

Retrieve a single object.

**Parameters**

- **id** (*Union[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*

gitlab: *Gitlab*

```
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: Optional[Callable[[...], Any]] = None, chunk_size: int = 1024, *,
        iterator: bool = False, **kwargs: Any) → Optional[Union[bytes, Iterator[Any]]]
```

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** ([Optional](#)[[Callable](#)[[...], [Any](#)]]) – 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

[Optional](#)[[Union](#)[[bytes](#), [Iterator](#)[[Any](#)]]]

**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](#)) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectSnippetAwardEmojiManager(gl: Gitlab, parent: Optional[RESTObject]
                                                         = None)
```

Bases: [NoUpdateMixin](#), [RESTManager](#)

### Object Creation

Required attributes for object create:

- `name`

### Object update

#### Parameters

- `gl` ([Gitlab](#)) –
- `parent` ([Optional\[RESTObject\]](#)) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → ProjectSnippetAwardEmoji
```

Retrieve a single object.

#### Parameters

- `id` ([Union\[str, int\]](#)) – 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](#)

**gitlab:** [Gitlab](#)

```
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: Optional[RESTObject]
                                                         = 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` (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [ProjectSnippetDiscussion](#)

Retrieve a single object.

#### Parameters

- `id` (*Union*[*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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectSnippetDiscussionNoteManager(gl: Gitlab, parent:
Optional[RESTObject] = 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` (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [ProjectSnippetDiscussionNote](#)

Retrieve a single object.

#### Parameters

- `id` (*Union*[*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](#)

**gitlab:** [Gitlab](#)

**class** `gitlab.v4.objects.ProjectSnippetManager`(*gl*: [Gitlab](#), *parent*: *Optional* [[RESTObject](#)] = *None*)

Bases: [CRUDMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- `title`
- `file_name`
- `content`
- `visibility`

Optional attributes for object create:

- `description`

#### Object update

Optional attributes for object update:

- `title`
- `file_name`

- content
- visibility
- description

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [ProjectSnippet](#)

Retrieve a single object.

#### Parameters

- **id** (*Union*[*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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectSnippetNoteAwardEmojiManager(gl: Gitlab, parent:
Optional[RESTObject] = None)
```

Bases: [NoUpdateMixin](#), [RESTManager](#)

### Object Creation

Required attributes for object create:

- **name**

### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → ProjectSnippetNoteAwardEmoji
```

Retrieve a single object.

#### Parameters

- **id** (*Union* [*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](#)

**gitlab:** [Gitlab](#)

```
class gitlab.v4.objects.ProjectSnippetNoteManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [CRUDMixin](#), [RESTManager](#)

### Object Creation

Required attributes for object create:

- **body**

### Object update

Required attributes for object update:

- **body**

#### Parameters

- **gl** ([Gitlab](#)) –

- **parent** (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [ProjectSnippetNote](#)

Retrieve a single object.

#### Parameters

- **id** (*Union*[*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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectStorageManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [GetWithoutIdMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**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***gitlab:** *Gitlab*

```
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*) –

**manager:** *RESTManager*

```
class gitlab.v4.objects.ProjectTagManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional[RESTObject]*) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → ProjectTag
```

Retrieve a single object.

**Parameters**

- **id** (*Union[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](#)

gitlab: [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectTriggerManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [CRUDMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- **description**

#### Object update

Required attributes for object update:

- **description**

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[[RESTObject](#)]*) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → ProjectTrigger
```

Retrieve a single object.

#### Parameters

- **id** (*Union[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

gitlab: [Gitlab](#)

```
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](#)) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectUserManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [ListMixin](#), [RESTManager](#)

#### Object listing filters

- `search`
- `skip_users`

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** ([Optional](#)[[RESTObject](#)]) –

gitlab: [Gitlab](#)

```
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](#)) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.ProjectVariableManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional* [[RESTObject](#)]) –

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → [ProjectVariable](#)

Retrieve a single object.

### Parameters

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

```
class gitlab.v4.objects.ProjectWiki(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*) –

**manager:** [RESTManager](#)

**class** gitlab.v4.objects.**ProjectWikiManager**(*gl: [Gitlab](#), parent: Optional[[RESTObject](#)] = 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** (*Optional[[RESTObject](#)]*) –

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → [ProjectWiki](#)

Retrieve a single object.

#### Parameters

- **id** (*Union[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](#)

gitlab: [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.RegistryRepositoryManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [GetMixin](#), [RESTManager](#)

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[[RESTObject](#)]*) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → RegistryRepository
```

Retrieve a single object.

##### Parameters

- **id** (*Union[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](#)

gitlab: [Gitlab](#)

```
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: Optional[str] = None, streamed: bool = False, action: Optional[Callable[[...], Any]] = None, chunk\_size: int = 1024, format: Optional[str] = None, path: Optional[str] = None, \*, iterator: bool = False, \*\*kwargs: Any*) → Optional[Union[bytes, Iterator[Any]]]

Return an archive of the repository.

**Parameters**

- **sha** (*Optional[str]*) – 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** (*Optional[Callable[[...], Any]]*) – Callable responsible of dealing with chunk of data
- **chunk\_size** (*int*) – Size of each chunk
- **format** (*Optional[str]*) – file format (tar.gz by default)
- **path** (*Optional[str]*) – 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**

*Optional[Union[bytes, Iterator[Any]]]*

**repository\_blob**(*sha: str, \*\*kwargs: Any*) → Union[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***Union[Dict[str, Any], Response]***repository\_compare**(*from\_*: str, *to*: str, *\*\*kwargs*: Any) → Union[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***Union[Dict[str, Any], Response]***repository\_contributors**(*\*\*kwargs*: Any) → Union[*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***Union[GitlabList, List[Dict[str, Any]]]***repository\_merge\_base**(*refs*: List[str], *\*\*kwargs*: Any) → Union[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**

*Union[Dict[str, Any], Response]*

**repository\_raw\_blob**(*sha: str, streamed: bool = False, action: Optional[Callable[[...], Any]] = None, chunk\_size: int = 1024, \*, iterator: bool = False, \*\*kwargs: Any*) → *Optional[Union[bytes, Iterator[Any]]]*

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** (*Optional[Callable[[...], Any]]*) – 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**

*Optional[Union[bytes, Iterator[Any]]]*

**repository\_tree**(*path: str = "", ref: str = "", recursive: bool = False, \*\*kwargs: Any*) → *Union[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**

*Union[GitlabList, List[Dict[str, Any]]]*

**update\_submodule**(*submodule: str, branch: str, commit\_sha: str, \*\*kwargs: Any*) → Union[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**

*Union[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*) –

**manager:** [RESTManager](#)

**class** gitlab.v4.objects.RunnerAllManager(*gl*: [Gitlab](#), *parent*: *Optional*[[RESTObject](#)] = *None*)

Bases: [ListMixin](#), [RESTManager](#)

#### Object listing filters

- *scope*
- *type*
- *status*
- *paused*
- *tag\_list*

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

**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*) –

**manager:** [RESTManager](#)

**class** gitlab.v4.objects.RunnerJobManager(*gl*: [Gitlab](#), *parent*: *Optional*[[RESTObject](#)] = *None*)

Bases: [ListMixin](#), [RESTManager](#)

#### Object listing filters

- *status*

#### Object Creation

#### Object update

##### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

**class** gitlab.v4.objects.**RunnerManager**(gl: [Gitlab](#), parent: *Optional*[[RESTObject](#)] = 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** (*Optional* [[RESTObject](#)]) –

**all**(scope: *Optional*[str] = None, **\*\*kwargs**: Any) → List[[Runner](#)]

List all the runners.

#### Parameters



- **scope** (*Optional[str]*) – 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: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → *Runner*

Retrieve a single object.

**Parameters**

- **id** (*Union[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*

**gitlab:** *Gitlab*

**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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.SharedProjectManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional[RESTObject]*) –

**gitlab:** [Gitlab](#)

```
class gitlab.v4.objects.SidekiqManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional* [[RESTObject](#)]) –

**compound\_metrics**(\*\*kwargs: Any) → Union[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**

*Union[Dict[str, Any], Response]*

**gitlab:** [Gitlab](#)

**job\_stats**(\*\*kwargs: Any) → Union[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**

*Union[Dict[str, Any], Response]*

**process\_metrics**(\*\*kwargs: Any) → Union[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**

*Union[Dict[str, Any], Response]*

**queue\_metrics**(\*\*kwargs: Any) → Union[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**

*Union[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: Optional[Callable[[...], Any]] = None, chunk\_size: int = 1024, \*, iterator: bool = False, \*\*kwargs: Any*) → *Optional[Union[bytes, Iterator[Any]]]*

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** (*Optional[Callable[[...], Any]]*) – 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**

*Optional[Union[bytes, Iterator[Any]]]*

**manager:** *RESTManager*

```
class gitlab.v4.objects.SnippetManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: *CRUDMixin, RESTManager*

**Object Creation**

Required attributes for object create:

- `title`
- `file_name`
- `content`

Optional attributes for object create:

- `lifetime`
- `visibility`

### Object update

Optional attributes for object update:

- `title`
- `file_name`
- `content`
- `visibility`

#### Parameters

- `gl` ([Gitlab](#)) –
- `parent` (*Optional* [[RESTObject](#)]) –

**get**(*id*: *Union*[*str*, *int*], *lazy*: *bool* = *False*, *\*\*kwargs*: *Any*) → [Snippet](#)

Retrieve a single object.

#### Parameters

- `id` (*Union*[*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](#)

**gitlab:** [Gitlab](#)

**public**(*\*\*kwargs*: *Any*) → *Union*[[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**

*Union[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*) –

**manager:** *RESTManager*

```
class gitlab.v4.objects.StarredProjectManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional[RESTObject]*) –

gitlab: [Gitlab](#)

**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) –

**manager:** [RESTManager](#)

**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: *Optional[RESTObject]* = None)

Bases: [ListMixin](#), [DeleteMixin](#), [RESTManager](#)

#### Object listing filters

- action
- author\_id
- project\_id
- state
- type

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional[RESTObject]*) –

gitlab: [Gitlab](#)

**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*) –

**manager:** [RESTManager](#)

**class** gitlab.v4.objects.**TopicManager**(*gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional[RESTObject]*) –



**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → *Topic*

Retrieve a single object.

#### Parameters

- **id** (Union[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*

**gitlab:** *Gitlab*

**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) → Union[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

Union[Dict[str, Any], Response]

**approve**(*\*\*kwargs*: Any) → Union[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**

*Union[Dict[str, Any], Response]*

**ban**(\*\*kwargs: Any) → Union[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**

*Union[Dict[str, Any], Response]*

**block**(\*\*kwargs: Any) → Optional[bool]

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**

*Optional[bool]*

**customattributes:** [\*UserCustomAttributeManager\*](#)

**deactivate**(\*\*kwargs: Any) → Union[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**

*Union[Dict[str, Any], Response]*

**emails:** [UserEmailManager](#)

**events:** [UserEventManager](#)

**follow**(\*\*kwargs: Any) → Union[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

Union[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) → Union[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

Union[Dict[str, Any], Response]

**starred\_projects:** [StarredProjectManager](#)

**status:** [UserStatusManager](#)

**unban**(\*\*kwargs: Any) → Union[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**

*Union[Dict[str, Any], Response]*

**unlock**(\*\*kwargs: Any) → Optional[bool]

Unblock the user.

**Parameters**

**\*\*kwargs** (Any) – Extra options to send to the server (e.g. sudo)

**Raises**

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUnlockError** – If the user could not be unblocked

**Returns**

Whether the user status has been changed

**Return type**

*Optional[bool]*

**unfollow**(\*\*kwargs: Any) → Union[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**

*Union[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*) –

**manager:** [RESTManager](#)

**class** gitlab.v4.objects.**UserActivitiesManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: [ListMixin](#), [RESTManager](#)

**Object Creation**

**Object update**

**Parameters**

- **gl** (*Gitlab*) –
- **parent** (*Optional[RESTObject]*) –

**gitlab:** [Gitlab](#)

**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*) –

**manager:** [RESTManager](#)

**class** gitlab.v4.objects.**UserCustomAttributeManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: [RetrieveMixin](#), [SetMixin](#), [DeleteMixin](#), [RESTManager](#)

**Object Creation**

**Object update**

**Parameters**

- **gl** (*Gitlab*) –
- **parent** (*Optional[RESTObject]*) –

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → [UserCustomAttribute](#)

Retrieve a single object.

**Parameters**

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

```
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](#)) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.UserEmailManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [RetrieveMixin](#), [CreateMixin](#), [DeleteMixin](#), [RESTManager](#)

**Object Creation**

Required attributes for object create:

- email

**Object update****Parameters**

- **gl** ([Gitlab](#)) –
- **parent** ([Optional](#) [[RESTObject](#)]) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → UserEmail
```

Retrieve a single object.

**Parameters**

- **id** ([Union](#)[[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](#)

**gitlab:** [Gitlab](#)

```
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) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.UserEventManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [EventManager](#)

#### Object listing filters

- action
- target\_type
- before
- after
- sort

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (Optional[[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
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) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.UserGPGKeyManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [RetrieveMixin](#), [CreateMixin](#), [DeleteMixin](#), [RESTManager](#)

#### Object Creation

Required attributes for object create:

- key

## Object update

### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (*Optional* [[RESTObject](#)]) –

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → [UserGPGKey](#)

Retrieve a single object.

### Parameters

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

```
class gitlab.v4.objects.UserIdentityProviderManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional* [[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

```
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*) –



**manager:** [RESTManager](#)

```
class gitlab.v4.objects.UserImpersonationTokenManager(gl: Gitlab, parent: Optional[RESTObject] =
                                                    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** (*Optional* [[RESTObject](#)]) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → UserImpersonationToken
```

Retrieve a single object.

##### Parameters

- **id** (*Union[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](#)

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [\*RESTManager\*](#)

**class** gitlab.v4.objects.**UserKeyManager**(*gl: Gitlab, parent: Optional[RESTObject] = None*)

Bases: [\*RetrieveMixin\*](#), [\*CreateMixin\*](#), [\*DeleteMixin\*](#), [\*RESTManager\*](#)

#### Object Creation

Required attributes for object create:

- **title**
- **key**

#### Object update

##### Parameters

- **gl** ([\*Gitlab\*](#)) –
- **parent** (*Optional[[\*RESTObject\*](#)]*) –

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → [\*UserKey\*](#)

Retrieve a single object.

##### Parameters

- **id** (*Union[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\*](#)

**gitlab:** [\*Gitlab\*](#)

**class** gitlab.v4.objects.**UserManager**(*gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional* [[RESTObject](#)]) –

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → [User](#)

Retrieve a single object.

#### Parameters

- **id** (*Union[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](#)

gitlab: [Gitlab](#)

```
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](#)) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.UserMembershipManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [RetrieveMixin](#), [RESTManager](#)

#### Object listing filters

- **type**

#### Object Creation

#### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** ([Optional](#) [[RESTObject](#)]) –

```
get(id: Union[str, int], lazy: bool = False, **kwargs: Any) → UserMembership
```

Retrieve a single object.

#### Parameters

- **id** ([Union](#)[[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](#)

gitlab: [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.UserPersonalAccessTokenManager(gl: Gitlab, parent: Optional[RESTObject] =  
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** (*Optional[[RESTObject](#)]*) –

**gitlab:** [Gitlab](#)

```
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*) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.UserProjectManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (*Optional* [[RESTObject](#)]) –

**gitlab:** [Gitlab](#)

**list**(\*\*kwargs: Any) → Union[[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

Union[[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) –

**manager:** [RESTManager](#)

```
class gitlab.v4.objects.UserStatusManager(gl: Gitlab, parent: Optional[RESTObject] = None)
```

Bases: [GetWithoutIdMixin](#), [RESTManager](#)

#### Object Creation

##### Object update

#### Parameters

- **gl** ([Gitlab](#)) –
- **parent** (Optional[[RESTObject](#)]) –

**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`**gitlab:** `Gitlab`

```
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`) –

**manager:** `RESTManager`

```
class gitlab.v4.objects.VariableManager(gl: Gitlab, parent: Optional[RESTObject] = 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** (`Optional[RESTObject]`) –

**get**(*id*: Union[str, int], *lazy*: bool = False, *\*\*kwargs*: Any) → Variable

Retrieve a single object.

#### Parameters

- **id** (Union[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

**gitlab:** *Gitlab*

## Module contents

### 6.6.3 Submodules

#### 6.6.4 gitlab.base module

**class** gitlab.base.RESTManager(*gl*: *Gitlab*, *parent*: Optional[RESTObject] = 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** (Optional[RESTObject]) –

**gitlab:** *Gitlab*

**property** parent\_attrs: Optional[Dict[str, Any]]

**property** path: Optional[str]

**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:** *Optional[Union[int, str]]*

Ensure that the ID is url-encoded so that it can be safely used in a URL path

**get\_id()** → *Optional[Union[int, str]]*

Returns the id of the resource.

#### Return type

*Optional[Union[int, str]]*

**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:** `Optional[int]`

The next page number.

If None, the current page is the last.

**property per\_page:** `Optional[int]`

The number of items per page.

**property prev\_page:** `Optional[int]`

The previous page number.

If None, the current page is the first.

**property total:** `Optional[int]`

The total number of items.

**property total\_pages:** `Optional[int]`

The total number of pages.

## 6.6.5 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*: `Optional[Exception]` = `None`) → `None`

**Parameters**

- **msg** (`str`) –
- **e** (`Optional[Exception]`) –

**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: module)` → `Type[RESTObject]`**Parameters**

- `gitlab_resource` (*str*) –
- `namespace` (*module*) –

**Return type***Type[RESTObject]*`gitlab.cli.main()` → None**Return type**

None

`gitlab.cli.register_custom_action(cls_names: Union[str, Tuple[str, ...]], mandatory: Tuple[str, ...] = (), optional: Tuple[str, ...] = (), custom_action: Optional[str] = None)` → `Callable[[__F], __F]`

**Parameters**

- `cls_names` (*Union[str, Tuple[str, ...]]*) –
- `mandatory` (*Tuple[str, ...]*) –
- `optional` (*Tuple[str, ...]*) –
- `custom_action` (*Optional[str]*) –

**Return type***Callable[[\_\_F], \_\_F]*

## 6.6.6 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: Optional[str] = None, config_files: Optional[List[str]] = None)`

Bases: `object`**Parameters**

- `gitlab_id` (*Optional[str]*) –

- `config_files` (*Optional[List[str]]*) –

**exception** `gitlab.config.GitlabDataError`

Bases: `ConfigError`

**exception** `gitlab.config.GitlabIDError`

Bases: `ConfigError`

### 6.6.7 gitlab.const module

**class** `gitlab.const.AccessLevel(value)`

Bases: `IntEnum`

An enumeration.

**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)`

Bases: `GitlabEnum`

An enumeration.

**CUSTOM:** `str = 'custom'`

**DISABLED:** `str = 'disabled'`

**GLOBAL:** `str = 'global'`

**MENTION:** `str = 'mention'`

**PARTICIPATING:** `str = 'participating'`

**WATCH:** `str = 'watch'`

**class** `gitlab.const.SearchScope(value)`

Bases: `GitlabEnum`

An enumeration.

**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)
    Bases: GitlabEnum
    An enumeration.

    INTERNAL: str = 'internal'

    PRIVATE: str = 'private'

    PUBLIC: str = 'public'

```

### 6.6.8 gitlab.exceptions module

```

exception gitlab.exceptions.GitlabActivateError(error_message: Union[str, bytes] = "", response_code:
Optional[int] = None, response_body:
Optional[bytes] = None)

```

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** ([Union\[str, bytes\]](#)) –
- **response\_code** ([Optional\[int\]](#)) –
- **response\_body** ([Optional\[bytes\]](#)) –

#### Return type

None

```

exception gitlab.exceptions.GitlabAttachFileError(error_message: Union[str, bytes] = "",
response_code: Optional[int] = None,
response_body: Optional[bytes] = None)

```

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** ([Union\[str, bytes\]](#)) –
- **response\_code** ([Optional\[int\]](#)) –
- **response\_body** ([Optional\[bytes\]](#)) –

#### Return type

None

**exception** `gitlab.exceptions.GitlabAuthenticationError`(*error\_message*: *Union[str, bytes]* = "",  
*response\_code*: *Optional[int]* = None,  
*response\_body*: *Optional[bytes]* = None)

Bases: [\*GitlabError\*](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.exceptions.GitlabBanError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*:  
*Optional[int]* = None, *response\_body*: *Optional[bytes]* =  
None)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.exceptions.GitlabBlockError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*:  
*Optional[int]* = None, *response\_body*: *Optional[bytes]* =  
None)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.exceptions.GitlabBuildCancelError`(*error\_message*: *Union[str, bytes]* = "",  
*response\_code*: *Optional[int]* = None,  
*response\_body*: *Optional[bytes]* = None)

Bases: [\*GitlabCancelError\*](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None



**exception** `gitlab.exceptions.GitlabBuildEraseError`(*error\_message*: *Union[str, bytes]* = "",  
*response\_code*: *Optional[int]* = None,  
*response\_body*: *Optional[bytes]* = None)

Bases: [\*GitlabRetryError\*](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** `gitlab.exceptions.GitlabBuildPlayError`(*error\_message*: *Union[str, bytes]* = "",  
*response\_code*: *Optional[int]* = None,  
*response\_body*: *Optional[bytes]* = None)

Bases: [\*GitlabRetryError\*](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** `gitlab.exceptions.GitlabBuildRetryError`(*error\_message*: *Union[str, bytes]* = "",  
*response\_code*: *Optional[int]* = None,  
*response\_body*: *Optional[bytes]* = None)

Bases: [\*GitlabRetryError\*](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** `gitlab.exceptions.GitlabCancelError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*:  
*Optional[int]* = None, *response\_body*: *Optional[bytes]*  
= None)

Bases: [\*GitlabOperationError\*](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

```
exception gitlab.exceptions.GitlabCherryPickError(error_message: Union[str, bytes] = "",  
                                                response_code: Optional[int] = None,  
                                                response_body: Optional[bytes] = None)
```

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

```
exception gitlab.exceptions.GitlabCiLintError(error_message: Union[str, bytes] = "", response_code:  
                                              Optional[int] = None, response_body: Optional[bytes]  
                                              = None)
```

Bases: [GitlabError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

```
exception gitlab.exceptions.GitlabConnectionError(error_message: Union[str, bytes] = "",  
                                                  response_code: Optional[int] = None,  
                                                  response_body: Optional[bytes] = None)
```

Bases: [GitlabError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

```
exception gitlab.exceptions.GitlabCreateError(error_message: Union[str, bytes] = "", response_code:  
                                              Optional[int] = None, response_body: Optional[bytes]  
                                              = None)
```

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

```
exception gitlab.exceptions.GitlabDeactivateError(error_message: Union[str, bytes] = ",  

response_code: Optional[int] = None,  

response_body: Optional[bytes] = None)
```

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

```
exception gitlab.exceptions.GitlabDeleteError(error_message: Union[str, bytes] = ", response_code:  

Optional[int] = None, response_body: Optional[bytes]  

= None)
```

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

```
exception gitlab.exceptions.GitlabError(error_message: Union[str, bytes] = ", response_code:  

Optional[int] = None, response_body: Optional[bytes] = None)
```

Bases: [Exception](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

```
exception gitlab.exceptions.GitlabFollowError(error_message: Union[str, bytes] = ", response_code:  

Optional[int] = None, response_body: Optional[bytes]  

= None)
```

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** gitlab.exceptions.**GitlabGetError**(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.exceptions.**GitlabGroupTransferError**(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.exceptions.**GitlabHeadError**(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.exceptions.**GitlabHousekeepingError**(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.exceptions.GitlabHttpError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** gitlab.exceptions.GitlabImportError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** gitlab.exceptions.GitlabInvitationError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** gitlab.exceptions.GitlabJobCancelError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabCancelError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** `gitlab.exceptions.GitlabJobEraseError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = *None*, *response\_body*: *Optional[bytes]* = *None*)

Bases: [\*GitlabRetryError\*](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.exceptions.GitlabJobPlayError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = *None*, *response\_body*: *Optional[bytes]* = *None*)

Bases: [\*GitlabRetryError\*](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.exceptions.GitlabJobRetryError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = *None*, *response\_body*: *Optional[bytes]* = *None*)

Bases: [\*GitlabRetryError\*](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** `gitlab.exceptions.GitlabLicenseError`(*error\_message*: *Union[str, bytes]* = "", *response\_code*: *Optional[int]* = *None*, *response\_body*: *Optional[bytes]* = *None*)

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.exceptions.**GitlabListError**(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** gitlab.exceptions.**GitlabMRApprovalError**(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** gitlab.exceptions.**GitlabMRClosedError**(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** gitlab.exceptions.**GitlabMRForbiddenError**(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

```
exception gitlab.exceptions.GitlabMROnBuildSuccessError(error_message: Union[str, bytes] = "",  
                                                    response_code: Optional[int] = None,  
                                                    response_body: Optional[bytes] = None)
```

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

```
exception gitlab.exceptions.GitlabMRRebaseError(error_message: Union[str, bytes] = "", response_code:  
                                              Optional[int] = None, response_body:  
                                              Optional[bytes] = None)
```

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

```
exception gitlab.exceptions.GitlabMarkdownError(error_message: Union[str, bytes] = "", response_code:  
                                              Optional[int] = None, response_body:  
                                              Optional[bytes] = None)
```

Bases: [\*GitlabOperationError\*](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

```
exception gitlab.exceptions.GitlabOperationError(error_message: Union[str, bytes] = "",  
                                              response_code: Optional[int] = None,  
                                              response_body: Optional[bytes] = None)
```

Bases: [\*GitlabError\*](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None



```
exception gitlab.exceptions.GitlabOwnershipError(error_message: Union[str, bytes] = "",
                                                  response_code: Optional[int] = None,
                                                  response_body: Optional[bytes] = None)
```

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

#### Return type

None

```
exception gitlab.exceptions.GitlabParsingError(error_message: Union[str, bytes] = "", response_code:
                                                Optional[int] = None, response_body: Optional[bytes]
                                                = None)
```

Bases: [GitlabError](#)

#### Parameters

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

#### Return type

None

```
exception gitlab.exceptions.GitlabPipelineCancelError(error_message: Union[str, bytes] = "",
                                                         response_code: Optional[int] = None,
                                                         response_body: Optional[bytes] = None)
```

Bases: [GitlabCancelError](#)

#### Parameters

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

#### Return type

None

```
exception gitlab.exceptions.GitlabPipelinePlayError(error_message: Union[str, bytes] = "",
                                                       response_code: Optional[int] = None,
                                                       response_body: Optional[bytes] = None)
```

Bases: [GitlabRetryError](#)

#### Parameters

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

#### Return type

None

```
exception gitlab.exceptions.GitlabPipelineRetryError(error_message: Union[str, bytes] = "",  
                                                    response_code: Optional[int] = None,  
                                                    response_body: Optional[bytes] = None)
```

Bases: [GitlabRetryError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

```
exception gitlab.exceptions.GitlabProjectDeployKeyError(error_message: Union[str, bytes] = "",  
                                                         response_code: Optional[int] = None,  
                                                         response_body: Optional[bytes] = None)
```

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

```
exception gitlab.exceptions.GitlabPromoteError(error_message: Union[str, bytes] = "", response_code:  
                                                Optional[int] = None, response_body: Optional[bytes]  
                                                = None)
```

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

```
exception gitlab.exceptions.GitlabProtectError(error_message: Union[str, bytes] = "", response_code:  
                                                Optional[int] = None, response_body: Optional[bytes]  
                                                = None)
```

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

**exception** gitlab.exceptions.**GitlabRenderError**(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

#### Return type

None

**exception** gitlab.exceptions.**GitlabRepairError**(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

#### Return type

None

**exception** gitlab.exceptions.**GitlabRestoreError**(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

#### Return type

None

**exception** gitlab.exceptions.**GitlabRetryError**(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

#### Return type

None

**exception** gitlab.exceptions.**GitlabRevertError**(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

**exception** gitlab.exceptions.**GitlabSearchError**(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

**exception** gitlab.exceptions.**GitlabSetError**(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

**exception** gitlab.exceptions.**GitlabStopError**(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

```
exception gitlab.exceptions.GitlabSubscribeError(error_message: Union[str, bytes] = "",  
response_code: Optional[int] = None,  
response_body: Optional[bytes] = None)
```

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

```
exception gitlab.exceptions.GitlabTimeTrackingError(error_message: Union[str, bytes] = "",  
response_code: Optional[int] = None,  
response_body: Optional[bytes] = None)
```

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

```
exception gitlab.exceptions.GitlabTodoError(error_message: Union[str, bytes] = "", response_code:  
Optional[int] = None, response_body: Optional[bytes] =  
None)
```

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

```
exception gitlab.exceptions.GitlabTransferProjectError(error_message: Union[str, bytes] = "",  
response_code: Optional[int] = None,  
response_body: Optional[bytes] = None)
```

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** `gitlab.exceptions.GitlabUnbanError`(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

**exception** `gitlab.exceptions.GitlabUnblockError`(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

**exception** `gitlab.exceptions.GitlabUnfollowError`(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

**exception** `gitlab.exceptions.GitlabUnsubscribeError`(*error\_message*: Union[str, bytes] = "", *response\_code*: Optional[int] = None, *response\_body*: Optional[bytes] = None)

Bases: [`GitlabOperationError`](#)

**Parameters**

- **error\_message** (Union[str, bytes]) –
- **response\_code** (Optional[int]) –
- **response\_body** (Optional[bytes]) –

**Return type**

None

**exception** gitlab.exceptions.GitlabUpdateError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** gitlab.exceptions.GitlabUploadError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** gitlab.exceptions.GitlabUserApproveError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** gitlab.exceptions.GitlabUserRejectError(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

#### Parameters

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

#### Return type

None

**exception** gitlab.exceptions.**GitlabVerifyError**(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabOperationError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

**exception** gitlab.exceptions.**RedirectError**(*error\_message: Union[str, bytes] = "", response\_code: Optional[int] = None, response\_body: Optional[bytes] = None*)

Bases: [GitlabError](#)

**Parameters**

- **error\_message** (*Union[str, bytes]*) –
- **response\_code** (*Optional[int]*) –
- **response\_body** (*Optional[bytes]*) –

**Return type**

None

## 6.6.9 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*: Optional[Dict[str, Any]] = None, *\*\*kwargs*: Any) → [RESTObject](#)

Create a new object.

#### Parameters

- **data** (Optional[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

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*: Optional[Union[int, str]] = None, *\*\*kwargs*: Any) → None

Delete an object on the server.

#### Parameters

- **id** (Optional[Union[int, str]]) – 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: Optional[Callable[[bytes], None]] = None, chunk\_size: int = 1024, \*, iterator: bool = False, \*\*kwargs: Any*) → *Optional[Union[bytes, Iterator[Any]]]*

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** (*Optional[Callable[[bytes], 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**

*Optional[Union[bytes, Iterator[Any]]]*

**manager:** [\*RESTManager\*](#)

**class** gitlab.mixins.GetMixin

Bases: HeadMixin, object

**get**(*id: Union[str, int], lazy: bool = False, \*\*kwargs: Any*) → [\*RESTObject\*](#)

Retrieve a single object.

**Parameters**

- **id** (*Union[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.GetWithoutIdMixinBases: [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.ListMixinBases: [HeadMixin](#), object**gitlab:** [Gitlab](#)**list**(\*\*kwargs: Any) → Union[[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**Union[[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.RetrieveMixin

Bases: [ListMixin](#), [GetMixin](#)

**gitlab:** [Gitlab](#)

**class** gitlab.mixins.SaveMixin

Bases: object

Mixin for RESTObject's that can be updated.

**manager:** [RESTManager](#)

**save**(\*\*kwargs: Any) → Optional[Dict[str, Any]]

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**

*Optional[Dict[str, Any]]*

**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: Optional[Union[int, str]] = None, new\_data: Optional[Dict[str, Any]] = None, \*\*kwargs: Any) → Dict[str, Any]

Update an object on the server.

**Parameters**

- **id** (Optional[Union[int, str]]) – ID of the object to update (can be None if not required)
- **new\_data** (Optional[Dict[str, Any]]) – 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]*

## 6.6.10 gitlab.utils module

**class** gitlab.utils.**EncodedId**(*value: Union[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** (*Union[str, int, EncodedId]*) –

**Return type**

*EncodedId*

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.**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: Optional[Callable[[bytes], None]], chunk\_size: int, \*, iterator: bool*) → Optional[Union[bytes, Iterator[Any]]]

**Parameters**

- **response** (*Response*) –
- **streamed** (*bool*) –
- **action** (*Optional[Callable[[bytes], None]]*) –
- **chunk\_size** (*int*) –
- **iterator** (*bool*) –

**Return type***Optional[Union[bytes, Iterator[Any]]]*

`gitlab.utils.warn(message: str, *, category: Optional[Type[Warning]] = None, source: Optional[Any] = 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** (*Optional[Type[Warning]]*) –
- **source** (*Optional[Any]*) –

**Return type**

None

## 6.7 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.

### 6.7.1 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] [--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]
    application
    application-appearance
    application-settings
    audit-event
    broadcast-message
```

(continues on next page)

(continued from previous page)

```
ci-lint
current-user
current-user-email
current-user-gpg-key
current-user-key
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
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-label
group-member
group-member-all
group-merge-request
group-milestone
group-notification-settings
group-package
group-project
group-push-rules
group-runner
```

(continues on next page)

(continued from previous page)

```
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
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
```

(continues on next page)

(continued from previous page)

```
project-issue-note-award-emoji
project-issue-resource-label-event
project-issue-resource-milestone-event
project-issue-resource-state-event
project-issues-statistics
project-job
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
project-merge-request-discussion
project-merge-request-discussion-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-train
project-milestone
project-note
project-notification-settings
project-package
project-package-file
project-pages-domain
project-pipeline
project-pipeline-bridge
project-pipeline-job
project-pipeline-schedule
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-runner
project-service
project-snippet
project-snippet-award-emoji
```

(continues on next page)

(continued from previous page)

```
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
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>  
Whether SSL certificates should be validated. [env var: GITLAB\_SSL\_VERIFY]

**--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]

### 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] [--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.  
**--all**  
Deprecated. Use `--get-all` instead.

### 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_
```

(continues on next page)

(continued from previous page)

```

↪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]

```

**-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 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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 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]
                                     [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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]
                                         [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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] [--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.

### **--all**

Deprecated. Use `--get-all` instead.

## 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] [--page PAGE]
                        [--per-page PER_PAGE] [--get-all] [--all]
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--action** <action>

**--target-type** <target\_type>

**--before** <before>

**--after** <after>

**--sort** <sort>

**--page** <page>

**--per-page** <per\_page>



**--get-all**

Return all items from the server, without pagination.

**--all**

Deprecated. Use `--get-all` instead.

**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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

## 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] [--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.

### --all

Deprecated. Use `--get-all` instead.

## gitlab group

```
usage: gitlab group [-h]
  list
  get
  create
  update
  delete
  transfer-project
  transfer
  search
  add-ldap-group-link
  delete-ldap-group-link
```

(continues on next page)

(continued from previous page)

```

ldap-sync
share
unshare
...

```

**-h, --help**

show this help message and exit

**gitlab group add-ldap-group-link**

```

usage: gitlab group add-ldap-group-link [-h] --id ID --cn CN --group-access
GROUP_ACCESS --provider PROVIDER

```

**-h, --help**

show this help message and exit

**--id** <id>**--cn** <cn>**--group-access** <group\_access>**--provider** <provider>**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 delete-ldap-group-link

```
usage: gitlab group delete-ldap-group-link [-h] --id ID --cn CN
                                           [--provider PROVIDER]
```

**-h, --help**  
show this help message and exit

**--id** <id>

**--cn** <cn>

**--provider** <provider>

### 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]
                        [--top-level-only TOP_LEVEL_ONLY] [--page PAGE]
                        [--per-page PER_PAGE] [--get-all] [--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.  
**--all**  
Deprecated. Use `--get-all` instead.

### 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]
```

(continues on next page)

(continued from previous page)

```

    [--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]
```

**-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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### gitlab group-access-token

```
usage: gitlab group-access-token [-h] {list,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 list

```
usage: gitlab group-access-token list [-h] [--sudo SUDO] --group-id GROUP_ID
                                         [--page PAGE] [--per-page PER_PAGE]
                                         [--get-all] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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]
                                [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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]
[--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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]
                                [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--board-id** <board\_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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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]
                                         [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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]
                                       [--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.

**--all**  
Deprecated. Use `--get-all` instead.

## 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]
                                           [--owned OWNED]
                                           [--with-custom-attributes WITH_CUSTOM_
↪ATTRIBUTES]
                                           [--min-access-level MIN_ACCESS_LEVEL]
                                           [--page PAGE] [--per-page PER_PAGE]
                                           [--get-all] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

**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-iid EPIC_IID
                                                --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-iid** <epic\_iid>

**--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-iid EPIC_IID
                                                --discussion-id DISCUSSION_ID
                                                --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--epic-iid** <epic\_iid>

**--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-iid EPIC_IID
                                             --discussion-id DISCUSSION_ID
                                             --id ID
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--epic-iid** <epic\_iid>

**--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-iid EPIC_IID
                                                  --discussion-id DISCUSSION_ID
                                                  --id ID --body BODY
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>

**--epic-iid** <epic\_iid>

**--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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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-iid EPIC_IID --body BODY
                                     [--created-at CREATED_AT]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--group-id** <group\_id>  
**--epic-iid** <epic\_iid>  
**--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-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-note get

```
usage: gitlab group-epic-note 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-note list

```
usage: gitlab group-epic-note list [-h] [--sudo SUDO] --group-id GROUP_ID
                                   --epic-iid EPIC_IID [--page PAGE]
                                   [--per-page PER_PAGE] [--get-all] [--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.  
**--all**  
 Deprecated. Use `--get-all` instead.

### gitlab group-epic-note update

```
usage: gitlab group-epic-note update [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --epic-iid EPIC_IID --id ID --body BODY
```

**-h, --help**  
 show this help message and exit  
**--sudo** <sudo>  
**--group-id** <group\_id>  
**--epic-iid** <epic\_iid>  
**--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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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] [--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.  
**--all**  
Deprecated. Use `--get-all` instead.

### 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]
                               [--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.

**--all**

Deprecated. Use `--get-all` instead.

**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]
```

```
-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>
```

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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]
                               [--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] [--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>

**--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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]
                               [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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-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] [--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.

### **--all**

Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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] [--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.
--all
    Deprecated. Use --get-all instead.
```

### 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] [--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.  
**--all**  
Deprecated. Use `--get-all` instead.

### 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] [--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.
--all
    Deprecated. Use --get-all instead.

```

## 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] [--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.
--all
    Deprecated. Use --get-all instead.
```

### 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]
                                     [--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] [--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.

### --all

Deprecated. Use `--get-all` instead.

## 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]
                                   [--page PAGE] [--per-page PER_PAGE]
                                   [--get-all] [--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.  
**--all**  
Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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] [--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] [--all]
```

### **-h, --help**

show this help message and exit

**--sudo** <sudo>

**--state** <state>

**--labels** <labels>

**--milestone** <milestone>

**--scope** <scope>

**--author-id** <author\_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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

### **--all**

Deprecated. Use `--get-all` instead.

## 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]
                          [--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.

### --all

Deprecated. Use --get-all instead.

## 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]
```

(continues on next page)

(continued from previous page)

```
[--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] [--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.
--all
    Deprecated. Use --get-all instead.

```

### 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]
                             [--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.

### **--all**

Deprecated. Use `--get-all` instead.

## 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]
```

(continues on next page)



(continued from previous page)

`↩REQUEST]``[--reassign-merge-request REASSIGN_MERGE_``[--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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

**gitlab personal-access-token**

```
usage: gitlab personal-access-token [-h] {list,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 list**

```
usage: gitlab personal-access-token list [-h] [--sudo SUDO]
                                         [--user-id USER_ID] [--page PAGE]
                                         [--per-page PER_PAGE] [--get-all]
                                         [--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.

**--all**

Deprecated. Use `--get-all` instead.

## gitlab project

```
usage: gitlab project [-h]
    list
    get
    create
    update
    delete
    artifacts
    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
    upload
    restore
    snapshot
    search
    mirror-pull
    transfer
    transfer-project
    artifact
    ...
```

### **-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 artifact

```
usage: gitlab project artifact [-h] --id ID --ref-name REF_NAME
                               --artifact-path ARTIFACT_PATH --job JOB
```

**-h, --help**  
show this help message and exit

**--id** <id>

**--ref-name** <ref\_name>

**--artifact-path** <artifact\_path>

**--job** <job>

## gitlab project artifacts

```
usage: gitlab project artifacts [-h] --id ID --ref-name REF_NAME --job JOB
                               [--job-token JOB_TOKEN]
```

**-h, --help**  
show this help message and exit

**--id** <id>

**--ref-name** <ref\_name>

**--job** <job>

**--job-token** <job\_token>

## gitlab project create

```
usage: gitlab project create [-h] [--sudo SUDO] [--name NAME] [--path PATH]
                             [--allow-merge-on-skipped-pipeline ALLOW_MERGE_ON_SKIPPED_
↳PIPELINE]
                             [--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]
```

(continues on next page)

(continued from previous page)

```

[--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]
[--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]
[--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>

**--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-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>
--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>
--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]
                        [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### gitlab project mirror-pull

```
usage: gitlab project mirror-pull [-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 transfer-project

```
usage: gitlab project transfer-project [-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]
                                [--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]
```

(continues on next page)

(continued from previous page)

```

[--container-expiration-policy-attributes CONTAINER_
↪ EXPIRATION_POLICY_ATTRIBUTES]
[--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]
[--import-url IMPORT_URL]
[--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-overwrites-diverged-branches MIRROR_OVERWRITES_
↪ DIVERGED_BRANCHES]
[--mirror-trigger-builds MIRROR_TRIGGER_BUILDS]
[--mirror-user-id MIRROR_USER_ID]
[--mirror MIRROR] [--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]
[--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]
[--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]
[--squash-option SQUASH_OPTION]
[--suggestion-commit-message SUGGESTION_COMMIT_MESSAGE]
[--tag-list TAG_LIST] [--topics TOPICS]
[--visibility VISIBILITY]

```

(continues on next page)



(continued from previous page)

```

[--wiki-access-level WIKI_ACCESS_LEVEL]
[--wiki-enabled WIKI_ENABLED]
[--issues-template ISSUES_TEMPLATE]
[--merge-requests-template MERGE_REQUESTS_TEMPLATE]

```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>

**--allow-merge-on-skipped-pipeline** <allow\_merge\_on\_skipped\_pipeline>

**--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>

**--container-expiration-policy-attributes** <container\_expiration\_policy\_attributes>

**--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>

**--import-url** <import\_url>

**--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-overwrites-diverged-branches <mirror_overwrites_diverged_branches>
--mirror-trigger-builds <mirror_trigger_builds>
--mirror-user-id <mirror_user_id>
--mirror <mirror>
--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>
--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>
--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>
--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>
--issues-template <issues_template>
--merge-requests-template <merge_requests_template>

```

### 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]
                                           [--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.

**--all**

Deprecated. Use `--get-all` instead.

## gitlab project-access-token

```
usage: gitlab project-access-token [-h] {list,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 list

```
usage: gitlab project-access-token list [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID [--page PAGE]
                                           [--per-page PER_PAGE] [--get-all]
                                           [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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]
```

**-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>

### 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]
                                           [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**

Deprecated. Use --get-all instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--board-id** <board\_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] [--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.

### --all

Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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
                                   [--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] [--all]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--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.  
**--all**  
 Deprecated. Use `--get-all` instead.

### 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] [--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.
--all
    Deprecated. Use --get-all instead.
```

### 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]
                                           [--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.  
**--all**  
Deprecated. Use `--get-all` instead.

## 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] [--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.  
**--all**  
Deprecated. Use `--get-all` instead.

### 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]
                                           [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

## gitlab project-deployment

```
usage: gitlab project-deployment [-h] {list,get,create,update} ...
```

### **-h, --help**

show this help message and exit

## 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]
[--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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]
                                                    [--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] [--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.  
**--all**  
 Deprecated. Use `--get-all` instead.

### gitlab project-environment

```
usage: gitlab project-environment [-h]
  list
  get
  create
  update
  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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

**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] [--page PAGE]
                                [--per-page PER_PAGE] [--get-all] [--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>

**--page** <page>

**--per-page** <per\_page>

**--get-all**  
Return all items from the server, without pagination.

**--all**  
Deprecated. Use `--get-all` instead.

### 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] [--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.
--all
    Deprecated. Use --get-all instead.

```

## 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]

```

(continues on next page)

(continued from previous page)

```
[--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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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]
                                         [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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]
```

**-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>

### 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]
                                     [--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.

### **--all**

Deprecated. Use `--get-all` instead.

## 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]
                                [--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] [--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>  
**--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.  
**--all**  
Deprecated. Use `--get-all` instead.

### 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]
      [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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]
                                           [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

**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] [--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.

--all
    Deprecated. Use --get-all instead.

```

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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-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]
                                [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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]
                                     [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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
  closes-issues
  commits
  changes
  approve
  unapprove
  rebase
  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
```

**-h, --help**  
 show this help message and exit  
**--project-id** <project\_id>  
**--sudo** <sudo>  
**--iid** <iid>

### 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
                                           [--assignee-id ASSIGNEE_ID]
                                           [--description DESCRIPTION]
                                           [--target-project-id TARGET_PROJECT_ID]
                                           [--labels LABELS]
                                           [--milestone-id MILESTONE_ID]
                                           [--remove-source-branch REMOVE_SOURCE_BRANCH]
                                           [--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>

**--source-branch** <source\_branch>

```

--target-branch <target_branch>
--title <title>
--assignee-id <assignee_id>
--description <description>
--target-project-id <target_project_id>
--labels <labels>
--milestone-id <milestone_id>
--remove-source-branch <remove_source_branch>
--allow-maintainer-to-push <allow_maintainer_to_push>
--squash <squash>
--reviewer-ids <reviewer_ids>

```

### 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] [--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.
--all
    Deprecated. Use --get-all instead.

```

### 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-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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

**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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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-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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

**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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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] [--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.

### --all

Deprecated. Use `--get-all` instead.

## 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]
```

(continues on next page)

(continued from previous page)

<code>↪REQUEST]</code>	<code>[--reopen-issue REOPEN_ISSUE]</code>
<code>↪REQUEST]</code>	<code>[--close-issue CLOSE_ISSUE]</code>
<code>↪REQUEST]</code>	<code>[--reassign-issue REASSIGN_ISSUE]</code>
<code>↪MERGE_REQUEST]</code>	<code>[--new-merge-request NEW_MERGE_</code>
<code>↪REQUEST]</code>	<code>[--reopen-merge-request REOPEN_MERGE_</code>
	<code>[--close-merge-request CLOSE_MERGE_</code>
	<code>[--reassign-merge-request REASSIGN_</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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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]
                                         [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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,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]
                                     [--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] [--all]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--scope** <scope>

**--status** <status>

**--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.



**--all**Deprecated. Use `--get-all` instead.**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] [--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.

**--all**Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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]
                                           [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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-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]
      [--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.

**--all**

Deprecated. Use `--get-all` instead.

## gitlab project-protected-branch

```
usage: gitlab project-protected-branch [-h] {list,get,create,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]
      [--allowed-to-push ALLOWED_TO_PUSH]
      [--allowed-to-merge ALLOWED_TO_MERGE]
```

(continues on next page)



(continued from previous page)

```
↪ UNPROTECT]
↪ APPROVAL_REQUIRED]
```

```
[--allowed-to-unprotect ALLOWED_TO_
[--code-owner-approval-required CODE_OWNER_
```

**-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>**--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]
        [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

**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]
                                         [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.



## 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]
                                         [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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
                                     --description DESCRIPTION [--name NAME]
                                     [--ref REF] [--assets ASSETS]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--tag-name** <tag\_name>

**--description** <description>

**--name** <name>

**--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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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]
                                           [--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>

**--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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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]
                                           [--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>
--link-type <link_type>

```

### gitlab project-remote-mirror

```
usage: gitlab project-remote-mirror [-h] {list,create,update} ...
```

**-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 list

```
usage: gitlab project-remote-mirror list [-h] [--sudo SUDO] --project-id
                                           PROJECT_ID [--page PAGE]
                                           [--per-page PER_PAGE] [--get-all]
                                           [--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.  
**--all**  
Deprecated. Use `--get-all` instead.

### 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-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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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] [--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.  
**--all**  
 Deprecated. Use `--get-all` instead.

### 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 --file-name
                                     FILE_NAME --content CONTENT --visibility
                                     VISIBILITY [--description DESCRIPTION]
```

**-h, --help**  
show this help message and exit

**--sudo** <sudo>

**--project-id** <project\_id>

**--title** <title>

**--file-name** <file\_name>

**--content** <content>

**--visibility** <visibility>

**--description** <description>

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## gitlab project-snippet update

```
usage: gitlab project-snippet update [-h] [--sudo SUDO] --project-id
                                     PROJECT_ID --id ID [--title TITLE]
                                     [--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>****--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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

**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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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]
                               [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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} ...
```

**-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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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 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] [--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.

### **--all**

Deprecated. Use `--get-all` instead.

## 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] [--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.  
**--all**  
 Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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]
```

(continues on next page)

(continued from previous page)

```
[--with-custom-attributes WITH_CUSTOM_ATTRIBUTES]
[--page PAGE] [--per-page PER_PAGE]
[--get-all] [--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.

**--all**Deprecated. Use `--get-all` instead.

## 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 --file-name
                             FILE_NAME --content CONTENT [--lifetime LIFETIME]
                             [--visibility VISIBILITY]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--title** <title>

**--file-name** <file\_name>

**--content** <content>

**--lifetime** <lifetime>

**--visibility** <visibility>

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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]
                        [--file-name FILE_NAME] [--content CONTENT]
                        [--visibility VISIBILITY]
```

**-h, --help**

show this help message and exit

**--sudo** <sudo>

**--id** <id>

**--title** <title>  
**--file-name** <file\_name>  
**--content** <content>  
**--visibility** <visibility>

### 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] [--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.
--all
    Deprecated. Use --get-all instead.

```

## 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] [--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.

### **--all**

Deprecated. Use `--get-all` instead.

## 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} ...
```

### -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] [--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.

### **--all**

Deprecated. Use `--get-all` instead.

## 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
```

(continues on next page)

(continued from previous page)

```
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]
                        [--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.

**--all**

Deprecated. Use `--get-all` instead.



### 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]
```

(continues on next page)

(continued from previous page)

```
[--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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**  
Deprecated. Use `--get-all` instead.

### 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]
                               [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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]
                               [--page PAGE] [--per-page PER_PAGE] [--get-all]
                               [--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>

**--page** <page>

**--per-page** <per\_page>

**--get-all**

Return all items from the server, without pagination.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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]
                                             [--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.

**--all**  
Deprecated. Use `--get-all` instead.

## 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

### 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] [--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.  
**--all**  
Deprecated. Use `--get-all` instead.

### 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] [--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.  
**--all**  
 Deprecated. Use `--get-all` instead.

### 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] [--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.

**--all**

Deprecated. Use `--get-all` instead.

## 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>**

## 6.8 Changelog

### 6.8.1 v3.9.0 (2022-08-28)

#### Feature

- Add support for merge\_base API ([dd4fbd5](#))

## 6.8.2 v3.8.1 (2022-08-10)

### Fix

- **client:** Do not assume user attrs returned for `auth()` ([a07547c](#))

## 6.8.3 v3.8.0 (2022-08-04)

### Feature

- **client:** Warn user on misconfigured URL in `auth()` ([0040b43](#))
- Support downloading archive subpaths ([cadb0e5](#))

### Fix

- **client:** Ensure encoded query params are never duplicated ([1398426](#))
- Optionally keep user-provided base URL for pagination ([#2149](#)) ([e2ea8b8](#))

### Documentation

- Describe self-revoking personal access tokens ([5ea48fc](#))

## 6.8.4 v3.7.0 (2022-07-28)

### Feature

- Allow sort/ordering for project releases ([b1dd284](#))
- **cli:** Add a custom help formatter ([005ba93](#))
- Add support for iterations API ([194ee01](#))
- **groups:** Add support for shared projects API ([66461ba](#))
- **issues:** Add support for issue reorder API ([8703324](#))
- **namespaces:** Add support for namespace existence API ([4882cb2](#))
- Add support for group and project invitations API ([7afd340](#))
- **projects:** Add support for project restore API ([4794ecc](#))
- Add support for filtering jobs by scope ([0e1c0dd](#))
- Add `asdict()` and `to_json()` methods to Gitlab Objects ([08ac071](#))
- **api:** Add support for instance-level registry repositories ([284d739](#))
- **groups:** Add support for group-level registry repositories ([70148c6](#))
- Add `'merge_pipelines_enabled'` project attribute ([fc33c93](#))
- Support validating CI lint results ([3b1ede4](#))
- **cli:** Add support for global CI lint ([3f67c4b](#))
- **objects:** Add Project CI Lint support ([b213dd3](#))

- Add support for group push rules ([b5cdc09](#))
- **api:** Add support for `get` for a MR approval rule ([89c18c6](#))

### Fix

- Support array types for most resources ([d9126cd](#))
- Use the `[]` after key names for array variables in `params` ([1af44ce](#))
- **runners:** Fix listing for `/runners/all` ([c6dd57c](#))
- **config:** Raise error when gitlab id provided but no config section found ([1ef7018](#))
- **config:** Raise error when gitlab id provided but no config file found ([ac46c1c](#))
- Add `get_all` param (and `--get-all`) to allow passing `all` to API ([7c71d5d](#))
- Results returned by `attributes` property to show updates ([e5affc8](#))
- Enable epic notes ([5fc3216](#))
- **cli:** Remove irrelevant MR approval rule list filters ([0daec5f](#))
- Ensure path elements are escaped ([5d9c198](#))

### Documentation

- **cli:** Showcase use of token scopes ([4a6f8d6](#))
- **projects:** Document export with upload to URL ([03f5484](#))
- Describe fetching existing export status ([9c5b8d5](#))
- **authors:** Add John ([e2afb84](#))
- Document CI Lint usage ([d5de4b1](#))
- **users:** Add docs about listing a user's projects ([065a1a5](#))
- Update return type of pushrules ([53cbecc](#))
- Describe ROPC flow in place of password authentication ([91c17b7](#))
- **readme:** Remove redundant `-v` that breaks the command ([c523e18](#))

## 6.8.5 v3.6.0 (2022-06-28)

### Feature

- **downloads:** Allow streaming downloads access to response iterator ([#1956](#)) ([b644721](#))
- **api:** Support `head()` method for `get` and `list` endpoints ([ce9216c](#))
- **api:** Implement `HEAD` method ([90635a7](#))
- **users:** Add `approve` and `reject` methods to `User` ([f57139d](#))
- **api:** Convert `gitlab.const` to `Enums` ([c3c6086](#))
- Add support for Protected Environments ([1dc9d0f](#))
- **users:** Add `ban` and `unban` methods ([0d44b11](#))

- **docker:** Provide a Debian-based slim image ([384031c](#))
- Support mutually exclusive attributes and consolidate validation to fix board lists ([#2037](#)) ([3fa330c](#))
- **client:** Introduce `iterator=True` and deprecate `as_list=False` in `list()` ([cdc6605](#))

## Fix

- **base:** Do not fail `repr()` on lazy objects ([1efb123](#))
- **cli:** Project-merge-request-approval-rule ([15a242c](#))
- **cli:** Fix project export download for CLI ([5d14867](#))

## Documentation

- **api:** Add separate section for advanced usage ([22ae101](#))
- **api:** Document usage of `head()` methods ([f555bfb](#))
- **projects:** Provide more detailed import examples ([8f8611a](#))
- **projects:** Document 404 gotcha with unactivated integrations ([522ecff](#))
- **variables:** Instruct users to follow GitLab rules for values ([194b6be](#))
- **api:** Stop linking to `python-requests.org` ([49c7e83](#))
- **api:** Fix incorrect docs for `merge_request_approvals` ([#2094](#)) ([5583eaa](#))
- **api-usage:** Add `import os` in example ([2194a44](#))
- Drop deprecated `setuptools` `build_sphinx` ([048d66a](#))
- **usage:** Refer to upstream docs instead of custom attributes ([ae7d3b0](#))
- **ext:** Fix rendering for `RequiredOptional` dataclass ([4d431e5](#))
- Documentation updates to reflect addition of mutually exclusive attributes ([24b720e](#))
- Use `as_list=False` or `all=True` in Getting started ([de8c6e8](#))

## 6.8.6 v3.5.0 (2022-05-28)

### Feature

- **objects:** Support get project storage endpoint ([8867ee5](#))
- Display human-readable attribute in `repr()` if present ([6b47c26](#))
- **ux:** Display `project.name_with_namespace` on project `repr` ([e598762](#))

## Fix

- **cli:** Changed default `allow_abbrev` value to fix arguments collision problem (#2013) (d68cacf)
- Duplicate subparsers being added to `argparse` (f553fd3)

## Documentation

- Update issue example and extend API usage docs (aad71d2)
- **CONTRIBUTING.rst:** Fix link to conventional-changelog commit format documentation (2373a4f)
- Add missing Admin access const value (3e0d4d9)
- **merge\_requests:** Add new possible merge request state and link to the upstream docs (e660fa8)

## 6.8.7 v3.4.0 (2022-04-28)

### Feature

- Emit a warning when using a `list()` method returns max (1339d64)
- **objects:** Support getting project/group deploy tokens by id (fcd37fe)
- **user:** Support getting user SSH key by id (6f93c05)
- **api:** Re-add topic delete endpoint (d1d96bd)

## Fix

- Add `ChunkedEncodingError` to list of retryable exceptions (7beb20f)
- Avoid passing redundant arguments to API (3431887)
- **cli:** Add missing filters for project commit list (149d244)
- Add 52x range to retry transient failures and tests (c3ef1b5)
- Also retry HTTP-based transient errors (3b49e4d)

## Documentation

- **api-docs:** Docs fix for application scopes (e1ad93d)

## 6.8.8 v3.3.0 (2022-03-28)

### Feature

- **object:** Add pipeline test report summary support (a97e0cf)

## Fix

- Support RateLimit-Reset header ([4060146](#))

## Documentation

- Fix typo and incorrect style ([2828b10](#))
- Add pipeline test report summary support ([d78afb3](#))
- **chore:** Include docs .js files in sdist ([3010b40](#))

## 6.8.9 v3.2.0 (2022-02-28)

### Feature

- **merge\_request\_approvals:** Add support for deleting MR approval rules ([85a734f](#))
- **artifacts:** Add support for project artifacts delete API ([c01c034](#))
- **mixins:** Allow deleting resources without IDs ([0717517](#))
- **objects:** Add a complete artifacts manager ([c8c2fa7](#))

## Fix

- **services:** Use slug for id\_attr instead of custom methods ([e30f39d](#))
- Remove custom delete method for labels ([0841a2a](#))

## Documentation

- Enable gitter chat directly in docs ([bd1ecdd](#))
- Add delete methods for runners and project artifacts ([5e711fd](#))
- Add retry\_transient infos ([bb1f054](#))
- Add transient errors retry info ([b7a1266](#))
- **artifacts:** Deprecate artifacts() and artifact() methods ([64d01ef](#))
- Revert “chore: add temporary banner for v3” ([#1864](#)) ([7a13b9b](#))

## 6.8.10 v3.1.1 (2022-01-28)

### Fix

- **cli:** Make ‘per\_page’ and ‘page’ type explicit ([d493a5e](#))
- **cli:** Make ‘timeout’ type explicit ([bbb7df5](#))
- **cli:** Allow custom methods in managers ([8dfed0c](#))
- **objects:** Make resource access tokens and repos available in CLI ([e0a3a41](#))

## Documentation

- Enhance release docs for CI\_JOB\_TOKEN usage ([5d973de](#))
- **changelog:** Add missing changelog items ([01755fb](#))

### 6.8.11 v3.1.0 (2022-01-14)

#### Feature

- add support for Group Access Token API ([c01b7c4](#))
- Add support for Groups API method `transfer()` ([0007006](#))
- **api:** Add `project.transfer()` and deprecate `transfer_project()` ([259668a](#))
- **api:** Return result from `SaveMixin.save()` ([e6258a4](#))

#### Fix

- **cli:** Add missing list filters for environments ([6f64d40](#))
- Use url-encoded ID in all paths ([12435d7](#))
- **members:** Use new `*All` objects for `*AllManager` managers ([755e0a3](#))
- **api:** Services: add missing `lazy` parameter ([888f332](#))
- broken URL for FAQ about attribute-error-list ([1863f30](#))
- remove default arguments for `mergerequests.merge()` ([8e589c4](#))
- remove custom URL encoding ([3d49e5e](#))

## Documentation

- **cli:** make examples more easily navigable by generating TOC ([f33c523](#))
- update project access token API reference link ([73ae955](#))

### 6.8.12 v3.0.0 (2022-01-05)

#### Feature

- **docker:** Remove custom entrypoint from image ([80754a1](#))
- **cli:** Allow options from args and environment variables ([ca58008](#))
- **api:** Support file format for repository archive ([83dcabf](#))
- Add support for `squash_option` in Projects ([a246ce8](#))
- **cli:** Do not require config file to run CLI ([92a893b](#))
- **api:** Add support for Topics API ([e7559bf](#))
- Add delete on `package_file` object ([124667b](#))
- Add support for `projects.groups.list()` ([68ff595](#))

- **api:** Add support for epic notes ([7f4edb5](#))
- Remove support for Python 3.6, require 3.7 or higher ([414009d](#))
- **api:** Add project milestone promotion ([f068520](#))
- **api:** Add merge trains ([fd73a73](#))
- **api:** Add merge request approval state ([f41b093](#))
- **api:** Add project label promotion ([6d7c88a](#))
- **objects:** Support delete package files API ([4518046](#))
- **objects:** List starred projects of a user ([47a5606](#))
- **build:** Officially support and test python 3.10 ([c042ddc](#))
- **objects:** Support Create and Revoke personal access token API ([e19314d](#))
- Default to gitlab.com if no URL given ([8236281](#))
- Allow global `retry_transient_errors` setup ([3b1d3a4](#))

## Fix

- Handle situation where GitLab does not return values ([cb824a4](#))
- Stop encoding ‘.’ to ‘%2E’ ([702e41d](#))
- **build:** Do not include docs in wheel package ([68a97ce](#))
- **api:** Delete invalid ‘project-runner get’ command ([#1628](#)) ([905781b](#))
- **api:** Replace deprecated attribute in `delete_in_bulk()` ([#1536](#)) ([c59fbdb](#))
- **objects:** Rename confusing `to_project_id` argument ([ce4bc0d](#))
- Raise error if there is a 301/302 redirection ([d56a434](#))
- **build:** Do not package tests in wheel ([969dccc](#))

## Breaking

- 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](#))
- 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](#))
- 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](#))
- As of python-gitlab 3.0.0, Python 3.6 is no longer supported. Python 3.7 or higher is required. ([414009d](#))
- As of python-gitlab 3.0.0, the default branch for development has changed from `master` to `main`. ([545f8ed](#))
- remove deprecated branch protect methods in favor of the more complete protected branches API. ([9656a16](#))
- The deprecated `name_regex` attribute has been removed in favor of `name_regex_delete`. (see <https://gitlab.com/gitlab-org/gitlab/-/commit/ce99813cf54>) ([c59fbdb](#))



- 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](#))
- remove deprecated constants defined in `gitlab.v4.objects`, and use only `gitlab.const` module ([3f320af](#))
- remove deprecated tag release API. This was removed in GitLab 14.0 ([2b8a94a](#))
- remove deprecated `project.issuesstatistics` in favor of `project.issues_statistics` ([ca7777e](#))
- remove deprecated `members.all()` method in favor of `members_all.list()` ([4d7b848](#))
- remove deprecated `pipelines()` methods in favor of `pipelines.list()` ([c4f5ec6](#))
- python-gitlab will now default to `gitlab.com` if no URL is given ([8236281](#))
- raise error if there is a 301/302 redirection ([d56a434](#))

## Documentation

- Switch to Furo and refresh introduction pages ([ee6b024](#))
- Correct documentation for updating discussion note ([ee66f4a](#))
- Rename documentation files to match names of code files ([ee3f865](#))
- **project:** Remove redundant encoding parameter ([fed613f](#))
- Use annotations for return types ([79e785e](#))
- Update docs to use `gitlab.const` for constants ([b3b0b5f](#))
- Only use type annotations for documentation ([b7dde0d](#))
- Add links to the GitLab API docs ([e3b5d27](#))
- Fix API delete key example ([b31bb05](#))
- **pipelines:** Document `take_ownership` method ([69461f6](#))
- **api:** Document the update method for project variables ([7992911](#))
- **api:** Clarify job token usage with `auth()` ([3f423ef](#))
- Fix a few typos ([7ea4ddc](#))
- Consolidate changelogs and remove v3 API docs ([90da8ba](#))
- Correct documented return type ([acabf63](#))

### 6.8.13 v2.10.1 (2021-08-28)

#### Fix

- **mixins:** Improve deprecation warning ([57e0187](#))
- **deps:** Upgrade requests to 2.25.0 (see CVE-2021-33503) ([ce995b2](#))

## Documentation

- **mergequests:** Gl.mergequests.list documentation was misleading ([5b5a7bc](#))

## 6.8.14 v2.10.0 (2021-07-28)

### Feature

- **api:** Add merge\_ref for merge requests ([1e24ab2](#))
- **api:** Add name\_regex\_keep attribute in delete\_in\_bulk() ([e49ff3f](#))

### Fix

- **api:** Do not require Release name for creation ([98cd03b](#))

## Documentation

- **readme:** Move contributing docs to CONTRIBUTING.rst ([edf49a3](#))
- Add example for mr.merge\_ref ([b30b8ac](#))
- **project:** Add example on getting a single project using name with namespace ([ef16a97](#))

## 6.8.15 v2.9.0 (2021-06-28)

### Feature

- **release:** Allow to update release ([b4c4787](#))
- **api:** Add group hooks ([4a7e9b8](#))
- **api:** Remove responsibility for API inconsistencies for MR reviewers ([3d985ee](#))
- **api:** Add MR pipeline manager and deprecate pipelines() method ([954357c](#))
- **api:** Add support for creating/editing reviewers in project merge requests ([676d1f6](#))

## Documentation

- **tags:** Remove deprecated functions ([1b1a827](#))
- **release:** Add update example ([6254a5f](#))
- Make Gitlab class usable for intersphinx ([8753add](#))

## 6.8.16 v2.8.0 (2021-06-10)

### Feature

- Add keys endpoint ([a81525a](#))
- **objects:** Add support for Group wikis ([#1484](#)) ([74f5e62](#))
- **objects:** Add support for generic packages API ([79d88bd](#))
- **api:** Add deployment mergerequests interface ([fbbc0d4](#))
- **objects:** Support all issues statistics endpoints ([f731707](#))
- **objects:** Add support for descendant groups API ([1b70580](#))
- **objects:** Add pipeline test report support ([ee9f96e](#))
- **objects:** Add support for billable members ([fb0b083](#))
- Add feature to get inherited member for project/group ([e444b39](#))
- Add code owner approval as attribute ([fdc46ba](#))
- Indicate that we are a typed package ([e4421ca](#))
- Add support for lists of integers to ListAttribute ([115938b](#))

### Fix

- Catch invalid type used to initialize RESTObject ([c7bcc25](#))
- Functional project service test ([#1500](#)) ([093db9d](#))
- Ensure kwargs are passed appropriately for ObjectDeleteMixin ([4e690c2](#))
- **cli:** Add missing list filter for jobs ([b3d1c26](#))
- Change mr.merge() to use 'post\_data' ([cb6a3c6](#))
- **cli:** Fix parsing CLI objects to classnames ([4252070](#))
- **objects:** Return server data in cancel/retry methods ([9fed061](#))
- **objects:** Add missing group attributes ([d20ff4f](#))
- **objects:** Allow lists for filters for in all objects ([603a351](#))
- Iids not working as a list in projects.issues.list() ([45f806c](#))
- Add a check to ensure the MRO is correct ([565d548](#))

### Documentation

- Fix typo in http\_delete docstring ([5226f09](#))
- **api:** Add behavior in local attributes when updating objects ([38f65e8](#))
- Fail on warnings during sphinx build ([cbd4d52](#))

### 6.8.17 v2.7.1 (2021-04-26)

- `fix(files)`: do not url-encode file paths twice

### 6.8.18 v2.7.0 (2021-04-25)

#### Bug Fixes

- update user's bool data and avatar (3ba27ffb)
- argument type was not a tuple as expected (062f8f6a)
- correct some type-hints in `gitlab/mixins.py` (8bd31240)
- only append kwargs as query parameters (b9ecc9a8)
- only add `query_parameters` to `GitlabList` once (1386)
- checking if `RESTManager._from_parent_attrs` is set (8224b406)
- handling config value in `_get_values_from_helper` (9dfb4cd9)
- let the `homedir` be expanded in path of helper (fc7387a0)
- make secret helper more user friendly (fc2798fc)
- linting issues and test (b04dd2c0)
- handle tags like `debian/2%2.6-21` as identifiers (b4dac5ce)
- remove duplicate class definitions in `v4/objects/users.py` (7c4e6259)
- wrong variable name (15ec41ca)
- tox pep8 target, so that it can run (f518e87b)
- undefined name errors (48ec9e0f)
- extend wait timeout for `test_delete_user()` (19fde8ed)
- `test_update_group()` dependency on ordering (e78a8d63)
- honor parameter value passed (c2f8f0e7)
- **objects**: add single get endpoint for instance audit events (c3f0a6f1)
- **types**: prevent `dir` from producing duplicates (5bf7525d)

#### Features

- add `ProjectPackageFile` (#1372)
- add option to add a helper to lookup token (8ecf5592)
- add project audit endpoint (6660dbef)
- add personal access token API (2bb16fac)
- add import from bitbucket server (ff3013a2)
- **api,cli**: make user agent configurable (4bb201b9)
- **issues**: add missing get verb to `IssueManager` (f78ebe06)
- **objects**:

- add support for resource state events API (d4799c40)
- add support for group audit events API (2a0fbdf9)
- add Release Links API support (28d75181)
- **projects:** add project access token api (1becef02)
- **users:** add follow/unfollow API (e456869d)

## Documentation

- correct ProjectFile.decode() documentation (b180bafd)
- update doc for token helper (3ac6fa12)
- better real life token lookup example (9ef83118)

## 6.8.19 v2.6.0 (2021-01-29)

### Features

- support multipart uploads (2fa3004d)
- add MINIMAL\_ACCESS constant (49eb3ca7)
- unit tests added (f37ebf5f)
- added support for pipeline bridges (05cbdc22)
- adds support for project merge request approval rules (#1199) (c6fbf399)
- **api:**
  - added wip filter param for merge requests (d6078f80)
  - added wip filter param for merge requests (aa6e80d5)
  - add support for user identity provider deletion (e78e1215)
- **tests:** test label getter (a41af902)

### Bug Fixes

- docs changed using the consts (650b65c3)
- typo (9baa9053)
- **api:**
  - use RetrieveMixin for ProjectLabelManager (1a143952)
  - add missing runner access\_level param (92669f2e)
- **base:** really refresh object (e1e0d8cb), closes (#1155)
- **cli:**
  - write binary data to stdout buffer (0733ec6c)
  - add missing args for project lists (c73e2374)

### 6.8.20 v2.5.0 (2020-09-01)

#### Features

- add support to resource milestone events (88f8cc78), closes #1154
- add share/unshare group with group (7c6e541d)
- add support for instance variables (4492fc42)
- add support for Packages API (71495d12)
- add endpoint for latest ref artifacts (b7a07fca)

#### Bug Fixes

- wrong reconfirmation parameter when updating user's email (b5c267e1)
- tests fail when using REUSE\_CONTAINER option (0078f899, closes #1146)
- implement Gitlab's behavior change for owned=True (99777991)

### 6.8.21 v2.4.0 (2020-07-09)

#### Bug Fixes

- do not check if kwargs is none (a349b90e)
- make query kwargs consistent between call in init and next (72ffa016)
- pass kwargs to subsequent queries in gitlab list (1d011ac7)
- **merge:** parse arguments as query\_data (878098b7)

#### Features

- add NO\_ACCESS const (dab4d0a1)
- add masked parameter for variables command (b6339bf8)

### 6.8.22 v2.3.1 (2020-06-09)

- revert keyset pagination by default

### 6.8.23 v2.3.0 (2020-06-08)

#### Features

- add group runners api (49439916)
- add play command to project pipeline schedules (07b99881)
- allow an environment variable to specify config location (401e702a)
- **api:** added support in the GroupManager to upload Group avatars (28eb7eab)

- **services:** add project service list API (fc522218)
- **types:** add **dir** to RESTObject to expose attributes (cad134c0)

### Bug Fixes

- use keyset pagination by default for /projects > 50000 (f86ef3bb)
- **config:** fix duplicate code (ee2df6f1), closes (#1094)
- **project:** add missing project parameters (ad8c67d6)

## 6.8.24 v2.2.0 (2020-04-07)

### Bug Fixes

- add missing import\_project param (9b16614b)
- **types:** do not split single value string in ListAttribute (a26e5858)

### Features

- add commit GPG signature API (da7a8097)
- add create from template args to ProjectManager (f493b73e)
- add remote mirrors API (#1056) (4cfaa2fd)
- add Gitlab Deploy Token API (01de524c)
- add Group Import/Export API (#1037) (6cb9d923)

## 6.8.25 v2.1.2 (2020-03-09)

### Bug Fixes

- Fix regression, when using keyset pagination with merge requests. Related to [https://github.com/python-gitlab/issues/1044](https://github.com/python-gitlab/python-gitlab/issues/1044)

## 6.8.26 v2.1.1 (2020-03-09)

### Bug Fixes

**users:** update user attributes

This change was made to migrate an issue in Gitlab (again). Fix available in: [https://gitlab.com/gitlab-org/gitlab/-/merge\\_requests/26792](https://gitlab.com/gitlab-org/gitlab/-/merge_requests/26792)

### 6.8.27 v2.1.0 (2020-03-08)

#### Bug Fixes

- do not require empty data dict for create() (99d959f7)
- remove trailing slashes from base URL (#913) (2e396e4a)
- return response with commit data (b77b945c)
- remove null values from features POST data, because it fails with HTTP 500 (1ec1816d)
- **docs:**
  - fix typo in user memberships example (33889becb)
  - update to new set approvers call for # of approvers (8e0c5262)
  - update docs and tests for set\_approvers (2cf12c79)
- **objects:**
  - add default name data and use http post (70c0cfb6)
  - update set\_approvers function call (65ecadcf)
  - update to new gitlab api for path, and args (e512cddd)

#### Features

- add support for user memberships API (#1009) (c313c2b0)
- add support for commit revert API (#991) (5298964e)
- add capability to control GitLab features per project or group (7f192b4f)
- use keyset pagination by default for all=True (99b4484d)
- add support for GitLab OAuth Applications API (4e12356d)

### 6.8.28 v2.0.1 (2020-02-05)

#### Changes

- **users:** update user attributes

This change was made to mitigate an issue in Gitlab. See: <https://gitlab.com/gitlab-org/gitlab/issues/202070>

### 6.8.29 v2.0.0 (2020-01-26)

**This releases drops support for python < 3.6**

#### Bug Fixes

- **projects:** adjust snippets to match the API (e104e213)



## Features

- add global order\_by option to ease pagination (d1879253)
- support keyset pagination globally (0b71ba4d)
- add appearance API (4c4ac5ca)
- add autocompletion support (973cb8b9)

### 6.8.30 v1.15.0 (2019-12-16)

#### Bug Fixes

- ignore all parameter, when as\_list=True 137d72b3, closes #962

## Features

- allow cfg timeout to be overridden via kwargs e9a8289a
- add support for /import/github aa4d41b7
- nicer stacktrace 697cda24
- retry transient HTTP errors 59fe2714, closes #970
- access project's issues statistics 482e57ba, closes #966
- adding project stats db0b00a9, closes #967
- add variable\_type/protected to projects ci variables 4724c50e
- add variable\_type to groups ci variables 0986c931

### 6.8.31 v1.14.0 (2019-12-07)

#### Bug Fixes

- added missing attributes for project approvals 460ed63c
- **labels:** ensure label.save() works 727f5361
- **project-fork:**
  - copy create fix from ProjectPipelineManager 516307f1
  - correct path computation for project-fork list 44a7c278

## Features

- add audit endpoint 2534020b
- add project and group clusters ebd053e7
- add support for include\_subgroups filter adbcd83f

### 6.8.32 v1.13.0 (2019-11-02)

## Features

- add users activate, deactivate functionality (32ad6692)
- send python-gitlab version as user-agent (c22d49d0)
- add deployment creation (ca256a07), closes [#917]
- **auth:** remove deprecated session auth (b751cdf4)
- **doc:** remove refs to api v3 in docs (6beea99)
- **test:** unused unittest2, type -> isinstance (33b18012)

## Bug Fixes

- **projects:** support approval\_rules endpoint for projects (2cef2bb4)

### 6.8.33 v1.12.1 (2019-10-07)

## Bug Fixes

fix: fix not working without auth provided

### 6.8.34 v1.12.0 (2019-10-06)

## Features

- add support for job token
- **project:**
  - implement update\_submodule
  - add file blame api
- **user:** add status api

### Bug Fixes

- **cli:** fix cli command user-project list
- **labels:** don't mangle label name on update
- **todo:** mark\_all\_as\_done doesn't return anything

## 6.8.35 v1.11.0 (2019-08-31)

### Features

- add methods to retrieve an individual project environment
- group labels with subscriptable mixin

### Bug Fixes

- projects: avatar uploading for projects
- remove empty list default arguments
- remove empty dict default arguments
- add project and group label update without id to fix cli

## 6.8.36 v1.10.0 (2019-07-22)

### Features

- add mr rebase method bc4280c2
- get artifact by ref and job cda11745
- add support for board update 908d79fa, closes #801
- add support for issue.related\_merge\_requests 90a36315, closes #794

### Bug Fixes

- improve pickle support b4b5decb
- **cli:**
  - allow `--recursive` parameter in repository tree 7969a78c, closes #718, #731
  - don't fail when the short print attr value is None 8d1552a0, closes #717, #727
  - fix update value for key not working b7662039

### 6.8.37 v1.9.0 (2019-06-19)

#### Features

- implement artifacts deletion
- add endpoint to get the variables of a pipeline
- delete ProjectPipeline
- implement **eq** and **hash** methods
- Allow runpy invocation of CLI tool (python -m gitlab)
- add project releases api
- merged new release & registry apis

#### Bug Fixes

- convert # to %23 in URLs
- pep8 errors
- use python2 compatible syntax for super
- Make MemberManager.all() return a list of objects
- 
- Re-enable command specific help messages
- dont ask for id attr if this is \*Manager originating custom action
- fix -/\_ replacament for \*Manager custom actions
- fix repository\_id marshaling in cli
- register cli action for delete\_in\_bulk

### 6.8.38 v1.8.0 (2019-02-22)

- docs(setup): use proper readme on PyPI
- docs(readme): provide commit message guidelines
- fix(api): make reset\_time\_estimate() work again
- fix: handle empty 'Retry-After' header from GitLab
- fix: remove decode() on error\_message string
- chore: release tags to PyPI automatically
- fix(api): avoid parameter conflicts with python and gitlab
- fix(api): Don't try to parse raw downloads
- feat: Added approve & unapprove method for Mergerequests
- fix all kwarg behaviour

### 6.8.39 v1.7.0 (2018-12-09)

- **docs:** Fix the owned/starred usage documentation
- **docs:** Add a warning about http to https redirects
- Fix the https redirection test
- **docs:** Add a note about GroupProject limited API
- Add missing comma in ProjectIssueManager \_create\_attrs
- More flexible docker image
- Add project protected tags management
- **cli:** Print help and usage without config file
- Rename MASTER\_ACCESS to MAINTAINER\_ACCESS
- **docs:** Add docs build information
- Use docker image with current sources
- **docs:** Add PyYAML requirement notice
- Add Gitter badge to README
- **docs:** Add an example of pipeline schedule vars listing
- **cli:** Exit on config parse error, instead of crashing
- Add support for resource label events
- **docs:** Fix the milestone filetring doc (iid -> iids)
- **docs:** Fix typo in custom attributes example
- Improve error message handling in exceptions
- Add support for members all() method
- Add access control options to protected branch creation

### 6.8.40 v1.6.0 (2018-08-25)

- **docs:** Don't use hardcoded values for ids
- **docs:** Improve the snippets examples
- **cli:** Output: handle bytes in API responses
- **cli:** Fix the case where we have nothing to print
- Project import: fix the override\_params parameter
- Support group and global MR listing
- Implement MR.pipelines()
- MR: add the squash attribute for create/update
- Added support for listing forks of a project
- **docs:** Add/update notes about read-only objects
- Raise an exception on https redirects for PUT/POST

- **docs:** Add a FAQ
- **cli:** Fix the project-export download

#### 6.8.41 v1.5.1 (2018-06-23)

- Fix the ProjectPipelineJob base class (regression)

#### 6.8.42 v1.5.0 (2018-06-22)

- Drop API v3 support
- Drop GetFromListMixin
- Update the sphinx extension for v4 objects
- Add support for user avatar upload
- Add support for project import/export
- Add support for the search API
- Add a global per\_page config option
- Add support for the discussions API
- Add support for merged branches deletion
- Add support for Project badges
- Implement user\_agent\_detail for snippets
- Implement commit.refs()
- Add commit.merge\_requests() support
- Deployment: add list filters
- Deploy key: add missing attributes
- Add support for environment stop()
- Add feature flags deletion support
- Update some group attributes
- Issues: add missing attributes and methods
- Fix the participants() decorator
- Add support for group boards
- Implement the markdown rendering API
- Update MR attributes
- Add pipeline listing filters
- Add missing project attributes
- Implement runner jobs listing
- Runners can be created (registered)
- Implement runner token validation
- Update the settings attributes

- Add support for the gitlab CI lint API
- Add support for group badges
- Fix the IssueManager path to avoid redirections
- time\_stats(): use an existing attribute if available
- Make ProjectCommitStatus.create work with CLI
- Tests: default to python 3
- ProjectPipelineJob was defined twice
- Silence logs/warnings in unittests
- Add support for MR approval configuration (EE)
- Change post\_data default value to None
- Add geo nodes API support (EE)
- Add support for issue links (EE)
- Add support for LDAP groups (EE)
- Add support for board creation/deletion (EE)
- Add support for Project.pull\_mirror (EE)
- Add project push rules configuration (EE)
- Add support for the EE license API
- Add support for the LDAP groups API (EE)
- Add support for epics API (EE)
- Fix the non-verbose output of ProjectCommitComment

#### 6.8.43 v1.4.0 (2018-05-19)

- Require requests>=2.4.2
- ProjectKeys can be updated
- Add support for unsharing projects (v3/v4)
- **cli:** fix listing for json and yaml output
- Fix typos in documentation
- Introduce RefreshMixin
- **docs:** Fix the time tracking examples
- **docs:** Commits: add an example of binary file creation
- **cli:** Allow to read args from files
- Add support for recursive tree listing
- **cli:** Restore the --help option behavior
- Add basic unit tests for v4 CLI
- **cli:** Fix listing of strings
- Support downloading a single artifact file

- Update docs copyright years
- Implement attribute types to handle special cases
- **docs:** fix GitLab reference for notes
- Expose additional properties for Gitlab objects
- Fix the impersonation token deletion example
- feat: obey the rate limit
- Fix URL encoding on branch methods
- **docs:** add a code example for listing commits of a MR
- **docs:** update service.available() example for API v4
- **tests:** fix functional tests for python3
- api-usage: bit more detail for listing with all
- More efficient .get() for group members
- Add docs for the files arg in http\_\*
- Deprecate GetFromListMixin

#### 6.8.44 v1.3.0 (2018-02-18)

- Add support for pipeline schedules and schedule variables
- Clarify information about supported python version
- Add manager for jobs within a pipeline
- Fix wrong tag example
- Update the groups documentation
- Add support for MR participants API
- Add support for getting list of user projects
- Add Gitlab and User events support
- Make trigger\_pipeline return the pipeline
- Config: support api\_version in the global section
- Gitlab can be used as context manager
- Default to API v4
- Add a simplified example for streamed artifacts
- Add documentation about labels update



### 6.8.45 v1.2.0 (2018-01-01)

- Add mattermost service support
- Add users custom attributes support
- **doc:** Fix project.triggers.create example with v4 API
- OAuth token support
- Remove deprecated objects/methods
- Rework authentication args handling
- Add support for oauth and anonymous auth in config/CLI
- Add support for impersonation tokens API
- Add support for user activities
- Update user docs with gitlab URLs
- **docs:** Bad arguments in projects file documentation
- Add support for user\_agent\_detail (issues)
- Add a SetMixin
- Add support for project housekeeping
- Expected HTTP response for subscribe is 201
- Update pagination docs for ProjectCommit
- Add doc to get issue from iid
- Make todo() raise GitlabTodoError on error
- Add support for award emojis
- Update project services docs for v4
- Avoid sending empty update data to issue.save
- **docstrings:** Explicitly document pagination arguments
- **docs:** Add a note about password auth being removed from GitLab
- Submanagers: allow having undefined parameters
- ProjectFile.create(): don't modify the input data
- Update testing tools for /session removal
- Update groups tests
- Allow per\_page to be used with generators
- Add groups listing attributes
- Add support for subgroups listing
- Add supported python versions in setup.py
- Add support for pagesdomains
- Add support for features flags
- Add support for project and group custom variables
- Add support for user/group/project filter by custom attribute

- Respect content of REQUESTS\_CA\_BUNDLE and \*\_proxy envvars

#### 6.8.46 v1.1.0 (2017-11-03)

- Fix trigger variables in v4 API
- Make the delete() method handle / in ids
- **docs:** update the file upload samples
- Tags release description: support / in tag names
- **docs:** improve the labels usage documentation
- Add support for listing project users
- ProjectFileManager.create: handle / in file paths
- Change ProjectUser and GroupProject base class
- **docs:** document get\_create\_attrs in the API tutorial
- Document the Gitlab session parameter
- ProjectFileManager: custom update() method
- Project: add support for printing\_merge\_request\_link\_enabled attr
- Update the ssl\_verify docstring
- Add support for group milestones
- Add support for GPG keys
- Add support for wiki pages
- Update the repository\_blob documentation
- Fix the CLI for objects without ID (API v4)
- Add a contributed Dockerfile
- Pagination generators: expose more information
- Module's base objects serialization
- **doc:** Add sample code for client-side certificates

#### 6.8.47 v1.0.2 (2017-09-29)

- **docs:** remove example usage of submanagers
- Properly handle the labels attribute in ProjectMergeRequest
- ProjectFile: handle / in path for delete() and save()

#### 6.8.48 v1.0.1 (2017-09-21)

- Tags can be retrieved by ID
- Add the server response in GitlabError exceptions
- Add support for project file upload
- Minor typo fix in “Switching to v4” documentation
- Fix password authentication for v4
- Fix the labels attrs on MR and issues
- Exceptions: use a proper error message
- Fix http\_get method in get artifacts and job trace
- CommitStatus: sha is parent attribute
- Fix a couple listing calls to allow proper pagination
- Add missing doc file

#### 6.8.49 v1.0.0 (2017-09-08)

- Support for API v4. See <http://python-gitlab.readthedocs.io/en/master/switching-to-v4.html>
- Support SSL verification via internal CA bundle
- Docs: Add link to gitlab docs on obtaining a token
- Added dependency injection support for Session
- Fixed repository\_compare examples
- Fix changelog and release notes inclusion in sdist
- Missing expires\_at in GroupMembers update
- Add lower-level methods for Gitlab()

#### 6.8.50 v0.21.2 (2017-06-11)

- Install doc: use sudo for system commands
- **v4:** Make MR work properly
- Remove extra\_attrs argument from \_raw\_list
- **v4:** Make project issues work properly
- Fix urlencode() usage (python 2/3) (#268)
- Fixed spelling mistake (#269)
- Add new event types to ProjectHook

### 6.8.51 v0.21.1 (2017-05-25)

- Fix the manager name for jobs in the Project class
- Fix the docs

### 6.8.52 v0.21 (2017-05-24)

- Add time\_stats to ProjectMergeRequest
- Update User options for creation and update (#246)
- Add milestone.merge\_requests() API
- Fix docs typo (s/correspnding/corresponding/)
- Support milestone start date (#251)
- Add support for priority attribute in labels (#256)
- Add support for nested groups (#257)
- Make GroupProjectManager a subclass of ProjectManager (#255)
- Available services: return a list instead of JSON (#258)
- MR: add support for time tracking features (#248)
- Fixed repository\_tree and repository\_blob path encoding (#265)
- Add 'search' attribute to projects.list()
- Initial gitlab API v4 support
- Reorganise the code to handle v3 and v4 objects
- Allow 202 as delete return code
- Deprecate parameter related methods in gitlab.Gitlab

### 6.8.53 v0.20 (2017-03-25)

- Add time tracking support (#222)
- Improve changelog (#229, #230)
- Make sure that manager objects are never overwritten (#209)
- Include chanlog and release notes in docs
- Add DeployKey{,Manager} classes (#212)
- Add support for merge request notes deletion (#227)
- Properly handle extra args when listing with all=True (#233)
- Implement pipeline creation API (#237)
- Fix spent\_time methods
- Add 'delete source branch' option when creating MR (#241)
- Provide API wrapper for cherry picking commits (#236)
- Stop listing if recursion limit is hit (#234)

### 6.8.54 v0.19 (2017-02-21)

- Update project.archive() docs
- Support the scope attribute in runners.list()
- Add support for project runners
- Add support for commit creation
- Fix install doc
- Add builds-email and pipelines-email services
- Deploy keys: rework enable/disable
- Document the dynamic aspect of objects
- Add pipeline\_events to ProjectHook attrs
- Add due\_date attribute to ProjectIssue
- Handle settings.domain\_whitelist, partly
- {Project,Group}Member: support expires\_at attribute

### 6.8.55 v0.18 (2016-12-27)

- Fix JIRA service editing for GitLab 8.14+
- Add jira\_issue\_transition\_id to the JIRA service optional fields
- Added support for Snippets (new API in Gitlab 8.15)
- **docs:** update pagination section
- **docs:** artifacts example: open file in wb mode
- **CLI:** ignore empty arguments
- **CLI:** Fix wrong use of arguments
- **docs:** Add doc for snippets
- Fix duplicated data in API docs
- Update known attributes for projects
- sudo: always use strings

### 6.8.56 v0.17 (2016-12-02)

- README: add badges for pypi and RTD
- Fix ProjectBuild.play (raised error on success)
- Pass kwargs to the object factory
- Add .tox to ignore to respect default tox settings
- Convert response list to single data source for iid requests
- Add support for boards API
- Add support for Gitlab.version()

- Add support for broadcast messages API
- Add support for the notification settings API
- Don't overwrite attributes returned by the server
- Fix bug when retrieving changes for merge request
- Feature: enable / disable the deploy key in a project
- Docs: add a note for python 3.5 for file content update
- ProjectHook: support the token attribute
- Rework the API documentation
- Fix docstring for `http_{username,password}`
- Build managers on demand on `GitlabObject`'s
- API docs: add managers doc in `GitlabObject`'s
- Sphinx ext: factorize the build methods
- Implement `__repr__` for gitlab objects
- Add a 'report a bug' link on doc
- Remove deprecated methods
- Implement merge requests diff support
- Make the manager objects creation more dynamic
- Add support for templates API
- Add attr 'created\_at' to `ProjectIssueNote`
- Add attr 'updated\_at' to `ProjectIssue`
- CLI: add support for project all -all
- Add support for triggering a new build
- Rework requests arguments (support latest requests release)
- Fix `should_remove_source_branch`

### 6.8.57 v0.16 (2016-10-16)

- Add the ability to fork to a specific namespace
- JIRA service - add `api_url` to optional attributes
- Fix bug: Missing coma concatenates array values
- docs: branch protection notes
- Create a project in a group
- Add `only_allow_merge_if_build_succeeds` option to project objects
- Add support for -all in CLI
- Fix examples for file modification
- Use the plural `merge_requests` URL everywhere
- Rework travis and tox setup

- Workaround gitlab setup failure in tests
- Add ProjectBuild.erase()
- Implement ProjectBuild.play()

#### 6.8.58 v0.15.1 (2016-10-16)

- docs: improve the pagination section
- Fix and test pagination
- ‘path’ is an existing gitlab attr, don’t use it as method argument

#### 6.8.59 v0.15 (2016-08-28)

- Add a basic HTTP debug method
- Run more tests in travis
- Fix fork creation documentation
- Add more API examples in docs
- Update the ApplicationSettings attributes
- Implement the todo API
- Add sidekiq metrics support
- Move the constants at the gitlab root level
- Remove methods marked as deprecated 7 months ago
- Refactor the Gitlab class
- Remove `_get_list_or_object()` and its tests
- Fix `canGet` attribute (typo)
- Remove unused `ProjectTagReleaseManager` class
- Add support for project services API
- Add support for project pipelines
- Add support for access requests
- Add support for project deployments

#### 6.8.60 v0.14 (2016-08-07)

- Remove ‘next\_url’ from kwargs before passing it to the cls constructor.
- List projects under group
- Add support for subscribe and unsubscribe in issues
- Project issue: doc and CLI for (un)subscribe
- Added support for HTTP basic authentication
- Add support for build artifacts and trace

- `-title` is a required argument for `ProjectMilestone`
- Commit status: add optional context url
- Commit status: optional get attrs
- Add support for commit comments
- Issues: add optional listing parameters
- Issues: add missing optional listing parameters
- Project issue: proper update attributes
- Add support for project-issue move
- Update `ProjectLabel` attributes
- Milestone: optional listing attrs
- Add support for namespaces
- Add support for label (un)subscribe
- MR: add (un)subscribe support
- Add `note_events` to project hooks attributes
- Add code examples for a bunch of resources
- Implement user emails support
- Project: add `VISIBILITY_*` constants
- Fix the `Project.archive` call
- Implement `archive/unarchive` for a projet
- Update `ProjectSnippet` attributes
- Fix `ProjectMember` update
- Implement sharing project with a group
- Implement CLI for project `archive/unarchive/share`
- Implement runners global API
- Gitlab: add managers for build-related resources
- Implement `ProjectBuild.keep_artifacts`
- Allow to stream the downloads when appropriate
- Groups can be updated
- Replace `Snippet.Content()` with a new `content()` method
- CLI: refactor `_die()`
- Improve commit statuses and comments
- Add support from listing group issues
- Added a new project attribute to enable the container registry.
- Add a contributing section in README
- Add support for global deploy key listing
- Add support for project environments



- MR: get list of changes and commits
- Fix the listing of some resources
- MR: fix updates
- Handle empty messages from server in exceptions
- MR (un)subscribe: don't fail if state doesn't change
- MR merge(): update the object

### 6.8.61 v0.13 (2016-05-16)

- Add support for MergeRequest validation
- MR: add support for cancel\_merge\_when\_build\_succeeds
- MR: add support for closes\_issues
- Add “external” parameter for users
- Add deletion support for issues and MR
- Add missing group creation parameters
- Add a Session instance for all HTTP requests
- Enable updates on ProjectIssueNotes
- Add support for Project raw\_blob
- Implement project compare
- Implement project contributors
- Drop the next\_url attribute when listing
- Remove unnecessary canUpdate property from ProjectIssuesNote
- Add new optional attributes for projects
- Enable deprecation warnings for gitlab only
- Rework merge requests update
- Rework the Gitlab.delete method
- ProjectFile: file\_path is required for deletion
- Rename some methods to better match the API URLs
- Deprecate the file\_\* methods in favor of the files manager
- Implement star/unstar for projects
- Implement list/get licenses
- Manage optional parameters for list() and get()

### 6.8.62 v0.12.2 (2016-03-19)

- Add new ProjectHook attributes
- Add support for user block/unblock
- Fix GitlabObject creation in `_custom_list`
- Add support for more CLI subcommands
- Add some unit tests for CLI
- Add a coverage tox env
- Define `GitlabObject.as_dict()` to dump object as a dict
- Define `GitlabObject.__eq__()` and `__ne__()` equivalence methods
- Define `UserManager.search()` to search for users
- Define `UserManager.get_by_username()` to get a user by username
- Implement “user search” CLI
- Improve the doc for `UserManager`
- CLI: implement user get-by-username
- Re-implement `_custom_list` in the `Gitlab` class
- Fix the ‘invalid syntax’ error on Python 3.2
- `Gitlab.update()`: use the proper attributes if defined

### 6.8.63 v0.12.1 (2016-02-03)

- Fix a broken upload to pypi

### 6.8.64 v0.12 (2016-02-03)

- Improve documentation
- Improve unit tests
- Improve test scripts
- Skip `BaseManager` attributes when encoding to JSON
- Fix the `json()` method for python 3
- Add Travis CI support
- Add a decode method for `ProjectFile`
- Make connection exceptions more explicit
- Fix `ProjectLabel` get and delete
- Implement `ProjectMilestone.issues()`
- `ProjectTag` supports deletion
- Implement setting release info on a tag
- Implement project triggers support

- Implement project variables support
- Add support for application settings
- Fix the ‘password’ requirement for User creation
- Add sudo support
- Fix project update
- Fix Project.tree()
- Add support for project builds

#### 6.8.65 v0.11.1 (2016-01-17)

- Fix discovery of parents object attrs for managers
- Support setting commit status
- Support deletion without getting the object first
- Improve the documentation

#### 6.8.66 v0.11 (2016-01-09)

- functional\_tests.sh: support python 2 and 3
- Add a get method for GitlabObject
- CLI: Add the -g short option for –gitlab
- Provide a create method for GitlabObject’s
- Rename the \_created attribute \_from\_api
- More unit tests
- CLI: fix error when arguments are missing (python 3)
- Remove deprecated methods
- Implement managers to get access to resources
- Documentation improvements
- Add fork project support
- Deprecate the “old” Gitlab methods
- Add support for groups search

#### 6.8.67 v0.10 (2015-12-29)

- Implement pagination for list() (#63)
- Fix url when fetching a single MergeRequest
- Add support to update MergeRequestNotes
- API: Provide a Gitlab.from\_config method
- setup.py: require requests>=1 (#69)

- Fix deletion of object not using 'id' as ID (#68)
- Fix GET/POST for project files
- Make 'confirm' an optional attribute for user creation
- Python 3 compatibility fixes
- Add support for group members update (#73)

#### 6.8.68 v0.9.2 (2015-07-11)

- CLI: fix the update and delete subcommands (#62)

#### 6.8.69 v0.9.1 (2015-05-15)

- Fix the setup.py script

#### 6.8.70 v0.9 (2015-05-15)

- Implement argparse library for parsing argument on CLI
- Provide unit tests and (a few) functional tests
- Provide PEP8 tests
- Use tox to run the tests
- CLI: provide a `--config-file` option
- Turn the gitlab module into a proper package
- Allow projects to be updated
- Use more pythonic names for some methods
- 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`
- Sphinx: don't hardcode the version in `conf.py`

#### 6.8.71 v0.8 (2014-10-26)

- Better python 2.6 and python 3 support
- Timeout support in HTTP requests
- `Gitlab.get()` raised `GitlabListError` instead of `GitlabGetError`
- Support api-objects which don't have id in api response
- Add `ProjectLabel` and `ProjectFile` classes
- Moved url attributes to separate list
- Added list for delete attributes

### 6.8.72 v0.7 (2014-08-21)

- Fix license classifier in `setup.py`
- Fix encoding error when printing to redirected output
- Fix encoding error when updating with redirected output
- Add support for UserKey listing and deletion
- Add support for branches creation and deletion
- Support `state_event` in ProjectMilestone (#30)
- Support namespace/name for project id (#28)
- Fix handling of boolean values (#22)

### 6.8.73 v0.6 (2014-01-16)

- IDs can be unicode (#15)
- ProjectMember: constructor should not create a User object
- Add support for extra parameters when listing all projects (#12)
- Projects listing: explicitly define arguments for pagination

### 6.8.74 v0.5 (2013-12-26)

- Add SSH key for user
- Fix comments
- Add support for project events
- Support creation of projects for users
- Project: add methods for create/update/delete files
- Support projects listing: search, all, owned
- System hooks can't be updated
- Project.archive(): download tarball of the project
- Define new optional attributes for user creation
- Provide constants for access permissions in groups

### 6.8.75 v0.4 (2013-09-26)

- Fix strings encoding (Closes #6)
- Allow to get a project commit (GitLab 6.1)
- ProjectMergeRequest: fix Note() method
- Gitlab 6.1 methods: diff, blob (commit), tree, blob (project)
- Add support for Gitlab 6.1 group members

### 6.8.76 v0.3 (2013-08-27)

- Use PRIVATE-TOKEN header for passing the auth token
- provide an AUTHORS file
- cli: support ssl\_verify config option
- Add ssl\_verify option to Gitlab object. Defaults to True
- Correct url for merge requests API.

### 6.8.77 v0.2 (2013-08-08)

- provide a pip requirements.txt
- drop some debug statements

### 6.8.78 v0.1 (2013-07-08)

- Initial release

## 6.9 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.

### 6.9.1 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.

### 6.9.2 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.

### 6.9.3 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.

### 6.9.4 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.

### 6.9.5 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.

## 6.9.6 Changes from 1.2 to 1.3

- gitlab.Gitlab objects can be used as context managers in a with block.

## 6.9.7 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

## 6.9.8 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()
```



### 6.9.9 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.

### 6.9.10 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.

### 6.9.11 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>

## 6.10 FAQ

### 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*.

### I get an `AttributeError` when accessing attributes of an object retrieved via a `list()` call.

Fetching a list of objects, doesn't 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)
```

### How can I clone the repository of a project?

python-gitlab doesn't 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])
```

### I get an `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.

### I passed `all=True` (or `--all` via the CLI) to the API and I still cannot see all items returned.

Use `get_all=True` (or `--get-all` via the CLI). See *Pagination* for more details.

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### g

- `gitlab.base`, [422](#)
- `gitlab.cli`, [424](#)
- `gitlab.config`, [425](#)
- `gitlab.const`, [426](#)
- `gitlab.exceptions`, [427](#)
- `gitlab.mixins`, [444](#)
- `gitlab.utils`, [453](#)
- `gitlab.v4`, [422](#)
- `gitlab.v4.objects`, [182](#)



## Symbols

- access-level
  - gitlab-group-access-request-approve command line option, [489](#)
  - gitlab-group-access-token-create command line option, [490](#)
  - gitlab-group-invitation-create command line option, [522](#)
  - gitlab-group-invitation-update command line option, [524](#)
  - gitlab-group-member-create command line option, [530](#)
  - gitlab-group-member-update command line option, [531](#)
  - gitlab-project-access-request-approve command line option, [576](#)
  - gitlab-project-access-token-create command line option, [577](#)
  - gitlab-project-invitation-create command line option, [625](#)
  - gitlab-project-invitation-update command line option, [626](#)
  - gitlab-project-member-create command line option, [657](#)
  - gitlab-project-member-update command line option, [659](#)
  - gitlab-runner-create command line option, [749](#)
  - gitlab-runner-update command line option, [751](#)
- action
  - gitlab-event-list command line option, [476](#)
  - gitlab-project-event-list command line option, [614](#)
  - gitlab-todo-list command line option, [758](#)
  - gitlab-user-event-list command line option, [770](#)
- actions
  - gitlab-project-commit-create command line option, [595](#)
- active
  - gitlab-project-pipeline-schedule-create command line option, [703](#)
  - gitlab-project-pipeline-schedule-update command line option, [705](#)
  - gitlab-runner-create command line option, [749](#)
  - gitlab-runner-update command line option, [750](#)
  - gitlab-user-list command line option, [764](#)
- admin
  - gitlab-user-create command line option, [762](#)
  - gitlab-user-update command line option, [766](#)
- after
  - gitlab-event-list command line option, [476](#)
  - gitlab-project-event-list command line option, [614](#)
  - gitlab-user-event-list command line option, [770](#)
- after-sign-out-path
  - gitlab-application-settings-update command line option, [465](#)
- all
  - gitlab-application-list command line option, [460](#)
  - gitlab-audit-event-list command line option, [466](#)
  - gitlab-broadcast-message-list command line option, [468](#)
  - gitlab-current-user-email-list command line option, [471](#)
  - gitlab-current-user-gpg-key-list command line option, [472](#)
  - gitlab-current-user-key-list command line option, [473](#)
  - gitlab-deploy-key-list command line option, [474](#)
  - gitlab-deploy-token-list command line option, [475](#)
  - gitlab-dockerfile-list command line

- option, [476](#)
- gitlab-event-list command line option, [477](#)
- gitlab-feature-list command line option, [477](#)
- gitlab-geo-node-list command line option, [480](#)
- gitlab-gitignore-list command line option, [481](#)
- gitlab-gitlabciyaml-list command line option, [482](#)
- gitlab-group-access-request-list command line option, [490](#)
- gitlab-group-access-token-list command line option, [491](#)
- gitlab-group-audit-event-list command line option, [492](#)
- gitlab-group-badge-list command line option, [494](#)
- gitlab-group-billable-member-list command line option, [495](#)
- gitlab-group-billable-member-membership-list command line option, [496](#)
- gitlab-group-board-list command line option, [497](#)
- gitlab-group-board-list-list command line option, [499](#)
- gitlab-group-cluster-list command line option, [501](#)
- gitlab-group-custom-attribute-list command line option, [503](#)
- gitlab-group-deploy-token-list command line option, [504](#)
- gitlab-group-descendant-group-list command line option, [505](#)
- gitlab-group-epic-award-emoji-list command line option, [510](#)
- gitlab-group-epic-issue-list command line option, [513](#)
- gitlab-group-epic-list command line option, [507](#)
- gitlab-group-epic-note-award-emoji-list command line option, [517](#)
- gitlab-group-epic-note-list command line option, [515](#)
- gitlab-group-epic-resource-label-event-list command line option, [518](#)
- gitlab-group-hook-list command line option, [521](#)
- gitlab-group-invitation-list command line option, [524](#)
- gitlab-group-issue-list command line option, [525](#)
- gitlab-group-iteration-list command line option, [526](#)
- gitlab-group-label-list command line option, [528](#)
- gitlab-group-list command line option, [486](#)
- gitlab-group-member-all-list command line option, [532](#)
- gitlab-group-member-list command line option, [531](#)
- gitlab-group-merge-request-list command line option, [534](#)
- gitlab-group-milestone-list command line option, [536](#)
- gitlab-group-package-list command line option, [539](#)
- gitlab-group-project-list command line option, [540](#)
- gitlab-group-runner-list command line option, [543](#)
- gitlab-group-subgroup-list command line option, [544](#)
- gitlab-group-variable-list command line option, [545](#)
- gitlab-group-wiki-list command line option, [547](#)
- gitlab-hook-list command line option, [549](#)
- gitlab-issue-list command line option, [550](#)
- gitlab-ldap-group-list command line option, [552](#)
- gitlab-license-list command line option, [553](#)
- gitlab-merge-request-list command line option, [555](#)
- gitlab-namespace-list command line option, [556](#)
- gitlab-pages-domain-list command line option, [558](#)
- gitlab-personal-access-token-list command line option, [558](#)
- gitlab-project-access-request-list command line option, [577](#)
- gitlab-project-access-token-list command line option, [578](#)
- gitlab-project-approval-rule-list command line option, [581](#)
- gitlab-project-audit-event-list command line option, [583](#)
- gitlab-project-badge-list command line option, [585](#)
- gitlab-project-board-list command line option, [587](#)
- gitlab-project-board-list-list command line option, [589](#)



- gitlab-project-branch-list command line option, [591](#)
- gitlab-project-cluster-list command line option, [594](#)
- gitlab-project-commit-comment-list command line option, [599](#)
- gitlab-project-commit-discussion-list command line option, [600](#)
- gitlab-project-commit-list command line option, [597](#)
- gitlab-project-commit-status-list command line option, [604](#)
- gitlab-project-custom-attribute-list command line option, [605](#)
- gitlab-project-deploy-token-list command line option, [606](#)
- gitlab-project-deployment-list command line option, [608](#)
- gitlab-project-deployment-merge-request-list command line option, [611](#)
- gitlab-project-environment-list command line option, [613](#)
- gitlab-project-event-list command line option, [614](#)
- gitlab-project-fork-list command line option, [619](#)
- gitlab-project-hook-list command line option, [621](#)
- gitlab-project-integration-list command line option, [624](#)
- gitlab-project-invitation-list command line option, [626](#)
- gitlab-project-issue-award-emoji-list command line option, [636](#)
- gitlab-project-issue-discussion-list command line option, [638](#)
- gitlab-project-issue-link-list command line option, [641](#)
- gitlab-project-issue-list command line option, [630](#)
- gitlab-project-issue-note-award-emoji-list command line option, [645](#)
- gitlab-project-issue-note-list command line option, [643](#)
- gitlab-project-issue-resource-label-event-list command line option, [646](#)
- gitlab-project-issue-resource-milestone-event-list command line option, [647](#)
- gitlab-project-issue-resource-state-event-list command line option, [648](#)
- gitlab-project-job-list command line option, [651](#)
- gitlab-project-key-list command line option, [654](#)
- gitlab-project-label-list command line option, [656](#)
- gitlab-project-list command line option, [566](#)
- gitlab-project-member-all-list command line option, [660](#)
- gitlab-project-member-list command line option, [658](#)
- gitlab-project-merge-request-approval-rule-list command line option, [672](#)
- gitlab-project-merge-request-award-emoji-list command line option, [676](#)
- gitlab-project-merge-request-diff-list command line option, [677](#)
- gitlab-project-merge-request-discussion-list command line option, [678](#)
- gitlab-project-merge-request-list command line option, [665](#)
- gitlab-project-merge-request-note-award-emoji-list command line option, [685](#)
- gitlab-project-merge-request-note-list command line option, [682](#)
- gitlab-project-merge-request-pipeline-list command line option, [685](#)
- gitlab-project-merge-request-resource-label-event-list command line option, [686](#)
- gitlab-project-merge-request-resource-milestone-event-list command line option, [687](#)
- gitlab-project-merge-request-resource-state-event-list command line option, [688](#)
- gitlab-project-merge-train-list command line option, [689](#)
- gitlab-project-milestone-list command line option, [691](#)
- gitlab-project-note-list command line option, [693](#)
- gitlab-project-package-file-list command line option, [696](#)
- gitlab-project-package-list command line option, [696](#)
- gitlab-project-pages-domain-list command line option, [698](#)
- gitlab-project-pipeline-bridge-list command line option, [701](#)
- gitlab-project-pipeline-job-list command line option, [702](#)
- gitlab-project-pipeline-list command line option, [700](#)
- gitlab-project-pipeline-schedule-list command line option, [704](#)
- gitlab-project-pipeline-variable-list command line option, [708](#)
- gitlab-project-protected-branch-list command line option, [710](#)

`gitlab-project-protected-environment-list`  
command line option, [712](#)

`gitlab-project-protected-tag-list`  
command line option, [713](#)

`gitlab-project-registry-repository-list`  
command line option, [716](#)

`gitlab-project-registry-tag-list`  
command line option, [718](#)

`gitlab-project-release-link-list`  
command line option, [722](#)

`gitlab-project-release-list` command  
line option, [720](#)

`gitlab-project-remote-mirror-list`  
command line option, [724](#)

`gitlab-project-runner-list` command line  
option, [725](#)

`gitlab-project-service-list` command  
line option, [727](#)

`gitlab-project-snippet-award-emoji-list`  
command line option, [731](#)

`gitlab-project-snippet-discussion-list`  
command line option, [733](#)

`gitlab-project-snippet-list` command  
line option, [729](#)

`gitlab-project-snippet-note-award-emoji-list`  
command line option, [739](#)

`gitlab-project-snippet-note-list`  
command line option, [737](#)

`gitlab-project-tag-list` command line  
option, [741](#)

`gitlab-project-trigger-list` command  
line option, [742](#)

`gitlab-project-user-list` command line  
option, [743](#)

`gitlab-project-variable-list` command  
line option, [745](#)

`gitlab-project-wiki-list` command line  
option, [747](#)

`gitlab-runner-all-list` command line  
option, [751](#)

`gitlab-runner-job-list` command line  
option, [752](#)

`gitlab-runner-list` command line option,  
[750](#)

`gitlab-shared-project-list` command line  
option, [753](#)

`gitlab-snippet-list` command line option,  
[755](#)

`gitlab-starred-project-list` command  
line option, [757](#)

`gitlab-todo-list` command line option, [758](#)

`gitlab-topic-list` command line option,  
[760](#)

`gitlab-user-activities-list` command  
line option, [767](#)

`gitlab-user-custom-attribute-list`  
command line option, [768](#)

`gitlab-user-email-list` command line  
option, [770](#)

`gitlab-user-event-list` command line  
option, [770](#)

`gitlab-user-gpg-key-list` command line  
option, [772](#)

`gitlab-user-impersonation-token-list`  
command line option, [773](#)

`gitlab-user-key-list` command line  
option, [775](#)

`gitlab-user-list` command line option, [764](#)

`gitlab-user-membership-list` command  
line option, [776](#)

`gitlab-user-project-list` command line  
option, [779](#)

`gitlab-variable-list` command line  
option, [781](#)

`--all-available`  
`gitlab-group-descendant-group-list`  
command line option, [505](#)

`gitlab-group-list` command line option,  
[485](#)

`gitlab-group-subgroup-list` command line  
option, [544](#)

`--allow-local-requests-from-hooks-and-services`  
`gitlab-application-settings-update`  
command line option, [465](#)

`--allow-local-requests-from-system-hooks`  
`gitlab-application-settings-update`  
command line option, [465](#)

`--allow-local-requests-from-web-hooks-and-services`  
`gitlab-application-settings-update`  
command line option, [465](#)

`--allow-maintainer-to-push`  
`gitlab-project-merge-request-create`  
command line option, [663](#)

`gitlab-project-merge-request-update`  
command line option, [669](#)

`--allow-merge-on-skipped-pipeline`  
`gitlab-project-create` command line  
option, [562](#)

`gitlab-project-update` command line  
option, [573](#)

`--allowed-to-merge`  
`gitlab-project-protected-branch-create`  
command line option, [709](#)

`--allowed-to-push`  
`gitlab-project-protected-branch-create`  
command line option, [709](#)

`--allowed-to-unprotect`  
`gitlab-project-protected-branch-create`

- command line option, [709](#)
- analytics-access-level
  - gitlab-project-create command line option, [562](#)
  - gitlab-project-update command line option, [573](#)
- api-version
  - gitlab command line option, [459](#)
- approval-project-rule-id
  - gitlab-project-merge-request-approval-rule-create option, [778](#)
  - command line option, [671](#)
- approval-rule-id
  - gitlab-project-merge-request-approval-rule-update option, [560](#)
  - command line option, [673](#)
- approval-rules
  - gitlab-project-protected-environment-create-asset-proxy-allowlist command line option, [711](#)
- approvals-before-merge
  - gitlab-project-approval-update command line option, [579](#)
  - gitlab-project-create command line option, [562](#)
  - gitlab-project-update command line option, [573](#)
- approvals-required
  - gitlab-project-approval-rule-create command line option, [580](#)
  - gitlab-project-merge-request-approval-rule-create command line option, [671](#)
  - gitlab-project-merge-request-approval-rule-update command line option, [673](#)
  - gitlab-project-merge-request-approval-update command line option, [670](#)
- approved-by-ids
  - gitlab-group-merge-request-list command line option, [533](#)
  - gitlab-merge-request-list command line option, [554](#)
  - gitlab-project-deployment-merge-request-list command line option, [610](#)
  - gitlab-project-merge-request-list command line option, [665](#)
- approver-ids
  - gitlab-group-merge-request-list command line option, [533](#)
  - gitlab-merge-request-list command line option, [554](#)
  - gitlab-project-deployment-merge-request-list command line option, [610](#)
  - gitlab-project-merge-request-list command line option, [665](#)
- archived
  - gitlab-group-project-list command line option, [539](#)
- gitlab-project-fork-list command line option, [618](#)
- gitlab-project-list command line option, [566](#)
- gitlab-shared-project-list command line option, [753](#)
- gitlab-starred-project-list command line option, [756](#)
- gitlab-user-project-list command line option, [756](#)
- artifact-path
  - gitlab-project-artifact command line option, [582](#)
  - gitlab-project-artifact-raw command line option, [582](#)
- gitlab-application-settings-update command line option, [465](#)
- asset-proxy-enabled
  - gitlab-application-settings-update command line option, [465](#)
- asset-proxy-url
  - gitlab-application-settings-update command line option, [465](#)
- asset-proxy-whitelist
  - gitlab-application-settings-update command line option, [465](#)
- gitlab-project-release-create command line option, [719](#)
- assignee-id
  - gitlab-group-issue-list command line option, [525](#)
  - gitlab-group-merge-request-list command line option, [533](#)
  - gitlab-issue-list command line option, [550](#)
  - gitlab-merge-request-list command line option, [554](#)
  - gitlab-project-deployment-merge-request-list command line option, [610](#)
  - gitlab-project-issue-create command line option, [628](#)
  - gitlab-project-issue-list command line option, [630](#)
  - gitlab-project-issue-update command line option, [634](#)
  - gitlab-project-merge-request-create command line option, [663](#)
  - gitlab-project-merge-request-list command line option, [665](#)
  - gitlab-project-merge-request-update command line option, [669](#)
- assignee-ids

- gitlab-project-issue-create command
  - line option, [628](#)
- gitlab-project-issue-update command
  - line option, [634](#)
- author-email
  - gitlab-project-commit-create command
    - line option, [595](#)
  - gitlab-project-file-create command line
    - option, [616](#)
  - gitlab-project-file-update command line
    - option, [618](#)
- author-email-regex
  - gitlab-group-push-rules-create command
    - line option, [541](#)
  - gitlab-group-push-rules-update command
    - line option, [542](#)
  - gitlab-project-push-rules-create
    - command line option, [714](#)
  - gitlab-project-push-rules-update
    - command line option, [715](#)
- author-id
  - gitlab-group-epic-list command line
    - option, [507](#)
  - gitlab-group-issue-list command line
    - option, [525](#)
  - gitlab-group-merge-request-list command
    - line option, [533](#)
  - gitlab-issue-list command line option,
    - [550](#)
  - gitlab-merge-request-list command line
    - option, [554](#)
  - gitlab-project-deployment-merge-request-list
    - command line option, [610](#)
  - gitlab-project-issue-list command line
    - option, [630](#)
  - gitlab-project-merge-request-list
    - command line option, [665](#)
  - gitlab-todo-list command line option, [758](#)
- author-name
  - gitlab-project-commit-create command
    - line option, [596](#)
  - gitlab-project-file-create command line
    - option, [616](#)
  - gitlab-project-file-update command line
    - option, [618](#)
- author-username
  - gitlab-merge-request-list command line
    - option, [554](#)
  - gitlab-project-deployment-merge-request-list
    - command line option, [610](#)
- auto-cancel-pending-pipelines
  - gitlab-project-create command line
    - option, [562](#)
  - gitlab-project-update command line
    - option, [573](#)
- auto-devops-deploy-strategy
  - gitlab-project-create command line
    - option, [562](#)
  - gitlab-project-update command line
    - option, [573](#)
- auto-devops-enabled
  - gitlab-group-create command line option,
    - [484](#)
  - gitlab-group-update command line option,
    - [488](#)
  - gitlab-project-create command line
    - option, [562](#)
  - gitlab-project-update command line
    - option, [573](#)
- autoclose-referenced-issues
  - gitlab-project-create command line
    - option, [562](#)
  - gitlab-project-update command line
    - option, [573](#)
- avatar
  - gitlab-group-create command line option,
    - [484](#)
  - gitlab-group-update command line option,
    - [488](#)
  - gitlab-project-create command line
    - option, [562](#)
  - gitlab-project-update command line
    - option, [573](#)
  - gitlab-topic-create command line option,
    - [759](#)
  - gitlab-topic-update command line option,
    - [760](#)
  - gitlab-user-create command line option,
    - [763](#)
  - gitlab-user-update command line option,
    - [766](#)
- before
  - gitlab-event-list command line option,
    - [476](#)
  - gitlab-project-event-list command line
    - option, [614](#)
  - gitlab-user-event-list command line
    - option, [770](#)
- bio
  - gitlab-user-create command line option,
    - [762](#)
  - gitlab-user-update command line option,
    - [766](#)
- blocked
  - gitlab-user-list command line option, [764](#)
- board-id
  - gitlab-group-board-list-create command
    - line option, [498](#)

- gitlab-group-board-list-delete command line option, [498](#)
- gitlab-group-board-list-get command line option, [499](#)
- gitlab-group-board-list-list command line option, [499](#)
- gitlab-group-board-list-update command line option, [499](#)
- gitlab-project-board-list-create command line option, [588](#)
- gitlab-project-board-list-delete command line option, [588](#)
- gitlab-project-board-list-get command line option, [588](#)
- gitlab-project-board-list-list command line option, [589](#)
- gitlab-project-board-list-update command line option, [589](#)
- body
  - gitlab-group-epic-discussion-note-create command line option, [510](#)
  - gitlab-group-epic-discussion-note-update command line option, [511](#)
  - gitlab-group-epic-note-create command line option, [514](#)
  - gitlab-group-epic-note-update command line option, [515](#)
  - gitlab-project-commit-discussion-create command line option, [600](#)
  - gitlab-project-commit-discussion-note-create command line option, [601](#)
  - gitlab-project-commit-discussion-note-update command line option, [602](#)
  - gitlab-project-issue-discussion-create command line option, [637](#)
  - gitlab-project-issue-discussion-note-create-build-coverage-regex command line option, [638](#)
  - gitlab-project-issue-discussion-note-update command line option, [640](#)
  - gitlab-project-issue-note-create command line option, [641](#)
  - gitlab-project-issue-note-update command line option, [643](#)
  - gitlab-project-merge-request-discussion-create command line option, [677](#)
  - gitlab-project-merge-request-discussion-note-create-build-timeout command line option, [679](#)
  - gitlab-project-merge-request-discussion-note-update command line option, [680](#)
  - gitlab-project-merge-request-note-create command line option, [681](#)
  - gitlab-project-merge-request-note-update command line option, [683](#)
  - gitlab-project-snippet-discussion-create command line option, [732](#)
  - gitlab-project-snippet-discussion-note-create command line option, [734](#)
  - gitlab-project-snippet-discussion-note-update command line option, [735](#)
  - gitlab-project-snippet-note-create command line option, [735](#)
  - gitlab-project-snippet-note-update command line option, [737](#)
- branch
  - gitlab-project-branch-create command line option, [590](#)
  - gitlab-project-commit-cherry-pick command line option, [595](#)
  - gitlab-project-commit-create command line option, [595](#)
  - gitlab-project-commit-revert command line option, [598](#)
  - gitlab-project-file-create command line option, [616](#)
  - gitlab-project-file-delete command line option, [616](#)
  - gitlab-project-file-update command line option, [617](#)
  - gitlab-project-update-submodule command line option, [575](#)
- branch-name-regex
  - gitlab-group-push-rules-create command line option, [541](#)
  - gitlab-group-push-rules-update command line option, [542](#)
  - gitlab-project-push-rules-create command line option, [714](#)
  - gitlab-project-push-rules-update command line option, [715](#)
- build-coverage-regex
  - gitlab-project-create command line option, [562](#)
  - gitlab-project-update command line option, [573](#)
- build-git-strategy
  - gitlab-project-create command line option, [562](#)
  - gitlab-project-update command line option, [573](#)
- build-timeout
  - gitlab-project-create command line option, [562](#)
  - gitlab-project-update command line option, [573](#)
- builds-access-level
  - gitlab-project-create command line option, [562](#)
  - gitlab-project-update command line

- option, [573](#)
- `--builds-enabled`
  - `gitlab-user-project-create` command line option, [777](#)
- `--can-create-group`
  - `gitlab-user-create` command line option, [762](#)
  - `gitlab-user-update` command line option, [766](#)
- `--can-push`
  - `gitlab-project-key-create` command line option, [653](#)
  - `gitlab-project-key-update` command line option, [654](#)
- `--certificate`
  - `gitlab-project-pages-domain-create` command line option, [697](#)
  - `gitlab-project-pages-domain-update` command line option, [698](#)
- `--ci-config-path`
  - `gitlab-project-create` command line option, [562](#)
  - `gitlab-project-update` command line option, [573](#)
- `--ci-default-git-depth`
  - `gitlab-project-update` command line option, [573](#)
- `--ci-forward-deployment-enabled`
  - `gitlab-project-update` command line option, [573](#)
- `--close-issue`
  - `gitlab-group-notification-settings-update` command line option, [538](#)
  - `gitlab-notification-settings-update` command line option, [557](#)
  - `gitlab-project-notification-settings-update` command line option, [694](#)
- `--close-merge-request`
  - `gitlab-group-notification-settings-update` command line option, [538](#)
  - `gitlab-notification-settings-update` command line option, [557](#)
  - `gitlab-project-notification-settings-update` command line option, [694](#)
- `--cn`
  - `gitlab-group-add-ldap-group-link` command line option, [483](#)
  - `gitlab-group-delete-ldap-group-link` command line option, [485](#)
- `--code-owner-approval-required`
  - `gitlab-project-protected-branch-create` command line option, [709](#)
- `--color`
  - `gitlab-broadcast-message-create` command line option, [467](#)
- `gitlab-broadcast-message-update` command line option, [468](#)
- `gitlab-group-label-create` command line option, [527](#)
- `gitlab-group-label-update` command line option, [529](#)
- `gitlab-project-label-create` command line option, [655](#)
- `gitlab-project-label-update` command line option, [657](#)
- `--color-scheme-id`
  - `gitlab-user-create` command line option, [763](#)
  - `gitlab-user-update` command line option, [766](#)
- `--commit-committer-check`
  - `gitlab-group-push-rules-create` command line option, [541](#)
  - `gitlab-group-push-rules-update` command line option, [542](#)
  - `gitlab-project-push-rules-create` command line option, [714](#)
  - `gitlab-project-push-rules-update` command line option, [715](#)
- `--commit-id`
  - `gitlab-project-commit-comment-create` command line option, [598](#)
  - `gitlab-project-commit-comment-list` command line option, [599](#)
  - `gitlab-project-commit-discussion-create` command line option, [600](#)
  - `gitlab-project-commit-discussion-get` command line option, [600](#)
  - `gitlab-project-commit-discussion-list` command line option, [600](#)
  - `gitlab-project-commit-discussion-note-create` command line option, [601](#)
  - `gitlab-project-commit-discussion-note-delete` command line option, [601](#)
  - `gitlab-project-commit-discussion-note-get` command line option, [602](#)
  - `gitlab-project-commit-discussion-note-update` command line option, [602](#)
  - `gitlab-project-commit-status-create` command line option, [603](#)
  - `gitlab-project-commit-status-list` command line option, [604](#)
- `--commit-message`
  - `gitlab-project-commit-create` command line option, [595](#)
  - `gitlab-project-file-create` command line option, [616](#)
  - `gitlab-project-file-delete` command line



- option, [616](#)
- gitlab-project-file-update command line option, [617](#)
- commit-message-negative-regex
  - gitlab-group-push-rules-create command line option, [541](#)
  - gitlab-group-push-rules-update command line option, [542](#)
  - gitlab-project-push-rules-create command line option, [714](#)
  - gitlab-project-push-rules-update command line option, [715](#)
- commit-message-regex
  - gitlab-group-push-rules-create command line option, [541](#)
  - gitlab-group-push-rules-update command line option, [542](#)
  - gitlab-project-push-rules-create command line option, [714](#)
  - gitlab-project-push-rules-update command line option, [715](#)
- commit-sha
  - gitlab-project-update-submodule command line option, [575](#)
- confidential
  - gitlab-application-create command line option, [460](#)
  - gitlab-project-issue-create command line option, [628](#)
  - gitlab-project-issue-update command line option, [634](#)
- confidential-issues-events
  - gitlab-group-hook-create command line option, [519](#)
  - gitlab-group-hook-update command line option, [521](#)
  - gitlab-project-hook-create command line option, [620](#)
  - gitlab-project-hook-update command line option, [622](#)
- confidential-note-events
  - gitlab-group-hook-create command line option, [519](#)
  - gitlab-group-hook-update command line option, [521](#)
- config-file
  - gitlab command line option, [458](#)
- container-expiration-policy-attributes
  - gitlab-project-create command line option, [562](#)
  - gitlab-project-update command line option, [573](#)
- container-registry-enabled
  - gitlab-project-create command line option, [562](#)
- gitlab-project-update command line option, [573](#)
- container-registry-token-expire-delay
  - gitlab-application-settings-update command line option, [465](#)
- content
  - gitlab-ci-lint-create command line option, [469](#)
  - gitlab-ci-lint-validate command line option, [469](#)
  - gitlab-group-wiki-create command line option, [546](#)
  - gitlab-group-wiki-update command line option, [548](#)
  - gitlab-project-ci-lint-create command line option, [591](#)
  - gitlab-project-ci-lint-validate command line option, [592](#)
  - gitlab-project-file-create command line option, [616](#)
  - gitlab-project-file-update command line option, [617](#)
  - gitlab-project-snippet-create command line option, [728](#)
  - gitlab-project-snippet-update command line option, [729](#)
  - gitlab-project-wiki-create command line option, [746](#)
  - gitlab-project-wiki-update command line option, [748](#)
  - gitlab-snippet-create command line option, [754](#)
  - gitlab-snippet-update command line option, [756](#)
- context
  - gitlab-project-commit-status-create command line option, [603](#)
- coverage
  - gitlab-project-commit-status-create command line option, [603](#)
- create-access-level
  - gitlab-project-protected-tag-create command line option, [712](#)
- created-after
  - gitlab-audit-event-list command line option, [466](#)
  - gitlab-group-audit-event-list command line option, [492](#)
  - gitlab-group-issue-list command line option, [525](#)
  - gitlab-group-merge-request-list command line option, [533](#)
  - gitlab-issue-list command line option,

- 550
- gitlab-merge-request-list command line option, 554
- gitlab-project-audit-event-list command line option, 583
- gitlab-project-deployment-merge-request-listcron command line option, 610
- gitlab-project-issue-list command line option, 630
- gitlab-project-merge-request-list command line option, 664
- created-at
  - gitlab-application-settings-update command line option, 464
  - gitlab-group-epic-discussion-note-create command line option, 510
  - gitlab-group-epic-note-create command line option, 514
  - gitlab-project-commit-discussion-create command line option, 600
  - gitlab-project-commit-discussion-note-create command line option, 601
  - gitlab-project-issue-create command line option, 628
  - gitlab-project-issue-discussion-create command line option, 637
  - gitlab-project-issue-discussion-note-create command line option, 638
  - gitlab-project-issue-note-create command line option, 641
  - gitlab-project-merge-request-discussion-create command line option, 677
  - gitlab-project-merge-request-discussion-note-create command line option, 679
  - gitlab-project-snippet-discussion-create command line option, 732
  - gitlab-project-snippet-discussion-note-create command line option, 734
- created-before
  - gitlab-audit-event-list command line option, 466
  - gitlab-group-audit-event-list command line option, 492
  - gitlab-group-issue-list command line option, 525
  - gitlab-group-merge-request-list command line option, 533
  - gitlab-issue-list command line option, 550
  - gitlab-merge-request-list command line option, 554
  - gitlab-project-audit-event-list command line option, 583
  - gitlab-project-deployment-merge-request-list command line option, 610
  - gitlab-project-issue-list command line option, 630
  - gitlab-project-merge-request-list command line option, 664
  - gitlab-project-pipeline-schedule-create command line option, 703
  - gitlab-project-pipeline-schedule-update command line option, 705
- cron-timezone
  - gitlab-project-pipeline-schedule-create command line option, 703
  - gitlab-project-pipeline-schedule-update command line option, 705
- custom-attributes
  - gitlab-user-list command line option, 764
- debug
  - gitlab command line option, 458
- default-branch
  - gitlab-project-create command line option, 562
  - gitlab-project-update command line option, 573
  - gitlab-user-project-create command line option, 777
- default-branch-protection
  - gitlab-application-settings-update command line option, 464
  - gitlab-group-create command line option, 484
  - gitlab-group-update command line option, 488
- default-group-visibility
  - gitlab-application-settings-update command line option, 464
- default-project-visibility
  - gitlab-application-settings-update command line option, 464
- default-projects-limit
  - gitlab-application-settings-update command line option, 464
- default-snippet-visibility
  - gitlab-application-settings-update command line option, 464
- deny-delete-tag
  - gitlab-group-push-rules-create command line option, 541
  - gitlab-group-push-rules-update command line option, 542
  - gitlab-project-push-rules-create command line option, 714
  - gitlab-project-push-rules-update command line option, 715



---

```

--deploy-access-levels
    gitlab-project-protected-environment-create
        command line option, 710
--deployed-after
    gitlab-merge-request-list command line
        option, 555
    gitlab-project-deployment-merge-request-list
        command line option, 611
--deployed-before
    gitlab-merge-request-list command line
        option, 555
    gitlab-project-deployment-merge-request-list
        command line option, 610
--deployment-events
    gitlab-group-hook-create command line
        option, 519
    gitlab-group-hook-update command line
        option, 521
--deployment-id
    gitlab-project-deployment-merge-request-list
        command line option, 610
--description
    gitlab-application-appearance-update
        command line option, 461
    gitlab-group-create command line option,
        484
    gitlab-group-epic-create command line
        option, 506
    gitlab-group-epic-update command line
        option, 508
    gitlab-group-label-create command line
        option, 527
    gitlab-group-label-update command line
        option, 529
    gitlab-group-milestone-create command
        line option, 534
    gitlab-group-milestone-update command
        line option, 536
    gitlab-group-update command line option,
        488
    gitlab-project-commit-status-create
        command line option, 603
    gitlab-project-create command line
        option, 562
    gitlab-project-export-create command
        line option, 614
    gitlab-project-issue-create command
        line option, 628
    gitlab-project-issue-update command
        line option, 634
    gitlab-project-label-create command
        line option, 655
    gitlab-project-label-update command
        line option, 657
    gitlab-project-merge-request-create
        command line option, 663
    gitlab-project-merge-request-update
        command line option, 669
    gitlab-project-milestone-create command
        line option, 690
    gitlab-project-milestone-update command
        line option, 692
    gitlab-project-pipeline-schedule-create
        command line option, 703
    gitlab-project-pipeline-schedule-update
        command line option, 705
    gitlab-project-release-create command
        line option, 719
    gitlab-project-release-update command
        line option, 720
    gitlab-project-snippet-create command
        line option, 728
    gitlab-project-snippet-update command
        line option, 729
    gitlab-project-trigger-create command
        line option, 741
    gitlab-project-trigger-update command
        line option, 743
    gitlab-project-update command line
        option, 573
    gitlab-runner-create command line
        option, 749
    gitlab-runner-update command line
        option, 750
    gitlab-topic-create command line option,
        759
    gitlab-topic-update command line option,
        760
    gitlab-user-project-create command line
        option, 777
--disable-overriding-approvers-per-merge-request
    gitlab-project-approval-update command
        line option, 580
--disabled-oauth-sign-in-sources
    gitlab-application-settings-update
        command line option, 464
--discussion-id
    gitlab-group-epic-discussion-note-create
        command line option, 510
    gitlab-group-epic-discussion-note-delete
        command line option, 510
    gitlab-group-epic-discussion-note-get
        command line option, 511
    gitlab-group-epic-discussion-note-update
        command line option, 511
    gitlab-project-commit-discussion-note-create
        command line option, 601
    gitlab-project-commit-discussion-note-delete

```

command line option, [602](#)

gitlab-project-commit-discussion-note-get command line option, [602](#)

gitlab-project-commit-discussion-note-update command line option, [602](#)

gitlab-project-issue-discussion-note-create command line option, [638](#)

gitlab-project-issue-discussion-note-delete command line option, [639](#)

gitlab-project-issue-discussion-note-get command line option, [639](#)

gitlab-project-issue-discussion-note-update command line option, [640](#)

gitlab-project-merge-request-discussion-note-create command line option, [679](#)

gitlab-project-merge-request-discussion-note-delete command line option, [680](#)

gitlab-project-merge-request-discussion-note-get command line option, [680](#)

gitlab-project-merge-request-discussion-note-update command line option, [680](#)

gitlab-project-snippet-discussion-note-create command line option, [734](#)

gitlab-project-snippet-discussion-note-delete command line option, [734](#)

gitlab-project-snippet-discussion-note-get command line option, [734](#)

gitlab-project-snippet-discussion-note-update command line option, [735](#)

--discussion-locked

gitlab-project-issue-update command line option, [634](#)

gitlab-project-merge-request-update command line option, [669](#)

--discussion-to-resolve

gitlab-project-issue-create command line option, [628](#)

--domain

gitlab-group-cluster-create command line option, [500](#)

gitlab-group-cluster-update command line option, [502](#)

gitlab-project-cluster-create command line option, [593](#)

gitlab-project-cluster-update command line option, [594](#)

gitlab-project-pages-domain-create command line option, [697](#)

gitlab-project-pages-domain-delete command line option, [697](#)

gitlab-project-pages-domain-get command line option, [698](#)

gitlab-project-pages-domain-update command line option, [698](#)

--domain-allowlist

gitlab-application-settings-update command line option, [464](#)

domain-blacklist

gitlab-application-settings-update command line option, [464](#)

--domain-blacklist-enabled

gitlab-application-settings-update command line option, [464](#)

--domain-denylist

gitlab-application-settings-update command line option, [464](#)

--domain-denylist-enabled

gitlab-application-settings-update command line option, [464](#)

domain-whitelist

gitlab-application-settings-update command line option, [464](#)

--dry-run

gitlab-project-ci-lint-create command line option, [591](#)

gitlab-project-ci-lint-validate command line option, [592](#)

ecdsa-key-restriction

gitlab-application-settings-update command line option, [465](#)

--due-date

gitlab-group-milestone-create command line option, [534](#)

gitlab-group-milestone-update command line option, [536](#)

gitlab-project-issue-create command line option, [628](#)

gitlab-project-issue-update command line option, [634](#)

gitlab-project-milestone-create command line option, [690](#)

gitlab-project-milestone-update command line option, [692](#)

--duration

gitlab-project-issue-add-spent-time command line option, [627](#)

gitlab-project-issue-time-estimate command line option, [633](#)

gitlab-project-merge-request-add-spent-time command line option, [661](#)

gitlab-project-merge-request-time-estimate command line option, [667](#)

--ecdsa-key-restriction

gitlab-application-settings-update command line option, [465](#)

--ed25519-key-restriction

gitlab-application-settings-update command line option, [465](#)

---

```

--email
  gitlab-current-user-email-create
    command line option, 470
  gitlab-group-invitation-delete command
    line option, 523
  gitlab-group-invitation-get command
    line option, 523
  gitlab-group-invitation-update command
    line option, 524
  gitlab-project-invitation-delete
    command line option, 625
  gitlab-project-invitation-get command
    line option, 625
  gitlab-project-invitation-update
    command line option, 626
  gitlab-user-create command line option,
    762
  gitlab-user-email-create command line
    option, 769
  gitlab-user-update command line option,
    766
--email-header-and-footer-enabled
  gitlab-application-appearance-update
    command line option, 461
--emails-disabled
  gitlab-group-create command line option,
    484
  gitlab-group-update command line option,
    488
  gitlab-project-create command line
    option, 562
  gitlab-project-update command line
    option, 573
--emoji
  gitlab-current-user-status-update
    command line option, 474
--enable-ssl-verification
  gitlab-group-hook-create command line
    option, 520
  gitlab-group-hook-update command line
    option, 522
  gitlab-project-hook-create command line
    option, 620
  gitlab-project-hook-update command line
    option, 622
--enabled
  gitlab-geo-node-update command line
    option, 480
  gitlab-group-cluster-create command
    line option, 500
  gitlab-project-cluster-create command
    line option, 593
  gitlab-project-remote-mirror-create
    command line option, 723
  gitlab-project-remote-mirror-update
    command line option, 724
--encoding
  gitlab-project-file-create command line
    option, 616
  gitlab-project-file-update command line
    option, 618
--end-date
  gitlab-group-epic-create command line
    option, 506
  gitlab-group-epic-update command line
    option, 508
--ends-at
  gitlab-broadcast-message-create command
    line option, 467
  gitlab-broadcast-message-update command
    line option, 468
--enforce-terms
  gitlab-application-settings-update
    command line option, 465
--entity-id
  gitlab-audit-event-list command line
    option, 466
--entity-type
  gitlab-audit-event-list command line
    option, 466
--environment
  gitlab-merge-request-list command line
    option, 555
  gitlab-project-deployment-create
    command line option, 607
  gitlab-project-deployment-list command
    line option, 608
  gitlab-project-deployment-merge-request-list
    command line option, 610
--environment-scope
  gitlab-group-cluster-create command
    line option, 500
  gitlab-group-cluster-update command
    line option, 502
  gitlab-project-cluster-create command
    line option, 593
  gitlab-project-cluster-update command
    line option, 594
  gitlab-project-variable-create command
    line option, 744
  gitlab-project-variable-update command
    line option, 746
--epic-id
  gitlab-group-epic-resource-label-event-get
    command line option, 517
  gitlab-group-epic-resource-label-event-list
    command line option, 517
--epic-iid

```

- gitlab-group-epic-award-emoji-create
  - command line option, [508](#)
- gitlab-group-epic-award-emoji-delete
  - command line option, [509](#)
- gitlab-group-epic-award-emoji-get
  - command line option, [509](#)
- gitlab-group-epic-award-emoji-list
  - command line option, [509](#)
- gitlab-group-epic-discussion-note-create
  - command line option, [510](#)
- gitlab-group-epic-discussion-note-delete
  - command line option, [510](#)
- gitlab-group-epic-discussion-note-get
  - command line option, [511](#)
- gitlab-group-epic-discussion-note-update
  - command line option, [511](#)
- gitlab-group-epic-issue-create command line option, [512](#)
- gitlab-group-epic-issue-delete command line option, [512](#)
- gitlab-group-epic-issue-list command line option, [512](#)
- gitlab-group-epic-issue-update command line option, [513](#)
- gitlab-group-epic-note-award-emoji-create
  - command line option, [515](#)
- gitlab-group-epic-note-award-emoji-delete
  - command line option, [516](#)
- gitlab-group-epic-note-award-emoji-get
  - command line option, [516](#)
- gitlab-group-epic-note-award-emoji-list
  - command line option, [516](#)
- gitlab-group-epic-note-create command line option, [514](#)
- gitlab-group-epic-note-delete command line option, [514](#)
- gitlab-group-epic-note-get command line option, [514](#)
- gitlab-group-epic-note-list command line option, [514](#)
- gitlab-group-epic-note-update command line option, [515](#)
- epic-issue-id
  - gitlab-group-epic-issue-delete command line option, [512](#)
  - gitlab-group-epic-issue-update command line option, [513](#)
- exclude-subgroups
  - gitlab-group-package-list command line option, [538](#)
- expires-at
  - gitlab-group-access-token-create command line option, [490](#)
  - gitlab-group-deploy-token-create
    - command line option, [503](#)
- gitlab-group-invitation-create command line option, [523](#)
- gitlab-group-invitation-update command line option, [524](#)
- gitlab-group-member-create command line option, [530](#)
- gitlab-group-member-update command line option, [531](#)
- gitlab-group-share command line option, [486](#)
- gitlab-project-access-token-create
  - command line option, [577](#)
- gitlab-project-deploy-token-create
  - command line option, [605](#)
- gitlab-project-invitation-create
  - command line option, [625](#)
- gitlab-project-invitation-update
  - command line option, [626](#)
- gitlab-project-member-create command line option, [657](#)
- gitlab-project-member-update command line option, [659](#)
- gitlab-project-share command line option, [569](#)
- gitlab-user-impersonation-token-create
  - command line option, [772](#)
- gitlab-user-personal-access-token-create
  - command line option, [776](#)
- extern-uid
  - gitlab-user-create command line option, [762](#)
  - gitlab-user-list command line option, [764](#)
  - gitlab-user-update command line option, [766](#)
- external
  - gitlab-user-create command line option, [762](#)
  - gitlab-user-list command line option, [764](#)
  - gitlab-user-update command line option, [766](#)
- external-authorization-classification-label
  - gitlab-project-create command line option, [562](#)
  - gitlab-project-update command line option, [573](#)
- external-authorization-service-default-label
  - gitlab-application-settings-update command line option, [464](#)
- external-authorization-service-enabled
  - gitlab-application-settings-update command line option, [464](#)
- external-authorization-service-timeout
  - gitlab-application-settings-update

command line option, [464](#)  
 --external-authorization-service-url  
     gitlab-application-settings-update  
         command line option, [464](#)  
 --external-url  
     gitlab-project-environment-create  
         command line option, [611](#)  
     gitlab-project-environment-update  
         command line option, [613](#)  
 --extra-shared-runners-minutes-limit  
     gitlab-group-create command line option,  
         [484](#)  
     gitlab-group-update command line option,  
         [488](#)  
 --fancy  
     gitlab command line option, [458](#)  
 --favicon  
     gitlab-application-appearance-update  
         command line option, [461](#)  
 --fields  
     gitlab command line option, [458](#)  
 --file-name  
     gitlab-generic-package-download command  
         line option, [478](#)  
     gitlab-generic-package-upload command  
         line option, [478](#)  
     gitlab-project-snippet-create command  
         line option, [728](#)  
     gitlab-project-snippet-update command  
         line option, [729](#)  
     gitlab-snippet-create command line  
         option, [754](#)  
     gitlab-snippet-update command line  
         option, [756](#)  
 --file-name-regex  
     gitlab-group-push-rules-create command  
         line option, [541](#)  
     gitlab-group-push-rules-update command  
         line option, [542](#)  
     gitlab-project-push-rules-create  
         command line option, [714](#)  
     gitlab-project-push-rules-update  
         command line option, [715](#)  
 --file-path  
     gitlab-project-file-blame command line  
         option, [615](#)  
     gitlab-project-file-create command line  
         option, [616](#)  
     gitlab-project-file-delete command line  
         option, [616](#)  
     gitlab-project-file-get command line  
         option, [617](#)  
     gitlab-project-file-raw command line  
         option, [617](#)  
     gitlab-project-file-update command line  
         option, [617](#)  
 --file-template-project-id  
     gitlab-application-settings-update  
         command line option, [465](#)  
     gitlab-group-update command line option,  
         [488](#)  
 --filename  
     gitlab-project-upload command line  
         option, [575](#)  
 --filepath  
     gitlab-project-release-link-create  
         command line option, [721](#)  
     gitlab-project-release-link-update  
         command line option, [723](#)  
     gitlab-project-upload command line  
         option, [575](#)  
 --files-max-capacity  
     gitlab-geo-node-update command line  
         option, [481](#)  
 --first-day-of-week  
     gitlab-application-settings-update  
         command line option, [465](#)  
 --first-parent  
     gitlab-project-commit-list command line  
         option, [597](#)  
 --font  
     gitlab-broadcast-message-create command  
         line option, [467](#)  
     gitlab-broadcast-message-update command  
         line option, [468](#)  
 --footer-message  
     gitlab-application-appearance-update  
         command line option, [461](#)  
 --forked-from-id  
     gitlab-project-create-fork-relation  
         command line option, [564](#)  
 --forking-access-level  
     gitlab-project-create command line  
         option, [562](#)  
     gitlab-project-update command line  
         option, [573](#)  
 --format  
     gitlab-group-wiki-create command line  
         option, [546](#)  
     gitlab-group-wiki-update command line  
         option, [548](#)  
     gitlab-project-repository-archive  
         command line option, [567](#)  
     gitlab-project-wiki-create command line  
         option, [746](#)  
     gitlab-project-wiki-update command line  
         option, [748](#)  
 --from-

gitlab-project-repository-compare  
command line option, [567](#)

--fullname  
gitlab-license-get command line option,  
[552](#)

--geo-node-allowed-ips  
gitlab-application-settings-update  
command line option, [465](#)

--get-all  
gitlab-application-list command line  
option, [460](#)  
gitlab-audit-event-list command line  
option, [466](#)  
gitlab-broadcast-message-list command  
line option, [468](#)  
gitlab-current-user-email-list command  
line option, [471](#)  
gitlab-current-user-gpg-key-list  
command line option, [472](#)  
gitlab-current-user-key-list command  
line option, [473](#)  
gitlab-deploy-key-list command line  
option, [474](#)  
gitlab-deploy-token-list command line  
option, [475](#)  
gitlab-dockerfile-list command line  
option, [476](#)  
gitlab-event-list command line option,  
[476](#)  
gitlab-feature-list command line option,  
[477](#)  
gitlab-geo-node-list command line  
option, [480](#)  
gitlab-gitignore-list command line  
option, [481](#)  
gitlab-gitlabciyaml-list command line  
option, [482](#)  
gitlab-group-access-request-list  
command line option, [490](#)  
gitlab-group-access-token-list command  
line option, [491](#)  
gitlab-group-audit-event-list command  
line option, [492](#)  
gitlab-group-badge-list command line  
option, [494](#)  
gitlab-group-billable-member-list  
command line option, [495](#)  
gitlab-group-billable-member-membership-list  
command line option, [496](#)  
gitlab-group-board-list command line  
option, [497](#)  
gitlab-group-board-list-list command  
line option, [499](#)  
gitlab-group-cluster-list command line  
option, [501](#)  
gitlab-group-custom-attribute-list  
command line option, [503](#)  
gitlab-group-deploy-token-list command  
line option, [504](#)  
gitlab-group-descendant-group-list  
command line option, [505](#)  
gitlab-group-epic-award-emoji-list  
command line option, [509](#)  
gitlab-group-epic-issue-list command  
line option, [513](#)  
gitlab-group-epic-list command line  
option, [507](#)  
gitlab-group-epic-note-award-emoji-list  
command line option, [517](#)  
gitlab-group-epic-note-list command  
line option, [515](#)  
gitlab-group-epic-resource-label-event-list  
command line option, [518](#)  
gitlab-group-hook-list command line  
option, [520](#)  
gitlab-group-invitation-list command  
line option, [524](#)  
gitlab-group-issue-list command line  
option, [525](#)  
gitlab-group-iteration-list command  
line option, [526](#)  
gitlab-group-label-list command line  
option, [528](#)  
gitlab-group-list command line option,  
[486](#)  
gitlab-group-member-all-list command  
line option, [532](#)  
gitlab-group-member-list command line  
option, [531](#)  
gitlab-group-merge-request-list command  
line option, [534](#)  
gitlab-group-milestone-list command  
line option, [536](#)  
gitlab-group-package-list command line  
option, [539](#)  
gitlab-group-project-list command line  
option, [540](#)  
gitlab-group-runner-list command line  
option, [543](#)  
gitlab-group-subgroup-list command line  
option, [544](#)  
gitlab-group-variable-list command line  
option, [545](#)  
gitlab-group-wiki-list command line  
option, [547](#)  
gitlab-hook-list command line option, [549](#)  
gitlab-issue-list command line option,  
[550](#)



- gitlab-ldap-group-list command line option, [552](#)
- gitlab-license-list command line option, [553](#)
- gitlab-merge-request-list command line option, [555](#)
- gitlab-namespace-list command line option, [556](#)
- gitlab-pages-domain-list command line option, [557](#)
- gitlab-personal-access-token-list command line option, [558](#)
- gitlab-project-access-request-list command line option, [577](#)
- gitlab-project-access-token-list command line option, [578](#)
- gitlab-project-approval-rule-list command line option, [581](#)
- gitlab-project-audit-event-list command line option, [583](#)
- gitlab-project-badge-list command line option, [585](#)
- gitlab-project-board-list command line option, [587](#)
- gitlab-project-board-list-list command line option, [589](#)
- gitlab-project-branch-list command line option, [591](#)
- gitlab-project-cluster-list command line option, [594](#)
- gitlab-project-commit-comment-list command line option, [599](#)
- gitlab-project-commit-discussion-list command line option, [600](#)
- gitlab-project-commit-list command line option, [597](#)
- gitlab-project-commit-status-list command line option, [604](#)
- gitlab-project-custom-attribute-list command line option, [605](#)
- gitlab-project-deploy-token-list command line option, [606](#)
- gitlab-project-deployment-list command line option, [608](#)
- gitlab-project-deployment-merge-request-list command line option, [611](#)
- gitlab-project-environment-list command line option, [612](#)
- gitlab-project-event-list command line option, [614](#)
- gitlab-project-fork-list command line option, [619](#)
- gitlab-project-hook-list command line option, [621](#)
- gitlab-project-integration-list command line option, [624](#)
- gitlab-project-invitation-list command line option, [626](#)
- gitlab-project-issue-award-emoji-list command line option, [636](#)
- gitlab-project-issue-discussion-list command line option, [638](#)
- gitlab-project-issue-link-list command line option, [641](#)
- gitlab-project-issue-list command line option, [630](#)
- gitlab-project-issue-note-award-emoji-list command line option, [645](#)
- gitlab-project-issue-note-list command line option, [642](#)
- gitlab-project-issue-resource-label-event-list command line option, [646](#)
- gitlab-project-issue-resource-milestone-event-list command line option, [647](#)
- gitlab-project-issue-resource-state-event-list command line option, [648](#)
- gitlab-project-job-list command line option, [651](#)
- gitlab-project-key-list command line option, [654](#)
- gitlab-project-label-list command line option, [656](#)
- gitlab-project-list command line option, [566](#)
- gitlab-project-member-all-list command line option, [660](#)
- gitlab-project-member-list command line option, [658](#)
- gitlab-project-merge-request-approval-rule-list command line option, [672](#)
- gitlab-project-merge-request-award-emoji-list command line option, [676](#)
- gitlab-project-merge-request-diff-list command line option, [676](#)
- gitlab-project-merge-request-discussion-list command line option, [678](#)
- gitlab-project-merge-request-list command line option, [665](#)
- gitlab-project-merge-request-note-award-emoji-list command line option, [684](#)
- gitlab-project-merge-request-note-list command line option, [682](#)
- gitlab-project-merge-request-pipeline-list command line option, [685](#)
- gitlab-project-merge-request-resource-label-event-list command line option, [686](#)
- gitlab-project-merge-request-resource-milestone-event-list command line option, [687](#)

`gitlab-project-merge-request-resource-state-event-list` command line option, [688](#)  
`gitlab-project-merge-train-list` command line option, [689](#)  
`gitlab-project-milestone-list` command line option, [691](#)  
`gitlab-project-note-list` command line option, [693](#)  
`gitlab-project-package-file-list` command line option, [696](#)  
`gitlab-project-package-list` command line option, [695](#)  
`gitlab-project-pages-domain-list` command line option, [698](#)  
`gitlab-project-pipeline-bridge-list` command line option, [701](#)  
`gitlab-project-pipeline-job-list` command line option, [702](#)  
`gitlab-project-pipeline-list` command line option, [700](#)  
`gitlab-project-pipeline-schedule-list` command line option, [704](#)  
`gitlab-project-pipeline-variable-list` command line option, [708](#)  
`gitlab-project-protected-branch-list` command line option, [710](#)  
`gitlab-project-protected-environment-list` command line option, [711](#)  
`gitlab-project-protected-tag-list` command line option, [713](#)  
`gitlab-project-registry-repository-list` command line option, [716](#)  
`gitlab-project-registry-tag-list` command line option, [718](#)  
`gitlab-project-release-link-list` command line option, [722](#)  
`gitlab-project-release-list` command line option, [720](#)  
`gitlab-project-remote-mirror-list` command line option, [724](#)  
`gitlab-project-runner-list` command line option, [725](#)  
`gitlab-project-service-list` command line option, [727](#)  
`gitlab-project-snippet-award-emoji-list` command line option, [731](#)  
`gitlab-project-snippet-discussion-list` command line option, [733](#)  
`gitlab-project-snippet-list` command line option, [729](#)  
`gitlab-project-snippet-note-award-emoji-list` command line option, [739](#)  
`gitlab-project-snippet-note-list` command line option, [736](#)  
`gitlab-project-tag-list` command line option, [741](#)  
`gitlab-project-trigger-list` command line option, [742](#)  
`gitlab-project-user-list` command line option, [743](#)  
`gitlab-project-variable-list` command line option, [745](#)  
`gitlab-project-wiki-list` command line option, [747](#)  
`gitlab-runner-all-list` command line option, [751](#)  
`gitlab-runner-job-list` command line option, [752](#)  
`gitlab-runner-list` command line option, [750](#)  
`gitlab-shared-project-list` command line option, [753](#)  
`gitlab-snippet-list` command line option, [755](#)  
`gitlab-starred-project-list` command line option, [757](#)  
`gitlab-todo-list` command line option, [758](#)  
`gitlab-topic-list` command line option, [760](#)  
`gitlab-user-activities-list` command line option, [767](#)  
`gitlab-user-custom-attribute-list` command line option, [768](#)  
`gitlab-user-email-list` command line option, [770](#)  
`gitlab-user-event-list` command line option, [770](#)  
`gitlab-user-gpg-key-list` command line option, [772](#)  
`gitlab-user-impersonation-token-list` command line option, [773](#)  
`gitlab-user-key-list` command line option, [775](#)  
`gitlab-user-list` command line option, [764](#)  
`gitlab-user-membership-list` command line option, [776](#)  
`gitlab-user-project-list` command line option, [779](#)  
`gitlab-variable-list` command line option, [781](#)  
`--gitlab`  
    `gitlab` command line option, [458](#)  
`--gravatar-enabled`  
    `gitlab-application-settings-update` command line option, [464](#)  
`--group-access`  
    `gitlab-group-add-ldap-group-link` command line option, [483](#)



- gitlab-group-share command line option, [486](#)
- gitlab-project-share command line option, [569](#)
- group-id
  - gitlab-group-access-request-approve command line option, [489](#)
  - gitlab-group-access-request-create command line option, [489](#)
  - gitlab-group-access-request-delete command line option, [489](#)
  - gitlab-group-access-request-list command line option, [490](#)
  - gitlab-group-access-token-create command line option, [490](#)
  - gitlab-group-access-token-delete command line option, [491](#)
  - gitlab-group-access-token-list command line option, [491](#)
  - gitlab-group-audit-event-get command line option, [492](#)
  - gitlab-group-audit-event-list command line option, [492](#)
  - gitlab-group-badge-create command line option, [493](#)
  - gitlab-group-badge-delete command line option, [493](#)
  - gitlab-group-badge-get command line option, [493](#)
  - gitlab-group-badge-list command line option, [494](#)
  - gitlab-group-badge-render command line option, [494](#)
  - gitlab-group-badge-update command line option, [494](#)
  - gitlab-group-billable-member-delete command line option, [495](#)
  - gitlab-group-billable-member-list command line option, [495](#)
  - gitlab-group-billable-member-membership-list command line option, [496](#)
  - gitlab-group-board-create command line option, [496](#)
  - gitlab-group-board-delete command line option, [497](#)
  - gitlab-group-board-get command line option, [497](#)
  - gitlab-group-board-list command line option, [497](#)
  - gitlab-group-board-list-create command line option, [498](#)
  - gitlab-group-board-list-delete command line option, [498](#)
  - gitlab-group-board-list-get command line option, [499](#)
  - gitlab-group-board-list-list command line option, [499](#)
  - gitlab-group-board-list-update command line option, [499](#)
  - gitlab-group-board-update command line option, [498](#)
  - gitlab-group-cluster-create command line option, [500](#)
  - gitlab-group-cluster-delete command line option, [500](#)
  - gitlab-group-cluster-get command line option, [501](#)
  - gitlab-group-cluster-list command line option, [501](#)
  - gitlab-group-cluster-update command line option, [501](#)
  - gitlab-group-custom-attribute-delete command line option, [502](#)
  - gitlab-group-custom-attribute-get command line option, [502](#)
  - gitlab-group-custom-attribute-list command line option, [503](#)
  - gitlab-group-deploy-token-create command line option, [503](#)
  - gitlab-group-deploy-token-delete command line option, [504](#)
  - gitlab-group-deploy-token-get command line option, [504](#)
  - gitlab-group-deploy-token-list command line option, [504](#)
  - gitlab-group-descendant-group-list command line option, [505](#)
  - gitlab-group-epic-award-emoji-create command line option, [508](#)
  - gitlab-group-epic-award-emoji-delete command line option, [509](#)
  - gitlab-group-epic-award-emoji-get command line option, [509](#)
  - gitlab-group-epic-award-emoji-list command line option, [509](#)
  - gitlab-group-epic-create command line option, [506](#)
  - gitlab-group-epic-delete command line option, [506](#)
  - gitlab-group-epic-discussion-note-create command line option, [510](#)
  - gitlab-group-epic-discussion-note-delete command line option, [510](#)
  - gitlab-group-epic-discussion-note-get command line option, [511](#)
  - gitlab-group-epic-discussion-note-update command line option, [511](#)
  - gitlab-group-epic-get command line

option, [507](#)

gitlab-group-epic-issue-create command  
line option, [512](#)

gitlab-group-epic-issue-delete command  
line option, [512](#)

gitlab-group-epic-issue-list command  
line option, [512](#)

gitlab-group-epic-issue-update command  
line option, [513](#)

gitlab-group-epic-list command line  
option, [507](#)

gitlab-group-epic-note-award-emoji-create  
command line option, [515](#)

gitlab-group-epic-note-award-emoji-delete  
command line option, [516](#)

gitlab-group-epic-note-award-emoji-get  
command line option, [516](#)

gitlab-group-epic-note-award-emoji-list  
command line option, [516](#)

gitlab-group-epic-note-create command  
line option, [513](#)

gitlab-group-epic-note-delete command  
line option, [514](#)

gitlab-group-epic-note-get command line  
option, [514](#)

gitlab-group-epic-note-list command  
line option, [514](#)

gitlab-group-epic-note-update command  
line option, [515](#)

gitlab-group-epic-resource-label-event-get  
command line option, [517](#)

gitlab-group-epic-resource-label-event-list  
command line option, [517](#)

gitlab-group-epic-update command line  
option, [508](#)

gitlab-group-export-create command line  
option, [518](#)

gitlab-group-export-download command  
line option, [518](#)

gitlab-group-export-get command line  
option, [518](#)

gitlab-group-hook-create command line  
option, [519](#)

gitlab-group-hook-delete command line  
option, [520](#)

gitlab-group-hook-get command line  
option, [520](#)

gitlab-group-hook-list command line  
option, [520](#)

gitlab-group-hook-update command line  
option, [521](#)

gitlab-group-import-get command line  
option, [522](#)

gitlab-group-invitation-create command  
line option, [522](#)

gitlab-group-invitation-delete command  
line option, [523](#)

gitlab-group-invitation-get command  
line option, [523](#)

gitlab-group-invitation-list command  
line option, [523](#)

gitlab-group-invitation-update command  
line option, [524](#)

gitlab-group-issue-list command line  
option, [525](#)

gitlab-group-issues-statistics-get  
command line option, [526](#)

gitlab-group-iteration-list command  
line option, [526](#)

gitlab-group-label-create command line  
option, [527](#)

gitlab-group-label-delete command line  
option, [527](#)

gitlab-group-label-get command line  
option, [528](#)

gitlab-group-label-list command line  
option, [528](#)

gitlab-group-label-subscribe command  
line option, [528](#)

gitlab-group-label-unsubscribe command  
line option, [529](#)

gitlab-group-label-update command line  
option, [529](#)

gitlab-group-member-all-get command  
line option, [532](#)

gitlab-group-member-all-list command  
line option, [532](#)

gitlab-group-member-create command line  
option, [530](#)

gitlab-group-member-delete command line  
option, [530](#)

gitlab-group-member-get command line  
option, [530](#)

gitlab-group-member-list command line  
option, [531](#)

gitlab-group-member-update command line  
option, [531](#)

gitlab-group-merge-request-list command  
line option, [533](#)

gitlab-group-milestone-create command  
line option, [534](#)

gitlab-group-milestone-delete command  
line option, [535](#)

gitlab-group-milestone-get command line  
option, [535](#)

gitlab-group-milestone-issues command  
line option, [535](#)

gitlab-group-milestone-list command

- line option, [536](#)
- gitlab-group-milestone-merge-requests command line option, [536](#)
- gitlab-group-milestone-update command line option, [536](#)
- gitlab-group-notification-settings-get command line option, [537](#)
- gitlab-group-notification-settings-update command line option, [537](#)
- gitlab-group-package-list command line option, [538](#)
- gitlab-group-project-list command line option, [539](#)
- gitlab-group-push-rules-create command line option, [541](#)
- gitlab-group-push-rules-delete command line option, [541](#)
- gitlab-group-push-rules-get command line option, [541](#)
- gitlab-group-push-rules-update command line option, [542](#)
- gitlab-group-runner-list command line option, [543](#)
- gitlab-group-share command line option, [486](#)
- gitlab-group-subgroup-list command line option, [543](#)
- gitlab-group-transfer command line option, [487](#)
- gitlab-group-unshare command line option, [487](#)
- gitlab-group-variable-create command line option, [544](#)
- gitlab-group-variable-delete command line option, [545](#)
- gitlab-group-variable-get command line option, [545](#)
- gitlab-group-variable-list command line option, [545](#)
- gitlab-group-variable-update command line option, [546](#)
- gitlab-group-wiki-create command line option, [546](#)
- gitlab-group-wiki-delete command line option, [547](#)
- gitlab-group-wiki-get command line option, [547](#)
- gitlab-group-wiki-list command line option, [547](#)
- gitlab-group-wiki-update command line option, [548](#)
- gitlab-project-share command line option, [569](#)
- gitlab-project-unshare command line option, [571](#)
- gitlab-shared-project-list command line option, [753](#)
- group-ids
  - gitlab-project-approval-rule-create command line option, [580](#)
  - gitlab-project-merge-request-approval-rule-create command line option, [671](#)
  - gitlab-project-merge-request-approval-rule-update command line option, [673](#)
- group-with-project-templates-id
  - gitlab-project-create command line option, [562](#)
- header-logo
  - gitlab-application-appearance-update command line option, [461](#)
- header-message
  - gitlab-application-appearance-update command line option, [461](#)
- help
  - gitlab command line option, [458](#)
  - gitlab-application command line option, [459](#)
  - gitlab-application-appearance command line option, [460](#)
  - gitlab-application-appearance-get command line option, [461](#)
  - gitlab-application-appearance-update command line option, [461](#)
  - gitlab-application-create command line option, [459](#)
  - gitlab-application-delete command line option, [460](#)
  - gitlab-application-list command line option, [460](#)
  - gitlab-application-settings command line option, [462](#)
  - gitlab-application-settings-get command line option, [462](#)
  - gitlab-application-settings-update command line option, [463](#)
  - gitlab-audit-event command line option, [466](#)
  - gitlab-audit-event-get command line option, [466](#)
  - gitlab-audit-event-list command line option, [466](#)
  - gitlab-broadcast-message command line option, [467](#)
  - gitlab-broadcast-message-create command line option, [467](#)
  - gitlab-broadcast-message-delete command line option, [467](#)
  - gitlab-broadcast-message-get command

- line option, [467](#)
- gitlab-broadcast-message-list command line option, [468](#)
- gitlab-broadcast-message-update command line option, [468](#)
- gitlab-ci-lint command line option, [469](#)
- gitlab-ci-lint-create command line option, [469](#)
- gitlab-ci-lint-validate command line option, [469](#)
- gitlab-current-user command line option, [469](#)
- gitlab-current-user-email command line option, [470](#)
- gitlab-current-user-email-create command line option, [470](#)
- gitlab-current-user-email-delete command line option, [470](#)
- gitlab-current-user-email-get command line option, [470](#)
- gitlab-current-user-email-list command line option, [471](#)
- gitlab-current-user-get command line option, [470](#)
- gitlab-current-user-gpg-key command line option, [471](#)
- gitlab-current-user-gpg-key-create command line option, [471](#)
- gitlab-current-user-gpg-key-delete command line option, [471](#)
- gitlab-current-user-gpg-key-get command line option, [472](#)
- gitlab-current-user-gpg-key-list command line option, [472](#)
- gitlab-current-user-key command line option, [472](#)
- gitlab-current-user-key-create command line option, [472](#)
- gitlab-current-user-key-delete command line option, [473](#)
- gitlab-current-user-key-get command line option, [473](#)
- gitlab-current-user-key-list command line option, [473](#)
- gitlab-current-user-status command line option, [473](#)
- gitlab-current-user-status-get command line option, [474](#)
- gitlab-current-user-status-update command line option, [474](#)
- gitlab-deploy-key command line option, [474](#)
- gitlab-deploy-key-list command line option, [474](#)
- gitlab-deploy-token command line option, [475](#)
- gitlab-deploy-token-list command line option, [475](#)
- gitlab-dockerfile command line option, [475](#)
- gitlab-dockerfile-get command line option, [475](#)
- gitlab-dockerfile-list command line option, [476](#)
- gitlab-event command line option, [476](#)
- gitlab-event-list command line option, [476](#)
- gitlab-feature command line option, [477](#)
- gitlab-feature-delete command line option, [477](#)
- gitlab-feature-list command line option, [477](#)
- gitlab-generic-package command line option, [478](#)
- gitlab-generic-package-download command line option, [478](#)
- gitlab-generic-package-upload command line option, [478](#)
- gitlab-geo-node command line option, [479](#)
- gitlab-geo-node-current-failures command line option, [479](#)
- gitlab-geo-node-delete command line option, [479](#)
- gitlab-geo-node-get command line option, [479](#)
- gitlab-geo-node-list command line option, [480](#)
- gitlab-geo-node-repair command line option, [480](#)
- gitlab-geo-node-status command line option, [480](#)
- gitlab-geo-node-update command line option, [480](#)
- gitlab-gitignore command line option, [481](#)
- gitlab-gitignore-get command line option, [481](#)
- gitlab-gitignore-list command line option, [481](#)
- gitlab-gitlabciyaml command line option, [482](#)
- gitlab-gitlabciyaml-get command line option, [482](#)
- gitlab-gitlabciyaml-list command line option, [482](#)
- gitlab-group command line option, [483](#)
- gitlab-group-access-request command line option, [489](#)
- gitlab-group-access-request-approve

- command line option, [489](#)
- gitlab-group-access-request-create
  - command line option, [489](#)
- gitlab-group-access-request-delete
  - command line option, [489](#)
- gitlab-group-access-request-list
  - command line option, [490](#)
- gitlab-group-access-token command line option, [490](#)
- gitlab-group-access-token-create
  - command line option, [490](#)
- gitlab-group-access-token-delete
  - command line option, [491](#)
- gitlab-group-access-token-list command line option, [491](#)
- gitlab-group-add-ldap-group-link
  - command line option, [483](#)
- gitlab-group-audit-event command line option, [491](#)
- gitlab-group-audit-event-get command line option, [492](#)
- gitlab-group-audit-event-list command line option, [492](#)
- gitlab-group-badge command line option, [492](#)
- gitlab-group-badge-create command line option, [493](#)
- gitlab-group-badge-delete command line option, [493](#)
- gitlab-group-badge-get command line option, [493](#)
- gitlab-group-badge-list command line option, [493](#)
- gitlab-group-badge-render command line option, [494](#)
- gitlab-group-badge-update command line option, [494](#)
- gitlab-group-billable-member command line option, [495](#)
- gitlab-group-billable-member-delete
  - command line option, [495](#)
- gitlab-group-billable-member-list
  - command line option, [495](#)
- gitlab-group-billable-member-membership
  - command line option, [496](#)
- gitlab-group-billable-member-membership-list
  - command line option, [496](#)
- gitlab-group-board command line option, [496](#)
- gitlab-group-board-create command line option, [496](#)
- gitlab-group-board-delete command line option, [497](#)
- gitlab-group-board-get command line option, [497](#)
- gitlab-group-board-list command line option, [497](#), [498](#)
- gitlab-group-board-list-create command line option, [498](#)
- gitlab-group-board-list-delete command line option, [498](#)
- gitlab-group-board-list-get command line option, [499](#)
- gitlab-group-board-list-list command line option, [499](#)
- gitlab-group-board-list-update command line option, [499](#)
- gitlab-group-board-update command line option, [498](#)
- gitlab-group-cluster command line option, [500](#)
- gitlab-group-cluster-create command line option, [500](#)
- gitlab-group-cluster-delete command line option, [500](#)
- gitlab-group-cluster-get command line option, [501](#)
- gitlab-group-cluster-list command line option, [501](#)
- gitlab-group-cluster-update command line option, [501](#)
- gitlab-group-create command line option, [483](#)
- gitlab-group-custom-attribute command line option, [502](#)
- gitlab-group-custom-attribute-delete
  - command line option, [502](#)
- gitlab-group-custom-attribute-get
  - command line option, [502](#)
- gitlab-group-custom-attribute-list
  - command line option, [503](#)
- gitlab-group-delete command line option, [484](#)
- gitlab-group-delete-ldap-group-link
  - command line option, [485](#)
- gitlab-group-deploy-token command line option, [503](#)
- gitlab-group-deploy-token-create
  - command line option, [503](#)
- gitlab-group-deploy-token-delete
  - command line option, [504](#)
- gitlab-group-deploy-token-get command line option, [504](#)
- gitlab-group-deploy-token-list command line option, [504](#)
- gitlab-group-descendant-group command line option, [505](#)
- gitlab-group-descendant-group-list

- command line option, [505](#)
- gitlab-group-epic command line option, [506](#)
- gitlab-group-epic-award-emoji command line option, [508](#)
- gitlab-group-epic-award-emoji-create command line option, [508](#)
- gitlab-group-epic-award-emoji-delete command line option, [509](#)
- gitlab-group-epic-award-emoji-get command line option, [509](#)
- gitlab-group-epic-award-emoji-list command line option, [509](#)
- gitlab-group-epic-create command line option, [506](#)
- gitlab-group-epic-delete command line option, [506](#)
- gitlab-group-epic-discussion-note command line option, [510](#)
- gitlab-group-epic-discussion-note-create command line option, [510](#)
- gitlab-group-epic-discussion-note-delete command line option, [510](#)
- gitlab-group-epic-discussion-note-get command line option, [511](#)
- gitlab-group-epic-discussion-note-update command line option, [511](#)
- gitlab-group-epic-get command line option, [507](#)
- gitlab-group-epic-issue command line option, [512](#)
- gitlab-group-epic-issue-create command line option, [512](#)
- gitlab-group-epic-issue-delete command line option, [512](#)
- gitlab-group-epic-issue-list command line option, [512](#)
- gitlab-group-epic-issue-update command line option, [513](#)
- gitlab-group-epic-list command line option, [507](#)
- gitlab-group-epic-note command line option, [513](#)
- gitlab-group-epic-note-award-emoji command line option, [515](#)
- gitlab-group-epic-note-award-emoji-create command line option, [515](#)
- gitlab-group-epic-note-award-emoji-delete command line option, [516](#)
- gitlab-group-epic-note-award-emoji-get command line option, [516](#)
- gitlab-group-epic-note-award-emoji-list command line option, [516](#)
- gitlab-group-epic-note-create command line option, [513](#)
- gitlab-group-epic-note-delete command line option, [514](#)
- gitlab-group-epic-note-get command line option, [514](#)
- gitlab-group-epic-note-list command line option, [514](#)
- gitlab-group-epic-note-update command line option, [515](#)
- gitlab-group-epic-resource-label-event command line option, [517](#)
- gitlab-group-epic-resource-label-event-get command line option, [517](#)
- gitlab-group-epic-resource-label-event-list command line option, [517](#)
- gitlab-group-epic-update command line option, [508](#)
- gitlab-group-export command line option, [518](#)
- gitlab-group-export-create command line option, [518](#)
- gitlab-group-export-download command line option, [518](#)
- gitlab-group-export-get command line option, [518](#)
- gitlab-group-get command line option, [485](#)
- gitlab-group-hook command line option, [519](#)
- gitlab-group-hook-create command line option, [519](#)
- gitlab-group-hook-delete command line option, [520](#)
- gitlab-group-hook-get command line option, [520](#)
- gitlab-group-hook-list command line option, [520](#)
- gitlab-group-hook-update command line option, [521](#)
- gitlab-group-import command line option, [522](#)
- gitlab-group-import-get command line option, [522](#)
- gitlab-group-invitation command line option, [522](#)
- gitlab-group-invitation-create command line option, [522](#)
- gitlab-group-invitation-delete command line option, [523](#)
- gitlab-group-invitation-get command line option, [523](#)
- gitlab-group-invitation-list command line option, [523](#)
- gitlab-group-invitation-update command line option, [524](#)

- gitlab-group-issue command line option, [524](#)
- gitlab-group-issue-list command line option, [524](#)
- gitlab-group-issues-statistics command line option, [525](#)
- gitlab-group-issues-statistics-get command line option, [526](#)
- gitlab-group-iteration command line option, [526](#)
- gitlab-group-iteration-list command line option, [526](#)
- gitlab-group-label command line option, [527](#)
- gitlab-group-label-create command line option, [527](#)
- gitlab-group-label-delete command line option, [527](#)
- gitlab-group-label-get command line option, [528](#)
- gitlab-group-label-list command line option, [528](#)
- gitlab-group-label-subscribe command line option, [528](#)
- gitlab-group-label-unsubscribe command line option, [529](#)
- gitlab-group-label-update command line option, [529](#)
- gitlab-group-ldap-sync command line option, [485](#)
- gitlab-group-list command line option, [485](#)
- gitlab-group-member command line option, [529](#)
- gitlab-group-member-all command line option, [531](#)
- gitlab-group-member-all-get command line option, [532](#)
- gitlab-group-member-all-list command line option, [532](#)
- gitlab-group-member-create command line option, [530](#)
- gitlab-group-member-delete command line option, [530](#)
- gitlab-group-member-get command line option, [530](#)
- gitlab-group-member-list command line option, [531](#)
- gitlab-group-member-update command line option, [531](#)
- gitlab-group-merge-request command line option, [532](#)
- gitlab-group-merge-request-list command line option, [533](#)
- gitlab-group-milestone command line option, [534](#)
- gitlab-group-milestone-create command line option, [534](#)
- gitlab-group-milestone-delete command line option, [535](#)
- gitlab-group-milestone-get command line option, [535](#)
- gitlab-group-milestone-issues command line option, [535](#)
- gitlab-group-milestone-list command line option, [535](#)
- gitlab-group-milestone-merge-requests command line option, [536](#)
- gitlab-group-milestone-update command line option, [536](#)
- gitlab-group-notification-settings command line option, [537](#)
- gitlab-group-notification-settings-get command line option, [537](#)
- gitlab-group-notification-settings-update command line option, [537](#)
- gitlab-group-package command line option, [538](#)
- gitlab-group-package-list command line option, [538](#)
- gitlab-group-project command line option, [539](#)
- gitlab-group-project-list command line option, [539](#)
- gitlab-group-push-rules command line option, [540](#)
- gitlab-group-push-rules-create command line option, [540](#)
- gitlab-group-push-rules-delete command line option, [541](#)
- gitlab-group-push-rules-get command line option, [541](#)
- gitlab-group-push-rules-update command line option, [542](#)
- gitlab-group-runner command line option, [542](#)
- gitlab-group-runner-list command line option, [543](#)
- gitlab-group-search command line option, [486](#)
- gitlab-group-share command line option, [486](#)
- gitlab-group-subgroup command line option, [543](#)
- gitlab-group-subgroup-list command line option, [543](#)
- gitlab-group-transfer command line option, [487](#)



gitlab-group-transfer-project command line option, [487](#)  
gitlab-group-unshare command line option, [487](#)  
gitlab-group-update command line option, [488](#)  
gitlab-group-variable command line option, [544](#)  
gitlab-group-variable-create command line option, [544](#)  
gitlab-group-variable-delete command line option, [545](#)  
gitlab-group-variable-get command line option, [545](#)  
gitlab-group-variable-list command line option, [545](#)  
gitlab-group-variable-update command line option, [546](#)  
gitlab-group-wiki command line option, [546](#)  
gitlab-group-wiki-create command line option, [546](#)  
gitlab-group-wiki-delete command line option, [547](#)  
gitlab-group-wiki-get command line option, [547](#)  
gitlab-group-wiki-list command line option, [547](#)  
gitlab-group-wiki-update command line option, [548](#)  
gitlab-hook command line option, [548](#)  
gitlab-hook-create command line option, [548](#)  
gitlab-hook-delete command line option, [548](#)  
gitlab-hook-get command line option, [549](#)  
gitlab-hook-list command line option, [549](#)  
gitlab-issue command line option, [549](#)  
gitlab-issue-get command line option, [549](#)  
gitlab-issue-list command line option, [550](#)  
gitlab-issues-statistics command line option, [551](#)  
gitlab-issues-statistics-get command line option, [551](#)  
gitlab-key command line option, [551](#)  
gitlab-key-get command line option, [551](#)  
gitlab-ldap-group command line option, [551](#)  
gitlab-ldap-group-list command line option, [552](#)  
gitlab-license command line option, [552](#)  
gitlab-license-get command line option, [552](#)  
gitlab-license-list command line option, [553](#)  
gitlab-merge-request command line option, [553](#)  
gitlab-merge-request-list command line option, [554](#)  
gitlab-namespace command line option, [555](#)  
gitlab-namespace-exists command line option, [555](#)  
gitlab-namespace-get command line option, [555](#)  
gitlab-namespace-list command line option, [556](#)  
gitlab-notification-settings command line option, [556](#)  
gitlab-notification-settings-get command line option, [556](#)  
gitlab-notification-settings-update command line option, [557](#)  
gitlab-pages-domain command line option, [557](#)  
gitlab-pages-domain-list command line option, [557](#)  
gitlab-personal-access-token command line option, [558](#)  
gitlab-personal-access-token-delete command line option, [558](#)  
gitlab-personal-access-token-list command line option, [558](#)  
gitlab-project command line option, [559](#)  
gitlab-project-access-request command line option, [576](#)  
gitlab-project-access-request-approve command line option, [576](#)  
gitlab-project-access-request-create command line option, [576](#)  
gitlab-project-access-request-delete command line option, [576](#)  
gitlab-project-access-request-list command line option, [577](#)  
gitlab-project-access-token command line option, [577](#)  
gitlab-project-access-token-create command line option, [577](#)  
gitlab-project-access-token-delete command line option, [578](#)  
gitlab-project-access-token-list command line option, [578](#)  
gitlab-project-additional-statistics command line option, [578](#)  
gitlab-project-additional-statistics-get command line option, [579](#)  
gitlab-project-approval command line option, [579](#)



- gitlab-project-approval-get command line option, [579](#)
- gitlab-project-approval-rule command line option, [580](#)
- gitlab-project-approval-rule-create command line option, [580](#)
- gitlab-project-approval-rule-delete command line option, [580](#)
- gitlab-project-approval-rule-list command line option, [581](#)
- gitlab-project-approval-rule-update command line option, [581](#)
- gitlab-project-approval-update command line option, [579](#)
- gitlab-project-archive command line option, [559](#)
- gitlab-project-artifact command line option, [560](#), [581](#)
- gitlab-project-artifact-delete command line option, [582](#)
- gitlab-project-artifact-download command line option, [582](#)
- gitlab-project-artifact-raw command line option, [582](#)
- gitlab-project-artifacts command line option, [560](#)
- gitlab-project-audit-event command line option, [583](#)
- gitlab-project-audit-event-get command line option, [583](#)
- gitlab-project-audit-event-list command line option, [583](#)
- gitlab-project-badge command line option, [584](#)
- gitlab-project-badge-create command line option, [584](#)
- gitlab-project-badge-delete command line option, [584](#)
- gitlab-project-badge-get command line option, [584](#)
- gitlab-project-badge-list command line option, [585](#)
- gitlab-project-badge-render command line option, [585](#)
- gitlab-project-badge-update command line option, [585](#)
- gitlab-project-board command line option, [586](#)
- gitlab-project-board-create command line option, [586](#)
- gitlab-project-board-delete command line option, [586](#)
- gitlab-project-board-get command line option, [586](#)
- gitlab-project-board-list command line option, [587](#)
- gitlab-project-board-list-create command line option, [588](#)
- gitlab-project-board-list-delete command line option, [588](#)
- gitlab-project-board-list-get command line option, [588](#)
- gitlab-project-board-list-list command line option, [589](#)
- gitlab-project-board-list-update command line option, [589](#)
- gitlab-project-board-update command line option, [587](#)
- gitlab-project-branch command line option, [589](#)
- gitlab-project-branch-create command line option, [590](#)
- gitlab-project-branch-delete command line option, [590](#)
- gitlab-project-branch-get command line option, [590](#)
- gitlab-project-branch-list command line option, [591](#)
- gitlab-project-ci-lint command line option, [591](#)
- gitlab-project-ci-lint-create command line option, [591](#)
- gitlab-project-ci-lint-get command line option, [592](#)
- gitlab-project-ci-lint-validate command line option, [592](#)
- gitlab-project-cluster command line option, [592](#)
- gitlab-project-cluster-create command line option, [592](#)
- gitlab-project-cluster-delete command line option, [593](#)
- gitlab-project-cluster-get command line option, [593](#)
- gitlab-project-cluster-list command line option, [594](#)
- gitlab-project-cluster-update command line option, [594](#)
- gitlab-project-commit command line option, [595](#)
- gitlab-project-commit-cherry-pick command line option, [595](#)
- gitlab-project-commit-comment command line option, [598](#)
- gitlab-project-commit-comment-create command line option, [598](#)
- gitlab-project-commit-comment-list command line option, [599](#)

`gitlab-project-commit-create` command line option, [595](#)  
`gitlab-project-commit-diff` command line option, [596](#)  
`gitlab-project-commit-discussion` command line option, [599](#)  
`gitlab-project-commit-discussion-create` command line option, [599](#)  
`gitlab-project-commit-discussion-get` command line option, [600](#)  
`gitlab-project-commit-discussion-list` command line option, [600](#)  
`gitlab-project-commit-discussion-note` command line option, [601](#)  
`gitlab-project-commit-discussion-note-create` command line option, [601](#)  
`gitlab-project-commit-discussion-note-delete` command line option, [601](#)  
`gitlab-project-commit-discussion-note-get` command line option, [602](#)  
`gitlab-project-commit-discussion-note-update` command line option, [602](#)  
`gitlab-project-commit-get` command line option, [596](#)  
`gitlab-project-commit-list` command line option, [596](#)  
`gitlab-project-commit-merge-requests` command line option, [597](#)  
`gitlab-project-commit-refs` command line option, [597](#)  
`gitlab-project-commit-revert` command line option, [598](#)  
`gitlab-project-commit-signature` command line option, [598](#)  
`gitlab-project-commit-status` command line option, [603](#)  
`gitlab-project-commit-status-create` command line option, [603](#)  
`gitlab-project-commit-status-list` command line option, [603](#)  
`gitlab-project-create` command line option, [561](#)  
`gitlab-project-create-fork-relation` command line option, [564](#)  
`gitlab-project-custom-attribute` command line option, [604](#)  
`gitlab-project-custom-attribute-delete` command line option, [604](#)  
`gitlab-project-custom-attribute-get` command line option, [604](#)  
`gitlab-project-custom-attribute-list` command line option, [605](#)  
`gitlab-project-delete` command line option, [564](#)  
`gitlab-project-delete-fork-relation` command line option, [564](#)  
`gitlab-project-delete-merged-branches` command line option, [564](#)  
`gitlab-project-deploy-token` command line option, [605](#)  
`gitlab-project-deploy-token-create` command line option, [605](#)  
`gitlab-project-deploy-token-delete` command line option, [606](#)  
`gitlab-project-deploy-token-get` command line option, [606](#)  
`gitlab-project-deploy-token-list` command line option, [606](#)  
`gitlab-project-deployment` command line option, [607](#)  
`gitlab-project-deployment-create` command line option, [607](#)  
`gitlab-project-deployment-get` command line option, [607](#)  
`gitlab-project-deployment-list` command line option, [608](#)  
`gitlab-project-deployment-merge-request` command line option, [609](#)  
`gitlab-project-deployment-merge-request-list` command line option, [609](#)  
`gitlab-project-deployment-update` command line option, [608](#)  
`gitlab-project-environment` command line option, [611](#)  
`gitlab-project-environment-create` command line option, [611](#)  
`gitlab-project-environment-delete` command line option, [612](#)  
`gitlab-project-environment-get` command line option, [612](#)  
`gitlab-project-environment-list` command line option, [612](#)  
`gitlab-project-environment-stop` command line option, [613](#)  
`gitlab-project-environment-update` command line option, [613](#)  
`gitlab-project-event` command line option, [613](#)  
`gitlab-project-event-list` command line option, [614](#)  
`gitlab-project-export` command line option, [614](#)  
`gitlab-project-export-create` command line option, [614](#)  
`gitlab-project-export-download` command line option, [615](#)  
`gitlab-project-export-get` command line option, [615](#)

- gitlab-project-file command line option, [615](#)
- gitlab-project-file-blame command line option, [615](#)
- gitlab-project-file-create command line option, [616](#)
- gitlab-project-file-delete command line option, [616](#)
- gitlab-project-file-get command line option, [617](#)
- gitlab-project-file-raw command line option, [617](#)
- gitlab-project-file-update command line option, [617](#)
- gitlab-project-fork command line option, [618](#)
- gitlab-project-fork-create command line option, [618](#)
- gitlab-project-fork-list command line option, [618](#)
- gitlab-project-get command line option, [565](#)
- gitlab-project-hook command line option, [619](#)
- gitlab-project-hook-create command line option, [620](#)
- gitlab-project-hook-delete command line option, [620](#)
- gitlab-project-hook-get command line option, [621](#)
- gitlab-project-hook-list command line option, [621](#)
- gitlab-project-hook-update command line option, [621](#)
- gitlab-project-housekeeping command line option, [565](#)
- gitlab-project-import command line option, [622](#)
- gitlab-project-import-get command line option, [622](#)
- gitlab-project-integration command line option, [623](#)
- gitlab-project-integration-available command line option, [623](#)
- gitlab-project-integration-delete command line option, [623](#)
- gitlab-project-integration-get command line option, [623](#)
- gitlab-project-integration-list command line option, [624](#)
- gitlab-project-integration-update command line option, [624](#)
- gitlab-project-invitation command line option, [624](#)
- gitlab-project-invitation-create command line option, [625](#)
- gitlab-project-invitation-delete command line option, [625](#)
- gitlab-project-invitation-get command line option, [625](#)
- gitlab-project-invitation-list command line option, [626](#)
- gitlab-project-invitation-update command line option, [626](#)
- gitlab-project-issue command line option, [627](#)
- gitlab-project-issue-add-spent-time command line option, [627](#)
- gitlab-project-issue-award-emoji command line option, [635](#)
- gitlab-project-issue-award-emoji-create command line option, [635](#)
- gitlab-project-issue-award-emoji-delete command line option, [635](#)
- gitlab-project-issue-award-emoji-get command line option, [636](#)
- gitlab-project-issue-award-emoji-list command line option, [636](#)
- gitlab-project-issue-closed-by command line option, [628](#)
- gitlab-project-issue-create command line option, [628](#)
- gitlab-project-issue-delete command line option, [629](#)
- gitlab-project-issue-discussion command line option, [636](#)
- gitlab-project-issue-discussion-create command line option, [637](#)
- gitlab-project-issue-discussion-get command line option, [637](#)
- gitlab-project-issue-discussion-list command line option, [637](#)
- gitlab-project-issue-discussion-note command line option, [638](#)
- gitlab-project-issue-discussion-note-create command line option, [638](#)
- gitlab-project-issue-discussion-note-delete command line option, [639](#)
- gitlab-project-issue-discussion-note-get command line option, [639](#)
- gitlab-project-issue-discussion-note-update command line option, [639](#)
- gitlab-project-issue-get command line option, [629](#)
- gitlab-project-issue-link command line option, [640](#)
- gitlab-project-issue-link-create command line option, [640](#)

`gitlab-project-issue-link-delete`  
command line option, [640](#)

`gitlab-project-issue-link-list` command  
line option, [641](#)

`gitlab-project-issue-list` command line  
option, [629](#)

`gitlab-project-issue-move` command line  
option, [630](#)

`gitlab-project-issue-note` command line  
option, [641](#)

`gitlab-project-issue-note-award-emoji`  
command line option, [643](#)

`gitlab-project-issue-note-award-emoji-create`  
command line option, [643](#)

`gitlab-project-issue-note-award-emoji-delete`  
command line option, [644](#)

`gitlab-project-issue-note-award-emoji-get`  
command line option, [644](#)

`gitlab-project-issue-note-award-emoji-list`  
command line option, [644](#)

`gitlab-project-issue-note-create`  
command line option, [641](#)

`gitlab-project-issue-note-delete`  
command line option, [642](#)

`gitlab-project-issue-note-get` command  
line option, [642](#)

`gitlab-project-issue-note-list` command  
line option, [642](#)

`gitlab-project-issue-note-update`  
command line option, [643](#)

`gitlab-project-issue-participants`  
command line option, [631](#)

`gitlab-project-issue-related-merge-requests`  
command line option, [631](#)

`gitlab-project-issue-reorder` command  
line option, [631](#)

`gitlab-project-issue-reset-spent-time`  
command line option, [632](#)

`gitlab-project-issue-reset-time-estimate`  
command line option, [632](#)

`gitlab-project-issue-resource-label-event`  
command line option, [645](#)

`gitlab-project-issue-resource-label-event-get`  
command line option, [645](#)

`gitlab-project-issue-resource-label-event-list`  
command line option, [645](#)

`gitlab-project-issue-resource-milestone-event`  
command line option, [646](#)

`gitlab-project-issue-resource-milestone-event-get`  
command line option, [646](#)

`gitlab-project-issue-resource-milestone-event-list`  
command line option, [647](#)

`gitlab-project-issue-resource-state-event`  
command line option, [647](#)

`gitlab-project-issue-resource-state-event-get`  
command line option, [647](#)

`gitlab-project-issue-resource-state-event-list`  
command line option, [648](#)

`gitlab-project-issue-subscribe` command  
line option, [632](#)

`gitlab-project-issue-time-estimate`  
command line option, [632](#)

`gitlab-project-issue-time-stats` command  
line option, [633](#)

`gitlab-project-issue-todo` command line  
option, [633](#)

`gitlab-project-issue-unsubscribe`  
command line option, [633](#)

`gitlab-project-issue-update` command  
line option, [634](#)

`gitlab-project-issue-user-agent-detail`  
command line option, [634](#)

`gitlab-project-issues-statistics`  
command line option, [648](#)

`gitlab-project-issues-statistics-get`  
command line option, [648](#)

`gitlab-project-job` command line option,  
[649](#)

`gitlab-project-job-artifact` command  
line option, [649](#)

`gitlab-project-job-artifacts` command  
line option, [649](#)

`gitlab-project-job-cancel` command line  
option, [650](#)

`gitlab-project-job-delete-artifacts`  
command line option, [650](#)

`gitlab-project-job-erase` command line  
option, [650](#)

`gitlab-project-job-get` command line  
option, [650](#)

`gitlab-project-job-keep-artifacts`  
command line option, [651](#)

`gitlab-project-job-list` command line  
option, [651](#)

`gitlab-project-job-play` command line  
option, [651](#)

`gitlab-project-job-retry` command line  
option, [652](#)

`gitlab-project-job-trace` command line  
option, [652](#)

`gitlab-project-key` command line option,  
[652](#)

`gitlab-project-key-create` command line  
option, [652](#)

`gitlab-project-key-delete` command line  
option, [653](#)

`gitlab-project-key-enable` command line  
option, [653](#)

- gitlab-project-key-get command line option, [653](#)
- gitlab-project-key-list command line option, [654](#)
- gitlab-project-key-update command line option, [654](#)
- gitlab-project-label command line option, [654](#)
- gitlab-project-label-create command line option, [655](#)
- gitlab-project-label-delete command line option, [655](#)
- gitlab-project-label-get command line option, [655](#)
- gitlab-project-label-list command line option, [656](#)
- gitlab-project-label-subscribe command line option, [656](#)
- gitlab-project-label-unsubscribe command line option, [656](#)
- gitlab-project-label-update command line option, [657](#)
- gitlab-project-languages command line option, [655](#)
- gitlab-project-list command line option, [655](#)
- gitlab-project-member command line option, [657](#)
- gitlab-project-member-all command line option, [659](#)
- gitlab-project-member-all-get command line option, [659](#)
- gitlab-project-member-all-list command line option, [659](#)
- gitlab-project-member-create command line option, [657](#)
- gitlab-project-member-delete command line option, [658](#)
- gitlab-project-member-get command line option, [658](#)
- gitlab-project-member-list command line option, [658](#)
- gitlab-project-member-update command line option, [659](#)
- gitlab-project-merge-request command line option, [660](#)
- gitlab-project-merge-request-add-spent-time command line option, [660](#)
- gitlab-project-merge-request-approval command line option, [670](#)
- gitlab-project-merge-request-approval-get command line option, [670](#)
- gitlab-project-merge-request-approval-rule command line option, [670](#)
- gitlab-project-merge-request-approval-rule-create command line option, [671](#)
- gitlab-project-merge-request-approval-rule-delete command line option, [671](#)
- gitlab-project-merge-request-approval-rule-get command line option, [672](#)
- gitlab-project-merge-request-approval-rule-list command line option, [672](#)
- gitlab-project-merge-request-approval-rule-update command line option, [673](#)
- gitlab-project-merge-request-approval-state command line option, [673](#)
- gitlab-project-merge-request-approval-state-get command line option, [674](#)
- gitlab-project-merge-request-approval-update command line option, [670](#)
- gitlab-project-merge-request-approve command line option, [661](#)
- gitlab-project-merge-request-award-emoji command line option, [674](#)
- gitlab-project-merge-request-award-emoji-create command line option, [674](#)
- gitlab-project-merge-request-award-emoji-delete command line option, [675](#)
- gitlab-project-merge-request-award-emoji-get command line option, [675](#)
- gitlab-project-merge-request-award-emoji-list command line option, [675](#)
- gitlab-project-merge-request-cancel-merge-when-pipeline command line option, [661](#)
- gitlab-project-merge-request-changes command line option, [661](#)
- gitlab-project-merge-request-closes-issues command line option, [662](#)
- gitlab-project-merge-request-commits command line option, [662](#)
- gitlab-project-merge-request-create command line option, [662](#)
- gitlab-project-merge-request-delete command line option, [663](#)
- gitlab-project-merge-request-diff command line option, [676](#)
- gitlab-project-merge-request-diff-get command line option, [676](#)
- gitlab-project-merge-request-diff-list command line option, [676](#)
- gitlab-project-merge-request-discussion command line option, [677](#)
- gitlab-project-merge-request-discussion-create command line option, [677](#)
- gitlab-project-merge-request-discussion-get command line option, [677](#)
- gitlab-project-merge-request-discussion-list command line option, [678](#)

gitlab-project-merge-request-discussion-note command line option, 679	gitlab-project-merge-request-reset-time-estimate command line option, 667
gitlab-project-merge-request-discussion-note-delete command line option, 679	gitlab-project-merge-request-resource-label-event command line option, 686
gitlab-project-merge-request-discussion-note-delete command line option, 679	gitlab-project-merge-request-resource-label-event-get command line option, 686
gitlab-project-merge-request-discussion-note-get command line option, 680	gitlab-project-merge-request-resource-label-event-list command line option, 686
gitlab-project-merge-request-discussion-note-update command line option, 680	gitlab-project-merge-request-resource-milestone-event command line option, 687
gitlab-project-merge-request-discussion-update command line option, 678	gitlab-project-merge-request-resource-milestone-event- command line option, 687
gitlab-project-merge-request-get command line option, 663	gitlab-project-merge-request-resource-milestone-event- command line option, 687
gitlab-project-merge-request-list command line option, 664	gitlab-project-merge-request-resource-state-event command line option, 688
gitlab-project-merge-request-merge command line option, 665	gitlab-project-merge-request-resource-state-event-get command line option, 688
gitlab-project-merge-request-merge-ref command line option, 666	gitlab-project-merge-request-resource-state-event-list command line option, 688
gitlab-project-merge-request-note command line option, 681	gitlab-project-merge-request-subscribe command line option, 667
gitlab-project-merge-request-note-award-emoji command line option, 683	gitlab-project-merge-request-time-estimate command line option, 667
gitlab-project-merge-request-note-award-emoji-create command line option, 683	gitlab-project-merge-request-time-stats command line option, 668
gitlab-project-merge-request-note-award-emoji-delete command line option, 683	gitlab-project-merge-request-todo command line option, 668
gitlab-project-merge-request-note-award-emoji-get command line option, 684	gitlab-project-merge-request-unapprove command line option, 668
gitlab-project-merge-request-note-award-emoji-list command line option, 684	gitlab-project-merge-request-unsubscribe command line option, 668
gitlab-project-merge-request-note-create command line option, 681	gitlab-project-merge-request-update command line option, 669
gitlab-project-merge-request-note-delete command line option, 681	gitlab-project-merge-train command line option, 689
gitlab-project-merge-request-note-get command line option, 682	gitlab-project-merge-train-list command line option, 689
gitlab-project-merge-request-note-list command line option, 682	gitlab-project-milestone command line option, 689
gitlab-project-merge-request-note-update command line option, 682	gitlab-project-milestone-create command line option, 690
gitlab-project-merge-request-participants command line option, 666	gitlab-project-milestone-delete command line option, 690
gitlab-project-merge-request-pipeline command line option, 685	gitlab-project-milestone-get command line option, 690
gitlab-project-merge-request-pipeline-create command line option, 685	gitlab-project-milestone-issues command line option, 691
gitlab-project-merge-request-pipeline-list command line option, 685	gitlab-project-milestone-list command line option, 691
gitlab-project-merge-request-rebase command line option, 666	gitlab-project-milestone-merge-requests command line option, 691
gitlab-project-merge-request-reset-spent-time command line option, 666	gitlab-project-milestone-update command line option, 692



- gitlab-project-mirror-pull command line option, [567](#)
- gitlab-project-note command line option, [692](#)
- gitlab-project-note-get command line option, [692](#)
- gitlab-project-note-list command line option, [693](#)
- gitlab-project-notification-settings command line option, [693](#)
- gitlab-project-notification-settings-get command line option, [693](#)
- gitlab-project-notification-settings-update command line option, [694](#)
- gitlab-project-package command line option, [694](#)
- gitlab-project-package-delete command line option, [695](#)
- gitlab-project-package-file command line option, [696](#)
- gitlab-project-package-file-delete command line option, [696](#)
- gitlab-project-package-file-list command line option, [696](#)
- gitlab-project-package-get command line option, [695](#)
- gitlab-project-package-list command line option, [695](#)
- gitlab-project-pages-domain command line option, [697](#)
- gitlab-project-pages-domain-create command line option, [697](#)
- gitlab-project-pages-domain-delete command line option, [697](#)
- gitlab-project-pages-domain-get command line option, [698](#)
- gitlab-project-pages-domain-list command line option, [698](#)
- gitlab-project-pages-domain-update command line option, [698](#)
- gitlab-project-pipeline command line option, [699](#)
- gitlab-project-pipeline-bridge command line option, [701](#)
- gitlab-project-pipeline-bridge-list command line option, [701](#)
- gitlab-project-pipeline-cancel command line option, [699](#)
- gitlab-project-pipeline-create command line option, [699](#)
- gitlab-project-pipeline-delete command line option, [699](#)
- gitlab-project-pipeline-get command line option, [700](#)
- gitlab-project-pipeline-job command line option, [702](#)
- gitlab-project-pipeline-job-list command line option, [702](#)
- gitlab-project-pipeline-list command line option, [700](#)
- gitlab-project-pipeline-retry command line option, [701](#)
- gitlab-project-pipeline-schedule command line option, [702](#)
- gitlab-project-pipeline-schedule-create command line option, [703](#)
- gitlab-project-pipeline-schedule-delete command line option, [703](#)
- gitlab-project-pipeline-schedule-get command line option, [703](#)
- gitlab-project-pipeline-schedule-list command line option, [704](#)
- gitlab-project-pipeline-schedule-play command line option, [704](#)
- gitlab-project-pipeline-schedule-take-ownership command line option, [704](#)
- gitlab-project-pipeline-schedule-update command line option, [705](#)
- gitlab-project-pipeline-schedule-variable command line option, [705](#)
- gitlab-project-pipeline-schedule-variable-create command line option, [705](#)
- gitlab-project-pipeline-schedule-variable-delete command line option, [706](#)
- gitlab-project-pipeline-schedule-variable-update command line option, [706](#)
- gitlab-project-pipeline-test-report command line option, [707](#)
- gitlab-project-pipeline-test-report-get command line option, [707](#)
- gitlab-project-pipeline-test-report-summary command line option, [707](#)
- gitlab-project-pipeline-test-report-summary-get command line option, [707](#)
- gitlab-project-pipeline-variable command line option, [708](#)
- gitlab-project-pipeline-variable-list command line option, [708](#)
- gitlab-project-protected-branch command line option, [708](#)
- gitlab-project-protected-branch-create command line option, [709](#)
- gitlab-project-protected-branch-delete command line option, [709](#)
- gitlab-project-protected-branch-get command line option, [709](#)
- gitlab-project-protected-branch-list command line option, [710](#)

`gitlab-project-protected-environment`  
command line option, [710](#)

`gitlab-project-protected-environment-create`  
command line option, [710](#)

`gitlab-project-protected-environment-delete`  
command line option, [711](#)

`gitlab-project-protected-environment-get`  
command line option, [711](#)

`gitlab-project-protected-environment-list`  
command line option, [711](#)

`gitlab-project-protected-tag` command  
line option, [712](#)

`gitlab-project-protected-tag-create`  
command line option, [712](#)

`gitlab-project-protected-tag-delete`  
command line option, [712](#)

`gitlab-project-protected-tag-get`  
command line option, [713](#)

`gitlab-project-protected-tag-list`  
command line option, [713](#)

`gitlab-project-push-rules` command line  
option, [713](#)

`gitlab-project-push-rules-create`  
command line option, [714](#)

`gitlab-project-push-rules-delete`  
command line option, [714](#)

`gitlab-project-push-rules-get` command  
line option, [715](#)

`gitlab-project-push-rules-update`  
command line option, [715](#)

`gitlab-project-registry-repository`  
command line option, [716](#)

`gitlab-project-registry-repository-delete`  
command line option, [716](#)

`gitlab-project-registry-repository-list`  
command line option, [716](#)

`gitlab-project-registry-tag` command  
line option, [717](#)

`gitlab-project-registry-tag-delete`  
command line option, [717](#)

`gitlab-project-registry-tag-delete-in-bulk`  
command line option, [717](#)

`gitlab-project-registry-tag-get` command  
line option, [718](#)

`gitlab-project-registry-tag-list`  
command line option, [718](#)

`gitlab-project-release` command line  
option, [718](#)

`gitlab-project-release-create` command  
line option, [719](#)

`gitlab-project-release-delete` command  
line option, [719](#)

`gitlab-project-release-get` command line  
option, [719](#)

`gitlab-project-release-link` command  
line option, [721](#)

`gitlab-project-release-link-create`  
command line option, [721](#)

`gitlab-project-release-link-delete`  
command line option, [721](#)

`gitlab-project-release-link-get` command  
line option, [722](#)

`gitlab-project-release-link-list`  
command line option, [722](#)

`gitlab-project-release-link-update`  
command line option, [722](#)

`gitlab-project-release-list` command  
line option, [720](#)

`gitlab-project-release-update` command  
line option, [720](#)

`gitlab-project-remote-mirror` command  
line option, [723](#)

`gitlab-project-remote-mirror-create`  
command line option, [723](#)

`gitlab-project-remote-mirror-list`  
command line option, [723](#)

`gitlab-project-remote-mirror-update`  
command line option, [724](#)

`gitlab-project-repository-archive`  
command line option, [567](#)

`gitlab-project-repository-blob` command  
line option, [567](#)

`gitlab-project-repository-compare`  
command line option, [567](#)

`gitlab-project-repository-contributors`  
command line option, [568](#)

`gitlab-project-repository-merge-base`  
command line option, [568](#)

`gitlab-project-repository-raw-blob`  
command line option, [568](#)

`gitlab-project-repository-tree` command  
line option, [568](#)

`gitlab-project-restore` command line  
option, [569](#)

`gitlab-project-runner` command line  
option, [724](#)

`gitlab-project-runner-create` command  
line option, [724](#)

`gitlab-project-runner-delete` command  
line option, [725](#)

`gitlab-project-runner-list` command line  
option, [725](#)

`gitlab-project-search` command line  
option, [569](#)

`gitlab-project-service` command line  
option, [725](#)

`gitlab-project-service-available`  
command line option, [726](#)



- gitlab-project-service-delete command line option, [726](#)
- gitlab-project-service-get command line option, [726](#)
- gitlab-project-service-list command line option, [726](#)
- gitlab-project-service-update command line option, [727](#)
- gitlab-project-share command line option, [569](#)
- gitlab-project-snapshot command line option, [569](#)
- gitlab-project-snippet command line option, [727](#)
- gitlab-project-snippet-award-emoji command line option, [730](#)
- gitlab-project-snippet-award-emoji-create command line option, [730](#)
- gitlab-project-snippet-award-emoji-delete command line option, [730](#)
- gitlab-project-snippet-award-emoji-get command line option, [731](#)
- gitlab-project-snippet-award-emoji-list command line option, [731](#)
- gitlab-project-snippet-content command line option, [727](#)
- gitlab-project-snippet-create command line option, [728](#)
- gitlab-project-snippet-delete command line option, [728](#)
- gitlab-project-snippet-discussion command line option, [732](#)
- gitlab-project-snippet-discussion-create command line option, [732](#)
- gitlab-project-snippet-discussion-get command line option, [732](#)
- gitlab-project-snippet-discussion-list command line option, [733](#)
- gitlab-project-snippet-discussion-note command line option, [733](#)
- gitlab-project-snippet-discussion-note-create command line option, [733](#)
- gitlab-project-snippet-discussion-note-delete command line option, [734](#)
- gitlab-project-snippet-discussion-note-get command line option, [734](#)
- gitlab-project-snippet-discussion-note-update command line option, [735](#)
- gitlab-project-snippet-get command line option, [728](#)
- gitlab-project-snippet-list command line option, [729](#)
- gitlab-project-snippet-note command line option, [735](#)
- gitlab-project-snippet-note-award-emoji command line option, [737](#)
- gitlab-project-snippet-note-award-emoji-create command line option, [737](#)
- gitlab-project-snippet-note-award-emoji-delete command line option, [738](#)
- gitlab-project-snippet-note-award-emoji-get command line option, [738](#)
- gitlab-project-snippet-note-award-emoji-list command line option, [739](#)
- gitlab-project-snippet-note-create command line option, [735](#)
- gitlab-project-snippet-note-delete command line option, [736](#)
- gitlab-project-snippet-note-get command line option, [736](#)
- gitlab-project-snippet-note-list command line option, [736](#)
- gitlab-project-snippet-note-update command line option, [737](#)
- gitlab-project-snippet-update command line option, [729](#)
- gitlab-project-snippet-user-agent-detail command line option, [730](#)
- gitlab-project-star command line option, [570](#)
- gitlab-project-storage command line option, [739](#)
- gitlab-project-storage-get command line option, [739](#)
- gitlab-project-tag command line option, [740](#)
- gitlab-project-tag-create command line option, [740](#)
- gitlab-project-tag-delete command line option, [740](#)
- gitlab-project-tag-get command line option, [740](#)
- gitlab-project-tag-list command line option, [741](#)
- gitlab-project-transfer command line option, [570](#)
- gitlab-project-transfer-project command line option, [570](#)
- gitlab-project-trigger command line option, [741](#)
- gitlab-project-trigger-create command line option, [741](#)
- gitlab-project-trigger-delete command line option, [742](#)
- gitlab-project-trigger-get command line option, [742](#)
- gitlab-project-trigger-list command line option, [742](#)

gitlab-project-trigger-pipeline command line option, [570](#)  
gitlab-project-trigger-update command line option, [743](#)  
gitlab-project-unarchive command line option, [571](#)  
gitlab-project-unshare command line option, [571](#)  
gitlab-project-unstar command line option, [571](#)  
gitlab-project-update command line option, [573](#)  
gitlab-project-update-submodule command line option, [575](#)  
gitlab-project-upload command line option, [575](#)  
gitlab-project-user command line option, [743](#)  
gitlab-project-user-list command line option, [743](#)  
gitlab-project-variable command line option, [744](#)  
gitlab-project-variable-create command line option, [744](#)  
gitlab-project-variable-delete command line option, [744](#)  
gitlab-project-variable-get command line option, [745](#)  
gitlab-project-variable-list command line option, [745](#)  
gitlab-project-variable-update command line option, [745](#)  
gitlab-project-wiki command line option, [746](#)  
gitlab-project-wiki-create command line option, [746](#)  
gitlab-project-wiki-delete command line option, [746](#)  
gitlab-project-wiki-get command line option, [747](#)  
gitlab-project-wiki-list command line option, [747](#)  
gitlab-project-wiki-update command line option, [747](#)  
gitlab-registry-repository command line option, [748](#)  
gitlab-registry-repository-get command line option, [748](#)  
gitlab-runner command line option, [748](#)  
gitlab-runner-all command line option, [748](#), [751](#)  
gitlab-runner-all-list command line option, [751](#)  
gitlab-runner-create command line option, [749](#)  
gitlab-runner-delete command line option, [749](#)  
gitlab-runner-get command line option, [749](#)  
gitlab-runner-job command line option, [752](#)  
gitlab-runner-job-list command line option, [752](#)  
gitlab-runner-list command line option, [750](#)  
gitlab-runner-update command line option, [750](#)  
gitlab-runner-verify command line option, [751](#)  
gitlab-shared-project command line option, [752](#)  
gitlab-shared-project-list command line option, [753](#)  
gitlab-snippet command line option, [753](#)  
gitlab-snippet-content command line option, [754](#)  
gitlab-snippet-create command line option, [754](#)  
gitlab-snippet-delete command line option, [754](#)  
gitlab-snippet-get command line option, [755](#)  
gitlab-snippet-list command line option, [755](#)  
gitlab-snippet-public command line option, [755](#)  
gitlab-snippet-update command line option, [755](#)  
gitlab-snippet-user-agent-detail command line option, [756](#)  
gitlab-starred-project command line option, [756](#)  
gitlab-starred-project-list command line option, [756](#)  
gitlab-todo command line option, [757](#)  
gitlab-todo-delete command line option, [757](#)  
gitlab-todo-list command line option, [758](#)  
gitlab-todo-mark-all-as-done command line option, [758](#)  
gitlab-todo-mark-as-done command line option, [758](#)  
gitlab-topic command line option, [759](#)  
gitlab-topic-create command line option, [759](#)  
gitlab-topic-delete command line option, [759](#)  
gitlab-topic-get command line option, [759](#)

- gitlab-topic-list command line option, [760](#)
- gitlab-topic-update command line option, [760](#)
- gitlab-user command line option, [761](#)
- gitlab-user-activate command line option, [761](#)
- gitlab-user-activities command line option, [767](#)
- gitlab-user-activities-list command line option, [767](#)
- gitlab-user-approve command line option, [761](#)
- gitlab-user-ban command line option, [761](#)
- gitlab-user-block command line option, [761](#)
- gitlab-user-create command line option, [762](#)
- gitlab-user-custom-attribute command line option, [767](#)
- gitlab-user-custom-attribute-delete command line option, [767](#)
- gitlab-user-custom-attribute-get command line option, [768](#)
- gitlab-user-custom-attribute-list command line option, [768](#)
- gitlab-user-deactivate command line option, [763](#)
- gitlab-user-delete command line option, [763](#)
- gitlab-user-email command line option, [768](#)
- gitlab-user-email-create command line option, [769](#)
- gitlab-user-email-delete command line option, [769](#)
- gitlab-user-email-get command line option, [769](#)
- gitlab-user-email-list command line option, [769](#)
- gitlab-user-event command line option, [770](#)
- gitlab-user-event-list command line option, [770](#)
- gitlab-user-follow command line option, [763](#)
- gitlab-user-get command line option, [764](#)
- gitlab-user-gpg-key command line option, [771](#)
- gitlab-user-gpg-key-create command line option, [771](#)
- gitlab-user-gpg-key-delete command line option, [771](#)
- gitlab-user-gpg-key-get command line option, [771](#)
- gitlab-user-gpg-key-list command line option, [772](#)
- gitlab-user-impersonation-token command line option, [772](#)
- gitlab-user-impersonation-token-create command line option, [772](#)
- gitlab-user-impersonation-token-delete command line option, [773](#)
- gitlab-user-impersonation-token-get command line option, [773](#)
- gitlab-user-impersonation-token-list command line option, [773](#)
- gitlab-user-key command line option, [774](#)
- gitlab-user-key-create command line option, [774](#)
- gitlab-user-key-delete command line option, [774](#)
- gitlab-user-key-get command line option, [774](#)
- gitlab-user-key-list command line option, [775](#)
- gitlab-user-list command line option, [764](#)
- gitlab-user-membership command line option, [775](#)
- gitlab-user-membership-get command line option, [775](#)
- gitlab-user-membership-list command line option, [775](#)
- gitlab-user-personal-access-token command line option, [776](#)
- gitlab-user-personal-access-token-create command line option, [776](#)
- gitlab-user-project command line option, [776](#)
- gitlab-user-project-create command line option, [777](#)
- gitlab-user-project-list command line option, [778](#)
- gitlab-user-reject command line option, [765](#)
- gitlab-user-status command line option, [779](#)
- gitlab-user-status-get command line option, [779](#)
- gitlab-user-unban command line option, [765](#)
- gitlab-user-unblock command line option, [765](#)
- gitlab-user-unfollow command line option, [765](#)
- gitlab-user-update command line option, [766](#)
- gitlab-variable command line option, [779](#)

gitlab-variable-create command line option, [780](#)  
gitlab-variable-delete command line option, [780](#)  
gitlab-variable-get command line option, [780](#)  
gitlab-variable-list command line option, [781](#)  
gitlab-variable-update command line option, [781](#)  
--home-page-url  
    gitlab-application-settings-update command line option, [464](#)  
--id  
    gitlab-application-delete command line option, [460](#)  
    gitlab-application-settings-update command line option, [464](#)  
    gitlab-audit-event-get command line option, [466](#)  
    gitlab-broadcast-message-delete command line option, [467](#)  
    gitlab-broadcast-message-get command line option, [467](#)  
    gitlab-broadcast-message-update command line option, [468](#)  
    gitlab-current-user-email-delete command line option, [470](#)  
    gitlab-current-user-email-get command line option, [470](#)  
    gitlab-current-user-gpg-key-delete command line option, [471](#)  
    gitlab-current-user-gpg-key-get command line option, [472](#)  
    gitlab-current-user-key-delete command line option, [473](#)  
    gitlab-current-user-key-get command line option, [473](#)  
    gitlab-geo-node-current-failures command line option, [479](#)  
    gitlab-geo-node-delete command line option, [479](#)  
    gitlab-geo-node-get command line option, [479](#)  
    gitlab-geo-node-repair command line option, [480](#)  
    gitlab-geo-node-status command line option, [480](#)  
    gitlab-geo-node-update command line option, [480](#)  
    gitlab-group-access-request-approve command line option, [489](#)  
    gitlab-group-access-request-delete command line option, [489](#)  
    gitlab-group-access-token-delete command line option, [491](#)  
    gitlab-group-add-ldap-group-link command line option, [483](#)  
    gitlab-group-audit-event-get command line option, [492](#)  
    gitlab-group-badge-delete command line option, [493](#)  
    gitlab-group-badge-get command line option, [493](#)  
    gitlab-group-badge-render command line option, [494](#)  
    gitlab-group-badge-update command line option, [494](#)  
    gitlab-group-billable-member-delete command line option, [495](#)  
    gitlab-group-board-delete command line option, [497](#)  
    gitlab-group-board-get command line option, [497](#)  
    gitlab-group-board-list-delete command line option, [498](#)  
    gitlab-group-board-list-get command line option, [499](#)  
    gitlab-group-board-list-update command line option, [500](#)  
    gitlab-group-board-update command line option, [498](#)  
    gitlab-group-cluster-delete command line option, [500](#)  
    gitlab-group-cluster-get command line option, [501](#)  
    gitlab-group-cluster-update command line option, [501](#)  
    gitlab-group-delete command line option, [484](#)  
    gitlab-group-delete-ldap-group-link command line option, [485](#)  
    gitlab-group-deploy-token-delete command line option, [504](#)  
    gitlab-group-deploy-token-get command line option, [504](#)  
    gitlab-group-epic-award-emoji-delete command line option, [509](#)  
    gitlab-group-epic-award-emoji-get command line option, [509](#)  
    gitlab-group-epic-discussion-note-delete command line option, [511](#)  
    gitlab-group-epic-discussion-note-get command line option, [511](#)  
    gitlab-group-epic-discussion-note-update command line option, [511](#)  
    gitlab-group-epic-note-award-emoji-delete command line option, [516](#)

gitlab-group-epic-note-award-emoji-get  
command line option, [516](#)

gitlab-group-epic-note-delete command  
line option, [514](#)

gitlab-group-epic-note-get command line  
option, [514](#)

gitlab-group-epic-note-update command  
line option, [515](#)

gitlab-group-epic-resource-label-event-get  
command line option, [517](#)

gitlab-group-get command line option, [485](#)

gitlab-group-hook-delete command line  
option, [520](#)

gitlab-group-hook-get command line  
option, [520](#)

gitlab-group-hook-update command line  
option, [521](#)

gitlab-group-ldap-sync command line  
option, [485](#)

gitlab-group-member-all-get command  
line option, [532](#)

gitlab-group-member-delete command line  
option, [530](#)

gitlab-group-member-get command line  
option, [530](#)

gitlab-group-member-update command line  
option, [531](#)

gitlab-group-milestone-delete command  
line option, [535](#)

gitlab-group-milestone-get command line  
option, [535](#)

gitlab-group-milestone-issues command  
line option, [535](#)

gitlab-group-milestone-merge-requests  
command line option, [536](#)

gitlab-group-milestone-update command  
line option, [536](#)

gitlab-group-search command line option,  
[486](#)

gitlab-group-share command line option,  
[486](#)

gitlab-group-transfer command line  
option, [487](#)

gitlab-group-transfer-project command  
line option, [487](#)

gitlab-group-unshare command line  
option, [487](#)

gitlab-group-update command line option,  
[488](#)

gitlab-hook-delete command line option,  
[548](#)

gitlab-hook-get command line option, [549](#)

gitlab-issue-get command line option, [549](#)

gitlab-key-get command line option, [551](#)

gitlab-namespace-exists command line  
option, [555](#)

gitlab-namespace-get command line  
option, [555](#)

gitlab-personal-access-token-delete  
command line option, [558](#)

gitlab-project-access-request-approve  
command line option, [576](#)

gitlab-project-access-request-delete  
command line option, [576](#)

gitlab-project-access-token-delete  
command line option, [578](#)

gitlab-project-approval-rule-delete  
command line option, [580](#)

gitlab-project-approval-rule-update  
command line option, [581](#)

gitlab-project-archive command line  
option, [559](#)

gitlab-project-artifact command line  
option, [560](#)

gitlab-project-artifacts command line  
option, [560](#)

gitlab-project-audit-event-get command  
line option, [583](#)

gitlab-project-badge-delete command  
line option, [584](#)

gitlab-project-badge-get command line  
option, [584](#)

gitlab-project-badge-render command  
line option, [585](#)

gitlab-project-badge-update command  
line option, [585](#)

gitlab-project-board-delete command  
line option, [586](#)

gitlab-project-board-get command line  
option, [587](#)

gitlab-project-board-list-delete  
command line option, [588](#)

gitlab-project-board-list-get command  
line option, [588](#)

gitlab-project-board-list-update  
command line option, [589](#)

gitlab-project-board-update command  
line option, [587](#)

gitlab-project-cluster-delete command  
line option, [593](#)

gitlab-project-cluster-get command line  
option, [593](#)

gitlab-project-cluster-update command  
line option, [594](#)

gitlab-project-commit-cherry-pick  
command line option, [595](#)

gitlab-project-commit-diff command line  
option, [596](#)

`gitlab-project-commit-discussion-get`  
command line option, [600](#)

`gitlab-project-commit-discussion-note-delete`  
command line option, [602](#)

`gitlab-project-commit-discussion-note-get`  
command line option, [602](#)

`gitlab-project-commit-discussion-note-update`  
command line option, [602](#)

`gitlab-project-commit-get` command line  
option, [596](#)

`gitlab-project-commit-merge-requests`  
command line option, [597](#)

`gitlab-project-commit-refs` command line  
option, [597](#)

`gitlab-project-commit-revert` command  
line option, [598](#)

`gitlab-project-commit-signature` command  
line option, [598](#)

`gitlab-project-create-fork-relation`  
command line option, [564](#)

`gitlab-project-delete` command line  
option, [564](#)

`gitlab-project-delete-fork-relation`  
command line option, [564](#)

`gitlab-project-delete-merged-branches`  
command line option, [564](#)

`gitlab-project-deploy-token-delete`  
command line option, [606](#)

`gitlab-project-deploy-token-get` command  
line option, [606](#)

`gitlab-project-deployment-get` command  
line option, [607](#)

`gitlab-project-deployment-update`  
command line option, [608](#)

`gitlab-project-environment-delete`  
command line option, [612](#)

`gitlab-project-environment-get` command  
line option, [612](#)

`gitlab-project-environment-stop` command  
line option, [613](#)

`gitlab-project-environment-update`  
command line option, [613](#)

`gitlab-project-get` command line option,  
[565](#)

`gitlab-project-hook-delete` command line  
option, [620](#)

`gitlab-project-hook-get` command line  
option, [621](#)

`gitlab-project-hook-update` command line  
option, [622](#)

`gitlab-project-housekeeping` command  
line option, [565](#)

`gitlab-project-issue-award-emoji-delete`  
command line option, [635](#)

`gitlab-project-issue-award-emoji-get`  
command line option, [636](#)

`gitlab-project-issue-discussion-get`  
command line option, [637](#)

`gitlab-project-issue-discussion-note-delete`  
command line option, [639](#)

`gitlab-project-issue-discussion-note-get`  
command line option, [639](#)

`gitlab-project-issue-discussion-note-update`  
command line option, [640](#)

`gitlab-project-issue-note-award-emoji-delete`  
command line option, [644](#)

`gitlab-project-issue-note-award-emoji-get`  
command line option, [644](#)

`gitlab-project-issue-note-delete`  
command line option, [642](#)

`gitlab-project-issue-note-get` command  
line option, [642](#)

`gitlab-project-issue-note-update`  
command line option, [643](#)

`gitlab-project-issue-resource-label-event-get`  
command line option, [645](#)

`gitlab-project-issue-resource-milestone-event-get`  
command line option, [646](#)

`gitlab-project-issue-resource-state-event-get`  
command line option, [647](#)

`gitlab-project-job-artifact` command  
line option, [649](#)

`gitlab-project-job-artifacts` command  
line option, [649](#)

`gitlab-project-job-cancel` command line  
option, [650](#)

`gitlab-project-job-delete-artifacts`  
command line option, [650](#)

`gitlab-project-job-erase` command line  
option, [650](#)

`gitlab-project-job-get` command line  
option, [651](#)

`gitlab-project-job-keep-artifacts`  
command line option, [651](#)

`gitlab-project-job-play` command line  
option, [652](#)

`gitlab-project-job-retry` command line  
option, [652](#)

`gitlab-project-job-trace` command line  
option, [652](#)

`gitlab-project-key-delete` command line  
option, [653](#)

`gitlab-project-key-enable` command line  
option, [653](#)

`gitlab-project-key-get` command line  
option, [653](#)

`gitlab-project-key-update` command line  
option, [654](#)



- gitlab-project-languages command line option, 565
- gitlab-project-member-all-get command line option, 659
- gitlab-project-member-delete command line option, 658
- gitlab-project-member-get command line option, 658
- gitlab-project-member-update command line option, 659
- gitlab-project-merge-request-approval-rule-create command line option, 671
- gitlab-project-merge-request-approval-rule-delete command line option, 672
- gitlab-project-merge-request-approval-rule-get command line option, 672
- gitlab-project-merge-request-approval-rule-update command line option, 673
- gitlab-project-merge-request-approval-state-get command line option, 674
- gitlab-project-merge-request-award-emoji-delete command line option, 675
- gitlab-project-merge-request-award-emoji-get command line option, 675
- gitlab-project-merge-request-diff-get command line option, 676
- gitlab-project-merge-request-discussion-get command line option, 678
- gitlab-project-merge-request-discussion-note-delete command line option, 680
- gitlab-project-merge-request-discussion-note-get command line option, 680
- gitlab-project-merge-request-discussion-note-update command line option, 680
- gitlab-project-merge-request-discussion-update command line option, 678
- gitlab-project-merge-request-note-award-emoji-delete command line option, 684
- gitlab-project-merge-request-note-award-emoji-get command line option, 684
- gitlab-project-merge-request-note-delete command line option, 681
- gitlab-project-merge-request-note-get command line option, 682
- gitlab-project-merge-request-note-update command line option, 683
- gitlab-project-merge-request-resource-label-event-get command line option, 686
- gitlab-project-merge-request-resource-milestone-get command line option, 687
- gitlab-project-merge-request-resource-state-event-get command line option, 688
- gitlab-project-milestone-delete command line option, 690
- gitlab-project-milestone-get command line option, 690
- gitlab-project-milestone-issues command line option, 691
- gitlab-project-milestone-merge-requests command line option, 692
- gitlab-project-milestone-update command line option, 692
- gitlab-project-mirror-pull command line option, 567
- gitlab-project-note-get command line option, 692
- gitlab-project-package-delete command line option, 695
- gitlab-project-package-file-delete command line option, 696
- gitlab-project-package-get command line option, 695
- gitlab-project-pipeline-cancel command line option, 699
- gitlab-project-pipeline-delete command line option, 699
- gitlab-project-pipeline-get command line option, 700
- gitlab-project-pipeline-retry command line option, 701
- gitlab-project-pipeline-schedule-delete command line option, 703
- gitlab-project-pipeline-schedule-get command line option, 703
- gitlab-project-pipeline-schedule-play command line option, 704
- gitlab-project-pipeline-schedule-take-ownership command line option, 704
- gitlab-project-pipeline-schedule-update command line option, 705
- gitlab-project-registry-repository-delete command line option, 716
- gitlab-project-release-link-delete command line option, 721
- gitlab-project-release-link-get command line option, 722
- gitlab-project-release-link-update command line option, 723
- gitlab-project-remote-mirror-update command line option, 724
- gitlab-project-repository-archive command line option, 567
- gitlab-project-repository-blob command line option, 567
- gitlab-project-repository-compare command line option, 567
- gitlab-project-repository-contributors command line option, 568

gitlab-project-repository-merge-base  
command line option, [568](#)

gitlab-project-repository-raw-blob  
command line option, [568](#)

gitlab-project-repository-tree command  
line option, [568](#)

gitlab-project-restore command line  
option, [569](#)

gitlab-project-runner-delete command  
line option, [725](#)

gitlab-project-search command line  
option, [569](#)

gitlab-project-share command line  
option, [569](#)

gitlab-project-snapshot command line  
option, [569](#)

gitlab-project-snippet-award-emoji-delete  
command line option, [731](#)

gitlab-project-snippet-award-emoji-get  
command line option, [731](#)

gitlab-project-snippet-content command  
line option, [727](#)

gitlab-project-snippet-delete command  
line option, [728](#)

gitlab-project-snippet-discussion-get  
command line option, [732](#)

gitlab-project-snippet-discussion-note-delete  
command line option, [734](#)

gitlab-project-snippet-discussion-note-get  
command line option, [734](#)

gitlab-project-snippet-discussion-note-update  
command line option, [735](#)

gitlab-project-snippet-get command line  
option, [728](#)

gitlab-project-snippet-note-award-emoji-delete  
command line option, [738](#)

gitlab-project-snippet-note-award-emoji-get  
command line option, [738](#)

gitlab-project-snippet-note-delete  
command line option, [736](#)

gitlab-project-snippet-note-get command  
line option, [736](#)

gitlab-project-snippet-note-update  
command line option, [737](#)

gitlab-project-snippet-update command  
line option, [729](#)

gitlab-project-snippet-user-agent-detail  
command line option, [730](#)

gitlab-project-star command line option,  
[570](#)

gitlab-project-storage-get command line  
option, [739](#)

gitlab-project-transfer command line  
option, [570](#)

gitlab-project-transfer-project command  
line option, [570](#)

gitlab-project-trigger-delete command  
line option, [742](#)

gitlab-project-trigger-get command line  
option, [742](#)

gitlab-project-trigger-pipeline command  
line option, [570](#)

gitlab-project-trigger-update command  
line option, [743](#)

gitlab-project-unarchive command line  
option, [571](#)

gitlab-project-unshare command line  
option, [571](#)

gitlab-project-unstar command line  
option, [571](#)

gitlab-project-update command line  
option, [573](#)

gitlab-project-update-submodule command  
line option, [575](#)

gitlab-project-upload command line  
option, [575](#)

gitlab-registry-repository-get command  
line option, [748](#)

gitlab-runner-all command line option,  
[748](#)

gitlab-runner-delete command line  
option, [749](#)

gitlab-runner-get command line option,  
[749](#)

gitlab-runner-update command line  
option, [750](#)

gitlab-runner-verify command line  
option, [751](#)

gitlab-snippet-content command line  
option, [754](#)

gitlab-snippet-delete command line  
option, [754](#)

gitlab-snippet-get command line option,  
[755](#)

gitlab-snippet-public command line  
option, [755](#)

gitlab-snippet-update command line  
option, [755](#)

gitlab-snippet-user-agent-detail  
command line option, [756](#)

gitlab-todo-delete command line option,  
[757](#)

gitlab-todo-mark-all-as-done command  
line option, [758](#)

gitlab-todo-mark-as-done command line  
option, [758](#)

gitlab-topic-delete command line option,  
[759](#)



- gitlab-topic-get command line option, [759](#)
- gitlab-topic-update command line option, [760](#)
- gitlab-user-activate command line option, [761](#)
- gitlab-user-approve command line option, [761](#)
- gitlab-user-ban command line option, [761](#)
- gitlab-user-block command line option, [761](#)
- gitlab-user-deactivate command line option, [763](#)
- gitlab-user-delete command line option, [763](#)
- gitlab-user-email-delete command line option, [769](#)
- gitlab-user-email-get command line option, [769](#)
- gitlab-user-follow command line option, [763](#)
- gitlab-user-get command line option, [764](#)
- gitlab-user-gpg-key-delete command line option, [771](#)
- gitlab-user-gpg-key-get command line option, [771](#)
- gitlab-user-impersonation-token-delete command line option, [773](#)
- gitlab-user-impersonation-token-get command line option, [773](#)
- gitlab-user-key-delete command line option, [774](#)
- gitlab-user-key-get command line option, [774](#)
- gitlab-user-reject command line option, [765](#)
- gitlab-user-unban command line option, [765](#)
- gitlab-user-unblock command line option, [765](#)
- gitlab-user-unfollow command line option, [765](#)
- gitlab-user-update command line option, [766](#)
- id-after
  - gitlab-project-list command line option, [566](#)
  - gitlab-user-project-list command line option, [779](#)
- id-before
  - gitlab-project-list command line option, [566](#)
  - gitlab-user-project-list command line option, [779](#)
- iid
  - gitlab-group-epic-delete command line option, [506](#)
  - gitlab-group-epic-get command line option, [507](#)
  - gitlab-group-epic-update command line option, [508](#)
  - gitlab-project-issue-add-spent-time command line option, [627](#)
  - gitlab-project-issue-closed-by command line option, [628](#)
  - gitlab-project-issue-delete command line option, [629](#)
  - gitlab-project-issue-get command line option, [629](#)
  - gitlab-project-issue-move command line option, [630](#)
  - gitlab-project-issue-participants command line option, [631](#)
  - gitlab-project-issue-related-merge-requests command line option, [631](#)
  - gitlab-project-issue-reorder command line option, [631](#)
  - gitlab-project-issue-reset-spent-time command line option, [632](#)
  - gitlab-project-issue-reset-time-estimate command line option, [632](#)
  - gitlab-project-issue-subscribe command line option, [632](#)
  - gitlab-project-issue-time-estimate command line option, [633](#)
  - gitlab-project-issue-time-stats command line option, [633](#)
  - gitlab-project-issue-todo command line option, [633](#)
  - gitlab-project-issue-unsubscribe command line option, [633](#)
  - gitlab-project-issue-update command line option, [634](#)
  - gitlab-project-issue-user-agent-detail command line option, [635](#)
  - gitlab-project-merge-request-add-spent-time command line option, [661](#)
  - gitlab-project-merge-request-approve command line option, [661](#)
  - gitlab-project-merge-request-cancel-merge-when-pipeline command line option, [661](#)
  - gitlab-project-merge-request-changes command line option, [661](#)
  - gitlab-project-merge-request-closes-issues command line option, [662](#)
  - gitlab-project-merge-request-commits command line option, [662](#)
  - gitlab-project-merge-request-delete command line option, [663](#)

```
gitlab-project-merge-request-get
    command line option, 663
gitlab-project-merge-request-merge
    command line option, 665
gitlab-project-merge-request-merge-ref
    command line option, 666
gitlab-project-merge-request-participants
    command line option, 666
gitlab-project-merge-request-rebase
    command line option, 666
gitlab-project-merge-request-reset-spent-time
    command line option, 667
gitlab-project-merge-request-reset-time-estimate
    command line option, 667
gitlab-project-merge-request-subscribe
    command line option, 667
gitlab-project-merge-request-time-estimate
    command line option, 667
gitlab-project-merge-request-time-stats
    command line option, 668
gitlab-project-merge-request-todo
    command line option, 668
gitlab-project-merge-request-unapprove
    command line option, 668
gitlab-project-merge-request-unsubscribe
    command line option, 669
gitlab-project-merge-request-update
    command line option, 669
--iids
    gitlab-group-issue-list command line
        option, 525
    gitlab-group-milestone-list command
        line option, 536
    gitlab-issue-list command line option,
        550
    gitlab-project-issue-list command line
        option, 629
    gitlab-project-merge-request-list
        command line option, 665
    gitlab-project-milestone-list command
        line option, 691
--image-url
    gitlab-group-badge-create command line
        option, 493
    gitlab-group-badge-render command line
        option, 494
    gitlab-group-badge-update command line
        option, 494
    gitlab-project-badge-create command
        line option, 584
    gitlab-project-badge-render command
        line option, 585
    gitlab-project-badge-update command
        line option, 586
    --import-sources
        gitlab-application-settings-update
            command line option, 464
    --import-url
        gitlab-project-create command line
            option, 562
        gitlab-project-update command line
            option, 573
        gitlab-user-project-create command line
            option, 777
    gitlab-merge-request-list command line
        option, 555
    gitlab-project-deployment-merge-request-list
        command line option, 610
    --include-ancestors
        gitlab-group-iteration-list command
            line option, 526
    --include-diverged-commits-count
        gitlab-project-merge-request-get
            command line option, 664
    --include-html-description
        gitlab-project-release-list command
            line option, 720
    --include-jobs
        gitlab-ci-lint-create command line
            option, 469
        gitlab-ci-lint-validate command line
            option, 469
        gitlab-project-ci-lint-create command
            line option, 591
        gitlab-project-ci-lint-validate command
            line option, 592
    --include-merged-yaml
        gitlab-ci-lint-create command line
            option, 469
        gitlab-ci-lint-validate command line
            option, 469
    --include-rebase-in-progress
        gitlab-project-merge-request-get
            command line option, 664
    --include-retried
        gitlab-project-pipeline-job-list
            command line option, 702
    --include-subgroups
        gitlab-group-project-list command line
            option, 540
    --info
        gitlab-runner-create command line
            option, 749
    --initialize-with-readme
        gitlab-project-create command line
            option, 562
    --instance-statistics-visibility-private
```

---

```

gitlab-application-settings-update
    command line option, 465
--invite-source
    gitlab-group-invitation-create command
        line option, 523
    gitlab-project-invitation-create
        command line option, 625
--issue-id
    gitlab-group-epic-issue-create command
        line option, 512
--issue-iid
    gitlab-project-issue-award-emoji-create
        command line option, 635
    gitlab-project-issue-award-emoji-delete
        command line option, 635
    gitlab-project-issue-award-emoji-get
        command line option, 636
    gitlab-project-issue-award-emoji-list
        command line option, 636
    gitlab-project-issue-discussion-create
        command line option, 637
    gitlab-project-issue-discussion-get
        command line option, 637
    gitlab-project-issue-discussion-list
        command line option, 637
    gitlab-project-issue-discussion-note-create
        command line option, 638
    gitlab-project-issue-discussion-note-delete
        command line option, 639
    gitlab-project-issue-discussion-note-get
        command line option, 639
    gitlab-project-issue-discussion-note-update
        command line option, 640
    gitlab-project-issue-link-create
        command line option, 640
    gitlab-project-issue-link-delete
        command line option, 640
    gitlab-project-issue-link-list command
        line option, 641
    gitlab-project-issue-note-award-emoji-create
        command line option, 643
    gitlab-project-issue-note-award-emoji-delete
        command line option, 644
    gitlab-project-issue-note-award-emoji-get
        command line option, 644
    gitlab-project-issue-note-award-emoji-list
        command line option, 645
    gitlab-project-issue-note-create
        command line option, 641
    gitlab-project-issue-note-delete
        command line option, 642
    gitlab-project-issue-note-get command
        line option, 642
    gitlab-project-issue-note-list command
        line option, 642
    gitlab-project-issue-note-update
        command line option, 643
    gitlab-project-issue-resource-label-event-get
        command line option, 645
    gitlab-project-issue-resource-label-event-list
        command line option, 646
    gitlab-project-issue-resource-milestone-event-get
        command line option, 646
    gitlab-project-issue-resource-milestone-event-list
        command line option, 647
    gitlab-project-issue-resource-state-event-get
        command line option, 647
    gitlab-project-issue-resource-state-event-list
        command line option, 648
--issue-link-id
    gitlab-project-issue-link-delete
        command line option, 640
--issues-access-level
    gitlab-project-create command line
        option, 562
    gitlab-project-update command line
        option, 573
--issues-enabled
    gitlab-project-create command line
        option, 562
    gitlab-project-update command line
        option, 573
    gitlab-user-project-create command line
        option, 777
--issues-events
    gitlab-group-hook-create command line
        option, 519
    gitlab-group-hook-update command line
        option, 521
    gitlab-project-hook-create command line
        option, 620
    gitlab-project-hook-update command line
        option, 622
issues-template
    gitlab-project-update command line
        option, 575
--job
    gitlab-project-artifact command line
        option, 560
    gitlab-project-artifact-download
        command line option, 582
    gitlab-project-artifact-raw command
        line option, 582
    gitlab-project-artifacts command line
        option, 560
--job-events
    gitlab-group-hook-create command line
        option, 519

```

- gitlab-group-hook-update command line option, [521](#)
- gitlab-project-hook-create command line option, [620](#)
- gitlab-project-hook-update command line option, [622](#)
- job-token
  - gitlab command line option, [459](#)
  - gitlab-project-artifact-download command line option, [582](#)
  - gitlab-project-artifacts command line option, [560](#)
- jobs-enabled
  - gitlab-project-create command line option, [562](#)
  - gitlab-project-update command line option, [574](#)
- keep-n
  - gitlab-project-registry-tag-delete-in-bulk command line option, [717](#)
- key
  - gitlab-current-user-gpg-key-create command line option, [471](#)
  - gitlab-current-user-key-create command line option, [472](#)
  - gitlab-group-custom-attribute-delete command line option, [502](#)
  - gitlab-group-custom-attribute-get command line option, [502](#)
  - gitlab-group-variable-create command line option, [544](#)
  - gitlab-group-variable-delete command line option, [545](#)
  - gitlab-group-variable-get command line option, [545](#)
  - gitlab-group-variable-update command line option, [546](#)
  - gitlab-license-get command line option, [552](#)
  - gitlab-project-custom-attribute-delete command line option, [604](#)
  - gitlab-project-custom-attribute-get command line option, [604](#)
  - gitlab-project-key-create command line option, [653](#)
  - gitlab-project-pages-domain-create command line option, [697](#)
  - gitlab-project-pages-domain-update command line option, [699](#)
  - gitlab-project-pipeline-schedule-variable-create command line option, [706](#)
  - gitlab-project-pipeline-schedule-variable-delete command line option, [706](#)
  - gitlab-project-pipeline-schedule-variable-update command line option, [706](#)
- gitlab-project-variable-create command line option, [744](#)
- gitlab-project-variable-delete command line option, [744](#)
- gitlab-project-variable-get command line option, [745](#)
- gitlab-project-variable-update command line option, [745](#)
- gitlab-user-custom-attribute-delete command line option, [767](#)
- gitlab-user-custom-attribute-get command line option, [768](#)
- gitlab-user-gpg-key-create command line option, [771](#)
- gitlab-user-key-create command line option, [774](#)
- gitlab-variable-create command line option, [780](#)
- gitlab-variable-delete command line option, [780](#)
- gitlab-variable-get command line option, [780](#)
- gitlab-variable-update command line option, [781](#)
- key-id
  - gitlab-project-key-enable command line option, [653](#)
- labels
  - gitlab-group-epic-create command line option, [506](#)
  - gitlab-group-epic-list command line option, [507](#)
  - gitlab-group-epic-update command line option, [508](#)
  - gitlab-group-issue-list command line option, [525](#)
  - gitlab-group-merge-request-list command line option, [533](#)
  - gitlab-issue-list command line option, [550](#)
  - gitlab-merge-request-list command line option, [554](#)
  - gitlab-project-deployment-merge-request-list command line option, [610](#)
  - gitlab-project-issue-create command line option, [628](#)
  - gitlab-project-issue-list command line option, [630](#)
  - gitlab-project-issue-update command line option, [634](#)
  - gitlab-project-merge-request-create command line option, [663](#)
  - gitlab-project-merge-request-list

- command line option, [664](#)
  - gitlab-project-merge-request-update
    - command line option, [669](#)
- last-activity-after
  - gitlab-project-list command line option, [566](#)
- last-activity-before
  - gitlab-project-list command line option, [566](#)
- level
  - gitlab-group-notification-settings-update
    - command line option, [538](#)
  - gitlab-notification-settings-update
    - command line option, [557](#)
  - gitlab-project-notification-settings-update
    - command line option, [694](#)
- lfs-enabled
  - gitlab-group-create command line option, [484](#)
  - gitlab-group-update command line option, [488](#)
  - gitlab-project-create command line option, [562](#)
  - gitlab-project-update command line option, [574](#)
- lifetime
  - gitlab-snippet-create command line option, [754](#)
- line
  - gitlab-project-commit-comment-create
    - command line option, [599](#)
- line-type
  - gitlab-project-commit-comment-create
    - command line option, [599](#)
- link-type
  - gitlab-project-release-link-create
    - command line option, [721](#)
  - gitlab-project-release-link-update
    - command line option, [723](#)
- link-url
  - gitlab-group-badge-create command line option, [493](#)
  - gitlab-group-badge-render command line option, [494](#)
  - gitlab-group-badge-update command line option, [494](#)
  - gitlab-project-badge-create command line option, [584](#)
  - gitlab-project-badge-render command line option, [585](#)
  - gitlab-project-badge-update command line option, [586](#)
- linkedin
  - gitlab-user-create command line option, [762](#)
  - gitlab-user-update command line option, [766](#)
- local-markdown-version
  - gitlab-application-settings-update
    - command line option, [465](#)
- location
  - gitlab-user-create command line option, [763](#)
  - gitlab-user-update command line option, [766](#)
- locked
  - gitlab-runner-create command line option, [749](#)
  - gitlab-runner-update command line option, [750](#)
- logo
  - gitlab-application-appearance-update
    - command line option, [461](#)
- managed
  - gitlab-group-cluster-create command line option, [500](#)
  - gitlab-project-cluster-create command line option, [593](#)
- management-project-id
  - gitlab-group-cluster-update command line option, [502](#)
  - gitlab-project-cluster-update command line option, [594](#)
- masked
  - gitlab-group-variable-create command line option, [545](#)
  - gitlab-group-variable-update command line option, [546](#)
  - gitlab-project-variable-create command line option, [744](#)
  - gitlab-project-variable-update command line option, [746](#)
  - gitlab-variable-create command line option, [780](#)
  - gitlab-variable-update command line option, [781](#)
- max-attachment-size
  - gitlab-application-settings-update
    - command line option, [464](#)
- max-file-size
  - gitlab-group-push-rules-create command line option, [541](#)
  - gitlab-group-push-rules-update command line option, [542](#)
  - gitlab-project-push-rules-create
    - command line option, [714](#)
  - gitlab-project-push-rules-update
    - command line option, [715](#)

--maximum-timeout  
gitlab-runner-create command line  
option, [749](#)  
gitlab-runner-update command line  
option, [751](#)

--member-check  
gitlab-group-push-rules-create command  
line option, [541](#)  
gitlab-group-push-rules-update command  
line option, [542](#)  
gitlab-project-push-rules-create  
command line option, [714](#)  
gitlab-project-push-rules-update  
command line option, [716](#)

--membership  
gitlab-project-fork-list command line  
option, [619](#)  
gitlab-project-list command line option,  
[566](#)  
gitlab-starred-project-list command  
line option, [756](#)  
gitlab-user-project-list command line  
option, [778](#)

--membership-lock  
gitlab-group-create command line option,  
[484](#)  
gitlab-group-update command line option,  
[488](#)

--mentions-disabled  
gitlab-group-create command line option,  
[484](#)  
gitlab-group-update command line option,  
[488](#)

--merge-access-level  
gitlab-project-protected-branch-create  
command line option, [709](#)

--merge-commit-message  
gitlab-project-merge-request-merge  
command line option, [665](#)

--merge-merge-request  
gitlab-group-notification-settings-update  
command line option, [538](#)  
gitlab-notification-settings-update  
command line option, [557](#)  
gitlab-project-notification-settings-update  
command line option, [694](#)

--merge-method  
gitlab-project-create command line  
option, [563](#)  
gitlab-project-update command line  
option, [574](#)

--merge-pipelines-enabled  
gitlab-project-create command line  
option, [563](#)  
gitlab-project-update command line  
option, [574](#)

--merge-request-iid  
gitlab-project-merge-request-approval-rule-create  
command line option, [671](#)  
gitlab-project-merge-request-approval-rule-update  
command line option, [673](#)

--merge-request-to-resolve-discussions-of  
gitlab-project-issue-create command  
line option, [628](#)

--merge-requests-access-level  
gitlab-project-create command line  
option, [563](#)  
gitlab-project-update command line  
option, [574](#)

--merge-requests-author-approval  
gitlab-project-approval-update command  
line option, [580](#)

--merge-requests-disable-committers-approval  
gitlab-project-approval-update command  
line option, [580](#)

--merge-requests-enabled  
gitlab-project-create command line  
option, [563](#)  
gitlab-project-update command line  
option, [574](#)  
gitlab-user-project-create command line  
option, [777](#)

--merge-requests-events  
gitlab-group-hook-create command line  
option, [519](#)  
gitlab-group-hook-update command line  
option, [521](#)  
gitlab-project-hook-create command line  
option, [620](#)  
gitlab-project-hook-update command line  
option, [622](#)

--merge-requests-template  
gitlab-project-update command line  
option, [575](#)

--merge-when-pipeline-succeeds  
gitlab-project-merge-request-merge  
command line option, [665](#)

--message  
gitlab-broadcast-message-create command  
line option, [467](#)  
gitlab-broadcast-message-update command  
line option, [468](#)  
gitlab-current-user-status-update  
command line option, [474](#)  
gitlab-project-tag-create command line  
option, [740](#)

--message-background-color  
gitlab-application-appearance-update

---

```

        command line option, 461
--message-font-color
    gitlab-application-appearance-update
        command line option, 461
--milestone
    gitlab-group-issue-list command line
        option, 525
    gitlab-group-merge-request-list command
        line option, 533
    gitlab-issue-list command line option,
        550
    gitlab-merge-request-list command line
        option, 554
    gitlab-project-deployment-merge-request-list
        command line option, 610
    gitlab-project-issue-list command line
        option, 630
    gitlab-project-merge-request-list
        command line option, 664
--milestone-id
    gitlab-project-issue-create command
        line option, 628
    gitlab-project-issue-update command
        line option, 634
    gitlab-project-merge-request-create
        command line option, 663
    gitlab-project-merge-request-update
        command line option, 669
--milestones
    gitlab-project-release-update command
        line option, 720
--min-access-level
    gitlab-group-descendant-group-list
        command line option, 505
    gitlab-group-list command line option,
        486
    gitlab-group-project-list command line
        option, 540
    gitlab-group-subgroup-list command line
        option, 544
    gitlab-project-list command line option,
        566
    gitlab-shared-project-list command line
        option, 753
    gitlab-starred-project-list command
        line option, 757
    gitlab-user-project-list command line
        option, 779
--mirror
    gitlab-project-create command line
        option, 563
    gitlab-project-update command line
        option, 574
--mirror-overwrites-diverged-branches
    gitlab-project-update command line
        option, 574
--mirror-trigger-builds
    gitlab-project-create command line
        option, 563
    gitlab-project-update command line
        option, 574
--mirror-user-id
    gitlab-project-update command line
        option, 574
--move-after-id
    gitlab-group-epic-issue-update command
        line option, 513
    gitlab-project-issue-reorder command
        line option, 631
--move-before-id
    gitlab-group-epic-issue-update command
        line option, 513
    gitlab-project-issue-reorder command
        line option, 631
--mr-iid
    gitlab-project-merge-request-approval-get
        command line option, 670
    gitlab-project-merge-request-approval-rule-create
        command line option, 671
    gitlab-project-merge-request-approval-rule-delete
        command line option, 672
    gitlab-project-merge-request-approval-rule-get
        command line option, 672
    gitlab-project-merge-request-approval-rule-list
        command line option, 672
    gitlab-project-merge-request-approval-rule-update
        command line option, 673
    gitlab-project-merge-request-approval-state-get
        command line option, 674
    gitlab-project-merge-request-approval-update
        command line option, 670
    gitlab-project-merge-request-award-emoji-create
        command line option, 674
    gitlab-project-merge-request-award-emoji-delete
        command line option, 675
    gitlab-project-merge-request-award-emoji-get
        command line option, 675
    gitlab-project-merge-request-award-emoji-list
        command line option, 675
    gitlab-project-merge-request-diff-get
        command line option, 676
    gitlab-project-merge-request-diff-list
        command line option, 676
    gitlab-project-merge-request-discussion-create
        command line option, 677
    gitlab-project-merge-request-discussion-get
        command line option, 678
    gitlab-project-merge-request-discussion-list

```



command line option, 678  
 gitlab-project-merge-request-discussion-note-create command line option, 679  
 gitlab-project-merge-request-discussion-note-delete command line option, 680  
 gitlab-project-merge-request-discussion-note-get command line option, 680  
 gitlab-project-merge-request-discussion-note-update command line option, 680  
 gitlab-project-merge-request-discussion-update command line option, 678  
 gitlab-project-merge-request-note-award-emoji command line option, 683  
 gitlab-project-merge-request-note-award-emoji-delete command line option, 684  
 gitlab-project-merge-request-note-award-emoji-get command line option, 684  
 gitlab-project-merge-request-note-award-emoji-list command line option, 684  
 gitlab-project-merge-request-note-create command line option, 681  
 gitlab-project-merge-request-note-delete command line option, 681  
 gitlab-project-merge-request-note-get command line option, 682  
 gitlab-project-merge-request-note-list command line option, 682  
 gitlab-project-merge-request-note-update command line option, 683  
 gitlab-project-merge-request-pipeline-create command line option, 685  
 gitlab-project-merge-request-pipeline-list command line option, 685  
 gitlab-project-merge-request-resource-label-event-get command line option, 686  
 gitlab-project-merge-request-resource-label-event-list command line option, 686  
 gitlab-project-merge-request-resource-milestone-event-get command line option, 687  
 gitlab-project-merge-request-resource-milestone-event-list command line option, 687  
 gitlab-project-merge-request-resource-state-event-get command line option, 688  
 gitlab-project-merge-request-resource-state-event-list command line option, 688  
 --my-reaction-emoji  
 gitlab-group-issue-list command line option, 525  
 gitlab-group-merge-request-list command line option, 533  
 gitlab-issue-list command line option, 550  
 gitlab-merge-request-list command line option, 554  
 gitlab-project-deployment-merge-request-list command line option, 610  
 gitlab-project-issue-list command line option, 630  
 gitlab-project-merge-request-list command line option, 665  
 --name  
 gitlab-application-create command line option, 459  
 gitlab-dockerfile-get command line option, 475  
 gitlab-feature-delete command line option, 477  
 gitlab-gitignore-get command line option, 481  
 gitlab-gitlabciyaml-get command line option, 482  
 gitlab-group-access-token-create command line option, 490  
 gitlab-group-board-create command line option, 497  
 gitlab-group-cluster-create command line option, 500  
 gitlab-group-cluster-update command line option, 501  
 gitlab-group-create command line option, 484  
 gitlab-group-deploy-token-create command line option, 503  
 gitlab-group-epic-award-emoji-create command line option, 508  
 gitlab-group-epic-note-award-emoji-create command line option, 515  
 gitlab-group-label-create command line option, 527  
 gitlab-group-label-delete command line option, 527  
 gitlab-group-label-get command line option, 528  
 gitlab-group-label-list command line option, 528  
 gitlab-group-label-subscribe command line option, 529  
 gitlab-group-label-unsubscribe command line option, 529  
 gitlab-group-label-update command line option, 529  
 gitlab-group-update command line option, 488  
 gitlab-project-access-token-create command line option, 577  
 gitlab-project-approval-rule-create command line option, 580  
 gitlab-project-board-create command line option, 586  
 gitlab-project-branch-delete command



- line option, [590](#)
- gitlab-project-branch-get command line option, [590](#)
- gitlab-project-cluster-create command line option, [593](#)
- gitlab-project-cluster-update command line option, [594](#)
- gitlab-project-commit-status-create command line option, [603](#)
- gitlab-project-create command line option, [562](#)
- gitlab-project-deploy-token-create command line option, [605](#)
- gitlab-project-environment-create command line option, [611](#)
- gitlab-project-environment-list command line option, [612](#)
- gitlab-project-environment-update command line option, [613](#)
- gitlab-project-issue-award-emoji-create command line option, [635](#)
- gitlab-project-issue-note-award-emoji-create command line option, [643](#)
- gitlab-project-label-create command line option, [655](#)
- gitlab-project-label-delete command line option, [655](#)
- gitlab-project-label-get command line option, [655](#)
- gitlab-project-label-subscribe command line option, [656](#)
- gitlab-project-label-unsubscribe command line option, [656](#)
- gitlab-project-label-update command line option, [657](#)
- gitlab-project-merge-request-approval-rule-create command line option, [671](#)
- gitlab-project-merge-request-approval-rule-update command line option, [673](#)
- gitlab-project-merge-request-award-emoji-create command line option, [674](#)
- gitlab-project-merge-request-note-award-emoji-create command line option, [683](#)
- gitlab-project-pipeline-list command line option, [700](#)
- gitlab-project-protected-branch-create command line option, [709](#)
- gitlab-project-protected-branch-delete command line option, [709](#)
- gitlab-project-protected-branch-get command line option, [709](#)
- gitlab-project-protected-environment-create command line option, [710](#)
- gitlab-project-protected-environment-delete-namespace command line option, [711](#)
- gitlab-project-protected-environment-get command line option, [711](#)
- gitlab-project-protected-tag-create command line option, [712](#)
- gitlab-project-protected-tag-delete command line option, [712](#)
- gitlab-project-protected-tag-get command line option, [713](#)
- gitlab-project-registry-tag-delete command line option, [717](#)
- gitlab-project-registry-tag-delete-in-bulk command line option, [717](#)
- gitlab-project-registry-tag-get command line option, [718](#)
- gitlab-project-release-create command line option, [719](#)
- gitlab-project-release-link-create command line option, [721](#)
- gitlab-project-release-link-update command line option, [723](#)
- gitlab-project-release-update command line option, [720](#)
- gitlab-project-snippet-award-emoji-create command line option, [730](#)
- gitlab-project-snippet-note-award-emoji-create command line option, [738](#)
- gitlab-project-tag-delete command line option, [740](#)
- gitlab-project-tag-get command line option, [740](#)
- gitlab-project-update command line option, [574](#)
- gitlab-topic-create command line option, [759](#)
- gitlab-topic-update command line option, [760](#)
- gitlab-user-create command line option, [762](#)
- gitlab-user-impersonation-token-create command line option, [772](#)
- gitlab-user-personal-access-token-create command line option, [776](#)
- gitlab-user-project-create command line option, [777](#)
- gitlab-user-update command line option, [766](#)
- name-regex-delete
  - gitlab-project-registry-tag-delete-in-bulk command line option, [717](#)
- name-regex-keep
  - gitlab-project-registry-tag-delete-in-bulk command line option, [717](#)

```
gitlab-namespace-exists command line
  option, 555
gitlab-project-fork-create command line
  option, 618
--namespace-id
  gitlab-project-create command line
    option, 563
--new-issue
  gitlab-group-notification-settings-update
    command line option, 538
  gitlab-notification-settings-update
    command line option, 557
  gitlab-project-notification-settings-update
    command line option, 694
--new-merge-request
  gitlab-group-notification-settings-update
    command line option, 538
  gitlab-notification-settings-update
    command line option, 557
  gitlab-project-notification-settings-update
    command line option, 694
--new-name
  gitlab-group-label-update command line
    option, 529
  gitlab-project-label-update command
    line option, 657
--new-note
  gitlab-group-notification-settings-update
    command line option, 538
  gitlab-notification-settings-update
    command line option, 557
  gitlab-project-notification-settings-update
    command line option, 694
--new-project-guidelines
  gitlab-application-appearance-update
    command line option, 461
--not
  gitlab-merge-request-list command line
    option, 555
  gitlab-project-deployment-merge-request-list
    command line option, 610
--note
  gitlab-project-commit-comment-create
    command line option, 599
--note-events
  gitlab-group-hook-create command line
    option, 519
  gitlab-group-hook-update command line
    option, 521
  gitlab-project-hook-create command line
    option, 620
  gitlab-project-hook-update command line
    option, 622
--note-id
  gitlab-group-epic-note-award-emoji-create
    command line option, 515
  gitlab-group-epic-note-award-emoji-delete
    command line option, 516
  gitlab-group-epic-note-award-emoji-get
    command line option, 516
  gitlab-group-epic-note-award-emoji-list
    command line option, 516
  gitlab-project-issue-note-award-emoji-create
    command line option, 643
  gitlab-project-issue-note-award-emoji-delete
    command line option, 644
  gitlab-project-issue-note-award-emoji-get
    command line option, 644
  gitlab-project-issue-note-award-emoji-list
    command line option, 645
  gitlab-project-merge-request-note-award-emoji-create
    command line option, 683
  gitlab-project-merge-request-note-award-emoji-delete
    command line option, 684
  gitlab-project-merge-request-note-award-emoji-get
    command line option, 684
  gitlab-project-merge-request-note-award-emoji-list
    command line option, 684
  gitlab-project-snippet-note-award-emoji-create
    command line option, 737
  gitlab-project-snippet-note-award-emoji-delete
    command line option, 738
  gitlab-project-snippet-note-award-emoji-get
    command line option, 738
  gitlab-project-snippet-note-award-emoji-list
    command line option, 739
--notification-email
  gitlab-group-notification-settings-update
    command line option, 538
  gitlab-notification-settings-update
    command line option, 557
  gitlab-project-notification-settings-update
    command line option, 694
--oauth-token
  gitlab command line option, 459
--older-than
  gitlab-project-registry-tag-delete-in-bulk
    command line option, 717
--only-allow-merge-if-all-discussions-are-resolved
  gitlab-project-create command line
    option, 563
  gitlab-project-update command line
    option, 574
--only-allow-merge-if-build-succeeds
  gitlab-user-project-create command line
    option, 777
--only-allow-merge-if-pipeline-succeeds
  gitlab-project-create command line
```

- option, [563](#)
- gitlab-project-update command line option, [574](#)
- only-mirror-protected-branches
  - gitlab-project-update command line option, [574](#)
- only-protected-branches
  - gitlab-project-remote-mirror-create command line option, [723](#)
  - gitlab-project-remote-mirror-update command line option, [724](#)
- operations-access-level
  - gitlab-project-create command line option, [563](#)
  - gitlab-project-update command line option, [574](#)
- order
  - gitlab-project-commit-list command line option, [597](#)
- order-by
  - gitlab command line option, [459](#)
  - gitlab-group-descendant-group-list command line option, [505](#)
  - gitlab-group-epic-list command line option, [507](#)
  - gitlab-group-issue-list command line option, [525](#)
  - gitlab-group-list command line option, [486](#)
  - gitlab-group-merge-request-list command line option, [533](#)
  - gitlab-group-package-list command line option, [538](#)
  - gitlab-group-project-list command line option, [539](#)
  - gitlab-group-subgroup-list command line option, [544](#)
  - gitlab-issue-list command line option, [550](#)
  - gitlab-merge-request-list command line option, [554](#)
  - gitlab-project-deployment-list command line option, [608](#)
  - gitlab-project-deployment-merge-request-list command line option, [610](#)
  - gitlab-project-fork-list command line option, [619](#)
  - gitlab-project-issue-list command line option, [630](#)
  - gitlab-project-list command line option, [566](#)
  - gitlab-project-merge-request-list command line option, [664](#)
  - gitlab-project-package-list command line option, [695](#)
  - gitlab-project-pipeline-list command line option, [700](#)
  - gitlab-project-release-list command line option, [720](#)
  - gitlab-shared-project-list command line option, [753](#)
  - gitlab-starred-project-list command line option, [757](#)
  - gitlab-user-project-list command line option, [778](#)
- organization
  - gitlab-user-create command line option, [763](#)
  - gitlab-user-update command line option, [766](#)
- outbound-local-requests-whitelist
  - gitlab-application-settings-update command line option, [464](#)
- output
  - gitlab command line option, [458](#)
- owned
  - gitlab-group-descendant-group-list command line option, [505](#)
  - gitlab-group-list command line option, [486](#)
  - gitlab-group-project-list command line option, [540](#)
  - gitlab-group-subgroup-list command line option, [544](#)
  - gitlab-project-fork-list command line option, [619](#)
  - gitlab-project-list command line option, [566](#)
  - gitlab-starred-project-list command line option, [757](#)
  - gitlab-user-project-list command line option, [778](#)
- package-id
  - gitlab-project-package-file-delete command line option, [696](#)
  - gitlab-project-package-file-list command line option, [696](#)
- package-name
  - gitlab-generic-package-download command line option, [478](#)
  - gitlab-generic-package-upload command line option, [478](#)
- package-type
  - gitlab-group-package-list command line option, [539](#)
  - gitlab-project-package-list command line option, [695](#)

- option, [538](#)
- gitlab-project-package-list command line option, [695](#)
- package-version
  - gitlab-generic-package-download command line option, [478](#)
  - gitlab-generic-package-upload command line option, [478](#)
- packages-enabled
  - gitlab-project-create command line option, [563](#)
  - gitlab-project-update command line option, [574](#)
- page
  - gitlab-application-list command line option, [460](#)
  - gitlab-audit-event-list command line option, [466](#)
  - gitlab-broadcast-message-list command line option, [468](#)
  - gitlab-current-user-email-list command line option, [471](#)
  - gitlab-current-user-gpg-key-list command line option, [472](#)
  - gitlab-current-user-key-list command line option, [473](#)
  - gitlab-deploy-key-list command line option, [474](#)
  - gitlab-deploy-token-list command line option, [475](#)
  - gitlab-dockerfile-list command line option, [476](#)
  - gitlab-event-list command line option, [476](#)
  - gitlab-feature-list command line option, [477](#)
  - gitlab-geo-node-list command line option, [480](#)
  - gitlab-gitignore-list command line option, [481](#)
  - gitlab-gitlabciyaml-list command line option, [482](#)
  - gitlab-group-access-request-list command line option, [490](#)
  - gitlab-group-access-token-list command line option, [491](#)
  - gitlab-group-audit-event-list command line option, [492](#)
  - gitlab-group-badge-list command line option, [494](#)
  - gitlab-group-billable-member-list command line option, [495](#)
  - gitlab-group-billable-member-membership-list command line option, [496](#)
  - gitlab-group-board-list command line option, [497](#)
  - gitlab-group-board-list-list command line option, [499](#)
  - gitlab-group-cluster-list command line option, [501](#)
  - gitlab-group-custom-attribute-list command line option, [503](#)
  - gitlab-group-deploy-token-list command line option, [504](#)
  - gitlab-group-descendant-group-list command line option, [505](#)
  - gitlab-group-epic-award-emoji-list command line option, [509](#)
  - gitlab-group-epic-issue-list command line option, [513](#)
  - gitlab-group-epic-list command line option, [507](#)
  - gitlab-group-epic-note-award-emoji-list command line option, [517](#)
  - gitlab-group-epic-note-list command line option, [515](#)
  - gitlab-group-epic-resource-label-event-list command line option, [518](#)
  - gitlab-group-hook-list command line option, [520](#)
  - gitlab-group-invitation-list command line option, [523](#)
  - gitlab-group-issue-list command line option, [525](#)
  - gitlab-group-iteration-list command line option, [526](#)
  - gitlab-group-label-list command line option, [528](#)
  - gitlab-group-list command line option, [486](#)
  - gitlab-group-member-all-list command line option, [532](#)
  - gitlab-group-member-list command line option, [531](#)
  - gitlab-group-merge-request-list command line option, [534](#)
  - gitlab-group-milestone-list command line option, [536](#)
  - gitlab-group-package-list command line option, [539](#)
  - gitlab-group-project-list command line option, [540](#)
  - gitlab-group-runner-list command line option, [543](#)
  - gitlab-group-subgroup-list command line option, [544](#)
  - gitlab-group-variable-list command line option, [545](#)

gitlab-group-wiki-list command line option, [547](#)  
gitlab-hook-list command line option, [549](#)  
gitlab-issue-list command line option, [550](#)  
gitlab-ldap-group-list command line option, [552](#)  
gitlab-license-list command line option, [553](#)  
gitlab-merge-request-list command line option, [555](#)  
gitlab-namespace-list command line option, [556](#)  
gitlab-pages-domain-list command line option, [557](#)  
gitlab-personal-access-token-list command line option, [558](#)  
gitlab-project-access-request-list command line option, [577](#)  
gitlab-project-access-token-list command line option, [578](#)  
gitlab-project-approval-rule-list command line option, [581](#)  
gitlab-project-audit-event-list command line option, [583](#)  
gitlab-project-badge-list command line option, [585](#)  
gitlab-project-board-list command line option, [587](#)  
gitlab-project-board-list-list command line option, [589](#)  
gitlab-project-branch-list command line option, [591](#)  
gitlab-project-cluster-list command line option, [594](#)  
gitlab-project-commit-comment-list command line option, [599](#)  
gitlab-project-commit-discussion-list command line option, [600](#)  
gitlab-project-commit-list command line option, [597](#)  
gitlab-project-commit-status-list command line option, [604](#)  
gitlab-project-custom-attribute-list command line option, [605](#)  
gitlab-project-deploy-token-list command line option, [606](#)  
gitlab-project-deployment-list command line option, [608](#)  
gitlab-project-deployment-merge-request-list command line option, [611](#)  
gitlab-project-environment-list command line option, [612](#)  
gitlab-project-event-list command line option, [614](#)  
gitlab-project-fork-list command line option, [619](#)  
gitlab-project-hook-list command line option, [621](#)  
gitlab-project-integration-list command line option, [624](#)  
gitlab-project-invitation-list command line option, [626](#)  
gitlab-project-issue-award-emoji-list command line option, [636](#)  
gitlab-project-issue-discussion-list command line option, [637](#)  
gitlab-project-issue-link-list command line option, [641](#)  
gitlab-project-issue-list command line option, [630](#)  
gitlab-project-issue-note-award-emoji-list command line option, [645](#)  
gitlab-project-issue-note-list command line option, [642](#)  
gitlab-project-issue-resource-label-event-list command line option, [646](#)  
gitlab-project-issue-resource-milestone-event-list command line option, [647](#)  
gitlab-project-issue-resource-state-event-list command line option, [648](#)  
gitlab-project-job-list command line option, [651](#)  
gitlab-project-key-list command line option, [654](#)  
gitlab-project-label-list command line option, [656](#)  
gitlab-project-list command line option, [566](#)  
gitlab-project-member-all-list command line option, [660](#)  
gitlab-project-member-list command line option, [658](#)  
gitlab-project-merge-request-approval-rule-list command line option, [672](#)  
gitlab-project-merge-request-award-emoji-list command line option, [675](#)  
gitlab-project-merge-request-diff-list command line option, [676](#)  
gitlab-project-merge-request-discussion-list command line option, [678](#)  
gitlab-project-merge-request-list command line option, [665](#)  
gitlab-project-merge-request-note-award-emoji-list command line option, [684](#)  
gitlab-project-merge-request-note-list command line option, [682](#)  
gitlab-project-merge-request-pipeline-list

command line option, [685](#)  
gitlab-project-merge-request-resource-label-event-list  
command line option, [686](#)  
gitlab-project-merge-request-resource-milestone-event-list  
command line option, [687](#)  
gitlab-project-merge-request-resource-state-event-list  
command line option, [688](#)  
gitlab-project-merge-train-list command  
line option, [689](#)  
gitlab-project-milestone-list command  
line option, [691](#)  
gitlab-project-note-list command line  
option, [693](#)  
gitlab-project-package-file-list  
command line option, [696](#)  
gitlab-project-package-list command  
line option, [695](#)  
gitlab-project-pages-domain-list  
command line option, [698](#)  
gitlab-project-pipeline-bridge-list  
command line option, [701](#)  
gitlab-project-pipeline-job-list  
command line option, [702](#)  
gitlab-project-pipeline-list command  
line option, [700](#)  
gitlab-project-pipeline-schedule-list  
command line option, [704](#)  
gitlab-project-pipeline-variable-list  
command line option, [708](#)  
gitlab-project-protected-branch-list  
command line option, [710](#)  
gitlab-project-protected-environment-list  
command line option, [711](#)  
gitlab-project-protected-tag-list  
command line option, [713](#)  
gitlab-project-registry-repository-list  
command line option, [716](#)  
gitlab-project-registry-tag-list  
command line option, [718](#)  
gitlab-project-release-link-list  
command line option, [722](#)  
gitlab-project-release-list command  
line option, [720](#)  
gitlab-project-remote-mirror-list  
command line option, [724](#)  
gitlab-project-runner-list command line  
option, [725](#)  
gitlab-project-service-list command  
line option, [727](#)  
gitlab-project-snippet-award-emoji-list  
command line option, [731](#)  
gitlab-project-snippet-discussion-list  
command line option, [733](#)  
gitlab-project-snippet-list command  
line option, [729](#)  
gitlab-project-snippet-note-award-emoji-list  
command line option, [739](#)  
gitlab-project-snippet-note-list  
command line option, [736](#)  
gitlab-project-tag-list command line  
option, [741](#)  
gitlab-project-trigger-list command  
line option, [742](#)  
gitlab-project-user-list command line  
option, [743](#)  
gitlab-project-variable-list command  
line option, [745](#)  
gitlab-project-wiki-list command line  
option, [747](#)  
gitlab-runner-all-list command line  
option, [751](#)  
gitlab-runner-job-list command line  
option, [752](#)  
gitlab-runner-list command line option,  
[750](#)  
gitlab-shared-project-list command line  
option, [753](#)  
gitlab-snippet-list command line option,  
[755](#)  
gitlab-starred-project-list command  
line option, [757](#)  
gitlab-todo-list command line option, [758](#)  
gitlab-topic-list command line option,  
[760](#)  
gitlab-user-activities-list command  
line option, [767](#)  
gitlab-user-custom-attribute-list  
command line option, [768](#)  
gitlab-user-email-list command line  
option, [769](#)  
gitlab-user-event-list command line  
option, [770](#)  
gitlab-user-gpg-key-list command line  
option, [772](#)  
gitlab-user-impersonation-token-list  
command line option, [773](#)  
gitlab-user-key-list command line  
option, [775](#)  
gitlab-user-list command line option, [764](#)  
gitlab-user-membership-list command  
line option, [776](#)  
gitlab-user-project-list command line  
option, [779](#)  
gitlab-variable-list command line  
option, [781](#)  
--pages-access-level  
gitlab-project-create command line  
option, [563](#)



- gitlab-project-update command line option, [574](#)
- pagination
  - gitlab command line option, [459](#)
- parent-id
  - gitlab-group-create command line option, [484](#)
  - gitlab-namespace-exists command line option, [555](#)
- password
  - gitlab-user-create command line option, [762](#)
  - gitlab-user-update command line option, [766](#)
- password-authentication-enabled-for-web
  - gitlab-application-settings-update command line option, [464](#)
- path
  - gitlab-generic-package-upload command line option, [478](#)
  - gitlab-group-create command line option, [484](#)
  - gitlab-group-update command line option, [488](#)
  - gitlab-project-commit-comment-create command line option, [599](#)
  - gitlab-project-commit-list command line option, [597](#)
  - gitlab-project-create command line option, [562](#)
  - gitlab-project-repository-tree command line option, [568](#)
  - gitlab-project-update command line option, [574](#)
- paused
  - gitlab-runner-all-list command line option, [751](#)
  - gitlab-runner-list command line option, [750](#)
- per-page
  - gitlab command line option, [459](#)
  - gitlab-application-list command line option, [460](#)
  - gitlab-audit-event-list command line option, [466](#)
  - gitlab-broadcast-message-list command line option, [468](#)
  - gitlab-current-user-email-list command line option, [471](#)
  - gitlab-current-user-gpg-key-list command line option, [472](#)
  - gitlab-current-user-key-list command line option, [473](#)
  - gitlab-deploy-key-list command line option, [474](#)
  - gitlab-deploy-token-list command line option, [475](#)
  - gitlab-dockerfile-list command line option, [476](#)
  - gitlab-event-list command line option, [476](#)
  - gitlab-feature-list command line option, [477](#)
  - gitlab-geo-node-list command line option, [480](#)
  - gitlab-gitignore-list command line option, [481](#)
  - gitlab-gitlabciyaml-list command line option, [482](#)
  - gitlab-group-access-request-list command line option, [490](#)
  - gitlab-group-access-token-list command line option, [491](#)
  - gitlab-group-audit-event-list command line option, [492](#)
  - gitlab-group-badge-list command line option, [494](#)
  - gitlab-group-billable-member-list command line option, [495](#)
  - gitlab-group-billable-member-membership-list command line option, [496](#)
  - gitlab-group-board-list command line option, [497](#)
  - gitlab-group-board-list-list command line option, [499](#)
  - gitlab-group-cluster-list command line option, [501](#)
  - gitlab-group-custom-attribute-list command line option, [503](#)
  - gitlab-group-deploy-token-list command line option, [504](#)
  - gitlab-group-descendant-group-list command line option, [505](#)
  - gitlab-group-epic-award-emoji-list command line option, [509](#)
  - gitlab-group-epic-issue-list command line option, [513](#)
  - gitlab-group-epic-list command line option, [507](#)
  - gitlab-group-epic-note-award-emoji-list command line option, [517](#)
  - gitlab-group-epic-note-list command line option, [515](#)
  - gitlab-group-epic-resource-label-event-list command line option, [518](#)
  - gitlab-group-hook-list command line option, [520](#)
  - gitlab-group-invitation-list command

- line option, [524](#)
- gitlab-group-issue-list command line option, [525](#)
- gitlab-group-iteration-list command line option, [526](#)
- gitlab-group-label-list command line option, [528](#)
- gitlab-group-list command line option, [486](#)
- gitlab-group-member-all-list command line option, [532](#)
- gitlab-group-member-list command line option, [531](#)
- gitlab-group-merge-request-list command line option, [534](#)
- gitlab-group-milestone-list command line option, [536](#)
- gitlab-group-package-list command line option, [539](#)
- gitlab-group-project-list command line option, [540](#)
- gitlab-group-runner-list command line option, [543](#)
- gitlab-group-subgroup-list command line option, [544](#)
- gitlab-group-variable-list command line option, [545](#)
- gitlab-group-wiki-list command line option, [547](#)
- gitlab-hook-list command line option, [549](#)
- gitlab-issue-list command line option, [550](#)
- gitlab-ldap-group-list command line option, [552](#)
- gitlab-license-list command line option, [553](#)
- gitlab-merge-request-list command line option, [555](#)
- gitlab-namespace-list command line option, [556](#)
- gitlab-pages-domain-list command line option, [557](#)
- gitlab-personal-access-token-list command line option, [558](#)
- gitlab-project-access-request-list command line option, [577](#)
- gitlab-project-access-token-list command line option, [578](#)
- gitlab-project-approval-rule-list command line option, [581](#)
- gitlab-project-audit-event-list command line option, [583](#)
- gitlab-project-badge-list command line option, [585](#)
- gitlab-project-board-list command line option, [587](#)
- gitlab-project-board-list-list command line option, [589](#)
- gitlab-project-branch-list command line option, [591](#)
- gitlab-project-cluster-list command line option, [594](#)
- gitlab-project-commit-comment-list command line option, [599](#)
- gitlab-project-commit-discussion-list command line option, [600](#)
- gitlab-project-commit-list command line option, [597](#)
- gitlab-project-commit-status-list command line option, [604](#)
- gitlab-project-custom-attribute-list command line option, [605](#)
- gitlab-project-deploy-token-list command line option, [606](#)
- gitlab-project-deployment-list command line option, [608](#)
- gitlab-project-deployment-merge-request-list command line option, [611](#)
- gitlab-project-environment-list command line option, [612](#)
- gitlab-project-event-list command line option, [614](#)
- gitlab-project-fork-list command line option, [619](#)
- gitlab-project-hook-list command line option, [621](#)
- gitlab-project-integration-list command line option, [624](#)
- gitlab-project-invitation-list command line option, [626](#)
- gitlab-project-issue-award-emoji-list command line option, [636](#)
- gitlab-project-issue-discussion-list command line option, [638](#)
- gitlab-project-issue-link-list command line option, [641](#)
- gitlab-project-issue-list command line option, [630](#)
- gitlab-project-issue-note-award-emoji-list command line option, [645](#)
- gitlab-project-issue-note-list command line option, [642](#)
- gitlab-project-issue-resource-label-event-list command line option, [646](#)
- gitlab-project-issue-resource-milestone-event-list command line option, [647](#)
- gitlab-project-issue-resource-state-event-list command line option, [648](#)



- gitlab-project-job-list command line option, [651](#)
- gitlab-project-key-list command line option, [654](#)
- gitlab-project-label-list command line option, [656](#)
- gitlab-project-list command line option, [566](#)
- gitlab-project-member-all-list command line option, [660](#)
- gitlab-project-member-list command line option, [658](#)
- gitlab-project-merge-request-approval-rule-list command line option, [672](#)
- gitlab-project-merge-request-award-emoji-list command line option, [675](#)
- gitlab-project-merge-request-diff-list command line option, [676](#)
- gitlab-project-merge-request-discussion-list command line option, [678](#)
- gitlab-project-merge-request-list command line option, [665](#)
- gitlab-project-merge-request-note-award-emoji-list command line option, [684](#)
- gitlab-project-merge-request-note-list command line option, [682](#)
- gitlab-project-merge-request-pipeline-list command line option, [685](#)
- gitlab-project-merge-request-resource-label-event-list command line option, [686](#)
- gitlab-project-merge-request-resource-milestone-event-list command line option, [687](#)
- gitlab-project-merge-request-resource-state-event-list command line option, [688](#)
- gitlab-project-merge-train-list command line option, [689](#)
- gitlab-project-milestone-list command line option, [691](#)
- gitlab-project-note-list command line option, [693](#)
- gitlab-project-package-file-list command line option, [696](#)
- gitlab-project-package-list command line option, [695](#)
- gitlab-project-pages-domain-list command line option, [698](#)
- gitlab-project-pipeline-bridge-list command line option, [701](#)
- gitlab-project-pipeline-job-list command line option, [702](#)
- gitlab-project-pipeline-list command line option, [700](#)
- gitlab-project-pipeline-schedule-list command line option, [704](#)
- gitlab-project-pipeline-variable-list command line option, [708](#)
- gitlab-project-protected-branch-list command line option, [710](#)
- gitlab-project-protected-environment-list command line option, [711](#)
- gitlab-project-protected-tag-list command line option, [713](#)
- gitlab-project-registry-repository-list command line option, [716](#)
- gitlab-project-registry-tag-list command line option, [718](#)
- gitlab-project-release-link-list command line option, [722](#)
- gitlab-project-release-list command line option, [720](#)
- gitlab-project-remote-mirror-list command line option, [724](#)
- gitlab-project-runner-list command line option, [725](#)
- gitlab-project-service-list command line option, [727](#)
- gitlab-project-snippet-award-emoji-list command line option, [731](#)
- gitlab-project-snippet-discussion-list command line option, [733](#)
- gitlab-project-snippet-list command line option, [729](#)
- gitlab-project-snippet-note-award-emoji-list command line option, [739](#)
- gitlab-project-snippet-note-list command line option, [736](#)
- gitlab-project-tag-list command line option, [741](#)
- gitlab-project-trigger-list command line option, [742](#)
- gitlab-project-user-list command line option, [743](#)
- gitlab-project-variable-list command line option, [745](#)
- gitlab-project-wiki-list command line option, [747](#)
- gitlab-runner-all-list command line option, [751](#)
- gitlab-runner-job-list command line option, [752](#)
- gitlab-runner-list command line option, [750](#)
- gitlab-shared-project-list command line option, [753](#)
- gitlab-snippet-list command line option, [755](#)
- gitlab-starred-project-list command line option, [757](#)

gitlab-todo-list command line option, [758](#)  
gitlab-topic-list command line option, [760](#)  
gitlab-user-activities-list command line option, [767](#)  
gitlab-user-custom-attribute-list command line option, [768](#)  
gitlab-user-email-list command line option, [770](#)  
gitlab-user-event-list command line option, [770](#)  
gitlab-user-gpg-key-list command line option, [772](#)  
gitlab-user-impersonation-token-list command line option, [773](#)  
gitlab-user-key-list command line option, [775](#)  
gitlab-user-list command line option, [764](#)  
gitlab-user-membership-list command line option, [776](#)  
gitlab-user-project-list command line option, [779](#)  
gitlab-variable-list command line option, [781](#)  
--performance-bar-allowed-group-id gitlab-application-settings-update command line option, [465](#)  
--pipeline-events  
gitlab-group-hook-create command line option, [519](#)  
gitlab-group-hook-update command line option, [521](#)  
gitlab-project-hook-create command line option, [620](#)  
gitlab-project-hook-update command line option, [622](#)  
--pipeline-id  
gitlab-project-pipeline-bridge-list command line option, [701](#)  
gitlab-project-pipeline-job-list command line option, [702](#)  
gitlab-project-pipeline-test-report-get command line option, [707](#)  
gitlab-project-pipeline-test-report-summary-get command line option, [707](#)  
gitlab-project-pipeline-variable-list command line option, [708](#)  
--pipeline-schedule-id  
gitlab-project-pipeline-schedule-variable-create command line option, [706](#)  
gitlab-project-pipeline-schedule-variable-delete command line option, [706](#)  
gitlab-project-pipeline-schedule-variable-update command line option, [706](#)  
--plantuml-enabled gitlab-application-settings-update command line option, [465](#)  
--plantuml-url gitlab-application-settings-update command line option, [465](#)  
--platform-kubernetes-attributes  
gitlab-group-cluster-create command line option, [500](#)  
gitlab-group-cluster-update command line option, [502](#)  
gitlab-project-cluster-create command line option, [593](#)  
gitlab-project-cluster-update command line option, [594](#)  
--polling-interval-multiplier gitlab-application-settings-update command line option, [465](#)  
--popular  
gitlab-license-list command line option, [553](#)  
--position  
gitlab-group-board-list-update command line option, [500](#)  
gitlab-project-board-list-update command line option, [589](#)  
gitlab-project-commit-discussion-note-create command line option, [601](#)  
gitlab-project-merge-request-discussion-create command line option, [677](#)  
--prevent-forking-outside-group  
gitlab-group-update command line option, [488](#)  
--prevent-secrets  
gitlab-group-push-rules-create command line option, [541](#)  
gitlab-group-push-rules-update command line option, [542](#)  
gitlab-project-push-rules-create command line option, [714](#)  
gitlab-project-push-rules-update command line option, [716](#)  
--printing-merge-request-link-enabled  
gitlab-project-create command line option, [563](#)  
--priority  
gitlab-group-label-create command line option, [527](#)  
gitlab-group-label-update command line option, [529](#)  
gitlab-project-label-create command line option, [655](#)  
gitlab-project-label-update command line option, [657](#)

- private-profile
  - gitlab-user-create command line option, [763](#)
  - gitlab-user-update command line option, [766](#)
- private-token
  - gitlab command line option, [459](#)
- project
  - gitlab-license-get command line option, [552](#)
- project-creation-level
  - gitlab-group-create command line option, [484](#)
  - gitlab-group-update command line option, [488](#)
- project-id
  - gitlab-generic-package-download command line option, [478](#)
  - gitlab-generic-package-upload command line option, [478](#)
  - gitlab-group-transfer-project command line option, [487](#)
  - gitlab-project-access-request-approve command line option, [576](#)
  - gitlab-project-access-request-create command line option, [576](#)
  - gitlab-project-access-request-delete command line option, [576](#)
  - gitlab-project-access-request-list command line option, [577](#)
  - gitlab-project-access-token-create command line option, [577](#)
  - gitlab-project-access-token-delete command line option, [578](#)
  - gitlab-project-access-token-list command line option, [578](#)
  - gitlab-project-additional-statistics-get command line option, [579](#)
  - gitlab-project-approval-get command line option, [579](#)
  - gitlab-project-approval-rule-create command line option, [580](#)
  - gitlab-project-approval-rule-delete command line option, [580](#)
  - gitlab-project-approval-rule-list command line option, [581](#)
  - gitlab-project-approval-rule-update command line option, [581](#)
  - gitlab-project-approval-update command line option, [579](#)
  - gitlab-project-artifact-delete command line option, [582](#)
  - gitlab-project-artifact-download command line option, [582](#)
  - gitlab-project-artifact-raw command line option, [582](#)
  - gitlab-project-audit-event-get command line option, [583](#)
  - gitlab-project-audit-event-list command line option, [583](#)
  - gitlab-project-badge-create command line option, [584](#)
  - gitlab-project-badge-delete command line option, [584](#)
  - gitlab-project-badge-get command line option, [584](#)
  - gitlab-project-badge-list command line option, [585](#)
  - gitlab-project-badge-render command line option, [585](#)
  - gitlab-project-badge-update command line option, [585](#)
  - gitlab-project-board-create command line option, [586](#)
  - gitlab-project-board-delete command line option, [586](#)
  - gitlab-project-board-get command line option, [586](#)
  - gitlab-project-board-list command line option, [587](#)
  - gitlab-project-board-list-create command line option, [588](#)
  - gitlab-project-board-list-delete command line option, [588](#)
  - gitlab-project-board-list-get command line option, [588](#)
  - gitlab-project-board-list-list command line option, [589](#)
  - gitlab-project-board-list-update command line option, [589](#)
  - gitlab-project-board-update command line option, [587](#)
  - gitlab-project-branch-create command line option, [590](#)
  - gitlab-project-branch-delete command line option, [590](#)
  - gitlab-project-branch-get command line option, [590](#)
  - gitlab-project-branch-list command line option, [591](#)
  - gitlab-project-ci-lint-create command line option, [591](#)
  - gitlab-project-ci-lint-get command line option, [592](#)
  - gitlab-project-ci-lint-validate command line option, [592](#)
  - gitlab-project-cluster-create command line option, [593](#)

`gitlab-project-cluster-delete` command line option, [593](#)  
`gitlab-project-cluster-get` command line option, [593](#)  
`gitlab-project-cluster-list` command line option, [594](#)  
`gitlab-project-cluster-update` command line option, [594](#)  
`gitlab-project-commit-cherry-pick` command line option, [595](#)  
`gitlab-project-commit-comment-create` command line option, [598](#)  
`gitlab-project-commit-comment-list` command line option, [599](#)  
`gitlab-project-commit-create` command line option, [595](#)  
`gitlab-project-commit-diff` command line option, [596](#)  
`gitlab-project-commit-discussion-create` command line option, [600](#)  
`gitlab-project-commit-discussion-get` command line option, [600](#)  
`gitlab-project-commit-discussion-list` command line option, [600](#)  
`gitlab-project-commit-discussion-note-create` command line option, [601](#)  
`gitlab-project-commit-discussion-note-delete` command line option, [601](#)  
`gitlab-project-commit-discussion-note-get` command line option, [602](#)  
`gitlab-project-commit-discussion-note-update` command line option, [602](#)  
`gitlab-project-commit-get` command line option, [596](#)  
`gitlab-project-commit-list` command line option, [596](#)  
`gitlab-project-commit-merge-requests` command line option, [597](#)  
`gitlab-project-commit-refs` command line option, [597](#)  
`gitlab-project-commit-revert` command line option, [598](#)  
`gitlab-project-commit-signature` command line option, [598](#)  
`gitlab-project-commit-status-create` command line option, [603](#)  
`gitlab-project-commit-status-list` command line option, [603](#)  
`gitlab-project-custom-attribute-delete` command line option, [604](#)  
`gitlab-project-custom-attribute-get` command line option, [604](#)  
`gitlab-project-custom-attribute-list` command line option, [605](#)  
`gitlab-project-deploy-token-create` command line option, [605](#)  
`gitlab-project-deploy-token-delete` command line option, [606](#)  
`gitlab-project-deploy-token-get` command line option, [606](#)  
`gitlab-project-deploy-token-list` command line option, [606](#)  
`gitlab-project-deployment-create` command line option, [607](#)  
`gitlab-project-deployment-get` command line option, [607](#)  
`gitlab-project-deployment-list` command line option, [608](#)  
`gitlab-project-deployment-merge-request-list` command line option, [610](#)  
`gitlab-project-deployment-update` command line option, [608](#)  
`gitlab-project-environment-create` command line option, [611](#)  
`gitlab-project-environment-delete` command line option, [612](#)  
`gitlab-project-environment-get` command line option, [612](#)  
`gitlab-project-environment-list` command line option, [612](#)  
`gitlab-project-environment-stop` command line option, [613](#)  
`gitlab-project-environment-update` command line option, [613](#)  
`gitlab-project-event-list` command line option, [614](#)  
`gitlab-project-export-create` command line option, [614](#)  
`gitlab-project-export-download` command line option, [615](#)  
`gitlab-project-export-get` command line option, [615](#)  
`gitlab-project-file-blame` command line option, [615](#)  
`gitlab-project-file-create` command line option, [616](#)  
`gitlab-project-file-delete` command line option, [616](#)  
`gitlab-project-file-get` command line option, [617](#)  
`gitlab-project-file-raw` command line option, [617](#)  
`gitlab-project-file-update` command line option, [617](#)  
`gitlab-project-fork-create` command line option, [618](#)  
`gitlab-project-fork-list` command line option, [618](#)

gitlab-project-hook-create command line option, [620](#)  
gitlab-project-hook-delete command line option, [620](#)  
gitlab-project-hook-get command line option, [621](#)  
gitlab-project-hook-list command line option, [621](#)  
gitlab-project-hook-update command line option, [622](#)  
gitlab-project-import-get command line option, [622](#)  
gitlab-project-integration-available command line option, [623](#)  
gitlab-project-integration-delete command line option, [623](#)  
gitlab-project-integration-get command line option, [623](#)  
gitlab-project-integration-list command line option, [624](#)  
gitlab-project-integration-update command line option, [624](#)  
gitlab-project-invitation-create command line option, [625](#)  
gitlab-project-invitation-delete command line option, [625](#)  
gitlab-project-invitation-get command line option, [625](#)  
gitlab-project-invitation-list command line option, [626](#)  
gitlab-project-invitation-update command line option, [626](#)  
gitlab-project-issue-add-spent-time command line option, [627](#)  
gitlab-project-issue-award-emoji-create command line option, [635](#)  
gitlab-project-issue-award-emoji-delete command line option, [635](#)  
gitlab-project-issue-award-emoji-get command line option, [636](#)  
gitlab-project-issue-award-emoji-list command line option, [636](#)  
gitlab-project-issue-closed-by command line option, [628](#)  
gitlab-project-issue-create command line option, [628](#)  
gitlab-project-issue-delete command line option, [629](#)  
gitlab-project-issue-discussion-create command line option, [637](#)  
gitlab-project-issue-discussion-get command line option, [637](#)  
gitlab-project-issue-discussion-list command line option, [637](#)  
gitlab-project-issue-discussion-note-create command line option, [638](#)  
gitlab-project-issue-discussion-note-delete command line option, [639](#)  
gitlab-project-issue-discussion-note-get command line option, [639](#)  
gitlab-project-issue-discussion-note-update command line option, [640](#)  
gitlab-project-issue-get command line option, [629](#)  
gitlab-project-issue-link-create command line option, [640](#)  
gitlab-project-issue-link-delete command line option, [640](#)  
gitlab-project-issue-link-list command line option, [641](#)  
gitlab-project-issue-list command line option, [629](#)  
gitlab-project-issue-move command line option, [630](#)  
gitlab-project-issue-note-award-emoji-create command line option, [643](#)  
gitlab-project-issue-note-award-emoji-delete command line option, [644](#)  
gitlab-project-issue-note-award-emoji-get command line option, [644](#)  
gitlab-project-issue-note-award-emoji-list command line option, [645](#)  
gitlab-project-issue-note-create command line option, [641](#)  
gitlab-project-issue-note-delete command line option, [642](#)  
gitlab-project-issue-note-get command line option, [642](#)  
gitlab-project-issue-note-list command line option, [642](#)  
gitlab-project-issue-note-update command line option, [643](#)  
gitlab-project-issue-participants command line option, [631](#)  
gitlab-project-issue-related-merge-requests command line option, [631](#)  
gitlab-project-issue-reorder command line option, [631](#)  
gitlab-project-issue-reset-spent-time command line option, [632](#)  
gitlab-project-issue-reset-time-estimate command line option, [632](#)  
gitlab-project-issue-resource-label-event-get command line option, [645](#)  
gitlab-project-issue-resource-label-event-list command line option, [646](#)  
gitlab-project-issue-resource-milestone-event-get command line option, [646](#)

`gitlab-project-issue-resource-milestone-event-get` `gitlab-project-key-update` command line option, [647](#)

`gitlab-project-issue-resource-state-event-get` `gitlab-project-label-create` command line option, [647](#)

`gitlab-project-issue-resource-state-event-list` `gitlab-project-label-delete` command line option, [648](#)

`gitlab-project-issue-subscribe` command line option, [632](#)

`gitlab-project-issue-time-estimate` `gitlab-project-label-get` command line option, [632](#)

`gitlab-project-issue-time-stats` command line option, [633](#)

`gitlab-project-issue-todo` command line option, [633](#)

`gitlab-project-issue-unsubscribe` `gitlab-project-label-list` command line option, [633](#)

`gitlab-project-issue-update` command line option, [634](#)

`gitlab-project-issue-user-agent-detail` `gitlab-project-label-subscribe` command line option, [634](#)

`gitlab-project-issues-statistics-get` `gitlab-project-label-unsubscribe` command line option, [634](#)

`gitlab-project-job-artifact` command line option, [649](#)

`gitlab-project-job-artifacts` command line option, [649](#)

`gitlab-project-job-cancel` command line option, [650](#)

`gitlab-project-job-delete-artifacts` `gitlab-project-label-update` command line option, [650](#)

`gitlab-project-job-erase` command line option, [650](#)

`gitlab-project-job-get` command line option, [650](#)

`gitlab-project-job-keep-artifacts` `gitlab-project-member-all-get` command line option, [651](#)

`gitlab-project-job-list` command line option, [651](#)

`gitlab-project-job-play` command line option, [651](#)

`gitlab-project-job-retry` command line option, [652](#)

`gitlab-project-job-trace` command line option, [652](#)

`gitlab-project-key-create` command line option, [652](#)

`gitlab-project-key-delete` command line option, [653](#)

`gitlab-project-key-enable` command line option, [653](#)

`gitlab-project-key-get` command line option, [653](#)

`gitlab-project-key-list` command line option, [654](#)

`gitlab-project-key-update` command line option, [654](#)

`gitlab-project-label-create` command line option, [655](#)

`gitlab-project-label-delete` command line option, [655](#)

`gitlab-project-label-get` command line option, [655](#)

`gitlab-project-label-list` command line option, [656](#)

`gitlab-project-label-subscribe` command line option, [656](#)

`gitlab-project-label-unsubscribe` command line option, [656](#)

`gitlab-project-label-update` command line option, [657](#)

`gitlab-project-member-all-get` command line option, [659](#)

`gitlab-project-member-all-list` command line option, [659](#)

`gitlab-project-member-create` command line option, [657](#)

`gitlab-project-member-delete` command line option, [658](#)

`gitlab-project-member-get` command line option, [658](#)

`gitlab-project-member-list` command line option, [658](#)

`gitlab-project-member-update` command line option, [659](#)

`gitlab-project-merge-request-add-spent-time` command line option, [660](#)

`gitlab-project-merge-request-approval-get` command line option, [670](#)

`gitlab-project-merge-request-approval-rule-create` command line option, [671](#)

`gitlab-project-merge-request-approval-rule-delete` command line option, [671](#)

`gitlab-project-merge-request-approval-rule-get` command line option, [672](#)

`gitlab-project-merge-request-approval-rule-list` command line option, [672](#)

`gitlab-project-merge-request-approval-rule-update` command line option, [673](#)

`gitlab-project-merge-request-approval-state-get` command line option, [674](#)

`gitlab-project-merge-request-approval-update` command line option, [670](#)

`gitlab-project-merge-request-approve` command line option, [661](#)

`gitlab-project-merge-request-award-emoji-create` command line option, [674](#)

`gitlab-project-merge-request-award-emoji-delete` command line option, [675](#)



gitlab-project-merge-request-award-emoji-get command line option, 675	gitlab-project-merge-request-note-delete command line option, 681
gitlab-project-merge-request-award-emoji-list command line option, 675	gitlab-project-merge-request-note-get command line option, 682
gitlab-project-merge-request-cancel-merge-when-pipeline-fails command line option, 661	gitlab-project-merge-request-note-list command line option, 682
gitlab-project-merge-request-changes command line option, 661	gitlab-project-merge-request-note-update command line option, 683
gitlab-project-merge-request-closes-issues command line option, 662	gitlab-project-merge-request-participants command line option, 666
gitlab-project-merge-request-commits command line option, 662	gitlab-project-merge-request-pipeline-create command line option, 685
gitlab-project-merge-request-create command line option, 662	gitlab-project-merge-request-pipeline-list command line option, 685
gitlab-project-merge-request-delete command line option, 663	gitlab-project-merge-request-rebase command line option, 666
gitlab-project-merge-request-diff-get command line option, 676	gitlab-project-merge-request-reset-spent-time command line option, 666
gitlab-project-merge-request-diff-list command line option, 676	gitlab-project-merge-request-reset-time-estimate command line option, 667
gitlab-project-merge-request-discussion-create command line option, 677	gitlab-project-merge-request-resource-label-event-get command line option, 686
gitlab-project-merge-request-discussion-get command line option, 677	gitlab-project-merge-request-resource-label-event-list command line option, 686
gitlab-project-merge-request-discussion-list command line option, 678	gitlab-project-merge-request-resource-milestone-event- command line option, 687
gitlab-project-merge-request-discussion-note-create command line option, 679	gitlab-project-merge-request-resource-milestone-event- command line option, 687
gitlab-project-merge-request-discussion-note-delete command line option, 679	gitlab-project-merge-request-resource-state-event-get command line option, 688
gitlab-project-merge-request-discussion-note-get command line option, 680	gitlab-project-merge-request-resource-state-event-list command line option, 688
gitlab-project-merge-request-discussion-note-update command line option, 680	gitlab-project-merge-request-subscribe command line option, 667
gitlab-project-merge-request-discussion-update command line option, 678	gitlab-project-merge-request-time-estimate command line option, 667
gitlab-project-merge-request-get command line option, 663	gitlab-project-merge-request-time-stats command line option, 668
gitlab-project-merge-request-list command line option, 664	gitlab-project-merge-request-todo command line option, 668
gitlab-project-merge-request-merge command line option, 665	gitlab-project-merge-request-unapprove command line option, 668
gitlab-project-merge-request-merge-ref command line option, 666	gitlab-project-merge-request-unsubscribe command line option, 668
gitlab-project-merge-request-note-award-emoji-create command line option, 683	gitlab-project-merge-request-update command line option, 669
gitlab-project-merge-request-note-award-emoji-delete command line option, 683	gitlab-project-merge-request-update command line option, 669
gitlab-project-merge-request-note-award-emoji-get command line option, 684	gitlab-project-merge-request-update command line option, 669
gitlab-project-merge-request-note-award-emoji-list command line option, 684	gitlab-project-milestone-create command line option, 690
gitlab-project-merge-request-note-award-emoji-update command line option, 684	gitlab-project-milestone-delete command line option, 690
gitlab-project-merge-request-note-create command line option, 681	gitlab-project-milestone-get command line option, 690

`gitlab-project-milestone-issues` command line option, [691](#)

`gitlab-project-milestone-list` command line option, [691](#)

`gitlab-project-milestone-merge-requests` command line option, [691](#)

`gitlab-project-milestone-update` command line option, [692](#)

`gitlab-project-note-get` command line option, [692](#)

`gitlab-project-note-list` command line option, [693](#)

`gitlab-project-notification-settings-get` command line option, [693](#)

`gitlab-project-notification-settings-update` command line option, [694](#)

`gitlab-project-package-delete` command line option, [695](#)

`gitlab-project-package-file-delete` command line option, [696](#)

`gitlab-project-package-file-list` command line option, [696](#)

`gitlab-project-package-get` command line option, [695](#)

`gitlab-project-package-list` command line option, [695](#)

`gitlab-project-pages-domain-create` command line option, [697](#)

`gitlab-project-pages-domain-delete` command line option, [697](#)

`gitlab-project-pages-domain-get` command line option, [698](#)

`gitlab-project-pages-domain-list` command line option, [698](#)

`gitlab-project-pages-domain-update` command line option, [698](#)

`gitlab-project-pipeline-bridge-list` command line option, [701](#)

`gitlab-project-pipeline-cancel` command line option, [699](#)

`gitlab-project-pipeline-create` command line option, [699](#)

`gitlab-project-pipeline-delete` command line option, [699](#)

`gitlab-project-pipeline-get` command line option, [700](#)

`gitlab-project-pipeline-job-list` command line option, [702](#)

`gitlab-project-pipeline-list` command line option, [700](#)

`gitlab-project-pipeline-retry` command line option, [701](#)

`gitlab-project-pipeline-schedule-create` command line option, [703](#)

`gitlab-project-pipeline-schedule-delete` command line option, [703](#)

`gitlab-project-pipeline-schedule-get` command line option, [703](#)

`gitlab-project-pipeline-schedule-list` command line option, [704](#)

`gitlab-project-pipeline-schedule-play` command line option, [704](#)

`gitlab-project-pipeline-schedule-take-ownership` command line option, [704](#)

`gitlab-project-pipeline-schedule-update` command line option, [705](#)

`gitlab-project-pipeline-schedule-variable-create` command line option, [705](#)

`gitlab-project-pipeline-schedule-variable-delete` command line option, [706](#)

`gitlab-project-pipeline-schedule-variable-update` command line option, [706](#)

`gitlab-project-pipeline-test-report-get` command line option, [707](#)

`gitlab-project-pipeline-test-report-summary-get` command line option, [707](#)

`gitlab-project-pipeline-variable-list` command line option, [708](#)

`gitlab-project-protected-branch-create` command line option, [709](#)

`gitlab-project-protected-branch-delete` command line option, [709](#)

`gitlab-project-protected-branch-get` command line option, [709](#)

`gitlab-project-protected-branch-list` command line option, [710](#)

`gitlab-project-protected-environment-create` command line option, [710](#)

`gitlab-project-protected-environment-delete` command line option, [711](#)

`gitlab-project-protected-environment-get` command line option, [711](#)

`gitlab-project-protected-environment-list` command line option, [711](#)

`gitlab-project-protected-tag-create` command line option, [712](#)

`gitlab-project-protected-tag-delete` command line option, [712](#)

`gitlab-project-protected-tag-get` command line option, [713](#)

`gitlab-project-protected-tag-list` command line option, [713](#)

`gitlab-project-push-rules-create` command line option, [714](#)

`gitlab-project-push-rules-delete` command line option, [715](#)

`gitlab-project-push-rules-get` command line option, [715](#)



gitlab-project-push-rules-update  
command line option, [715](#)

gitlab-project-registry-repository-delete  
command line option, [716](#)

gitlab-project-registry-repository-list  
command line option, [716](#)

gitlab-project-registry-tag-delete  
command line option, [717](#)

gitlab-project-registry-tag-delete-in-bulk  
command line option, [717](#)

gitlab-project-registry-tag-get command  
line option, [718](#)

gitlab-project-registry-tag-list  
command line option, [718](#)

gitlab-project-release-create command  
line option, [719](#)

gitlab-project-release-delete command  
line option, [719](#)

gitlab-project-release-get command line  
option, [719](#)

gitlab-project-release-link-create  
command line option, [721](#)

gitlab-project-release-link-delete  
command line option, [721](#)

gitlab-project-release-link-get command  
line option, [722](#)

gitlab-project-release-link-list  
command line option, [722](#)

gitlab-project-release-link-update  
command line option, [723](#)

gitlab-project-release-list command  
line option, [720](#)

gitlab-project-release-update command  
line option, [720](#)

gitlab-project-remote-mirror-create  
command line option, [723](#)

gitlab-project-remote-mirror-list  
command line option, [724](#)

gitlab-project-remote-mirror-update  
command line option, [724](#)

gitlab-project-runner-create command  
line option, [724](#)

gitlab-project-runner-delete command  
line option, [725](#)

gitlab-project-runner-list command line  
option, [725](#)

gitlab-project-service-available  
command line option, [726](#)

gitlab-project-service-delete command  
line option, [726](#)

gitlab-project-service-get command line  
option, [726](#)

gitlab-project-service-list command  
line option, [726](#)

gitlab-project-service-update command  
line option, [727](#)

gitlab-project-snippet-award-emoji-create  
command line option, [730](#)

gitlab-project-snippet-award-emoji-delete  
command line option, [730](#)

gitlab-project-snippet-award-emoji-get  
command line option, [731](#)

gitlab-project-snippet-award-emoji-list  
command line option, [731](#)

gitlab-project-snippet-content command  
line option, [727](#)

gitlab-project-snippet-create command  
line option, [728](#)

gitlab-project-snippet-delete command  
line option, [728](#)

gitlab-project-snippet-discussion-create  
command line option, [732](#)

gitlab-project-snippet-discussion-get  
command line option, [732](#)

gitlab-project-snippet-discussion-list  
command line option, [733](#)

gitlab-project-snippet-discussion-note-create  
command line option, [733](#)

gitlab-project-snippet-discussion-note-delete  
command line option, [734](#)

gitlab-project-snippet-discussion-note-get  
command line option, [734](#)

gitlab-project-snippet-discussion-note-update  
command line option, [735](#)

gitlab-project-snippet-get command line  
option, [728](#)

gitlab-project-snippet-list command  
line option, [729](#)

gitlab-project-snippet-note-award-emoji-create  
command line option, [737](#)

gitlab-project-snippet-note-award-emoji-delete  
command line option, [738](#)

gitlab-project-snippet-note-award-emoji-get  
command line option, [738](#)

gitlab-project-snippet-note-award-emoji-list  
command line option, [739](#)

gitlab-project-snippet-note-create  
command line option, [735](#)

gitlab-project-snippet-note-delete  
command line option, [736](#)

gitlab-project-snippet-note-get command  
line option, [736](#)

gitlab-project-snippet-note-list  
command line option, [736](#)

gitlab-project-snippet-note-update  
command line option, [737](#)

gitlab-project-snippet-update command  
line option, [729](#)

- gitlab-project-snippet-user-agent-detail
  - command line option, [730](#)
- gitlab-project-storage-get command line option, [739](#)
- gitlab-project-tag-create command line option, [740](#)
- gitlab-project-tag-delete command line option, [740](#)
- gitlab-project-tag-get command line option, [740](#)
- gitlab-project-tag-list command line option, [741](#)
- gitlab-project-trigger-create command line option, [741](#)
- gitlab-project-trigger-delete command line option, [742](#)
- gitlab-project-trigger-get command line option, [742](#)
- gitlab-project-trigger-list command line option, [742](#)
- gitlab-project-trigger-update command line option, [743](#)
- gitlab-project-user-list command line option, [743](#)
- gitlab-project-variable-create command line option, [744](#)
- gitlab-project-variable-delete command line option, [744](#)
- gitlab-project-variable-get command line option, [745](#)
- gitlab-project-variable-list command line option, [745](#)
- gitlab-project-variable-update command line option, [745](#)
- gitlab-project-wiki-create command line option, [746](#)
- gitlab-project-wiki-delete command line option, [746](#)
- gitlab-project-wiki-get command line option, [747](#)
- gitlab-project-wiki-list command line option, [747](#)
- gitlab-project-wiki-update command line option, [747](#)
- gitlab-todo-list command line option, [758](#)
- projects-limit
  - gitlab-user-create command line option, [762](#)
  - gitlab-user-update command line option, [766](#)
- protected
  - gitlab-group-variable-create command line option, [544](#)
  - gitlab-group-variable-update command line option, [546](#)
  - gitlab-project-variable-create command line option, [744](#)
  - gitlab-project-variable-update command line option, [746](#)
  - gitlab-variable-create command line option, [780](#)
  - gitlab-variable-update command line option, [781](#)
- protected-branch-ids
  - gitlab-project-approval-rule-create command line option, [580](#)
- provider
  - gitlab-group-add-ldap-group-link command line option, [483](#)
  - gitlab-group-delete-ldap-group-link command line option, [485](#)
  - gitlab-ldap-group-list command line option, [552](#)
  - gitlab-user-create command line option, [762](#)
  - gitlab-user-list command line option, [764](#)
  - gitlab-user-update command line option, [766](#)
- public
  - gitlab-user-project-create command line option, [777](#)
- public-builds
  - gitlab-project-create command line option, [563](#)
  - gitlab-project-update command line option, [574](#)
  - gitlab-user-project-create command line option, [777](#)
- public-email
  - gitlab-user-create command line option, [763](#)
  - gitlab-user-update command line option, [766](#)
- push-access-level
  - gitlab-project-protected-branch-create command line option, [709](#)
- push-events
  - gitlab-group-hook-create command line option, [519](#)
  - gitlab-group-hook-update command line option, [521](#)
  - gitlab-project-hook-create command line option, [620](#)
  - gitlab-project-hook-update command line option, [622](#)
- query
  - gitlab-group-invitation-list command line option, [523](#)

---

```

gitlab-project-invitation-list command
  line option, 626
--reassign-issue
  gitlab-group-notification-settings-update
    command line option, 538
  gitlab-notification-settings-update
    command line option, 557
  gitlab-project-notification-settings-update
    command line option, 694
--reassign-merge-request
  gitlab-group-notification-settings-update
    command line option, 538
  gitlab-notification-settings-update
    command line option, 557
  gitlab-project-notification-settings-update-refs
    command line option, 694
--recursive
  gitlab-project-repository-tree command
    line option, 568
--redirect-uri
  gitlab-application-create command line
    option, 460
--ref
  gitlab-project-branch-create command
    line option, 590
  gitlab-project-ci-lint-create command
    line option, 591
  gitlab-project-ci-lint-validate command
    line option, 592
  gitlab-project-commit-status-create
    command line option, 603
  gitlab-project-deployment-create
    command line option, 607
  gitlab-project-file-blame command line
    option, 615
  gitlab-project-file-get command line
    option, 617
  gitlab-project-file-raw command line
    option, 617
  gitlab-project-pipeline-create command
    line option, 699
  gitlab-project-pipeline-list command
    line option, 700
  gitlab-project-pipeline-schedule-create
    command line option, 703
  gitlab-project-pipeline-schedule-update
    command line option, 705
  gitlab-project-release-create command
    line option, 719
  gitlab-project-repository-tree command
    line option, 568
  gitlab-project-tag-create command line
    option, 740
  gitlab-project-trigger-pipeline command
    line option, 570
--ref-name
  gitlab-project-artifact command line
    option, 560
  gitlab-project-artifact-delete command
    line option, 582
  gitlab-project-artifact-download
    command line option, 582
  gitlab-project-artifact-raw command
    line option, 582
  gitlab-project-artifacts command line
    option, 560
  gitlab-project-commit-list command line
    option, 596
  gitlab-project-repository-merge-base
    command line option, 568
--reject-unsigned-commits
  gitlab-group-push-rules-create command
    line option, 541
  gitlab-group-push-rules-update command
    line option, 542
  gitlab-project-push-rules-create
    command line option, 714
  gitlab-project-push-rules-update
    command line option, 716
--released-at
  gitlab-project-release-update command
    line option, 720
--releases-events
  gitlab-group-hook-create command line
    option, 520
  gitlab-group-hook-update command line
    option, 522
--remove-source-branch
  gitlab-project-merge-request-create
    command line option, 663
  gitlab-project-merge-request-update
    command line option, 669
--remove-source-branch-after-merge
  gitlab-project-create command line
    option, 563
  gitlab-project-update command line
    option, 574
--render-html
  gitlab-project-merge-request-get
    command line option, 663
--reopen-issue
  gitlab-group-notification-settings-update
    command line option, 538
  gitlab-notification-settings-update
    command line option, 557
  gitlab-project-notification-settings-update
    command line option, 694

```

--reopen-merge-request  
    gitlab-group-notification-settings-update  
        command line option, [538](#)  
    gitlab-notification-settings-update  
        command line option, [557](#)  
    gitlab-project-notification-settings-update  
        command line option, [694](#)

--repos-max-capacity  
    gitlab-geo-node-update command line  
        option, [481](#)

--repository-access-level  
    gitlab-project-create command line  
        option, [563](#)  
    gitlab-project-update command line  
        option, [574](#)

--repository-checksum-failed  
    gitlab-project-list command line option,  
        [566](#)  
    gitlab-user-project-list command line  
        option, [778](#)

--repository-id  
    gitlab-project-registry-tag-delete  
        command line option, [717](#)  
    gitlab-project-registry-tag-delete-in-bulk  
        command line option, [717](#)  
    gitlab-project-registry-tag-get command  
        line option, [718](#)  
    gitlab-project-registry-tag-list  
        command line option, [718](#)

--repository-storage  
    gitlab-project-create command line  
        option, [563](#)  
    gitlab-project-list command line option,  
        [566](#)  
    gitlab-project-update command line  
        option, [574](#)

--repository-storages  
    gitlab-application-settings-update  
        command line option, [465](#)

--request-access-enabled  
    gitlab-group-create command line option,  
        [484](#)  
    gitlab-group-update command line option,  
        [488](#)  
    gitlab-project-create command line  
        option, [563](#)  
    gitlab-project-update command line  
        option, [574](#)

--require-two-factor-authentication  
    gitlab-group-create command line option,  
        [484](#)  
    gitlab-group-update command line option,  
        [488](#)

--required-approval-count  
    gitlab-project-protected-environment-create  
        command line option, [710](#)

--requirements-access-level  
    gitlab-project-create command line  
        option, [563](#)  
    gitlab-project-update command line  
        option, [574](#)

--reset-approvals-on-push  
    gitlab-project-approval-update command  
        line option, [579](#)

--reset-password  
    gitlab-user-create command line option,  
        [762](#)

--resolve-outdated-diff-discussions  
    gitlab-project-create command line  
        option, [563](#)  
    gitlab-project-update command line  
        option, [574](#)

--resolved  
    gitlab-project-merge-request-discussion-update  
        command line option, [678](#)

--restrict-user-defined-variables  
    gitlab-project-update command line  
        option, [574](#)

--restricted-visibility-levels  
    gitlab-application-settings-update  
        command line option, [464](#)

--reviewer-id  
    gitlab-merge-request-list command line  
        option, [554](#)  
    gitlab-project-deployment-merge-request-list  
        command line option, [610](#)

--reviewer-ids  
    gitlab-project-merge-request-create  
        command line option, [663](#)  
    gitlab-project-merge-request-update  
        command line option, [669](#)

--reviewer-username  
    gitlab-merge-request-list command line  
        option, [554](#)  
    gitlab-project-deployment-merge-request-list  
        command line option, [610](#)

--rsa-key-restriction  
    gitlab-application-settings-update  
        command line option, [465](#)

--run-untagged  
    gitlab-runner-create command line  
        option, [749](#)  
    gitlab-runner-update command line  
        option, [750](#)

--runner-id  
    gitlab-project-runner-create command  
        line option, [724](#)  
    gitlab-runner-job-list command line

- option, [752](#)
- scope
  - gitlab-group-merge-request-list command line option, [533](#)
  - gitlab-group-runner-list command line option, [543](#)
  - gitlab-group-search command line option, [486](#)
  - gitlab-issue-list command line option, [550](#)
  - gitlab-merge-request-list command line option, [554](#)
  - gitlab-project-deployment-merge-request-list command line option, [610](#)
  - gitlab-project-issue-list command line option, [630](#)
  - gitlab-project-job-list command line option, [651](#)
  - gitlab-project-merge-request-list command line option, [664](#)
  - gitlab-project-merge-train-list command line option, [689](#)
  - gitlab-project-pipeline-bridge-list command line option, [701](#)
  - gitlab-project-pipeline-job-list command line option, [702](#)
  - gitlab-project-pipeline-list command line option, [700](#)
  - gitlab-project-runner-list command line option, [725](#)
  - gitlab-project-search command line option, [569](#)
  - gitlab-runner-all command line option, [748](#)
  - gitlab-runner-all-list command line option, [751](#)
  - gitlab-runner-list command line option, [750](#)
- scopes
  - gitlab-application-create command line option, [460](#)
  - gitlab-group-access-token-create command line option, [490](#)
  - gitlab-group-deploy-token-create command line option, [503](#)
  - gitlab-group-deploy-token-list command line option, [504](#)
  - gitlab-project-access-token-create command line option, [577](#)
  - gitlab-project-deploy-token-create command line option, [605](#)
  - gitlab-project-deploy-token-list command line option, [606](#)
  - gitlab-user-impersonation-token-create command line option, [772](#)
- gitlab-user-personal-access-token-create command line option, [776](#)
- search
  - gitlab-group-billable-member-list command line option, [495](#)
  - gitlab-group-descendant-group-list command line option, [505](#)
  - gitlab-group-epic-list command line option, [507](#)
  - gitlab-group-issue-list command line option, [525](#)
  - gitlab-group-iteration-list command line option, [526](#)
  - gitlab-group-list command line option, [486](#)
  - gitlab-group-merge-request-list command line option, [534](#)
  - gitlab-group-milestone-list command line option, [536](#)
  - gitlab-group-project-list command line option, [540](#)
  - gitlab-group-search command line option, [486](#)
  - gitlab-group-subgroup-list command line option, [544](#)
  - gitlab-issue-list command line option, [550](#)
  - gitlab-ldap-group-list command line option, [552](#)
  - gitlab-merge-request-list command line option, [554](#)
  - gitlab-namespace-list command line option, [556](#)
  - gitlab-project-deployment-merge-request-list command line option, [610](#)
  - gitlab-project-environment-list command line option, [612](#)
  - gitlab-project-fork-list command line option, [619](#)
  - gitlab-project-issue-list command line option, [630](#)
  - gitlab-project-list command line option, [566](#)
  - gitlab-project-merge-request-list command line option, [665](#)
  - gitlab-project-milestone-list command line option, [691](#)
  - gitlab-project-search command line option, [569](#)
  - gitlab-project-user-list command line option, [743](#)
  - gitlab-shared-project-list command line option, [753](#)

- gitlab-starred-project-list command line option, [757](#)
- gitlab-user-list command line option, [764](#)
- gitlab-user-project-list command line option, [778](#)
- search-namespaces
  - gitlab-project-list command line option, [566](#)
- server-url
  - gitlab command line option, [459](#)
- service-desk-enabled
  - gitlab-project-update command line option, [574](#)
- session-expire-delay
  - gitlab-application-settings-update command line option, [464](#)
- sha
  - gitlab-project-deployment-create command line option, [607](#)
  - gitlab-project-merge-request-approve command line option, [661](#)
  - gitlab-project-pipeline-list command line option, [700](#)
  - gitlab-project-repository-archive command line option, [567](#)
  - gitlab-project-repository-blob command line option, [567](#)
  - gitlab-project-repository-raw-blob command line option, [568](#)
- share-with-group-lock
  - gitlab-group-create command line option, [484](#)
  - gitlab-group-update command line option, [488](#)
- shared-runners-enabled
  - gitlab-project-create command line option, [563](#)
  - gitlab-project-update command line option, [574](#)
- shared-runners-minutes-limit
  - gitlab-group-create command line option, [484](#)
  - gitlab-group-update command line option, [488](#)
- shared-runners-setting
  - gitlab-group-update command line option, [488](#)
- should-remove-source-branch
  - gitlab-project-merge-request-merge command line option, [665](#)
- show-default-award-emojis
  - gitlab-project-create command line option, [563](#)
  - gitlab-project-update command line option, [574](#)
- sign-in-text
  - gitlab-application-settings-update command line option, [464](#)
- signup-enabled
  - gitlab-application-settings-update command line option, [464](#)
- simple
  - gitlab-group-project-list command line option, [540](#)
  - gitlab-project-fork-list command line option, [619](#)
  - gitlab-project-list command line option, [566](#)
  - gitlab-shared-project-list command line option, [753](#)
  - gitlab-starred-project-list command line option, [757](#)
  - gitlab-user-project-list command line option, [778](#)
- since
  - gitlab-project-commit-list command line option, [596](#)
- skip-confirmation
  - gitlab-user-create command line option, [762](#)
- skip-groups
  - gitlab-group-descendant-group-list command line option, [505](#)
  - gitlab-group-list command line option, [485](#)
  - gitlab-group-subgroup-list command line option, [544](#)
- skip-reconfirmation
  - gitlab-user-update command line option, [766](#)
- skip-users
  - gitlab-project-user-list command line option, [743](#)
- skype
  - gitlab-user-create command line option, [762](#)
  - gitlab-user-update command line option, [766](#)
- slug
  - gitlab-group-wiki-delete command line option, [547](#)
  - gitlab-group-wiki-get command line option, [547](#)
  - gitlab-group-wiki-update command line option, [548](#)
  - gitlab-project-integration-available command line option, [623](#)
  - gitlab-project-integration-delete

command line option, [623](#)  
 gitlab-project-integration-get command line option, [623](#)  
 gitlab-project-integration-update command line option, [624](#)  
 gitlab-project-service-available command line option, [726](#)  
 gitlab-project-service-delete command line option, [726](#)  
 gitlab-project-service-get command line option, [726](#)  
 gitlab-project-service-update command line option, [727](#)  
 gitlab-project-wiki-delete command line option, [746](#)  
 gitlab-project-wiki-get command line option, [747](#)  
 gitlab-project-wiki-update command line option, [747](#)  
 --snippet-id  
 gitlab-project-snippet-award-emoji-create command line option, [730](#)  
 gitlab-project-snippet-award-emoji-delete command line option, [731](#)  
 gitlab-project-snippet-award-emoji-get command line option, [731](#)  
 gitlab-project-snippet-award-emoji-list command line option, [731](#)  
 gitlab-project-snippet-discussion-create command line option, [732](#)  
 gitlab-project-snippet-discussion-get command line option, [732](#)  
 gitlab-project-snippet-discussion-list command line option, [733](#)  
 gitlab-project-snippet-discussion-note-create command line option, [733](#)  
 gitlab-project-snippet-discussion-note-delete command line option, [734](#)  
 gitlab-project-snippet-discussion-note-get command line option, [734](#)  
 gitlab-project-snippet-discussion-note-update command line option, [735](#)  
 gitlab-project-snippet-note-award-emoji-create command line option, [737](#)  
 gitlab-project-snippet-note-award-emoji-delete command line option, [738](#)  
 gitlab-project-snippet-note-award-emoji-get command line option, [738](#)  
 gitlab-project-snippet-note-award-emoji-list command line option, [739](#)  
 gitlab-project-snippet-note-create command line option, [735](#)  
 gitlab-project-snippet-note-delete command line option, [736](#)  
 gitlab-project-snippet-note-get command line option, [736](#)  
 gitlab-project-snippet-note-list command line option, [736](#)  
 gitlab-project-snippet-note-update command line option, [737](#)  
 --snippets-access-level  
 gitlab-project-create command line option, [563](#)  
 gitlab-project-update command line option, [574](#)  
 --snippets-enabled  
 gitlab-project-create command line option, [563](#)  
 gitlab-project-update command line option, [574](#)  
 gitlab-user-project-create command line option, [777](#)  
 --sort  
 gitlab-event-list command line option, [476](#)  
 gitlab-group-billable-member-list command line option, [495](#)  
 gitlab-group-descendant-group-list command line option, [505](#)  
 gitlab-group-epic-list command line option, [507](#)  
 gitlab-group-issue-list command line option, [525](#)  
 gitlab-group-list command line option, [486](#)  
 gitlab-group-merge-request-list command line option, [533](#)  
 gitlab-group-package-list command line option, [538](#)  
 gitlab-group-project-list command line option, [539](#)  
 gitlab-group-subgroup-list command line option, [544](#)  
 gitlab-issue-list command line option, [550](#)  
 gitlab-merge-request-list command line option, [554](#)  
 gitlab-project-deployment-list command line option, [608](#)  
 gitlab-project-deployment-merge-request-list command line option, [610](#)  
 gitlab-project-event-list command line option, [614](#)  
 gitlab-project-fork-list command line option, [619](#)  
 gitlab-project-issue-list command line option, [630](#)  
 gitlab-project-list command line option,



- 566
- gitlab-project-merge-request-list
  - command line option, 664
- gitlab-project-package-list command line option, 695
- gitlab-project-pipeline-list command line option, 700
- gitlab-project-release-list command line option, 720
- gitlab-shared-project-list command line option, 753
- gitlab-starred-project-list command line option, 757
- gitlab-user-event-list command line option, 770
- gitlab-user-project-list command line option, 778
- source-branch
  - gitlab-group-merge-request-list command line option, 534
  - gitlab-merge-request-list command line option, 554
  - gitlab-project-deployment-merge-request-liststate command line option, 610
  - gitlab-project-merge-request-create command line option, 662
  - gitlab-project-merge-request-list command line option, 665
- source-id
  - gitlab-user-membership-get command line option, 775
- squash
  - gitlab-project-merge-request-create command line option, 663
  - gitlab-project-merge-request-update command line option, 669
- squash-option
  - gitlab-project-create command line option, 563
  - gitlab-project-update command line option, 575
  - gitlab-user-project-create command line option, 777
- ssl-verify
  - gitlab command line option, 459
- starred
  - gitlab-group-project-list command line option, 540
  - gitlab-project-fork-list command line option, 619
  - gitlab-project-list command line option, 566
  - gitlab-shared-project-list command line option, 753
  - gitlab-starred-project-list command line option, 757
  - gitlab-user-project-list command line option, 778
- start-date
  - gitlab-group-epic-create command line option, 506
  - gitlab-group-epic-update command line option, 508
  - gitlab-group-milestone-create command line option, 534
  - gitlab-group-milestone-update command line option, 537
  - gitlab-project-milestone-create command line option, 690
  - gitlab-project-milestone-update command line option, 692
- starts-at
  - gitlab-broadcast-message-create command line option, 467
  - gitlab-broadcast-message-update command line option, 468
- gitlab-group-issue-list command line option, 525
- gitlab-group-iteration-list command line option, 526
- gitlab-group-merge-request-list command line option, 533
- gitlab-group-milestone-list command line option, 536
- gitlab-issue-list command line option, 550
- gitlab-merge-request-list command line option, 554
- gitlab-project-commit-status-create command line option, 603
- gitlab-project-deployment-merge-request-list command line option, 610
- gitlab-project-issue-list command line option, 629
- gitlab-project-merge-request-list command line option, 664
- gitlab-project-milestone-list command line option, 691
- gitlab-todo-list command line option, 758
- gitlab-user-impersonation-token-list command line option, 773
- state-event
  - gitlab-group-milestone-update command line option, 537
  - gitlab-project-issue-update command line option, 634
  - gitlab-project-merge-request-update



- command line option, 669
  - gitlab-project-milestone-create command line option, 690
  - gitlab-project-milestone-update command line option, 692
- states
  - gitlab-project-environment-list command line option, 612
- statistics
  - gitlab-group-descendant-group-list command line option, 505
  - gitlab-group-list command line option, 486
  - gitlab-group-subgroup-list command line option, 544
  - gitlab-project-fork-list command line option, 619
  - gitlab-project-list command line option, 566
  - gitlab-starred-project-list command line option, 757
  - gitlab-user-project-list command line option, 778
- status
  - gitlab-project-deployment-create command line option, 607
  - gitlab-project-deployment-list command line option, 608
  - gitlab-project-pipeline-list command line option, 700
  - gitlab-runner-all-list command line option, 751
  - gitlab-runner-job-list command line option, 752
  - gitlab-runner-list command line option, 750
  - gitlab-user-list command line option, 764
- subgroup-creation-level
  - gitlab-group-create command line option, 484
  - gitlab-group-update command line option, 488
- subgroup-events
  - gitlab-group-hook-create command line option, 520
  - gitlab-group-hook-update command line option, 522
- submodule
  - gitlab-project-update-submodule command line option, 575
- sudo
  - gitlab-application-appearance-get command line option, 461
  - gitlab-application-appearance-update command line option, 461
  - gitlab-application-create command line option, 459
  - gitlab-application-delete command line option, 460
  - gitlab-application-list command line option, 460
  - gitlab-application-settings-get command line option, 462
  - gitlab-application-settings-update command line option, 464
  - gitlab-audit-event-get command line option, 466
  - gitlab-audit-event-list command line option, 466
  - gitlab-broadcast-message-create command line option, 467
  - gitlab-broadcast-message-delete command line option, 467
  - gitlab-broadcast-message-get command line option, 467
  - gitlab-broadcast-message-list command line option, 468
  - gitlab-broadcast-message-update command line option, 468
  - gitlab-ci-lint-create command line option, 469
  - gitlab-current-user-email-create command line option, 470
  - gitlab-current-user-email-delete command line option, 470
  - gitlab-current-user-email-get command line option, 470
  - gitlab-current-user-email-list command line option, 471
  - gitlab-current-user-get command line option, 470
  - gitlab-current-user-gpg-key-create command line option, 471
  - gitlab-current-user-gpg-key-delete command line option, 471
  - gitlab-current-user-gpg-key-get command line option, 472
  - gitlab-current-user-gpg-key-list command line option, 472
  - gitlab-current-user-key-create command line option, 472
  - gitlab-current-user-key-delete command line option, 473
  - gitlab-current-user-key-get command line option, 473
  - gitlab-current-user-key-list command line option, 473
  - gitlab-current-user-status-get command

- line option, [474](#)
- gitlab-current-user-status-update
  - command line option, [474](#)
- gitlab-deploy-key-list command line option, [474](#)
- gitlab-deploy-token-list command line option, [475](#)
- gitlab-dockerfile-get command line option, [475](#)
- gitlab-dockerfile-list command line option, [476](#)
- gitlab-event-list command line option, [476](#)
- gitlab-feature-delete command line option, [477](#)
- gitlab-feature-list command line option, [477](#)
- gitlab-generic-package-download command line option, [478](#)
- gitlab-generic-package-upload command line option, [478](#)
- gitlab-geo-node-delete command line option, [479](#)
- gitlab-geo-node-get command line option, [479](#)
- gitlab-geo-node-list command line option, [480](#)
- gitlab-geo-node-update command line option, [480](#)
- gitlab-gitignore-get command line option, [481](#)
- gitlab-gitignore-list command line option, [481](#)
- gitlab-gitlabciyaml-get command line option, [482](#)
- gitlab-gitlabciyaml-list command line option, [482](#)
- gitlab-group-access-request-approve command line option, [489](#)
- gitlab-group-access-request-create command line option, [489](#)
- gitlab-group-access-request-delete command line option, [489](#)
- gitlab-group-access-request-list command line option, [490](#)
- gitlab-group-access-token-create command line option, [490](#)
- gitlab-group-access-token-delete command line option, [491](#)
- gitlab-group-access-token-list command line option, [491](#)
- gitlab-group-audit-event-get command line option, [492](#)
- gitlab-group-audit-event-list command line option, [492](#)
- gitlab-group-badge-create command line option, [493](#)
- gitlab-group-badge-delete command line option, [493](#)
- gitlab-group-badge-get command line option, [493](#)
- gitlab-group-badge-list command line option, [493](#)
- gitlab-group-badge-render command line option, [494](#)
- gitlab-group-badge-update command line option, [494](#)
- gitlab-group-billable-member-delete command line option, [495](#)
- gitlab-group-billable-member-list command line option, [495](#)
- gitlab-group-billable-member-membership-list command line option, [496](#)
- gitlab-group-board-create command line option, [496](#)
- gitlab-group-board-delete command line option, [497](#)
- gitlab-group-board-get command line option, [497](#)
- gitlab-group-board-list command line option, [497](#)
- gitlab-group-board-list-create command line option, [498](#)
- gitlab-group-board-list-delete command line option, [498](#)
- gitlab-group-board-list-get command line option, [499](#)
- gitlab-group-board-list-list command line option, [499](#)
- gitlab-group-board-list-update command line option, [499](#)
- gitlab-group-board-update command line option, [498](#)
- gitlab-group-cluster-create command line option, [500](#)
- gitlab-group-cluster-delete command line option, [500](#)
- gitlab-group-cluster-get command line option, [501](#)
- gitlab-group-cluster-list command line option, [501](#)
- gitlab-group-cluster-update command line option, [501](#)
- gitlab-group-create command line option, [483](#)
- gitlab-group-custom-attribute-delete command line option, [502](#)
- gitlab-group-custom-attribute-get

- command line option, [502](#)
- gitlab-group-custom-attribute-list
  - command line option, [503](#)
- gitlab-group-delete command line option, [484](#)
- gitlab-group-deploy-token-create
  - command line option, [503](#)
- gitlab-group-deploy-token-delete
  - command line option, [504](#)
- gitlab-group-deploy-token-get command line option, [504](#)
- gitlab-group-deploy-token-list command line option, [504](#)
- gitlab-group-descendant-group-list
  - command line option, [505](#)
- gitlab-group-epic-award-emoji-create
  - command line option, [508](#)
- gitlab-group-epic-award-emoji-delete
  - command line option, [509](#)
- gitlab-group-epic-award-emoji-get
  - command line option, [509](#)
- gitlab-group-epic-award-emoji-list
  - command line option, [509](#)
- gitlab-group-epic-create command line option, [506](#)
- gitlab-group-epic-delete command line option, [506](#)
- gitlab-group-epic-discussion-note-create
  - command line option, [510](#)
- gitlab-group-epic-discussion-note-delete
  - command line option, [510](#)
- gitlab-group-epic-discussion-note-get
  - command line option, [511](#)
- gitlab-group-epic-discussion-note-update
  - command line option, [511](#)
- gitlab-group-epic-get command line option, [507](#)
- gitlab-group-epic-issue-create command line option, [512](#)
- gitlab-group-epic-issue-delete command line option, [512](#)
- gitlab-group-epic-issue-list command line option, [512](#)
- gitlab-group-epic-issue-update command line option, [513](#)
- gitlab-group-epic-list command line option, [507](#)
- gitlab-group-epic-note-award-emoji-create
  - command line option, [515](#)
- gitlab-group-epic-note-award-emoji-delete
  - command line option, [516](#)
- gitlab-group-epic-note-award-emoji-get
  - command line option, [516](#)
- gitlab-group-epic-note-award-emoji-list
  - command line option, [516](#)
- gitlab-group-epic-note-create
  - command line option, [513](#)
- gitlab-group-epic-note-delete
  - command line option, [514](#)
- gitlab-group-epic-note-get command line option, [514](#)
- gitlab-group-epic-note-list
  - command line option, [514](#)
- gitlab-group-epic-note-update
  - command line option, [515](#)
- gitlab-group-epic-resource-label-event-get
  - command line option, [517](#)
- gitlab-group-epic-resource-label-event-list
  - command line option, [517](#)
- gitlab-group-epic-update command line option, [508](#)
- gitlab-group-export-create command line option, [518](#)
- gitlab-group-export-download
  - command line option, [518](#)
- gitlab-group-export-get command line option, [518](#)
- gitlab-group-get command line option, [485](#)
- gitlab-group-hook-create command line option, [519](#)
- gitlab-group-hook-delete
  - command line option, [520](#)
- gitlab-group-hook-get
  - command line option, [520](#)
- gitlab-group-hook-list
  - command line option, [520](#)
- gitlab-group-hook-update
  - command line option, [521](#)
- gitlab-group-import-get
  - command line option, [522](#)
- gitlab-group-invitation-create
  - command line option, [522](#)
- gitlab-group-invitation-delete
  - command line option, [523](#)
- gitlab-group-invitation-get
  - command line option, [523](#)
- gitlab-group-invitation-list
  - command line option, [523](#)
- gitlab-group-invitation-update
  - command line option, [524](#)
- gitlab-group-issue-list
  - command line option, [525](#)
- gitlab-group-issues-statistics-get
  - command line option, [526](#)
- gitlab-group-iteration-list
  - command line option, [526](#)
- gitlab-group-label-create
  - command line option, [527](#)

gitlab-group-label-delete command line option, [527](#)  
gitlab-group-label-get command line option, [528](#)  
gitlab-group-label-list command line option, [528](#)  
gitlab-group-label-subscribe command line option, [528](#)  
gitlab-group-label-unsubscribe command line option, [529](#)  
gitlab-group-label-update command line option, [529](#)  
gitlab-group-list command line option, [485](#)  
gitlab-group-member-all-get command line option, [532](#)  
gitlab-group-member-all-list command line option, [532](#)  
gitlab-group-member-create command line option, [530](#)  
gitlab-group-member-delete command line option, [530](#)  
gitlab-group-member-get command line option, [530](#)  
gitlab-group-member-list command line option, [531](#)  
gitlab-group-member-update command line option, [531](#)  
gitlab-group-merge-request-list command line option, [533](#)  
gitlab-group-milestone-create command line option, [534](#)  
gitlab-group-milestone-delete command line option, [535](#)  
gitlab-group-milestone-get command line option, [535](#)  
gitlab-group-milestone-issues command line option, [535](#)  
gitlab-group-milestone-list command line option, [535](#)  
gitlab-group-milestone-merge-requests command line option, [536](#)  
gitlab-group-milestone-update command line option, [536](#)  
gitlab-group-notification-settings-get command line option, [537](#)  
gitlab-group-notification-settings-update command line option, [537](#)  
gitlab-group-package-list command line option, [538](#)  
gitlab-group-project-list command line option, [539](#)  
gitlab-group-push-rules-create command line option, [541](#)  
gitlab-group-push-rules-delete command line option, [541](#)  
gitlab-group-push-rules-get command line option, [541](#)  
gitlab-group-push-rules-update command line option, [542](#)  
gitlab-group-runner-list command line option, [543](#)  
gitlab-group-subgroup-list command line option, [543](#)  
gitlab-group-update command line option, [488](#)  
gitlab-group-variable-create command line option, [544](#)  
gitlab-group-variable-delete command line option, [545](#)  
gitlab-group-variable-get command line option, [545](#)  
gitlab-group-variable-list command line option, [545](#)  
gitlab-group-variable-update command line option, [546](#)  
gitlab-group-wiki-create command line option, [546](#)  
gitlab-group-wiki-delete command line option, [547](#)  
gitlab-group-wiki-get command line option, [547](#)  
gitlab-group-wiki-list command line option, [547](#)  
gitlab-group-wiki-update command line option, [548](#)  
gitlab-hook-create command line option, [548](#)  
gitlab-hook-delete command line option, [548](#)  
gitlab-hook-get command line option, [549](#)  
gitlab-hook-list command line option, [549](#)  
gitlab-issue-get command line option, [549](#)  
gitlab-issue-list command line option, [550](#)  
gitlab-issues-statistics-get command line option, [551](#)  
gitlab-key-get command line option, [551](#)  
gitlab-ldap-group-list command line option, [552](#)  
gitlab-license-get command line option, [552](#)  
gitlab-license-list command line option, [553](#)  
gitlab-merge-request-list command line option, [554](#)  
gitlab-namespace-get command line option, [555](#)

- gitlab-namespace-list command line option, [556](#)
- gitlab-notification-settings-get command line option, [556](#)
- gitlab-notification-settings-update command line option, [557](#)
- gitlab-pages-domain-list command line option, [557](#)
- gitlab-personal-access-token-delete command line option, [558](#)
- gitlab-personal-access-token-list command line option, [558](#)
- gitlab-project-access-request-approve command line option, [576](#)
- gitlab-project-access-request-create command line option, [576](#)
- gitlab-project-access-request-delete command line option, [576](#)
- gitlab-project-access-request-list command line option, [577](#)
- gitlab-project-access-token-create command line option, [577](#)
- gitlab-project-access-token-delete command line option, [578](#)
- gitlab-project-access-token-list command line option, [578](#)
- gitlab-project-additional-statistics-get command line option, [579](#)
- gitlab-project-approval-get command line option, [579](#)
- gitlab-project-approval-rule-create command line option, [580](#)
- gitlab-project-approval-rule-delete command line option, [580](#)
- gitlab-project-approval-rule-list command line option, [581](#)
- gitlab-project-approval-rule-update command line option, [581](#)
- gitlab-project-approval-update command line option, [579](#)
- gitlab-project-artifact-delete command line option, [582](#)
- gitlab-project-artifact-download command line option, [582](#)
- gitlab-project-artifact-raw command line option, [582](#)
- gitlab-project-audit-event-get command line option, [583](#)
- gitlab-project-audit-event-list command line option, [583](#)
- gitlab-project-badge-create command line option, [584](#)
- gitlab-project-badge-delete command line option, [584](#)
- gitlab-project-badge-get command line option, [584](#)
- gitlab-project-badge-list command line option, [585](#)
- gitlab-project-badge-render command line option, [585](#)
- gitlab-project-badge-update command line option, [585](#)
- gitlab-project-board-create command line option, [586](#)
- gitlab-project-board-delete command line option, [586](#)
- gitlab-project-board-get command line option, [586](#)
- gitlab-project-board-list command line option, [587](#)
- gitlab-project-board-list-create command line option, [588](#)
- gitlab-project-board-list-delete command line option, [588](#)
- gitlab-project-board-list-get command line option, [588](#)
- gitlab-project-board-list-list command line option, [589](#)
- gitlab-project-board-list-update command line option, [589](#)
- gitlab-project-board-update command line option, [587](#)
- gitlab-project-branch-create command line option, [590](#)
- gitlab-project-branch-delete command line option, [590](#)
- gitlab-project-branch-get command line option, [590](#)
- gitlab-project-branch-list command line option, [591](#)
- gitlab-project-ci-lint-create command line option, [591](#)
- gitlab-project-ci-lint-get command line option, [592](#)
- gitlab-project-ci-lint-validate command line option, [592](#)
- gitlab-project-cluster-create command line option, [593](#)
- gitlab-project-cluster-delete command line option, [593](#)
- gitlab-project-cluster-get command line option, [593](#)
- gitlab-project-cluster-list command line option, [594](#)
- gitlab-project-cluster-update command line option, [594](#)
- gitlab-project-commit-cherry-pick command line option, [595](#)

`gitlab-project-commit-comment-create`  
command line option, [598](#)

`gitlab-project-commit-comment-list`  
command line option, [599](#)

`gitlab-project-commit-create` command  
line option, [595](#)

`gitlab-project-commit-diff` command line  
option, [596](#)

`gitlab-project-commit-discussion-create`  
command line option, [599](#)

`gitlab-project-commit-discussion-get`  
command line option, [600](#)

`gitlab-project-commit-discussion-list`  
command line option, [600](#)

`gitlab-project-commit-discussion-note-create`  
command line option, [601](#)

`gitlab-project-commit-discussion-note-delete`  
command line option, [601](#)

`gitlab-project-commit-discussion-note-get`  
command line option, [602](#)

`gitlab-project-commit-discussion-note-update`  
command line option, [602](#)

`gitlab-project-commit-get` command line  
option, [596](#)

`gitlab-project-commit-list` command line  
option, [596](#)

`gitlab-project-commit-merge-requests`  
command line option, [597](#)

`gitlab-project-commit-refs` command line  
option, [597](#)

`gitlab-project-commit-revert` command  
line option, [598](#)

`gitlab-project-commit-signature` command  
line option, [598](#)

`gitlab-project-commit-status-create`  
command line option, [603](#)

`gitlab-project-commit-status-list`  
command line option, [603](#)

`gitlab-project-create` command line  
option, [562](#)

`gitlab-project-custom-attribute-delete`  
command line option, [604](#)

`gitlab-project-custom-attribute-get`  
command line option, [604](#)

`gitlab-project-custom-attribute-list`  
command line option, [605](#)

`gitlab-project-delete` command line  
option, [564](#)

`gitlab-project-deploy-token-create`  
command line option, [605](#)

`gitlab-project-deploy-token-delete`  
command line option, [606](#)

`gitlab-project-deploy-token-get` command  
line option, [606](#)

`gitlab-project-deploy-token-list`  
command line option, [606](#)

`gitlab-project-deployment-create`  
command line option, [607](#)

`gitlab-project-deployment-get` command  
line option, [607](#)

`gitlab-project-deployment-list` command  
line option, [608](#)

`gitlab-project-deployment-merge-request-list`  
command line option, [609](#)

`gitlab-project-deployment-update`  
command line option, [608](#)

`gitlab-project-environment-create`  
command line option, [611](#)

`gitlab-project-environment-delete`  
command line option, [612](#)

`gitlab-project-environment-get` command  
line option, [612](#)

`gitlab-project-environment-list` command  
line option, [612](#)

`gitlab-project-environment-stop` command  
line option, [613](#)

`gitlab-project-environment-update`  
command line option, [613](#)

`gitlab-project-event-list` command line  
option, [614](#)

`gitlab-project-export-create` command  
line option, [614](#)

`gitlab-project-export-download` command  
line option, [615](#)

`gitlab-project-export-get` command line  
option, [615](#)

`gitlab-project-file-blame` command line  
option, [615](#)

`gitlab-project-file-create` command line  
option, [616](#)

`gitlab-project-file-delete` command line  
option, [616](#)

`gitlab-project-file-get` command line  
option, [617](#)

`gitlab-project-file-raw` command line  
option, [617](#)

`gitlab-project-file-update` command line  
option, [617](#)

`gitlab-project-fork-create` command line  
option, [618](#)

`gitlab-project-fork-list` command line  
option, [618](#)

`gitlab-project-get` command line option,  
[565](#)

`gitlab-project-hook-create` command line  
option, [620](#)

`gitlab-project-hook-delete` command line  
option, [620](#)

gitlab-project-hook-get command line option, [621](#)  
gitlab-project-hook-list command line option, [621](#)  
gitlab-project-hook-update command line option, [622](#)  
gitlab-project-import-get command line option, [622](#)  
gitlab-project-integration-available command line option, [623](#)  
gitlab-project-integration-delete command line option, [623](#)  
gitlab-project-integration-get command line option, [623](#)  
gitlab-project-integration-list command line option, [624](#)  
gitlab-project-integration-update command line option, [624](#)  
gitlab-project-invitation-create command line option, [625](#)  
gitlab-project-invitation-delete command line option, [625](#)  
gitlab-project-invitation-get command line option, [625](#)  
gitlab-project-invitation-list command line option, [626](#)  
gitlab-project-invitation-update command line option, [626](#)  
gitlab-project-issue-add-spent-time command line option, [627](#)  
gitlab-project-issue-award-emoji-create command line option, [635](#)  
gitlab-project-issue-award-emoji-delete command line option, [635](#)  
gitlab-project-issue-award-emoji-get command line option, [636](#)  
gitlab-project-issue-award-emoji-list command line option, [636](#)  
gitlab-project-issue-closed-by command line option, [628](#)  
gitlab-project-issue-create command line option, [628](#)  
gitlab-project-issue-delete command line option, [629](#)  
gitlab-project-issue-discussion-create command line option, [637](#)  
gitlab-project-issue-discussion-get command line option, [637](#)  
gitlab-project-issue-discussion-list command line option, [637](#)  
gitlab-project-issue-discussion-note-create command line option, [638](#)  
gitlab-project-issue-discussion-note-delete command line option, [639](#)  
gitlab-project-issue-discussion-note-get command line option, [639](#)  
gitlab-project-issue-discussion-note-update command line option, [639](#)  
gitlab-project-issue-get command line option, [629](#)  
gitlab-project-issue-link-create command line option, [640](#)  
gitlab-project-issue-link-delete command line option, [640](#)  
gitlab-project-issue-link-list command line option, [641](#)  
gitlab-project-issue-list command line option, [629](#)  
gitlab-project-issue-move command line option, [630](#)  
gitlab-project-issue-note-award-emoji-create command line option, [643](#)  
gitlab-project-issue-note-award-emoji-delete command line option, [644](#)  
gitlab-project-issue-note-award-emoji-get command line option, [644](#)  
gitlab-project-issue-note-award-emoji-list command line option, [644](#)  
gitlab-project-issue-note-create command line option, [641](#)  
gitlab-project-issue-note-delete command line option, [642](#)  
gitlab-project-issue-note-get command line option, [642](#)  
gitlab-project-issue-note-list command line option, [642](#)  
gitlab-project-issue-note-update command line option, [643](#)  
gitlab-project-issue-participants command line option, [631](#)  
gitlab-project-issue-related-merge-requests command line option, [631](#)  
gitlab-project-issue-reorder command line option, [631](#)  
gitlab-project-issue-reset-spent-time command line option, [632](#)  
gitlab-project-issue-reset-time-estimate command line option, [632](#)  
gitlab-project-issue-resource-label-event-get command line option, [645](#)  
gitlab-project-issue-resource-label-event-list command line option, [646](#)  
gitlab-project-issue-resource-milestone-event-get command line option, [646](#)  
gitlab-project-issue-resource-milestone-event-list command line option, [647](#)  
gitlab-project-issue-resource-state-event-get command line option, [647](#)



gitlab-project-issue-resource-state-event-list command line option, [648](#)  
gitlab-project-issue-subscribe command line option, [632](#)  
gitlab-project-issue-time-estimate command line option, [632](#)  
gitlab-project-issue-time-stats command line option, [633](#)  
gitlab-project-issue-todo command line option, [633](#)  
gitlab-project-issue-unsubscribe command line option, [633](#)  
gitlab-project-issue-update command line option, [634](#)  
gitlab-project-issue-user-agent-detail command line option, [634](#)  
gitlab-project-issues-statistics-get command line option, [648](#)  
gitlab-project-job-artifact command line option, [649](#)  
gitlab-project-job-artifacts command line option, [649](#)  
gitlab-project-job-cancel command line option, [650](#)  
gitlab-project-job-delete-artifacts command line option, [650](#)  
gitlab-project-job-erase command line option, [650](#)  
gitlab-project-job-get command line option, [650](#)  
gitlab-project-job-keep-artifacts command line option, [651](#)  
gitlab-project-job-list command line option, [651](#)  
gitlab-project-job-play command line option, [651](#)  
gitlab-project-job-retry command line option, [652](#)  
gitlab-project-job-trace command line option, [652](#)  
gitlab-project-key-create command line option, [652](#)  
gitlab-project-key-delete command line option, [653](#)  
gitlab-project-key-enable command line option, [653](#)  
gitlab-project-key-get command line option, [653](#)  
gitlab-project-key-list command line option, [654](#)  
gitlab-project-key-update command line option, [654](#)  
gitlab-project-label-create command line option, [655](#)  
gitlab-project-label-delete command line option, [655](#)  
gitlab-project-label-get command line option, [655](#)  
gitlab-project-label-list command line option, [656](#)  
gitlab-project-label-subscribe command line option, [656](#)  
gitlab-project-label-unsubscribe command line option, [656](#)  
gitlab-project-label-update command line option, [657](#)  
gitlab-project-list command line option, [566](#)  
gitlab-project-member-all-get command line option, [659](#)  
gitlab-project-member-all-list command line option, [659](#)  
gitlab-project-member-create command line option, [657](#)  
gitlab-project-member-delete command line option, [658](#)  
gitlab-project-member-get command line option, [658](#)  
gitlab-project-member-list command line option, [658](#)  
gitlab-project-member-update command line option, [659](#)  
gitlab-project-merge-request-add-spent-time command line option, [661](#)  
gitlab-project-merge-request-approval-get command line option, [670](#)  
gitlab-project-merge-request-approval-rule-create command line option, [671](#)  
gitlab-project-merge-request-approval-rule-delete command line option, [671](#)  
gitlab-project-merge-request-approval-rule-get command line option, [672](#)  
gitlab-project-merge-request-approval-rule-list command line option, [672](#)  
gitlab-project-merge-request-approval-rule-update command line option, [673](#)  
gitlab-project-merge-request-approval-state-get command line option, [674](#)  
gitlab-project-merge-request-approval-update command line option, [670](#)  
gitlab-project-merge-request-approve command line option, [661](#)  
gitlab-project-merge-request-award-emoji-create command line option, [674](#)  
gitlab-project-merge-request-award-emoji-delete command line option, [675](#)  
gitlab-project-merge-request-award-emoji-get command line option, [675](#)



gitlab-project-merge-request-award-emoji-list command line option, 675  
 gitlab-project-merge-request-cancel-merge-when-pipeline-is-ready command line option, 661  
 gitlab-project-merge-request-changes command line option, 661  
 gitlab-project-merge-request-closes-issues command line option, 662  
 gitlab-project-merge-request-commits command line option, 662  
 gitlab-project-merge-request-create command line option, 662  
 gitlab-project-merge-request-delete command line option, 663  
 gitlab-project-merge-request-diff-get command line option, 676  
 gitlab-project-merge-request-diff-list command line option, 676  
 gitlab-project-merge-request-discussion-create command line option, 677  
 gitlab-project-merge-request-discussion-get command line option, 677  
 gitlab-project-merge-request-discussion-list command line option, 678  
 gitlab-project-merge-request-discussion-note-create command line option, 679  
 gitlab-project-merge-request-discussion-note-delete command line option, 679  
 gitlab-project-merge-request-discussion-note-get command line option, 680  
 gitlab-project-merge-request-discussion-note-update command line option, 680  
 gitlab-project-merge-request-discussion-update command line option, 678  
 gitlab-project-merge-request-get command line option, 663  
 gitlab-project-merge-request-list command line option, 664  
 gitlab-project-merge-request-merge command line option, 665  
 gitlab-project-merge-request-merge-ref command line option, 666  
 gitlab-project-merge-request-note-award-emoji-create command line option, 683  
 gitlab-project-merge-request-note-award-emoji-delete command line option, 683  
 gitlab-project-merge-request-note-award-emoji-get command line option, 684  
 gitlab-project-merge-request-note-award-emoji-list command line option, 684  
 gitlab-project-merge-request-note-create command line option, 681  
 gitlab-project-merge-request-note-delete command line option, 681  
 gitlab-project-merge-request-note-get command line option, 682  
 gitlab-project-merge-request-note-list command line option, 682  
 gitlab-project-merge-request-note-update command line option, 682  
 gitlab-project-merge-request-participants command line option, 666  
 gitlab-project-merge-request-pipeline-create command line option, 685  
 gitlab-project-merge-request-pipeline-list command line option, 685  
 gitlab-project-merge-request-rebase command line option, 666  
 gitlab-project-merge-request-reset-spent-time command line option, 666  
 gitlab-project-merge-request-reset-time-estimate command line option, 667  
 gitlab-project-merge-request-resource-label-event-get command line option, 686  
 gitlab-project-merge-request-resource-label-event-list command line option, 686  
 gitlab-project-merge-request-resource-milestone-event-create command line option, 687  
 gitlab-project-merge-request-resource-milestone-event-delete command line option, 687  
 gitlab-project-merge-request-resource-milestone-event-get command line option, 688  
 gitlab-project-merge-request-resource-milestone-event-list command line option, 688  
 gitlab-project-merge-request-subscribe command line option, 667  
 gitlab-project-merge-request-time-estimate command line option, 667  
 gitlab-project-merge-request-time-stats command line option, 668  
 gitlab-project-merge-request-todo command line option, 668  
 gitlab-project-merge-request-unapprove command line option, 668  
 gitlab-project-merge-request-unsubscribe command line option, 668  
 gitlab-project-merge-request-update command line option, 669  
 gitlab-project-merge-train-list command line option, 689  
 gitlab-project-milestone-create command line option, 690  
 gitlab-project-milestone-delete command line option, 690  
 gitlab-project-milestone-get command line option, 690  
 gitlab-project-milestone-issues command line option, 691

`gitlab-project-milestone-list` command line option, [691](#)

`gitlab-project-milestone-merge-requests` command line option, [691](#)

`gitlab-project-milestone-update` command line option, [692](#)

`gitlab-project-note-get` command line option, [692](#)

`gitlab-project-note-list` command line option, [693](#)

`gitlab-project-notification-settings-get` command line option, [693](#)

`gitlab-project-notification-settings-update` command line option, [694](#)

`gitlab-project-package-delete` command line option, [695](#)

`gitlab-project-package-file-delete` command line option, [696](#)

`gitlab-project-package-file-list` command line option, [696](#)

`gitlab-project-package-get` command line option, [695](#)

`gitlab-project-package-list` command line option, [695](#)

`gitlab-project-pages-domain-create` command line option, [697](#)

`gitlab-project-pages-domain-delete` command line option, [697](#)

`gitlab-project-pages-domain-get` command line option, [698](#)

`gitlab-project-pages-domain-list` command line option, [698](#)

`gitlab-project-pages-domain-update` command line option, [698](#)

`gitlab-project-pipeline-bridge-list` command line option, [701](#)

`gitlab-project-pipeline-cancel` command line option, [699](#)

`gitlab-project-pipeline-create` command line option, [699](#)

`gitlab-project-pipeline-delete` command line option, [699](#)

`gitlab-project-pipeline-get` command line option, [700](#)

`gitlab-project-pipeline-job-list` command line option, [702](#)

`gitlab-project-pipeline-list` command line option, [700](#)

`gitlab-project-pipeline-retry` command line option, [701](#)

`gitlab-project-pipeline-schedule-create` command line option, [703](#)

`gitlab-project-pipeline-schedule-delete` command line option, [703](#)

`gitlab-project-pipeline-schedule-get` command line option, [703](#)

`gitlab-project-pipeline-schedule-list` command line option, [704](#)

`gitlab-project-pipeline-schedule-play` command line option, [704](#)

`gitlab-project-pipeline-schedule-take-ownership` command line option, [704](#)

`gitlab-project-pipeline-schedule-update` command line option, [705](#)

`gitlab-project-pipeline-schedule-variable-create` command line option, [705](#)

`gitlab-project-pipeline-schedule-variable-delete` command line option, [706](#)

`gitlab-project-pipeline-schedule-variable-update` command line option, [706](#)

`gitlab-project-pipeline-test-report-get` command line option, [707](#)

`gitlab-project-pipeline-test-report-summary-get` command line option, [707](#)

`gitlab-project-pipeline-variable-list` command line option, [708](#)

`gitlab-project-protected-branch-create` command line option, [709](#)

`gitlab-project-protected-branch-delete` command line option, [709](#)

`gitlab-project-protected-branch-get` command line option, [709](#)

`gitlab-project-protected-branch-list` command line option, [710](#)

`gitlab-project-protected-environment-create` command line option, [710](#)

`gitlab-project-protected-environment-delete` command line option, [711](#)

`gitlab-project-protected-environment-get` command line option, [711](#)

`gitlab-project-protected-environment-list` command line option, [711](#)

`gitlab-project-protected-tag-create` command line option, [712](#)

`gitlab-project-protected-tag-delete` command line option, [712](#)

`gitlab-project-protected-tag-get` command line option, [713](#)

`gitlab-project-protected-tag-list` command line option, [713](#)

`gitlab-project-push-rules-create` command line option, [714](#)

`gitlab-project-push-rules-delete` command line option, [714](#)

`gitlab-project-push-rules-get` command line option, [715](#)

`gitlab-project-push-rules-update` command line option, [715](#)

gitlab-project-registry-repository-delete  
command line option, [716](#)

gitlab-project-registry-repository-list  
command line option, [716](#)

gitlab-project-registry-tag-delete  
command line option, [717](#)

gitlab-project-registry-tag-delete-in-bulk  
command line option, [717](#)

gitlab-project-registry-tag-get command  
line option, [718](#)

gitlab-project-registry-tag-list  
command line option, [718](#)

gitlab-project-release-create command  
line option, [719](#)

gitlab-project-release-delete command  
line option, [719](#)

gitlab-project-release-get command line  
option, [719](#)

gitlab-project-release-link-create  
command line option, [721](#)

gitlab-project-release-link-delete  
command line option, [721](#)

gitlab-project-release-link-get command  
line option, [722](#)

gitlab-project-release-link-list  
command line option, [722](#)

gitlab-project-release-link-update  
command line option, [722](#)

gitlab-project-release-list command  
line option, [720](#)

gitlab-project-release-update command  
line option, [720](#)

gitlab-project-remote-mirror-create  
command line option, [723](#)

gitlab-project-remote-mirror-list  
command line option, [723](#)

gitlab-project-remote-mirror-update  
command line option, [724](#)

gitlab-project-runner-create command  
line option, [724](#)

gitlab-project-runner-delete command  
line option, [725](#)

gitlab-project-runner-list command line  
option, [725](#)

gitlab-project-service-available  
command line option, [726](#)

gitlab-project-service-delete command  
line option, [726](#)

gitlab-project-service-get command line  
option, [726](#)

gitlab-project-service-list command  
line option, [726](#)

gitlab-project-service-update command  
line option, [727](#)

gitlab-project-snippet-award-emoji-create  
command line option, [730](#)

gitlab-project-snippet-award-emoji-delete  
command line option, [730](#)

gitlab-project-snippet-award-emoji-get  
command line option, [731](#)

gitlab-project-snippet-award-emoji-list  
command line option, [731](#)

gitlab-project-snippet-content command  
line option, [727](#)

gitlab-project-snippet-create command  
line option, [728](#)

gitlab-project-snippet-delete command  
line option, [728](#)

gitlab-project-snippet-discussion-create  
command line option, [732](#)

gitlab-project-snippet-discussion-get  
command line option, [732](#)

gitlab-project-snippet-discussion-list  
command line option, [733](#)

gitlab-project-snippet-discussion-note-create  
command line option, [733](#)

gitlab-project-snippet-discussion-note-delete  
command line option, [734](#)

gitlab-project-snippet-discussion-note-get  
command line option, [734](#)

gitlab-project-snippet-discussion-note-update  
command line option, [735](#)

gitlab-project-snippet-get command line  
option, [728](#)

gitlab-project-snippet-list command  
line option, [729](#)

gitlab-project-snippet-note-award-emoji-create  
command line option, [737](#)

gitlab-project-snippet-note-award-emoji-delete  
command line option, [738](#)

gitlab-project-snippet-note-award-emoji-get  
command line option, [738](#)

gitlab-project-snippet-note-award-emoji-list  
command line option, [739](#)

gitlab-project-snippet-note-create  
command line option, [735](#)

gitlab-project-snippet-note-delete  
command line option, [736](#)

gitlab-project-snippet-note-get command  
line option, [736](#)

gitlab-project-snippet-note-list  
command line option, [736](#)

gitlab-project-snippet-note-update  
command line option, [737](#)

gitlab-project-snippet-update command  
line option, [729](#)

gitlab-project-snippet-user-agent-detail  
command line option, [730](#)

`gitlab-project-storage-get` command line option, [739](#)  
`gitlab-project-tag-create` command line option, [740](#)  
`gitlab-project-tag-delete` command line option, [740](#)  
`gitlab-project-tag-get` command line option, [740](#)  
`gitlab-project-tag-list` command line option, [741](#)  
`gitlab-project-trigger-create` command line option, [741](#)  
`gitlab-project-trigger-delete` command line option, [742](#)  
`gitlab-project-trigger-get` command line option, [742](#)  
`gitlab-project-trigger-list` command line option, [742](#)  
`gitlab-project-trigger-update` command line option, [743](#)  
`gitlab-project-update` command line option, [573](#)  
`gitlab-project-user-list` command line option, [743](#)  
`gitlab-project-variable-create` command line option, [744](#)  
`gitlab-project-variable-delete` command line option, [744](#)  
`gitlab-project-variable-get` command line option, [745](#)  
`gitlab-project-variable-list` command line option, [745](#)  
`gitlab-project-variable-update` command line option, [745](#)  
`gitlab-project-wiki-create` command line option, [746](#)  
`gitlab-project-wiki-delete` command line option, [746](#)  
`gitlab-project-wiki-get` command line option, [747](#)  
`gitlab-project-wiki-list` command line option, [747](#)  
`gitlab-project-wiki-update` command line option, [747](#)  
`gitlab-registry-repository-get` command line option, [748](#)  
`gitlab-runner-all-list` command line option, [751](#)  
`gitlab-runner-create` command line option, [749](#)  
`gitlab-runner-delete` command line option, [749](#)  
`gitlab-runner-get` command line option, [749](#)  
`gitlab-runner-job-list` command line option, [752](#)  
`gitlab-runner-list` command line option, [750](#)  
`gitlab-runner-update` command line option, [750](#)  
`gitlab-shared-project-list` command line option, [753](#)  
`gitlab-snippet-create` command line option, [754](#)  
`gitlab-snippet-delete` command line option, [754](#)  
`gitlab-snippet-get` command line option, [755](#)  
`gitlab-snippet-list` command line option, [755](#)  
`gitlab-snippet-update` command line option, [755](#)  
`gitlab-starred-project-list` command line option, [756](#)  
`gitlab-todo-delete` command line option, [757](#)  
`gitlab-todo-list` command line option, [758](#)  
`gitlab-topic-create` command line option, [759](#)  
`gitlab-topic-delete` command line option, [759](#)  
`gitlab-topic-get` command line option, [759](#)  
`gitlab-topic-list` command line option, [760](#)  
`gitlab-topic-update` command line option, [760](#)  
`gitlab-user-activities-list` command line option, [767](#)  
`gitlab-user-create` command line option, [762](#)  
`gitlab-user-custom-attribute-delete` command line option, [767](#)  
`gitlab-user-custom-attribute-get` command line option, [768](#)  
`gitlab-user-custom-attribute-list` command line option, [768](#)  
`gitlab-user-delete` command line option, [763](#)  
`gitlab-user-email-create` command line option, [769](#)  
`gitlab-user-email-delete` command line option, [769](#)  
`gitlab-user-email-get` command line option, [769](#)  
`gitlab-user-email-list` command line option, [769](#)  
`gitlab-user-event-list` command line option, [770](#)

---

```

gitlab-user-get command line option, 764
gitlab-user-gpg-key-create command line
  option, 771
gitlab-user-gpg-key-delete command line
  option, 771
gitlab-user-gpg-key-get command line
  option, 771
gitlab-user-gpg-key-list command line
  option, 772
gitlab-user-impersonation-token-create
  command line option, 772
gitlab-user-impersonation-token-delete
  command line option, 773
gitlab-user-impersonation-token-get
  command line option, 773
gitlab-user-impersonation-token-list
  command line option, 773
gitlab-user-key-create command line
  option, 774
gitlab-user-key-delete command line
  option, 774
gitlab-user-key-get command line option,
  774
gitlab-user-key-list command line
  option, 775
gitlab-user-list command line option, 764
gitlab-user-membership-get command line
  option, 775
gitlab-user-membership-list command
  line option, 775
gitlab-user-personal-access-token-create
  command line option, 776
gitlab-user-project-create command line
  option, 777
gitlab-user-project-list command line
  option, 778
gitlab-user-status-get command line
  option, 779
gitlab-user-update command line option,
  766
gitlab-variable-create command line
  option, 780
gitlab-variable-delete command line
  option, 780
gitlab-variable-get command line option,
  780
gitlab-variable-list command line
  option, 781
gitlab-variable-update command line
  option, 781
--suggestion-commit-message
  gitlab-project-update command line
    option, 575
--tag
  gitlab-project-deployment-create
    command line option, 607
--tag-list
  gitlab-group-runner-list command line
    option, 543
  gitlab-project-create command line
    option, 563
  gitlab-project-runner-list command line
    option, 725
  gitlab-project-update command line
    option, 575
  gitlab-runner-all-list command line
    option, 751
  gitlab-runner-create command line
    option, 749
  gitlab-runner-list command line option,
    750
  gitlab-runner-update command line
    option, 750
--tag-name
  gitlab-project-release-create command
    line option, 719
  gitlab-project-release-delete command
    line option, 719
  gitlab-project-release-get command line
    option, 719
  gitlab-project-release-link-create
    command line option, 721
  gitlab-project-release-link-delete
    command line option, 721
  gitlab-project-release-link-get command
    line option, 722
  gitlab-project-release-link-list
    command line option, 722
  gitlab-project-release-link-update
    command line option, 723
  gitlab-project-release-update command
    line option, 720
  gitlab-project-tag-create command line
    option, 740
--tag-push-events
  gitlab-group-hook-create command line
    option, 519
  gitlab-group-hook-update command line
    option, 521
  gitlab-project-hook-create command line
    option, 620
  gitlab-project-hook-update command line
    option, 622
--target-access-levels
  gitlab-broadcast-message-create command
    line option, 467
  gitlab-broadcast-message-update command
    line option, 468

```

```
--target-branch
  gitlab-group-merge-request-list command
    line option, 534
  gitlab-merge-request-list command line
    option, 554
  gitlab-project-deployment-merge-request-list
    command line option, 610
  gitlab-project-merge-request-create
    command line option, 662
  gitlab-project-merge-request-list
    command line option, 665
  gitlab-project-merge-request-update
    command line option, 669
--target-issue-iid
  gitlab-project-issue-link-create
    command line option, 640
--target-project-id
  gitlab-project-issue-link-create
    command line option, 640
  gitlab-project-merge-request-create
    command line option, 663
--target-type
  gitlab-event-list command line option,
    476
  gitlab-project-event-list command line
    option, 614
  gitlab-user-event-list command line
    option, 770
--target-url
  gitlab-project-commit-status-create
    command line option, 603
--tasks-project-id
  gitlab-group-invitation-create command
    line option, 523
  gitlab-project-invitation-create
    command line option, 625
--tasks-to-be-done
  gitlab-group-invitation-create command
    line option, 523
  gitlab-group-member-create command line
    option, 530
  gitlab-project-invitation-create
    command line option, 625
  gitlab-project-member-create command
    line option, 657
--template-name
  gitlab-project-create command line
    option, 563
--template-project-id
  gitlab-project-create command line
    option, 563
--terminal-max-session-time
  gitlab-application-settings-update
    command line option, 465
--terms
  gitlab-application-settings-update
    command line option, 465
--theme-id
  gitlab-user-create command line option,
    763
  gitlab-user-update command line option,
    766
--timeout
  gitlab command line option, 459
--title
  gitlab-application-appearance-update
    command line option, 461
  gitlab-current-user-key-create command
    line option, 472
  gitlab-group-epic-create command line
    option, 506
  gitlab-group-epic-update command line
    option, 508
  gitlab-group-milestone-create command
    line option, 534
  gitlab-group-milestone-update command
    line option, 536
  gitlab-group-wiki-create command line
    option, 546
  gitlab-group-wiki-update command line
    option, 548
  gitlab-project-issue-create command
    line option, 628
  gitlab-project-issue-update command
    line option, 634
  gitlab-project-key-create command line
    option, 652
  gitlab-project-key-update command line
    option, 654
  gitlab-project-merge-request-create
    command line option, 663
  gitlab-project-merge-request-update
    command line option, 669
  gitlab-project-milestone-create command
    line option, 690
  gitlab-project-milestone-update command
    line option, 692
  gitlab-project-snippet-create command
    line option, 728
  gitlab-project-snippet-update command
    line option, 729
  gitlab-project-wiki-create command line
    option, 746
  gitlab-project-wiki-update command line
    option, 747
  gitlab-snippet-create command line
    option, 754
  gitlab-snippet-update command line
```



- option, [755](#)
- gitlab-topic-create command line option, [759](#)
- gitlab-user-key-create command line option, [774](#)
- to
  - gitlab-project-repository-compare command line option, [567](#)
- to-namespace
  - gitlab-project-transfer command line option, [570](#)
  - gitlab-project-transfer-project command line option, [570](#)
- to-project-id
  - gitlab-project-issue-move command line option, [630](#)
- token
  - gitlab-group-hook-create command line option, [520](#)
  - gitlab-group-hook-update command line option, [522](#)
  - gitlab-project-hook-create command line option, [620](#)
  - gitlab-project-hook-update command line option, [622](#)
  - gitlab-project-trigger-pipeline command line option, [570](#)
  - gitlab-runner-create command line option, [749](#)
  - gitlab-runner-verify command line option, [751](#)
- top-level-only
  - gitlab-group-list command line option, [486](#)
- topic
  - gitlab-project-list command line option, [566](#)
- topics
  - gitlab-project-create command line option, [563](#)
  - gitlab-project-update command line option, [575](#)
- trailers
  - gitlab-project-commit-list command line option, [597](#)
- twitter
  - gitlab-user-create command line option, [762](#)
  - gitlab-user-update command line option, [766](#)
- two-factor
  - gitlab-user-list command line option, [764](#)
- two-factor-grace-period
  - gitlab-group-create command line option, [484](#)
  - gitlab-group-update command line option, [488](#)
- type
  - gitlab-project-commit-refs command line option, [597](#)
  - gitlab-runner-all-list command line option, [751](#)
  - gitlab-runner-list command line option, [750](#)
  - gitlab-todo-list command line option, [758](#)
  - gitlab-user-membership-list command line option, [776](#)
- unprotect-access-level
  - gitlab-project-protected-branch-create command line option, [709](#)
- until
  - gitlab-project-commit-list command line option, [597](#)
- updated-after
  - gitlab-group-issue-list command line option, [525](#)
  - gitlab-group-merge-request-list command line option, [533](#)
  - gitlab-issue-list command line option, [550](#)
  - gitlab-merge-request-list command line option, [554](#)
  - gitlab-project-deployment-list command line option, [608](#)
  - gitlab-project-deployment-merge-request-list command line option, [610](#)
  - gitlab-project-issue-list command line option, [630](#)
  - gitlab-project-merge-request-list command line option, [664](#)
- updated-at
  - gitlab-application-settings-update command line option, [464](#)
  - gitlab-project-issue-update command line option, [634](#)
- updated-before
  - gitlab-group-issue-list command line option, [525](#)
  - gitlab-group-merge-request-list command line option, [533](#)
  - gitlab-issue-list command line option, [550](#)
  - gitlab-merge-request-list command line option, [554](#)
  - gitlab-project-deployment-list command line option, [608](#)
  - gitlab-project-deployment-merge-request-list command line option, [610](#)

```
gitlab-project-issue-list command line
  option, 630
gitlab-project-merge-request-list
  command line option, 664
--url
  gitlab-geo-node-update command line
    option, 481
  gitlab-group-hook-create command line
    option, 519
  gitlab-group-hook-update command line
    option, 521
  gitlab-hook-create command line option,
    548
  gitlab-project-hook-create command line
    option, 620
  gitlab-project-hook-update command line
    option, 622
  gitlab-project-release-link-create
    command line option, 721
  gitlab-project-release-link-update
    command line option, 723
  gitlab-project-remote-mirror-create
    command line option, 723
--use-custom-template
  gitlab-project-create command line
    option, 563
--user-agent
  gitlab command line option, 459
--user-id
  gitlab-group-billable-member-membership-list
    command line option, 496
  gitlab-group-member-create command line
    option, 530
  gitlab-personal-access-token-list
    command line option, 558
  gitlab-project-member-create command
    line option, 657
  gitlab-starred-project-list command
    line option, 756
  gitlab-user-custom-attribute-delete
    command line option, 767
  gitlab-user-custom-attribute-get
    command line option, 768
  gitlab-user-custom-attribute-list
    command line option, 768
  gitlab-user-email-create command line
    option, 769
  gitlab-user-email-delete command line
    option, 769
  gitlab-user-email-get command line
    option, 769
  gitlab-user-email-list command line
    option, 769
  gitlab-user-event-list command line
    option, 770
  gitlab-user-gpg-key-create command line
    option, 771
  gitlab-user-gpg-key-delete command line
    option, 771
  gitlab-user-gpg-key-get command line
    option, 771
  gitlab-user-gpg-key-list command line
    option, 772
  gitlab-user-impersonation-token-create
    command line option, 772
  gitlab-user-impersonation-token-delete
    command line option, 773
  gitlab-user-impersonation-token-get
    command line option, 773
  gitlab-user-impersonation-token-list
    command line option, 773
  gitlab-user-key-create command line
    option, 774
  gitlab-user-key-delete command line
    option, 774
  gitlab-user-key-get command line option,
    774
  gitlab-user-key-list command line
    option, 775
  gitlab-user-membership-get command line
    option, 775
  gitlab-user-membership-list command
    line option, 776
  gitlab-user-personal-access-token-create
    command line option, 776
  gitlab-user-project-create command line
    option, 777
  gitlab-user-project-list command line
    option, 778
  gitlab-user-status-get command line
    option, 779
--user-ids
  gitlab-project-approval-rule-create
    command line option, 580
  gitlab-project-merge-request-approval-rule-create
    command line option, 671
  gitlab-project-merge-request-approval-rule-update
    command line option, 673
--user-oauth-applications
  gitlab-application-settings-update
    command line option, 465
--user-show-add-ssh-key-message
  gitlab-application-settings-update
    command line option, 465
--username
  gitlab-group-deploy-token-create
    command line option, 503
  gitlab-project-deploy-token-create
```



- command line option, [605](#)
- gitlab-project-pipeline-list command line option, [700](#)
- gitlab-user-create command line option, [762](#)
- gitlab-user-list command line option, [764](#)
- gitlab-user-update command line option, [766](#)
- value
  - gitlab-group-variable-create command line option, [544](#)
  - gitlab-group-variable-update command line option, [546](#)
  - gitlab-project-pipeline-schedule-variable-create command line option, [706](#)
  - gitlab-project-pipeline-schedule-variable-update command line option, [706](#)
  - gitlab-project-variable-create command line option, [744](#)
  - gitlab-project-variable-update command line option, [746](#)
  - gitlab-variable-create command line option, [780](#)
  - gitlab-variable-update command line option, [781](#)
- variable-type
  - gitlab-group-variable-create command line option, [545](#)
  - gitlab-group-variable-update command line option, [546](#)
  - gitlab-project-variable-create command line option, [744](#)
  - gitlab-project-variable-update command line option, [746](#)
  - gitlab-variable-create command line option, [780](#)
  - gitlab-variable-update command line option, [781](#)
- verbose
  - gitlab command line option, [458](#)
- version
  - gitlab command line option, [458](#)
- view
  - gitlab-group-merge-request-list command line option, [533](#)
  - gitlab-merge-request-list command line option, [554](#)
  - gitlab-project-deployment-merge-request-list command line option, [610](#)
  - gitlab-project-merge-request-list command line option, [664](#)
- visibility
  - gitlab-group-create command line option, [484](#)
  - gitlab-group-project-list command line option, [539](#)
  - gitlab-group-update command line option, [488](#)
  - gitlab-project-create command line option, [563](#)
  - gitlab-project-fork-list command line option, [619](#)
  - gitlab-project-list command line option, [566](#)
  - gitlab-project-snippet-create command line option, [728](#)
  - gitlab-project-snippet-update command line option, [729](#)
  - gitlab-project-update command line option, [575](#)
  - gitlab-shared-project-list command line option, [753](#)
  - gitlab-snippet-create command line option, [754](#)
  - gitlab-snippet-update command line option, [756](#)
  - gitlab-starred-project-list command line option, [757](#)
  - gitlab-user-project-create command line option, [777](#)
  - gitlab-user-project-list command line option, [778](#)
- wall-enabled
  - gitlab-user-project-create command line option, [777](#)
- website-url
  - gitlab-user-create command line option, [762](#)
  - gitlab-user-update command line option, [766](#)
- wiki
  - gitlab-project-snapshot command line option, [569](#)
- wiki-access-level
  - gitlab-project-create command line option, [564](#)
  - gitlab-project-update command line option, [575](#)
- wiki-checksum-failed
  - gitlab-project-list command line option, [566](#)
- wiki-enabled
  - gitlab-user-project-list command line option, [778](#)
  - gitlab-project-create command line option, [564](#)
  - gitlab-project-update command line option, [575](#)

```
gitlab-user-project-create command line
  option, 777
--wiki-events
  gitlab-project-hook-update command line
    option, 622
--wiki-page-events
  gitlab-group-hook-create command line
    option, 519
  gitlab-group-hook-update command line
    option, 521
  gitlab-project-hook-create command line
    option, 620
--wip
  gitlab-group-merge-request-list command
    line option, 534
  gitlab-merge-request-list command line
    option, 555
  gitlab-project-deployment-merge-request-list
    command line option, 610
  gitlab-project-merge-request-list
    command line option, 665
--with-content
  gitlab-group-wiki-list command line
    option, 547
  gitlab-project-wiki-list command line
    option, 747
--with-custom-attributes
  gitlab-group-descendant-group-list
    command line option, 505
  gitlab-group-list command line option,
    486
  gitlab-group-project-list command line
    option, 540
  gitlab-group-subgroup-list command line
    option, 544
  gitlab-project-fork-list command line
    option, 619
  gitlab-project-list command line option,
    566
  gitlab-shared-project-list command line
    option, 753
  gitlab-starred-project-list command
    line option, 757
  gitlab-user-project-list command line
    option, 778
--with-issues-enabled
  gitlab-group-project-list command line
    option, 540
  gitlab-project-fork-list command line
    option, 619
  gitlab-project-list command line option,
    566
  gitlab-shared-project-list command line
    option, 753
  gitlab-starred-project-list command
    line option, 757
  gitlab-user-project-list command line
    option, 778
--with-labels-details
  gitlab-merge-request-list command line
    option, 554
  gitlab-project-deployment-merge-request-list
    command line option, 610
--with-merge-requests-enabled
  gitlab-group-project-list command line
    option, 540
  gitlab-project-fork-list command line
    option, 619
  gitlab-project-list command line option,
    566
  gitlab-shared-project-list command line
    option, 753
  gitlab-starred-project-list command
    line option, 757
  gitlab-user-project-list command line
    option, 778
--with-merge-status-recheck
  gitlab-merge-request-list command line
    option, 554
  gitlab-project-deployment-merge-request-list
    command line option, 610
--with-programming-language
  gitlab-project-list command line option,
    566
  gitlab-user-project-list command line
    option, 778
--with-security-reports
  gitlab-group-project-list command line
    option, 540
--with-shared
  gitlab-group-project-list command line
    option, 540
--with-stats
  gitlab-project-commit-list command line
    option, 597
--yaml-errors
  gitlab-project-pipeline-list command
    line option, 700
-c
  gitlab command line option, 458
-d
  gitlab command line option, 458
-f
  gitlab command line option, 458
-g
  gitlab command line option, 458
-h
  gitlab command line option, 458
```

gitlab-application command line option, 459

gitlab-application-appearance command line option, 460

gitlab-application-appearance-get command line option, 461

gitlab-application-appearance-update command line option, 461

gitlab-application-create command line option, 459

gitlab-application-delete command line option, 460

gitlab-application-list command line option, 460

gitlab-application-settings command line option, 462

gitlab-application-settings-get command line option, 462

gitlab-application-settings-update command line option, 463

gitlab-audit-event command line option, 466

gitlab-audit-event-get command line option, 466

gitlab-audit-event-list command line option, 466

gitlab-broadcast-message command line option, 467

gitlab-broadcast-message-create command line option, 467

gitlab-broadcast-message-delete command line option, 467

gitlab-broadcast-message-get command line option, 467

gitlab-broadcast-message-list command line option, 468

gitlab-broadcast-message-update command line option, 468

gitlab-ci-lint command line option, 469

gitlab-ci-lint-create command line option, 469

gitlab-ci-lint-validate command line option, 469

gitlab-current-user command line option, 469

gitlab-current-user-email command line option, 470

gitlab-current-user-email-create command line option, 470

gitlab-current-user-email-delete command line option, 470

gitlab-current-user-email-get command line option, 470

gitlab-current-user-email-list command line option, 471

gitlab-current-user-get command line option, 470

gitlab-current-user-gpg-key command line option, 471

gitlab-current-user-gpg-key-create command line option, 471

gitlab-current-user-gpg-key-delete command line option, 471

gitlab-current-user-gpg-key-get command line option, 472

gitlab-current-user-gpg-key-list command line option, 472

gitlab-current-user-key command line option, 472

gitlab-current-user-key-create command line option, 472

gitlab-current-user-key-delete command line option, 473

gitlab-current-user-key-get command line option, 473

gitlab-current-user-key-list command line option, 473

gitlab-current-user-status command line option, 473

gitlab-current-user-status-get command line option, 474

gitlab-current-user-status-update command line option, 474

gitlab-deploy-key command line option, 474

gitlab-deploy-key-list command line option, 474

gitlab-deploy-token command line option, 475

gitlab-deploy-token-list command line option, 475

gitlab-dockerfile command line option, 475

gitlab-dockerfile-get command line option, 475

gitlab-dockerfile-list command line option, 476

gitlab-event command line option, 476

gitlab-event-list command line option, 476

gitlab-feature command line option, 477

gitlab-feature-delete command line option, 477

gitlab-feature-list command line option, 477

gitlab-generic-package command line option, 478

gitlab-generic-package-download command

- line option, [478](#)
- gitlab-generic-package-upload command
  - line option, [478](#)
- gitlab-geo-node command line option, [479](#)
- gitlab-geo-node-current-failures
  - command line option, [479](#)
- gitlab-geo-node-delete command line option, [479](#)
- gitlab-geo-node-get command line option, [479](#)
- gitlab-geo-node-list command line option, [480](#)
- gitlab-geo-node-repair command line option, [480](#)
- gitlab-geo-node-status command line option, [480](#)
- gitlab-geo-node-update command line option, [480](#)
- gitlab-gitignore command line option, [481](#)
- gitlab-gitignore-get command line option, [481](#)
- gitlab-gitignore-list command line option, [481](#)
- gitlab-gitlabciyaml command line option, [482](#)
- gitlab-gitlabciyaml-get command line option, [482](#)
- gitlab-gitlabciyaml-list command line option, [482](#)
- gitlab-group command line option, [483](#)
- gitlab-group-access-request command line option, [489](#)
- gitlab-group-access-request-approve command line option, [489](#)
- gitlab-group-access-request-create command line option, [489](#)
- gitlab-group-access-request-delete command line option, [489](#)
- gitlab-group-access-request-list command line option, [490](#)
- gitlab-group-access-token command line option, [490](#)
- gitlab-group-access-token-create command line option, [490](#)
- gitlab-group-access-token-delete command line option, [491](#)
- gitlab-group-access-token-list command line option, [491](#)
- gitlab-group-add-ldap-group-link command line option, [483](#)
- gitlab-group-audit-event command line option, [491](#)
- gitlab-group-audit-event-get command line option, [492](#)
- gitlab-group-audit-event-list command line option, [492](#)
- gitlab-group-badge command line option, [492](#)
- gitlab-group-badge-create command line option, [493](#)
- gitlab-group-badge-delete command line option, [493](#)
- gitlab-group-badge-get command line option, [493](#)
- gitlab-group-badge-list command line option, [493](#)
- gitlab-group-badge-render command line option, [494](#)
- gitlab-group-badge-update command line option, [494](#)
- gitlab-group-billable-member command line option, [495](#)
- gitlab-group-billable-member-delete command line option, [495](#)
- gitlab-group-billable-member-list command line option, [495](#)
- gitlab-group-billable-member-membership command line option, [496](#)
- gitlab-group-billable-member-membership-list command line option, [496](#)
- gitlab-group-board command line option, [496](#)
- gitlab-group-board-create command line option, [496](#)
- gitlab-group-board-delete command line option, [497](#)
- gitlab-group-board-get command line option, [497](#)
- gitlab-group-board-list command line option, [497](#), [498](#)
- gitlab-group-board-list-create command line option, [498](#)
- gitlab-group-board-list-delete command line option, [498](#)
- gitlab-group-board-list-get command line option, [499](#)
- gitlab-group-board-list-list command line option, [499](#)
- gitlab-group-board-list-update command line option, [499](#)
- gitlab-group-board-update command line option, [498](#)
- gitlab-group-cluster command line option, [500](#)
- gitlab-group-cluster-create command line option, [500](#)
- gitlab-group-cluster-delete command line option, [500](#)

gitlab-group-cluster-get command line option, [501](#)  
gitlab-group-cluster-list command line option, [501](#)  
gitlab-group-cluster-update command line option, [501](#)  
gitlab-group-create command line option, [483](#)  
gitlab-group-custom-attribute command line option, [502](#)  
gitlab-group-custom-attribute-delete command line option, [502](#)  
gitlab-group-custom-attribute-get command line option, [502](#)  
gitlab-group-custom-attribute-list command line option, [503](#)  
gitlab-group-delete command line option, [484](#)  
gitlab-group-delete-ldap-group-link command line option, [485](#)  
gitlab-group-deploy-token command line option, [503](#)  
gitlab-group-deploy-token-create command line option, [503](#)  
gitlab-group-deploy-token-delete command line option, [504](#)  
gitlab-group-deploy-token-get command line option, [504](#)  
gitlab-group-deploy-token-list command line option, [504](#)  
gitlab-group-descendant-group command line option, [505](#)  
gitlab-group-descendant-group-list command line option, [505](#)  
gitlab-group-epic command line option, [506](#)  
gitlab-group-epic-award-emoji command line option, [508](#)  
gitlab-group-epic-award-emoji-create command line option, [508](#)  
gitlab-group-epic-award-emoji-delete command line option, [509](#)  
gitlab-group-epic-award-emoji-get command line option, [509](#)  
gitlab-group-epic-award-emoji-list command line option, [509](#)  
gitlab-group-epic-create command line option, [506](#)  
gitlab-group-epic-delete command line option, [506](#)  
gitlab-group-epic-discussion-note command line option, [510](#)  
gitlab-group-epic-discussion-note-create command line option, [510](#)  
gitlab-group-epic-discussion-note-delete command line option, [510](#)  
gitlab-group-epic-discussion-note-get command line option, [511](#)  
gitlab-group-epic-discussion-note-update command line option, [511](#)  
gitlab-group-epic-get command line option, [507](#)  
gitlab-group-epic-issue command line option, [512](#)  
gitlab-group-epic-issue-create command line option, [512](#)  
gitlab-group-epic-issue-delete command line option, [512](#)  
gitlab-group-epic-issue-list command line option, [512](#)  
gitlab-group-epic-issue-update command line option, [513](#)  
gitlab-group-epic-list command line option, [507](#)  
gitlab-group-epic-note command line option, [513](#)  
gitlab-group-epic-note-award-emoji command line option, [515](#)  
gitlab-group-epic-note-award-emoji-create command line option, [515](#)  
gitlab-group-epic-note-award-emoji-delete command line option, [516](#)  
gitlab-group-epic-note-award-emoji-get command line option, [516](#)  
gitlab-group-epic-note-award-emoji-list command line option, [516](#)  
gitlab-group-epic-note-create command line option, [513](#)  
gitlab-group-epic-note-delete command line option, [514](#)  
gitlab-group-epic-note-get command line option, [514](#)  
gitlab-group-epic-note-list command line option, [514](#)  
gitlab-group-epic-note-update command line option, [515](#)  
gitlab-group-epic-resource-label-event command line option, [517](#)  
gitlab-group-epic-resource-label-event-get command line option, [517](#)  
gitlab-group-epic-resource-label-event-list command line option, [517](#)  
gitlab-group-epic-update command line option, [508](#)  
gitlab-group-export command line option, [518](#)  
gitlab-group-export-create command line option, [518](#)

gitlab-group-export-download command line option, [518](#)  
gitlab-group-export-get command line option, [518](#)  
gitlab-group-get command line option, [485](#)  
gitlab-group-hook command line option, [519](#)  
gitlab-group-hook-create command line option, [519](#)  
gitlab-group-hook-delete command line option, [520](#)  
gitlab-group-hook-get command line option, [520](#)  
gitlab-group-hook-list command line option, [520](#)  
gitlab-group-hook-update command line option, [521](#)  
gitlab-group-import command line option, [522](#)  
gitlab-group-import-get command line option, [522](#)  
gitlab-group-invitation command line option, [522](#)  
gitlab-group-invitation-create command line option, [522](#)  
gitlab-group-invitation-delete command line option, [523](#)  
gitlab-group-invitation-get command line option, [523](#)  
gitlab-group-invitation-list command line option, [523](#)  
gitlab-group-invitation-update command line option, [524](#)  
gitlab-group-issue command line option, [524](#)  
gitlab-group-issue-list command line option, [524](#)  
gitlab-group-issues-statistics command line option, [525](#)  
gitlab-group-issues-statistics-get command line option, [526](#)  
gitlab-group-iteration command line option, [526](#)  
gitlab-group-iteration-list command line option, [526](#)  
gitlab-group-label command line option, [527](#)  
gitlab-group-label-create command line option, [527](#)  
gitlab-group-label-delete command line option, [527](#)  
gitlab-group-label-get command line option, [528](#)  
gitlab-group-label-list command line option, [528](#)  
gitlab-group-label-subscribe command line option, [528](#)  
gitlab-group-label-unsubscribe command line option, [529](#)  
gitlab-group-label-update command line option, [529](#)  
gitlab-group-ldap-sync command line option, [485](#)  
gitlab-group-list command line option, [485](#)  
gitlab-group-member command line option, [529](#)  
gitlab-group-member-all command line option, [531](#)  
gitlab-group-member-all-get command line option, [532](#)  
gitlab-group-member-all-list command line option, [532](#)  
gitlab-group-member-create command line option, [530](#)  
gitlab-group-member-delete command line option, [530](#)  
gitlab-group-member-get command line option, [530](#)  
gitlab-group-member-list command line option, [531](#)  
gitlab-group-member-update command line option, [531](#)  
gitlab-group-merge-request command line option, [532](#)  
gitlab-group-merge-request-list command line option, [533](#)  
gitlab-group-milestone command line option, [534](#)  
gitlab-group-milestone-create command line option, [534](#)  
gitlab-group-milestone-delete command line option, [535](#)  
gitlab-group-milestone-get command line option, [535](#)  
gitlab-group-milestone-issues command line option, [535](#)  
gitlab-group-milestone-list command line option, [535](#)  
gitlab-group-milestone-merge-requests command line option, [536](#)  
gitlab-group-milestone-update command line option, [536](#)  
gitlab-group-notification-settings command line option, [537](#)  
gitlab-group-notification-settings-get command line option, [537](#)  
gitlab-group-notification-settings-update

- command line option, [537](#)
- gitlab-group-package command line option, [538](#)
- gitlab-group-package-list command line option, [538](#)
- gitlab-group-project command line option, [539](#)
- gitlab-group-project-list command line option, [539](#)
- gitlab-group-push-rules command line option, [540](#)
- gitlab-group-push-rules-create command line option, [540](#)
- gitlab-group-push-rules-delete command line option, [541](#)
- gitlab-group-push-rules-get command line option, [541](#)
- gitlab-group-push-rules-update command line option, [542](#)
- gitlab-group-runner command line option, [542](#)
- gitlab-group-runner-list command line option, [543](#)
- gitlab-group-search command line option, [486](#)
- gitlab-group-share command line option, [486](#)
- gitlab-group-subgroup command line option, [543](#)
- gitlab-group-subgroup-list command line option, [543](#)
- gitlab-group-transfer command line option, [487](#)
- gitlab-group-transfer-project command line option, [487](#)
- gitlab-group-unshare command line option, [487](#)
- gitlab-group-update command line option, [488](#)
- gitlab-group-variable command line option, [544](#)
- gitlab-group-variable-create command line option, [544](#)
- gitlab-group-variable-delete command line option, [545](#)
- gitlab-group-variable-get command line option, [545](#)
- gitlab-group-variable-list command line option, [545](#)
- gitlab-group-variable-update command line option, [546](#)
- gitlab-group-wiki command line option, [546](#)
- gitlab-group-wiki-create command line option, [546](#)
- gitlab-group-wiki-delete command line option, [547](#)
- gitlab-group-wiki-get command line option, [547](#)
- gitlab-group-wiki-list command line option, [547](#)
- gitlab-group-wiki-update command line option, [548](#)
- gitlab-hook command line option, [548](#)
- gitlab-hook-create command line option, [548](#)
- gitlab-hook-delete command line option, [548](#)
- gitlab-hook-get command line option, [549](#)
- gitlab-hook-list command line option, [549](#)
- gitlab-issue command line option, [549](#)
- gitlab-issue-get command line option, [549](#)
- gitlab-issue-list command line option, [550](#)
- gitlab-issues-statistics command line option, [551](#)
- gitlab-issues-statistics-get command line option, [551](#)
- gitlab-key command line option, [551](#)
- gitlab-key-get command line option, [551](#)
- gitlab-ldap-group command line option, [551](#)
- gitlab-ldap-group-list command line option, [552](#)
- gitlab-license command line option, [552](#)
- gitlab-license-get command line option, [552](#)
- gitlab-license-list command line option, [553](#)
- gitlab-merge-request command line option, [553](#)
- gitlab-merge-request-list command line option, [554](#)
- gitlab-namespace command line option, [555](#)
- gitlab-namespace-exists command line option, [555](#)
- gitlab-namespace-get command line option, [555](#)
- gitlab-namespace-list command line option, [556](#)
- gitlab-notification-settings command line option, [556](#)
- gitlab-notification-settings-get command line option, [556](#)
- gitlab-notification-settings-update command line option, [557](#)
- gitlab-pages-domain command line option, [557](#)



gitlab-pages-domain-list command line option, [557](#)  
gitlab-personal-access-token command line option, [558](#)  
gitlab-personal-access-token-delete command line option, [558](#)  
gitlab-personal-access-token-list command line option, [558](#)  
gitlab-project command line option, [559](#)  
gitlab-project-access-request command line option, [576](#)  
gitlab-project-access-request-approve command line option, [576](#)  
gitlab-project-access-request-create command line option, [576](#)  
gitlab-project-access-request-delete command line option, [576](#)  
gitlab-project-access-request-list command line option, [577](#)  
gitlab-project-access-token command line option, [577](#)  
gitlab-project-access-token-create command line option, [577](#)  
gitlab-project-access-token-delete command line option, [578](#)  
gitlab-project-access-token-list command line option, [578](#)  
gitlab-project-additional-statistics command line option, [578](#)  
gitlab-project-additional-statistics-get command line option, [579](#)  
gitlab-project-approval command line option, [579](#)  
gitlab-project-approval-get command line option, [579](#)  
gitlab-project-approval-rule command line option, [580](#)  
gitlab-project-approval-rule-create command line option, [580](#)  
gitlab-project-approval-rule-delete command line option, [580](#)  
gitlab-project-approval-rule-list command line option, [581](#)  
gitlab-project-approval-rule-update command line option, [581](#)  
gitlab-project-approval-update command line option, [579](#)  
gitlab-project-archive command line option, [559](#)  
gitlab-project-artifact command line option, [560](#), [581](#)  
gitlab-project-artifact-delete command line option, [582](#)  
gitlab-project-artifact-download command line option, [582](#)  
gitlab-project-artifact-raw command line option, [582](#)  
gitlab-project-artifacts command line option, [560](#)  
gitlab-project-audit-event command line option, [583](#)  
gitlab-project-audit-event-get command line option, [583](#)  
gitlab-project-audit-event-list command line option, [583](#)  
gitlab-project-badge command line option, [584](#)  
gitlab-project-badge-create command line option, [584](#)  
gitlab-project-badge-delete command line option, [584](#)  
gitlab-project-badge-get command line option, [584](#)  
gitlab-project-badge-list command line option, [585](#)  
gitlab-project-badge-render command line option, [585](#)  
gitlab-project-badge-update command line option, [585](#)  
gitlab-project-board command line option, [586](#)  
gitlab-project-board-create command line option, [586](#)  
gitlab-project-board-delete command line option, [586](#)  
gitlab-project-board-get command line option, [586](#)  
gitlab-project-board-list command line option, [587](#)  
gitlab-project-board-list-create command line option, [588](#)  
gitlab-project-board-list-delete command line option, [588](#)  
gitlab-project-board-list-get command line option, [588](#)  
gitlab-project-board-list-list command line option, [589](#)  
gitlab-project-board-list-update command line option, [589](#)  
gitlab-project-board-update command line option, [587](#)  
gitlab-project-branch command line option, [589](#)  
gitlab-project-branch-create command line option, [590](#)  
gitlab-project-branch-delete command line option, [590](#)  
gitlab-project-branch-get command line



- option, [590](#)
- gitlab-project-branch-list command line option, [591](#)
- gitlab-project-ci-lint command line option, [591](#)
- gitlab-project-ci-lint-create command line option, [591](#)
- gitlab-project-ci-lint-get command line option, [592](#)
- gitlab-project-ci-lint-validate command line option, [592](#)
- gitlab-project-cluster command line option, [592](#)
- gitlab-project-cluster-create command line option, [592](#)
- gitlab-project-cluster-delete command line option, [593](#)
- gitlab-project-cluster-get command line option, [593](#)
- gitlab-project-cluster-list command line option, [594](#)
- gitlab-project-cluster-update command line option, [594](#)
- gitlab-project-commit command line option, [595](#)
- gitlab-project-commit-cherry-pick command line option, [595](#)
- gitlab-project-commit-comment command line option, [598](#)
- gitlab-project-commit-comment-create command line option, [598](#)
- gitlab-project-commit-comment-list command line option, [599](#)
- gitlab-project-commit-create command line option, [595](#)
- gitlab-project-commit-diff command line option, [596](#)
- gitlab-project-commit-discussion command line option, [599](#)
- gitlab-project-commit-discussion-create command line option, [599](#)
- gitlab-project-commit-discussion-get command line option, [600](#)
- gitlab-project-commit-discussion-list command line option, [600](#)
- gitlab-project-commit-discussion-note command line option, [601](#)
- gitlab-project-commit-discussion-note-create command line option, [601](#)
- gitlab-project-commit-discussion-note-delete command line option, [601](#)
- gitlab-project-commit-discussion-note-get command line option, [602](#)
- gitlab-project-commit-discussion-note-update command line option, [602](#)
- gitlab-project-commit-get command line option, [596](#)
- gitlab-project-commit-list command line option, [596](#)
- gitlab-project-commit-merge-requests command line option, [597](#)
- gitlab-project-commit-refs command line option, [597](#)
- gitlab-project-commit-revert command line option, [598](#)
- gitlab-project-commit-signature command line option, [598](#)
- gitlab-project-commit-status command line option, [603](#)
- gitlab-project-commit-status-create command line option, [603](#)
- gitlab-project-commit-status-list command line option, [603](#)
- gitlab-project-create command line option, [561](#)
- gitlab-project-create-fork-relation command line option, [564](#)
- gitlab-project-custom-attribute command line option, [604](#)
- gitlab-project-custom-attribute-delete command line option, [604](#)
- gitlab-project-custom-attribute-get command line option, [604](#)
- gitlab-project-custom-attribute-list command line option, [605](#)
- gitlab-project-delete command line option, [564](#)
- gitlab-project-delete-fork-relation command line option, [564](#)
- gitlab-project-delete-merged-branches command line option, [564](#)
- gitlab-project-deploy-token command line option, [605](#)
- gitlab-project-deploy-token-create command line option, [605](#)
- gitlab-project-deploy-token-delete command line option, [606](#)
- gitlab-project-deploy-token-get command line option, [606](#)
- gitlab-project-deploy-token-list command line option, [606](#)
- gitlab-project-deployment command line option, [607](#)
- gitlab-project-deployment-create command line option, [607](#)
- gitlab-project-deployment-get command line option, [607](#)
- gitlab-project-deployment-list command

- line option, [608](#)
- gitlab-project-deployment-merge-request
  - command line option, [609](#)
- gitlab-project-deployment-merge-request-list
  - command line option, [609](#)
- gitlab-project-deployment-update
  - command line option, [608](#)
- gitlab-project-environment command line option, [611](#)
- gitlab-project-environment-create
  - command line option, [611](#)
- gitlab-project-environment-delete
  - command line option, [612](#)
- gitlab-project-environment-get command line option, [612](#)
- gitlab-project-environment-list command line option, [612](#)
- gitlab-project-environment-stop command line option, [613](#)
- gitlab-project-environment-update
  - command line option, [613](#)
- gitlab-project-event command line option, [613](#)
- gitlab-project-event-list command line option, [614](#)
- gitlab-project-export command line option, [614](#)
- gitlab-project-export-create command line option, [614](#)
- gitlab-project-export-download command line option, [615](#)
- gitlab-project-export-get command line option, [615](#)
- gitlab-project-file command line option, [615](#)
- gitlab-project-file-blame command line option, [615](#)
- gitlab-project-file-create command line option, [616](#)
- gitlab-project-file-delete command line option, [616](#)
- gitlab-project-file-get command line option, [617](#)
- gitlab-project-file-raw command line option, [617](#)
- gitlab-project-file-update command line option, [617](#)
- gitlab-project-fork command line option, [618](#)
- gitlab-project-fork-create command line option, [618](#)
- gitlab-project-fork-list command line option, [618](#)
- gitlab-project-get command line option, [565](#)
- gitlab-project-hook command line option, [619](#)
- gitlab-project-hook-create command line option, [620](#)
- gitlab-project-hook-delete command line option, [620](#)
- gitlab-project-hook-get command line option, [621](#)
- gitlab-project-hook-list command line option, [621](#)
- gitlab-project-hook-update command line option, [621](#)
- gitlab-project-housekeeping command line option, [565](#)
- gitlab-project-import command line option, [622](#)
- gitlab-project-import-get command line option, [622](#)
- gitlab-project-integration command line option, [623](#)
- gitlab-project-integration-available
  - command line option, [623](#)
- gitlab-project-integration-delete
  - command line option, [623](#)
- gitlab-project-integration-get command line option, [623](#)
- gitlab-project-integration-list command line option, [624](#)
- gitlab-project-integration-update
  - command line option, [624](#)
- gitlab-project-invitation command line option, [624](#)
- gitlab-project-invitation-create
  - command line option, [625](#)
- gitlab-project-invitation-delete
  - command line option, [625](#)
- gitlab-project-invitation-get command line option, [625](#)
- gitlab-project-invitation-list command line option, [626](#)
- gitlab-project-invitation-update
  - command line option, [626](#)
- gitlab-project-issue command line option, [627](#)
- gitlab-project-issue-add-spent-time
  - command line option, [627](#)
- gitlab-project-issue-award-emoji
  - command line option, [635](#)
- gitlab-project-issue-award-emoji-create
  - command line option, [635](#)
- gitlab-project-issue-award-emoji-delete
  - command line option, [635](#)
- gitlab-project-issue-award-emoji-get

command line option, [636](#)  
gitlab-project-issue-award-emoji-list  
command line option, [636](#)  
gitlab-project-issue-closed-by command  
line option, [628](#)  
gitlab-project-issue-create command  
line option, [628](#)  
gitlab-project-issue-delete command  
line option, [629](#)  
gitlab-project-issue-discussion command  
line option, [636](#)  
gitlab-project-issue-discussion-create  
command line option, [637](#)  
gitlab-project-issue-discussion-get  
command line option, [637](#)  
gitlab-project-issue-discussion-list  
command line option, [637](#)  
gitlab-project-issue-discussion-note  
command line option, [638](#)  
gitlab-project-issue-discussion-note-create  
command line option, [638](#)  
gitlab-project-issue-discussion-note-delete  
command line option, [639](#)  
gitlab-project-issue-discussion-note-get  
command line option, [639](#)  
gitlab-project-issue-discussion-note-update  
command line option, [639](#)  
gitlab-project-issue-get command line  
option, [629](#)  
gitlab-project-issue-link command line  
option, [640](#)  
gitlab-project-issue-link-create  
command line option, [640](#)  
gitlab-project-issue-link-delete  
command line option, [640](#)  
gitlab-project-issue-link-list command  
line option, [641](#)  
gitlab-project-issue-list command line  
option, [629](#)  
gitlab-project-issue-move command line  
option, [630](#)  
gitlab-project-issue-note command line  
option, [641](#)  
gitlab-project-issue-note-award-emoji  
command line option, [643](#)  
gitlab-project-issue-note-award-emoji-create  
command line option, [643](#)  
gitlab-project-issue-note-award-emoji-delete  
command line option, [644](#)  
gitlab-project-issue-note-award-emoji-get  
command line option, [644](#)  
gitlab-project-issue-note-award-emoji-list  
command line option, [644](#)  
gitlab-project-issue-note-create  
command line option, [641](#)  
gitlab-project-issue-note-delete  
command line option, [642](#)  
gitlab-project-issue-note-get command  
line option, [642](#)  
gitlab-project-issue-note-list command  
line option, [642](#)  
gitlab-project-issue-note-update  
command line option, [643](#)  
gitlab-project-issue-participants  
command line option, [631](#)  
gitlab-project-issue-related-merge-requests  
command line option, [631](#)  
gitlab-project-issue-reorder command  
line option, [631](#)  
gitlab-project-issue-reset-spent-time  
command line option, [632](#)  
gitlab-project-issue-reset-time-estimate  
command line option, [632](#)  
gitlab-project-issue-resource-label-event  
command line option, [645](#)  
gitlab-project-issue-resource-label-event-get  
command line option, [645](#)  
gitlab-project-issue-resource-label-event-list  
command line option, [645](#)  
gitlab-project-issue-resource-milestone-event  
command line option, [646](#)  
gitlab-project-issue-resource-milestone-event-get  
command line option, [646](#)  
gitlab-project-issue-resource-milestone-event-list  
command line option, [647](#)  
gitlab-project-issue-resource-state-event  
command line option, [647](#)  
gitlab-project-issue-resource-state-event-get  
command line option, [647](#)  
gitlab-project-issue-resource-state-event-list  
command line option, [648](#)  
gitlab-project-issue-subscribe command  
line option, [632](#)  
gitlab-project-issue-time-estimate  
command line option, [632](#)  
gitlab-project-issue-time-stats command  
line option, [633](#)  
gitlab-project-issue-todo command line  
option, [633](#)  
gitlab-project-issue-unsubscribe  
command line option, [633](#)  
gitlab-project-issue-update command  
line option, [634](#)  
gitlab-project-issue-user-agent-detail  
command line option, [634](#)  
gitlab-project-issues-statistics  
command line option, [648](#)  
gitlab-project-issues-statistics-get

- command line option, [648](#)
- gitlab-project-job command line option, [649](#)
- gitlab-project-job-artifact command line option, [649](#)
- gitlab-project-job-artifacts command line option, [649](#)
- gitlab-project-job-cancel command line option, [650](#)
- gitlab-project-job-delete-artifacts command line option, [650](#)
- gitlab-project-job-erase command line option, [650](#)
- gitlab-project-job-get command line option, [650](#)
- gitlab-project-job-keep-artifacts command line option, [651](#)
- gitlab-project-job-list command line option, [651](#)
- gitlab-project-job-play command line option, [651](#)
- gitlab-project-job-retry command line option, [652](#)
- gitlab-project-job-trace command line option, [652](#)
- gitlab-project-key command line option, [652](#)
- gitlab-project-key-create command line option, [652](#)
- gitlab-project-key-delete command line option, [653](#)
- gitlab-project-key-enable command line option, [653](#)
- gitlab-project-key-get command line option, [653](#)
- gitlab-project-key-list command line option, [654](#)
- gitlab-project-key-update command line option, [654](#)
- gitlab-project-label command line option, [654](#)
- gitlab-project-label-create command line option, [655](#)
- gitlab-project-label-delete command line option, [655](#)
- gitlab-project-label-get command line option, [655](#)
- gitlab-project-label-list command line option, [656](#)
- gitlab-project-label-subscribe command line option, [656](#)
- gitlab-project-label-unsubscribe command line option, [656](#)
- gitlab-project-label-update command

- line option, [657](#)
- gitlab-project-languages command line option, [655](#)
- gitlab-project-list command line option, [655](#)
- gitlab-project-member command line option, [657](#)
- gitlab-project-member-all command line option, [659](#)
- gitlab-project-member-all-get command line option, [659](#)
- gitlab-project-member-all-list command line option, [659](#)
- gitlab-project-member-create command line option, [657](#)
- gitlab-project-member-delete command line option, [658](#)
- gitlab-project-member-get command line option, [658](#)
- gitlab-project-member-list command line option, [658](#)
- gitlab-project-member-update command line option, [659](#)
- gitlab-project-merge-request command line option, [660](#)
- gitlab-project-merge-request-add-spent-time command line option, [660](#)
- gitlab-project-merge-request-approval command line option, [670](#)
- gitlab-project-merge-request-approval-get command line option, [670](#)
- gitlab-project-merge-request-approval-rule command line option, [670](#)
- gitlab-project-merge-request-approval-rule-create command line option, [671](#)
- gitlab-project-merge-request-approval-rule-delete command line option, [671](#)
- gitlab-project-merge-request-approval-rule-get command line option, [672](#)
- gitlab-project-merge-request-approval-rule-list command line option, [672](#)
- gitlab-project-merge-request-approval-rule-update command line option, [673](#)
- gitlab-project-merge-request-approval-state command line option, [673](#)
- gitlab-project-merge-request-approval-state-get command line option, [674](#)
- gitlab-project-merge-request-approval-update command line option, [670](#)
- gitlab-project-merge-request-approve command line option, [661](#)
- gitlab-project-merge-request-award-emoji command line option, [674](#)
- gitlab-project-merge-request-award-emoji-create

- command line option, 674
- gitlab-project-merge-request-award-emoji-delete
  - command line option, 675
- gitlab-project-merge-request-award-emoji-get
  - command line option, 675
- gitlab-project-merge-request-award-emoji-list
  - command line option, 675
- gitlab-project-merge-request-cancel-merge-when-pipeline-fails
  - command line option, 661
- gitlab-project-merge-request-changes
  - command line option, 661
- gitlab-project-merge-request-closes-issues
  - command line option, 662
- gitlab-project-merge-request-commits
  - command line option, 662
- gitlab-project-merge-request-create
  - command line option, 662
- gitlab-project-merge-request-delete
  - command line option, 663
- gitlab-project-merge-request-diff
  - command line option, 676
- gitlab-project-merge-request-diff-get
  - command line option, 676
- gitlab-project-merge-request-diff-list
  - command line option, 676
- gitlab-project-merge-request-discussion
  - command line option, 677
- gitlab-project-merge-request-discussion-create
  - command line option, 677
- gitlab-project-merge-request-discussion-get
  - command line option, 677
- gitlab-project-merge-request-discussion-list
  - command line option, 678
- gitlab-project-merge-request-discussion-note
  - command line option, 679
- gitlab-project-merge-request-discussion-note-create
  - command line option, 679
- gitlab-project-merge-request-discussion-note-delete
  - command line option, 679
- gitlab-project-merge-request-discussion-note-get
  - command line option, 680
- gitlab-project-merge-request-discussion-note-update
  - command line option, 680
- gitlab-project-merge-request-discussion-update
  - command line option, 678
- gitlab-project-merge-request-get
  - command line option, 663
- gitlab-project-merge-request-list
  - command line option, 664
- gitlab-project-merge-request-merge
  - command line option, 665
- gitlab-project-merge-request-merge-ref
  - command line option, 666
- gitlab-project-merge-request-note
  - command line option, 681
- gitlab-project-merge-request-note-award-emoji
  - command line option, 683
- gitlab-project-merge-request-note-award-emoji-create
  - command line option, 683
- gitlab-project-merge-request-note-award-emoji-delete
  - command line option, 683
- gitlab-project-merge-request-note-award-emoji-get
  - command line option, 684
- gitlab-project-merge-request-note-award-emoji-list
  - command line option, 684
- gitlab-project-merge-request-note-create
  - command line option, 681
- gitlab-project-merge-request-note-delete
  - command line option, 681
- gitlab-project-merge-request-note-get
  - command line option, 682
- gitlab-project-merge-request-note-list
  - command line option, 682
- gitlab-project-merge-request-note-update
  - command line option, 682
- gitlab-project-merge-request-participants
  - command line option, 666
- gitlab-project-merge-request-pipeline
  - command line option, 685
- gitlab-project-merge-request-pipeline-create
  - command line option, 685
- gitlab-project-merge-request-pipeline-list
  - command line option, 685
- gitlab-project-merge-request-rebase
  - command line option, 666
- gitlab-project-merge-request-reset-spent-time
  - command line option, 666
- gitlab-project-merge-request-reset-time-estimate
  - command line option, 667
- gitlab-project-merge-request-resource-label-event
  - command line option, 686
- gitlab-project-merge-request-resource-label-event-get
  - command line option, 686
- gitlab-project-merge-request-resource-label-event-list
  - command line option, 686
- gitlab-project-merge-request-resource-milestone-event
  - command line option, 687
- gitlab-project-merge-request-resource-milestone-event-
  - command line option, 687
- gitlab-project-merge-request-resource-milestone-event-
  - command line option, 687
- gitlab-project-merge-request-resource-state-event
  - command line option, 688
- gitlab-project-merge-request-resource-state-event-get
  - command line option, 688
- gitlab-project-merge-request-resource-state-event-list
  - command line option, 688
- gitlab-project-merge-request-subscribe

- command line option, [667](#)
- gitlab-project-merge-request-time-estimate
  - command line option, [667](#)
- gitlab-project-merge-request-time-stats
  - command line option, [668](#)
- gitlab-project-merge-request-todo
  - command line option, [668](#)
- gitlab-project-merge-request-unapprove
  - command line option, [668](#)
- gitlab-project-merge-request-unsubscribe
  - command line option, [668](#)
- gitlab-project-merge-request-update
  - command line option, [669](#)
- gitlab-project-merge-train command line option, [689](#)
- gitlab-project-merge-train-list command line option, [689](#)
- gitlab-project-milestone command line option, [689](#)
- gitlab-project-milestone-create command line option, [690](#)
- gitlab-project-milestone-delete command line option, [690](#)
- gitlab-project-milestone-get command line option, [690](#)
- gitlab-project-milestone-issues command line option, [691](#)
- gitlab-project-milestone-list command line option, [691](#)
- gitlab-project-milestone-merge-requests
  - command line option, [691](#)
- gitlab-project-milestone-update command line option, [692](#)
- gitlab-project-mirror-pull command line option, [567](#)
- gitlab-project-note command line option, [692](#)
- gitlab-project-note-get command line option, [692](#)
- gitlab-project-note-list command line option, [693](#)
- gitlab-project-notification-settings
  - command line option, [693](#)
- gitlab-project-notification-settings-get
  - command line option, [693](#)
- gitlab-project-notification-settings-update
  - command line option, [694](#)
- gitlab-project-package command line option, [694](#)
- gitlab-project-package-delete command line option, [695](#)
- gitlab-project-package-file command line option, [696](#)
- gitlab-project-package-file-delete
  - command line option, [696](#)
- gitlab-project-package-file-list
  - command line option, [696](#)
- gitlab-project-package-get command line option, [695](#)
- gitlab-project-package-list command line option, [695](#)
- gitlab-project-pages-domain command line option, [697](#)
- gitlab-project-pages-domain-create
  - command line option, [697](#)
- gitlab-project-pages-domain-delete
  - command line option, [697](#)
- gitlab-project-pages-domain-get command line option, [698](#)
- gitlab-project-pages-domain-list
  - command line option, [698](#)
- gitlab-project-pages-domain-update
  - command line option, [698](#)
- gitlab-project-pipeline command line option, [699](#)
- gitlab-project-pipeline-bridge command line option, [701](#)
- gitlab-project-pipeline-bridge-list
  - command line option, [701](#)
- gitlab-project-pipeline-cancel command line option, [699](#)
- gitlab-project-pipeline-create command line option, [699](#)
- gitlab-project-pipeline-delete command line option, [699](#)
- gitlab-project-pipeline-get command line option, [700](#)
- gitlab-project-pipeline-job command line option, [702](#)
- gitlab-project-pipeline-job-list
  - command line option, [702](#)
- gitlab-project-pipeline-list command line option, [700](#)
- gitlab-project-pipeline-retry command line option, [701](#)
- gitlab-project-pipeline-schedule
  - command line option, [702](#)
- gitlab-project-pipeline-schedule-create
  - command line option, [703](#)
- gitlab-project-pipeline-schedule-delete
  - command line option, [703](#)
- gitlab-project-pipeline-schedule-get
  - command line option, [703](#)
- gitlab-project-pipeline-schedule-list
  - command line option, [704](#)
- gitlab-project-pipeline-schedule-play
  - command line option, [704](#)
- gitlab-project-pipeline-schedule-take-ownership



- command line option, [704](#)
- gitlab-project-pipeline-schedule-update
  - command line option, [705](#)
- gitlab-project-pipeline-schedule-variable
  - command line option, [705](#)
- gitlab-project-pipeline-schedule-variable-create
  - command line option, [705](#)
- gitlab-project-pipeline-schedule-variable-delete
  - command line option, [706](#)
- gitlab-project-pipeline-schedule-variable-update
  - command line option, [706](#)
- gitlab-project-pipeline-test-report
  - command line option, [707](#)
- gitlab-project-pipeline-test-report-get
  - command line option, [707](#)
- gitlab-project-pipeline-test-report-summary
  - command line option, [707](#)
- gitlab-project-pipeline-test-report-summary-get
  - command line option, [707](#)
- gitlab-project-pipeline-variable
  - command line option, [708](#)
- gitlab-project-pipeline-variable-list
  - command line option, [708](#)
- gitlab-project-protected-branch
  - command line option, [708](#)
- gitlab-project-protected-branch-create
  - command line option, [709](#)
- gitlab-project-protected-branch-delete
  - command line option, [709](#)
- gitlab-project-protected-branch-get
  - command line option, [709](#)
- gitlab-project-protected-branch-list
  - command line option, [710](#)
- gitlab-project-protected-environment
  - command line option, [710](#)
- gitlab-project-protected-environment-create
  - command line option, [710](#)
- gitlab-project-protected-environment-delete
  - command line option, [711](#)
- gitlab-project-protected-environment-get
  - command line option, [711](#)
- gitlab-project-protected-environment-list
  - command line option, [711](#)
- gitlab-project-protected-tag
  - command line option, [712](#)
- gitlab-project-protected-tag-create
  - command line option, [712](#)
- gitlab-project-protected-tag-delete
  - command line option, [712](#)
- gitlab-project-protected-tag-get
  - command line option, [713](#)
- gitlab-project-protected-tag-list
  - command line option, [713](#)
- gitlab-project-push-rules
  - command line option, [713](#)
- gitlab-project-push-rules-create
  - command line option, [714](#)
- gitlab-project-push-rules-delete
  - command line option, [714](#)
- gitlab-project-push-rules-get
  - command line option, [715](#)
- gitlab-project-push-rules-update
  - command line option, [715](#)
- gitlab-project-registry-repository
  - command line option, [716](#)
- gitlab-project-registry-repository-delete
  - command line option, [716](#)
- gitlab-project-registry-repository-list
  - command line option, [716](#)
- gitlab-project-registry-tag
  - command line option, [717](#)
- gitlab-project-registry-tag-delete
  - command line option, [717](#)
- gitlab-project-registry-tag-delete-in-bulk
  - command line option, [717](#)
- gitlab-project-registry-tag-get
  - command line option, [718](#)
- gitlab-project-registry-tag-list
  - command line option, [718](#)
- gitlab-project-release
  - command line option, [718](#)
- gitlab-project-release-create
  - command line option, [719](#)
- gitlab-project-release-delete
  - command line option, [719](#)
- gitlab-project-release-get
  - command line option, [719](#)
- gitlab-project-release-link
  - command line option, [721](#)
- gitlab-project-release-link-create
  - command line option, [721](#)
- gitlab-project-release-link-delete
  - command line option, [721](#)
- gitlab-project-release-link-get
  - command line option, [722](#)
- gitlab-project-release-link-list
  - command line option, [722](#)
- gitlab-project-release-link-update
  - command line option, [722](#)
- gitlab-project-release-list
  - command line option, [720](#)
- gitlab-project-release-update
  - command line option, [720](#)
- gitlab-project-remote-mirror
  - command line option, [723](#)
- gitlab-project-remote-mirror-create
  - command line option, [723](#)
- gitlab-project-remote-mirror-list

- command line option, [723](#)
- gitlab-project-remote-mirror-update
  - command line option, [724](#)
- gitlab-project-repository-archive
  - command line option, [567](#)
- gitlab-project-repository-blob
  - command line option, [567](#)
- gitlab-project-repository-compare
  - command line option, [567](#)
- gitlab-project-repository-contributors
  - command line option, [568](#)
- gitlab-project-repository-merge-base
  - command line option, [568](#)
- gitlab-project-repository-raw-blob
  - command line option, [568](#)
- gitlab-project-repository-tree
  - command line option, [568](#)
- gitlab-project-restore
  - command line option, [569](#)
- gitlab-project-runner
  - command line option, [724](#)
- gitlab-project-runner-create
  - command line option, [724](#)
- gitlab-project-runner-delete
  - command line option, [725](#)
- gitlab-project-runner-list
  - command line option, [725](#)
- gitlab-project-search
  - command line option, [569](#)
- gitlab-project-service
  - command line option, [725](#)
- gitlab-project-service-available
  - command line option, [726](#)
- gitlab-project-service-delete
  - command line option, [726](#)
- gitlab-project-service-get
  - command line option, [726](#)
- gitlab-project-service-list
  - command line option, [726](#)
- gitlab-project-service-update
  - command line option, [727](#)
- gitlab-project-share
  - command line option, [569](#)
- gitlab-project-snapshot
  - command line option, [569](#)
- gitlab-project-snippet
  - command line option, [727](#)
- gitlab-project-snippet-award-emoji
  - command line option, [730](#)
- gitlab-project-snippet-award-emoji-create
  - command line option, [730](#)
- gitlab-project-snippet-award-emoji-delete
  - command line option, [730](#)
- gitlab-project-snippet-award-emoji-get
  - command line option, [731](#)
- gitlab-project-snippet-award-emoji-list
  - command line option, [731](#)
- gitlab-project-snippet-content
  - command line option, [727](#)
- gitlab-project-snippet-create
  - command line option, [728](#)
- gitlab-project-snippet-delete
  - command line option, [728](#)
- gitlab-project-snippet-discussion
  - command line option, [732](#)
- gitlab-project-snippet-discussion-create
  - command line option, [732](#)
- gitlab-project-snippet-discussion-get
  - command line option, [732](#)
- gitlab-project-snippet-discussion-list
  - command line option, [733](#)
- gitlab-project-snippet-discussion-note
  - command line option, [733](#)
- gitlab-project-snippet-discussion-note-create
  - command line option, [733](#)
- gitlab-project-snippet-discussion-note-delete
  - command line option, [734](#)
- gitlab-project-snippet-discussion-note-get
  - command line option, [734](#)
- gitlab-project-snippet-discussion-note-update
  - command line option, [735](#)
- gitlab-project-snippet-get
  - command line option, [728](#)
- gitlab-project-snippet-list
  - command line option, [729](#)
- gitlab-project-snippet-note
  - command line option, [735](#)
- gitlab-project-snippet-note-award-emoji
  - command line option, [737](#)
- gitlab-project-snippet-note-award-emoji-create
  - command line option, [737](#)
- gitlab-project-snippet-note-award-emoji-delete
  - command line option, [738](#)
- gitlab-project-snippet-note-award-emoji-get
  - command line option, [738](#)
- gitlab-project-snippet-note-award-emoji-list
  - command line option, [739](#)
- gitlab-project-snippet-note-create
  - command line option, [735](#)
- gitlab-project-snippet-note-delete
  - command line option, [736](#)
- gitlab-project-snippet-note-get
  - command line option, [736](#)
- gitlab-project-snippet-note-list
  - command line option, [736](#)
- gitlab-project-snippet-note-update
  - command line option, [737](#)
- gitlab-project-snippet-update
  - command line option, [731](#)



- line option, [729](#)
- gitlab-project-snippet-user-agent-detail command line option, [730](#)
- gitlab-project-star command line option, [570](#)
- gitlab-project-storage command line option, [739](#)
- gitlab-project-storage-get command line option, [739](#)
- gitlab-project-tag command line option, [740](#)
- gitlab-project-tag-create command line option, [740](#)
- gitlab-project-tag-delete command line option, [740](#)
- gitlab-project-tag-get command line option, [740](#)
- gitlab-project-tag-list command line option, [741](#)
- gitlab-project-transfer command line option, [570](#)
- gitlab-project-transfer-project command line option, [570](#)
- gitlab-project-trigger command line option, [741](#)
- gitlab-project-trigger-create command line option, [741](#)
- gitlab-project-trigger-delete command line option, [742](#)
- gitlab-project-trigger-get command line option, [742](#)
- gitlab-project-trigger-list command line option, [742](#)
- gitlab-project-trigger-pipeline command line option, [570](#)
- gitlab-project-trigger-update command line option, [743](#)
- gitlab-project-unarchive command line option, [571](#)
- gitlab-project-unshare command line option, [571](#)
- gitlab-project-unstar command line option, [571](#)
- gitlab-project-update command line option, [573](#)
- gitlab-project-update-submodule command line option, [575](#)
- gitlab-project-upload command line option, [575](#)
- gitlab-project-user command line option, [743](#)
- gitlab-project-user-list command line option, [743](#)
- gitlab-project-variable command line option, [744](#)
- gitlab-project-variable-create command line option, [744](#)
- gitlab-project-variable-delete command line option, [744](#)
- gitlab-project-variable-get command line option, [745](#)
- gitlab-project-variable-list command line option, [745](#)
- gitlab-project-variable-update command line option, [745](#)
- gitlab-project-wiki command line option, [746](#)
- gitlab-project-wiki-create command line option, [746](#)
- gitlab-project-wiki-delete command line option, [746](#)
- gitlab-project-wiki-get command line option, [747](#)
- gitlab-project-wiki-list command line option, [747](#)
- gitlab-project-wiki-update command line option, [747](#)
- gitlab-registry-repository command line option, [748](#)
- gitlab-registry-repository-get command line option, [748](#)
- gitlab-runner command line option, [748](#)
- gitlab-runner-all command line option, [748](#), [751](#)
- gitlab-runner-all-list command line option, [751](#)
- gitlab-runner-create command line option, [749](#)
- gitlab-runner-delete command line option, [749](#)
- gitlab-runner-get command line option, [749](#)
- gitlab-runner-job command line option, [752](#)
- gitlab-runner-job-list command line option, [752](#)
- gitlab-runner-list command line option, [750](#)
- gitlab-runner-update command line option, [750](#)
- gitlab-runner-verify command line option, [751](#)
- gitlab-shared-project command line option, [752](#)
- gitlab-shared-project-list command line option, [753](#)
- gitlab-snippet command line option, [753](#)
- gitlab-snippet-content command line

- option, [754](#)
- gitlab-snippet-create command line option, [754](#)
- gitlab-snippet-delete command line option, [754](#)
- gitlab-snippet-get command line option, [755](#)
- gitlab-snippet-list command line option, [755](#)
- gitlab-snippet-public command line option, [755](#)
- gitlab-snippet-update command line option, [755](#)
- gitlab-snippet-user-agent-detail command line option, [756](#)
- gitlab-starred-project command line option, [756](#)
- gitlab-starred-project-list command line option, [756](#)
- gitlab-todo command line option, [757](#)
- gitlab-todo-delete command line option, [757](#)
- gitlab-todo-list command line option, [758](#)
- gitlab-todo-mark-all-as-done command line option, [758](#)
- gitlab-todo-mark-as-done command line option, [758](#)
- gitlab-topic command line option, [759](#)
- gitlab-topic-create command line option, [759](#)
- gitlab-topic-delete command line option, [759](#)
- gitlab-topic-get command line option, [759](#)
- gitlab-topic-list command line option, [760](#)
- gitlab-topic-update command line option, [760](#)
- gitlab-user command line option, [761](#)
- gitlab-user-activate command line option, [761](#)
- gitlab-user-activities command line option, [767](#)
- gitlab-user-activities-list command line option, [767](#)
- gitlab-user-approve command line option, [761](#)
- gitlab-user-ban command line option, [761](#)
- gitlab-user-block command line option, [761](#)
- gitlab-user-create command line option, [762](#)
- gitlab-user-custom-attribute command line option, [767](#)
- gitlab-user-custom-attribute-delete command line option, [767](#)
- gitlab-user-custom-attribute-get command line option, [768](#)
- gitlab-user-custom-attribute-list command line option, [768](#)
- gitlab-user-deactivate command line option, [763](#)
- gitlab-user-delete command line option, [763](#)
- gitlab-user-email command line option, [768](#)
- gitlab-user-email-create command line option, [769](#)
- gitlab-user-email-delete command line option, [769](#)
- gitlab-user-email-get command line option, [769](#)
- gitlab-user-email-list command line option, [769](#)
- gitlab-user-event command line option, [770](#)
- gitlab-user-event-list command line option, [770](#)
- gitlab-user-follow command line option, [763](#)
- gitlab-user-get command line option, [764](#)
- gitlab-user-gpg-key command line option, [771](#)
- gitlab-user-gpg-key-create command line option, [771](#)
- gitlab-user-gpg-key-delete command line option, [771](#)
- gitlab-user-gpg-key-get command line option, [771](#)
- gitlab-user-gpg-key-list command line option, [772](#)
- gitlab-user-impersonation-token command line option, [772](#)
- gitlab-user-impersonation-token-create command line option, [772](#)
- gitlab-user-impersonation-token-delete command line option, [773](#)
- gitlab-user-impersonation-token-get command line option, [773](#)
- gitlab-user-impersonation-token-list command line option, [773](#)
- gitlab-user-key command line option, [774](#)
- gitlab-user-key-create command line option, [774](#)
- gitlab-user-key-delete command line option, [774](#)
- gitlab-user-key-get command line option, [774](#)
- gitlab-user-key-list command line

option, 775

gitlab-user-list command line option, 764

gitlab-user-membership command line option, 775

gitlab-user-membership-get command line option, 775

gitlab-user-membership-list command line option, 775

gitlab-user-personal-access-token command line option, 776

gitlab-user-personal-access-token-create command line option, 776

gitlab-user-project command line option, 776

gitlab-user-project-create command line option, 777

gitlab-user-project-list command line option, 778

gitlab-user-reject command line option, 765

gitlab-user-status command line option, 779

gitlab-user-status-get command line option, 779

gitlab-user-unban command line option, 765

gitlab-user-unblock command line option, 765

gitlab-user-unfollow command line option, 765

gitlab-user-update command line option, 766

gitlab-variable command line option, 779

gitlab-variable-create command line option, 780

gitlab-variable-delete command line option, 780

gitlab-variable-get command line option, 780

gitlab-variable-list command line option, 781

gitlab-variable-update command line option, 781

-o

gitlab command line option, 458

-v

gitlab command line option, 458

## A

access\_tokens (*gitlab.v4.objects.Group* attribute), 203

access\_tokens (*gitlab.v4.objects.Project* attribute), 259

AccessLevel (*class in gitlab.const*), 426

AccessRequestMixin (*class in gitlab.mixins*), 444

accessrequests (*gitlab.v4.objects.Group* attribute), 203

accessrequests (*gitlab.v4.objects.Project* attribute), 259

activate() (*gitlab.v4.objects.User* method), 405

add\_ldap\_group\_link() (*gitlab.v4.objects.Group* method), 203

add\_spent\_time() (*gitlab.mixins.TimeTrackingMixin* method), 450

additionalstatistics (*gitlab.v4.objects.Project* attribute), 259

ADMIN (*gitlab.const.AccessLevel* attribute), 426

all() (*gitlab.v4.objects.RunnerManager* method), 396

api\_url (*gitlab.Gitlab* property), 172

api\_version (*gitlab.Gitlab* property), 172

appearance (*gitlab.Gitlab* attribute), 172

Application (*class in gitlab.v4.objects*), 182

ApplicationAppearance (*class in gitlab.v4.objects*), 182

ApplicationAppearanceManager (*class in gitlab.v4.objects*), 182

ApplicationManager (*class in gitlab.v4.objects*), 183

applications (*gitlab.Gitlab* attribute), 172

ApplicationSettings (*class in gitlab.v4.objects*), 184

ApplicationSettingsManager (*class in gitlab.v4.objects*), 184

approval\_rule\_id (*gitlab.v4.objects.ProjectMergeRequestApprovalRule* attribute), 338

approval\_rules (*gitlab.v4.objects.ProjectMergeRequest* attribute), 334

approval\_state (*gitlab.v4.objects.ProjectMergeRequest* attribute), 334

approvalrules (*gitlab.v4.objects.Project* attribute), 259

approvals (*gitlab.v4.objects.Project* attribute), 259

approvals (*gitlab.v4.objects.ProjectMergeRequest* attribute), 334

approve() (*gitlab.mixins.AccessRequestMixin* method), 444

approve() (*gitlab.v4.objects.ProjectMergeRequest* method), 334

approve() (*gitlab.v4.objects.User* method), 405

archive() (*gitlab.v4.objects.Project* method), 259

artifact() (*gitlab.v4.objects.Project* method), 259

artifact() (*gitlab.v4.objects.ProjectJob* method), 317

artifacts (*gitlab.v4.objects.Project* attribute), 260

artifacts() (*gitlab.v4.objects.ProjectJob* method), 318

asdict() (*gitlab.base.RESTObject* method), 423

attributes (*gitlab.base.RESTObject* property), 423

audit\_events (*gitlab.Gitlab* attribute), 172

audit\_events (*gitlab.v4.objects.Group* attribute), 203

audit\_events (*gitlab.v4.objects.Project* attribute), 260

AuditEvent (*class in gitlab.v4.objects*), 187

AuditEventManager (*class in gitlab.v4.objects*), 187

`auth()` (*gitlab.Gitlab* method), 172  
`available()` (*gitlab.v4.objects.ProjectIntegrationManager* method), 303  
`awardemojis` (*gitlab.v4.objects.GroupEpicNote* attribute), 222  
`awardemojis` (*gitlab.v4.objects.ProjectIssue* attribute), 305  
`awardemojis` (*gitlab.v4.objects.ProjectIssueNote* attribute), 312  
`awardemojis` (*gitlab.v4.objects.ProjectMergeRequest* attribute), 334  
`awardemojis` (*gitlab.v4.objects.ProjectMergeRequestNote* attribute), 347  
`awardemojis` (*gitlab.v4.objects.ProjectSnippet* attribute), 378  
`awardemojis` (*gitlab.v4.objects.ProjectSnippetNote* attribute), 382

## B

`BadgeRenderMixin` (*class in gitlab.mixins*), 444  
`badges` (*gitlab.v4.objects.Group* attribute), 203  
`badges` (*gitlab.v4.objects.Project* attribute), 260  
`ban()` (*gitlab.v4.objects.User* method), 406  
`billable_members` (*gitlab.v4.objects.Group* attribute), 203  
`blame()` (*gitlab.v4.objects.ProjectFileManager* method), 296  
`BLOBS` (*gitlab.const.SearchScope* attribute), 426  
`block()` (*gitlab.v4.objects.User* method), 406  
`boards` (*gitlab.v4.objects.Group* attribute), 203  
`boards` (*gitlab.v4.objects.Project* attribute), 260  
`branch` (*gitlab.v4.objects.ProjectFile* attribute), 295  
`branches` (*gitlab.v4.objects.Project* attribute), 260  
`bridges` (*gitlab.v4.objects.ProjectPipeline* attribute), 359  
`BroadcastMessage` (*class in gitlab.v4.objects*), 188  
`BroadcastMessageManager` (*class in gitlab.v4.objects*), 188  
`broadcastmessages` (*gitlab.Gitlab* attribute), 172

## C

`cancel()` (*gitlab.v4.objects.ProjectJob* method), 318  
`cancel()` (*gitlab.v4.objects.ProjectPipeline* method), 359  
`cancel_merge_when_pipeline_succeeds()` (*gitlab.v4.objects.ProjectMergeRequest* method), 334  
`changes()` (*gitlab.v4.objects.ProjectMergeRequest* method), 334  
`cherry_pick()` (*gitlab.v4.objects.ProjectCommit* method), 280  
`ci_lint` (*gitlab.Gitlab* attribute), 172  
`ci_lint` (*gitlab.v4.objects.Project* attribute), 260  
`CiLint` (*class in gitlab.v4.objects*), 189

`CiLintManager` (*class in gitlab.v4.objects*), 189  
`closed_by()` (*gitlab.v4.objects.ProjectIssue* method), 305  
`closes_issues()` (*gitlab.v4.objects.ProjectMergeRequest* method), 335  
`cls_to_gitlab_resource()` (*in module gitlab.cli*), 424  
`clusters` (*gitlab.v4.objects.Group* attribute), 203  
`clusters` (*gitlab.v4.objects.Project* attribute), 260  
`comments` (*gitlab.v4.objects.ProjectCommit* attribute), 281  
`commit_message` (*gitlab.v4.objects.ProjectFile* attribute), 295  
`COMMITTS` (*gitlab.const.SearchScope* attribute), 426  
`commits` (*gitlab.v4.objects.Project* attribute), 260  
`commits()` (*gitlab.v4.objects.ProjectMergeRequest* method), 335  
`compound_metrics()` (*gitlab.v4.objects.SidekiqManager* method), 399  
`ConfigError`, 425  
`content()` (*gitlab.v4.objects.ProjectSnippet* method), 378  
`content()` (*gitlab.v4.objects.Snippet* method), 400  
`copy_dict()` (*in module gitlab.utils*), 453  
`create()` (*gitlab.mixins.CreateMixin* method), 445  
`create()` (*gitlab.v4.objects.GroupClusterManager* method), 213  
`create()` (*gitlab.v4.objects.GroupEpicIssueManager* method), 220  
`create()` (*gitlab.v4.objects.ProjectClusterManager* method), 279  
`create()` (*gitlab.v4.objects.ProjectCommitStatusManager* method), 287  
`create()` (*gitlab.v4.objects.ProjectFileManager* method), 297  
`create()` (*gitlab.v4.objects.ProjectForkManager* method), 299  
`create()` (*gitlab.v4.objects.ProjectIssueLinkManager* method), 309  
`create()` (*gitlab.v4.objects.ProjectMergeRequestApprovalRuleManager* method), 339  
`create()` (*gitlab.v4.objects.ProjectPipelineManager* method), 361  
`create_fork_relation()` (*gitlab.v4.objects.Project* method), 260  
`CreateMixin` (*class in gitlab.mixins*), 445  
`CRUDMixin` (*class in gitlab.mixins*), 445  
`current_failures()` (*gitlab.v4.objects.GeoNodeManager* method), 200  
`current_page` (*gitlab.base.RESTObjectList* property), 424

[current\\_page](#) (*gitlab.GitlabList* property), 181  
[CurrentUser](#) (class in *gitlab.v4.objects*), 190  
[CurrentUserEmail](#) (class in *gitlab.v4.objects*), 190  
[CurrentUserEmailManager](#) (class in *gitlab.v4.objects*), 190  
[CurrentUserGPGKey](#) (class in *gitlab.v4.objects*), 191  
[CurrentUserGPGKeyManager](#) (class in *gitlab.v4.objects*), 191  
[CurrentUserKey](#) (class in *gitlab.v4.objects*), 192  
[CurrentUserKeyManager](#) (class in *gitlab.v4.objects*), 192  
[CurrentUserManager](#) (class in *gitlab.v4.objects*), 193  
[CurrentUserStatus](#) (class in *gitlab.v4.objects*), 193  
[CurrentUserStatusManager](#) (class in *gitlab.v4.objects*), 193  
[CUSTOM](#) (*gitlab.const.NotificationLevel* attribute), 426  
[customattributes](#) (*gitlab.v4.objects.Group* attribute), 203  
[customattributes](#) (*gitlab.v4.objects.Project* attribute), 260  
[customattributes](#) (*gitlab.v4.objects.User* attribute), 406

## D

[deactivate\(\)](#) (*gitlab.v4.objects.User* method), 406  
[decode\(\)](#) (*gitlab.v4.objects.ProjectFile* method), 295  
[delete\(\)](#) (*gitlab.mixins.DeleteMixin* method), 445  
[delete\(\)](#) (*gitlab.mixins.ObjectDeleteMixin* method), 448  
[delete\(\)](#) (*gitlab.v4.objects.ProjectArtifactManager* method), 271  
[delete\(\)](#) (*gitlab.v4.objects.ProjectFile* method), 295  
[delete\(\)](#) (*gitlab.v4.objects.ProjectFileManager* method), 297  
[delete\\_artifacts\(\)](#) (*gitlab.v4.objects.ProjectJob* method), 319  
[delete\\_fork\\_relation\(\)](#) (*gitlab.v4.objects.Project* method), 260  
[delete\\_in\\_bulk\(\)](#) (*gitlab.v4.objects.ProjectRegistryTagManager* method), 372  
[delete\\_ldap\\_group\\_link\(\)](#) (*gitlab.v4.objects.Group* method), 203  
[delete\\_merged\\_branches\(\)](#) (*gitlab.v4.objects.RepositoryMixin* method), 390  
[DeleteMixin](#) (class in *gitlab.mixins*), 445  
[DeployKey](#) (class in *gitlab.v4.objects*), 194  
[DeployKeyManager](#) (class in *gitlab.v4.objects*), 194  
[deploykeys](#) (*gitlab.Gitlab* attribute), 173  
[deployments](#) (*gitlab.v4.objects.Project* attribute), 260  
[DeployToken](#) (class in *gitlab.v4.objects*), 195  
[DeployTokenManager](#) (class in *gitlab.v4.objects*), 195  
[deploytokens](#) (*gitlab.Gitlab* attribute), 173

[deploytokens](#) (*gitlab.v4.objects.Group* attribute), 204  
[deploytokens](#) (*gitlab.v4.objects.Project* attribute), 260  
[descendant\\_groups](#) (*gitlab.v4.objects.Group* attribute), 204  
[DEVELOPER](#) (*gitlab.const.AccessLevel* attribute), 426  
[die\(\)](#) (in module *gitlab.cli*), 424  
[diff\(\)](#) (*gitlab.v4.objects.ProjectCommit* method), 281  
[diffs](#) (*gitlab.v4.objects.ProjectMergeRequest* attribute), 335  
[DISABLED](#) (*gitlab.const.NotificationLevel* attribute), 426  
[discussions](#) (*gitlab.v4.objects.ProjectCommit* attribute), 281  
[discussions](#) (*gitlab.v4.objects.ProjectIssue* attribute), 305  
[discussions](#) (*gitlab.v4.objects.ProjectMergeRequest* attribute), 335  
[discussions](#) (*gitlab.v4.objects.ProjectSnippet* attribute), 378  
[Dockerfile](#) (class in *gitlab.v4.objects*), 195  
[DockerfileManager](#) (class in *gitlab.v4.objects*), 195  
[dockerfiles](#) (*gitlab.Gitlab* attribute), 173  
[docs\(\)](#) (in module *gitlab.cli*), 425  
[download\(\)](#) (*gitlab.mixins.DownloadMixin* method), 446  
[download\(\)](#) (*gitlab.v4.objects.GenericPackageManager* method), 198  
[download\(\)](#) (*gitlab.v4.objects.ProjectArtifactManager* method), 271  
[DownloadMixin](#) (class in *gitlab.mixins*), 446

## E

[emails](#) (*gitlab.v4.objects.CurrentUser* attribute), 190  
[emails](#) (*gitlab.v4.objects.User* attribute), 406  
[enable\(\)](#) (*gitlab.v4.objects.ProjectKeyManager* method), 322  
[enable\\_debug\(\)](#) (*gitlab.Gitlab* static method), 173  
[encoded\\_id](#) (*gitlab.base.RESTObject* property), 423  
[EncodedId](#) (class in *gitlab.utils*), 453  
[environments](#) (*gitlab.v4.objects.Project* attribute), 260  
[epics](#) (*gitlab.v4.objects.Group* attribute), 204  
[erase\(\)](#) (*gitlab.v4.objects.ProjectJob* method), 319  
[Event](#) (class in *gitlab.v4.objects*), 196  
[EventManager](#) (class in *gitlab.v4.objects*), 196  
[events](#) (*gitlab.Gitlab* attribute), 173  
[events](#) (*gitlab.v4.objects.Project* attribute), 260  
[events](#) (*gitlab.v4.objects.User* attribute), 407  
[exists\(\)](#) (*gitlab.v4.objects.NamespaceManager* method), 256  
[exports](#) (*gitlab.v4.objects.Group* attribute), 204  
[exports](#) (*gitlab.v4.objects.Project* attribute), 260

## F

[Feature](#) (class in *gitlab.v4.objects*), 197  
[FeatureManager](#) (class in *gitlab.v4.objects*), 197



`features` (*gitlab.Gitlab* attribute), 173  
`file_path` (*gitlab.v4.objects.ProjectFile* attribute), 295  
`files` (*gitlab.v4.objects.Project* attribute), 261  
`follow()` (*gitlab.v4.objects.User* method), 407  
`followers_users` (*gitlab.v4.objects.User* attribute), 407  
`following_users` (*gitlab.v4.objects.User* attribute), 407  
`forks` (*gitlab.v4.objects.Project* attribute), 261  
`format_help()` (*gitlab.cli.VerticalHelpFormatter* method), 424  
`from_config()` (*gitlab.Gitlab* class method), 173

## G

`generic_packages` (*gitlab.v4.objects.Project* attribute), 261  
`GenericPackage` (class in *gitlab.v4.objects*), 197  
`GenericPackageManager` (class in *gitlab.v4.objects*), 198  
`GeoNode` (class in *gitlab.v4.objects*), 199  
`GeoNodeManager` (class in *gitlab.v4.objects*), 200  
`geonodes` (*gitlab.Gitlab* attribute), 173  
`get()` (*gitlab.mixins.GetMixin* method), 446  
`get()` (*gitlab.mixins.GetWithoutIdMixin* method), 447  
`get()` (*gitlab.v4.objects.ApplicationAppearanceManager* method), 183  
`get()` (*gitlab.v4.objects.ApplicationSettingsManager* method), 186  
`get()` (*gitlab.v4.objects.AuditEventManager* method), 187  
`get()` (*gitlab.v4.objects.BroadcastMessageManager* method), 188  
`get()` (*gitlab.v4.objects.CurrentUserEmailManager* method), 190  
`get()` (*gitlab.v4.objects.CurrentUserGPGKeyManager* method), 191  
`get()` (*gitlab.v4.objects.CurrentUserKeyManager* method), 192  
`get()` (*gitlab.v4.objects.CurrentUserManager* method), 193  
`get()` (*gitlab.v4.objects.CurrentUserStatusManager* method), 194  
`get()` (*gitlab.v4.objects.DockerfileManager* method), 195  
`get()` (*gitlab.v4.objects.GeoNodeManager* method), 200  
`get()` (*gitlab.v4.objects.GitignoreManager* method), 201  
`get()` (*gitlab.v4.objects.GitlabciyamlManager* method), 202  
`get()` (*gitlab.v4.objects.GroupAuditEventManager* method), 208  
`get()` (*gitlab.v4.objects.GroupBadgeManager* method), 209  
`get()` (*gitlab.v4.objects.GroupBoardListManager* method), 211  
`get()` (*gitlab.v4.objects.GroupBoardManager* method), 212  
`get()` (*gitlab.v4.objects.GroupClusterManager* method), 214  
`get()` (*gitlab.v4.objects.GroupCustomAttributeManager* method), 215  
`get()` (*gitlab.v4.objects.GroupDeployTokenManager* method), 216  
`get()` (*gitlab.v4.objects.GroupEpicAwardEmojiManager* method), 218  
`get()` (*gitlab.v4.objects.GroupEpicDiscussionNoteManager* method), 219  
`get()` (*gitlab.v4.objects.GroupEpicManager* method), 221  
`get()` (*gitlab.v4.objects.GroupEpicNoteAwardEmojiManager* method), 222  
`get()` (*gitlab.v4.objects.GroupEpicNoteManager* method), 223  
`get()` (*gitlab.v4.objects.GroupEpicResourceLabelEventManager* method), 224  
`get()` (*gitlab.v4.objects.GroupExportManager* method), 225  
`get()` (*gitlab.v4.objects.GroupHookManager* method), 226  
`get()` (*gitlab.v4.objects.GroupImportManager* method), 227  
`get()` (*gitlab.v4.objects.GroupInvitationManager* method), 228  
`get()` (*gitlab.v4.objects.GroupIssuesStatisticsManager* method), 230  
`get()` (*gitlab.v4.objects.GroupLabelManager* method), 232  
`get()` (*gitlab.v4.objects.GroupManager* method), 235  
`get()` (*gitlab.v4.objects.GroupMemberAllManager* method), 236  
`get()` (*gitlab.v4.objects.GroupMemberManager* method), 237  
`get()` (*gitlab.v4.objects.GroupMilestoneManager* method), 240  
`get()` (*gitlab.v4.objects.GroupNotificationSettingsManager* method), 241  
`get()` (*gitlab.v4.objects.GroupPushRulesManager* method), 244  
`get()` (*gitlab.v4.objects.GroupVariableManager* method), 247  
`get()` (*gitlab.v4.objects.GroupWikiManager* method), 248  
`get()` (*gitlab.v4.objects.HookManager* method), 249  
`get()` (*gitlab.v4.objects.IssueManager* method), 250  
`get()` (*gitlab.v4.objects.IssuesStatisticsManager* method), 251  
`get()` (*gitlab.v4.objects.KeyManager* method), 252  
`get()` (*gitlab.v4.objects.LicenseManager* method), 254  
`get()` (*gitlab.v4.objects.NamespaceManager* method),

- 256
- `get()` (`gitlab.v4.objects.NotificationSettingsManager` method), 258
- `get()` (`gitlab.v4.objects.ProjectAdditionalStatisticsManager` method), 268
- `get()` (`gitlab.v4.objects.ProjectApprovalManager` method), 269
- `get()` (`gitlab.v4.objects.ProjectAuditEventManager` method), 273
- `get()` (`gitlab.v4.objects.ProjectBadgeManager` method), 274
- `get()` (`gitlab.v4.objects.ProjectBoardListManager` method), 275
- `get()` (`gitlab.v4.objects.ProjectBoardManager` method), 276
- `get()` (`gitlab.v4.objects.ProjectBranchManager` method), 277
- `get()` (`gitlab.v4.objects.ProjectCiLintManager` method), 278
- `get()` (`gitlab.v4.objects.ProjectClusterManager` method), 280
- `get()` (`gitlab.v4.objects.ProjectCommitDiscussionManager` method), 283
- `get()` (`gitlab.v4.objects.ProjectCommitDiscussionNoteManager` method), 285
- `get()` (`gitlab.v4.objects.ProjectCommitManager` method), 286
- `get()` (`gitlab.v4.objects.ProjectCustomAttributeManager` method), 288
- `get()` (`gitlab.v4.objects.ProjectDeploymentManager` method), 290
- `get()` (`gitlab.v4.objects.ProjectDeployTokenManager` method), 289
- `get()` (`gitlab.v4.objects.ProjectEnvironmentManager` method), 293
- `get()` (`gitlab.v4.objects.ProjectExportManager` method), 294
- `get()` (`gitlab.v4.objects.ProjectFileManager` method), 297
- `get()` (`gitlab.v4.objects.ProjectHookManager` method), 301
- `get()` (`gitlab.v4.objects.ProjectImportManager` method), 302
- `get()` (`gitlab.v4.objects.ProjectIntegrationManager` method), 303
- `get()` (`gitlab.v4.objects.ProjectInvitationManager` method), 304
- `get()` (`gitlab.v4.objects.ProjectIssueAwardEmojiManager` method), 306
- `get()` (`gitlab.v4.objects.ProjectIssueDiscussionManager` method), 307
- `get()` (`gitlab.v4.objects.ProjectIssueDiscussionNoteManager` method), 309
- `get()` (`gitlab.v4.objects.ProjectIssueManager` method), 311
- `get()` (`gitlab.v4.objects.ProjectIssueNoteAwardEmojiManager` method), 312
- `get()` (`gitlab.v4.objects.ProjectIssueNoteManager` method), 313
- `get()` (`gitlab.v4.objects.ProjectIssueResourceLabelEventManager` method), 314
- `get()` (`gitlab.v4.objects.ProjectIssueResourceMilestoneEventManager` method), 315
- `get()` (`gitlab.v4.objects.ProjectIssueResourceStateEventManager` method), 316
- `get()` (`gitlab.v4.objects.ProjectIssuesStatisticsManager` method), 317
- `get()` (`gitlab.v4.objects.ProjectJobManager` method), 321
- `get()` (`gitlab.v4.objects.ProjectKeyManager` method), 322
- `get()` (`gitlab.v4.objects.ProjectLabelManager` method), 323
- `get()` (`gitlab.v4.objects.ProjectManager` method), 328
- `get()` (`gitlab.v4.objects.ProjectMemberAllManager` method), 332
- `get()` (`gitlab.v4.objects.ProjectMemberManager` method), 333
- `get()` (`gitlab.v4.objects.ProjectMergeRequestApprovalManager` method), 337
- `get()` (`gitlab.v4.objects.ProjectMergeRequestApprovalRuleManager` method), 340
- `get()` (`gitlab.v4.objects.ProjectMergeRequestApprovalStateManager` method), 340
- `get()` (`gitlab.v4.objects.ProjectMergeRequestAwardEmojiManager` method), 341
- `get()` (`gitlab.v4.objects.ProjectMergeRequestDiffManager` method), 342
- `get()` (`gitlab.v4.objects.ProjectMergeRequestDiscussionManager` method), 343
- `get()` (`gitlab.v4.objects.ProjectMergeRequestDiscussionNoteManager` method), 344
- `get()` (`gitlab.v4.objects.ProjectMergeRequestManager` method), 346
- `get()` (`gitlab.v4.objects.ProjectMergeRequestNoteAwardEmojiManager` method), 347
- `get()` (`gitlab.v4.objects.ProjectMergeRequestNoteManager` method), 348
- `get()` (`gitlab.v4.objects.ProjectMergeRequestResourceLabelEventManager` method), 350
- `get()` (`gitlab.v4.objects.ProjectMergeRequestResourceMilestoneEventManager` method), 350
- `get()` (`gitlab.v4.objects.ProjectMergeRequestResourceStateEventManager` method), 351
- `get()` (`gitlab.v4.objects.ProjectMilestoneManager` method), 354
- `get()` (`gitlab.v4.objects.ProjectNoteManager` method), 355

`get()` (`gitlab.v4.objects.ProjectNotificationSettingsManager` method), 356

`get()` (`gitlab.v4.objects.ProjectPackageManager` method), 357

`get()` (`gitlab.v4.objects.ProjectPagesDomainManager` method), 359

`get()` (`gitlab.v4.objects.ProjectPipelineManager` method), 362

`get()` (`gitlab.v4.objects.ProjectPipelineScheduleManager` method), 363

`get()` (`gitlab.v4.objects.ProjectPipelineTestReportManager` method), 365

`get()` (`gitlab.v4.objects.ProjectPipelineTestReportSummaryManager` method), 366

`get()` (`gitlab.v4.objects.ProjectProtectedBranchManager` method), 367

`get()` (`gitlab.v4.objects.ProjectProtectedEnvironmentManager` method), 368

`get()` (`gitlab.v4.objects.ProjectProtectedTagManager` method), 369

`get()` (`gitlab.v4.objects.ProjectPushRulesManager` method), 371

`get()` (`gitlab.v4.objects.ProjectRegistryTagManager` method), 372

`get()` (`gitlab.v4.objects.ProjectReleaseLinkManager` method), 374

`get()` (`gitlab.v4.objects.ProjectReleaseManager` method), 375

`get()` (`gitlab.v4.objects.ProjectServiceManager` method), 377

`get()` (`gitlab.v4.objects.ProjectSnippetAwardEmojiManager` method), 379

`get()` (`gitlab.v4.objects.ProjectSnippetDiscussionManager` method), 380

`get()` (`gitlab.v4.objects.ProjectSnippetDiscussionNoteManager` method), 381

`get()` (`gitlab.v4.objects.ProjectSnippetManager` method), 382

`get()` (`gitlab.v4.objects.ProjectSnippetNoteAwardEmojiManager` method), 383

`get()` (`gitlab.v4.objects.ProjectSnippetNoteManager` method), 384

`get()` (`gitlab.v4.objects.ProjectStorageManager` method), 384

`get()` (`gitlab.v4.objects.ProjectTagManager` method), 385

`get()` (`gitlab.v4.objects.ProjectTriggerManager` method), 386

`get()` (`gitlab.v4.objects.ProjectVariableManager` method), 388

`get()` (`gitlab.v4.objects.ProjectWikiManager` method), 389

`get()` (`gitlab.v4.objects.RegistryRepositoryManager` method), 390

`get()` (`gitlab.v4.objects.RunnerManager` method), 397

`get()` (`gitlab.v4.objects.SnippetManager` method), 401

`get()` (`gitlab.v4.objects.TopicManager` method), 404

`get()` (`gitlab.v4.objects.UserCustomAttributeManager` method), 409

`get()` (`gitlab.v4.objects.UserEmailManager` method), 410

`get()` (`gitlab.v4.objects.UserGPGKeyManager` method), 412

`get()` (`gitlab.v4.objects.UserImpersonationTokenManager` method), 413

`get()` (`gitlab.v4.objects.UserKeyManager` method), 414

`get()` (`gitlab.v4.objects.UserManager` method), 416

`get()` (`gitlab.v4.objects.UserMembershipManager` method), 417

`get()` (`gitlab.v4.objects.UserStatusManager` method), 420

`get()` (`gitlab.v4.objects.VariableManager` method), 421

`get_id()` (`gitlab.base.RESTObject` method), 423

`get_license()` (`gitlab.Gitlab` method), 173

`GetMixin` (class in `gitlab.mixins`), 446

`GetWithoutIdMixin` (class in `gitlab.mixins`), 447

`Gitignore` (class in `gitlab.v4.objects`), 201

`GitignoreManager` (class in `gitlab.v4.objects`), 201

`gitignores` (`gitlab.Gitlab` attribute), 174

`Gitlab` (class in `gitlab`), 171

`gitlab` (`gitlab.base.RESTManager` attribute), 422

`gitlab` (`gitlab.mixins.CreateMixin` attribute), 445

`gitlab` (`gitlab.mixins.CRUDMixin` attribute), 445

`gitlab` (`gitlab.mixins.DeleteMixin` attribute), 446

`gitlab` (`gitlab.mixins.GetMixin` attribute), 447

`gitlab` (`gitlab.mixins.GetWithoutIdMixin` attribute), 447

`gitlab` (`gitlab.mixins.ListMixin` attribute), 447

`gitlab` (`gitlab.mixins.NoUpdateMixin` attribute), 448

`gitlab` (`gitlab.mixins.RetrieveMixin` attribute), 449

`gitlab` (`gitlab.mixins.SetMixin` attribute), 449

`gitlab` (`gitlab.mixins.UpdateMixin` attribute), 452

`gitlab` (`gitlab.v4.objects.ApplicationAppearanceManager` attribute), 183

`gitlab` (`gitlab.v4.objects.ApplicationManager` attribute), 184

`gitlab` (`gitlab.v4.objects.ApplicationSettingsManager` attribute), 186

`gitlab` (`gitlab.v4.objects.AuditEventManager` attribute), 187

`gitlab` (`gitlab.v4.objects.BroadcastMessageManager` attribute), 189

`gitlab` (`gitlab.v4.objects.CiLintManager` attribute), 189

`gitlab` (`gitlab.v4.objects.CurrentUserEmailManager` attribute), 191

`gitlab` (`gitlab.v4.objects.CurrentUserGPGKeyManager` attribute), 192

`gitlab` (`gitlab.v4.objects.CurrentUserKeyManager` attribute), 193



- gitlab (*gitlab.v4.objects.CurrentUserManager attribute*), 193
- gitlab (*gitlab.v4.objects.CurrentUserStatusManager attribute*), 194
- gitlab (*gitlab.v4.objects.DeployKeyManager attribute*), 195
- gitlab (*gitlab.v4.objects.DeployTokenManager attribute*), 195
- gitlab (*gitlab.v4.objects.DockerfileManager attribute*), 196
- gitlab (*gitlab.v4.objects.EventManager attribute*), 196
- gitlab (*gitlab.v4.objects.FeatureManager attribute*), 197
- gitlab (*gitlab.v4.objects.GenericPackageManager attribute*), 198
- gitlab (*gitlab.v4.objects.GeoNodeManager attribute*), 201
- gitlab (*gitlab.v4.objects.GitignoreManager attribute*), 202
- gitlab (*gitlab.v4.objects.GitlabciyamlManager attribute*), 203
- gitlab (*gitlab.v4.objects.GroupAccessRequestManager attribute*), 207
- gitlab (*gitlab.v4.objects.GroupAccessTokenManager attribute*), 208
- gitlab (*gitlab.v4.objects.GroupAuditEventManager attribute*), 208
- gitlab (*gitlab.v4.objects.GroupBadgeManager attribute*), 209
- gitlab (*gitlab.v4.objects.GroupBillableMemberManager attribute*), 210
- gitlab (*gitlab.v4.objects.GroupBillableMemberMembershipManager attribute*), 211
- gitlab (*gitlab.v4.objects.GroupBoardListManager attribute*), 212
- gitlab (*gitlab.v4.objects.GroupBoardManager attribute*), 212
- gitlab (*gitlab.v4.objects.GroupClusterManager attribute*), 214
- gitlab (*gitlab.v4.objects.GroupCustomAttributeManager attribute*), 215
- gitlab (*gitlab.v4.objects.GroupDeployTokenManager attribute*), 216
- gitlab (*gitlab.v4.objects.GroupEpicAwardEmojiManager attribute*), 218
- gitlab (*gitlab.v4.objects.GroupEpicDiscussionNoteManager attribute*), 219
- gitlab (*gitlab.v4.objects.GroupEpicIssueManager attribute*), 220
- gitlab (*gitlab.v4.objects.GroupEpicManager attribute*), 221
- gitlab (*gitlab.v4.objects.GroupEpicNoteAwardEmojiManager attribute*), 222
- gitlab (*gitlab.v4.objects.GroupEpicNoteManager attribute*), 223
- gitlab (*gitlab.v4.objects.GroupEpicResourceLabelEventManager attribute*), 224
- gitlab (*gitlab.v4.objects.GroupExportManager attribute*), 225
- gitlab (*gitlab.v4.objects.GroupHookManager attribute*), 227
- gitlab (*gitlab.v4.objects.GroupImportManager attribute*), 227
- gitlab (*gitlab.v4.objects.GroupInvitationManager attribute*), 229
- gitlab (*gitlab.v4.objects.GroupIssueManager attribute*), 230
- gitlab (*gitlab.v4.objects.GroupIssuesStatisticsManager attribute*), 230
- gitlab (*gitlab.v4.objects.GroupIterationManager attribute*), 231
- gitlab (*gitlab.v4.objects.GroupLabelManager attribute*), 232
- gitlab (*gitlab.v4.objects.GroupManager attribute*), 235
- gitlab (*gitlab.v4.objects.GroupMemberAllManager attribute*), 236
- gitlab (*gitlab.v4.objects.GroupMemberManager attribute*), 237
- gitlab (*gitlab.v4.objects.GroupMergeRequestManager attribute*), 238
- gitlab (*gitlab.v4.objects.GroupMilestoneManager attribute*), 240
- gitlab (*gitlab.v4.objects.GroupNotificationSettingsManager attribute*), 241
- gitlab (*gitlab.v4.objects.GroupPackageManager attribute*), 242
- gitlab (*gitlab.v4.objects.GroupProjectManager attribute*), 243
- gitlab (*gitlab.v4.objects.GroupPushRulesManager attribute*), 244
- gitlab (*gitlab.v4.objects.GroupRegistryRepositoryManager attribute*), 245
- gitlab (*gitlab.v4.objects.GroupRunnerManager attribute*), 245
- gitlab (*gitlab.v4.objects.GroupVariableManager attribute*), 247
- gitlab (*gitlab.v4.objects.GroupWikiManager attribute*), 248
- gitlab (*gitlab.v4.objects.HookManager attribute*), 249
- gitlab (*gitlab.v4.objects.IssueManager attribute*), 251
- gitlab (*gitlab.v4.objects.IssuesStatisticsManager attribute*), 251
- gitlab (*gitlab.v4.objects.KeyManager attribute*), 252
- gitlab (*gitlab.v4.objects.LDAPGroupManager attribute*), 253
- gitlab (*gitlab.v4.objects.LicenseManager attribute*), 254
- gitlab (*gitlab.v4.objects.MergeRequestManager attribute*), 254

- tribute), 255
- gitlab (gitlab.v4.objects.NamespaceManager attribute), 257
- gitlab (gitlab.v4.objects.NotificationSettingsManager attribute), 258
- gitlab (gitlab.v4.objects.PagesDomainManager attribute), 258
- gitlab (gitlab.v4.objects.PersonalAccessTokenManager attribute), 259
- gitlab (gitlab.v4.objects.ProjectAccessRequestManager attribute), 267
- gitlab (gitlab.v4.objects.ProjectAccessTokenManager attribute), 267
- gitlab (gitlab.v4.objects.ProjectAdditionalStatisticsManager attribute), 268
- gitlab (gitlab.v4.objects.ProjectApprovalManager attribute), 269
- gitlab (gitlab.v4.objects.ProjectApprovalRuleManager attribute), 270
- gitlab (gitlab.v4.objects.ProjectArtifactManager attribute), 271
- gitlab (gitlab.v4.objects.ProjectAuditEventManager attribute), 273
- gitlab (gitlab.v4.objects.ProjectAuditManager attribute), 274
- gitlab (gitlab.v4.objects.ProjectBadgeManager attribute), 275
- gitlab (gitlab.v4.objects.ProjectBoardListManager attribute), 276
- gitlab (gitlab.v4.objects.ProjectBoardManager attribute), 276
- gitlab (gitlab.v4.objects.ProjectBranchManager attribute), 277
- gitlab (gitlab.v4.objects.ProjectCiLintManager attribute), 278
- gitlab (gitlab.v4.objects.ProjectClusterManager attribute), 280
- gitlab (gitlab.v4.objects.ProjectCommitCommentManager attribute), 283
- gitlab (gitlab.v4.objects.ProjectCommitDiscussionManager attribute), 284
- gitlab (gitlab.v4.objects.ProjectCommitDiscussionNoteManager attribute), 285
- gitlab (gitlab.v4.objects.ProjectCommitManager attribute), 286
- gitlab (gitlab.v4.objects.ProjectCommitStatusManager attribute), 287
- gitlab (gitlab.v4.objects.ProjectCustomAttributeManager attribute), 288
- gitlab (gitlab.v4.objects.ProjectDeploymentManager attribute), 290
- gitlab (gitlab.v4.objects.ProjectDeploymentMergeRequestManager attribute), 291
- gitlab (gitlab.v4.objects.ProjectDeployTokenManager attribute), 289
- gitlab (gitlab.v4.objects.ProjectEnvironmentManager attribute), 293
- gitlab (gitlab.v4.objects.ProjectEventManager attribute), 294
- gitlab (gitlab.v4.objects.ProjectExportManager attribute), 294
- gitlab (gitlab.v4.objects.ProjectFileManager attribute), 298
- gitlab (gitlab.v4.objects.ProjectForkManager attribute), 300
- gitlab (gitlab.v4.objects.ProjectHookManager attribute), 301
- gitlab (gitlab.v4.objects.ProjectImportManager attribute), 302
- gitlab (gitlab.v4.objects.ProjectIntegrationManager attribute), 303
- gitlab (gitlab.v4.objects.ProjectInvitationManager attribute), 304
- gitlab (gitlab.v4.objects.ProjectIssueAwardEmojiManager attribute), 307
- gitlab (gitlab.v4.objects.ProjectIssueDiscussionManager attribute), 308
- gitlab (gitlab.v4.objects.ProjectIssueDiscussionNoteManager attribute), 309
- gitlab (gitlab.v4.objects.ProjectIssueLinkManager attribute), 310
- gitlab (gitlab.v4.objects.ProjectIssueManager attribute), 312
- gitlab (gitlab.v4.objects.ProjectIssueNoteAwardEmojiManager attribute), 313
- gitlab (gitlab.v4.objects.ProjectIssueNoteManager attribute), 313
- gitlab (gitlab.v4.objects.ProjectIssueResourceLabelEventManager attribute), 314
- gitlab (gitlab.v4.objects.ProjectIssueResourceMilestoneEventManager attribute), 315
- gitlab (gitlab.v4.objects.ProjectIssueResourceStateEventManager attribute), 316
- gitlab (gitlab.v4.objects.ProjectIssuesStatisticsManager attribute), 317
- gitlab (gitlab.v4.objects.ProjectIterationManager attribute), 317
- gitlab (gitlab.v4.objects.ProjectJobManager attribute), 321
- gitlab (gitlab.v4.objects.ProjectKeyManager attribute), 322
- gitlab (gitlab.v4.objects.ProjectLabelManager attribute), 324
- gitlab (gitlab.v4.objects.ProjectManager attribute), 329
- gitlab (gitlab.v4.objects.ProjectMemberAllManager attribute), 332
- gitlab (gitlab.v4.objects.ProjectMemberManager attribute), 333

[`gitlab \(gitlab.v4.objects.ProjectMergeRequestApprovalManager attribute\)`](#), 337  
[`gitlab \(gitlab.v4.objects.ProjectMergeRequestApprovalRulesManager attribute\)`](#), 340  
[`gitlab \(gitlab.v4.objects.ProjectMergeRequestApprovalStagesManager attribute\)`](#), 341  
[`gitlab \(gitlab.v4.objects.ProjectMergeRequestAwardEmojiManager attribute\)`](#), 342  
[`gitlab \(gitlab.v4.objects.ProjectMergeRequestDiffManager attribute\)`](#), 343  
[`gitlab \(gitlab.v4.objects.ProjectMergeRequestDiscussionManager attribute\)`](#), 344  
[`gitlab \(gitlab.v4.objects.ProjectMergeRequestDiscussionNotesManager attribute\)`](#), 345  
[`gitlab \(gitlab.v4.objects.ProjectMergeRequestManager attribute\)`](#), 347  
[`gitlab \(gitlab.v4.objects.ProjectMergeRequestNoteAwardEmojiManager attribute\)`](#), 348  
[`gitlab \(gitlab.v4.objects.ProjectMergeRequestNoteManager attribute\)`](#), 348  
[`gitlab \(gitlab.v4.objects.ProjectMergeRequestPipelineManager attribute\)`](#), 349  
[`gitlab \(gitlab.v4.objects.ProjectMergeRequestResourceLabelManager attribute\)`](#), 350  
[`gitlab \(gitlab.v4.objects.ProjectMergeRequestResourceMilestoneManager attribute\)`](#), 351  
[`gitlab \(gitlab.v4.objects.ProjectMergeRequestResourceStagesManager attribute\)`](#), 352  
[`gitlab \(gitlab.v4.objects.ProjectMergeTrainManager attribute\)`](#), 352  
[`gitlab \(gitlab.v4.objects.ProjectMilestoneManager attribute\)`](#), 354  
[`gitlab \(gitlab.v4.objects.ProjectNoteManager attribute\)`](#), 355  
[`gitlab \(gitlab.v4.objects.ProjectNotificationSettingsManager attribute\)`](#), 356  
[`gitlab \(gitlab.v4.objects.ProjectPackageFileManager attribute\)`](#), 357  
[`gitlab \(gitlab.v4.objects.ProjectPackageManager attribute\)`](#), 358  
[`gitlab \(gitlab.v4.objects.ProjectPagesDomainManager attribute\)`](#), 359  
[`gitlab \(gitlab.v4.objects.ProjectPipelineBridgeManager attribute\)`](#), 360  
[`gitlab \(gitlab.v4.objects.ProjectPipelineJobManager attribute\)`](#), 361  
[`gitlab \(gitlab.v4.objects.ProjectPipelineManager attribute\)`](#), 362  
[`gitlab \(gitlab.v4.objects.ProjectPipelineScheduleManager attribute\)`](#), 364  
[`gitlab \(gitlab.v4.objects.ProjectPipelineScheduleVariableManager attribute\)`](#), 365  
[`gitlab \(gitlab.v4.objects.ProjectPipelineTestReportManager attribute\)`](#), 365  
[`gitlab \(gitlab.v4.objects.ProjectPipelineTestReportSummaryManager attribute\)`](#), 366  
[`gitlab \(gitlab.v4.objects.ProjectPipelineVariableManager attribute\)`](#), 367  
[`gitlab \(gitlab.v4.objects.ProjectProtectedBranchManager attribute\)`](#), 368  
[`gitlab \(gitlab.v4.objects.ProjectProtectedEnvironmentManager attribute\)`](#), 369  
[`gitlab \(gitlab.v4.objects.ProjectProtectedTagManager attribute\)`](#), 370  
[`gitlab \(gitlab.v4.objects.ProjectPushRulesManager attribute\)`](#), 371  
[`gitlab \(gitlab.v4.objects.ProjectRegistryRepositoryManager attribute\)`](#), 372  
[`gitlab \(gitlab.v4.objects.ProjectRegistryTagManager attribute\)`](#), 373  
[`gitlab \(gitlab.v4.objects.ProjectReleaseLinkManager attribute\)`](#), 374  
[`gitlab \(gitlab.v4.objects.ProjectReleaseManager attribute\)`](#), 375  
[`gitlab \(gitlab.v4.objects.ProjectRemoteMirrorManager attribute\)`](#), 376  
[`gitlab \(gitlab.v4.objects.ProjectRunnerManager attribute\)`](#), 377  
[`gitlab \(gitlab.v4.objects.ProjectServiceManager attribute\)`](#), 377  
[`gitlab \(gitlab.v4.objects.ProjectSnippetAwardEmojiManager attribute\)`](#), 379  
[`gitlab \(gitlab.v4.objects.ProjectSnippetDiscussionManager attribute\)`](#), 380  
[`gitlab \(gitlab.v4.objects.ProjectSnippetDiscussionNoteManager attribute\)`](#), 381  
[`gitlab \(gitlab.v4.objects.ProjectSnippetManager attribute\)`](#), 382  
[`gitlab \(gitlab.v4.objects.ProjectSnippetNoteAwardEmojiManager attribute\)`](#), 383  
[`gitlab \(gitlab.v4.objects.ProjectSnippetNoteManager attribute\)`](#), 384  
[`gitlab \(gitlab.v4.objects.ProjectStorageManager attribute\)`](#), 385  
[`gitlab \(gitlab.v4.objects.ProjectTagManager attribute\)`](#), 386  
[`gitlab \(gitlab.v4.objects.ProjectTriggerManager attribute\)`](#), 387  
[`gitlab \(gitlab.v4.objects.ProjectUserManager attribute\)`](#), 387  
[`gitlab \(gitlab.v4.objects.ProjectVariableManager attribute\)`](#), 388  
[`gitlab \(gitlab.v4.objects.ProjectWikiManager attribute\)`](#), 390  
[`gitlab \(gitlab.v4.objects.RegistryRepositoryManager attribute\)`](#), 390  
[`gitlab \(gitlab.v4.objects.RunnerAllManager attribute\)`](#), 395

`gitlab` (`gitlab.v4.objects.RunnerJobManager` attribute), 395

`gitlab` (`gitlab.v4.objects.RunnerManager` attribute), 397

`gitlab` (`gitlab.v4.objects.SharedProjectManager` attribute), 398

`gitlab` (`gitlab.v4.objects.SidekiqManager` attribute), 399

`gitlab` (`gitlab.v4.objects.SnippetManager` attribute), 401

`gitlab` (`gitlab.v4.objects.StarredProjectManager` attribute), 402

`gitlab` (`gitlab.v4.objects.TODOManager` attribute), 403

`gitlab` (`gitlab.v4.objects.TopicManager` attribute), 405

`gitlab` (`gitlab.v4.objects.UserActivitiesManager` attribute), 409

`gitlab` (`gitlab.v4.objects.UserCustomAttributeManager` attribute), 410

`gitlab` (`gitlab.v4.objects.UserEmailManager` attribute), 410

`gitlab` (`gitlab.v4.objects.UserEventManager` attribute), 411

`gitlab` (`gitlab.v4.objects.UserGPGKeyManager` attribute), 412

`gitlab` (`gitlab.v4.objects.UserIdentityProviderManager` attribute), 412

`gitlab` (`gitlab.v4.objects.UserImpersonationTokenManager` attribute), 413

`gitlab` (`gitlab.v4.objects.UserKeyManager` attribute), 414

`gitlab` (`gitlab.v4.objects.UserManager` attribute), 416

`gitlab` (`gitlab.v4.objects.UserMembershipManager` attribute), 417

`gitlab` (`gitlab.v4.objects.UserPersonalAccessTokenManager` attribute), 418

`gitlab` (`gitlab.v4.objects.UserProjectManager` attribute), 419

`gitlab` (`gitlab.v4.objects.UserStatusManager` attribute), 421

`gitlab` (`gitlab.v4.objects.VariableManager` attribute), 422

`gitlab` command line option

- `--api-version`, 459
- `--config-file`, 458
- `--debug`, 458
- `--fancy`, 458
- `--fields`, 458
- `--gitlab`, 458
- `--help`, 458
- `--job-token`, 459
- `--oauth-token`, 459
- `--order-by`, 459
- `--output`, 458
- `--pagination`, 459
- `--per-page`, 459
- `--private-token`, 459
- `--server-url`, 459
- `--ssl-verify`, 459
- `--timeout`, 459
- `--user-agent`, 459
- `--verbose`, 458
- `--version`, 458

`-c`, 458

`-d`, 458

`-f`, 458

`-g`, 458

`-h`, 458

`-o`, 458

`-v`, 458

`gitlab.base`

- module, 422

`gitlab.cli`

- module, 424

`gitlab.config`

- module, 425

`gitlab.const`

- module, 426

`gitlab.exceptions`

- module, 427

`gitlab.mixins`

- module, 444

`gitlab.utils`

- module, 453

`gitlab.v4`

- module, 422

`gitlab.v4.objects`

- module, 182

`gitlab_resource_to_cls()` (in module `gitlab.cli`), 425

`gitlab-application` command line option

- `--help`, 459
- `-h`, 459

`gitlab-application-appearance` command line option

- `--help`, 460
- `-h`, 460

`gitlab-application-appearance-get` command line option

- `--help`, 461
- `--sudo`, 461
- `-h`, 461

`gitlab-application-appearance-update` command line option

- `--description`, 461
- `--email-header-and-footer-enabled`, 461
- `--favicon`, 461
- `--footer-message`, 461
- `--header-logo`, 461
- `--header-message`, 461
- `--help`, 461
- `--logo`, 461

```

--message-background-color, 461
--message-font-color, 461
--new-project-guidelines, 461
--sudo, 461
--title, 461
-h, 461
gitlab-application-create command line
    option
--confidential, 460
--help, 459
--name, 459
--redirect-uri, 460
--scopes, 460
--sudo, 459
-h, 459
gitlab-application-delete command line
    option
--help, 460
--id, 460
--sudo, 460
-h, 460
gitlab-application-list command line option
--all, 460
--get-all, 460
--help, 460
--page, 460
--per-page, 460
--sudo, 460
-h, 460
gitlab-application-settings command line
    option
--help, 462
-h, 462
gitlab-application-settings-get command
    line option
--help, 462
--sudo, 462
-h, 462
gitlab-application-settings-update command
    line option
--after-sign-out-path, 465
--allow-local-requests-from-hooks-and-services, 465
--allow-local-requests-from-system-hooks, 465
--allow-local-requests-from-web-hooks-and-services, 465
--asset-proxy-allowlist, 465
--asset-proxy-enabled, 465
--asset-proxy-url, 465
--asset-proxy-whitelist, 465
--container-registry-token-expire-delay, 465
--created-at, 464
--default-branch-protection, 464
--default-group-visibility, 464
--default-project-visibility, 464
--default-projects-limit, 464
--default-snippet-visibility, 464
--disabled-oauth-sign-in-sources, 464
--domain-allowlist, 464
--domain-blacklist, 464
--domain-blacklist-enabled, 464
--domain-denylist, 464
--domain-denylist-enabled, 464
--domain-whitelist, 464
--dsa-key-restriction, 465
--ecdsa-key-restriction, 465
--ed25519-key-restriction, 465
--enforce-terms, 465
--external-authorization-service-default-label, 464
--external-authorization-service-enabled, 464
--external-authorization-service-timeout, 464
--external-authorization-service-url, 464
--file-template-project-id, 465
--first-day-of-week, 465
--geo-node-allowed-ips, 465
--gravatar-enabled, 464
--help, 463
--home-page-url, 464
--id, 464
--import-sources, 464
--instance-statistics-visibility-private, 465
--local-markdown-version, 465
--max-attachment-size, 464
--outbound-local-requests-whitelist, 464
--password-authentication-enabled-for-web, 464
--performance-bar-allowed-group-id, 465
--plantuml-enabled, 465
--plantuml-url, 465
--polling-interval-multiplier, 465
--repository-storages, 465
--restricted-visibility-levels, 464
--rsa-key-restriction, 465
--session-expire-delay, 464
--sign-in-text, 464
--signup-enabled, 464
--sudo, 464
--terminal-max-session-time, 465
--terms, 465
--updated-at, 464
--user-oauth-applications, 465
--user-show-add-ssh-key-message, 465

```



-h, 463  
gitlab-audit-event command line option  
--help, 466  
-h, 466  
gitlab-audit-event-get command line option  
--help, 466  
--id, 466  
--sudo, 466  
-h, 466  
gitlab-audit-event-list command line option  
--all, 466  
--created-after, 466  
--created-before, 466  
--entity-id, 466  
--entity-type, 466  
--get-all, 466  
--help, 466  
--page, 466  
--per-page, 466  
--sudo, 466  
-h, 466  
gitlab-broadcast-message command line option  
--help, 467  
-h, 467  
gitlab-broadcast-message-create command line option  
--color, 467  
--ends-at, 467  
--font, 467  
--help, 467  
--message, 467  
--starts-at, 467  
--sudo, 467  
--target-access-levels, 467  
-h, 467  
gitlab-broadcast-message-delete command line option  
--help, 467  
--id, 467  
--sudo, 467  
-h, 467  
gitlab-broadcast-message-get command line option  
--help, 467  
--id, 467  
--sudo, 467  
-h, 467  
gitlab-broadcast-message-list command line option  
--all, 468  
--get-all, 468  
--help, 468  
--page, 468  
--per-page, 468  
--sudo, 468  
-h, 468  
gitlab-broadcast-message-update command line option  
--color, 468  
--ends-at, 468  
--font, 468  
--help, 468  
--id, 468  
--message, 468  
--starts-at, 468  
--sudo, 468  
--target-access-levels, 468  
-h, 468  
gitlab-ci-lint command line option  
--help, 469  
-h, 469  
gitlab-ci-lint-create command line option  
--content, 469  
--help, 469  
--include-jobs, 469  
--include-merged-yaml, 469  
--sudo, 469  
-h, 469  
gitlab-ci-lint-validate command line option  
--content, 469  
--help, 469  
--include-jobs, 469  
--include-merged-yaml, 469  
-h, 469  
gitlab-current-user command line option  
--help, 469  
-h, 469  
gitlab-current-user-email command line option  
--help, 470  
-h, 470  
gitlab-current-user-email-create command line option  
--email, 470  
--help, 470  
--sudo, 470  
-h, 470  
gitlab-current-user-email-delete command line option  
--help, 470  
--id, 470  
--sudo, 470  
-h, 470  
gitlab-current-user-email-get command line option  
--help, 470  
--id, 470

```

--sudo, 470
-h, 470
gitlab-current-user-email-list command line
    option
--all, 471
--get-all, 471
--help, 471
--page, 471
--per-page, 471
--sudo, 471
-h, 471
gitlab-current-user-get command line option
--help, 470
--sudo, 470
-h, 470
gitlab-current-user-gpg-key command line
    option
--help, 471
-h, 471
gitlab-current-user-gpg-key-create command
    line option
--help, 471
--key, 471
--sudo, 471
-h, 471
gitlab-current-user-gpg-key-delete command
    line option
--help, 471
--id, 471
--sudo, 471
-h, 471
gitlab-current-user-gpg-key-get command
    line option
--help, 472
--id, 472
--sudo, 472
-h, 472
gitlab-current-user-gpg-key-list command
    line option
--all, 472
--get-all, 472
--help, 472
--page, 472
--per-page, 472
--sudo, 472
-h, 472
gitlab-current-user-key command line option
--help, 472
-h, 472
gitlab-current-user-key-create command line
    option
--help, 472
--key, 472
--sudo, 472
--title, 472
-h, 472
gitlab-current-user-key-delete command line
    option
--help, 473
--id, 473
--sudo, 473
-h, 473
gitlab-current-user-key-get command line
    option
--help, 473
--id, 473
--sudo, 473
-h, 473
gitlab-current-user-key-list command line
    option
--all, 473
--get-all, 473
--help, 473
--page, 473
--per-page, 473
--sudo, 473
-h, 473
gitlab-current-user-status command line
    option
--help, 473
-h, 473
gitlab-current-user-status-get command line
    option
--help, 474
--sudo, 474
-h, 474
gitlab-current-user-status-update command
    line option
--emoji, 474
--help, 474
--message, 474
--sudo, 474
-h, 474
gitlab-deploy-key command line option
--help, 474
-h, 474
gitlab-deploy-key-list command line option
--all, 474
--get-all, 474
--help, 474
--page, 474
--per-page, 474
--sudo, 474
-h, 474
gitlab-deploy-token command line option
--help, 475
-h, 475

```

gitlab-deploy-token-list command line option

--all, 475  
--get-all, 475  
--help, 475  
--page, 475  
--per-page, 475  
--sudo, 475  
-h, 475

gitlab-dockerfile command line option

--help, 475  
-h, 475

gitlab-dockerfile-get command line option

--help, 475  
--name, 475  
--sudo, 475  
-h, 475

gitlab-dockerfile-list command line option

--all, 476  
--get-all, 476  
--help, 476  
--page, 476  
--per-page, 476  
--sudo, 476  
-h, 476

gitlab-event command line option

--help, 476  
-h, 476

gitlab-event-list command line option

--action, 476  
--after, 476  
--all, 477  
--before, 476  
--get-all, 476  
--help, 476  
--page, 476  
--per-page, 476  
--sort, 476  
--sudo, 476  
--target-type, 476  
-h, 476

gitlab-feature command line option

--help, 477  
-h, 477

gitlab-feature-delete command line option

--help, 477  
--name, 477  
--sudo, 477  
-h, 477

gitlab-feature-list command line option

--all, 477  
--get-all, 477  
--help, 477  
--page, 477

--per-page, 477

--sudo, 477

-h, 477

gitlab-generic-package command line option

--help, 478

-h, 478

gitlab-generic-package-download command line option

--file-name, 478  
--help, 478  
--package-name, 478  
--package-version, 478  
--project-id, 478  
--sudo, 478  
-h, 478

gitlab-generic-package-upload command line option

--file-name, 478  
--help, 478  
--package-name, 478  
--package-version, 478  
--path, 478  
--project-id, 478  
--sudo, 478  
-h, 478

gitlab-geo-node command line option

--help, 479  
-h, 479

gitlab-geo-node-current-failures command line option

--help, 479  
--id, 479  
-h, 479

gitlab-geo-node-delete command line option

--help, 479  
--id, 479  
--sudo, 479  
-h, 479

gitlab-geo-node-get command line option

--help, 479  
--id, 479  
--sudo, 479  
-h, 479

gitlab-geo-node-list command line option

--all, 480  
--get-all, 480  
--help, 480  
--page, 480  
--per-page, 480  
--sudo, 480  
-h, 480

gitlab-geo-node-repair command line option

--help, 480  
--id, 480



```

    -h, 480
gitlab-geo-node-status command line option
    --help, 480
    --id, 480
    -h, 480
gitlab-geo-node-update command line option
    --enabled, 480
    --files-max-capacity, 481
    --help, 480
    --id, 480
    --repos-max-capacity, 481
    --sudo, 480
    --url, 481
    -h, 480
gitlab-gitignore command line option
    --help, 481
    -h, 481
gitlab-gitignore-get command line option
    --help, 481
    --name, 481
    --sudo, 481
    -h, 481
gitlab-gitignore-list command line option
    --all, 481
    --get-all, 481
    --help, 481
    --page, 481
    --per-page, 481
    --sudo, 481
    -h, 481
gitlab-gitlabciyaml command line option
    --help, 482
    -h, 482
gitlab-gitlabciyaml-get command line option
    --help, 482
    --name, 482
    --sudo, 482
    -h, 482
gitlab-gitlabciyaml-list command line option
    --all, 482
    --get-all, 482
    --help, 482
    --page, 482
    --per-page, 482
    --sudo, 482
    -h, 482
gitlab-group command line option
    --help, 483
    -h, 483
gitlab-group-access-request command line
    option
    --help, 489
    -h, 489
gitlab-group-access-request-approve command
    line option
    --access-level, 489
    --group-id, 489
    --help, 489
    --id, 489
    --sudo, 489
    -h, 489
gitlab-group-access-request-create command
    line option
    --group-id, 489
    --help, 489
    --sudo, 489
    -h, 489
gitlab-group-access-request-delete command
    line option
    --group-id, 489
    --help, 489
    --id, 489
    --sudo, 489
    -h, 489
gitlab-group-access-request-list command
    line option
    --all, 490
    --get-all, 490
    --group-id, 490
    --help, 490
    --page, 490
    --per-page, 490
    --sudo, 490
    -h, 490
gitlab-group-access-token command line
    option
    --help, 490
    -h, 490
gitlab-group-access-token-create command
    line option
    --access-level, 490
    --expires-at, 490
    --group-id, 490
    --help, 490
    --name, 490
    --scopes, 490
    --sudo, 490
    -h, 490
gitlab-group-access-token-delete command
    line option
    --group-id, 491
    --help, 491
    --id, 491
    --sudo, 491
    -h, 491
gitlab-group-access-token-list command line
    option

```

```
--all, 491
--get-all, 491
--group-id, 491
--help, 491
--page, 491
--per-page, 491
--sudo, 491
-h, 491
gitlab-group-add-ldap-group-link command
  line option
  --cn, 483
  --group-access, 483
  --help, 483
  --id, 483
  --provider, 483
  -h, 483
gitlab-group-audit-event command line
  option
  --help, 491
  -h, 491
gitlab-group-audit-event-get command line
  option
  --group-id, 492
  --help, 492
  --id, 492
  --sudo, 492
  -h, 492
gitlab-group-audit-event-list command line
  option
  --all, 492
  --created-after, 492
  --created-before, 492
  --get-all, 492
  --group-id, 492
  --help, 492
  --page, 492
  --per-page, 492
  --sudo, 492
  -h, 492
gitlab-group-badge command line option
  --help, 492
  -h, 492
gitlab-group-badge-create command line
  option
  --group-id, 493
  --help, 493
  --image-url, 493
  --link-url, 493
  --sudo, 493
  -h, 493
gitlab-group-badge-delete command line
  option
  --group-id, 493
  --help, 493
  --id, 493
  --sudo, 493
  -h, 493
gitlab-group-badge-get command line option
  --group-id, 493
  --help, 493
  --id, 493
  --sudo, 493
  -h, 493
gitlab-group-badge-list command line option
  --all, 494
  --get-all, 494
  --group-id, 494
  --help, 493
  --page, 494
  --per-page, 494
  --sudo, 493
  -h, 493
gitlab-group-badge-render command line
  option
  --group-id, 494
  --help, 494
  --id, 494
  --image-url, 494
  --link-url, 494
  --sudo, 494
  -h, 494
gitlab-group-badge-update command line
  option
  --group-id, 494
  --help, 494
  --id, 494
  --image-url, 494
  --link-url, 494
  --sudo, 494
  -h, 494
gitlab-group-billable-member command line
  option
  --help, 495
  -h, 495
gitlab-group-billable-member-delete command
  line option
  --group-id, 495
  --help, 495
  --id, 495
  --sudo, 495
  -h, 495
gitlab-group-billable-member-list command
  line option
  --all, 495
  --get-all, 495
  --group-id, 495
  --help, 495
  --page, 495
```

```

--per-page, 495
--search, 495
--sort, 495
--sudo, 495
-h, 495
gitlab-group-billable-member-membership
    command line option
--help, 496
-h, 496
gitlab-group-billable-member-membership-list
    command line option
--all, 496
--get-all, 496
--group-id, 496
--help, 496
--page, 496
--per-page, 496
--sudo, 496
--user-id, 496
-h, 496
gitlab-group-board command line option
--help, 496
-h, 496
gitlab-group-board-create command line
    option
--group-id, 496
--help, 496
--name, 497
--sudo, 496
-h, 496
gitlab-group-board-delete command line
    option
--group-id, 497
--help, 497
--id, 497
--sudo, 497
-h, 497
gitlab-group-board-get command line option
--group-id, 497
--help, 497
--id, 497
--sudo, 497
-h, 497
gitlab-group-board-list command line option
--all, 497
--get-all, 497
--group-id, 497
--help, 497, 498
--page, 497
--per-page, 497
--sudo, 497
-h, 497, 498
gitlab-group-board-list-create command line
    option
--board-id, 498
--group-id, 498
--help, 498
--sudo, 498
-h, 498
gitlab-group-board-list-delete command line
    option
--board-id, 498
--group-id, 498
--help, 498
--id, 498
--sudo, 498
-h, 498
gitlab-group-board-list-get command line
    option
--board-id, 499
--group-id, 499
--help, 499
--id, 499
--sudo, 499
-h, 499
gitlab-group-board-list-list command line
    option
--all, 499
--board-id, 499
--get-all, 499
--group-id, 499
--help, 499
--page, 499
--per-page, 499
--sudo, 499
-h, 499
gitlab-group-board-list-update command line
    option
--board-id, 499
--group-id, 499
--help, 499
--id, 500
--position, 500
--sudo, 499
-h, 499
gitlab-group-board-update command line
    option
--group-id, 498
--help, 498
--id, 498
--sudo, 498
-h, 498
gitlab-group-cluster command line option
--help, 500
-h, 500
gitlab-group-cluster-create command line
    option
--domain, 500

```

- `--enabled`, 500
  - `--environment-scope`, 500
  - `--group-id`, 500
  - `--help`, 500
  - `--managed`, 500
  - `--name`, 500
  - `--platform-kubernetes-attributes`, 500
  - `--sudo`, 500
  - `-h`, 500
- gitlab-group-cluster-delete command line
  - option
  - `--group-id`, 500
  - `--help`, 500
  - `--id`, 500
  - `--sudo`, 500
  - `-h`, 500
- gitlab-group-cluster-get command line
  - option
  - `--group-id`, 501
  - `--help`, 501
  - `--id`, 501
  - `--sudo`, 501
  - `-h`, 501
- gitlab-group-cluster-list command line
  - option
  - `--all`, 501
  - `--get-all`, 501
  - `--group-id`, 501
  - `--help`, 501
  - `--page`, 501
  - `--per-page`, 501
  - `--sudo`, 501
  - `-h`, 501
- gitlab-group-cluster-update command line
  - option
  - `--domain`, 502
  - `--environment-scope`, 502
  - `--group-id`, 501
  - `--help`, 501
  - `--id`, 501
  - `--management-project-id`, 502
  - `--name`, 501
  - `--platform-kubernetes-attributes`, 502
  - `--sudo`, 501
  - `-h`, 501
- gitlab-group-create command line option
  - `--auto-devops-enabled`, 484
  - `--avatar`, 484
  - `--default-branch-protection`, 484
  - `--description`, 484
  - `--emails-disabled`, 484
  - `--extra-shared-runners-minutes-limit`, 484
  - `--help`, 483
  - `--lfs-enabled`, 484
  - `--membership-lock`, 484
  - `--mentions-disabled`, 484
  - `--name`, 484
  - `--parent-id`, 484
  - `--path`, 484
  - `--project-creation-level`, 484
  - `--request-access-enabled`, 484
  - `--require-two-factor-authentication`, 484
  - `--share-with-group-lock`, 484
  - `--shared-runners-minutes-limit`, 484
  - `--subgroup-creation-level`, 484
  - `--sudo`, 483
  - `--two-factor-grace-period`, 484
  - `--visibility`, 484
  - `-h`, 483
- gitlab-group-custom-attribute command line
  - option
  - `--help`, 502
  - `-h`, 502
- gitlab-group-custom-attribute-delete command line option
  - `--group-id`, 502
  - `--help`, 502
  - `--key`, 502
  - `--sudo`, 502
  - `-h`, 502
- gitlab-group-custom-attribute-get command line option
  - `--group-id`, 502
  - `--help`, 502
  - `--key`, 502
  - `--sudo`, 502
  - `-h`, 502
- gitlab-group-custom-attribute-list command line option
  - `--all`, 503
  - `--get-all`, 503
  - `--group-id`, 503
  - `--help`, 503
  - `--page`, 503
  - `--per-page`, 503
  - `--sudo`, 503
  - `-h`, 503
- gitlab-group-delete command line option
  - `--help`, 484
  - `--id`, 484
  - `--sudo`, 484
  - `-h`, 484
- gitlab-group-delete-ldap-group-link command line option
  - `--cn`, 485
  - `--help`, 485
  - `--id`, 485
  - `--provider`, 485

```

    -h, 485
gitlab-group-deploy-token command line
    option
    --help, 503
    -h, 503
gitlab-group-deploy-token-create command
    line option
    --expires-at, 503
    --group-id, 503
    --help, 503
    --name, 503
    --scopes, 503
    --sudo, 503
    --username, 503
    -h, 503
gitlab-group-deploy-token-delete command
    line option
    --group-id, 504
    --help, 504
    --id, 504
    --sudo, 504
    -h, 504
gitlab-group-deploy-token-get command line
    option
    --group-id, 504
    --help, 504
    --id, 504
    --sudo, 504
    -h, 504
gitlab-group-deploy-token-list command line
    option
    --all, 504
    --get-all, 504
    --group-id, 504
    --help, 504
    --page, 504
    --per-page, 504
    --scopes, 504
    --sudo, 504
    -h, 504
gitlab-group-descendant-group command line
    option
    --help, 505
    -h, 505
gitlab-group-descendant-group-list command
    line option
    --all, 505
    --all-available, 505
    --get-all, 505
    --group-id, 505
    --help, 505
    --min-access-level, 505
    --order-by, 505
    --owned, 505
    --page, 505
    --per-page, 505
    --search, 505
    --skip-groups, 505
    --sort, 505
    --statistics, 505
    --sudo, 505
    --with-custom-attributes, 505
    -h, 505
gitlab-group-epic command line option
    --help, 506
    -h, 506
gitlab-group-epic-award-emoji command line
    option
    --help, 508
    -h, 508
gitlab-group-epic-award-emoji-create
    command line option
    --epic-iid, 508
    --group-id, 508
    --help, 508
    --name, 508
    --sudo, 508
    -h, 508
gitlab-group-epic-award-emoji-delete
    command line option
    --epic-iid, 509
    --group-id, 509
    --help, 509
    --id, 509
    --sudo, 509
    -h, 509
gitlab-group-epic-award-emoji-get command
    line option
    --epic-iid, 509
    --group-id, 509
    --help, 509
    --id, 509
    --sudo, 509
    -h, 509
gitlab-group-epic-award-emoji-list command
    line option
    --all, 510
    --epic-iid, 509
    --get-all, 509
    --group-id, 509
    --help, 509
    --page, 509
    --per-page, 509
    --sudo, 509
    -h, 509
gitlab-group-epic-create command line
    option
    --description, 506

```

```
--end-date, 506
--group-id, 506
--help, 506
--labels, 506
--start-date, 506
--sudo, 506
--title, 506
-h, 506
gitlab-group-epic-delete command line
    option
--group-id, 506
--help, 506
--iid, 506
--sudo, 506
-h, 506
gitlab-group-epic-discussion-note command
    line option
--help, 510
-h, 510
gitlab-group-epic-discussion-note-create
    command line option
--body, 510
--created-at, 510
--discussion-id, 510
--epic-iid, 510
--group-id, 510
--help, 510
--sudo, 510
-h, 510
gitlab-group-epic-discussion-note-delete
    command line option
--discussion-id, 510
--epic-iid, 510
--group-id, 510
--help, 510
--id, 511
--sudo, 510
-h, 510
gitlab-group-epic-discussion-note-get
    command line option
--discussion-id, 511
--epic-iid, 511
--group-id, 511
--help, 511
--id, 511
--sudo, 511
-h, 511
gitlab-group-epic-discussion-note-update
    command line option
--body, 511
--discussion-id, 511
--epic-iid, 511
--group-id, 511
--help, 511
--id, 511
--sudo, 511
-h, 511
gitlab-group-epic-get command line option
--group-id, 507
--help, 507
--iid, 507
--sudo, 507
-h, 507
gitlab-group-epic-issue command line option
--help, 512
-h, 512
gitlab-group-epic-issue-create command line
    option
--epic-iid, 512
--group-id, 512
--help, 512
--issue-id, 512
--sudo, 512
-h, 512
gitlab-group-epic-issue-delete command line
    option
--epic-iid, 512
--epic-issue-id, 512
--group-id, 512
--help, 512
--sudo, 512
-h, 512
gitlab-group-epic-issue-list command line
    option
--all, 513
--epic-iid, 512
--get-all, 513
--group-id, 512
--help, 512
--page, 513
--per-page, 513
--sudo, 512
-h, 512
gitlab-group-epic-issue-update command line
    option
--epic-iid, 513
--epic-issue-id, 513
--group-id, 513
--help, 513
--move-after-id, 513
--move-before-id, 513
--sudo, 513
-h, 513
gitlab-group-epic-list command line option
--all, 507
--author-id, 507
--get-all, 507
--group-id, 507
```

```

--help, 507
--labels, 507
--order-by, 507
--page, 507
--per-page, 507
--search, 507
--sort, 507
--sudo, 507
-h, 507
gitlab-group-epic-note command line option
--help, 513
-h, 513
gitlab-group-epic-note-award-emoji command
line option
--help, 515
-h, 515
gitlab-group-epic-note-award-emoji-create
command line option
--epic-iid, 515
--group-id, 515
--help, 515
--name, 515
--note-id, 515
--sudo, 515
-h, 515
gitlab-group-epic-note-award-emoji-delete
command line option
--epic-iid, 516
--group-id, 516
--help, 516
--id, 516
--note-id, 516
--sudo, 516
-h, 516
gitlab-group-epic-note-award-emoji-get
command line option
--epic-iid, 516
--group-id, 516
--help, 516
--id, 516
--note-id, 516
--sudo, 516
-h, 516
gitlab-group-epic-note-award-emoji-list
command line option
--all, 517
--epic-iid, 516
--get-all, 517
--group-id, 516
--help, 516
--note-id, 516
--page, 517
--per-page, 517
--sudo, 516
--h, 516
gitlab-group-epic-note-create command line
option
--body, 514
--created-at, 514
--epic-iid, 514
--group-id, 513
--help, 513
--sudo, 513
-h, 513
gitlab-group-epic-note-delete command line
option
--epic-iid, 514
--group-id, 514
--help, 514
--id, 514
--sudo, 514
-h, 514
gitlab-group-epic-note-get command line
option
--epic-iid, 514
--group-id, 514
--help, 514
--id, 514
--sudo, 514
-h, 514
gitlab-group-epic-note-list command line
option
--all, 515
--epic-iid, 514
--get-all, 515
--group-id, 514
--help, 514
--page, 515
--per-page, 515
--sudo, 514
-h, 514
gitlab-group-epic-note-update command line
option
--body, 515
--epic-iid, 515
--group-id, 515
--help, 515
--id, 515
--sudo, 515
-h, 515
gitlab-group-epic-resource-label-event
command line option
--help, 517
-h, 517
gitlab-group-epic-resource-label-event-get
command line option
--epic-id, 517
--group-id, 517

```

```
--help, 517
--id, 517
--sudo, 517
-h, 517
gitlab-group-epic-resource-label-event-list
    command line option
    --all, 518
    --epic-id, 517
    --get-all, 518
    --group-id, 517
    --help, 517
    --page, 518
    --per-page, 518
    --sudo, 517
    -h, 517
gitlab-group-epic-update command line
    option
    --description, 508
    --end-date, 508
    --group-id, 508
    --help, 508
    --iid, 508
    --labels, 508
    --start-date, 508
    --sudo, 508
    --title, 508
    -h, 508
gitlab-group-export command line option
    --help, 518
    -h, 518
gitlab-group-export-create command line
    option
    --group-id, 518
    --help, 518
    --sudo, 518
    -h, 518
gitlab-group-export-download command line
    option
    --group-id, 518
    --help, 518
    --sudo, 518
    -h, 518
gitlab-group-export-get command line option
    --group-id, 518
    --help, 518
    --sudo, 518
    -h, 518
gitlab-group-get command line option
    --help, 485
    --id, 485
    --sudo, 485
    -h, 485
gitlab-group-hook command line option
    --help, 519
    -h, 519
gitlab-group-hook-create command line
    option
    --confidential-issues-events, 519
    --confidential-note-events, 519
    --deployment-events, 519
    --enable-ssl-verification, 520
    --group-id, 519
    --help, 519
    --issues-events, 519
    --job-events, 519
    --merge-requests-events, 519
    --note-events, 519
    --pipeline-events, 519
    --push-events, 519
    --releases-events, 520
    --subgroup-events, 520
    --sudo, 519
    --tag-push-events, 519
    --token, 520
    --url, 519
    --wiki-page-events, 519
    -h, 519
gitlab-group-hook-delete command line
    option
    --group-id, 520
    --help, 520
    --id, 520
    --sudo, 520
    -h, 520
gitlab-group-hook-get command line option
    --group-id, 520
    --help, 520
    --id, 520
    --sudo, 520
    -h, 520
gitlab-group-hook-list command line option
    --all, 521
    --get-all, 520
    --group-id, 520
    --help, 520
    --page, 520
    --per-page, 520
    --sudo, 520
    -h, 520
gitlab-group-hook-update command line
    option
    --confidential-issues-events, 521
    --confidential-note-events, 521
    --deployment-events, 521
    --enable-ssl-verification, 522
    --group-id, 521
    --help, 521
    --id, 521
```



- `--issues-events`, 521
- `--job-events`, 521
- `--merge-requests-events`, 521
- `--note-events`, 521
- `--pipeline-events`, 521
- `--push-events`, 521
- `--releases-events`, 522
- `--subgroup-events`, 522
- `--sudo`, 521
- `--tag-push-events`, 521
- `--token`, 522
- `--url`, 521
- `--wiki-page-events`, 521
- `-h`, 521
- `gitlab-group-import` command line option
  - `--help`, 522
  - `-h`, 522
- `gitlab-group-import-get` command line option
  - `--group-id`, 522
  - `--help`, 522
  - `--sudo`, 522
  - `-h`, 522
- `gitlab-group-invitation` command line option
  - `--help`, 522
  - `-h`, 522
- `gitlab-group-invitation-create` command line option
  - `--access-level`, 522
  - `--expires-at`, 523
  - `--group-id`, 522
  - `--help`, 522
  - `--invite-source`, 523
  - `--sudo`, 522
  - `--tasks-project-id`, 523
  - `--tasks-to-be-done`, 523
  - `-h`, 522
- `gitlab-group-invitation-delete` command line option
  - `--email`, 523
  - `--group-id`, 523
  - `--help`, 523
  - `--sudo`, 523
  - `-h`, 523
- `gitlab-group-invitation-get` command line option
  - `--email`, 523
  - `--group-id`, 523
  - `--help`, 523
  - `--sudo`, 523
  - `-h`, 523
- `gitlab-group-invitation-list` command line option
  - `--all`, 524
  - `--get-all`, 524
- `--group-id`, 523
- `--help`, 523
- `--page`, 523
- `--per-page`, 524
- `--query`, 523
- `--sudo`, 523
- `-h`, 523
- `gitlab-group-invitation-update` command line option
  - `--access-level`, 524
  - `--email`, 524
  - `--expires-at`, 524
  - `--group-id`, 524
  - `--help`, 524
  - `--sudo`, 524
  - `-h`, 524
- `gitlab-group-issue` command line option
  - `--help`, 524
  - `-h`, 524
- `gitlab-group-issue-list` command line option
  - `--all`, 525
  - `--assignee-id`, 525
  - `--author-id`, 525
  - `--created-after`, 525
  - `--created-before`, 525
  - `--get-all`, 525
  - `--group-id`, 525
  - `--help`, 524
  - `--iids`, 525
  - `--labels`, 525
  - `--milestone`, 525
  - `--my-reaction-emoji`, 525
  - `--order-by`, 525
  - `--page`, 525
  - `--per-page`, 525
  - `--search`, 525
  - `--sort`, 525
  - `--state`, 525
  - `--sudo`, 525
  - `--updated-after`, 525
  - `--updated-before`, 525
  - `-h`, 524
- `gitlab-group-issues-statistics` command line option
  - `--help`, 525
  - `-h`, 525
- `gitlab-group-issues-statistics-get` command line option
  - `--group-id`, 526
  - `--help`, 526
  - `--sudo`, 526
  - `-h`, 526
- `gitlab-group-iteration` command line option
  - `--help`, 526

-h, 526  
gitlab-group-iteration-list command line option  
--all, 526  
--get-all, 526  
--group-id, 526  
--help, 526  
--include-ancestors, 526  
--page, 526  
--per-page, 526  
--search, 526  
--state, 526  
--sudo, 526  
-h, 526  
gitlab-group-label command line option  
--help, 527  
-h, 527  
gitlab-group-label-create command line option  
--color, 527  
--description, 527  
--group-id, 527  
--help, 527  
--name, 527  
--priority, 527  
--sudo, 527  
-h, 527  
gitlab-group-label-delete command line option  
--group-id, 527  
--help, 527  
--name, 527  
--sudo, 527  
-h, 527  
gitlab-group-label-get command line option  
--group-id, 528  
--help, 528  
--name, 528  
--sudo, 528  
-h, 528  
gitlab-group-label-list command line option  
--all, 528  
--get-all, 528  
--group-id, 528  
--help, 528  
--page, 528  
--per-page, 528  
--sudo, 528  
-h, 528  
gitlab-group-label-subscribe command line option  
--group-id, 528  
--help, 528  
--name, 528  
--sudo, 528  
-h, 528  
gitlab-group-label-unsubscribe command line option  
--group-id, 529  
--help, 529  
--name, 529  
--sudo, 529  
-h, 529  
gitlab-group-label-update command line option  
--color, 529  
--description, 529  
--group-id, 529  
--help, 529  
--name, 529  
--new-name, 529  
--priority, 529  
--sudo, 529  
-h, 529  
gitlab-group-ldap-sync command line option  
--help, 485  
--id, 485  
-h, 485  
gitlab-group-list command line option  
--all, 486  
--all-available, 485  
--get-all, 486  
--help, 485  
--min-access-level, 486  
--order-by, 486  
--owned, 486  
--page, 486  
--per-page, 486  
--search, 486  
--skip-groups, 485  
--sort, 486  
--statistics, 486  
--sudo, 485  
--top-level-only, 486  
--with-custom-attributes, 486  
-h, 485  
gitlab-group-member command line option  
--help, 529  
-h, 529  
gitlab-group-member-all command line option  
--help, 531  
-h, 531  
gitlab-group-member-all-get command line option  
--group-id, 532  
--help, 532  
--id, 532  
--sudo, 532

```

    -h, 532
gitlab-group-member-all-list command line
    option
    --all, 532
    --get-all, 532
    --group-id, 532
    --help, 532
    --page, 532
    --per-page, 532
    --sudo, 532
    -h, 532
gitlab-group-member-create command line
    option
    --access-level, 530
    --expires-at, 530
    --group-id, 530
    --help, 530
    --sudo, 530
    --tasks-to-be-done, 530
    --user-id, 530
    -h, 530
gitlab-group-member-delete command line
    option
    --group-id, 530
    --help, 530
    --id, 530
    --sudo, 530
    -h, 530
gitlab-group-member-get command line option
    --group-id, 530
    --help, 530
    --id, 530
    --sudo, 530
    -h, 530
gitlab-group-member-list command line
    option
    --all, 531
    --get-all, 531
    --group-id, 531
    --help, 531
    --page, 531
    --per-page, 531
    --sudo, 531
    -h, 531
gitlab-group-member-update command line
    option
    --access-level, 531
    --expires-at, 531
    --group-id, 531
    --help, 531
    --id, 531
    --sudo, 531
    -h, 531

gitlab-group-merge-request command line
    option
    --help, 532
    -h, 532
gitlab-group-merge-request-list command
    line option
    --all, 534
    --approved-by-ids, 533
    --approver-ids, 533
    --assignee-id, 533
    --author-id, 533
    --created-after, 533
    --created-before, 533
    --get-all, 534
    --group-id, 533
    --help, 533
    --labels, 533
    --milestone, 533
    --my-reaction-emoji, 533
    --order-by, 533
    --page, 534
    --per-page, 534
    --scope, 533
    --search, 534
    --sort, 533
    --source-branch, 534
    --state, 533
    --sudo, 533
    --target-branch, 534
    --updated-after, 533
    --updated-before, 533
    --view, 533
    --wip, 534
    -h, 533
gitlab-group-milestone command line option
    --help, 534
    -h, 534
gitlab-group-milestone-create command line
    option
    --description, 534
    --due-date, 534
    --group-id, 534
    --help, 534
    --start-date, 534
    --sudo, 534
    --title, 534
    -h, 534
gitlab-group-milestone-delete command line
    option
    --group-id, 535
    --help, 535
    --id, 535
    --sudo, 535
    -h, 535

```

gitlab-group-milestone-get command line option  
--group-id, 535  
--help, 535  
--id, 535  
--sudo, 535  
-h, 535

gitlab-group-milestone-issues command line option  
--group-id, 535  
--help, 535  
--id, 535  
--sudo, 535  
-h, 535

gitlab-group-milestone-list command line option  
--all, 536  
--get-all, 536  
--group-id, 536  
--help, 535  
--iids, 536  
--page, 536  
--per-page, 536  
--search, 536  
--state, 536  
--sudo, 535  
-h, 535

gitlab-group-milestone-merge-requests command line option  
--group-id, 536  
--help, 536  
--id, 536  
--sudo, 536  
-h, 536

gitlab-group-milestone-update command line option  
--description, 536  
--due-date, 536  
--group-id, 536  
--help, 536  
--id, 536  
--start-date, 537  
--state-event, 537  
--sudo, 536  
--title, 536  
-h, 536

gitlab-group-notification-settings command line option  
--help, 537  
-h, 537

gitlab-group-notification-settings-get command line option  
--group-id, 537  
--help, 537  
--sudo, 537  
-h, 537

gitlab-group-notification-settings-update command line option  
--close-issue, 538  
--close-merge-request, 538  
--group-id, 537  
--help, 537  
--level, 538  
--merge-merge-request, 538  
--new-issue, 538  
--new-merge-request, 538  
--new-note, 538  
--notification-email, 538  
--reassign-issue, 538  
--reassign-merge-request, 538  
--reopen-issue, 538  
--reopen-merge-request, 538  
--sudo, 537  
-h, 537

gitlab-group-package command line option  
--help, 538  
-h, 538

gitlab-group-package-list command line option  
--all, 539  
--exclude-subgroups, 538  
--get-all, 539  
--group-id, 538  
--help, 538  
--order-by, 538  
--package-name, 539  
--package-type, 538  
--page, 539  
--per-page, 539  
--sort, 538  
--sudo, 538  
-h, 538

gitlab-group-project command line option  
--help, 539  
-h, 539

gitlab-group-project-list command line option  
--all, 540  
--archived, 539  
--get-all, 540  
--group-id, 539  
--help, 539  
--include-subgroups, 540  
--min-access-level, 540  
--order-by, 539  
--owned, 540  
--page, 540  
--per-page, 540

```

--search, 540
--simple, 540
--sort, 539
--starred, 540
--sudo, 539
--visibility, 539
--with-custom-attributes, 540
--with-issues-enabled, 540
--with-merge-requests-enabled, 540
--with-security-reports, 540
--with-shared, 540
-h, 539
gitlab-group-push-rules command line option
--help, 540
-h, 540
gitlab-group-push-rules-create command line
    option
--author-email-regex, 541
--branch-name-regex, 541
--commit-committer-check, 541
--commit-message-negative-regex, 541
--commit-message-regex, 541
--deny-delete-tag, 541
--file-name-regex, 541
--group-id, 541
--help, 540
--max-file-size, 541
--member-check, 541
--prevent-secrets, 541
--reject-unsigned-commits, 541
--sudo, 541
-h, 540
gitlab-group-push-rules-delete command line
    option
--group-id, 541
--help, 541
--sudo, 541
-h, 541
gitlab-group-push-rules-get command line
    option
--group-id, 541
--help, 541
--sudo, 541
-h, 541
gitlab-group-push-rules-update command line
    option
--author-email-regex, 542
--branch-name-regex, 542
--commit-committer-check, 542
--commit-message-negative-regex, 542
--commit-message-regex, 542
--deny-delete-tag, 542
--file-name-regex, 542
--group-id, 542
--help, 542
--max-file-size, 542
--member-check, 542
--prevent-secrets, 542
--reject-unsigned-commits, 542
--sudo, 542
-h, 542
gitlab-group-runner command line option
--help, 542
-h, 542
gitlab-group-runner-list command line
    option
--all, 543
--get-all, 543
--group-id, 543
--help, 543
--page, 543
--per-page, 543
--scope, 543
--sudo, 543
--tag-list, 543
-h, 543
gitlab-group-search command line option
--help, 486
--id, 486
--scope, 486
--search, 486
-h, 486
gitlab-group-share command line option
--expires-at, 486
--group-access, 486
--group-id, 486
--help, 486
--id, 486
-h, 486
gitlab-group-subgroup command line option
--help, 543
-h, 543
gitlab-group-subgroup-list command line
    option
--all, 544
--all-available, 544
--get-all, 544
--group-id, 543
--help, 543
--min-access-level, 544
--order-by, 544
--owned, 544
--page, 544
--per-page, 544
--search, 544
--skip-groups, 544
--sort, 544
--statistics, 544

```

- `--sudo`, 543
  - `--with-custom-attributes`, 544
  - `-h`, 543
- gitlab-group-transfer command line option
  - `--group-id`, 487
  - `--help`, 487
  - `--id`, 487
  - `-h`, 487
- gitlab-group-transfer-project command line option
  - `--help`, 487
  - `--id`, 487
  - `--project-id`, 487
  - `-h`, 487
- gitlab-group-unshare command line option
  - `--group-id`, 487
  - `--help`, 487
  - `--id`, 487
  - `-h`, 487
- gitlab-group-update command line option
  - `--auto-devops-enabled`, 488
  - `--avatar`, 488
  - `--default-branch-protection`, 488
  - `--description`, 488
  - `--emails-disabled`, 488
  - `--extra-shared-runners-minutes-limit`, 488
  - `--file-template-project-id`, 488
  - `--help`, 488
  - `--id`, 488
  - `--lfs-enabled`, 488
  - `--membership-lock`, 488
  - `--mentions-disabled`, 488
  - `--name`, 488
  - `--path`, 488
  - `--prevent-forking-outside-group`, 488
  - `--project-creation-level`, 488
  - `--request-access-enabled`, 488
  - `--require-two-factor-authentication`, 488
  - `--share-with-group-lock`, 488
  - `--shared-runners-minutes-limit`, 488
  - `--shared-runners-setting`, 488
  - `--subgroup-creation-level`, 488
  - `--sudo`, 488
  - `--two-factor-grace-period`, 488
  - `--visibility`, 488
  - `-h`, 488
- gitlab-group-variable command line option
  - `--help`, 544
  - `-h`, 544
- gitlab-group-variable-create command line option
  - `--group-id`, 544
  - `--help`, 544
  - `--key`, 544
  - `--masked`, 545
  - `--protected`, 544
  - `--sudo`, 544
  - `--value`, 544
  - `--variable-type`, 545
  - `-h`, 544
- gitlab-group-variable-delete command line option
  - `--group-id`, 545
  - `--help`, 545
  - `--key`, 545
  - `--sudo`, 545
  - `-h`, 545
- gitlab-group-variable-get command line option
  - `--group-id`, 545
  - `--help`, 545
  - `--key`, 545
  - `--sudo`, 545
  - `-h`, 545
- gitlab-group-variable-list command line option
  - `--all`, 545
  - `--get-all`, 545
  - `--group-id`, 545
  - `--help`, 545
  - `--page`, 545
  - `--per-page`, 545
  - `--sudo`, 545
  - `-h`, 545
- gitlab-group-variable-update command line option
  - `--group-id`, 546
  - `--help`, 546
  - `--key`, 546
  - `--masked`, 546
  - `--protected`, 546
  - `--sudo`, 546
  - `--value`, 546
  - `--variable-type`, 546
  - `-h`, 546
- gitlab-group-wiki command line option
  - `--help`, 546
  - `-h`, 546
- gitlab-group-wiki-create command line option
  - `--content`, 546
  - `--format`, 546
  - `--group-id`, 546
  - `--help`, 546
  - `--sudo`, 546
  - `--title`, 546
  - `-h`, 546

---

```

gitlab-group-wiki-delete command line
    option
    --group-id, 547
    --help, 547
    --slug, 547
    --sudo, 547
    -h, 547
gitlab-group-wiki-get command line option
    --group-id, 547
    --help, 547
    --slug, 547
    --sudo, 547
    -h, 547
gitlab-group-wiki-list command line option
    --all, 547
    --get-all, 547
    --group-id, 547
    --help, 547
    --page, 547
    --per-page, 547
    --sudo, 547
    --with-content, 547
    -h, 547
gitlab-group-wiki-update command line
    option
    --content, 548
    --format, 548
    --group-id, 548
    --help, 548
    --slug, 548
    --sudo, 548
    --title, 548
    -h, 548
gitlab-hook command line option
    --help, 548
    -h, 548
gitlab-hook-create command line option
    --help, 548
    --sudo, 548
    --url, 548
    -h, 548
gitlab-hook-delete command line option
    --help, 548
    --id, 548
    --sudo, 548
    -h, 548
gitlab-hook-get command line option
    --help, 549
    --id, 549
    --sudo, 549
    -h, 549
gitlab-hook-list command line option
    --all, 549
    --get-all, 549
    --help, 549
    --page, 549
    --per-page, 549
    --sudo, 549
    -h, 549
gitlab-issue command line option
    --help, 549
    -h, 549
gitlab-issue-get command line option
    --help, 549
    --id, 549
    --sudo, 549
    -h, 549
gitlab-issue-list command line option
    --all, 550
    --assignee-id, 550
    --author-id, 550
    --created-after, 550
    --created-before, 550
    --get-all, 550
    --help, 550
    --iids, 550
    --labels, 550
    --milestone, 550
    --my-reaction-emoji, 550
    --order-by, 550
    --page, 550
    --per-page, 550
    --scope, 550
    --search, 550
    --sort, 550
    --state, 550
    --sudo, 550
    --updated-after, 550
    --updated-before, 550
    -h, 550
gitlab-issues-statistics command line
    option
    --help, 551
    -h, 551
gitlab-issues-statistics-get command line
    option
    --help, 551
    --sudo, 551
    -h, 551
gitlab-key command line option
    --help, 551
    -h, 551
gitlab-key-get command line option
    --help, 551
    --id, 551
    --sudo, 551
    -h, 551
gitlab-ldap-group command line option

```

- [--help](#), 551
  - [-h](#), 551
- gitlab-ldap-group-list command line option
  - [--all](#), 552
  - [--get-all](#), 552
  - [--help](#), 552
  - [--page](#), 552
  - [--per-page](#), 552
  - [--provider](#), 552
  - [--search](#), 552
  - [--sudo](#), 552
  - [-h](#), 552
- gitlab-license command line option
  - [--help](#), 552
  - [-h](#), 552
- gitlab-license-get command line option
  - [--fullname](#), 552
  - [--help](#), 552
  - [--key](#), 552
  - [--project](#), 552
  - [--sudo](#), 552
  - [-h](#), 552
- gitlab-license-list command line option
  - [--all](#), 553
  - [--get-all](#), 553
  - [--help](#), 553
  - [--page](#), 553
  - [--per-page](#), 553
  - [--popular](#), 553
  - [--sudo](#), 553
  - [-h](#), 553
- gitlab-merge-request command line option
  - [--help](#), 553
  - [-h](#), 553
- gitlab-merge-request-list command line option
  - [--all](#), 555
  - [--approved-by-ids](#), 554
  - [--approver-ids](#), 554
  - [--assignee-id](#), 554
  - [--author-id](#), 554
  - [--author-username](#), 554
  - [--created-after](#), 554
  - [--created-before](#), 554
  - [--deployed-after](#), 555
  - [--deployed-before](#), 555
  - [--environment](#), 555
  - [--get-all](#), 555
  - [--help](#), 554
  - [--in](#), 555
  - [--labels](#), 554
  - [--milestone](#), 554
  - [--my-reaction-emoji](#), 554
  - [--not](#), 555
  - [--order-by](#), 554
  - [--page](#), 555
  - [--per-page](#), 555
  - [--reviewer-id](#), 554
  - [--reviewer-username](#), 554
  - [--scope](#), 554
  - [--search](#), 554
  - [--sort](#), 554
  - [--source-branch](#), 554
  - [--state](#), 554
  - [--sudo](#), 554
  - [--target-branch](#), 554
  - [--updated-after](#), 554
  - [--updated-before](#), 554
  - [--view](#), 554
  - [--wip](#), 555
  - [--with-labels-details](#), 554
  - [--with-merge-status-recheck](#), 554
  - [-h](#), 554
- gitlab-namespace command line option
  - [--help](#), 555
  - [-h](#), 555
- gitlab-namespace-exists command line option
  - [--help](#), 555
  - [--id](#), 555
  - [--namespace](#), 555
  - [--parent-id](#), 555
  - [-h](#), 555
- gitlab-namespace-get command line option
  - [--help](#), 555
  - [--id](#), 555
  - [--sudo](#), 555
  - [-h](#), 555
- gitlab-namespace-list command line option
  - [--all](#), 556
  - [--get-all](#), 556
  - [--help](#), 556
  - [--page](#), 556
  - [--per-page](#), 556
  - [--search](#), 556
  - [--sudo](#), 556
  - [-h](#), 556
- gitlab-notification-settings command line option
  - [--help](#), 556
  - [-h](#), 556
- gitlab-notification-settings-get command line option
  - [--help](#), 556
  - [--sudo](#), 556
  - [-h](#), 556
- gitlab-notification-settings-update command line option
  - [--close-issue](#), 557



```

--close-merge-request, 557
--help, 557
--level, 557
--merge-merge-request, 557
--new-issue, 557
--new-merge-request, 557
--new-note, 557
--notification-email, 557
--reassign-issue, 557
--reassign-merge-request, 557
--reopen-issue, 557
--reopen-merge-request, 557
--sudo, 557
-h, 557
gitlab-pages-domain command line option
--help, 557
-h, 557
gitlab-pages-domain-list command line
  option
--all, 558
--get-all, 557
--help, 557
--page, 557
--per-page, 557
--sudo, 557
-h, 557
gitlab-personal-access-token command line
  option
--help, 558
-h, 558
gitlab-personal-access-token-delete command
  line option
--help, 558
--id, 558
--sudo, 558
-h, 558
gitlab-personal-access-token-list command
  line option
--all, 558
--get-all, 558
--help, 558
--page, 558
--per-page, 558
--sudo, 558
--user-id, 558
-h, 558
gitlab-project command line option
--help, 559
-h, 559
gitlab-project-access-request command line
  option
--help, 576
-h, 576
gitlab-project-access-request-approve
  command line option
--access-level, 576
--help, 576
--id, 576
--project-id, 576
--sudo, 576
-h, 576
gitlab-project-access-request-create
  command line option
--help, 576
--project-id, 576
--sudo, 576
-h, 576
gitlab-project-access-request-delete
  command line option
--help, 576
--id, 576
--project-id, 576
--sudo, 576
-h, 576
gitlab-project-access-request-list command
  line option
--all, 577
--get-all, 577
--help, 577
--page, 577
--per-page, 577
--project-id, 577
--sudo, 577
-h, 577
gitlab-project-access-token command line
  option
--help, 577
-h, 577
gitlab-project-access-token-create command
  line option
--access-level, 577
--expires-at, 577
--help, 577
--name, 577
--project-id, 577
--scopes, 577
--sudo, 577
-h, 577
gitlab-project-access-token-delete command
  line option
--help, 578
--id, 578
--project-id, 578
--sudo, 578
-h, 578
gitlab-project-access-token-list command
  line option

```

```
--all, 578
--get-all, 578
--help, 578
--page, 578
--per-page, 578
--project-id, 578
--sudo, 578
-h, 578
gitlab-project-additional-statistics
    command line option
    --help, 578
    -h, 578
gitlab-project-additional-statistics-get
    command line option
    --help, 579
    --project-id, 579
    --sudo, 579
    -h, 579
gitlab-project-approval command line option
    --help, 579
    -h, 579
gitlab-project-approval-get command line
    option
    --help, 579
    --project-id, 579
    --sudo, 579
    -h, 579
gitlab-project-approval-rule command line
    option
    --help, 580
    -h, 580
gitlab-project-approval-rule-create command
    line option
    --approvals-required, 580
    --group-ids, 580
    --help, 580
    --name, 580
    --project-id, 580
    --protected-branch-ids, 580
    --sudo, 580
    --user-ids, 580
    -h, 580
gitlab-project-approval-rule-delete command
    line option
    --help, 580
    --id, 580
    --project-id, 580
    --sudo, 580
    -h, 580
gitlab-project-approval-rule-list command
    line option
    --all, 581
    --get-all, 581
    --help, 581
    --page, 581
    --per-page, 581
    --project-id, 581
    --sudo, 581
    -h, 581
gitlab-project-approval-rule-update command
    line option
    --help, 581
    --id, 581
    --project-id, 581
    --sudo, 581
    -h, 581
gitlab-project-approval-update command line
    option
    --approvals-before-merge, 579
    --disable-overriding-approvers-per-merge-request,
        580
    --help, 579
    --merge-requests-author-approval, 580
    --merge-requests-disable-committers-approval,
        580
    --project-id, 579
    --reset-approvals-on-push, 579
    --sudo, 579
    -h, 579
gitlab-project-archive command line option
    --help, 559
    --id, 559
    -h, 559
gitlab-project-artifact command line option
    --artifact-path, 560
    --help, 560, 581
    --id, 560
    --job, 560
    --ref-name, 560
    -h, 560, 581
gitlab-project-artifact-delete command line
    option
    --help, 582
    --project-id, 582
    --ref-name, 582
    --sudo, 582
    -h, 582
gitlab-project-artifact-download command
    line option
    --help, 582
    --job, 582
    --job-token, 582
    --project-id, 582
    --ref-name, 582
    --sudo, 582
    -h, 582
gitlab-project-artifact-raw command line
    option
```

```

--artifact-path, 582
--help, 582
--job, 582
--project-id, 582
--ref-name, 582
--sudo, 582
-h, 582
gitlab-project-artifacts command line
    option
--help, 560
--id, 560
--job, 560
--job-token, 560
--ref-name, 560
-h, 560
gitlab-project-audit-event command line
    option
--help, 583
-h, 583
gitlab-project-audit-event-get command line
    option
--help, 583
--id, 583
--project-id, 583
--sudo, 583
-h, 583
gitlab-project-audit-event-list command
line option
--all, 583
--created-after, 583
--created-before, 583
--get-all, 583
--help, 583
--page, 583
--per-page, 583
--project-id, 583
--sudo, 583
-h, 583
gitlab-project-badge command line option
--help, 584
-h, 584
gitlab-project-badge-create command line
    option
--help, 584
--image-url, 584
--link-url, 584
--project-id, 584
--sudo, 584
-h, 584
gitlab-project-badge-delete command line
    option
--help, 584
--id, 584
--project-id, 584
--sudo, 584
-h, 584
gitlab-project-badge-get command line
    option
--help, 584
--id, 584
--project-id, 584
--sudo, 584
-h, 584
gitlab-project-badge-list command line
    option
--all, 585
--get-all, 585
--help, 585
--page, 585
--per-page, 585
--project-id, 585
--sudo, 585
-h, 585
gitlab-project-badge-render command line
    option
--help, 585
--id, 585
--image-url, 585
--link-url, 585
--project-id, 585
--sudo, 585
-h, 585
gitlab-project-badge-update command line
    option
--help, 585
--id, 585
--image-url, 586
--link-url, 586
--project-id, 585
--sudo, 585
-h, 585
gitlab-project-board command line option
--help, 586
-h, 586
gitlab-project-board-create command line
    option
--help, 586
--name, 586
--project-id, 586
--sudo, 586
-h, 586
gitlab-project-board-delete command line
    option
--help, 586
--id, 586
--project-id, 586
--sudo, 586
-h, 586

```

```
gitlab-project-board-get command line
    option
    --help, 586
    --id, 587
    --project-id, 586
    --sudo, 586
    -h, 586
gitlab-project-board-list command line
    option
    --all, 587
    --get-all, 587
    --help, 587
    --page, 587
    --per-page, 587
    --project-id, 587
    --sudo, 587
    -h, 587
gitlab-project-board-list-create command
    line option
    --board-id, 588
    --help, 588
    --project-id, 588
    --sudo, 588
    -h, 588
gitlab-project-board-list-delete command
    line option
    --board-id, 588
    --help, 588
    --id, 588
    --project-id, 588
    --sudo, 588
    -h, 588
gitlab-project-board-list-get command line
    option
    --board-id, 588
    --help, 588
    --id, 588
    --project-id, 588
    --sudo, 588
    -h, 588
gitlab-project-board-list-list command line
    option
    --all, 589
    --board-id, 589
    --get-all, 589
    --help, 589
    --page, 589
    --per-page, 589
    --project-id, 589
    --sudo, 589
    -h, 589
gitlab-project-board-list-update command
    line option
    --board-id, 589
    --help, 589
    --id, 589
    --position, 589
    --project-id, 589
    --sudo, 589
    -h, 589
gitlab-project-board-update command line
    option
    --help, 587
    --id, 587
    --project-id, 587
    --sudo, 587
    -h, 587
gitlab-project-branch command line option
    --help, 589
    -h, 589
gitlab-project-branch-create command line
    option
    --branch, 590
    --help, 590
    --project-id, 590
    --ref, 590
    --sudo, 590
    -h, 590
gitlab-project-branch-delete command line
    option
    --help, 590
    --name, 590
    --project-id, 590
    --sudo, 590
    -h, 590
gitlab-project-branch-get command line
    option
    --help, 590
    --name, 590
    --project-id, 590
    --sudo, 590
    -h, 590
gitlab-project-branch-list command line
    option
    --all, 591
    --get-all, 591
    --help, 591
    --page, 591
    --per-page, 591
    --project-id, 591
    --sudo, 591
    -h, 591
gitlab-project-ci-lint command line option
    --help, 591
    -h, 591
gitlab-project-ci-lint-create command line
    option
    --content, 591
```

```

--dry-run, 591
--help, 591
--include-jobs, 591
--project-id, 591
--ref, 591
--sudo, 591
-h, 591
gitlab-project-ci-lint-get command line
    option
--help, 592
--project-id, 592
--sudo, 592
-h, 592
gitlab-project-ci-lint-validate command
    line option
--content, 592
--dry-run, 592
--help, 592
--include-jobs, 592
--project-id, 592
--ref, 592
--sudo, 592
-h, 592
gitlab-project-cluster command line option
--help, 592
-h, 592
gitlab-project-cluster-create command line
    option
--domain, 593
--enabled, 593
--environment-scope, 593
--help, 592
--managed, 593
--name, 593
--platform-kubernetes-attributes, 593
--project-id, 593
--sudo, 593
-h, 592
gitlab-project-cluster-delete command line
    option
--help, 593
--id, 593
--project-id, 593
--sudo, 593
-h, 593
gitlab-project-cluster-get command line
    option
--help, 593
--id, 593
--project-id, 593
--sudo, 593
-h, 593
gitlab-project-cluster-list command line
    option
--all, 594
--get-all, 594
--help, 594
--page, 594
--per-page, 594
--project-id, 594
--sudo, 594
-h, 594
gitlab-project-cluster-update command line
    option
--domain, 594
--environment-scope, 594
--help, 594
--id, 594
--management-project-id, 594
--name, 594
--platform-kubernetes-attributes, 594
--project-id, 594
--sudo, 594
-h, 594
gitlab-project-commit command line option
--help, 595
-h, 595
gitlab-project-commit-cherry-pick command
    line option
--branch, 595
--help, 595
--id, 595
--project-id, 595
--sudo, 595
-h, 595
gitlab-project-commit-comment command line
    option
--help, 598
-h, 598
gitlab-project-commit-comment-create
    command line option
--commit-id, 598
--help, 598
--line, 599
--line-type, 599
--note, 599
--path, 599
--project-id, 598
--sudo, 598
-h, 598
gitlab-project-commit-comment-list command
    line option
--all, 599
--commit-id, 599
--get-all, 599
--help, 599
--page, 599
--per-page, 599

```

```
--project-id, 599
--sudo, 599
-h, 599
gitlab-project-commit-create command line
    option
--actions, 595
--author-email, 595
--author-name, 596
--branch, 595
--commit-message, 595
--help, 595
--project-id, 595
--sudo, 595
-h, 595
gitlab-project-commit-diff command line
    option
--help, 596
--id, 596
--project-id, 596
--sudo, 596
-h, 596
gitlab-project-commit-discussion command
    line option
--help, 599
-h, 599
gitlab-project-commit-discussion-create
    command line option
--body, 600
--commit-id, 600
--created-at, 600
--help, 599
--project-id, 600
--sudo, 599
-h, 599
gitlab-project-commit-discussion-get
    command line option
--commit-id, 600
--help, 600
--id, 600
--project-id, 600
--sudo, 600
-h, 600
gitlab-project-commit-discussion-list
    command line option
--all, 600
--commit-id, 600
--get-all, 600
--help, 600
--page, 600
--per-page, 600
--project-id, 600
--sudo, 600
-h, 600
gitlab-project-commit-discussion-note
    command line option
--help, 601
-h, 601
gitlab-project-commit-discussion-note-create
    command line option
--body, 601
--commit-id, 601
--created-at, 601
--discussion-id, 601
--help, 601
--position, 601
--project-id, 601
--sudo, 601
-h, 601
gitlab-project-commit-discussion-note-delete
    command line option
--commit-id, 601
--discussion-id, 602
--help, 601
--id, 602
--project-id, 601
--sudo, 601
-h, 601
gitlab-project-commit-discussion-note-get
    command line option
--commit-id, 602
--discussion-id, 602
--help, 602
--id, 602
--project-id, 602
--sudo, 602
-h, 602
gitlab-project-commit-discussion-note-update
    command line option
--body, 602
--commit-id, 602
--discussion-id, 602
--help, 602
--id, 602
--project-id, 602
--sudo, 602
-h, 602
gitlab-project-commit-get command line
    option
--help, 596
--id, 596
--project-id, 596
--sudo, 596
-h, 596
gitlab-project-commit-list command line
    option
--all, 597
--first-parent, 597
```

```

--get-all, 597
--help, 596
--order, 597
--page, 597
--path, 597
--per-page, 597
--project-id, 596
--ref-name, 596
--since, 596
--sudo, 596
--trailers, 597
--until, 597
--with-stats, 597
-h, 596
gitlab-project-commit-merge-requests
    command line option
    --help, 597
    --id, 597
    --project-id, 597
    --sudo, 597
    -h, 597
gitlab-project-commit-refs command line
    option
    --help, 597
    --id, 597
    --project-id, 597
    --sudo, 597
    --type, 597
    -h, 597
gitlab-project-commit-revert command line
    option
    --branch, 598
    --help, 598
    --id, 598
    --project-id, 598
    --sudo, 598
    -h, 598
gitlab-project-commit-signature command
    line option
    --help, 598
    --id, 598
    --project-id, 598
    --sudo, 598
    -h, 598
gitlab-project-commit-status command line
    option
    --help, 603
    -h, 603
gitlab-project-commit-status-create command
    line option
    --commit-id, 603
    --context, 603
    --coverage, 603
    --description, 603
    --help, 603
    --name, 603
    --project-id, 603
    --ref, 603
    --state, 603
    --sudo, 603
    --target-url, 603
    -h, 603
gitlab-project-commit-status-list command
    line option
    --all, 604
    --commit-id, 604
    --get-all, 604
    --help, 603
    --page, 604
    --per-page, 604
    --project-id, 603
    --sudo, 603
    -h, 603
gitlab-project-create command line option
    --allow-merge-on-skipped-pipeline, 562
    --analytics-access-level, 562
    --approvals-before-merge, 562
    --auto-cancel-pending-pipelines, 562
    --auto-devops-deploy-strategy, 562
    --auto-devops-enabled, 562
    --autoclose-referenced-issues, 562
    --avatar, 562
    --build-coverage-regex, 562
    --build-git-strategy, 562
    --build-timeout, 562
    --builds-access-level, 562
    --ci-config-path, 562
    --container-expiration-policy-attributes,
        562
    --container-registry-enabled, 562
    --default-branch, 562
    --description, 562
    --emails-disabled, 562
    --external-authorization-classification-label,
        562
    --forking-access-level, 562
    --group-with-project-templates-id, 562
    --help, 561
    --import-url, 562
    --initialize-with-readme, 562
    --issues-access-level, 562
    --issues-enabled, 562
    --jobs-enabled, 562
    --lfs-enabled, 562
    --merge-method, 563
    --merge-pipelines-enabled, 563
    --merge-requests-access-level, 563
    --merge-requests-enabled, 563

```

```
--mirror, 563
--mirror-trigger-builds, 563
--name, 562
--namespace-id, 563
--only-allow-merge-if-all-discussions-are-resolved, 563
--only-allow-merge-if-pipeline-succeeds, 563
--operations-access-level, 563
--packages-enabled, 563
--pages-access-level, 563
--path, 562
--printing-merge-request-link-enabled, 563
--public-builds, 563
--remove-source-branch-after-merge, 563
--repository-access-level, 563
--repository-storage, 563
--request-access-enabled, 563
--requirements-access-level, 563
--resolve-outdated-diff-discussions, 563
--shared-runners-enabled, 563
--show-default-award-emojis, 563
--snippets-access-level, 563
--snippets-enabled, 563
--squash-option, 563
--sudo, 562
--tag-list, 563
--template-name, 563
--template-project-id, 563
--topics, 563
--use-custom-template, 563
--visibility, 563
--wiki-access-level, 564
--wiki-enabled, 564
-h, 561
gitlab-project-create-fork-relation command
    line option
    --forked-from-id, 564
    --help, 564
    --id, 564
    -h, 564
gitlab-project-custom-attribute command
    line option
    --help, 604
    -h, 604
gitlab-project-custom-attribute-delete
    command line option
    --help, 604
    --key, 604
    --project-id, 604
    --sudo, 604
    -h, 604
gitlab-project-custom-attribute-get command
    line option
    --help, 604
    --key, 604
    --project-id, 604
    --sudo, 604
    -h, 604
gitlab-project-custom-attribute-list
    command line option
    --all, 605
    --get-all, 605
    --help, 605
    --page, 605
    --per-page, 605
    --project-id, 605
    --sudo, 605
    -h, 605
gitlab-project-delete command line option
    --help, 564
    --id, 564
    --sudo, 564
    -h, 564
gitlab-project-delete-fork-relation command
    line option
    --help, 564
    --id, 564
    -h, 564
gitlab-project-delete-merged-branches
    command line option
    --help, 564
    --id, 564
    -h, 564
gitlab-project-deploy-token command line
    option
    --help, 605
    -h, 605
gitlab-project-deploy-token-create command
    line option
    --expires-at, 605
    --help, 605
    --name, 605
    --project-id, 605
    --scopes, 605
    --sudo, 605
    --username, 605
    -h, 605
gitlab-project-deploy-token-delete command
    line option
    --help, 606
    --id, 606
    --project-id, 606
    --sudo, 606
    -h, 606
```



```

gitlab-project-deploy-token-get command
    line option
    --help, 606
    --id, 606
    --project-id, 606
    --sudo, 606
    -h, 606
gitlab-project-deploy-token-list command
    line option
    --all, 606
    --get-all, 606
    --help, 606
    --page, 606
    --per-page, 606
    --project-id, 606
    --scopes, 606
    --sudo, 606
    -h, 606
gitlab-project-deployment command line
    option
    --help, 607
    -h, 607
gitlab-project-deployment-create command
    line option
    --environment, 607
    --help, 607
    --project-id, 607
    --ref, 607
    --sha, 607
    --status, 607
    --sudo, 607
    --tag, 607
    -h, 607
gitlab-project-deployment-get command line
    option
    --help, 607
    --id, 607
    --project-id, 607
    --sudo, 607
    -h, 607
gitlab-project-deployment-list command line
    option
    --all, 608
    --environment, 608
    --get-all, 608
    --help, 608
    --order-by, 608
    --page, 608
    --per-page, 608
    --project-id, 608
    --sort, 608
    --status, 608
    --sudo, 608
    --updated-after, 608
    --updated-before, 608
    -h, 608
gitlab-project-deployment-merge-request
    command line option
    --help, 609
    -h, 609
gitlab-project-deployment-merge-request-list
    command line option
    --all, 611
    --approved-by-ids, 610
    --approver-ids, 610
    --assignee-id, 610
    --author-id, 610
    --author-username, 610
    --created-after, 610
    --created-before, 610
    --deployed-after, 611
    --deployed-before, 610
    --deployment-id, 610
    --environment, 610
    --get-all, 611
    --help, 609
    --in, 610
    --labels, 610
    --milestone, 610
    --my-reaction-emoji, 610
    --not, 610
    --order-by, 610
    --page, 611
    --per-page, 611
    --project-id, 610
    --reviewer-id, 610
    --reviewer-username, 610
    --scope, 610
    --search, 610
    --sort, 610
    --source-branch, 610
    --state, 610
    --sudo, 609
    --target-branch, 610
    --updated-after, 610
    --updated-before, 610
    --view, 610
    --wip, 610
    --with-labels-details, 610
    --with-merge-status-recheck, 610
    -h, 609
gitlab-project-deployment-update command
    line option
    --help, 608
    --id, 608
    --project-id, 608
    --sudo, 608
    -h, 608

```

gitlab-project-environment command line option  
--help, 611  
-h, 611

gitlab-project-environment-create command line option  
--external-url, 611  
--help, 611  
--name, 611  
--project-id, 611  
--sudo, 611  
-h, 611

gitlab-project-environment-delete command line option  
--help, 612  
--id, 612  
--project-id, 612  
--sudo, 612  
-h, 612

gitlab-project-environment-get command line option  
--help, 612  
--id, 612  
--project-id, 612  
--sudo, 612  
-h, 612

gitlab-project-environment-list command line option  
--all, 613  
--get-all, 612  
--help, 612  
--name, 612  
--page, 612  
--per-page, 612  
--project-id, 612  
--search, 612  
--states, 612  
--sudo, 612  
-h, 612

gitlab-project-environment-stop command line option  
--help, 613  
--id, 613  
--project-id, 613  
--sudo, 613  
-h, 613

gitlab-project-environment-update command line option  
--external-url, 613  
--help, 613  
--id, 613  
--name, 613  
--project-id, 613  
--sudo, 613  
-h, 613

gitlab-project-event command line option  
--help, 613  
-h, 613

gitlab-project-event-list command line option  
--action, 614  
--after, 614  
--all, 614  
--before, 614  
--get-all, 614  
--help, 614  
--page, 614  
--per-page, 614  
--project-id, 614  
--sort, 614  
--sudo, 614  
--target-type, 614  
-h, 614

gitlab-project-export command line option  
--help, 614  
-h, 614

gitlab-project-export-create command line option  
--description, 614  
--help, 614  
--project-id, 614  
--sudo, 614  
-h, 614

gitlab-project-export-download command line option  
--help, 615  
--project-id, 615  
--sudo, 615  
-h, 615

gitlab-project-export-get command line option  
--help, 615  
--project-id, 615  
--sudo, 615  
-h, 615

gitlab-project-file command line option  
--help, 615  
-h, 615

gitlab-project-file-blame command line option  
--file-path, 615  
--help, 615  
--project-id, 615  
--ref, 615  
--sudo, 615  
-h, 615

gitlab-project-file-create command line option

```

--author-email, 616
--author-name, 616
--branch, 616
--commit-message, 616
--content, 616
--encoding, 616
--file-path, 616
--help, 616
--project-id, 616
--sudo, 616
-h, 616
gitlab-project-file-delete command line
    option
--branch, 616
--commit-message, 616
--file-path, 616
--help, 616
--project-id, 616
--sudo, 616
-h, 616
gitlab-project-file-get command line option
--file-path, 617
--help, 617
--project-id, 617
--ref, 617
--sudo, 617
-h, 617
gitlab-project-file-raw command line option
--file-path, 617
--help, 617
--project-id, 617
--ref, 617
--sudo, 617
-h, 617
gitlab-project-file-update command line
    option
--author-email, 618
--author-name, 618
--branch, 617
--commit-message, 617
--content, 617
--encoding, 618
--file-path, 617
--help, 617
--project-id, 617
--sudo, 617
-h, 617
gitlab-project-fork command line option
--help, 618
-h, 618
gitlab-project-fork-create command line
    option
--help, 618
--namespace, 618
--project-id, 618
--sudo, 618
-h, 618
gitlab-project-fork-list command line
    option
--all, 619
--archived, 618
--get-all, 619
--help, 618
--membership, 619
--order-by, 619
--owned, 619
--page, 619
--per-page, 619
--project-id, 618
--search, 619
--simple, 619
--sort, 619
--starred, 619
--statistics, 619
--sudo, 618
--visibility, 619
--with-custom-attributes, 619
--with-issues-enabled, 619
--with-merge-requests-enabled, 619
-h, 618
gitlab-project-get command line option
--help, 565
--id, 565
--sudo, 565
-h, 565
gitlab-project-hook command line option
--help, 619
-h, 619
gitlab-project-hook-create command line
    option
--confidential-issues-events, 620
--enable-ssl-verification, 620
--help, 620
--issues-events, 620
--job-events, 620
--merge-requests-events, 620
--note-events, 620
--pipeline-events, 620
--project-id, 620
--push-events, 620
--sudo, 620
--tag-push-events, 620
--token, 620
--url, 620
--wiki-page-events, 620
-h, 620
gitlab-project-hook-delete command line
    option

```

```
--help, 620
--id, 620
--project-id, 620
--sudo, 620
-h, 620
gitlab-project-hook-get command line option
--help, 621
--id, 621
--project-id, 621
--sudo, 621
-h, 621
gitlab-project-hook-list command line
    option
--all, 621
--get-all, 621
--help, 621
--page, 621
--per-page, 621
--project-id, 621
--sudo, 621
-h, 621
gitlab-project-hook-update command line
    option
--confidential-issues-events, 622
--enable-ssl-verification, 622
--help, 621
--id, 622
--issues-events, 622
--job-events, 622
--merge-requests-events, 622
--note-events, 622
--pipeline-events, 622
--project-id, 622
--push-events, 622
--sudo, 622
--tag-push-events, 622
--token, 622
--url, 622
--wiki-events, 622
-h, 621
gitlab-project-housekeeping command line
    option
--help, 565
--id, 565
-h, 565
gitlab-project-import command line option
--help, 622
-h, 622
gitlab-project-import-get command line
    option
--help, 622
--project-id, 622
--sudo, 622
-h, 622

gitlab-project-integration command line
    option
--help, 623
-h, 623
gitlab-project-integration-available
    command line option
--help, 623
--project-id, 623
--slug, 623
--sudo, 623
-h, 623
gitlab-project-integration-delete command
    line option
--help, 623
--project-id, 623
--slug, 623
--sudo, 623
-h, 623
gitlab-project-integration-get command line
    option
--help, 623
--project-id, 623
--slug, 623
--sudo, 623
-h, 623
gitlab-project-integration-list command
    line option
--all, 624
--get-all, 624
--help, 624
--page, 624
--per-page, 624
--project-id, 624
--sudo, 624
-h, 624
gitlab-project-integration-update command
    line option
--help, 624
--project-id, 624
--slug, 624
--sudo, 624
-h, 624
gitlab-project-invitation command line
    option
--help, 624
-h, 624
gitlab-project-invitation-create command
    line option
--access-level, 625
--expires-at, 625
--help, 625
--invite-source, 625
--project-id, 625
--sudo, 625
```

```

--tasks-project-id, 625
--tasks-to-be-done, 625
-h, 625
gitlab-project-invitation-delete command
    line option
--email, 625
--help, 625
--project-id, 625
--sudo, 625
-h, 625
gitlab-project-invitation-get command line
    option
--email, 625
--help, 625
--project-id, 625
--sudo, 625
-h, 625
gitlab-project-invitation-list command line
    option
--all, 626
--get-all, 626
--help, 626
--page, 626
--per-page, 626
--project-id, 626
--query, 626
--sudo, 626
-h, 626
gitlab-project-invitation-update command
    line option
--access-level, 626
--email, 626
--expires-at, 626
--help, 626
--project-id, 626
--sudo, 626
-h, 626
gitlab-project-issue command line option
--help, 627
-h, 627
gitlab-project-issue-add-spent-time command
    line option
--duration, 627
--help, 627
--iid, 627
--project-id, 627
--sudo, 627
-h, 627
gitlab-project-issue-award-emoji command
    line option
--help, 635
-h, 635
gitlab-project-issue-award-emoji-create
    command line option
--help, 635
--issue-iid, 635
--name, 635
--project-id, 635
--sudo, 635
-h, 635
gitlab-project-issue-award-emoji-delete
    command line option
--help, 635
--id, 635
--issue-iid, 635
--project-id, 635
--sudo, 635
-h, 635
gitlab-project-issue-award-emoji-get
    command line option
--help, 636
--id, 636
--issue-iid, 636
--project-id, 636
--sudo, 636
-h, 636
gitlab-project-issue-award-emoji-list
    command line option
--all, 636
--get-all, 636
--help, 636
--issue-iid, 636
--page, 636
--per-page, 636
--project-id, 636
--sudo, 636
-h, 636
gitlab-project-issue-closed-by command line
    option
--help, 628
--iid, 628
--project-id, 628
--sudo, 628
-h, 628
gitlab-project-issue-create command line
    option
--assignee-id, 628
--assignee-ids, 628
--confidential, 628
--created-at, 628
--description, 628
--discussion-to-resolve, 628
--due-date, 628
--help, 628
--labels, 628
--merge-request-to-resolve-discussions-of,
    628
--milestone-id, 628

```

```
--project-id, 628
--sudo, 628
--title, 628
-h, 628
gitlab-project-issue-delete command line
    option
--help, 629
--iid, 629
--project-id, 629
--sudo, 629
-h, 629
gitlab-project-issue-discussion command
    line option
--help, 636
-h, 636
gitlab-project-issue-discussion-create
    command line option
--body, 637
--created-at, 637
--help, 637
--issue-iid, 637
--project-id, 637
--sudo, 637
-h, 637
gitlab-project-issue-discussion-get command
    line option
--help, 637
--id, 637
--issue-iid, 637
--project-id, 637
--sudo, 637
-h, 637
gitlab-project-issue-discussion-list
    command line option
--all, 638
--get-all, 638
--help, 637
--issue-iid, 637
--page, 637
--per-page, 638
--project-id, 637
--sudo, 637
-h, 637
gitlab-project-issue-discussion-note
    command line option
--help, 638
-h, 638
gitlab-project-issue-discussion-note-create
    command line option
--body, 638
--created-at, 638
--discussion-id, 638
--help, 638
--issue-iid, 638
--project-id, 638
--sudo, 638
-h, 638
gitlab-project-issue-discussion-note-delete
    command line option
--discussion-id, 639
--help, 639
--id, 639
--issue-iid, 639
--project-id, 639
--sudo, 639
-h, 639
gitlab-project-issue-discussion-note-get
    command line option
--discussion-id, 639
--help, 639
--id, 639
--issue-iid, 639
--project-id, 639
--sudo, 639
-h, 639
gitlab-project-issue-discussion-note-update
    command line option
--body, 640
--discussion-id, 640
--help, 639
--id, 640
--issue-iid, 640
--project-id, 640
--sudo, 639
-h, 639
gitlab-project-issue-get command line
    option
--help, 629
--iid, 629
--project-id, 629
--sudo, 629
-h, 629
gitlab-project-issue-link command line
    option
--help, 640
-h, 640
gitlab-project-issue-link-create command
    line option
--help, 640
--issue-iid, 640
--project-id, 640
--sudo, 640
--target-issue-iid, 640
--target-project-id, 640
-h, 640
gitlab-project-issue-link-delete command
    line option
--help, 640
```

```

--issue-iid, 640
--issue-link-id, 640
--project-id, 640
--sudo, 640
-h, 640
gitlab-project-issue-link-list command line
    option
--all, 641
--get-all, 641
--help, 641
--issue-iid, 641
--page, 641
--per-page, 641
--project-id, 641
--sudo, 641
-h, 641
gitlab-project-issue-list command line
    option
--all, 630
--assignee-id, 630
--author-id, 630
--created-after, 630
--created-before, 630
--get-all, 630
--help, 629
--iids, 629
--labels, 630
--milestone, 630
--my-reaction-emoji, 630
--order-by, 630
--page, 630
--per-page, 630
--project-id, 629
--scope, 630
--search, 630
--sort, 630
--state, 629
--sudo, 629
--updated-after, 630
--updated-before, 630
-h, 629
gitlab-project-issue-move command line
    option
--help, 630
--iid, 630
--project-id, 630
--sudo, 630
--to-project-id, 630
-h, 630
gitlab-project-issue-note command line
    option
--help, 641
-h, 641
gitlab-project-issue-note-award-emoji
    command line option
--help, 643
-h, 643
gitlab-project-issue-note-award-emoji-create
    command line option
--help, 643
--issue-iid, 643
--name, 643
--note-id, 643
--project-id, 643
--sudo, 643
-h, 643
gitlab-project-issue-note-award-emoji-delete
    command line option
--help, 644
--id, 644
--issue-iid, 644
--note-id, 644
--project-id, 644
--sudo, 644
-h, 644
gitlab-project-issue-note-award-emoji-get
    command line option
--help, 644
--id, 644
--issue-iid, 644
--note-id, 644
--project-id, 644
--sudo, 644
-h, 644
gitlab-project-issue-note-award-emoji-list
    command line option
--all, 645
--get-all, 645
--help, 644
--issue-iid, 645
--note-id, 645
--page, 645
--per-page, 645
--project-id, 645
--sudo, 644
-h, 644
gitlab-project-issue-note-create command
    line option
--body, 641
--created-at, 641
--help, 641
--issue-iid, 641
--project-id, 641
--sudo, 641
-h, 641
gitlab-project-issue-note-delete command
    line option

```

```
--help, 642
--id, 642
--issue-iid, 642
--project-id, 642
--sudo, 642
-h, 642
gitlab-project-issue-note-get command line
    option
--help, 642
--id, 642
--issue-iid, 642
--project-id, 642
--sudo, 642
-h, 642
gitlab-project-issue-note-list command line
    option
--all, 643
--get-all, 642
--help, 642
--issue-iid, 642
--page, 642
--per-page, 642
--project-id, 642
--sudo, 642
-h, 642
gitlab-project-issue-note-update command
    line option
--body, 643
--help, 643
--id, 643
--issue-iid, 643
--project-id, 643
--sudo, 643
-h, 643
gitlab-project-issue-participants command
    line option
--help, 631
--iid, 631
--project-id, 631
--sudo, 631
-h, 631
gitlab-project-issue-related-merge-requests
    command line option
--help, 631
--iid, 631
--project-id, 631
--sudo, 631
-h, 631
gitlab-project-issue-reorder command line
    option
--help, 631
--iid, 631
--move-after-id, 631
--move-before-id, 631
--project-id, 631
--sudo, 631
-h, 631
gitlab-project-issue-reset-spent-time
    command line option
--help, 632
--iid, 632
--project-id, 632
--sudo, 632
-h, 632
gitlab-project-issue-reset-time-estimate
    command line option
--help, 632
--iid, 632
--project-id, 632
--sudo, 632
-h, 632
gitlab-project-issue-resource-label-event
    command line option
--help, 645
-h, 645
gitlab-project-issue-resource-label-event-get
    command line option
--help, 645
--id, 645
--issue-iid, 645
--project-id, 645
--sudo, 645
-h, 645
gitlab-project-issue-resource-label-event-list
    command line option
--all, 646
--get-all, 646
--help, 645
--issue-iid, 646
--page, 646
--per-page, 646
--project-id, 646
--sudo, 646
-h, 645
gitlab-project-issue-resource-milestone-event
    command line option
--help, 646
-h, 646
gitlab-project-issue-resource-milestone-event-get
    command line option
--help, 646
--id, 646
--issue-iid, 646
--project-id, 646
--sudo, 646
-h, 646
gitlab-project-issue-resource-milestone-event-list
    command line option
```



```

--all, 647
--get-all, 647
--help, 647
--issue-iid, 647
--page, 647
--per-page, 647
--project-id, 647
--sudo, 647
-h, 647
gitlab-project-issue-resource-state-event
    command line option
    --help, 647
    -h, 647
gitlab-project-issue-resource-state-event-get
    command line option
    --help, 647
    --id, 647
    --issue-iid, 647
    --project-id, 647
    --sudo, 647
    -h, 647
gitlab-project-issue-resource-state-event-list
    command line option
    --all, 648
    --get-all, 648
    --help, 648
    --issue-iid, 648
    --page, 648
    --per-page, 648
    --project-id, 648
    --sudo, 648
    -h, 648
gitlab-project-issue-subscribe command line
    option
    --help, 632
    --iid, 632
    --project-id, 632
    --sudo, 632
    -h, 632
gitlab-project-issue-time-estimate command
    line option
    --duration, 633
    --help, 632
    --iid, 633
    --project-id, 632
    --sudo, 632
    -h, 632
gitlab-project-issue-time-stats command
    line option
    --help, 633
    --iid, 633
    --project-id, 633
    --sudo, 633
    -h, 633
gitlab-project-issue-todo command line
    option
    --help, 633
    --iid, 633
    --project-id, 633
    --sudo, 633
    -h, 633
gitlab-project-issue-unsubscribe command
    line option
    --help, 633
    --iid, 633
    --project-id, 633
    --sudo, 633
    -h, 633
gitlab-project-issue-update command line
    option
    --assignee-id, 634
    --assignee-ids, 634
    --confidential, 634
    --description, 634
    --discussion-locked, 634
    --due-date, 634
    --help, 634
    --iid, 634
    --labels, 634
    --milestone-id, 634
    --project-id, 634
    --state-event, 634
    --sudo, 634
    --title, 634
    --updated-at, 634
    -h, 634
gitlab-project-issue-user-agent-detail
    command line option
    --help, 634
    --iid, 635
    --project-id, 634
    --sudo, 634
    -h, 634
gitlab-project-issues-statistics command
    line option
    --help, 648
    -h, 648
gitlab-project-issues-statistics-get
    command line option
    --help, 648
    --project-id, 648
    --sudo, 648
    -h, 648
gitlab-project-job command line option
    --help, 649
    -h, 649
gitlab-project-job-artifact command line
    option

```

```
--help, 649
--id, 649
--project-id, 649
--sudo, 649
-h, 649
gitlab-project-job-artifacts command line
  option
--help, 649
--id, 649
--project-id, 649
--sudo, 649
-h, 649
gitlab-project-job-cancel command line
  option
--help, 650
--id, 650
--project-id, 650
--sudo, 650
-h, 650
gitlab-project-job-delete-artifacts command
  line option
--help, 650
--id, 650
--project-id, 650
--sudo, 650
-h, 650
gitlab-project-job-erase command line
  option
--help, 650
--id, 650
--project-id, 650
--sudo, 650
-h, 650
gitlab-project-job-get command line option
--help, 650
--id, 651
--project-id, 650
--sudo, 650
-h, 650
gitlab-project-job-keep-artifacts command
  line option
--help, 651
--id, 651
--project-id, 651
--sudo, 651
-h, 651
gitlab-project-job-list command line option
--all, 651
--get-all, 651
--help, 651
--page, 651
--per-page, 651
--project-id, 651
--scope, 651
--sudo, 651
-h, 651
gitlab-project-job-play command line option
--help, 651
--id, 652
--project-id, 651
--sudo, 651
-h, 651
gitlab-project-job-retry command line
  option
--help, 652
--id, 652
--project-id, 652
--sudo, 652
-h, 652
gitlab-project-job-trace command line
  option
--help, 652
--id, 652
--project-id, 652
--sudo, 652
-h, 652
gitlab-project-key command line option
--help, 652
-h, 652
gitlab-project-key-create command line
  option
--can-push, 653
--help, 652
--key, 653
--project-id, 652
--sudo, 652
--title, 652
-h, 652
gitlab-project-key-delete command line
  option
--help, 653
--id, 653
--project-id, 653
--sudo, 653
-h, 653
gitlab-project-key-enable command line
  option
--help, 653
--id, 653
--key-id, 653
--project-id, 653
--sudo, 653
-h, 653
gitlab-project-key-get command line option
--help, 653
--id, 653
--project-id, 653
--sudo, 653
```

---

```

    -h, 653
gitlab-project-key-list command line option
    --all, 654
    --get-all, 654
    --help, 654
    --page, 654
    --per-page, 654
    --project-id, 654
    --sudo, 654
    -h, 654
gitlab-project-key-update command line
    option
    --can-push, 654
    --help, 654
    --id, 654
    --project-id, 654
    --sudo, 654
    --title, 654
    -h, 654
gitlab-project-label command line option
    --help, 654
    -h, 654
gitlab-project-label-create command line
    option
    --color, 655
    --description, 655
    --help, 655
    --name, 655
    --priority, 655
    --project-id, 655
    --sudo, 655
    -h, 655
gitlab-project-label-delete command line
    option
    --help, 655
    --name, 655
    --project-id, 655
    --sudo, 655
    -h, 655
gitlab-project-label-get command line
    option
    --help, 655
    --name, 655
    --project-id, 655
    --sudo, 655
    -h, 655
gitlab-project-label-list command line
    option
    --all, 656
    --get-all, 656
    --help, 656
    --page, 656
    --per-page, 656
    --project-id, 656
    --sudo, 656
    -h, 656
gitlab-project-label-subscribe command line
    option
    --help, 656
    --name, 656
    --project-id, 656
    --sudo, 656
    -h, 656
gitlab-project-label-unsubscribe command
    line option
    --help, 656
    --name, 656
    --project-id, 656
    --sudo, 656
    -h, 656
gitlab-project-label-update command line
    option
    --color, 657
    --description, 657
    --help, 657
    --name, 657
    --new-name, 657
    --priority, 657
    --project-id, 657
    --sudo, 657
    -h, 657
gitlab-project-languages command line
    option
    --help, 565
    --id, 565
    -h, 565
gitlab-project-list command line option
    --all, 566
    --archived, 566
    --get-all, 566
    --help, 565
    --id-after, 566
    --id-before, 566
    --last-activity-after, 566
    --last-activity-before, 566
    --membership, 566
    --min-access-level, 566
    --order-by, 566
    --owned, 566
    --page, 566
    --per-page, 566
    --repository-checksum-failed, 566
    --repository-storage, 566
    --search, 566
    --search-namespaces, 566
    --simple, 566
    --sort, 566
    --starred, 566

```

```
--statistics, 566
--sudo, 566
--topic, 566
--visibility, 566
--wiki-checksum-failed, 566
--with-custom-attributes, 566
--with-issues-enabled, 566
--with-merge-requests-enabled, 566
--with-programming-language, 566
-h, 565
gitlab-project-member command line option
--help, 657
-h, 657
gitlab-project-member-all command line
    option
--help, 659
-h, 659
gitlab-project-member-all-get command line
    option
--help, 659
--id, 659
--project-id, 659
--sudo, 659
-h, 659
gitlab-project-member-all-list command line
    option
--all, 660
--get-all, 660
--help, 659
--page, 660
--per-page, 660
--project-id, 659
--sudo, 659
-h, 659
gitlab-project-member-create command line
    option
--access-level, 657
--expires-at, 657
--help, 657
--project-id, 657
--sudo, 657
--tasks-to-be-done, 657
--user-id, 657
-h, 657
gitlab-project-member-delete command line
    option
--help, 658
--id, 658
--project-id, 658
--sudo, 658
-h, 658
gitlab-project-member-get command line
    option
--help, 658
--id, 658
--project-id, 658
--sudo, 658
-h, 658
gitlab-project-member-list command line
    option
--all, 658
--get-all, 658
--help, 658
--page, 658
--per-page, 658
--project-id, 658
--sudo, 658
-h, 658
gitlab-project-member-update command line
    option
--access-level, 659
--expires-at, 659
--help, 659
--id, 659
--project-id, 659
--sudo, 659
-h, 659
gitlab-project-merge-request command line
    option
--help, 660
-h, 660
gitlab-project-merge-request-add-spent-time
    command line option
--duration, 661
--help, 660
--iid, 661
--project-id, 660
--sudo, 661
-h, 660
gitlab-project-merge-request-approval
    command line option
--help, 670
-h, 670
gitlab-project-merge-request-approval-get
    command line option
--help, 670
--mr-iid, 670
--project-id, 670
--sudo, 670
-h, 670
gitlab-project-merge-request-approval-rule
    command line option
--help, 670
-h, 670
gitlab-project-merge-request-approval-rule-create
    command line option
--approval-project-rule-id, 671
--approvals-required, 671
```

```

--group-ids, 671
--help, 671
--id, 671
--merge-request-iid, 671
--mr-iid, 671
--name, 671
--project-id, 671
--sudo, 671
--user-ids, 671
-h, 671
gitlab-project-merge-request-approval-rule-delete
    command line option
--help, 671
--id, 672
--mr-iid, 672
--project-id, 671
--sudo, 671
-h, 671
gitlab-project-merge-request-approval-rule-get
    command line option
--help, 672
--id, 672
--mr-iid, 672
--project-id, 672
--sudo, 672
-h, 672
gitlab-project-merge-request-approval-rule-list
    command line option
--all, 672
--get-all, 672
--help, 672
--mr-iid, 672
--page, 672
--per-page, 672
--project-id, 672
--sudo, 672
-h, 672
gitlab-project-merge-request-approval-rule-update
    command line option
--approval-rule-id, 673
--approvals-required, 673
--group-ids, 673
--help, 673
--id, 673
--merge-request-iid, 673
--mr-iid, 673
--name, 673
--project-id, 673
--sudo, 673
--user-ids, 673
-h, 673
gitlab-project-merge-request-approval-state
    command line option
--help, 673
-h, 673
gitlab-project-merge-request-approval-state-get
    command line option
--help, 674
--id, 674
--mr-iid, 674
--project-id, 674
--sudo, 674
-h, 674
gitlab-project-merge-request-approval-update
    command line option
--approvals-required, 670
--help, 670
--mr-iid, 670
--project-id, 670
--sudo, 670
-h, 670
gitlab-project-merge-request-approve
    command line option
--help, 661
--iid, 661
--project-id, 661
--sha, 661
--sudo, 661
-h, 661
gitlab-project-merge-request-award-emoji
    command line option
--help, 674
-h, 674
gitlab-project-merge-request-award-emoji-create
    command line option
--help, 674
--mr-iid, 674
--name, 674
--project-id, 674
--sudo, 674
-h, 674
gitlab-project-merge-request-award-emoji-delete
    command line option
--help, 675
--id, 675
--mr-iid, 675
--project-id, 675
--sudo, 675
-h, 675
gitlab-project-merge-request-award-emoji-get
    command line option
--help, 675
--id, 675
--mr-iid, 675
--project-id, 675
--sudo, 675
-h, 675

```

```
gitlab-project-merge-request-award-emoji-list
    command line option
    --all, 676
    --get-all, 676
    --help, 675
    --mr-iid, 675
    --page, 675
    --per-page, 675
    --project-id, 675
    --sudo, 675
    -h, 675
gitlab-project-merge-request-cancel-merge-when-pipeline-succeeds
    command line option
    --help, 661
    --iid, 661
    --project-id, 661
    --sudo, 661
    -h, 661
gitlab-project-merge-request-changes
    command line option
    --help, 661
    --iid, 661
    --project-id, 661
    --sudo, 661
    -h, 661
gitlab-project-merge-request-closes-issues
    command line option
    --help, 662
    --iid, 662
    --project-id, 662
    --sudo, 662
    -h, 662
gitlab-project-merge-request-commits
    command line option
    --help, 662
    --iid, 662
    --project-id, 662
    --sudo, 662
    -h, 662
gitlab-project-merge-request-create command
    line option
    --allow-maintainer-to-push, 663
    --assignee-id, 663
    --description, 663
    --help, 662
    --labels, 663
    --milestone-id, 663
    --project-id, 662
    --remove-source-branch, 663
    --reviewer-ids, 663
    --source-branch, 662
    --squash, 663
    --sudo, 662
    --target-branch, 662
    --target-project-id, 663
    --title, 663
    -h, 662
gitlab-project-merge-request-delete command
    line option
    --help, 663
    --iid, 663
    --project-id, 663
    --sudo, 663
    -h, 663
gitlab-project-merge-request-diff command
    line option
    --help, 676
    -h, 676
gitlab-project-merge-request-diff-get
    command line option
    --help, 676
    --id, 676
    --mr-iid, 676
    --project-id, 676
    --sudo, 676
    -h, 676
gitlab-project-merge-request-diff-list
    command line option
    --all, 677
    --get-all, 676
    --help, 676
    --mr-iid, 676
    --page, 676
    --per-page, 676
    --project-id, 676
    --sudo, 676
    -h, 676
gitlab-project-merge-request-discussion
    command line option
    --help, 677
    -h, 677
gitlab-project-merge-request-discussion-create
    command line option
    --body, 677
    --created-at, 677
    --help, 677
    --mr-iid, 677
    --position, 677
    --project-id, 677
    --sudo, 677
    -h, 677
gitlab-project-merge-request-discussion-get
    command line option
    --help, 677
    --id, 678
    --mr-iid, 678
    --project-id, 677
    --sudo, 677
```

```

    -h, 677
gitlab-project-merge-request-discussion-list
    command line option
    --all, 678
    --get-all, 678
    --help, 678
    --mr-iid, 678
    --page, 678
    --per-page, 678
    --project-id, 678
    --sudo, 678
    -h, 678
gitlab-project-merge-request-discussion-note
    command line option
    --help, 679
    -h, 679
gitlab-project-merge-request-discussion-note-create-render-html, 663
    command line option
    --body, 679
    --created-at, 679
    --discussion-id, 679
    --help, 679
    --mr-iid, 679
    --project-id, 679
    --sudo, 679
    -h, 679
gitlab-project-merge-request-discussion-note-delete
    command line option
    --discussion-id, 680
    --help, 679
    --id, 680
    --mr-iid, 680
    --project-id, 679
    --sudo, 679
    -h, 679
gitlab-project-merge-request-discussion-note-get
    command line option
    --discussion-id, 680
    --help, 680
    --id, 680
    --mr-iid, 680
    --project-id, 680
    --sudo, 680
    -h, 680
gitlab-project-merge-request-discussion-note-update
    command line option
    --body, 680
    --discussion-id, 680
    --help, 680
    --id, 680
    --mr-iid, 680
    --project-id, 680
    --sudo, 680
    -h, 680
gitlab-project-merge-request-discussion-update
    command line option
    --help, 678
    --id, 678
    --mr-iid, 678
    --project-id, 678
    --resolved, 678
    --sudo, 678
    -h, 678
gitlab-project-merge-request-get
    command line option
    --help, 663
    --iid, 663
    --include-diverged-commits-count, 664
    --include-rebase-in-progress, 664
    --project-id, 663
    --sudo, 663
    -h, 663
gitlab-project-merge-request-list
    command line option
    --all, 665
    --approved-by-ids, 665
    --approver-ids, 665
    --assignee-id, 665
    --author-id, 665
    --created-after, 664
    --created-before, 664
    --get-all, 665
    --help, 664
    --iids, 665
    --labels, 664
    --milestone, 664
    --my-reaction-emoji, 665
    --order-by, 664
    --page, 665
    --per-page, 665
    --project-id, 664
    --scope, 664
    --search, 665
    --sort, 664
    --source-branch, 665
    --state, 664
    --sudo, 664
    --target-branch, 665
    --updated-after, 664
    --updated-before, 664
    --view, 664
    --wip, 665
    -h, 664
gitlab-project-merge-request-merge
    command line option
    --help, 665
    --iid, 665

```

```
--merge-commit-message, 665
--merge-when-pipeline-succeeds, 665
--project-id, 665
--should-remove-source-branch, 665
--sudo, 665
-h, 665
gitlab-project-merge-request-merge-ref
    command line option
--help, 666
--iid, 666
--project-id, 666
--sudo, 666
-h, 666
gitlab-project-merge-request-note command
    line option
--help, 681
-h, 681
gitlab-project-merge-request-note-award-emoji
    command line option
--help, 683
-h, 683
gitlab-project-merge-request-note-award-emoji-create
    command line option
--help, 683
--mr-iid, 683
--name, 683
--note-id, 683
--project-id, 683
--sudo, 683
-h, 683
gitlab-project-merge-request-note-award-emoji-delete
    command line option
--help, 683
--id, 684
--mr-iid, 684
--note-id, 684
--project-id, 683
--sudo, 683
-h, 683
gitlab-project-merge-request-note-award-emoji-get
    command line option
--help, 684
--id, 684
--mr-iid, 684
--note-id, 684
--project-id, 684
--sudo, 684
-h, 684
gitlab-project-merge-request-note-award-emoji-list
    command line option
--all, 685
--get-all, 684
--help, 684
--mr-iid, 684
--note-id, 684
--project-id, 684
--sudo, 682
-h, 682
gitlab-project-merge-request-note-create
    command line option
--body, 681
--help, 681
--mr-iid, 681
--project-id, 681
--sudo, 681
-h, 681
gitlab-project-merge-request-note-delete
    command line option
--help, 681
--id, 681
--mr-iid, 681
--project-id, 681
--sudo, 681
-h, 681
gitlab-project-merge-request-note-get
    command line option
--help, 682
--id, 682
--mr-iid, 682
--project-id, 682
--sudo, 682
-h, 682
gitlab-project-merge-request-note-list
    command line option
--all, 682
--get-all, 682
--help, 682
--mr-iid, 682
--page, 682
--per-page, 682
--project-id, 682
--sudo, 682
-h, 682
gitlab-project-merge-request-note-update
    command line option
--body, 683
--help, 682
--id, 683
--mr-iid, 683
--project-id, 683
--sudo, 682
-h, 682
gitlab-project-merge-request-participants
    command line option
--help, 666
--iid, 666
```



```

--project-id, 666
--sudo, 666
-h, 666
gitlab-project-merge-request-pipeline
    command line option
--help, 685
-h, 685
gitlab-project-merge-request-pipeline-create
    command line option
--help, 685
--mr-iid, 685
--project-id, 685
--sudo, 685
-h, 685
gitlab-project-merge-request-pipeline-list
    command line option
--all, 685
--get-all, 685
--help, 685
--mr-iid, 685
--page, 685
--per-page, 685
--project-id, 685
--sudo, 685
-h, 685
gitlab-project-merge-request-rebase command
    line option
--help, 666
--iid, 666
--project-id, 666
--sudo, 666
-h, 666
gitlab-project-merge-request-reset-spent-time
    command line option
--help, 666
--iid, 667
--project-id, 666
--sudo, 666
-h, 666
gitlab-project-merge-request-reset-time-estimate
    command line option
--help, 667
--iid, 667
--project-id, 667
--sudo, 667
-h, 667
gitlab-project-merge-request-resource-label-event
    command line option
--help, 686
-h, 686
gitlab-project-merge-request-resource-label-event-list
    command line option
--help, 686
--id, 686
--mr-iid, 686
--project-id, 686
--sudo, 686
-h, 686
gitlab-project-merge-request-resource-milestone-event
    command line option
--help, 687
-h, 687
gitlab-project-merge-request-resource-milestone-event-get
    command line option
--help, 687
--id, 687
--mr-iid, 687
--project-id, 687
--sudo, 687
-h, 687
gitlab-project-merge-request-resource-milestone-event-list
    command line option
--all, 687
--get-all, 687
--help, 687
--mr-iid, 687
--page, 687
--per-page, 687
--project-id, 687
--sudo, 687
-h, 687
gitlab-project-merge-request-resource-state-event
    command line option
--help, 688
-h, 688
gitlab-project-merge-request-resource-state-event-get
    command line option
--help, 688
--id, 688
--mr-iid, 688
--project-id, 688
--sudo, 688
-h, 688
gitlab-project-merge-request-resource-state-event-list
    command line option
--all, 688
--get-all, 688

```

```
--help, 688
--mr-iid, 688
--page, 688
--per-page, 688
--project-id, 688
--sudo, 688
-h, 688
gitlab-project-merge-request-subscribe
    command line option
    --help, 667
    --iid, 667
    --project-id, 667
    --sudo, 667
    -h, 667
gitlab-project-merge-request-time-estimate
    command line option
    --duration, 667
    --help, 667
    --iid, 667
    --project-id, 667
    --sudo, 667
    -h, 667
gitlab-project-merge-request-time-stats
    command line option
    --help, 668
    --iid, 668
    --project-id, 668
    --sudo, 668
    -h, 668
gitlab-project-merge-request-todo command
    line option
    --help, 668
    --iid, 668
    --project-id, 668
    --sudo, 668
    -h, 668
gitlab-project-merge-request-unapprove
    command line option
    --help, 668
    --iid, 668
    --project-id, 668
    --sudo, 668
    -h, 668
gitlab-project-merge-request-unsubscribe
    command line option
    --help, 668
    --iid, 669
    --project-id, 668
    --sudo, 668
    -h, 668
gitlab-project-merge-request-update command
    line option
    --allow-maintainer-to-push, 669
    --assignee-id, 669
    --description, 669
    --discussion-locked, 669
    --help, 669
    --iid, 669
    --labels, 669
    --milestone-id, 669
    --project-id, 669
    --remove-source-branch, 669
    --reviewer-ids, 669
    --squash, 669
    --state-event, 669
    --sudo, 669
    --target-branch, 669
    --title, 669
    -h, 669
gitlab-project-merge-train command line
    option
    --help, 689
    -h, 689
gitlab-project-merge-train-list command
    line option
    --all, 689
    --get-all, 689
    --help, 689
    --page, 689
    --per-page, 689
    --project-id, 689
    --scope, 689
    --sudo, 689
    -h, 689
gitlab-project-milestone command line
    option
    --help, 689
    -h, 689
gitlab-project-milestone-create command
    line option
    --description, 690
    --due-date, 690
    --help, 690
    --project-id, 690
    --start-date, 690
    --state-event, 690
    --sudo, 690
    --title, 690
    -h, 690
gitlab-project-milestone-delete command
    line option
    --help, 690
    --id, 690
    --project-id, 690
    --sudo, 690
    -h, 690
gitlab-project-milestone-get command line
    option
```

```

--help, 690
--id, 690
--project-id, 690
--sudo, 690
-h, 690
gitlab-project-milestone-issues command
  line option
--help, 691
--id, 691
--project-id, 691
--sudo, 691
-h, 691
gitlab-project-milestone-list command line
  option
--all, 691
--get-all, 691
--help, 691
--iids, 691
--page, 691
--per-page, 691
--project-id, 691
--search, 691
--state, 691
--sudo, 691
-h, 691
gitlab-project-milestone-merge-requests
  command line option
--help, 691
--id, 692
--project-id, 691
--sudo, 691
-h, 691
gitlab-project-milestone-update command
  line option
--description, 692
--due-date, 692
--help, 692
--id, 692
--project-id, 692
--start-date, 692
--state-event, 692
--sudo, 692
--title, 692
-h, 692
gitlab-project-mirror-pull command line
  option
--help, 567
--id, 567
-h, 567
gitlab-project-note command line option
--help, 692
-h, 692
gitlab-project-note-get command line option
--help, 692
--id, 692
--project-id, 692
--sudo, 692
-h, 692
gitlab-project-note-list command line
  option
--all, 693
--get-all, 693
--help, 693
--page, 693
--per-page, 693
--project-id, 693
--sudo, 693
-h, 693
gitlab-project-notification-settings
  command line option
--help, 693
-h, 693
gitlab-project-notification-settings-get
  command line option
--help, 693
--project-id, 693
--sudo, 693
-h, 693
gitlab-project-notification-settings-update
  command line option
--close-issue, 694
--close-merge-request, 694
--help, 694
--level, 694
--merge-merge-request, 694
--new-issue, 694
--new-merge-request, 694
--new-note, 694
--notification-email, 694
--project-id, 694
--reassign-issue, 694
--reassign-merge-request, 694
--reopen-issue, 694
--reopen-merge-request, 694
--sudo, 694
-h, 694
gitlab-project-package command line option
--help, 694
-h, 694
gitlab-project-package-delete command line
  option
--help, 695
--id, 695
--project-id, 695
--sudo, 695
-h, 695
gitlab-project-package-file command line
  option

```

```
--help, 696
-h, 696
gitlab-project-package-file-delete command
    line option
    --help, 696
    --id, 696
    --package-id, 696
    --project-id, 696
    --sudo, 696
    -h, 696
gitlab-project-package-file-list command
    line option
    --all, 696
    --get-all, 696
    --help, 696
    --package-id, 696
    --page, 696
    --per-page, 696
    --project-id, 696
    --sudo, 696
    -h, 696
gitlab-project-package-get command line
    option
    --help, 695
    --id, 695
    --project-id, 695
    --sudo, 695
    -h, 695
gitlab-project-package-list command line
    option
    --all, 696
    --get-all, 695
    --help, 695
    --order-by, 695
    --package-name, 695
    --package-type, 695
    --page, 695
    --per-page, 695
    --project-id, 695
    --sort, 695
    --sudo, 695
    -h, 695
gitlab-project-pages-domain command line
    option
    --help, 697
    -h, 697
gitlab-project-pages-domain-create command
    line option
    --certificate, 697
    --domain, 697
    --help, 697
    --key, 697
    --project-id, 697
    --sudo, 697
    -h, 697
gitlab-project-pages-domain-delete command
    line option
    --domain, 697
    --help, 697
    --project-id, 697
    --sudo, 697
    -h, 697
gitlab-project-pages-domain-get command
    line option
    --domain, 698
    --help, 698
    --project-id, 698
    --sudo, 698
    -h, 698
gitlab-project-pages-domain-list command
    line option
    --all, 698
    --get-all, 698
    --help, 698
    --page, 698
    --per-page, 698
    --project-id, 698
    --sudo, 698
    -h, 698
gitlab-project-pages-domain-update command
    line option
    --certificate, 698
    --domain, 698
    --help, 698
    --key, 699
    --project-id, 698
    --sudo, 698
    -h, 698
gitlab-project-pipeline command line option
    --help, 699
    -h, 699
gitlab-project-pipeline-bridge command line
    option
    --help, 701
    -h, 701
gitlab-project-pipeline-bridge-list command
    line option
    --all, 701
    --get-all, 701
    --help, 701
    --page, 701
    --per-page, 701
    --pipeline-id, 701
    --project-id, 701
    --scope, 701
    --sudo, 701
    -h, 701
```

```

gitlab-project-pipeline-cancel command line
    option
    --help, 699
    --id, 699
    --project-id, 699
    --sudo, 699
    -h, 699
gitlab-project-pipeline-create command line
    option
    --help, 699
    --project-id, 699
    --ref, 699
    --sudo, 699
    -h, 699
gitlab-project-pipeline-delete command line
    option
    --help, 699
    --id, 699
    --project-id, 699
    --sudo, 699
    -h, 699
gitlab-project-pipeline-get command line
    option
    --help, 700
    --id, 700
    --project-id, 700
    --sudo, 700
    -h, 700
gitlab-project-pipeline-job command line
    option
    --help, 702
    -h, 702
gitlab-project-pipeline-job-list command
line option
    --all, 702
    --get-all, 702
    --help, 702
    --include-retried, 702
    --page, 702
    --per-page, 702
    --pipeline-id, 702
    --project-id, 702
    --scope, 702
    --sudo, 702
    -h, 702
gitlab-project-pipeline-list command line
    option
    --all, 700
    --get-all, 700
    --help, 700
    --name, 700
    --order-by, 700
    --page, 700
    --per-page, 700
    --project-id, 700
    --ref, 700
    --scope, 700
    --sha, 700
    --sort, 700
    --status, 700
    --sudo, 700
    --username, 700
    --yaml-errors, 700
    -h, 700
gitlab-project-pipeline-retry command line
    option
    --help, 701
    --id, 701
    --project-id, 701
    --sudo, 701
    -h, 701
gitlab-project-pipeline-schedule command
line option
    --help, 702
    -h, 702
gitlab-project-pipeline-schedule-create
command line option
    --active, 703
    --cron, 703
    --cron-timezone, 703
    --description, 703
    --help, 703
    --project-id, 703
    --ref, 703
    --sudo, 703
    -h, 703
gitlab-project-pipeline-schedule-delete
command line option
    --help, 703
    --id, 703
    --project-id, 703
    --sudo, 703
    -h, 703
gitlab-project-pipeline-schedule-get
command line option
    --help, 703
    --id, 703
    --project-id, 703
    --sudo, 703
    -h, 703
gitlab-project-pipeline-schedule-list
command line option
    --all, 704
    --get-all, 704
    --help, 704
    --page, 704
    --per-page, 704
    --project-id, 704

```

```
--sudo, 704
-h, 704
gitlab-project-pipeline-schedule-play
    command line option
    --help, 704
    --id, 704
    --project-id, 704
    --sudo, 704
    -h, 704
gitlab-project-pipeline-schedule-take-ownership
    command line option
    --help, 704
    --id, 704
    --project-id, 704
    --sudo, 704
    -h, 704
gitlab-project-pipeline-schedule-update
    command line option
    --active, 705
    --cron, 705
    --cron-timezone, 705
    --description, 705
    --help, 705
    --id, 705
    --project-id, 705
    --ref, 705
    --sudo, 705
    -h, 705
gitlab-project-pipeline-schedule-variable
    command line option
    --help, 705
    -h, 705
gitlab-project-pipeline-schedule-variable-create
    command line option
    --help, 705
    --key, 706
    --pipeline-schedule-id, 706
    --project-id, 705
    --sudo, 705
    --value, 706
    -h, 705
gitlab-project-pipeline-schedule-variable-delete
    command line option
    --help, 706
    --key, 706
    --pipeline-schedule-id, 706
    --project-id, 706
    --sudo, 706
    -h, 706
gitlab-project-pipeline-schedule-variable-update
    command line option
    --help, 706
    --key, 706
    --pipeline-schedule-id, 706
    --project-id, 706
    --sudo, 706
    -h, 706
gitlab-project-pipeline-test-report
    command line option
    --help, 707
    -h, 707
gitlab-project-pipeline-test-report-get
    command line option
    --help, 707
    --pipeline-id, 707
    --project-id, 707
    --sudo, 707
    -h, 707
gitlab-project-pipeline-test-report-summary
    command line option
    --help, 707
    -h, 707
gitlab-project-pipeline-test-report-summary-get
    command line option
    --help, 707
    --pipeline-id, 707
    --project-id, 707
    --sudo, 707
    -h, 707
gitlab-project-pipeline-variable
    command line option
    --help, 708
    -h, 708
gitlab-project-pipeline-variable-list
    command line option
    --all, 708
    --get-all, 708
    --help, 708
    --page, 708
    --per-page, 708
    --pipeline-id, 708
    --project-id, 708
    --sudo, 708
    -h, 708
gitlab-project-protected-branch
    command line option
    --help, 708
    -h, 708
gitlab-project-protected-branch-create
    command line option
    --allowed-to-merge, 709
    --allowed-to-push, 709
    --allowed-to-unprotect, 709
    --code-owner-approval-required, 709
    --help, 709
    --merge-access-level, 709
    --name, 709
```

```

--project-id, 709
--push-access-level, 709
--sudo, 709
--unprotect-access-level, 709
-h, 709
gitlab-project-protected-branch-delete
    command line option
--help, 709
--name, 709
--project-id, 709
--sudo, 709
-h, 709
gitlab-project-protected-branch-get command
    line option
--help, 709
--name, 709
--project-id, 709
--sudo, 709
-h, 709
gitlab-project-protected-branch-list
    command line option
--all, 710
--get-all, 710
--help, 710
--page, 710
--per-page, 710
--project-id, 710
--sudo, 710
-h, 710
gitlab-project-protected-environment
    command line option
--help, 710
-h, 710
gitlab-project-protected-environment-create
    command line option
--approval-rules, 711
--deploy-access-levels, 710
--help, 710
--name, 710
--project-id, 710
--required-approval-count, 710
--sudo, 710
-h, 710
gitlab-project-protected-environment-delete
    command line option
--help, 711
--name, 711
--project-id, 711
--sudo, 711
-h, 711
gitlab-project-protected-environment-get
    command line option
--help, 711
--name, 711
--project-id, 711
--sudo, 711
-h, 711
gitlab-project-protected-environment-list
    command line option
--all, 712
--get-all, 711
--help, 711
--page, 711
--per-page, 711
--project-id, 711
--sudo, 711
-h, 711
gitlab-project-protected-tag command line
    option
--help, 712
-h, 712
gitlab-project-protected-tag-create command
    line option
--create-access-level, 712
--help, 712
--name, 712
--project-id, 712
--sudo, 712
-h, 712
gitlab-project-protected-tag-delete command
    line option
--help, 712
--name, 712
--project-id, 712
--sudo, 712
-h, 712
gitlab-project-protected-tag-get command
    line option
--help, 713
--name, 713
--project-id, 713
--sudo, 713
-h, 713
gitlab-project-protected-tag-list command
    line option
--all, 713
--get-all, 713
--help, 713
--page, 713
--per-page, 713
--project-id, 713
--sudo, 713
-h, 713
gitlab-project-push-rules command line
    option
--help, 713
-h, 713

```

gitlab-project-push-rules-create command  
line option

- author-email-regex, 714
- branch-name-regex, 714
- commit-committer-check, 714
- commit-message-negative-regex, 714
- commit-message-regex, 714
- deny-delete-tag, 714
- file-name-regex, 714
- help, 714
- max-file-size, 714
- member-check, 714
- prevent-secrets, 714
- project-id, 714
- reject-unsigned-commits, 714
- sudo, 714
- h, 714

gitlab-project-push-rules-delete command  
line option

- help, 714
- project-id, 715
- sudo, 714
- h, 714

gitlab-project-push-rules-get command line  
option

- help, 715
- project-id, 715
- sudo, 715
- h, 715

gitlab-project-push-rules-update command  
line option

- author-email-regex, 715
- branch-name-regex, 715
- commit-committer-check, 715
- commit-message-negative-regex, 715
- commit-message-regex, 715
- deny-delete-tag, 715
- file-name-regex, 715
- help, 715
- max-file-size, 715
- member-check, 716
- prevent-secrets, 716
- project-id, 715
- reject-unsigned-commits, 716
- sudo, 715
- h, 715

gitlab-project-registry-repository command  
line option

- help, 716
- h, 716

gitlab-project-registry-repository-delete  
command line option

- help, 716
- id, 716

- project-id, 716
- sudo, 716
- h, 716

gitlab-project-registry-repository-list  
command line option

- all, 716
- get-all, 716
- help, 716
- page, 716
- per-page, 716
- project-id, 716
- sudo, 716
- h, 716

gitlab-project-registry-tag command line  
option

- help, 717
- h, 717

gitlab-project-registry-tag-delete command  
line option

- help, 717
- name, 717
- project-id, 717
- repository-id, 717
- sudo, 717
- h, 717

gitlab-project-registry-tag-delete-in-bulk  
command line option

- help, 717
- keep-n, 717
- name, 717
- name-regex-delete, 717
- name-regex-keep, 717
- older-than, 717
- project-id, 717
- repository-id, 717
- sudo, 717
- h, 717

gitlab-project-registry-tag-get command  
line option

- help, 718
- name, 718
- project-id, 718
- repository-id, 718
- sudo, 718
- h, 718

gitlab-project-registry-tag-list command  
line option

- all, 718
- get-all, 718
- help, 718
- page, 718
- per-page, 718
- project-id, 718
- repository-id, 718



```

--sudo, 718
-h, 718
gitlab-project-release command line option
--help, 718
-h, 718
gitlab-project-release-create command line
  option
--assets, 719
--description, 719
--help, 719
--name, 719
--project-id, 719
--ref, 719
--sudo, 719
--tag-name, 719
-h, 719
gitlab-project-release-delete command line
  option
--help, 719
--project-id, 719
--sudo, 719
--tag-name, 719
-h, 719
gitlab-project-release-get command line
  option
--help, 719
--project-id, 719
--sudo, 719
--tag-name, 719
-h, 719
gitlab-project-release-link command line
  option
--help, 721
-h, 721
gitlab-project-release-link-create command
  line option
--filepath, 721
--help, 721
--link-type, 721
--name, 721
--project-id, 721
--sudo, 721
--tag-name, 721
--url, 721
-h, 721
gitlab-project-release-link-delete command
  line option
--help, 721
--id, 721
--project-id, 721
--sudo, 721
--tag-name, 721
-h, 721
gitlab-project-release-link-get command
  line option
--help, 722
--id, 722
--project-id, 722
--sudo, 722
--tag-name, 722
-h, 722
gitlab-project-release-link-list command
  line option
--all, 722
--get-all, 722
--help, 722
--page, 722
--per-page, 722
--project-id, 722
--sudo, 722
--tag-name, 722
-h, 722
gitlab-project-release-link-update command
  line option
--filepath, 723
--help, 722
--id, 723
--link-type, 723
--name, 723
--project-id, 723
--sudo, 722
--tag-name, 723
--url, 723
-h, 722
gitlab-project-release-list command line
  option
--all, 720
--get-all, 720
--help, 720
--include-html-description, 720
--order-by, 720
--page, 720
--per-page, 720
--project-id, 720
--sort, 720
--sudo, 720
-h, 720
gitlab-project-release-update command line
  option
--description, 720
--help, 720
--milestones, 720
--name, 720
--project-id, 720
--released-at, 720
--sudo, 720
--tag-name, 720

```

-h, 720  
gitlab-project-remote-mirror command line option  
--help, 723  
-h, 723  
gitlab-project-remote-mirror-create command line option  
--enabled, 723  
--help, 723  
--only-protected-branches, 723  
--project-id, 723  
--sudo, 723  
--url, 723  
-h, 723  
gitlab-project-remote-mirror-list command line option  
--all, 724  
--get-all, 724  
--help, 723  
--page, 724  
--per-page, 724  
--project-id, 724  
--sudo, 723  
-h, 723  
gitlab-project-remote-mirror-update command line option  
--enabled, 724  
--help, 724  
--id, 724  
--only-protected-branches, 724  
--project-id, 724  
--sudo, 724  
-h, 724  
gitlab-project-repository-archive command line option  
--format, 567  
--help, 567  
--id, 567  
--sha, 567  
-h, 567  
gitlab-project-repository-blob command line option  
--help, 567  
--id, 567  
--sha, 567  
-h, 567  
gitlab-project-repository-compare command line option  
--from-, 567  
--help, 567  
--id, 567  
--to, 567  
-h, 567  
gitlab-project-repository-contributors command line option  
--help, 568  
--id, 568  
-h, 568  
gitlab-project-repository-merge-base command line option  
--help, 568  
--id, 568  
--refs, 568  
-h, 568  
gitlab-project-repository-raw-blob command line option  
--help, 568  
--id, 568  
--sha, 568  
-h, 568  
gitlab-project-repository-tree command line option  
--help, 568  
--id, 568  
--path, 568  
--recursive, 568  
--ref, 568  
-h, 568  
gitlab-project-restore command line option  
--help, 569  
--id, 569  
-h, 569  
gitlab-project-runner command line option  
--help, 724  
-h, 724  
gitlab-project-runner-create command line option  
--help, 724  
--project-id, 724  
--runner-id, 724  
--sudo, 724  
-h, 724  
gitlab-project-runner-delete command line option  
--help, 725  
--id, 725  
--project-id, 725  
--sudo, 725  
-h, 725  
gitlab-project-runner-list command line option  
--all, 725  
--get-all, 725  
--help, 725  
--page, 725  
--per-page, 725  
--project-id, 725

---

```

--scope, 725
--sudo, 725
--tag-list, 725
-h, 725
gitlab-project-search command line option
--help, 569
--id, 569
--scope, 569
--search, 569
-h, 569
gitlab-project-service command line option
--help, 725
-h, 725
gitlab-project-service-available command
line option
--help, 726
--project-id, 726
--slug, 726
--sudo, 726
-h, 726
gitlab-project-service-delete command line
option
--help, 726
--project-id, 726
--slug, 726
--sudo, 726
-h, 726
gitlab-project-service-get command line
option
--help, 726
--project-id, 726
--slug, 726
--sudo, 726
-h, 726
gitlab-project-service-list command line
option
--all, 727
--get-all, 727
--help, 726
--page, 727
--per-page, 727
--project-id, 726
--sudo, 726
-h, 726
gitlab-project-service-update command line
option
--help, 727
--project-id, 727
--slug, 727
--sudo, 727
-h, 727
gitlab-project-share command line option
--expires-at, 569
--group-access, 569
--group-id, 569
--help, 569
--id, 569
-h, 569
gitlab-project-snapshot command line option
--help, 569
--id, 569
--wiki, 569
-h, 569
gitlab-project-snippet command line option
--help, 727
-h, 727
gitlab-project-snippet-award-emoji command
line option
--help, 730
-h, 730
gitlab-project-snippet-award-emoji-create
command line option
--help, 730
--name, 730
--project-id, 730
--snippet-id, 730
--sudo, 730
-h, 730
gitlab-project-snippet-award-emoji-delete
command line option
--help, 730
--id, 731
--project-id, 730
--snippet-id, 731
--sudo, 730
-h, 730
gitlab-project-snippet-award-emoji-get
command line option
--help, 731
--id, 731
--project-id, 731
--snippet-id, 731
--sudo, 731
-h, 731
gitlab-project-snippet-award-emoji-list
command line option
--all, 731
--get-all, 731
--help, 731
--page, 731
--per-page, 731
--project-id, 731
--snippet-id, 731
--sudo, 731
-h, 731
gitlab-project-snippet-content command line
option
--help, 727

```

```
--id, 727
--project-id, 727
--sudo, 727
-h, 727
gitlab-project-snippet-create command line
    option
--content, 728
--description, 728
--file-name, 728
--help, 728
--project-id, 728
--sudo, 728
--title, 728
--visibility, 728
-h, 728
gitlab-project-snippet-delete command line
    option
--help, 728
--id, 728
--project-id, 728
--sudo, 728
-h, 728
gitlab-project-snippet-discussion command
    line option
--help, 732
-h, 732
gitlab-project-snippet-discussion-create
    command line option
--body, 732
--created-at, 732
--help, 732
--project-id, 732
--snippet-id, 732
--sudo, 732
-h, 732
gitlab-project-snippet-discussion-get
    command line option
--help, 732
--id, 732
--project-id, 732
--snippet-id, 732
--sudo, 732
-h, 732
gitlab-project-snippet-discussion-list
    command line option
--all, 733
--get-all, 733
--help, 733
--page, 733
--per-page, 733
--project-id, 733
--snippet-id, 733
--sudo, 733
-h, 733

gitlab-project-snippet-discussion-note
    command line option
--help, 733
-h, 733
gitlab-project-snippet-discussion-note-create
    command line option
--body, 734
--created-at, 734
--discussion-id, 734
--help, 733
--project-id, 733
--snippet-id, 733
--sudo, 733
-h, 733
gitlab-project-snippet-discussion-note-delete
    command line option
--discussion-id, 734
--help, 734
--id, 734
--project-id, 734
--snippet-id, 734
--sudo, 734
-h, 734
gitlab-project-snippet-discussion-note-get
    command line option
--discussion-id, 734
--help, 734
--id, 734
--project-id, 734
--snippet-id, 734
--sudo, 734
-h, 734
gitlab-project-snippet-discussion-note-update
    command line option
--body, 735
--discussion-id, 735
--help, 735
--id, 735
--project-id, 735
--snippet-id, 735
--sudo, 735
-h, 735
gitlab-project-snippet-get command line
    option
--help, 728
--id, 728
--project-id, 728
--sudo, 728
-h, 728
gitlab-project-snippet-list command line
    option
--all, 729
--get-all, 729
--help, 729
```

```

--page, 729
--per-page, 729
--project-id, 729
--sudo, 729
-h, 729
gitlab-project-snippet-note command line
    option
--help, 735
-h, 735
gitlab-project-snippet-note-award-emoji
    command line option
--help, 737
-h, 737
gitlab-project-snippet-note-award-emoji-create
    command line option
--help, 737
--name, 738
--note-id, 737
--project-id, 737
--snippet-id, 737
--sudo, 737
-h, 737
gitlab-project-snippet-note-award-emoji-deletegitlab-project-snippet-note-list command
    command line option
--help, 738
--id, 738
--note-id, 738
--project-id, 738
--snippet-id, 738
--sudo, 738
-h, 738
gitlab-project-snippet-note-award-emoji-get
    command line option
--help, 738
--id, 738
--note-id, 738
--project-id, 738
--snippet-id, 738
--sudo, 738
-h, 738
gitlab-project-snippet-note-award-emoji-list
    command line option
--all, 739
--get-all, 739
--help, 739
--note-id, 739
--page, 739
--per-page, 739
--project-id, 739
--snippet-id, 739
--sudo, 739
-h, 739
gitlab-project-snippet-note-create command
    line option
--body, 735
--help, 735
--project-id, 735
--snippet-id, 735
--sudo, 735
-h, 735
gitlab-project-snippet-note-delete command
    line option
--help, 736
--id, 736
--project-id, 736
--snippet-id, 736
--sudo, 736
-h, 736
gitlab-project-snippet-note-get command
    line option
--help, 736
--id, 736
--project-id, 736
--snippet-id, 736
--sudo, 736
-h, 736
gitlab-project-snippet-note-list command
    line option
--all, 737
--get-all, 736
--help, 736
--page, 736
--per-page, 736
--project-id, 736
--snippet-id, 736
--sudo, 736
-h, 736
gitlab-project-snippet-note-update command
    line option
--body, 737
--help, 737
--id, 737
--project-id, 737
--snippet-id, 737
--sudo, 737
-h, 737
gitlab-project-snippet-update command line
    option
--content, 729
--description, 729
--file-name, 729
--help, 729
--id, 729
--project-id, 729
--sudo, 729
--title, 729
--visibility, 729
-h, 729

```

gitlab-project-snippet-user-agent-detail  
    command line option  
    --help, 730  
    --id, 730  
    --project-id, 730  
    --sudo, 730  
    -h, 730

gitlab-project-star command line option  
    --help, 570  
    --id, 570  
    -h, 570

gitlab-project-storage command line option  
    --help, 739  
    -h, 739

gitlab-project-storage-get command line  
    option  
    --help, 739  
    --id, 739  
    --project-id, 739  
    --sudo, 739  
    -h, 739

gitlab-project-tag command line option  
    --help, 740  
    -h, 740

gitlab-project-tag-create command line  
    option  
    --help, 740  
    --message, 740  
    --project-id, 740  
    --ref, 740  
    --sudo, 740  
    --tag-name, 740  
    -h, 740

gitlab-project-tag-delete command line  
    option  
    --help, 740  
    --name, 740  
    --project-id, 740  
    --sudo, 740  
    -h, 740

gitlab-project-tag-get command line option  
    --help, 740  
    --name, 740  
    --project-id, 740  
    --sudo, 740  
    -h, 740

gitlab-project-tag-list command line option  
    --all, 741  
    --get-all, 741  
    --help, 741  
    --page, 741  
    --per-page, 741  
    --project-id, 741  
    --sudo, 741  
    -h, 741

gitlab-project-transfer command line option  
    --help, 570  
    --id, 570  
    --to-namespace, 570  
    -h, 570

gitlab-project-transfer-project command  
    line option  
    --help, 570  
    --id, 570  
    --to-namespace, 570  
    -h, 570

gitlab-project-trigger command line option  
    --help, 741  
    -h, 741

gitlab-project-trigger-create command line  
    option  
    --description, 741  
    --help, 741  
    --project-id, 741  
    --sudo, 741  
    -h, 741

gitlab-project-trigger-delete command line  
    option  
    --help, 742  
    --id, 742  
    --project-id, 742  
    --sudo, 742  
    -h, 742

gitlab-project-trigger-get command line  
    option  
    --help, 742  
    --id, 742  
    --project-id, 742  
    --sudo, 742  
    -h, 742

gitlab-project-trigger-list command line  
    option  
    --all, 742  
    --get-all, 742  
    --help, 742  
    --page, 742  
    --per-page, 742  
    --project-id, 742  
    --sudo, 742  
    -h, 742

gitlab-project-trigger-pipeline command  
    line option  
    --help, 570  
    --id, 570  
    --ref, 570  
    --token, 570  
    -h, 570

```

gitlab-project-trigger-update command line
    option
    --description, 743
    --help, 743
    --id, 743
    --project-id, 743
    --sudo, 743
    -h, 743
gitlab-project-unarchive command line
    option
    --help, 571
    --id, 571
    -h, 571
gitlab-project-unshare command line option
    --group-id, 571
    --help, 571
    --id, 571
    -h, 571
gitlab-project-unstar command line option
    --help, 571
    --id, 571
    -h, 571
gitlab-project-update command line option
    --allow-merge-on-skipped-pipeline, 573
    --analytics-access-level, 573
    --approvals-before-merge, 573
    --auto-cancel-pending-pipelines, 573
    --auto-devops-deploy-strategy, 573
    --auto-devops-enabled, 573
    --autoclose-referenced-issues, 573
    --avatar, 573
    --build-coverage-regex, 573
    --build-git-strategy, 573
    --build-timeout, 573
    --builds-access-level, 573
    --ci-config-path, 573
    --ci-default-git-depth, 573
    --ci-forward-deployment-enabled, 573
    --container-expiration-policy-attributes,
        573
    --container-registry-enabled, 573
    --default-branch, 573
    --description, 573
    --emails-disabled, 573
    --external-authorization-classification-label,
        573
    --forking-access-level, 573
    --help, 573
    --id, 573
    --import-url, 573
    --issues-access-level, 573
    --issues-enabled, 573
    --issues-template, 573
    --jobs-enabled, 574
    --lfs-enabled, 574
    --merge-method, 574
    --merge-pipelines-enabled, 574
    --merge-requests-access-level, 574
    --merge-requests-enabled, 574
    --merge-requests-template, 575
    --mirror, 574
    --mirror-overwrites-diverged-branches,
        574
    --mirror-trigger-builds, 574
    --mirror-user-id, 574
    --name, 574
    --only-allow-merge-if-all-discussions-are-resolved,
        574
    --only-allow-merge-if-pipeline-succeeds,
        574
    --only-mirror-protected-branches, 574
    --operations-access-level, 574
    --packages-enabled, 574
    --pages-access-level, 574
    --path, 574
    --public-builds, 574
    --remove-source-branch-after-merge, 574
    --repository-access-level, 574
    --repository-storage, 574
    --request-access-enabled, 574
    --requirements-access-level, 574
    --resolve-outdated-diff-discussions, 574
    --restrict-user-defined-variables, 574
    --service-desk-enabled, 574
    --shared-runners-enabled, 574
    --show-default-award-emojis, 574
    --snippets-access-level, 574
    --snippets-enabled, 574
    --squash-option, 575
    --sudo, 573
    --suggestion-commit-message, 575
    --tag-list, 575
    --topics, 575
    --visibility, 575
    --wiki-access-level, 575
    --wiki-enabled, 575
    -h, 573
gitlab-project-update-submodule command
    line option
    --branch, 575
    --commit-sha, 575
    --help, 575
    --id, 575
    --submodule, 575
    -h, 575
gitlab-project-upload command line option
    --filename, 575
    --filepath, 575

```

`--help`, 575  
`--id`, 575  
`-h`, 575  
`gitlab-project-user` command line option  
`--help`, 743  
`-h`, 743  
`gitlab-project-user-list` command line option  
`--all`, 743  
`--get-all`, 743  
`--help`, 743  
`--page`, 743  
`--per-page`, 743  
`--project-id`, 743  
`--search`, 743  
`--skip-users`, 743  
`--sudo`, 743  
`-h`, 743  
`gitlab-project-variable` command line option  
`--help`, 744  
`-h`, 744  
`gitlab-project-variable-create` command line option  
`--environment-scope`, 744  
`--help`, 744  
`--key`, 744  
`--masked`, 744  
`--project-id`, 744  
`--protected`, 744  
`--sudo`, 744  
`--value`, 744  
`--variable-type`, 744  
`-h`, 744  
`gitlab-project-variable-delete` command line option  
`--help`, 744  
`--key`, 744  
`--project-id`, 744  
`--sudo`, 744  
`-h`, 744  
`gitlab-project-variable-get` command line option  
`--help`, 745  
`--key`, 745  
`--project-id`, 745  
`--sudo`, 745  
`-h`, 745  
`gitlab-project-variable-list` command line option  
`--all`, 745  
`--get-all`, 745  
`--help`, 745  
`--page`, 745  
`--per-page`, 745  
`--project-id`, 745  
`--sudo`, 745  
`-h`, 745  
`gitlab-project-variable-update` command line option  
`--environment-scope`, 746  
`--help`, 745  
`--key`, 745  
`--masked`, 746  
`--project-id`, 745  
`--protected`, 746  
`--sudo`, 745  
`--value`, 746  
`--variable-type`, 746  
`-h`, 745  
`gitlab-project-wiki` command line option  
`--help`, 746  
`-h`, 746  
`gitlab-project-wiki-create` command line option  
`--content`, 746  
`--format`, 746  
`--help`, 746  
`--project-id`, 746  
`--sudo`, 746  
`--title`, 746  
`-h`, 746  
`gitlab-project-wiki-delete` command line option  
`--help`, 746  
`--project-id`, 746  
`--slug`, 746  
`--sudo`, 746  
`-h`, 746  
`gitlab-project-wiki-get` command line option  
`--help`, 747  
`--project-id`, 747  
`--slug`, 747  
`--sudo`, 747  
`-h`, 747  
`gitlab-project-wiki-list` command line option  
`--all`, 747  
`--get-all`, 747  
`--help`, 747  
`--page`, 747  
`--per-page`, 747  
`--project-id`, 747  
`--sudo`, 747  
`--with-content`, 747  
`-h`, 747  
`gitlab-project-wiki-update` command line option  
`--content`, 748



---

```

--format, 748
--help, 747
--project-id, 747
--slug, 747
--sudo, 747
--title, 747
-h, 747
gitlab-registry-repository command line
    option
--help, 748
-h, 748
gitlab-registry-repository-get command line
    option
--help, 748
--id, 748
--sudo, 748
-h, 748
gitlab-runner command line option
--help, 748
-h, 748
gitlab-runner-all command line option
--help, 748, 751
--id, 748
--scope, 748
-h, 748, 751
gitlab-runner-all-list command line option
--all, 751
--get-all, 751
--help, 751
--page, 751
--paused, 751
--per-page, 751
--scope, 751
--status, 751
--sudo, 751
--tag-list, 751
--type, 751
-h, 751
gitlab-runner-create command line option
--access-level, 749
--active, 749
--description, 749
--help, 749
--info, 749
--locked, 749
--maximum-timeout, 749
--run-untagged, 749
--sudo, 749
--tag-list, 749
--token, 749
-h, 749
gitlab-runner-delete command line option
--help, 749
--id, 749
--sudo, 749
-h, 749
gitlab-runner-get command line option
--help, 749
--id, 749
--sudo, 749
-h, 749
gitlab-runner-job command line option
--help, 752
-h, 752
gitlab-runner-job-list command line option
--all, 752
--get-all, 752
--help, 752
--page, 752
--per-page, 752
--runner-id, 752
--status, 752
--sudo, 752
-h, 752
gitlab-runner-list command line option
--all, 750
--get-all, 750
--help, 750
--page, 750
--paused, 750
--per-page, 750
--scope, 750
--status, 750
--sudo, 750
--tag-list, 750
--type, 750
-h, 750
gitlab-runner-update command line option
--access-level, 751
--active, 750
--description, 750
--help, 750
--id, 750
--locked, 750
--maximum-timeout, 751
--run-untagged, 750
--sudo, 750
--tag-list, 750
-h, 750
gitlab-runner-verify command line option
--help, 751
--id, 751
--token, 751
-h, 751
gitlab-shared-project command line option
--help, 752
-h, 752

```

gitlab-shared-project-list command line option

- all, [753](#)
- archived, [753](#)
- get-all, [753](#)
- group-id, [753](#)
- help, [753](#)
- min-access-level, [753](#)
- order-by, [753](#)
- page, [753](#)
- per-page, [753](#)
- search, [753](#)
- simple, [753](#)
- sort, [753](#)
- starred, [753](#)
- sudo, [753](#)
- visibility, [753](#)
- with-custom-attributes, [753](#)
- with-issues-enabled, [753](#)
- with-merge-requests-enabled, [753](#)
- h, [753](#)

gitlab-snippet command line option

- help, [753](#)
- h, [753](#)

gitlab-snippet-content command line option

- help, [754](#)
- id, [754](#)
- h, [754](#)

gitlab-snippet-create command line option

- content, [754](#)
- file-name, [754](#)
- help, [754](#)
- lifetime, [754](#)
- sudo, [754](#)
- title, [754](#)
- visibility, [754](#)
- h, [754](#)

gitlab-snippet-delete command line option

- help, [754](#)
- id, [754](#)
- sudo, [754](#)
- h, [754](#)

gitlab-snippet-get command line option

- help, [755](#)
- id, [755](#)
- sudo, [755](#)
- h, [755](#)

gitlab-snippet-list command line option

- all, [755](#)
- get-all, [755](#)
- help, [755](#)
- page, [755](#)
- per-page, [755](#)
- sudo, [755](#)

- h, [755](#)

gitlab-snippet-public command line option

- help, [755](#)
- id, [755](#)
- h, [755](#)

gitlab-snippet-update command line option

- content, [756](#)
- file-name, [756](#)
- help, [755](#)
- id, [755](#)
- sudo, [755](#)
- title, [755](#)
- visibility, [756](#)
- h, [755](#)

gitlab-snippet-user-agent-detail command line option

- help, [756](#)
- id, [756](#)
- h, [756](#)

gitlab-starred-project command line option

- help, [756](#)
- h, [756](#)

gitlab-starred-project-list command line option

- all, [757](#)
- archived, [756](#)
- get-all, [757](#)
- help, [756](#)
- membership, [756](#)
- min-access-level, [757](#)
- order-by, [757](#)
- owned, [757](#)
- page, [757](#)
- per-page, [757](#)
- search, [757](#)
- simple, [757](#)
- sort, [757](#)
- starred, [757](#)
- statistics, [757](#)
- sudo, [756](#)
- user-id, [756](#)
- visibility, [757](#)
- with-custom-attributes, [757](#)
- with-issues-enabled, [757](#)
- with-merge-requests-enabled, [757](#)
- h, [756](#)

gitlab-todo command line option

- help, [757](#)
- h, [757](#)

gitlab-todo-delete command line option

- help, [757](#)
- id, [757](#)
- sudo, [757](#)
- h, [757](#)

---

```

gitlab-todo-list command line option
  --action, 758
  --all, 758
  --author-id, 758
  --get-all, 758
  --help, 758
  --page, 758
  --per-page, 758
  --project-id, 758
  --state, 758
  --sudo, 758
  --type, 758
  -h, 758
gitlab-todo-mark-all-as-done command line
  option
  --help, 758
  --id, 758
  -h, 758
gitlab-todo-mark-as-done command line
  option
  --help, 758
  --id, 758
  -h, 758
gitlab-topic command line option
  --help, 759
  -h, 759
gitlab-topic-create command line option
  --avatar, 759
  --description, 759
  --help, 759
  --name, 759
  --sudo, 759
  --title, 759
  -h, 759
gitlab-topic-delete command line option
  --help, 759
  --id, 759
  --sudo, 759
  -h, 759
gitlab-topic-get command line option
  --help, 759
  --id, 759
  --sudo, 759
  -h, 759
gitlab-topic-list command line option
  --all, 760
  --get-all, 760
  --help, 760
  --page, 760
  --per-page, 760
  --sudo, 760
  -h, 760
gitlab-topic-update command line option
  --avatar, 760
  --description, 760
  --help, 760
  --id, 760
  --name, 760
  --sudo, 760
  -h, 760
gitlab-user command line option
  --help, 761
  -h, 761
gitlab-user-activate command line option
  --help, 761
  --id, 761
  -h, 761
gitlab-user-activities command line option
  --help, 767
  -h, 767
gitlab-user-activities-list command line
  option
  --all, 767
  --get-all, 767
  --help, 767
  --page, 767
  --per-page, 767
  --sudo, 767
  -h, 767
gitlab-user-approve command line option
  --help, 761
  --id, 761
  -h, 761
gitlab-user-ban command line option
  --help, 761
  --id, 761
  -h, 761
gitlab-user-block command line option
  --help, 761
  --id, 761
  -h, 761
gitlab-user-create command line option
  --admin, 762
  --avatar, 763
  --bio, 762
  --can-create-group, 762
  --color-scheme-id, 763
  --email, 762
  --extern-uid, 762
  --external, 762
  --help, 762
  --linkedin, 762
  --location, 763
  --name, 762
  --organization, 763
  --password, 762
  --private-profile, 763
  --projects-limit, 762

```

```
--provider, 762
--public-email, 763
--reset-password, 762
--skip-confirmation, 762
--skype, 762
--sudo, 762
--theme-id, 763
--twitter, 762
--username, 762
--website-url, 762
-h, 762
gitlab-user-custom-attribute command line
    option
--help, 767
-h, 767
gitlab-user-custom-attribute-delete command
    line option
--help, 767
--key, 767
--sudo, 767
--user-id, 767
-h, 767
gitlab-user-custom-attribute-get command
    line option
--help, 768
--key, 768
--sudo, 768
--user-id, 768
-h, 768
gitlab-user-custom-attribute-list command
    line option
--all, 768
--get-all, 768
--help, 768
--page, 768
--per-page, 768
--sudo, 768
--user-id, 768
-h, 768
gitlab-user-deactivate command line option
--help, 763
--id, 763
-h, 763
gitlab-user-delete command line option
--help, 763
--id, 763
--sudo, 763
-h, 763
gitlab-user-email command line option
--help, 768
-h, 768
gitlab-user-email-create command line
    option
--email, 769
--help, 769
--sudo, 769
--user-id, 769
-h, 769
gitlab-user-email-delete command line
    option
--help, 769
--id, 769
--sudo, 769
--user-id, 769
-h, 769
gitlab-user-email-get command line option
--help, 769
--id, 769
--sudo, 769
--user-id, 769
-h, 769
gitlab-user-email-list command line option
--all, 770
--get-all, 770
--help, 769
--page, 769
--per-page, 770
--sudo, 769
--user-id, 769
-h, 769
gitlab-user-event command line option
--help, 770
-h, 770
gitlab-user-event-list command line option
--action, 770
--after, 770
--all, 770
--before, 770
--get-all, 770
--help, 770
--page, 770
--per-page, 770
--sort, 770
--sudo, 770
--target-type, 770
--user-id, 770
-h, 770
gitlab-user-follow command line option
--help, 763
--id, 763
-h, 763
gitlab-user-get command line option
--help, 764
--id, 764
--sudo, 764
-h, 764
gitlab-user-gpg-key command line option
--help, 771
```

```

    -h, 771
gitlab-user-gpg-key-create command line
    option
    --help, 771
    --key, 771
    --sudo, 771
    --user-id, 771
    -h, 771
gitlab-user-gpg-key-delete command line
    option
    --help, 771
    --id, 771
    --sudo, 771
    --user-id, 771
    -h, 771
gitlab-user-gpg-key-get command line option
    --help, 771
    --id, 771
    --sudo, 771
    --user-id, 771
    -h, 771
gitlab-user-gpg-key-list command line
    option
    --all, 772
    --get-all, 772
    --help, 772
    --page, 772
    --per-page, 772
    --sudo, 772
    --user-id, 772
    -h, 772
gitlab-user-impersonation-token command
    line option
    --help, 772
    -h, 772
gitlab-user-impersonation-token-create
    command line option
    --expires-at, 772
    --help, 772
    --name, 772
    --scopes, 772
    --sudo, 772
    --user-id, 772
    -h, 772
gitlab-user-impersonation-token-delete
    command line option
    --help, 773
    --id, 773
    --sudo, 773
    --user-id, 773
    -h, 773
gitlab-user-impersonation-token-get command
    line option
    --help, 773
    --id, 773
    --sudo, 773
    --user-id, 773
    -h, 773
gitlab-user-key command line option
    --help, 774
    -h, 774
gitlab-user-key-create command line option
    --help, 774
    --key, 774
    --sudo, 774
    --title, 774
    --user-id, 774
    -h, 774
gitlab-user-key-delete command line option
    --help, 774
    --id, 774
    --sudo, 774
    --user-id, 774
    -h, 774
gitlab-user-key-get command line option
    --help, 774
    --id, 774
    --sudo, 774
    --user-id, 774
    -h, 774
gitlab-user-key-list command line option
    --all, 775
    --get-all, 775
    --help, 775
    --page, 775
    --per-page, 775
    --sudo, 775
    --user-id, 775
    -h, 775
gitlab-user-list command line option
    --active, 764
    --all, 764
    --blocked, 764
    --custom-attributes, 764
    --extern-uid, 764
    --external, 764
    --get-all, 764

```

```
--help, 764
--page, 764
--per-page, 764
--provider, 764
--search, 764
--status, 764
--sudo, 764
--two-factor, 764
--username, 764
-h, 764
gitlab-user-membership command line option
--help, 775
-h, 775
gitlab-user-membership-get command line
    option
--help, 775
--source-id, 775
--sudo, 775
--user-id, 775
-h, 775
gitlab-user-membership-list command line
    option
--all, 776
--get-all, 776
--help, 775
--page, 776
--per-page, 776
--sudo, 775
--type, 776
--user-id, 776
-h, 775
gitlab-user-personal-access-token command
    line option
--help, 776
-h, 776
gitlab-user-personal-access-token-create
    command line option
--expires-at, 776
--help, 776
--name, 776
--scopes, 776
--sudo, 776
--user-id, 776
-h, 776
gitlab-user-project command line option
--help, 776
-h, 776
gitlab-user-project-create command line
    option
--builds-enabled, 777
--default-branch, 777
--description, 777
--help, 777
--import-url, 777
--issues-enabled, 777
--merge-requests-enabled, 777
--name, 777
--only-allow-merge-if-build-succeeds, 777
--public, 777
--public-builds, 777
--snippets-enabled, 777
--squash-option, 777
--sudo, 777
--user-id, 777
--visibility, 777
--wall-enabled, 777
--wiki-enabled, 777
-h, 777
gitlab-user-project-list command line
    option
--all, 779
--archived, 778
--get-all, 779
--help, 778
--id-after, 779
--id-before, 779
--membership, 778
--min-access-level, 779
--order-by, 778
--owned, 778
--page, 779
--per-page, 779
--repository-checksum-failed, 778
--search, 778
--simple, 778
--sort, 778
--starred, 778
--statistics, 778
--sudo, 778
--user-id, 778
--visibility, 778
--wiki-checksum-failed, 778
--with-custom-attributes, 778
--with-issues-enabled, 778
--with-merge-requests-enabled, 778
--with-programming-language, 778
-h, 778
gitlab-user-reject command line option
--help, 765
--id, 765
-h, 765
gitlab-user-status command line option
--help, 779
-h, 779
gitlab-user-status-get command line option
--help, 779
--sudo, 779
--user-id, 779
```

- h, 779
- gitlab-user-unban command line option
  - help, 765
  - id, 765
  - h, 765
- gitlab-user-unblock command line option
  - help, 765
  - id, 765
  - h, 765
- gitlab-user-unfollow command line option
  - help, 765
  - id, 765
  - h, 765
- gitlab-user-update command line option
  - admin, 766
  - avatar, 766
  - bio, 766
  - can-create-group, 766
  - color-scheme-id, 766
  - email, 766
  - extern-uid, 766
  - external, 766
  - help, 766
  - id, 766
  - linkedin, 766
  - location, 766
  - name, 766
  - organization, 766
  - password, 766
  - private-profile, 766
  - projects-limit, 766
  - provider, 766
  - public-email, 766
  - skip-reconfirmation, 766
  - skype, 766
  - sudo, 766
  - theme-id, 766
  - twitter, 766
  - username, 766
  - website-url, 766
  - h, 766
- gitlab-variable command line option
  - help, 779
  - h, 779
- gitlab-variable-create command line option
  - help, 780
  - key, 780
  - masked, 780
  - protected, 780
  - sudo, 780
  - value, 780
  - variable-type, 780
  - h, 780
- gitlab-variable-delete command line option
  - help, 780
  - key, 780
  - sudo, 780
  - h, 780
- gitlab-variable-get command line option
  - help, 780
  - key, 780
  - sudo, 780
  - h, 780
- gitlab-variable-list command line option
  - all, 781
  - get-all, 781
  - help, 781
  - page, 781
  - per-page, 781
  - sudo, 781
  - h, 781
- gitlab-variable-update command line option
  - help, 781
  - key, 781
  - masked, 781
  - protected, 781
  - sudo, 781
  - value, 781
  - variable-type, 781
  - h, 781
- GitlabActivateError, 427
- GitlabAttachFileError, 427
- GitlabAuthenticationError, 427
- GitlabBanError, 428
- GitlabBlockError, 428
- GitlabBuildCancelError, 428
- GitlabBuildEraseError, 428
- GitlabBuildPlayError, 429
- GitlabBuildRetryError, 429
- GitlabCancelError, 429
- GitlabCherryPickError, 429
- GitlabCiLintError, 430
- Gitlabciyaml (*class in gitlab.v4.objects*), 202
- GitlabciyamlManager (*class in gitlab.v4.objects*), 202
- gitlabciyaml (*gitlab.Gitlab attribute*), 174
- GitlabConfigHelperError, 425
- GitlabConfigMissingError, 425
- GitlabConfigParser (*class in gitlab.config*), 425
- GitlabConnectionError, 430
- GitlabCreateError, 430
- GitlabDataError, 426
- GitlabDeactivateError, 430
- GitlabDeleteError, 431
- GitlabError, 431
- GitlabFollowError, 431
- GitlabGetError, 431
- GitlabGroupTransferError, 432
- GitlabHeadError, 432

GitlabHousekeepingError, 432  
GitlabHttpError, 432  
GitlabIDError, 426  
GitlabImportError, 433  
GitlabInvitationError, 433  
GitlabJobCancelError, 433  
GitlabJobEraseError, 433  
GitlabJobPlayError, 434  
GitlabJobRetryError, 434  
GitlabLicenseError, 434  
GitlabList (class in gitlab), 181  
GitlabListError, 434  
GitlabMarkdownError, 436  
GitlabMRApprovalError, 435  
GitlabMRClosedError, 435  
GitlabMRForbiddenError, 435  
GitlabMROnBuildSuccessError, 435  
GitlabMRRebaseError, 436  
GitlabOperationError, 436  
GitlabOwnershipError, 436  
GitlabParsingError, 437  
GitlabPipelineCancelError, 437  
GitlabPipelinePlayError, 437  
GitlabPipelineRetryError, 437  
GitlabProjectDeployKeyError, 438  
GitlabPromoteError, 438  
GitlabProtectError, 438  
GitlabRenderError, 438  
GitlabRepairError, 439  
GitlabRestoreError, 439  
GitlabRetryError, 439  
GitlabRevertError, 439  
GitlabSearchError, 440  
GitlabSetError, 440  
GitlabStopError, 440  
GitlabSubscribeError, 440  
GitlabTimeTrackingError, 441  
GitlabTodoError, 441  
GitlabTransferProjectError, 441  
GitlabUnbanError, 441  
GitlabUnblockError, 442  
GitlabUnfollowError, 442  
GitlabUnsubscribeError, 442  
GitlabUpdateError, 442  
GitlabUploadError, 443  
GitlabUserApproveError, 443  
GitlabUserRejectError, 443  
GitlabVerifyError, 443  
GLOBAL (gitlab.const.NotificationLevel attribute), 426  
GLOBAL\_SNIPPET\_TITLES (gitlab.const.SearchScope attribute), 426  
gpgkeys (gitlab.v4.objects.CurrentUser attribute), 190  
gpgkeys (gitlab.v4.objects.User attribute), 407  
Group (class in gitlab.v4.objects), 203  
GroupAccessRequest (class in gitlab.v4.objects), 206  
GroupAccessRequestManager (class in gitlab.v4.objects), 207  
GroupAccessToken (class in gitlab.v4.objects), 207  
GroupAccessTokenManager (class in gitlab.v4.objects), 207  
GroupAuditEvent (class in gitlab.v4.objects), 208  
GroupAuditEventManager (class in gitlab.v4.objects), 208  
GroupBadge (class in gitlab.v4.objects), 208  
GroupBadgeManager (class in gitlab.v4.objects), 209  
GroupBillableMember (class in gitlab.v4.objects), 210  
GroupBillableMemberManager (class in gitlab.v4.objects), 210  
GroupBillableMemberMembership (class in gitlab.v4.objects), 210  
GroupBillableMemberMembershipManager (class in gitlab.v4.objects), 210  
GroupBoard (class in gitlab.v4.objects), 211  
GroupBoardList (class in gitlab.v4.objects), 211  
GroupBoardListManager (class in gitlab.v4.objects), 211  
GroupBoardManager (class in gitlab.v4.objects), 212  
GroupCluster (class in gitlab.v4.objects), 213  
GroupClusterManager (class in gitlab.v4.objects), 213  
GroupCustomAttribute (class in gitlab.v4.objects), 214  
GroupCustomAttributeManager (class in gitlab.v4.objects), 214  
GroupDeployToken (class in gitlab.v4.objects), 215  
GroupDeployTokenManager (class in gitlab.v4.objects), 215  
GroupDescendantGroup (class in gitlab.v4.objects), 216  
GroupDescendantGroupManager (class in gitlab.v4.objects), 216  
GroupEpic (class in gitlab.v4.objects), 217  
GroupEpicAwardEmoji (class in gitlab.v4.objects), 217  
GroupEpicAwardEmojiManager (class in gitlab.v4.objects), 217  
GroupEpicDiscussionNote (class in gitlab.v4.objects), 218  
GroupEpicDiscussionNoteManager (class in gitlab.v4.objects), 218  
GroupEpicIssue (class in gitlab.v4.objects), 219  
GroupEpicIssueManager (class in gitlab.v4.objects), 219  
GroupEpicManager (class in gitlab.v4.objects), 220  
GroupEpicNote (class in gitlab.v4.objects), 221  
GroupEpicNoteAwardEmoji (class in gitlab.v4.objects), 222  
GroupEpicNoteAwardEmojiManager (class in gitlab.v4.objects), 222  
GroupEpicNoteManager (class in gitlab.v4.objects),



- 222
- GroupEpicResourceLabelEvent (class in gitlab.v4.objects), 223
- GroupEpicResourceLabelEventManager (class in gitlab.v4.objects), 224
- GroupExport (class in gitlab.v4.objects), 224
- GroupExportManager (class in gitlab.v4.objects), 224
- GroupHook (class in gitlab.v4.objects), 225
- GroupHookManager (class in gitlab.v4.objects), 225
- GroupImport (class in gitlab.v4.objects), 227
- GroupImportManager (class in gitlab.v4.objects), 227
- GroupInvitation (class in gitlab.v4.objects), 227
- GroupInvitationManager (class in gitlab.v4.objects), 228
- GroupIssue (class in gitlab.v4.objects), 229
- GroupIssueManager (class in gitlab.v4.objects), 229
- GroupIssuesStatistics (class in gitlab.v4.objects), 230
- GroupIssuesStatisticsManager (class in gitlab.v4.objects), 230
- GroupIteration (class in gitlab.v4.objects), 230
- GroupIterationManager (class in gitlab.v4.objects), 231
- GroupLabel (class in gitlab.v4.objects), 231
- GroupLabelManager (class in gitlab.v4.objects), 232
- GroupManager (class in gitlab.v4.objects), 233
- GroupMember (class in gitlab.v4.objects), 235
- GroupMemberAll (class in gitlab.v4.objects), 236
- GroupMemberAllManager (class in gitlab.v4.objects), 236
- GroupMemberManager (class in gitlab.v4.objects), 236
- GroupMergeRequest (class in gitlab.v4.objects), 237
- GroupMergeRequestManager (class in gitlab.v4.objects), 237
- GroupMilestone (class in gitlab.v4.objects), 238
- GroupMilestoneManager (class in gitlab.v4.objects), 239
- GroupNotificationSettings (class in gitlab.v4.objects), 240
- GroupNotificationSettingsManager (class in gitlab.v4.objects), 241
- GroupPackage (class in gitlab.v4.objects), 241
- GroupPackageManager (class in gitlab.v4.objects), 242
- GroupProject (class in gitlab.v4.objects), 242
- GroupProjectManager (class in gitlab.v4.objects), 242
- GroupPushRules (class in gitlab.v4.objects), 243
- GroupPushRulesManager (class in gitlab.v4.objects), 243
- GroupRegistryRepositoryManager (class in gitlab.v4.objects), 244
- GroupRunner (class in gitlab.v4.objects), 245
- GroupRunnerManager (class in gitlab.v4.objects), 245
- groups (gitlab.Gitlab attribute), 174
- groups (gitlab.v4.objects.Project attribute), 261
- GroupSubgroup (class in gitlab.v4.objects), 245
- GroupSubgroupManager (class in gitlab.v4.objects), 246
- GroupVariable (class in gitlab.v4.objects), 246
- GroupVariableManager (class in gitlab.v4.objects), 246
- GroupWiki (class in gitlab.v4.objects), 247
- GroupWikiManager (class in gitlab.v4.objects), 248
- GUEST (gitlab.const.AccessLevel attribute), 426
- ## H
- headers (gitlab.Gitlab attribute), 174
- Hook (class in gitlab.v4.objects), 248
- HookManager (class in gitlab.v4.objects), 249
- hooks (gitlab.Gitlab attribute), 174
- hooks (gitlab.v4.objects.Group attribute), 204
- hooks (gitlab.v4.objects.Project attribute), 261
- housekeeping() (gitlab.v4.objects.Project method), 261
- http\_delete() (gitlab.Gitlab method), 174
- http\_get() (gitlab.Gitlab method), 174
- http\_head() (gitlab.Gitlab method), 175
- http\_list() (gitlab.Gitlab method), 175
- http\_post() (gitlab.Gitlab method), 175
- http\_put() (gitlab.Gitlab method), 176
- http\_request() (gitlab.Gitlab method), 176
- ## I
- id (gitlab.v4.objects.ProjectMergeRequestApprovalRule attribute), 338
- identityproviders (gitlab.v4.objects.User attribute), 407
- impersonationtokens (gitlab.v4.objects.User attribute), 407
- import\_bitbucket\_server() (gitlab.v4.objects.ProjectManager method), 329
- import\_github() (gitlab.v4.objects.ProjectManager method), 330
- import\_group() (gitlab.v4.objects.GroupManager method), 235
- import\_project() (gitlab.v4.objects.ProjectManager method), 331
- imports (gitlab.v4.objects.Group attribute), 204
- imports (gitlab.v4.objects.Project attribute), 261
- integrations (gitlab.v4.objects.Project attribute), 261
- INTERNAL (gitlab.const.Visibility attribute), 427
- invitations (gitlab.v4.objects.Group attribute), 204
- invitations (gitlab.v4.objects.Project attribute), 261
- Issue (class in gitlab.v4.objects), 249
- IssueManager (class in gitlab.v4.objects), 250
- ISSUES (gitlab.const.SearchScope attribute), 426
- issues (gitlab.Gitlab attribute), 177
- issues (gitlab.v4.objects.Group attribute), 204

issues (*gitlab.v4.objects.GroupEpic* attribute), 217  
issues (*gitlab.v4.objects.Project* attribute), 261  
issues() (*gitlab.v4.objects.GroupMilestone* method), 239  
issues() (*gitlab.v4.objects.ProjectMilestone* method), 352  
issues\_statistics (*gitlab.Gitlab* attribute), 177  
issues\_statistics (*gitlab.v4.objects.Group* attribute), 204  
issues\_statistics (*gitlab.v4.objects.Project* attribute), 261  
IssuesStatistics (class in *gitlab.v4.objects*), 251  
IssuesStatisticsManager (class in *gitlab.v4.objects*), 251  
iterations (*gitlab.v4.objects.Group* attribute), 204  
iterations (*gitlab.v4.objects.Project* attribute), 261

## J

job\_stats() (*gitlab.v4.objects.SidekiqManager* method), 399  
jobs (*gitlab.v4.objects.Project* attribute), 261  
jobs (*gitlab.v4.objects.ProjectPipeline* attribute), 359  
jobs (*gitlab.v4.objects.Runner* attribute), 394

## K

keep\_artifacts() (*gitlab.v4.objects.ProjectJob* method), 319  
Key (class in *gitlab.v4.objects*), 251  
KeyManager (class in *gitlab.v4.objects*), 252  
keys (*gitlab.Gitlab* attribute), 177  
keys (*gitlab.v4.objects.CurrentUser* attribute), 190  
keys (*gitlab.v4.objects.Project* attribute), 261  
keys (*gitlab.v4.objects.User* attribute), 407

## L

labels (*gitlab.v4.objects.Group* attribute), 204  
labels (*gitlab.v4.objects.Project* attribute), 261  
languages() (*gitlab.v4.objects.Project* method), 261  
ldap\_sync() (*gitlab.v4.objects.Group* method), 204  
LDAPGroup (class in *gitlab.v4.objects*), 252  
LDAPGroupManager (class in *gitlab.v4.objects*), 252  
ldapgroups (*gitlab.Gitlab* attribute), 177  
License (class in *gitlab.v4.objects*), 253  
LicenseManager (class in *gitlab.v4.objects*), 253  
licenses (*gitlab.Gitlab* attribute), 177  
links (*gitlab.v4.objects.ProjectIssue* attribute), 305  
links (*gitlab.v4.objects.ProjectRelease* attribute), 373  
lint() (*gitlab.Gitlab* method), 177  
list() (*gitlab.mixins.ListMixin* method), 447  
list() (*gitlab.v4.objects.LDAPGroupManager* method), 253  
list() (*gitlab.v4.objects.UserProjectManager* method), 420

ListMixin (class in *gitlab.mixins*), 447  
lists (*gitlab.v4.objects.GroupBoard* attribute), 211  
lists (*gitlab.v4.objects.ProjectBoard* attribute), 275

## M

main() (in module *gitlab.cli*), 425  
MAINTAINER (*gitlab.const.AccessLevel* attribute), 426  
manager (*gitlab.base.RESTObject* attribute), 423  
manager (*gitlab.mixins.AccessRequestMixin* attribute), 444  
manager (*gitlab.mixins.DownloadMixin* attribute), 446  
manager (*gitlab.mixins.ObjectDeleteMixin* attribute), 448  
manager (*gitlab.mixins.ParticipantsMixin* attribute), 448  
manager (*gitlab.mixins.RefreshMixin* attribute), 448  
manager (*gitlab.mixins.SaveMixin* attribute), 449  
manager (*gitlab.mixins.SubscribableMixin* attribute), 450  
manager (*gitlab.mixins.TimeTrackingMixin* attribute), 451  
manager (*gitlab.mixins.TODOMixin* attribute), 452  
manager (*gitlab.mixins.UserAgentDetailMixin* attribute), 452  
manager (*gitlab.v4.objects.Application* attribute), 182  
manager (*gitlab.v4.objects.ApplicationAppearance* attribute), 182  
manager (*gitlab.v4.objects.ApplicationSettings* attribute), 184  
manager (*gitlab.v4.objects.AuditEvent* attribute), 187  
manager (*gitlab.v4.objects.BroadcastMessage* attribute), 188  
manager (*gitlab.v4.objects.CiLint* attribute), 189  
manager (*gitlab.v4.objects.CurrentUserEmail* attribute), 190  
manager (*gitlab.v4.objects.CurrentUserGPGKey* attribute), 191  
manager (*gitlab.v4.objects.CurrentUserKey* attribute), 192  
manager (*gitlab.v4.objects.CurrentUserStatus* attribute), 193  
manager (*gitlab.v4.objects.DeployKey* attribute), 194  
manager (*gitlab.v4.objects.DeployToken* attribute), 195  
manager (*gitlab.v4.objects.Dockerfile* attribute), 195  
manager (*gitlab.v4.objects.Event* attribute), 196  
manager (*gitlab.v4.objects.Feature* attribute), 197  
manager (*gitlab.v4.objects.GenericPackage* attribute), 198  
manager (*gitlab.v4.objects.GeoNode* attribute), 199  
manager (*gitlab.v4.objects.Gitignore* attribute), 201  
manager (*gitlab.v4.objects.Gitlabciyaml* attribute), 202  
manager (*gitlab.v4.objects.GroupAccessRequest* attribute), 207  
manager (*gitlab.v4.objects.GroupAccessToken* attribute), 207

- manager (*gitlab.v4.objects.GroupAuditEvent* attribute), 208
- manager (*gitlab.v4.objects.GroupBadge* attribute), 209
- manager (*gitlab.v4.objects.GroupBillableMemberMemberships* attribute), 210
- manager (*gitlab.v4.objects.GroupBoardList* attribute), 211
- manager (*gitlab.v4.objects.GroupCluster* attribute), 213
- manager (*gitlab.v4.objects.GroupCustomAttributes* attribute), 214
- manager (*gitlab.v4.objects.GroupDeployToken* attribute), 215
- manager (*gitlab.v4.objects.GroupDescendantGroup* attribute), 216
- manager (*gitlab.v4.objects.GroupEpicAwardEmoji* attribute), 217
- manager (*gitlab.v4.objects.GroupEpicDiscussionNote* attribute), 218
- manager (*gitlab.v4.objects.GroupEpicIssue* attribute), 219
- manager (*gitlab.v4.objects.GroupEpicNoteAwardEmoji* attribute), 222
- manager (*gitlab.v4.objects.GroupEpicResourceLabelEvent* attribute), 223
- manager (*gitlab.v4.objects.GroupExport* attribute), 224
- manager (*gitlab.v4.objects.GroupHook* attribute), 225
- manager (*gitlab.v4.objects.GroupImport* attribute), 227
- manager (*gitlab.v4.objects.GroupInvitation* attribute), 228
- manager (*gitlab.v4.objects.GroupIssue* attribute), 229
- manager (*gitlab.v4.objects.GroupIssuesStatistics* attribute), 230
- manager (*gitlab.v4.objects.GroupIteration* attribute), 231
- manager (*gitlab.v4.objects.GroupLabel* attribute), 231
- manager (*gitlab.v4.objects.GroupMember* attribute), 236
- manager (*gitlab.v4.objects.GroupMemberAll* attribute), 236
- manager (*gitlab.v4.objects.GroupMergeRequest* attribute), 237
- manager (*gitlab.v4.objects.GroupMilestone* attribute), 239
- manager (*gitlab.v4.objects.GroupNotificationSettings* attribute), 241
- manager (*gitlab.v4.objects.GroupPackage* attribute), 242
- manager (*gitlab.v4.objects.GroupProject* attribute), 242
- manager (*gitlab.v4.objects.GroupPushRules* attribute), 243
- manager (*gitlab.v4.objects.GroupRunner* attribute), 245
- manager (*gitlab.v4.objects.GroupSubgroup* attribute), 246
- manager (*gitlab.v4.objects.GroupVariable* attribute), 246
- manager (*gitlab.v4.objects.GroupWiki* attribute), 247
- manager (*gitlab.v4.objects.Hook* attribute), 249
- manager (*gitlab.v4.objects.Issue* attribute), 250
- manager (*gitlab.v4.objects.IssuesStatistics* attribute), 251
- manager (*gitlab.v4.objects.Key* attribute), 252
- manager (*gitlab.v4.objects.LDAPGroup* attribute), 252
- manager (*gitlab.v4.objects.License* attribute), 253
- manager (*gitlab.v4.objects.MergeRequest* attribute), 254
- manager (*gitlab.v4.objects.Namespace* attribute), 256
- manager (*gitlab.v4.objects.NotificationSettings* attribute), 257
- manager (*gitlab.v4.objects.PagesDomain* attribute), 258
- manager (*gitlab.v4.objects.PersonalAccessToken* attribute), 259
- manager (*gitlab.v4.objects.ProjectAccessRequest* attribute), 266
- manager (*gitlab.v4.objects.ProjectAccessToken* attribute), 267
- manager (*gitlab.v4.objects.ProjectAdditionalStatistics* attribute), 268
- manager (*gitlab.v4.objects.ProjectApproval* attribute), 268
- manager (*gitlab.v4.objects.ProjectApprovalRule* attribute), 270
- manager (*gitlab.v4.objects.ProjectArtifact* attribute), 270
- manager (*gitlab.v4.objects.ProjectAudit* attribute), 272
- manager (*gitlab.v4.objects.ProjectAuditEvent* attribute), 273
- manager (*gitlab.v4.objects.ProjectBadge* attribute), 274
- manager (*gitlab.v4.objects.ProjectBoardList* attribute), 275
- manager (*gitlab.v4.objects.ProjectBranch* attribute), 277
- manager (*gitlab.v4.objects.ProjectCiLint* attribute), 278
- manager (*gitlab.v4.objects.ProjectCluster* attribute), 279
- manager (*gitlab.v4.objects.ProjectCommitComment* attribute), 282
- manager (*gitlab.v4.objects.ProjectCommitDiscussionNote* attribute), 284
- manager (*gitlab.v4.objects.ProjectCommitStatus* attribute), 286
- manager (*gitlab.v4.objects.ProjectCustomAttribute* attribute), 287
- manager (*gitlab.v4.objects.ProjectDeploymentMergeRequest* attribute), 290
- manager (*gitlab.v4.objects.ProjectDeployToken* attribute), 288
- manager (*gitlab.v4.objects.ProjectEnvironment* attribute), 292
- manager (*gitlab.v4.objects.ProjectEvent* attribute), 293
- manager (*gitlab.v4.objects.ProjectExport* attribute), 294
- manager (*gitlab.v4.objects.ProjectFile* attribute), 295
- manager (*gitlab.v4.objects.ProjectFork* attribute), 299
- manager (*gitlab.v4.objects.ProjectHook* attribute), 300
- manager (*gitlab.v4.objects.ProjectImport* attribute), 302
- manager (*gitlab.v4.objects.ProjectIntegration* attribute), 302
- manager (*gitlab.v4.objects.ProjectInvitation* attribute),

- 303
- manager (*gitlab.v4.objects.ProjectIssueAwardEmoji attribute*), 306
- manager (*gitlab.v4.objects.ProjectIssueDiscussionNote attribute*), 308
- manager (*gitlab.v4.objects.ProjectIssueLink attribute*), 309
- manager (*gitlab.v4.objects.ProjectIssueNoteAwardEmoji attribute*), 312
- manager (*gitlab.v4.objects.ProjectIssueResourceLabelEvening attribute*), 314
- manager (*gitlab.v4.objects.ProjectIssueResourceMilestoneEvening attribute*), 315
- manager (*gitlab.v4.objects.ProjectIssueResourceStateEvent attribute*), 315
- manager (*gitlab.v4.objects.ProjectIssuesStatistics attribute*), 316
- manager (*gitlab.v4.objects.ProjectJob attribute*), 319
- manager (*gitlab.v4.objects.ProjectKey attribute*), 321
- manager (*gitlab.v4.objects.ProjectLabel attribute*), 322
- manager (*gitlab.v4.objects.ProjectMember attribute*), 332
- manager (*gitlab.v4.objects.ProjectMemberAll attribute*), 332
- manager (*gitlab.v4.objects.ProjectMergeRequestApproval attribute*), 337
- manager (*gitlab.v4.objects.ProjectMergeRequestApprovalState attribute*), 340
- manager (*gitlab.v4.objects.ProjectMergeRequestAwardEmoji attribute*), 341
- manager (*gitlab.v4.objects.ProjectMergeRequestDiff attribute*), 342
- manager (*gitlab.v4.objects.ProjectMergeRequestDiscussionNote attribute*), 344
- manager (*gitlab.v4.objects.ProjectMergeRequestNoteAwardEmoji attribute*), 347
- manager (*gitlab.v4.objects.ProjectMergeRequestPipeline attribute*), 349
- manager (*gitlab.v4.objects.ProjectMergeRequestResourceLabelEvening attribute*), 349
- manager (*gitlab.v4.objects.ProjectMergeRequestResourceMilestoneEvening attribute*), 350
- manager (*gitlab.v4.objects.ProjectMergeRequestResourceSnippet attribute*), 351
- manager (*gitlab.v4.objects.ProjectMergeTrain attribute*), 352
- manager (*gitlab.v4.objects.ProjectMilestone attribute*), 353
- manager (*gitlab.v4.objects.ProjectNote attribute*), 355
- manager (*gitlab.v4.objects.ProjectNotificationSettings attribute*), 355
- manager (*gitlab.v4.objects.ProjectPackageFile attribute*), 357
- manager (*gitlab.v4.objects.ProjectPagesDomain attribute*), 358
- manager (*gitlab.v4.objects.ProjectPipelineBridge attribute*), 360
- manager (*gitlab.v4.objects.ProjectPipelineJob attribute*), 361
- manager (*gitlab.v4.objects.ProjectPipelineScheduleVariable attribute*), 364
- manager (*gitlab.v4.objects.ProjectPipelineTestReport attribute*), 365
- manager (*gitlab.v4.objects.ProjectPipelineTestReportSummary attribute*), 366
- manager (*gitlab.v4.objects.ProjectPipelineVariable attribute*), 366
- manager (*gitlab.v4.objects.ProjectProtectedBranch attribute*), 367
- manager (*gitlab.v4.objects.ProjectProtectedEnvironment attribute*), 368
- manager (*gitlab.v4.objects.ProjectProtectedTag attribute*), 369
- manager (*gitlab.v4.objects.ProjectPushRules attribute*), 370
- manager (*gitlab.v4.objects.ProjectRegistryTag attribute*), 372
- manager (*gitlab.v4.objects.ProjectReleaseLink attribute*), 373
- manager (*gitlab.v4.objects.ProjectRemoteMirror attribute*), 375
- manager (*gitlab.v4.objects.ProjectRunner attribute*), 376
- manager (*gitlab.v4.objects.ProjectService attribute*), 377
- manager (*gitlab.v4.objects.ProjectSnippetAwardEmoji attribute*), 378
- manager (*gitlab.v4.objects.ProjectSnippetDiscussionNote attribute*), 380
- manager (*gitlab.v4.objects.ProjectSnippetNoteAwardEmoji attribute*), 383
- manager (*gitlab.v4.objects.ProjectStorage attribute*), 384
- manager (*gitlab.v4.objects.ProjectTag attribute*), 385
- manager (*gitlab.v4.objects.ProjectTrigger attribute*), 386
- manager (*gitlab.v4.objects.ProjectUser attribute*), 387
- manager (*gitlab.v4.objects.ProjectVariable attribute*), 389
- manager (*gitlab.v4.objects.ProjectWiki attribute*), 389
- manager (*gitlab.v4.objects.RegistryRepository attribute*), 390
- manager (*gitlab.v4.objects.RunnerAll attribute*), 394
- manager (*gitlab.v4.objects.RunnerJob attribute*), 395
- manager (*gitlab.v4.objects.SharedProject attribute*), 398
- manager (*gitlab.v4.objects.Snippet attribute*), 400
- manager (*gitlab.v4.objects.StarredProject attribute*), 402
- manager (*gitlab.v4.objects.TODO attribute*), 403
- manager (*gitlab.v4.objects.Topic attribute*), 404
- manager (*gitlab.v4.objects.UserActivities attribute*), 409
- manager (*gitlab.v4.objects.UserCustomAttribute attribute*), 409



- `manager` (`gitlab.v4.objects.UserEmail` attribute), 410
  - `manager` (`gitlab.v4.objects.UserEvent` attribute), 411
  - `manager` (`gitlab.v4.objects.UserGPGKey` attribute), 411
  - `manager` (`gitlab.v4.objects.UserImpersonationToken` attribute), 412
  - `manager` (`gitlab.v4.objects.UserKey` attribute), 414
  - `manager` (`gitlab.v4.objects.UserMembership` attribute), 417
  - `manager` (`gitlab.v4.objects.UserPersonalAccessToken` attribute), 418
  - `manager` (`gitlab.v4.objects.UserProject` attribute), 418
  - `manager` (`gitlab.v4.objects.UserStatus` attribute), 420
  - `manager` (`gitlab.v4.objects.Variable` attribute), 421
  - `mark_all_as_done()` (`gitlab.v4.objects.TODOManager` method), 403
  - `mark_as_done()` (`gitlab.v4.objects.TODO` method), 403
  - `markdown()` (`gitlab.Gitlab` method), 178
  - `members` (`gitlab.v4.objects.Group` attribute), 204
  - `members` (`gitlab.v4.objects.Project` attribute), 261
  - `members_all` (`gitlab.v4.objects.Group` attribute), 204
  - `members_all` (`gitlab.v4.objects.Project` attribute), 261
  - `memberships` (`gitlab.v4.objects.GroupBillableMember` attribute), 210
  - `memberships` (`gitlab.v4.objects.User` attribute), 407
  - `MENTION` (`gitlab.const.NotificationLevel` attribute), 426
  - `merge()` (`gitlab.v4.objects.ProjectMergeRequest` method), 335
  - `merge_config()` (`gitlab.Gitlab` class method), 178
  - `merge_ref()` (`gitlab.v4.objects.ProjectMergeRequest` method), 336
  - `merge_request_iid` (`gitlab.v4.objects.ProjectMergeRequestApprovalRule` attribute), 338
  - `MERGE_REQUESTS` (`gitlab.const.SearchScope` attribute), 427
  - `merge_requests()` (`gitlab.v4.objects.GroupMilestone` method), 239
  - `merge_requests()` (`gitlab.v4.objects.ProjectCommit` method), 281
  - `merge_requests()` (`gitlab.v4.objects.ProjectMilestone` method), 353
  - `merge_trains` (`gitlab.v4.objects.Project` attribute), 261
  - `MergeRequest` (class in `gitlab.v4.objects`), 254
  - `MergeRequestManager` (class in `gitlab.v4.objects`), 254
  - `mergerequests` (`gitlab.Gitlab` attribute), 179
  - `mergerequests` (`gitlab.v4.objects.Group` attribute), 204
  - `mergerequests` (`gitlab.v4.objects.Project` attribute), 262
  - `mergerequests` (`gitlab.v4.objects.ProjectDeployment` attribute), 289
  - `MILESTONES` (`gitlab.const.SearchScope` attribute), 427
  - `milestones` (`gitlab.v4.objects.Group` attribute), 204
  - `milestones` (`gitlab.v4.objects.Project` attribute), 262
  - `MINIMAL_ACCESS` (`gitlab.const.AccessLevel` attribute), 426
  - `mirror_pull()` (`gitlab.v4.objects.Project` method), 262
  - module
    - `gitlab.base`, 422
    - `gitlab.cli`, 424
    - `gitlab.config`, 425
    - `gitlab.const`, 426
    - `gitlab.exceptions`, 427
    - `gitlab.mixins`, 444
    - `gitlab.utils`, 453
    - `gitlab.v4`, 422
    - `gitlab.v4.objects`, 182
  - `move()` (`gitlab.v4.objects.ProjectIssue` method), 305
- ## N
- `Namespace` (class in `gitlab.v4.objects`), 255
  - `NamespaceManager` (class in `gitlab.v4.objects`), 256
  - `namespaces` (`gitlab.Gitlab` attribute), 179
  - `next()` (`gitlab.base.RESTObjectList` method), 424
  - `next()` (`gitlab.GitlabList` method), 181
  - `next_page` (`gitlab.base.RESTObjectList` property), 424
  - `next_page` (`gitlab.GitlabList` property), 181
  - `NO_ACCESS` (`gitlab.const.AccessLevel` attribute), 426
  - `notes` (`gitlab.v4.objects.GroupEpic` attribute), 217
  - `notes` (`gitlab.v4.objects.Project` attribute), 262
  - `notes` (`gitlab.v4.objects.ProjectCommitDiscussion` attribute), 283
  - `notes` (`gitlab.v4.objects.ProjectIssue` attribute), 305
  - `notes` (`gitlab.v4.objects.ProjectIssueDiscussion` attribute), 307
  - `notes` (`gitlab.v4.objects.ProjectMergeRequest` attribute), 336
  - `notes` (`gitlab.v4.objects.ProjectMergeRequestDiscussion` attribute), 343
  - `notes` (`gitlab.v4.objects.ProjectSnippet` attribute), 378
  - `notes` (`gitlab.v4.objects.ProjectSnippetDiscussion` attribute), 379
  - `NotificationLevel` (class in `gitlab.const`), 426
  - `NotificationSettings` (class in `gitlab.v4.objects`), 257
  - `notificationsettings` (`gitlab.Gitlab` attribute), 179
  - `notificationsettings` (`gitlab.v4.objects.Group` attribute), 204
  - `notificationsettings` (`gitlab.v4.objects.Project` attribute), 262
  - `NotificationSettingsManager` (class in `gitlab.v4.objects`), 257
  - `NoUpdateMixin` (class in `gitlab.mixins`), 447
- ## O
- `ObjectDeleteMixin` (class in `gitlab.mixins`), 448
  - `OWNER` (`gitlab.const.AccessLevel` attribute), 426
- ## P
- `package_files` (`gitlab.v4.objects.ProjectPackage`

- attribute*), 357
- packages* (*gitlab.v4.objects.Group* attribute), 205
- packages* (*gitlab.v4.objects.Project* attribute), 262
- PagesDomain* (class in *gitlab.v4.objects*), 258
- PagesDomainManager* (class in *gitlab.v4.objects*), 258
- pagesdomains* (*gitlab.Gitlab* attribute), 179
- pagesdomains* (*gitlab.v4.objects.Project* attribute), 262
- parent\_attrs* (*gitlab.base.RESTManager* property), 422
- participants()* (*gitlab.mixins.ParticipantsMixin* method), 448
- ParticipantsMixin* (class in *gitlab.mixins*), 448
- PARTICIPATING* (*gitlab.const.NotificationLevel* attribute), 426
- path* (*gitlab.base.RESTManager* property), 422
- per\_page* (*gitlab.base.RESTObjectList* property), 424
- per\_page* (*gitlab.GitlabList* property), 181
- personal\_access\_tokens* (*gitlab.Gitlab* attribute), 179
- personal\_access\_tokens* (*gitlab.v4.objects.User* attribute), 407
- PersonalAccessToken* (class in *gitlab.v4.objects*), 258
- PersonalAccessTokenManager* (class in *gitlab.v4.objects*), 259
- pformat()* (*gitlab.base.RESTObject* method), 423
- pipelines* (*gitlab.v4.objects.Project* attribute), 262
- pipelines* (*gitlab.v4.objects.ProjectMergeRequest* attribute), 336
- pipelineschedules* (*gitlab.v4.objects.Project* attribute), 262
- play()* (*gitlab.v4.objects.ProjectJob* method), 319
- play()* (*gitlab.v4.objects.ProjectPipelineSchedule* method), 362
- pprint()* (*gitlab.base.RESTObject* method), 423
- prev\_page* (*gitlab.base.RESTObjectList* property), 424
- prev\_page* (*gitlab.GitlabList* property), 181
- PRIVATE* (*gitlab.const.Visibility* attribute), 427
- process\_metrics()* (*gitlab.v4.objects.SidekiqManager* method), 399
- Project* (class in *gitlab.v4.objects*), 259
- PROJECT\_NOTES* (*gitlab.const.SearchScope* attribute), 427
- ProjectAccessRequest* (class in *gitlab.v4.objects*), 266
- ProjectAccessRequestManager* (class in *gitlab.v4.objects*), 267
- ProjectAccessToken* (class in *gitlab.v4.objects*), 267
- ProjectAccessTokenManager* (class in *gitlab.v4.objects*), 267
- ProjectAdditionalStatistics* (class in *gitlab.v4.objects*), 267
- ProjectAdditionalStatisticsManager* (class in *gitlab.v4.objects*), 268
- ProjectApproval* (class in *gitlab.v4.objects*), 268
- ProjectApprovalManager* (class in *gitlab.v4.objects*), 268
- ProjectApprovalRule* (class in *gitlab.v4.objects*), 270
- ProjectApprovalRuleManager* (class in *gitlab.v4.objects*), 270
- ProjectArtifact* (class in *gitlab.v4.objects*), 270
- ProjectArtifactManager* (class in *gitlab.v4.objects*), 270
- ProjectAudit* (class in *gitlab.v4.objects*), 272
- ProjectAuditEvent* (class in *gitlab.v4.objects*), 272
- ProjectAuditEventManager* (class in *gitlab.v4.objects*), 273
- ProjectAuditManager* (class in *gitlab.v4.objects*), 273
- ProjectBadge* (class in *gitlab.v4.objects*), 274
- ProjectBadgeManager* (class in *gitlab.v4.objects*), 274
- ProjectBoard* (class in *gitlab.v4.objects*), 275
- ProjectBoardList* (class in *gitlab.v4.objects*), 275
- ProjectBoardListManager* (class in *gitlab.v4.objects*), 275
- ProjectBoardManager* (class in *gitlab.v4.objects*), 276
- ProjectBranch* (class in *gitlab.v4.objects*), 277
- ProjectBranchManager* (class in *gitlab.v4.objects*), 277
- ProjectCiLint* (class in *gitlab.v4.objects*), 277
- ProjectCiLintManager* (class in *gitlab.v4.objects*), 278
- ProjectCluster* (class in *gitlab.v4.objects*), 279
- ProjectClusterManager* (class in *gitlab.v4.objects*), 279
- ProjectCommit* (class in *gitlab.v4.objects*), 280
- ProjectCommitComment* (class in *gitlab.v4.objects*), 282
- ProjectCommitCommentManager* (class in *gitlab.v4.objects*), 283
- ProjectCommitDiscussion* (class in *gitlab.v4.objects*), 283
- ProjectCommitDiscussionManager* (class in *gitlab.v4.objects*), 283
- ProjectCommitDiscussionNote* (class in *gitlab.v4.objects*), 284
- ProjectCommitDiscussionNoteManager* (class in *gitlab.v4.objects*), 284
- ProjectCommitManager* (class in *gitlab.v4.objects*), 285
- ProjectCommitStatus* (class in *gitlab.v4.objects*), 286
- ProjectCommitStatusManager* (class in *gitlab.v4.objects*), 286
- ProjectCustomAttribute* (class in *gitlab.v4.objects*), 287
- ProjectCustomAttributeManager* (class in *gitlab.v4.objects*), 287
- ProjectDeployment* (class in *gitlab.v4.objects*), 289
- ProjectDeploymentManager* (class in *gitlab.v4.objects*), 289

- ProjectDeploymentMergeRequest (class in *gitlab.v4.objects*), 290
- ProjectDeploymentMergeRequestManager (class in *gitlab.v4.objects*), 290
- ProjectDeployToken (class in *gitlab.v4.objects*), 288
- ProjectDeployTokenManager (class in *gitlab.v4.objects*), 288
- ProjectEnvironment (class in *gitlab.v4.objects*), 291
- ProjectEnvironmentManager (class in *gitlab.v4.objects*), 292
- ProjectEvent (class in *gitlab.v4.objects*), 293
- ProjectEventManager (class in *gitlab.v4.objects*), 293
- ProjectExport (class in *gitlab.v4.objects*), 294
- ProjectExportManager (class in *gitlab.v4.objects*), 294
- ProjectFile (class in *gitlab.v4.objects*), 294
- ProjectFileManager (class in *gitlab.v4.objects*), 295
- ProjectFork (class in *gitlab.v4.objects*), 298
- ProjectForkManager (class in *gitlab.v4.objects*), 299
- ProjectHook (class in *gitlab.v4.objects*), 300
- ProjectHookManager (class in *gitlab.v4.objects*), 300
- ProjectImport (class in *gitlab.v4.objects*), 301
- ProjectImportManager (class in *gitlab.v4.objects*), 302
- ProjectIntegration (class in *gitlab.v4.objects*), 302
- ProjectIntegrationManager (class in *gitlab.v4.objects*), 302
- ProjectInvitation (class in *gitlab.v4.objects*), 303
- ProjectInvitationManager (class in *gitlab.v4.objects*), 303
- ProjectIssue (class in *gitlab.v4.objects*), 304
- ProjectIssueAwardEmoji (class in *gitlab.v4.objects*), 306
- ProjectIssueAwardEmojiManager (class in *gitlab.v4.objects*), 306
- ProjectIssueDiscussion (class in *gitlab.v4.objects*), 307
- ProjectIssueDiscussionManager (class in *gitlab.v4.objects*), 307
- ProjectIssueDiscussionNote (class in *gitlab.v4.objects*), 308
- ProjectIssueDiscussionNoteManager (class in *gitlab.v4.objects*), 308
- ProjectIssueLink (class in *gitlab.v4.objects*), 309
- ProjectIssueLinkManager (class in *gitlab.v4.objects*), 309
- ProjectIssueManager (class in *gitlab.v4.objects*), 310
- ProjectIssueNote (class in *gitlab.v4.objects*), 312
- ProjectIssueNoteAwardEmoji (class in *gitlab.v4.objects*), 312
- ProjectIssueNoteAwardEmojiManager (class in *gitlab.v4.objects*), 312
- ProjectIssueNoteManager (class in *gitlab.v4.objects*), 313
- ProjectIssueResourceLabelEvent (class in *gitlab.v4.objects*), 313
- ProjectIssueResourceLabelEventManager (class in *gitlab.v4.objects*), 314
- ProjectIssueResourceMilestoneEvent (class in *gitlab.v4.objects*), 314
- ProjectIssueResourceMilestoneEventManager (class in *gitlab.v4.objects*), 315
- ProjectIssueResourceStateEvent (class in *gitlab.v4.objects*), 315
- ProjectIssueResourceStateEventManager (class in *gitlab.v4.objects*), 315
- ProjectIssuesStatistics (class in *gitlab.v4.objects*), 316
- ProjectIssuesStatisticsManager (class in *gitlab.v4.objects*), 316
- ProjectIterationManager (class in *gitlab.v4.objects*), 317
- ProjectJob (class in *gitlab.v4.objects*), 317
- ProjectJobManager (class in *gitlab.v4.objects*), 320
- ProjectKey (class in *gitlab.v4.objects*), 321
- ProjectKeyManager (class in *gitlab.v4.objects*), 321
- ProjectLabel (class in *gitlab.v4.objects*), 322
- ProjectLabelManager (class in *gitlab.v4.objects*), 323
- ProjectManager (class in *gitlab.v4.objects*), 324
- ProjectMember (class in *gitlab.v4.objects*), 331
- ProjectMemberAll (class in *gitlab.v4.objects*), 332
- ProjectMemberAllManager (class in *gitlab.v4.objects*), 332
- ProjectMemberManager (class in *gitlab.v4.objects*), 332
- ProjectMergeRequest (class in *gitlab.v4.objects*), 333
- ProjectMergeRequestApproval (class in *gitlab.v4.objects*), 337
- ProjectMergeRequestApprovalManager (class in *gitlab.v4.objects*), 337
- ProjectMergeRequestApprovalRule (class in *gitlab.v4.objects*), 338
- ProjectMergeRequestApprovalRuleManager (class in *gitlab.v4.objects*), 338
- ProjectMergeRequestApprovalState (class in *gitlab.v4.objects*), 340
- ProjectMergeRequestApprovalStateManager (class in *gitlab.v4.objects*), 340
- ProjectMergeRequestAwardEmoji (class in *gitlab.v4.objects*), 341
- ProjectMergeRequestAwardEmojiManager (class in *gitlab.v4.objects*), 341
- ProjectMergeRequestDiff (class in *gitlab.v4.objects*), 342
- ProjectMergeRequestDiffManager (class in *gitlab.v4.objects*), 342
- ProjectMergeRequestDiscussion (class in *gitlab.v4.objects*), 343

`ProjectMergeRequestDiscussionManager` (class in `gitlab.v4.objects`), 343

`ProjectMergeRequestDiscussionNote` (class in `gitlab.v4.objects`), 344

`ProjectMergeRequestDiscussionNoteManager` (class in `gitlab.v4.objects`), 344

`ProjectMergeRequestManager` (class in `gitlab.v4.objects`), 345

`ProjectMergeRequestNote` (class in `gitlab.v4.objects`), 347

`ProjectMergeRequestNoteAwardEmoji` (class in `gitlab.v4.objects`), 347

`ProjectMergeRequestNoteAwardEmojiManager` (class in `gitlab.v4.objects`), 347

`ProjectMergeRequestNoteManager` (class in `gitlab.v4.objects`), 348

`ProjectMergeRequestPipeline` (class in `gitlab.v4.objects`), 349

`ProjectMergeRequestPipelineManager` (class in `gitlab.v4.objects`), 349

`ProjectMergeRequestResourceLabelEvent` (class in `gitlab.v4.objects`), 349

`ProjectMergeRequestResourceLabelEventManager` (class in `gitlab.v4.objects`), 349

`ProjectMergeRequestResourceMilestoneEvent` (class in `gitlab.v4.objects`), 350

`ProjectMergeRequestResourceMilestoneEventManager` (class in `gitlab.v4.objects`), 350

`ProjectMergeRequestResourceStateEvent` (class in `gitlab.v4.objects`), 351

`ProjectMergeRequestResourceStateEventManager` (class in `gitlab.v4.objects`), 351

`ProjectMergeTrain` (class in `gitlab.v4.objects`), 352

`ProjectMergeTrainManager` (class in `gitlab.v4.objects`), 352

`ProjectMilestone` (class in `gitlab.v4.objects`), 352

`ProjectMilestoneManager` (class in `gitlab.v4.objects`), 353

`ProjectNote` (class in `gitlab.v4.objects`), 354

`ProjectNoteManager` (class in `gitlab.v4.objects`), 355

`ProjectNotificationSettings` (class in `gitlab.v4.objects`), 355

`ProjectNotificationSettingsManager` (class in `gitlab.v4.objects`), 355

`ProjectPackage` (class in `gitlab.v4.objects`), 356

`ProjectPackageFile` (class in `gitlab.v4.objects`), 357

`ProjectPackageFileManager` (class in `gitlab.v4.objects`), 357

`ProjectPackageManager` (class in `gitlab.v4.objects`), 357

`ProjectPagesDomain` (class in `gitlab.v4.objects`), 358

`ProjectPagesDomainManager` (class in `gitlab.v4.objects`), 358

`ProjectPipeline` (class in `gitlab.v4.objects`), 359

`ProjectPipelineBridge` (class in `gitlab.v4.objects`), 360

`ProjectPipelineBridgeManager` (class in `gitlab.v4.objects`), 360

`ProjectPipelineJob` (class in `gitlab.v4.objects`), 360

`ProjectPipelineJobManager` (class in `gitlab.v4.objects`), 361

`ProjectPipelineManager` (class in `gitlab.v4.objects`), 361

`ProjectPipelineSchedule` (class in `gitlab.v4.objects`), 362

`ProjectPipelineScheduleManager` (class in `gitlab.v4.objects`), 363

`ProjectPipelineScheduleVariable` (class in `gitlab.v4.objects`), 364

`ProjectPipelineScheduleVariableManager` (class in `gitlab.v4.objects`), 364

`ProjectPipelineTestReport` (class in `gitlab.v4.objects`), 365

`ProjectPipelineTestReportManager` (class in `gitlab.v4.objects`), 365

`ProjectPipelineTestReportSummary` (class in `gitlab.v4.objects`), 365

`ProjectPipelineTestReportSummaryManager` (class in `gitlab.v4.objects`), 366

`ProjectPipelineVariable` (class in `gitlab.v4.objects`), 366

`ProjectPipelineVariableManager` (class in `gitlab.v4.objects`), 366

`ProjectProtectedBranch` (class in `gitlab.v4.objects`), 367

`ProjectProtectedBranchManager` (class in `gitlab.v4.objects`), 367

`ProjectProtectedEnvironment` (class in `gitlab.v4.objects`), 368

`ProjectProtectedEnvironmentManager` (class in `gitlab.v4.objects`), 368

`ProjectProtectedTag` (class in `gitlab.v4.objects`), 369

`ProjectProtectedTagManager` (class in `gitlab.v4.objects`), 369

`ProjectPushRules` (class in `gitlab.v4.objects`), 370

`ProjectPushRulesManager` (class in `gitlab.v4.objects`), 370

`ProjectRegistryRepository` (class in `gitlab.v4.objects`), 371

`ProjectRegistryRepositoryManager` (class in `gitlab.v4.objects`), 371

`ProjectRegistryTag` (class in `gitlab.v4.objects`), 372

`ProjectRegistryTagManager` (class in `gitlab.v4.objects`), 372

`ProjectRelease` (class in `gitlab.v4.objects`), 373

`ProjectReleaseLink` (class in `gitlab.v4.objects`), 373

`ProjectReleaseLinkManager` (class in `gitlab.v4.objects`), 373



- ProjectReleaseManager (class in `gitlab.v4.objects`), 374
- ProjectRemoteMirror (class in `gitlab.v4.objects`), 375
- ProjectRemoteMirrorManager (class in `gitlab.v4.objects`), 376
- ProjectRunner (class in `gitlab.v4.objects`), 376
- ProjectRunnerManager (class in `gitlab.v4.objects`), 376
- PROJECTS (`gitlab.const.SearchScope` attribute), 427
- projects (`gitlab.Gitlab` attribute), 179
- projects (`gitlab.v4.objects.Group` attribute), 205
- projects (`gitlab.v4.objects.User` attribute), 407
- ProjectService (class in `gitlab.v4.objects`), 377
- ProjectServiceManager (class in `gitlab.v4.objects`), 377
- ProjectSnippet (class in `gitlab.v4.objects`), 377
- ProjectSnippetAwardEmoji (class in `gitlab.v4.objects`), 378
- ProjectSnippetAwardEmojiManager (class in `gitlab.v4.objects`), 378
- ProjectSnippetDiscussion (class in `gitlab.v4.objects`), 379
- ProjectSnippetDiscussionManager (class in `gitlab.v4.objects`), 379
- ProjectSnippetDiscussionNote (class in `gitlab.v4.objects`), 380
- ProjectSnippetDiscussionNoteManager (class in `gitlab.v4.objects`), 380
- ProjectSnippetManager (class in `gitlab.v4.objects`), 381
- ProjectSnippetNote (class in `gitlab.v4.objects`), 382
- ProjectSnippetNoteAwardEmoji (class in `gitlab.v4.objects`), 382
- ProjectSnippetNoteAwardEmojiManager (class in `gitlab.v4.objects`), 383
- ProjectSnippetNoteManager (class in `gitlab.v4.objects`), 383
- ProjectStorage (class in `gitlab.v4.objects`), 384
- ProjectStorageManager (class in `gitlab.v4.objects`), 384
- ProjectTag (class in `gitlab.v4.objects`), 385
- ProjectTagManager (class in `gitlab.v4.objects`), 385
- ProjectTrigger (class in `gitlab.v4.objects`), 386
- ProjectTriggerManager (class in `gitlab.v4.objects`), 386
- ProjectUser (class in `gitlab.v4.objects`), 387
- ProjectUserManager (class in `gitlab.v4.objects`), 387
- ProjectVariable (class in `gitlab.v4.objects`), 387
- ProjectVariableManager (class in `gitlab.v4.objects`), 387
- ProjectWiki (class in `gitlab.v4.objects`), 388
- ProjectWikiManager (class in `gitlab.v4.objects`), 389
- protected\_environments (`gitlab.v4.objects.Project` attribute), 262
- protectedbranches (`gitlab.v4.objects.Project` attribute), 262
- protectedtags (`gitlab.v4.objects.Project` attribute), 262
- PUBLIC (`gitlab.const.Visibility` attribute), 427
- public() (`gitlab.v4.objects.SnippetManager` method), 401
- pushrules (`gitlab.v4.objects.Group` attribute), 205
- pushrules (`gitlab.v4.objects.Project` attribute), 262
- ## Q
- queue\_metrics() (`gitlab.v4.objects.SidekiqManager` method), 399
- ## R
- raw() (`gitlab.v4.objects.ProjectArtifactManager` method), 272
- raw() (`gitlab.v4.objects.ProjectFileManager` method), 298
- rebase() (`gitlab.v4.objects.ProjectMergeRequest` method), 336
- RedirectError, 444
- refresh() (`gitlab.mixins.RefreshMixin` method), 448
- RefreshMixin (class in `gitlab.mixins`), 448
- refs() (`gitlab.v4.objects.ProjectCommit` method), 281
- register\_custom\_action() (in module `gitlab.cli`), 425
- registry\_repositories (`gitlab.Gitlab` attribute), 179
- registry\_repositories (`gitlab.v4.objects.Group` attribute), 205
- RegistryRepository (class in `gitlab.v4.objects`), 390
- RegistryRepositoryManager (class in `gitlab.v4.objects`), 390
- reject() (`gitlab.v4.objects.User` method), 407
- related\_merge\_requests() (`gitlab.v4.objects.ProjectIssue` method), 305
- releases (`gitlab.v4.objects.Project` attribute), 262
- remote\_mirrors (`gitlab.v4.objects.Project` attribute), 262
- remove\_none\_from\_dict() (in module `gitlab.utils`), 453
- render() (`gitlab.mixins.BadgeRenderMixin` method), 444
- reorder() (`gitlab.v4.objects.ProjectIssue` method), 306
- repair() (`gitlab.v4.objects.GeoNode` method), 199
- REPORTER (`gitlab.const.AccessLevel` attribute), 426
- repositories (`gitlab.v4.objects.Project` attribute), 262
- repository\_archive() (`gitlab.v4.objects.RepositoryMixin` method), 391
- repository\_blob() (`gitlab.v4.objects.RepositoryMixin` method), 391
- repository\_compare() (`gitlab.v4.objects.RepositoryMixin` method), 392

- `repository_contributors()` (*gitlab.v4.objects.RepositoryMixin* method), 392
  - `repository_merge_base()` (*gitlab.v4.objects.RepositoryMixin* method), 392
  - `repository_raw_blob()` (*gitlab.v4.objects.RepositoryMixin* method), 393
  - `repository_tree()` (*gitlab.v4.objects.RepositoryMixin* method), 393
  - `RepositoryMixin` (class in *gitlab.v4.objects*), 390
  - `reset_spent_time()` (*gitlab.mixins.TimeTrackingMixin* method), 451
  - `reset_time_estimate()` (*gitlab.mixins.TimeTrackingMixin* method), 451
  - `resourcelabelevents` (*gitlab.v4.objects.GroupEpic* attribute), 217
  - `resourcelabelevents` (*gitlab.v4.objects.ProjectIssue* attribute), 306
  - `resourcelabelevents` (*gitlab.v4.objects.ProjectMergeRequest* attribute), 336
  - `resourcemilestoneevents` (*gitlab.v4.objects.ProjectIssue* attribute), 306
  - `resourcemilestoneevents` (*gitlab.v4.objects.ProjectMergeRequest* attribute), 336
  - `resourcestateevents` (*gitlab.v4.objects.ProjectIssue* attribute), 306
  - `resourcestateevents` (*gitlab.v4.objects.ProjectMergeRequest* attribute), 336
  - `response_content()` (in module *gitlab.utils*), 453
  - `RESTManager` (class in *gitlab.base*), 422
  - `RESTObject` (class in *gitlab.base*), 422
  - `RESTObjectList` (class in *gitlab.base*), 423
  - `restore()` (*gitlab.v4.objects.Project* method), 262
  - `RetrieveMixin` (class in *gitlab.mixins*), 449
  - `retry()` (*gitlab.v4.objects.ProjectJob* method), 320
  - `retry()` (*gitlab.v4.objects.ProjectPipeline* method), 359
  - `revert()` (*gitlab.v4.objects.ProjectCommit* method), 282
  - `Runner` (class in *gitlab.v4.objects*), 394
  - `RunnerAll` (class in *gitlab.v4.objects*), 394
  - `RunnerAllManager` (class in *gitlab.v4.objects*), 395
  - `RunnerJob` (class in *gitlab.v4.objects*), 395
  - `RunnerJobManager` (class in *gitlab.v4.objects*), 395
  - `RunnerManager` (class in *gitlab.v4.objects*), 395
  - `runners` (*gitlab.Gitlab* attribute), 179
  - `runners` (*gitlab.v4.objects.Group* attribute), 205
  - `runners` (*gitlab.v4.objects.Project* attribute), 262
  - `runners_all` (*gitlab.Gitlab* attribute), 179
- ## S
- `save()` (*gitlab.mixins.SaveMixin* method), 449
  - `save()` (*gitlab.v4.objects.GroupEpicIssue* method), 219
  - `save()` (*gitlab.v4.objects.GroupLabel* method), 231
  - `save()` (*gitlab.v4.objects.ProjectFile* method), 295
  - `save()` (*gitlab.v4.objects.ProjectLabel* method), 323
  - `save()` (*gitlab.v4.objects.ProjectMergeRequestApprovalRule* method), 338
  - `SaveMixin` (class in *gitlab.mixins*), 449
  - `search()` (*gitlab.Gitlab* method), 179
  - `search()` (*gitlab.v4.objects.Group* method), 205
  - `search()` (*gitlab.v4.objects.Project* method), 262
  - `SearchScope` (class in *gitlab.const*), 426
  - `services` (*gitlab.v4.objects.Project* attribute), 263
  - `session` (*gitlab.Gitlab* attribute), 179
  - `set()` (*gitlab.mixins.SetMixin* method), 449
  - `set()` (*gitlab.v4.objects.FeatureManager* method), 197
  - `set_approvers()` (*gitlab.v4.objects.ProjectApprovalManager* method), 269
  - `set_approvers()` (*gitlab.v4.objects.ProjectMergeRequestApprovalManager* method), 338
  - `set_license()` (*gitlab.Gitlab* method), 179
  - `SetMixin` (class in *gitlab.mixins*), 449
  - `settings` (*gitlab.Gitlab* attribute), 180
  - `share()` (*gitlab.v4.objects.Group* method), 205
  - `share()` (*gitlab.v4.objects.Project* method), 263
  - `shared_projects` (*gitlab.v4.objects.Group* attribute), 205
  - `SharedProject` (class in *gitlab.v4.objects*), 398
  - `SharedProjectManager` (class in *gitlab.v4.objects*), 398
  - `sidekiq` (*gitlab.Gitlab* attribute), 180
  - `SidekiqManager` (class in *gitlab.v4.objects*), 398
  - `signature()` (*gitlab.v4.objects.ProjectCommit* method), 282
  - `snapshot()` (*gitlab.v4.objects.Project* method), 263
  - `Snippet` (class in *gitlab.v4.objects*), 400
  - `SnippetManager` (class in *gitlab.v4.objects*), 400
  - `snippets` (*gitlab.Gitlab* attribute), 180
  - `snippets` (*gitlab.v4.objects.Project* attribute), 264
  - `ssl_verify` (*gitlab.Gitlab* attribute), 180
  - `star()` (*gitlab.v4.objects.Project* method), 264
  - `starred_projects` (*gitlab.v4.objects.User* attribute), 407
  - `StarredProject` (class in *gitlab.v4.objects*), 402
  - `StarredProjectManager` (class in *gitlab.v4.objects*), 402
  - `status` (*gitlab.v4.objects.CurrentUser* attribute), 190
  - `status` (*gitlab.v4.objects.User* attribute), 407
  - `status()` (*gitlab.v4.objects.GeoNode* method), 199

status() (*gitlab.v4.objects.GeoNodeManager* method), 201  
 statuses (*gitlab.v4.objects.ProjectCommit* attribute), 282  
 stop() (*gitlab.v4.objects.ProjectEnvironment* method), 292  
 storage (*gitlab.v4.objects.Project* attribute), 264  
 subgroups (*gitlab.v4.objects.Group* attribute), 205  
 SubscribableMixin (class in *gitlab.mixins*), 450  
 subscribe() (*gitlab.mixins.SubscribableMixin* method), 450

## T

tags (*gitlab.v4.objects.Project* attribute), 264  
 tags (*gitlab.v4.objects.ProjectRegistryRepository* attribute), 371  
 take\_ownership() (*gitlab.v4.objects.ProjectPipelineSchedule* method), 363  
 test\_report (*gitlab.v4.objects.ProjectPipeline* attribute), 360  
 test\_report\_summary (*gitlab.v4.objects.ProjectPipeline* attribute), 360  
 time\_estimate() (*gitlab.mixins.TimeTrackingMixin* method), 451  
 time\_stats() (*gitlab.mixins.TimeTrackingMixin* method), 451  
 timeout (*gitlab.Gitlab* attribute), 180  
 TimeTrackingMixin (class in *gitlab.mixins*), 450  
 to\_json() (*gitlab.base.RESTObject* method), 423  
 Todo (class in *gitlab.v4.objects*), 403  
 todo() (*gitlab.mixins.TODOMixin* method), 452  
 TodoManager (class in *gitlab.v4.objects*), 403  
 TODOMixin (class in *gitlab.mixins*), 452  
 todos (*gitlab.Gitlab* attribute), 180  
 Topic (class in *gitlab.v4.objects*), 404  
 TopicManager (class in *gitlab.v4.objects*), 404  
 topics (*gitlab.Gitlab* attribute), 180  
 total (*gitlab.base.RESTObjectList* property), 424  
 total (*gitlab.GitlabList* property), 181  
 total\_pages (*gitlab.base.RESTObjectList* property), 424  
 total\_pages (*gitlab.GitlabList* property), 181  
 trace() (*gitlab.v4.objects.ProjectJob* method), 320  
 transfer() (*gitlab.v4.objects.Group* method), 205  
 transfer() (*gitlab.v4.objects.Project* method), 264  
 transfer\_project() (*gitlab.v4.objects.Group* method), 206  
 transfer\_project() (*gitlab.v4.objects.Project* method), 264  
 trigger\_pipeline() (*gitlab.v4.objects.Project* method), 264  
 triggers (*gitlab.v4.objects.Project* attribute), 265

## U

unapprove() (*gitlab.v4.objects.ProjectMergeRequest* method), 336  
 unarchive() (*gitlab.v4.objects.Project* method), 265  
 unban() (*gitlab.v4.objects.User* method), 407  
 unblock() (*gitlab.v4.objects.User* method), 408  
 unfollow() (*gitlab.v4.objects.User* method), 408  
 unshare() (*gitlab.v4.objects.Group* method), 206  
 unshare() (*gitlab.v4.objects.Project* method), 265  
 unstar() (*gitlab.v4.objects.Project* method), 265  
 unsubscribe() (*gitlab.mixins.SubscribableMixin* method), 450  
 update() (*gitlab.mixins.UpdateMixin* method), 452  
 update() (*gitlab.v4.objects.ApplicationAppearanceManager* method), 183  
 update() (*gitlab.v4.objects.ApplicationSettingsManager* method), 186  
 update() (*gitlab.v4.objects.GroupLabelManager* method), 233  
 update() (*gitlab.v4.objects.ProjectFileManager* method), 298  
 update() (*gitlab.v4.objects.ProjectLabelManager* method), 324  
 update\_submodule() (*gitlab.v4.objects.RepositoryMixin* method), 394  
 UpdateMixin (class in *gitlab.mixins*), 452  
 upload() (*gitlab.v4.objects.GenericPackageManager* method), 199  
 upload() (*gitlab.v4.objects.Project* method), 266  
 url (*gitlab.Gitlab* property), 180  
 User (class in *gitlab.v4.objects*), 405  
 user (*gitlab.Gitlab* attribute), 180  
 user\_activities (*gitlab.Gitlab* attribute), 180  
 user\_agent\_detail() (*gitlab.mixins.UserAgentDetailMixin* method), 452  
 UserActivities (class in *gitlab.v4.objects*), 408  
 UserActivitiesManager (class in *gitlab.v4.objects*), 409  
 UserAgentDetailMixin (class in *gitlab.mixins*), 452  
 UserCustomAttribute (class in *gitlab.v4.objects*), 409  
 UserCustomAttributeManager (class in *gitlab.v4.objects*), 409  
 UserEmail (class in *gitlab.v4.objects*), 410  
 UserEmailManager (class in *gitlab.v4.objects*), 410  
 UserEvent (class in *gitlab.v4.objects*), 410  
 UserEventManager (class in *gitlab.v4.objects*), 411  
 UserGPGKey (class in *gitlab.v4.objects*), 411  
 UserGPGKeyManager (class in *gitlab.v4.objects*), 411  
 UserIdentityProviderManager (class in *gitlab.v4.objects*), 412  
 UserImpersonationToken (class in *gitlab.v4.objects*), 412

UserImpersonationTokenManager (class in *gitlab.v4.objects*), 413  
UserKey (class in *gitlab.v4.objects*), 413  
UserKeyManager (class in *gitlab.v4.objects*), 414  
UserManager (class in *gitlab.v4.objects*), 414  
UserMembership (class in *gitlab.v4.objects*), 417  
UserMembershipManager (class in *gitlab.v4.objects*), 417  
UserPersonalAccessToken (class in *gitlab.v4.objects*), 417  
UserPersonalAccessTokenManager (class in *gitlab.v4.objects*), 418  
UserProject (class in *gitlab.v4.objects*), 418  
UserProjectManager (class in *gitlab.v4.objects*), 418  
USERS (*gitlab.const.SearchScope* attribute), 427  
users (*gitlab.Gitlab* attribute), 180  
users (*gitlab.v4.objects.Project* attribute), 266  
UserStatus (class in *gitlab.v4.objects*), 420  
UserStatusManager (class in *gitlab.v4.objects*), 420

## V

validate() (*gitlab.v4.objects.CiLintManager* method), 189  
validate() (*gitlab.v4.objects.ProjectCiLintManager* method), 278  
Variable (class in *gitlab.v4.objects*), 421  
VariableManager (class in *gitlab.v4.objects*), 421  
variables (*gitlab.Gitlab* attribute), 180  
variables (*gitlab.v4.objects.Group* attribute), 206  
variables (*gitlab.v4.objects.Project* attribute), 266  
variables (*gitlab.v4.objects.ProjectPipeline* attribute), 360  
variables (*gitlab.v4.objects.ProjectPipelineSchedule* attribute), 363  
verify() (*gitlab.v4.objects.RunnerManager* method), 397  
version() (*gitlab.Gitlab* method), 180  
VerticalHelpFormatter (class in *gitlab.cli*), 424  
Visibility (class in *gitlab.const*), 427

## W

warn() (in module *gitlab.utils*), 454  
WATCH (*gitlab.const.NotificationLevel* attribute), 426  
WIKI\_BLOBS (*gitlab.const.SearchScope* attribute), 427  
wikis (*gitlab.v4.objects.Group* attribute), 206  
wikis (*gitlab.v4.objects.Project* attribute), 266