python-gitlab Documentation

Release 8.4.0

Gauvain Pocentek, Mika Mäenpää

Jun 11, 2026

TABLE OF CONTENTS

1 Features **3**

2 Installation **5**

3 Using the docker images **7**
- 3.1 Usage inside GitLab CI . . . 7
- 3.2 Building the image . . . 8

4 Bug reports **9**

5 Gitter Community Chat **11**

6 Documentation **13**
- 6.1 Build the docs . . . 13

7 Contributing **15**
- 7.1 Using the CLI . . . 15
- 7.2 Using the REST API . . . 20
- 7.3 Advanced usage . . . 28
- 7.4 Using the GraphQL API (beta) . . . 31
- 7.5 CLI examples . . . 32
- 7.6 API examples . . . 37
- 7.7 API reference (`gitlab` package) . . . 169
- 7.8 CLI reference (`gitlab` command) . . . 463
- 7.9 Lower-level APIs . . . 824
- 7.10 CHANGELOG . . . 826
- 7.11 Release notes . . . 932
- 7.12 FAQ . . . 935

8 Indices and tables **939**

Python Module Index **941**

Index **943**

`python-gitlab` is a Python package providing access to the GitLab APIs.

It includes a client for GitLab's v4 REST API, synchronous and asynchronous GraphQL API clients, as well as a CLI tool (`gitlab`) wrapping REST API endpoints.

CHAPTER

ONE

FEATURES

`python-gitlab` enables you to:

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

CHAPTER

TWO

INSTALLATION

As of 7.0.0, `python-gitlab` is compatible with Python 3.10+.

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 and GitLab.com, 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
```

CHAPTER

THREE

USING THE DOCKER IMAGES

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

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

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

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

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

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

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

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

You can also mount your own config file:

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

3.1 Usage inside GitLab CI

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

```
Job Name:
   image:
      name: registry.gitlab.com/python-gitlab/python-gitlab:latest
      entrypoint: [""]
```

(continues on next page)

(continued from previous page)

```
  before_script:
    gitlab --version
  script:
    gitlab <command>
```

3.2 Building the image

To build your own image from this repository, run:

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

Run your own image:

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

Build a Debian slim-based image:

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

CHAPTER

FOUR

BUG REPORTS

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

CHAPTER

FIVE

GITTER COMMUNITY CHAT

We have a gitter 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!

CHAPTER

SIX

DOCUMENTATION

The full documentation for CLI and API is available on readthedocs.

6.1 Build the docs

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

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

CHAPTER

SEVEN

CONTRIBUTING

For guidelines for contributing to `python-gitlab`, refer to CONTRIBUTING.rst.

7.1 Using the CLI

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

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

7.1.1 Configuration

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

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

With no configuration provided, `gitlab` will default to unauthenticated requests against GitLab.com.

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

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

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

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

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

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

Configuration files

`gitlab` looks up 3 configuration files by default:

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

`/etc/python-gitlab.cfg`
System-wide configuration file

`~/.python-gitlab.cfg`
User configuration file

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

> **Warning**
>
> If the `PYTHON_GITLAB_CFG` environment variable is defined and the target file exists, it will be the only configuration file parsed by `gitlab`.
>
> If the environment variable is defined and the target file cannot be accessed, `gitlab` will fail explicitly.

Configuration file format

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

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

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

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

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

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

Table 1: Global options

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

You must define the url in each GitLab server section.

Warning

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

A URL that redirects using 301/302 (rather than 307/308) will most likely cause malformed POST and PUT requests.

python-gitlab will therefore raise a RedirectionError when it encounters a redirect which it believes will cause such an error, to avoid confusion between successful GET and failing POST/PUT requests on the same instance.

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

We recommend that you use *Credential helpers* to securely store your tokens.

Table 2: GitLab server options

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

Credential helpers

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

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

Example for a keyring helper:

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

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

Example for a pass helper with a wrapper script:

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

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

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

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

7.1.2 CLI

Objects and actions

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

```
$ gitlab project list
```

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

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

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

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

Optional arguments

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

`--verbose, -v`
: Outputs detail about retrieved objects. Available for legacy (default) output only.

--config-file, -c
Path to a configuration file.

--gitlab, -g
ID of a GitLab server defined in the configuration file.

--output, -o
Output format. Defaults to a custom format. Can also be `yaml` or `json`.

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

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

Example:

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

Reading values from files

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

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

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

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

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

7.1.3 Enabling shell autocompletion

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

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

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

Bash

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

tcsh

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

fish

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

Zsh

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

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

```
autoload -U bashcompinit
bashcompinit
```

Afterwards you can enable completion for gitlab:

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

7.2 Using the REST API

python-gitlab currently only supports v4 of the GitLab REST API.

7.2.1 `gitlab.Gitlab` class

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

> **Hint**
>
> You can use different types of tokens for authenticated requests against the GitLab API. You will most likely want to use a resource (project/group) access token or a personal access token.
>
> For the full list of available options and how to obtain these tokens, please see https://docs.gitlab.com/api/rest/authentication/.

```
import gitlab

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

(continues on next page)

(continued from previous page)

```
# 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='JVNSESs8EwWRx5yDxM5q')

# private token or personal token authentication (self-hosted GitLab instance)
gl = gitlab.Gitlab(url='https://gitlab.example.com', private_token='JVNSESs8EwWRx5yDxM5q
↪')

# 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/ci/jobs/ci_job_token
import os
gl = gitlab.Gitlab('https://gitlab.example.com', job_token=os.environ['CI_JOB_TOKEN'])

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

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

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

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

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

See the *Configuration* section for more information about configuration files.

Warning

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

A URL that redirects using 301/302 (rather than 307/308) will most likely cause malformed POST and PUT requests.

python-gitlab will therefore raise a `RedirectionError` when it encounters a redirect which it believes will cause such an error, to avoid confusion between successful GET and failing POST/PUT requests on the same instance.

Note on password authentication

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

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

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

7.2.2 Managers

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

Examples:

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

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

Warning

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

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

> **Note**
>
> python-gitlab attempts to sync the required, optional, and mutually exclusive attributes for resource creation and update with the upstream API.
>
> You are encouraged to follow upstream API documentation for each resource to find these - each resource documented here links to the corresponding upstream resource documentation at the top of the page.

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

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

Some objects also provide managers to access related GitLab resources:

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

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

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

7.2.3 Conflicting Parameters

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

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

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

7.2.4 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(get_all=True, state='opened')
for issue in issues:
   issue.my_super_awesome_feature_flag = "random_value"
   issue.save()
```

As a dictionary

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

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

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

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

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

Hint

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

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

As JSON

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

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

(continues on next page)

(continued from previous page)

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

7.2.5 Base types

The `gitlab` package provides some base types.

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

7.2.6 Lazy objects

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

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

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

7.2.7 `head()` methods

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

> **Note**
>
> In some cases, GitLab may omit specific headers. See more in the *Pagination* section.

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

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

7.2.8 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(get_all=True)
```

Reference: https://docs.gitlab.com/api/rest/#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/user/gitlab_com/index#pagination-response-headers

Note

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

Note

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

7.2.9 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')
```

Warning

When using `sudo`, its usage is not remembered. If you use `sudo` to retrieve an object and then later use `save()` to modify the object, it will not use `sudo`. You should use `save(sudo='user1')` if you want to perform subsequent actions as the user.

Updating with sudo

An example of how to `get` an object (using `sudo`), modify the object, and then `save` the object (using `sudo`):

```
group = gl.groups.get('example-group')
notification_setting = group.notificationsettings.get(sudo='user1')
notification_setting.level = gitlab.const.NOTIFICATION_LEVEL_GLOBAL
# Must use 'sudo' again when doing the save.
notification_setting.save(sudo='user1')
```

7.2.10 Logging

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

```
import os
import gitlab

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

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

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

7.2.11 Attributes in updated objects

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

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

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

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

7.3 Advanced usage

7.3.1 Using a custom session

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

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

```
import gitlab
import requests

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

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

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

7.3.2 Context manager

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

```
with gitlab.Gitlab(host, token) as gl:
    gl.statistics.get()
```

Warning

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

7.3.3 netrc authentication

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

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

```
import gitlab
import requests

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

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

7.3.4 Proxy configuration

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

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

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

7.3.5 SSL certificate verification

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

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

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

7.3.6 Client side certificate

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

```
import gitlab
import requests

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

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

7.3.7 Rate limits

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

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

```
import gitlab
import requests

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

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

```
import gitlab
import requests

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

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

7.3.8 Transient errors

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

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

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

```
import gitlab
import requests

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

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

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

7.3.9 Timeout

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

```
import gitlab

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

7.3.10 Typing

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

```
from typing import TYPE_CHECKING

import gitlab

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

(continues on next page)

(continued from previous page)

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

7.3.11 Per request HTTP headers override

The `extra_headers` keyword argument can be used to add and override the HTTP headers for a specific request. For example, it can be used do add `Range` header to download a part of artifacts archive:

```
import gitlab

gl = gitlab.Gitlab(url, token)
project = gl.projects.get(1)
job = project.jobs.get(123)

artifacts = job.artifacts(extra_headers={"Range": "bytes=0-9"})
```

7.4 Using the GraphQL API (beta)

python-gitlab provides basic support for executing GraphQL queries and mutations, providing both a synchronous and asynchronous client.

Danger

The GraphQL client is experimental and only provides basic support. It does not currently support pagination, obey rate limits, or attempt complex retries. You can use it to build simple queries and mutations.

It is currently unstable and its implementation may change. You can expect a more mature client in one of the upcoming versions.

7.4.1 The `gitlab.GraphQL` and `gitlab.AsyncGraphQL` classes

As with the REST client, you connect to a GitLab instance by creating a `gitlab.GraphQL` (for synchronous code) or `gitlab.AsyncGraphQL` instance (for asynchronous code):

```
import gitlab

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

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

# personal access token or OAuth2 token authentication (GitLab.com)
gq = gitlab.GraphQL(token='glpat-JVNSESs8EwWRx5yDxM5q')

# personal access token or OAuth2 token authentication (self-hosted GitLab instance)
gq = gitlab.GraphQL('https://gitlab.example.com', token='glpat-JVNSESs8EwWRx5yDxM5q')

# or the async equivalents
```

(continues on next page)

(continued from previous page)

```
async_gq = gitlab.AsyncGraphQL()
async_gq = gitlab.AsyncGraphQL('https://gitlab.example.com')
async_gq = gitlab.AsyncGraphQL(token='glpat-JVNSESs8EwWRx5yDxM5q')
async_gq = gitlab.AsyncGraphQL('https://gitlab.example.com', token='glpat-
↪JVNSESs8EwWRx5yDxM5q')
```

7.4.2 Sending queries

Get the result of a query:

```
query = """
{
    currentUser {
        name
    }
}
"""

result = gq.execute(query)
```

Get the result of a query using the async client:

```
query = """
{
    currentUser {
        name
    }
}
"""

result = await async_gq.execute(query)
```

7.5 CLI examples

See also

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

7.5.1 CI Lint

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

Lint a CI YAML configuration from a string:

Note

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

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

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

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

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

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

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

7.5.2 Project CI Lint

Lint a project's CI YAML configuration:

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

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

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

Lint a project's current CI YAML configuration:

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

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

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

7.5.3 Projects

List the projects (paginated):

```
$ gitlab project list
```

List all the projects:

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

List all projects of a group:

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

List all projects of a group and its subgroups:

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

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

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

Get a specific project (id 2):

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

7.5.4 Users

Get a specific user by id:

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

Create a user impersonation token (admin-only):

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

7.5.5 Deploy tokens

Create a deploy token for a project:

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

List deploy tokens for a group:

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

7.5.6 Personal access tokens

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

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

Revoke a personal access token by id:

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

Revoke the personal access token currently used:

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

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

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

7.5.7 Resource access tokens

Create a project access token:

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

List project access tokens:

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

Revoke a project access token:

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

Create a group access token:

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

List group access tokens:

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

Revoke a group access token:

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

7.5.8 Packages

List packages for a project:

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

List packages for a group:

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

Get a specific project package by id:

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

Delete a specific project package by id:

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

Upload a generic package to a project:

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

Download a project's generic package:

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

7.5.9 Issues

Get a list of issues for this project:

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

7.5.10 Snippets

Delete a snippet (id 3):

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

Update a snippet:

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

Create a snippet:

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

7.5.11 Commits

Get a specific project commit by its SHA id:

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

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

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

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

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

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

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

7.5.12 Artifacts

Download the artifacts zip archive of a job:

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

7.5.13 Runners

List owned runners:

```
$ gitlab runner list
```

List owned runners with a filter:

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

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

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

Get a runner's details:

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

7.5.14 Other

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

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

List values are comma-separated:

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

7.6 API examples

7.6.1 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/api/access_requests

Examples

List access requests from projects and groups:

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

Create an access request:

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

Approve an access request:

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

Deny (delete) an access request:

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

7.6.2 Appearance

Reference

- v4 API:
  - *gitlab.v4.objects.ApplicationAppearance*
  - *gitlab.v4.objects.ApplicationAppearanceManager*
  - *gitlab.Gitlab.appearance*
- GitLab API: https://docs.gitlab.com/api/appearance

Examples

Get the appearance:

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

Update the appearance:

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

7.6.3 Applications

Reference

- v4 API:
  - gitlab.v4.objects.Applications
  - *gitlab.v4.objects.ApplicationManager*

  - *gitlab.Gitlab.applications*

- GitLab API: https://docs.gitlab.com/api/applications

Examples

List all OAuth applications:

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

Create an application:

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

Delete an applications:

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

7.6.4 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/api/emoji_reactions/

Examples

List emojis for a resource:

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

Get a single emoji:

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

Add (create) an emoji:

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

Delete an emoji:

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

7.6.5 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/api/group_badges
  - https://docs.gitlab.com/api/project_badges

Examples

List badges:

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

Get a badge:

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

Create a badge:

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

Update a badge:

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

Delete a badge:

```
badge.delete()
```

Render a badge (preview the generate URLs):

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

7.6.6 Branches

References

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

Examples

Get the list of branches for a repository:

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

Get a single repository branch:

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

Create a repository branch:

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

Delete a repository branch:

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

Delete the merged branches for a project:

```
project.delete_merged_branches()
```

To manage protected branches, see *Protected branches*.

7.6.7 Migrations (Bulk Imports)

References

- v4 API:
  - *gitlab.v4.objects.BulkImport*
  - *gitlab.v4.objects.BulkImportManager*

  - *gitlab.Gitlab.bulk_imports*
  - *gitlab.v4.objects.BulkImportAllEntity*
  - *gitlab.v4.objects.BulkImportAllEntityManager*
  - *gitlab.Gitlab.bulk_import_entities*
  - *gitlab.v4.objects.BulkImportEntity*
  - *gitlab.v4.objects.BulkImportEntityManager*
  - *gitlab.v4.objects.BulkImport.entities*
- GitLab API: https://docs.gitlab.com/api/bulk_imports

Examples

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

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

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

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

List all migrations:

```
gl.bulk_imports.list(get_all=True)
```

List the entities of all migrations:

```
gl.bulk_import_entities.list(get_all=True)
```

Get a single migration by ID:

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

List the entities of a single migration:

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

Get a single entity of a migration by ID:

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

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

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

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

7.6.8 Broadcast messages

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

References

- v4 API:
  - *gitlab.v4.objects.BroadcastMessage*
  - *gitlab.v4.objects.BroadcastMessageManager*
  - *gitlab.Gitlab.broadcastmessages*
- GitLab API: https://docs.gitlab.com/api/broadcast_messages

Examples

List the messages:

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

Get a single message:

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

Create a message:

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

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

Update a message:

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

Delete a message:

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

7.6.9 CI Lint

Reference

- v4 API:
  - *gitlab.v4.objects.CiLint*
  - *gitlab.v4.objects.CiLintManager*
  - *gitlab.Gitlab.ci_lint*
  - *gitlab.v4.objects.ProjectCiLint*
  - *gitlab.v4.objects.ProjectCiLintManager*
  - *gitlab.v4.objects.Project.ci_lint*
- GitLab API: https://docs.gitlab.com/api/lint

Examples

Lint a CI YAML configuration:

```
gitlab_ci_yml = """.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_yml})

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 project's CI configuration from a specific branch or tag:

```
lint_result = project.ci_lint.get(content_ref="main")
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 project's CI configuration with dry run simulation:

```
lint_result = project.ci_lint.get(dry_run=True, dry_run_ref="develop")
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_yml})
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_yml})

# 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_yml})

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

7.6.10 Cluster agents

You can list and manage project cluster agents with the GitLab agent for Kubernetes.

> **Warning**
>
> Check the GitLab API documentation linked below for project permissions required to access specific cluster agent endpoints.

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectClusterAgent*
  - *gitlab.v4.objects.ProjectClusterAgentManager*
  - *gitlab.v4.objects.Project.cluster_agents*
- GitLab API: https://docs.gitlab.com/api/cluster_agents

Examples

List cluster agents for a project:

```
cluster_agents = project.cluster_agents.list(get_all=True)
```

Register a cluster agent with a project:

```
cluster_agent = project.cluster_agents.create({"name": "Agent 1"})
```

Retrieve a specific cluster agent for a project:

```
cluster_agent = project.cluster_agents.get(cluster_agent.id)
```

Delete a registered cluster agent from a project:

```
cluster_agent = project.cluster_agents.delete(cluster_agent.id)
# or
cluster.delete()
```

7.6.11 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(get_all=True)
```

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

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

List all commits for a project (see *Pagination*) on all branches:

commits = project.commits.list(get_all=True)

Create a commit:

```
# See https://docs.gitlab.com/api/commits#create-a-commit-with-multiple-files-and-actions
# for actions detail
data = {
    'branch': 'main',
    'commit_message': 'blah blah blah',
    'actions': [
        {
```

(continues on next page)

(continued from previous page)

```
            'action': 'create',
            'file_path': 'README.rst',
            'content': open('path/to/file.rst').read(),
        },
        {
            # Binary files need to be base64 encoded
            'action': 'create',
            'file_path': 'logo.png',
            'content': base64.b64encode(open('logo.png', mode='r+b').read()).decode(),
            'encoding': 'base64',
        }
    ]
}

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

Get a commit detail:

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

Get the diff for a commit:

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

Cherry-pick a commit into another branch:

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

Revert a commit on a given branch:

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

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

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

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

```
commit.signature()
```

List the merge requests related to a commit:

```
commit.merge_requests()
```

Commit comments

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectCommitComment*
  - *gitlab.v4.objects.ProjectCommitCommentManager*

  - *gitlab.v4.objects.ProjectCommit.comments*

- GitLab API: https://docs.gitlab.com/api/commits

Examples

Get the comments for a commit:

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

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/api/commits

Examples

List the statuses for a commit:

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

Change the status of a commit:

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

7.6.12 Deploy keys

Deploy keys

Reference

- v4 API:
  - *gitlab.v4.objects.DeployKey*
  - *gitlab.v4.objects.DeployKeyManager*
  - *gitlab.Gitlab.deploykeys*
- GitLab API: https://docs.gitlab.com/api/deploy_keys

Examples

Add an instance-wide deploy key (requires admin access):

```
keys = gl.deploykeys.create({'title': 'instance key', 'key': INSTANCE_KEY})
```

List all deploy keys:

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

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/api/deploy_keys

Examples

List keys for a project:

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

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, get_all=True)
# or
key.delete()
```

Enable a deploy key for a project:

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

Disable a deploy key for a project:

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

7.6.13 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/api/deploy_tokens

Examples

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

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

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/api/deploy_tokens#project-deploy-tokens

Examples

List the deploy tokens for a project:

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

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/api/deploy_tokens#group-deploy-tokens

Examples

List the deploy tokens for a group:

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

Get a deploy token for a group by id:

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

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

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

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

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

Warning

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

Remove a deploy token from the group:

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

7.6.14 Deployments

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectDeployment*
  - *gitlab.v4.objects.ProjectDeploymentManager*
  - *gitlab.v4.objects.Project.deployments*
- GitLab API: https://docs.gitlab.com/api/deployments

Examples

List deployments for a project:

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

Get a single deployment:

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

Create a new deployment:

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

Update a deployment:

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

Approve a deployment:

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

Reject a deployment:

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

Merge requests associated with a deployment

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectDeploymentMergeRequest*
  - *gitlab.v4.objects.ProjectDeploymentMergeRequestManager*
  - *gitlab.v4.objects.ProjectDeployment.mergerequests*
- GitLab API: https://docs.gitlab.com/api/deployments#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(get_all=True)
```

7.6.15 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/api/discussions

Examples

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

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

Get a single discussion:

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

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

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

Note

The notes are dicts, not objects.

You can add notes to existing discussions:

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

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

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

Create a new discussion:

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

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

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

7.6.16 Draft Notes

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

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectMergeRequestDraftNote*
  - *gitlab.v4.objects.ProjectMergeRequestDraftNoteManager*
  - *gitlab.v4.objects.ProjectMergeRequest.draft_notes*
- GitLab API: https://docs.gitlab.com/api/draft_notes

Examples

List all draft notes for a merge request:

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

Get a draft note for a merge request by ID:

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

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

Create a draft note for a merge request:

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

Update an existing draft note:

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

Delete an existing draft note:

```
draft_note.delete()
```

Publish an existing draft note:

```
draft_note.publish()
```

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

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

7.6.17 Environments

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectEnvironment*
  - *gitlab.v4.objects.ProjectEnvironmentManager*
  - *gitlab.v4.objects.Project.environments*
- GitLab API: https://docs.gitlab.com/api/environments

Examples

List environments for a project:

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

Create an environment for a project:

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

Retrieve a specific environment for a project:

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

Update an environment for a project:

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

Delete an environment for a project:

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

Stop an environment:

```
environment.stop()
```

To manage protected environments, see *Protected environments*.

7.6.18 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/api/events/

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(get_all=True)
```

List the issue events on a project:

```
events = project.events.list(target_type='issue', get_all=True)
```

List the user events:

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

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/api/resource_state_events

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_all=True)
```

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_all=True)
```

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

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

7.6.19 Epics

Epics

Reference

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

Examples

List the epics for a group:

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

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/api/epic_issues (EE feature)

Examples

List the issues associated with an issue:

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

Associate an issue with an epic:

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

Move an issue in the list:

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

Delete an issue association:

```
ei.delete()
```

7.6.20 GitLab Development Feature Flags

> **Note**
>
> This API is for managing GitLab's internal development feature flags and requires administrator access. For project-level feature flags, see *Project Feature Flags*.

Reference

- v4 API:
  - *gitlab.v4.objects.Feature*
  - *gitlab.v4.objects.FeatureManager*
  - *gitlab.Gitlab.features*
- GitLab API: https://docs.gitlab.com/api/features

Examples

List features:

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

Create or set a feature:

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

Delete a feature:

```
feature.delete()
```

7.6.21 Geo nodes

Reference

- v4 API:
  - *gitlab.v4.objects.GeoNode*
  - *gitlab.v4.objects.GeoNodeManager*
  - *gitlab.Gitlab.geonodes*
- GitLab API: https://docs.gitlab.com/api/geo_nodes (EE feature)

Examples

List the geo nodes:

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

Get the status of all the nodes:

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

Get a specific node and its status:

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

Edit a node configuration:

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

Delete a node:

```
node.delete()
```

List the sync failure on the current node:

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

7.6.22 Groups

Groups

Reference

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

Examples

List the groups:

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

Get a group's detail:

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

List a group's projects:

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

List a group's shared projects:

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

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

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

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

Create a group:

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

Warning

On GitLab.com, creating top-level groups is currently not permitted using the API. You can only use the API to create subgroups.

Create a subgroup under an existing group:

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

Update a group:

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

Set the avatar image for a group:

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

Remove the avatar image for a group:

```
group.avatar = ""
group.save()
```

Remove a group:

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

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

```
group.restore()
```

Share/unshare the group with a group:

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

Import / Export

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

Reference

- v4 API:
  - *gitlab.v4.objects.GroupExport*
  - *gitlab.v4.objects.GroupExportManager*
  - *gitlab.v4.objects.Group.exports*
  - *gitlab.v4.objects.GroupImport*
  - *gitlab.v4.objects.GroupImportManager*
  - *gitlab.v4.objects.Group.imports*
  - *gitlab.v4.objects.GroupManager.import_group*
- GitLab API: https://docs.gitlab.com/api/group_import_export

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(get_all=True)
```

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(get_all=True)
```

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(get_all=True)
```

> **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/api/custom_attributes

Examples

List custom attributes for a group:

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

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'}, get_all=True)
```

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/api/members

Billable group members are only available in GitLab EE.

Examples

List only direct group members:

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

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(get_all=True)
```

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(get_all=True)
```

LDAP group links

Add an LDAP group link to an existing GitLab group:

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

List a group's LDAP group links:

```
group.ldap_group_links.list(get_all=True)
```

Remove a link:

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

Sync the LDAP groups:

```
group.ldap_sync()
```

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

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

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

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

SAML group links

Add a SAML group link to an existing GitLab group:

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

List a group's SAML group links:

```
group.saml_group_links.list(get_all=True)
```

Get a SAML group link:

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

Remove a link:

```
saml_link.delete()
```

Groups hooks

Reference

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

Examples

List the group hooks:

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

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()
```

Test a group hook:

```
hook.test("push_events")
```

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/api/groups#push-rules

Examples

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

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

Get group push rules:

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

Edit group push rules:

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

Delete group push rules:

```
pr.delete()
```

Group Service Account

Reference

- v4 API:
  - *gitlab.v4.objects.GroupServiceAccount*
  - *gitlab.v4.objects.GroupServiceAccountManager*
  - gitlab.v4.objects.Group.serviceaccounts
- GitLab API: https://docs.gitlab.com/api/groups#service-accounts

Examples

Create group service account (only allowed at top level group):

```
group.serviceaccount.create({'name': 'group-service-account', 'username': 'group-service-
↪account'})
```

7.6.23 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/api/group_access_tokens

Examples

List group access tokens:

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

Get a group access token by id:

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

Create group access token:

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

Revoke a group access token:

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

Rotate a group access token and retrieve its new value:

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

Self-Rotate the group access token you are using to authenticate the request and retrieve its new value:

```
token = group.access_tokens.get(42, lazy=True)
token.rotate(self_rotate=True)
print(token.token)
```

7.6.24 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/api/invitations

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(get_all=True)
```

Warning

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

Update an invitation:

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

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

Delete an invitation:

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

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

7.6.25 Issues

Reported issues

Reference

- v4 API:
  - *gitlab.v4.objects.Issue*
  - *gitlab.v4.objects.IssueManager*
  - *gitlab.Gitlab.issues*
- GitLab API: https://docs.gitlab.com/api/issues

Examples

List the issues:

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

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

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(get_all=False)[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/api/issues

Examples

List the group issues:

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

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(get_all=False)[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/api/issues

Examples

List the project issues:

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

Get a project issue:

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

Create a new issue:

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

Update an issue:

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

Close / reopen an issue:

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

Delete an issue (admin or project owner only):

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

Assign the issues:

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

Note

The Gitlab API explicitly references that the *assignee_id* field is deprecated, so using a list of user IDs for *assignee_ids* is how to assign an issue to a user(s).

Subscribe / unsubscribe from an issue:

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

Move an issue to another project:

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

Reorder an issue on a board:

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

Make an issue as todo:

```
issue.todo()
```

Get time tracking stats:

```
issue.time_stats()
```

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

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

Set a time estimate for an issue:

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

Reset a time estimate for an issue:

```
issue.reset_time_estimate()
```

Add spent time for an issue:

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

Reset spent time for an issue:

```
issue.reset_spent_time()
```

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

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

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

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

Get the merge requests related to an issue:

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

Get the list of participants:

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

Get the list of iteration events:

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

Get the list of weight events:

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

Issue links

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectIssueLink*
  - *gitlab.v4.objects.ProjectIssueLinkManager*
  - *gitlab.v4.objects.ProjectIssue.links*
- GitLab API: https://docs.gitlab.com/api/issue_links

Examples

List the issues linked to `i1`:

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

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/api/issues_statistics/

Examples

Get statistics of all issues created by the current user:

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

Get statistics of all issues the user has access to:

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

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

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

Get statistics of all issues in a group:

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

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

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

Get statistics of all issues in a project:

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

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

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

7.6.26 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/api/iterations

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(get_all=True)
```

List iterations for a group:

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

Unavailable filters or keyword conflicts:

```
In case you are trying to pass a parameter that collides with a python
keyword (i.e. `in`) or with python-gitlab's internal arguments, you'll have
to use the `query_parameters` argument:

```
group.iterations.list(query_parameters={"in": "title"}, get_all=True)
```
```

7.6.27 CI/CD job token scope

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectJobTokenScope*
  - *gitlab.v4.objects.ProjectJobTokenScopeManager*
  - *gitlab.v4.objects.Project.job_token_scope*
- GitLab API: https://docs.gitlab.com/api/project_job_token_scopes

Examples

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

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

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

Update the job token scope settings:

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

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

Or update the job token scope settings directly:

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

Refresh the current state of job token scope:

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

Get a project's CI/CD job token inbound allowlist:

```
allowlist = scope.allowlist.list(get_all=True)
```

Add a project to the project's inbound allowlist:

```
allowed_project = scope.allowlist.create({"target_project_id": 42})
```

Remove a project from the project's inbound allowlist:

```
allowed_project.delete()
# or directly using a project ID
scope.allowlist.delete(42)
```

Warning

Similar to above, the ID attributes you receive from the create and list APIs are not consistent (in create() the id is returned as `source_project_id` whereas list() returns as `id`). To safely retrieve the ID of the allowlisted project regardless of how the object was created, always use its `.get_id()` method.

Using `.get_id()`:

```
resp = allowlist.create({"target_project_id": 2})
allowlist_id = resp.get_id()

for allowlist in project.allowlist.list(iterator=True):
  allowlist_id == allowlist.get_id()
```

Get a project's CI/CD job token inbound groups allowlist:

```
allowlist = scope.groups_allowlist.list(get_all=True)
```

Add a group to the project's inbound groups allowlist:

```
allowed_group = scope.groups_allowlist.create({"target_group_id": 42})
```

Remove a group from the project's inbound groups allowlist:

```
allowed_group.delete()
# or directly using a Group ID
scope.groups_allowlist.delete(42)
```

> **Warning**
>
> Similar to above, the ID attributes you receive from the create and list APIs are not consistent. To safely retrieve the ID of the allowlisted group regardless of how the object was created, always use its `.get_id()` method.

7.6.28 Keys

Keys

Reference

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

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/OweAM6jA8OjJ/gXs4N5fqUaREEJnz/EyfywfXY")
```

7.6.29 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/api/boards
  - https://docs.gitlab.com/api/group_boards

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_all=True)
```

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/api/boards
  - https://docs.gitlab.com/api/group_boards

Examples

List the issue lists for a board:

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

Get a single list:

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

Create a new list:

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

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

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

Delete a list:

```
b_list.delete()
```

7.6.30 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/api/labels

Examples

List labels for a project:

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

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/api/resource_label_events

Examples

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

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

Get a specific event for a resource:

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

7.6.31 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/api/notification_settings

Examples

Get the notifications settings:

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

Update the notifications settings:

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

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

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

7.6.32 Member Roles

You can configure member roles at the instance-level (admin only), or at group level.

Instance-level member roles

This endpoint requires admin access.

Reference

- v4 API
  - *gitlab.v4.objects.MemberRole*
  - *gitlab.v4.objects.MemberRoleManager*
  - *gitlab.Gitlab.member_roles*
- GitLab API
  - https://docs.gitlab.com/api/member_roles#manage-instance-member-roles

Examples

List member roles:

```
variables = gl.member_roles.list()
```

Create a member role:

```
variable = gl.member_roles.create({'name': 'Custom Role', 'base_access_level': value})
```

Remove a member role:

```
gl.member_roles.delete(member_role_id)
```

Group member role

Reference

- v4 API
  - *gitlab.v4.objects.GroupMemberRole*

  - *gitlab.v4.objects.GroupMemberRoleManager*
  - *gitlab.v4.objects.Group.member_roles*
- GitLab API
  - https://docs.gitlab.com/api/member_roles#manage-group-member-roles

Examples

List member roles:

```
member_roles = group.member_roles.list()
```

Create a member role:

```
member_roles = group.member_roles.create({'name': 'Custom Role', 'base_access_level':␣
↪value})
```

Remove a member role:

```
gl.member_roles.delete(member_role_id)
```

7.6.33 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/api/merge_trains

Examples

List merge trains for a project:

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

List active merge trains for a project:

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

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

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

7.6.34 Merge requests

You can use merge requests to notify a project that a branch is ready for merging. The owner of the target projet 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/api/merge_requests

Examples

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

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

List the merge requests available on the GitLab server:

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

List the merge requests for a group:

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

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(get_all=False)[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/api/merge_requests

Examples

List MRs for a project:

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

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/api/merge_requests#list-merge-requests

For example:

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

Get a single MR:

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

Get MR reviewer details:

```
mr = project.mergerequests.get(mr_iid)
reviewers = mr.reviewer_details.list(get_all=True)
```

Create a MR:

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

# Use a project MR description template
mr_description_template = project.merge_request_templates.get("Default")
mr = project.mergerequests.create({'source_branch': 'cool_feature',
                                   'target_branch': 'main',
                                   'title': 'merge cool feature',
                                   'description': mr_description_template.content})
```

Update a MR:

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

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

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

Delete a MR:

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

Accept a MR:

```
mr.merge()
```

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

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

Cancel a MR from merging when the pipeline succeeds:

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

List commits of a MR:

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

List the changes of a MR:

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

List issues related to this merge request:

```
related_issues = mr.related_issues()
```

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_all=True)
```

Get a diff for a merge request:

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

Get time tracking stats:

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

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

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

Set a time estimate for a merge request:

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

Reset a time estimate for a merge request:

```
mr.reset_time_estimate()
```

Add spent time for a merge request:

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

Reset spent time for a merge request:

```
mr.reset_spent_time()
```

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

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

Attempt to rebase an MR:

```
mr.rebase()
```

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

```
mr.reset_approvals()
```

Get status of a rebase for an MR:

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

For more info see: https://docs.gitlab.com/api/merge_requests#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/api/merge_requests#list-mr-pipelines

Examples

List pipelines for a merge request:

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

Create a pipeline for a merge request:

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

7.6.35 Merge request approvals settings

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

Group approval rules

References

- v4 API:
  - *gitlab.v4.objects.GroupApprovalRule*
  - *gitlab.v4.objects.GroupApprovalRuleManager*
- GitLab API: https://docs.gitlab.com/api/merge_request_approvals

Examples

List group-level MR approval rules:

```
group_approval_rules = group.approval_rules.list(get_all=True)
```

Change group-level MR approval rule:

```
g_approval_rule = group.approval_rules.get(123)
g_approval_rule.user_ids = [234]
g_approval_rule.save()
```

Create new group-level MR approval rule:

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

Project approval rules

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 API: https://docs.gitlab.com/api/merge_request_approvals

Examples

List project-level MR approval rules:

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

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 MR approvals settings:

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

Change project-level MR approvals settings:

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

Merge request approval rules

References

- v4 API:
  - *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/api/merge_request_approvals

Examples

Get 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 MR-level MR approvals settings:

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

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

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

List MR-level MR approval rules:

```
mr.approval_rules.list(get_all=True)
```

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(get_all=False)
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",
```

(continues on next page)

(continued from previous page)

```
    "user_ids": [105],
    "group_ids": [653, 654],
})
```

7.6.36 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/api/milestones
  - https://docs.gitlab.com/api/group_milestones

Examples

List the milestones for a project or a group:

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

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', get_all=True)
g_milestones = group.milestones.list(state='active', get_all=True)
```

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/api/resource_milestone_events

Examples

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

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

Get a specific milestone for a resource:

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

7.6.37 Namespaces

Reference

- v4 API:
  - *gitlab.v4.objects.Namespace*
  - *gitlab.v4.objects.NamespaceManager*
  - *gitlab.Gitlab.namespaces*
- GitLab API: https://docs.gitlab.com/api/namespaces

Examples

List namespaces:

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

Search namespaces:

```
namespaces = gl.namespaces.list(search='foo', get_all=True)
```

Get a namespace by ID or path:

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

Get existence of a namespace by path:

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

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

7.6.38 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/api/notes

Examples

List the notes for a resource:

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

Get a note for a resource:

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

Create a note for a resource:

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

Update a note for a resource:

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

Delete a note for a resource:

```
note.delete()
```

7.6.39 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/api/packages#within-a-project

Examples

List the packages in a project:

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

Filter the results by `package_type` or `package_name`

```
packages = project.packages.list(package_type='pypi', get_all=True)
```

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/api/packages#within-a-group

Examples

List the packages in a group:

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

Filter the results by `package_type` or `package_name`

```
packages = group.packages.list(package_type='pypi', get_all=True)
```

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/api/packages#list-package-files

Examples

List package files for package in project:

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

Delete a package file in a project:

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

Project Package Pipelines

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectPackagePipeline*
  - *gitlab.v4.objects.ProjectPackagePipelineManager*
  - *gitlab.v4.objects.ProjectPackage.pipelines*
- GitLab API: https://docs.gitlab.com/api/packages#list-package-pipelines

Examples

List package pipelines for package in project:

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

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/user/packages/generic_packages

Examples

Upload a generic package to a project:

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

Download a project's generic package:

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

Hint

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

7.6.40 Pages and Pages domains

Project pages

References

- v4 API:
  - *gitlab.v4.objects.ProjectPages*
  - *gitlab.v4.objects.ProjectPagesManager*
  - *gitlab.v4.objects.Project.pages*
- GitLab API: https://docs.gitlab.com/api/pages

Examples

Get Pages settings for a project:

```
pages = project.pages.get()
```

Update Pages settings for a project:

```
project.pages.update(new_data={'pages_https_only': True})
```

Delete (unpublish) Pages for a project (admin only):

```
project.pages.delete()
```

Pages domains (admin only)

References

- v4 API:
  - *gitlab.v4.objects.PagesDomain*
  - *gitlab.v4.objects.PagesDomainManager*
  - *gitlab.Gitlab.pagesdomains*
- GitLab API: https://docs.gitlab.com/api/pages_domains#list-all-pages-domains

Examples

List all the existing domains (admin only):

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

Project Pages domains

References

- v4 API:
  - *gitlab.v4.objects.ProjectPagesDomain*
  - *gitlab.v4.objects.ProjectPagesDomainManager*
  - *gitlab.v4.objects.Project.pagesdomains*
- GitLab API: https://docs.gitlab.com/api/pages_domains#list-pages-domains

Examples

List domains for a project:

```
domains = project.pagesdomains.list(get_all=True)
```

Get a single domain:

```
domain = project.pagesdomains.get('d1.example.com')
```

Create a new domain:

```
domain = project.pagesdomains.create({'domain': 'd2.example.com})
```

Update an existing domain:

```
domain.certificate = open('d2.crt').read()
domain.key = open('d2.key').read()
domain.save()
```

Delete an existing domain:

```
domain.delete
# or
project.pagesdomains.delete('d2.example.com')
```

7.6.41 Personal Access Tokens

References

- v4 API:
  - *gitlab.v4.objects.PersonalAccessToken*
  - gitlab.v4.objects.PersonalAcessTokenManager
  - *gitlab.Gitlab.personal_access_tokens*
  - *gitlab.v4.objects.UserPersonalAccessToken*
  - gitlab.v4.objects.UserPersonalAcessTokenManager
  - gitlab.Gitlab.User.personal_access_tokens
- GitLab API:
  - https://docs.gitlab.com/api/personal_access_tokens
  - https://docs.gitlab.com/api/users#create-a-personal-access-token

Examples

List personal access tokens:

```
access_tokens = gl.personal_access_tokens.list(get_all=True)
print(access_tokens[0].name)
```

List personal access tokens from other user_id (admin only):

```
access_tokens = gl.personal_access_tokens.list(user_id=25, get_all=True)
```

Get a personal access token by id:

```
gl.personal_access_tokens.get(123)
```

Get the personal access token currently used:

```
gl.personal_access_tokens.get("self")
```

Revoke a personal access token fetched via list:

```
access_token = access_tokens[0]
access_token.delete()
```

Revoke a personal access token by id:

```
gl.personal_access_tokens.delete(123)
```

Revoke the personal access token currently used:

```
gl.personal_access_tokens.delete("self")
```

Rotate a personal access token and retrieve its new value:

```
token = gl.personal_access_tokens.get(42, lazy=True)
token.rotate()
print(token.token)
# or directly using a token ID
new_token_dict = gl.personal_access_tokens.rotate(42)
print(new_token_dict)
```

Self-Rotate the personal access token you are using to authenticate the request and retrieve its new value:

```
token = gl.personal_access_tokens.get(42, lazy=True)
token.rotate(self_rotate=True)
print(token.token)
```

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

```
user = gl.users.get(25, lazy=True)
access_token = user.personal_access_tokens.create({"name": "test", "scopes": "api"})
```

Note

As you can see above, you can only create personal access tokens via the Users API, but you cannot revoke these objects directly. This is because the create API uses a different endpoint than the list and revoke APIs. You need to fetch the token via the list or get API first to revoke it.

7.6.42 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/api/pipelines

Examples

List pipelines for a project:

```
pipelines = project.pipelines.list(get_all=True)
```

Get a pipeline for a project:

```
pipeline = project.pipelines.get(pipeline_id)
```

Get variables of a pipeline:

```
variables = pipeline.variables.list(get_all=True)
```

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()
```

Get latest pipeline:

```
project.pipelines.latest(ref="main")
```

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/api/pipeline_triggers

Examples

List triggers:

```
triggers = project.triggers.list(get_all=True)
```

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_decription = 'my_trigger_id'
    for t in project.triggers.list(iterator=True):
        if t.description == trigger_decription:
            return t
    return project.triggers.create({'description': trigger_decription})

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/ci/triggers/#trigger-token

Pipeline schedules

You can schedule pipeline runs using a cron-like syntax. Variables can be associated with the scheduled pipelines.

Reference

- v4 API
  - *gitlab.v4.objects.ProjectPipelineSchedule*
  - *gitlab.v4.objects.ProjectPipelineScheduleManager*
  - *gitlab.v4.objects.Project.pipelineschedules*
  - *gitlab.v4.objects.ProjectPipelineScheduleVariable*
  - *gitlab.v4.objects.ProjectPipelineScheduleVariableManager*
  - *gitlab.v4.objects.ProjectPipelineSchedule.variables*
  - *gitlab.v4.objects.ProjectPipelineSchedulePipeline*
  - *gitlab.v4.objects.ProjectPipelineSchedulePipelineManager*
  - *gitlab.v4.objects.ProjectPipelineSchedule.pipelines*
- GitLab API: https://docs.gitlab.com/api/pipeline_schedules

Examples

List pipeline schedules:

```
scheds = project.pipelineschedules.list(get_all=True)
```

Get a single schedule:

```
sched = project.pipelineschedules.get(schedule_id)
```

Create a new schedule:

```
sched = project.pipelineschedules.create({
    'ref': 'main',
    'description': 'Daily test',
    'cron': '0 1 * * *'})
```

Update a schedule:

```
sched.cron = '1 2 * * *'
sched.save()
```

Take ownership of a schedule:

```
sched.take_ownership()
```

Trigger a pipeline schedule immediately:

```
sched = projects.pipelineschedules.get(schedule_id)
sched.play()
```

Delete a schedule:

```
sched.delete()
```

List schedule variables:

```
# note: you need to use get() to retrieve the schedule variables. The
# attribute is not present in the response of a list() call
sched = projects.pipelineschedules.get(schedule_id)
vars = sched.attributes['variables']
```

Create a schedule variable:

```
var = sched.variables.create({'key': 'foo', 'value': 'bar'})
```

Edit a schedule variable:

```
var.value = 'new_value'
var.save()
```

Delete a schedule variable:

```
var.delete()
```

List all pipelines triggered by a pipeline schedule:

```
pipelines = sched.pipelines.list(get_all=True)
```

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/api/jobs

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_all=True)
```

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(get_all=True)
```

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(get_all=False)[0]
job = project.jobs.get(pipeline_job.id, lazy=True)
job.retry()
```

Get the artifacts of a job:

```
build_or_job.artifacts()
```

Get the artifacts of a job by its name from the latest successful pipeline of a branch or tag:

```
project.artifacts.download(ref_name='main', job='build')
```

> **Warning**
>
> Artifacts are entirely stored in memory in this example.

You can download artifacts as a stream. Provide a callable to handle the stream:

```
with open("archive.zip", "wb") as f:
    build_or_job.artifacts(streamed=True, action=f.write)
```

You can also directly stream the output into a file, and unzip it afterwards:

```
zipfn = "___artifacts.zip"
with open(zipfn, "wb") as f:
    build_or_job.artifacts(streamed=True, action=f.write)
subprocess.run(["unzip", "-bo", zipfn])
os.unlink(zipfn)
```

Or, you can also use the underlying response iterator directly:

```
artifact_bytes_iterator = build_or_job.artifacts(iterator=True)
```

This can be used with frameworks that expect an iterator (such as FastAPI/Starlette's `StreamingResponse`) to forward a download from GitLab without having to download the entire content server-side first:

```
@app.get("/download_artifact")
def download_artifact():
    artifact_bytes_iterator = build_or_job.artifacts(iterator=True)
    return StreamingResponse(artifact_bytes_iterator, media_type="application/zip")
```

Delete all artifacts of a project that can be deleted:

```
project.artifacts.delete()
```

Get a single artifact file:

```
build_or_job.artifact('path/to/file')
```

Get a single artifact file by branch and job:

```
project.artifacts.raw('branch', 'path/to/file', 'job')
```

Mark a job artifact as kept when expiration is set:

```
build_or_job.keep_artifacts()
```

Delete the artifacts of a job:

```
build_or_job.delete_artifacts()
```

Get a job log file / trace:

```
build_or_job.trace()
```

Warning

Traces are entirely stored in memory unless you use the streaming feature. See *the artifacts example*.

Cancel/retry a job:

```
build_or_job.cancel()
build_or_job.retry()
```

Play (trigger) a job:

```
build_or_job.play()
```

Erase a job (artifacts and trace):

```
build_or_job.erase()
```

Pipeline bridges

Get a list of bridge jobs (including child pipelines) for a pipeline.

Reference

- v4 API
  - *gitlab.v4.objects.ProjectPipelineBridge*
  - *gitlab.v4.objects.ProjectPipelineBridgeManager*
  - *gitlab.v4.objects.ProjectPipeline.bridges*
- GitLab API: https://docs.gitlab.com/api/jobs#list-pipeline-bridges

Examples

List bridges for the pipeline:

```
bridges = pipeline.bridges.list(get_all=True)
```

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/api/pipelines#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/api/pipelines#get-a-pipelines-test-report-summary

Examples

Get the test report summary for a pipeline:

```
test_report_summary = pipeline.test_report_summary.get()
```

7.6.43 Projects

Projects

Reference

- v4 API:
  - *gitlab.v4.objects.Project*
  - *gitlab.v4.objects.ProjectManager*
  - *gitlab.Gitlab.projects*
- GitLab API: https://docs.gitlab.com/api/projects

Examples

List projects:

```
projects = gl.projects.list(get_all=True)
```

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, get_all=True)
# Limit to projects with a defined visibility
projects = gl.projects.list(visibility='public', get_all=True)

# List owned projects
projects = gl.projects.list(owned=True, get_all=True)

# List starred projects
projects = gl.projects.list(starred=True, get_all=True)

# Search projects
projects = gl.projects.list(search='keyword', get_all=True)
```

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', get_all=False)[0]
user_project = alice.projects.create({'name': 'project'})
user_projects = alice.projects.list(get_all=True)
```

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', get_all=False)[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(get_all=True)
```

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()
```

Remove the avatar image for a project:

```
project.avatar = ""
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(get_all=True)
```

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/api/repositories#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(get_all=True)

# search for users
users = p.users.list(search='pattern', get_all=True)
```

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/api/project_import_export

Examples

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

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

```
# Create the export
p = gl.projects.get(my_project)
export = p.exports.create()

# Wait for the 'finished' status
export.refresh()
while export.export_status != 'finished':
    time.sleep(1)
    export.refresh()

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

You can export and upload a project to an external URL (see upstream documentation for more details):

```
project.exports.create(
    {
        "upload":
            {
                "url": "http://localhost:8080",
                "method": "POST"
            }
    }
)
```

You can also get the status of an existing export, regardless of whether it was created via the API or the Web UI:

```
project = gl.projects.get(my_project)

# Gets the current export status
export = project.exports.get()
```

Import the project into the current user's namespace:

```
with open('/tmp/export.tgz', 'rb') as f:
    output = gl.projects.import_project(
        f, path='my_new_project', name='My New Project'
    )

# Get a ProjectImport object to track the import status
project_import = gl.projects.get(output['id'], lazy=True).imports.get()
while project_import.import_status != 'finished':
    time.sleep(1)
    project_import.refresh()
```

Import the project into a namespace and override parameters:

```
with open('/tmp/export.tgz', 'rb') as f:
    output = gl.projects.import_project(
        f,
        path='my_new_project',
        name='My New Project',
        namespace='my-group',
        override_params={'visibility': 'private'},
    )
```

Import the project using file stored on a remote URL:

```
output = gl.projects.remote_import(
    url="https://whatever.com/url/file.tar.gz",
    path="my_new_remote_project",
    name="My New Remote Project",
    namespace="my-group",
    override_params={'visibility': 'private'},
)
```

Import the project using file stored on AWS S3:

```
output = gl.projects.remote_import_s3(
    path="my_new_remote_project",
    region="aws-region",
    bucket_name="aws-bucket-name",
    file_key="aws-file-key",
    access_key_id="aws-access-key-id",
    secret_access_key="secret-access-key",
    name="My New Remote Project",
    namespace="my-group",
    override_params={'visibility': 'private'},
)
```

Project custom attributes

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectCustomAttribute*
  - *gitlab.v4.objects.ProjectCustomAttributeManager*
  - *gitlab.v4.objects.Project.customattributes*
- GitLab API: https://docs.gitlab.com/api/custom_attributes

Examples

List custom attributes for a project:

```
attrs = project.customattributes.list(get_all=True)
```

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'}, get_all=True)
```

Project feature flags

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectFeatureFlag*
  - *gitlab.v4.objects.ProjectFeatureFlagManager*
  - *gitlab.v4.objects.Project.feature_flags*
- GitLab API: https://docs.gitlab.com/api/feature_flags

Examples

See *Project Feature Flags*.

Project feature flag user lists

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectFeatureFlagUserList*
  - *gitlab.v4.objects.ProjectFeatureFlagUserListManager*
  - *gitlab.v4.objects.Project.feature_flags_user_lists*
- GitLab API: https://docs.gitlab.com/api/feature_flag_user_lists

Examples

See *Project Feature Flag User Lists*.

Project files

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectFile*
  - *gitlab.v4.objects.ProjectFileManager*
  - *gitlab.v4.objects.Project.files*
- GitLab API: https://docs.gitlab.com/api/repository_files

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/api/repository_files#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/api/tags

Examples

List the project tags:

```
tags = project.tags.list(get_all=True)
```

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/api/project_snippets

Examples

List the project snippets:

```
snippets = project.snippets.list(get_all=True)
```

Get a snippet:

```
snippet = project.snippets.get(snippet_id)
```

Get the content of a snippet:

```
print(snippet.content())
```

Warning

The snippet content is entirely stored in memory unless you use the streaming feature. See *the artifacts example*.

Create a snippet:

```
snippet = project.snippets.create({'title': 'sample 1',
                                   'files': [{
                                       'file_path': 'foo.py',
                                       'content': 'import gitlab'
                                    }],
```

(continues on next page)

(continued from previous page)

```
                                    '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/api/members

Examples

List only direct project members:

```
members = project.members.list(get_all=True)
```

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_all=True)
```

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/api/projects#hooks

Examples

List the project hooks:

```
hooks = project.hooks.list(get_all=True)
```

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()
```

Test a project hook:

```
hook.test("push_events")
```

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/api/integrations

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(get_all=True)
```

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/api/projects#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/api/projects#push-rules

Examples

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

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

Get project push rules:

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

Edit project push rules:

```
pr.branch_name_regex = '^(main|develop|support-\d+|release-\d+\..+|hotfix-.+|feature-.+)$
↪'
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/api/protected_tags

Examples

Get a list of protected tags from a project:

```
protected_tags = project.protectedtags.list(get_all=True)
```

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/api/project_statistics

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/api/projects#get-the-path-to-repository-storage

Examples

Get the repository storage details for a project:

```
storage = project.storage.get()
```

Get the repository storage disk path:

```
disk_path = project.storage.get().disk_path
```

7.6.44 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/api/project_access_tokens

Examples

List project access tokens:

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

Get a project access token by id:

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

Create project access token:

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

Revoke a project access token:

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

Rotate a project access token and retrieve its new value:

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

Self-Rotate the project access token you are using to authenticate the request and retrieve its new value:

```
token = project.access_tokens.get(42, lazy=True)
token.rotate(self_rotate=True)
print(new_token.token)
```

7.6.45 Project Feature Flags

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectFeatureFlag*
  - *gitlab.v4.objects.ProjectFeatureFlagManager*
  - *gitlab.v4.objects.Project.feature_flags*
- GitLab API: https://docs.gitlab.com/api/feature_flags

Examples

List feature flags:

```
flags = project.feature_flags.list()
```

Get a feature flag:

```
flag = project.feature_flags.get('my_feature_flag')
```

Create a feature flag:

```
flag = project.feature_flags.create({'name': 'my_feature_flag', 'version': 'new_version_
↪flag'})
```

Create a feature flag with strategies:

```
flag = project.feature_flags.create({
    'name': 'my_complex_flag',
    'version': 'new_version_flag',
    'strategies': [{
        'name': 'userWithId',
        'parameters': {'userIds': 'user1,user2'}
    }]
})
```

Update a feature flag:

```
flag.description = 'Updated description'
flag.save()
```

Rename a feature flag:

```
# You can rename a flag by changing its name attribute and calling save()
flag.name = 'new_flag_name'
flag.save()

# Alternatively, you can use the manager's update method
project.feature_flags.update('old_flag_name', {'name': 'new_flag_name'})
```

Delete a feature flag:

```
flag.delete()
```

See also

- *Project Feature Flag User Lists*

7.6.46 Project Feature Flag User Lists

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectFeatureFlagUserList*
  - *gitlab.v4.objects.ProjectFeatureFlagUserListManager*
  - *gitlab.v4.objects.Project.feature_flags_user_lists*
- GitLab API: https://docs.gitlab.com/api/feature_flag_user_lists

Examples

List user lists:

```
user_lists = project.feature_flags_user_lists.list()
```

Get a user list:

```
user_list = project.feature_flags_user_lists.get(list_iid)
```

Create a user list:

```
user_list = project.feature_flags_user_lists.create({
    'name': 'my_user_list',
    'user_xids': 'user1,user2,user3'
})
```

Update a user list:

```
user_list.name = 'updated_list_name'
user_list.user_xids = 'user1,user2'
user_list.save()
```

Delete a user list:

```
user_list.delete()
```

Search for a user list:

```
user_lists = project.feature_flags_user_lists.list(search='my_list')
```

See also

- *Project Feature Flags*

7.6.47 Protected branches

You can define a list of protected branch names on a repository or group. Names can use wildcards (*).

References

- v4 API:
  - *gitlab.v4.objects.ProjectProtectedBranch*
  - *gitlab.v4.objects.ProjectProtectedBranchManager*
  - *gitlab.v4.objects.Project.protectedbranches*
  - gitlab.v4.objects.GroupProtectedBranch
  - gitlab.v4.objects.GroupProtectedBranchManager
  - *gitlab.v4.objects.Group.protectedbranches*
- GitLab API: https://docs.gitlab.com/api/protected_branches#protected-branches-api

Examples

Get the list of protected branches for a project or group:

```
p_branches = project.protectedbranches.list()
p_branches = group.protectedbranches.list()
```

Get a single protected branch:

```
p_branch = project.protectedbranches.get('main')
p_branch = group.protectedbranches.get('main')
```

Update a protected branch:

```
p_branch.allow_force_push = True
p_branch.save()
```

Create a protected branch:

```
p_branch = project.protectedbranches.create({
    'name': '*-stable',
    'merge_access_level': gitlab.const.AccessLevel.DEVELOPER,
    'push_access_level': gitlab.const.AccessLevel.MAINTAINER
})
```

Create a protected branch with more granular access control:

```
p_branch = project.protectedbranches.create({
    'name': '*-stable',
    'allowed_to_push': [{"user_id": 99}, {"user_id": 98}],
    'allowed_to_merge': [{"group_id": 653}],
    'allowed_to_unprotect': [{"access_level": gitlab.const.AccessLevel.MAINTAINER}]
})
```

Delete a protected branch:

```
project.protectedbranches.delete('*-stable')
# or
p_branch.delete()
```

7.6.48 Protected container repositories

You can list and manage container registry protection rules in a project.

References

- v4 API:
  - `gitlab.v4.objects.ProjectRegistryRepositoryProtectionRuleRule`
  - `gitlab.v4.objects.ProjectRegistryRepositoryProtectionRuleRuleManager`
  - *`gitlab.v4.objects.Project.registry_protection_repository_rules`*
- GitLab API: https://docs.gitlab.com/api/container_repository_protection_rules

Examples

List the container registry protection rules for a project:

```
registry_rules = project.registry_protection_repository_rules.list(get_all=True)
```

Create a container registry protection rule:

```
registry_rule = project.registry_protection_repository_rules.create(
    {
        'repository_path_pattern': 'test/image',
        'minimum_access_level_for_push': 'maintainer',
        'minimum_access_level_for_delete': 'maintainer',
    }
)
```

Update a container registry protection rule:

```
registry_rule.minimum_access_level_for_push = 'owner'
registry_rule.save()
```

Delete a container registry protection rule:

```
registry_rule = project.registry_protection_repository_rules.delete(registry_rule.id)
# or
registry_rule.delete()
```

7.6.49 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/api/protected_environments

Examples

Get the list of protected environments for a project:

```
p_environments = project.protected_environments.list(get_all=True)
```

Get a single protected environment:

```
p_environments = project.protected_environments.get('production')
```

Protect an existing environment:

```
p_environment = project.protected_environments.create(
    {
        'name': 'production',
        'deploy_access_levels': [
            {'access_level': 40}
        ],
    }
)
```

Unprotect a protected environment:

```
p_environment = project.protected_environments.delete('production')
# or
p_environment.delete()
```

7.6.50 Protected packages

You can list and manage package protection rules in a project.

References

- v4 API:
  - *gitlab.v4.objects.ProjectPackageProtectionRule*
  - *gitlab.v4.objects.ProjectPackageProtectionRuleManager*
  - *gitlab.v4.objects.Project.package_protection_rules*
- GitLab API: https://docs.gitlab.com/api/project_packages_protection_rules

Examples

List the package protection rules for a project:

```
package_rules = project.package_protection_rules.list(get_all=True)
```

Create a package protection rule:

```
package_rule = project.package_protection_rules.create(
    {
        'package_name_pattern': 'v*',
        'package_type': 'npm',
        'minimum_access_level_for_push': 'maintainer'
    }
)
```

Update a package protection rule:

```
package_rule.minimum_access_level_for_push = 'developer'
package_rule.save()
```

Delete a package protection rule:

```
package_rule = project.package_protection_rules.delete(package_rule.id)
# or
package_rule.delete()
```

7.6.51 Project Pull Mirror

Pull Mirror allow you to set up pull mirroring for a project.

References

- v4 API:
  - *gitlab.v4.objects.ProjectPullMirror*
  - *gitlab.v4.objects.ProjectPullMirrorManager*
  - *gitlab.v4.objects.Project.pull_mirror*
- GitLab API: https://docs.gitlab.com/api/project_pull_mirroring/

Examples

Get the current pull mirror of a project:

```
mirrors = project.pull_mirror.get()
```

Create (and enable) a remote mirror for a project:

```
mirror = project.pull_mirror.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()
```

Start a sync of the pull mirror:

```
project.pull_mirror.start()
```

7.6.52 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/api/releases/index

Examples

Get a list of releases from a project:

```
project = gl.projects.get(project_id, lazy=True)
release = project.releases.list(get_all=True)
```

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/api/releases/links

Examples

Get a list of releases from a project:

```
links = release.links.list()
```

Get a single release link:

```
link = release.links.get(1)
```

Create a release link for a release:

```
link = release.links.create({"url": "https://example.com/asset", "name": "asset"})
```

Delete a release link:

```
# via its ID from release attributes
release.links.delete(1)

# delete object directly
link.delete()
```

7.6.53 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/api/runners

Examples

Use the `runners.list()` and `runners_all.list()` methods to list runners. `runners.list()` - Get a list of specific runners available to the user `runners_all.list()` - Get a list of all runners in the GitLab instance (specific and shared). Access is restricted to users with administrator access.

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

- `active`
- `paused`

- online
- specific (runners_all.list() only)
- shared (runners_all.list() only)

Note

The returned objects hold minimal information about the runners. Use the get() method to retrieve detail about a runner.

Runners returned via runners_all.list() also cannot be manipulated directly. You will need to use the get() method to create an editable object.

```
# List owned runners
runners = gl.runners.list(get_all=True)

# List owned runners with a filter
runners = gl.runners.list(scope='active', get_all=True)

# List all runners in the GitLab instance (specific and shared), using a filter
runners = gl.runners_all.list(scope='paused', get_all=True)
```

Get a runner's detail:

```
runner = gl.runners.get(runner_id)
```

Register a new runner:

```
runner = gl.runners.create({'token': secret_token})
```

Note

A new runner registration workflow has been introduced since GitLab 16.0. This new workflow comes with a new API endpoint to create runners, which does not use registration tokens.

The new endpoint can be called using gl.user.runners.create() after authenticating with gl.auth().

Update a runner:

```
runner = gl.runners.get(runner_id)
runner.tag_list.append('new_tag')
runner.save()
```

Remove a runner:

```
gl.runners.delete(runner_id)
# or
runner.delete()
```

Remove a runner by its authentication token:

```
gl.runners.delete(token="runner-auth-token")
```

Verify a registered runner token:

```
try:
    gl.runners.verify(runner_token)
    print("Valid token")
except GitlabVerifyError:
    print("Invalid token")
```

Project/Group runners

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectRunner*
  - *gitlab.v4.objects.ProjectRunnerManager*
  - *gitlab.v4.objects.Project.runners*
  - *gitlab.v4.objects.GroupRunner*
  - *gitlab.v4.objects.GroupRunnerManager*
  - *gitlab.v4.objects.Group.runners*
- GitLab API: https://docs.gitlab.com/api/runners

Examples

List the runners for a project:

```
runners = project.runners.list(get_all=True)
```

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/api/runners

Examples

List for jobs for a runner:

```
jobs = runner.jobs.list(get_all=True)
```

Filter the list using the jobs status:

```
# status can be 'running', 'success', 'failed' or 'canceled'
active_jobs = runner.jobs.list(status='running', get_all=True)
```

7.6.54 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/api/remote_mirrors

Examples

Get the list of a project's remote mirrors:

```
mirrors = project.remote_mirrors.list(get_all=True)
```

Create (and enable) a remote mirror for a project:

```
mirror = project.remote_mirrors.create({'url': 'https://gitlab.com/example.git',
                                        'enabled': True})
```

Update an existing remote mirror's attributes:

```
mirror.enabled = False
mirror.only_protected_branches = True
mirror.save()
```

Delete an existing remote mirror:

```
mirror.delete()
```

Force push mirror update:

```
mirror.sync()
```

7.6.55 Registry Repositories

References

- v4 API:

  - *gitlab.v4.objects.ProjectRegistryRepository*
  - *gitlab.v4.objects.ProjectRegistryRepositoryManager*
  - *gitlab.v4.objects.Project.repositories*
- Gitlab API: https://docs.gitlab.com/api/container_registry

Examples

Get the list of container registry repositories associated with the project:

```
repositories = project.repositories.list(get_all=True)
```

Get the list of all project container registry repositories in a group:

```
repositories = group.registry_repositories.list()
```

Delete repository:

```
project.repositories.delete(id=x)
# or
repository = repositories.pop()
repository.delete()
```

7.6.56 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/api/container_registry

Examples

Get the list of repository tags in given registry:

```
repositories = project.repositories.list(get_all=True)
repository = repositories.pop()
tags = repository.tags.list(get_all=True)
```

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/api/container_registry#delete-repository-tags-in-bulk

7.6.57 Resource Groups

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectResourceGroup*
  - *gitlab.v4.objects.ProjectResourceGroupManager*
  - *gitlab.v4.objects.Project.resource_groups*
  - *gitlab.v4.objects.ProjectResourceGroupUpcomingJob*
  - *gitlab.v4.objects.ProjectResourceGroupUpcomingJobManager*
  - *gitlab.v4.objects.ProjectResourceGroup.upcoming_jobs*
- Gitlab API: https://docs.gitlab.com/api/resource_groups

Examples

List resource groups for a project:

```
project = gl.projects.get(project_id, lazy=True)
resource_group = project.resource_groups.list(get_all=True)
```

Get a single resource group:

```
resource_group = project.resource_groups.get("production")
```

Edit a resource group:

```
resource_group.process_mode = "oldest_first"
resource_group.save()
```

List upcoming jobs for a resource group:

```
upcoming_jobs = resource_group.upcoming_jobs.list(get_all=True)
```

7.6.58 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/api/search

Examples

Search for issues matching a specific string:

```
# global search
gl.search(gitlab.const.SearchScope.ISSUES, 'regression')

# group search
group = gl.groups.get('mygroup')
group.search(gitlab.const.SearchScope.ISSUES, 'regression')

# project search
project = gl.projects.get('myproject')
project.search(gitlab.const.SearchScope.ISSUES, 'regression')
```

The `search()` methods implement the pagination support:

```
# get lists of 10 items, and start at page 2
gl.search(gitlab.const.SearchScope.ISSUES, search_str, page=2, per_page=10)

# get a generator that will automatically make required API calls for
# pagination
for item in gl.search(gitlab.const.SearchScope.ISSUES, search_str, iterator=True):
    do_something(item)
```

The search API doesn't return objects, but dicts. If you need to act on objects, you need to create them explicitly:

```
for item in gl.search(gitlab.const.SearchScope.ISSUES, search_str, iterator=True):
    issue_project = gl.projects.get(item['project_id'], lazy=True)
    issue = issue_project.issues.get(item['iid'])
    issue.state = 'closed'
    issue.save()
```

7.6.59 Secure Files

secure files

References

- v4 API:
  - *gitlab.v4.objects.ProjectSecureFile*
  - *gitlab.v4.objects.ProjectSecureFileManager*
  - *gitlab.v4.objects.Project.secure_files*
- GitLab API: https://docs.gitlab.com/api/secure_files

Examples

Get a project secure file:

```
secure_files = gl.projects.get(1, lazy=True).secure_files.get(1)
print(secure_files.name)
```

List project secure files:

```
secure_files = gl.projects.get(1, lazy=True).secure_files.list(get_all=True)
print(secure_files[0].name)
```

Create project secure file:

```
secure_file = gl.projects.get(1).secure_files.create({"name": "test", "file": "secure.txt
↪"})
```

Download a project secure file:

```
content = secure_file.download()
print(content)
with open("/tmp/secure.txt", "wb") as f:
    secure_file.download(streamed=True, action=f.write)
```

Remove a project secure file:

```
gl.projects.get(1).secure_files.delete(1)
# or
secure_file.delete()
```

7.6.60 Service Accounts

References

- v4 API:
  - *gitlab.v4.objects.ServiceAccount*
  - *gitlab.v4.objects.ServiceAccountManager*
  - *gitlab.v4.objects.GroupServiceAccount*
  - *gitlab.v4.objects.GroupServiceAccountManager*
  - *gitlab.v4.objects.GroupServiceAccountAccessToken*
  - *gitlab.v4.objects.GroupServiceAccountAccessTokenManager*
  - *gitlab.v4.objects.ProjectServiceAccount*
  - *gitlab.v4.objects.ProjectServiceAccountManager*
  - *gitlab.v4.objects.ProjectServiceAccountAccessToken*
  - *gitlab.v4.objects.ProjectServiceAccountAccessTokenManager*
- GitLab API: https://docs.gitlab.com/api/service_accounts/

Instance service accounts

List instance service accounts:

```
accounts = gl.service_accounts.list()
```

Create an instance service account:

```
sa = gl.service_accounts.create({})
# with optional attributes
sa = gl.service_accounts.create({"name": "my-bot", "username": "my-bot", "email": "my-
↪bot@example.com"})
```

Update an instance service account:

```
gl.service_accounts.update(sa.id, {"name": "renamed-bot"})
# or via the object
sa.name = "renamed-bot"
sa.save()
```

Group service accounts

List group service accounts:

```
accounts = group.service_accounts.list()
```

Create a group service account:

```
sa = group.service_accounts.create({})
# with optional attributes
sa = group.service_accounts.create({"name": "ci-bot", "username": "ci-bot"})
```

Update a group service account:

```
group.service_accounts.update(sa.id, {"name": "renamed-bot"})
# or via the object
sa.name = "renamed-bot"
sa.save()
```

Delete a group service account:

```
group.service_accounts.delete(sa.id)
# or via the object
sa.delete()
```

Group service account personal access tokens

List tokens for a group service account:

```
tokens = sa.access_tokens.list()
```

Create a token for a group service account:

```
token = sa.access_tokens.create({
    "name": "ci-token",
    "scopes": ["api"],
    "expires_at": "2026-01-01",
})
print(token.token)
```

Rotate a token:

```
token.rotate()
print(token.token)
# or directly using a token ID
new_token = sa.access_tokens.rotate(token.id)
print(new_token["token"])
```

Revoke a token:

```
sa.access_tokens.delete(token.id)
# or via the object
token.delete()
```

Project service accounts

List project service accounts:

```
accounts = project.service_accounts.list()
```

Create a project service account:

```
sa = project.service_accounts.create({})
# with optional attributes
sa = project.service_accounts.create({"name": "ci-bot", "username": "ci-bot"})
```

Update a project service account:

```
project.service_accounts.update(sa.id, {"name": "renamed-bot"})
# or via the object
sa.name = "renamed-bot"
sa.save()
```

Delete a project service account:

```
project.service_accounts.delete(sa.id)
# or via the object
sa.delete()
```

Project service account personal access tokens

List tokens for a project service account:

```
tokens = sa.access_tokens.list()
```

Create a token for a project service account:

```
token = sa.access_tokens.create({
    "name": "ci-token",
    "scopes": ["read_repository"],
    "expires_at": "2026-01-01",
})
print(token.token)
```

Rotate a token:

```
token.rotate()
print(token.token)
# or directly using a token ID
new_token = sa.access_tokens.rotate(token.id)
print(new_token["token"])
```

Revoke a token:

```
sa.access_tokens.delete(token.id)
# or via the object
token.delete()
```

7.6.61 Settings

Reference

- v4 API:
  - *gitlab.v4.objects.ApplicationSettings*
  - *gitlab.v4.objects.ApplicationSettingsManager*
  - *gitlab.Gitlab.settings*
- GitLab API: https://docs.gitlab.com/api/settings

Examples

Get the settings:

```
settings = gl.settings.get()
```

Update the settings:

```
settings.signin_enabled = False
settings.save()
```

7.6.62 Snippets

Reference

- v4 API:
  - *gitlab.v4.objects.Snippet*
  - gitlab.v4.objects.SnipptManager
  - *gitlab.Gitlab.snippets*
- GitLab API: https://docs.gitlab.com/api/snippets

Examples

List snippets owned by the current user:

```
snippets = gl.snippets.list(get_all=True)
```

List the public snippets:

```
public_snippets = gl.snippets.list_public()
```

List all snippets:

```
all_snippets = gl.snippets.list_all(get_all=True)
```

Warning

Only users with the Administrator or Auditor access levels can see all snippets (both personal and project). See the upstream API documentation for more details.

Get a snippet:

```
snippet = gl.snippets.get(snippet_id)
# get the content
content = snippet.content()
```

Warning

Blobs are entirely stored in memory unless you use the streaming feature. See *the artifacts example*.

Create a snippet:

```
snippet = gl.snippets.create({'title': 'snippet1',
                              'files': [{
                                  'file_path': 'foo.py',
                                  'content': 'import gitlab'
                               }],
                             })
```

Update the snippet attributes:

```
snippet.visibility_level = gitlab.const.Visibility.PUBLIC
snippet.save()
```

To update a snippet code you need to create a `ProjectSnippet` object:

```
snippet = gl.snippets.get(snippet_id)
project = gl.projects.get(snippet.projec_id, lazy=True)
editable_snippet = project.snippets.get(snippet.id)
editable_snippet.code = new_snippet_content
editable_snippet.save()
```

Delete a snippet:

```
gl.snippets.delete(snippet_id)
# or
snippet.delete()
```

Get user agent detail (admin only):

```
detail = snippet.user_agent_detail()
```

7.6.63 Statistics

Reference

- v4 API:
  - *gitlab.v4.objects.ApplicationStatistics*
  - *gitlab.v4.objects.ApplicationStatisticsManager*
  - *gitlab.Gitlab.statistics*
- GitLab API: https://docs.gitlab.com/api/statistics

Examples

Get the statistics:

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

7.6.64 External Status Checks

Manage external status checks for projects and merge requests.

Project external status checks

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectExternalStatusCheck*
  - *gitlab.v4.objects.ProjectExternalStatusCheckManager*
  - *gitlab.v4.objects.Project.external_status_checks*
- GitLab API: https://docs.gitlab.com/api/status_checks

Examples

List external status checks for a project:

```
status_checks = project.external_status_checks.list(get_all=True)
```

Create an external status check with shared secret:

```
status_checks = project.external_status_checks.create({
    "name": "mr_blocker",
    "external_url": "https://example.com/mr-status-check",
    "shared_secret": "secret-string"
})
```

Create an external status check with shared secret for protected branches:

```
protected_branch = project.protectedbranches.get('main')

status_check = project.external_status_checks.create({
    "name": "mr_blocker",
    "external_url": "https://example.com/mr-status-check",
    "shared_secret": "secret-string",
    "protected_branch_ids": [protected_branch.id]
})
```

Update an external status check:

```
status_check.external_url = "https://example.com/mr-blocker"
status_check.save()
```

Delete an external status check:

```
status_check.delete(status_check_id)
```

7.6.65 System hooks

Reference

- v4 API:
  - *gitlab.v4.objects.Hook*
  - *gitlab.v4.objects.HookManager*
  - *gitlab.Gitlab.hooks*

- GitLab API: https://docs.gitlab.com/api/system_hooks

Examples

List the system hooks:

```
hooks = gl.hooks.list(get_all=True)
```

Create a system hook:

```
gl.hooks.get(hook_id)
```

Test a system hook. The returned object is not usable (it misses the hook ID):

```
hook = gl.hooks.create({'url': 'http://your.target.url'})
```

Delete a system hook:

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

7.6.66 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/api/templates/licenses

Examples

List known license templates:

```
licenses = gl.licenses.list(get_all=True)
```

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/api/templates/gitignores

Examples

List known gitignore templates:

```
gitignores = gl.gitignores.list(get_all=True)
```

Get a gitignore template:

```
gitignore = gl.gitignores.get('Python')
print(gitignore.content)
```

GitLab CI templates

Reference

- v4 API:
  - *gitlab.v4.objects.Gitlabciyml*
  - *gitlab.v4.objects.GitlabciymlManager*
  - *gitlab.Gitlab.gitlabciymls*
- GitLab API: https://docs.gitlab.com/api/templates/gitlab_ci_ymls

Examples

List known GitLab CI templates:

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

Get a GitLab CI template:

```
gitlabciyml = gl.gitlabciymls.get('Pelican')
print(gitlabciyml.content)
```

Dockerfile templates

Reference

- v4 API:
  - *gitlab.v4.objects.Dockerfile*
  - *gitlab.v4.objects.DockerfileManager*
  - *gitlab.Gitlab.gitlabciymls*

- GitLab API: https://docs.gitlab.com/api/templates/dockerfiles

Examples

List known Dockerfile templates:

```
dockerfiles = gl.dockerfiles.list(get_all=True)
```

Get a Dockerfile template:

```
dockerfile = gl.dockerfiles.get('Python')
print(dockerfile.content)
```

Project templates

These templates are project-specific versions of the templates above, as well as issue and merge request templates.

Reference

- v4 API:
  - *gitlab.v4.objects.ProjectLicenseTemplate*
  - *gitlab.v4.objects.ProjectLicenseTemplateManager*
  - *gitlab.v4.objects.Project.license_templates*
  - *gitlab.v4.objects.ProjectGitignoreTemplate*
  - *gitlab.v4.objects.ProjectGitignoreTemplateManager*
  - *gitlab.v4.objects.Project.gitignore_templates*
  - *gitlab.v4.objects.ProjectGitlabciymlTemplate*
  - *gitlab.v4.objects.ProjectGitlabciymlTemplateManager*
  - *gitlab.v4.objects.Project.gitlabciyml_templates*
  - *gitlab.v4.objects.ProjectDockerfileTemplate*
  - *gitlab.v4.objects.ProjectDockerfileTemplateManager*
  - *gitlab.v4.objects.Project.dockerfile_templates*
  - *gitlab.v4.objects.ProjectIssueTemplate*
  - *gitlab.v4.objects.ProjectIssueTemplateManager*
  - *gitlab.v4.objects.Project.issue_templates*
  - *gitlab.v4.objects.ProjectMergeRequestTemplate*
  - *gitlab.v4.objects.ProjectMergeRequestTemplateManager*
  - *gitlab.v4.objects.Project.merge_request_templates*
- GitLab API: https://docs.gitlab.com/api/project_templates

Examples

List known project templates:

```
license_templates = project.license_templates.list(get_all=True)
gitignore_templates = project.gitignore_templates.list(get_all=True)
gitlabciyml_templates = project.gitlabciyml_templates.list(get_all=True)
dockerfile_templates = project.dockerfile_templates.list(get_all=True)
issue_templates = project.issue_templates.list(get_all=True)
merge_request_templates = project.merge_request_templates.list(get_all=True)
```

Get project templates:

```
license_template = project.license_templates.get('apache-2.0')
gitignore_template = project.gitignore_templates.get('Python')
gitlabciyml_template = project.gitlabciyml_templates.get('Pelican')
dockerfile_template = project.dockerfile_templates.get('Python')
issue_template = project.issue_templates.get('Default')
merge_request_template = project.merge_request_templates.get('Default')

print(license_template.content)
print(gitignore_template.content)
print(gitlabciyml_template.content)
print(dockerfile_template.content)
print(issue_template.content)
print(merge_request_template.content)
```

Create an issue or merge request using a description template:

```
issue = project.issues.create({'title': 'I have a bug',
                               'description': issue_template.content})
mr = project.mergerequests.create({'source_branch': 'cool_feature',
                                   'target_branch': 'main',
                                   'title': 'merge cool feature',
                                   'description': merge_request_template.content})
```

7.6.67 Todos

Reference

- v4 API:
  - Todo
  - TodoManager
  - *gitlab.Gitlab.todos*
- GitLab API: https://docs.gitlab.com/api/todos

Examples

List active todos:

```
todos = gl.todos.list(get_all=True)
```

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, get_all=True)
todos = gl.todos.list(state='done', type='Issue', get_all=True)
```

Mark a todo as done:

```
todos = gl.todos.list(project_id=1, get_all=True)
todos[0].mark_as_done()
```

Mark all the todos as done:

```
gl.todos.mark_all_as_done()
```

7.6.68 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/api/topics

This endpoint requires admin access for creating, updating and deleting objects.

Examples

List project topics on the GitLab instance:

```
topics = gl.topics.list(get_all=True)
```

Get a specific topic by its ID:

```
topic = gl.topics.get(topic_id)
```

Create a new topic:

```
topic = gl.topics.create({"name": "my-topic", "title": "my title"})
```

Update a topic:

```
topic.description = "My new topic"
topic.save()

# or
gl.topics.update(topic_id, {"description": "My new topic"})
```

Delete a topic:

```
topic.delete()

# or
gl.topics.delete(topic_id)
```

Merge a source topic into a target topic:

```
gl.topics.merge(topic_id, target_topic_id)
```

Set the avatar image for a topic:

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

Remove the avatar image for a topic:

```
topic.avatar = ""
topic.save()
```

7.6.69 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/api/users
  - https://docs.gitlab.com/api/projects#list-projects-starred-by-a-user

Examples

Get the list of users:

```
users = gl.users.list(get_all=True)
```

Search users whose username match a given string:

```
users = gl.users.list(search='foo', get_all=True)
```

Get a single user:

```
# by ID
user = gl.users.get(user_id)
# by username
user = gl.users.list(username='root', get_all=False)[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_all=True)
```

Get the followings of a user:

```
user.following_users.list(get_all=True)
```

List a user's contributed projects:

```
user.contributed_projects.list(get_all=True)
```

List a user's starred projects:

```
user.starred_projects.list(get_all=True)
```

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/api/custom_attributes

Examples

List custom attributes for a user:

```
attrs = user.customattributes.list(get_all=True)
```

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'}, get_all=True)
```

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/api/user_tokens#get-all-impersonation-tokens-of-a-user

List impersonation tokens for a user:

```
i_t = user.impersonationtokens.list(state='active', get_all=True)
i_t = user.impersonationtokens.list(state='inactive', get_all=True)
```

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/api/projects#list-a-users-projects

List visible projects in the user's namespace:

```
projects = user.projects.list(get_all=True)
```

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/api/users#list-projects-and-groups-that-a-user-is-a-member-of

List direct memberships for a user:

```
memberships = user.memberships.list(get_all=True)
```

List only direct project memberships:

```
memberships = user.memberships.list(type='Project', get_all=True)
```

List only direct group memberships:

```
memberships = user.memberships.list(type='Namespace', get_all=True)
```

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/api/users

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/api/user_keys#list-your-gpg-keys

Examples

List GPG keys for a user:

```
gpgkeys = user.gpgkeys.list(get_all=True)
```

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/api/user_keys#get-a-single-ssh-key

Examples

List SSH keys for a user:

```
keys = user.keys.list(get_all=True)
```

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/api/users#get-the-status-of-a-user

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/api/user_email_addresses

Examples

List emails for a user:

```
emails = user.emails.list(get_all=True)
```

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/api/users#list-a-users-activity

Examples

Get the users activities:

```
activities = gl.user_activities.list(
    query_parameters={'from': '2018-07-01'},
    get_all=True,
)
```

Create new runner

References

- New runner registration API endpoint (see Migrating to the new runner registration workflow)
- v4 API:
  - *gitlab.v4.objects.CurrentUserRunner*
  - *gitlab.v4.objects.CurrentUserRunnerManager*
  - gitlab.Gitlab.user.runners
- GitLab API : https://docs.gitlab.com/api/users#create-a-runner-linked-to-a-user

Examples

Create an instance-wide runner:

```
runner = gl.user.runners.create({
    "runner_type": "instance_type",
    "description": "My brand new runner",
    "paused": True,
    "locked": False,
    "run_untagged": True,
    "tag_list": ["linux", "docker", "testing"],
    "access_level": "not_protected"
})
```

Create a group runner:

```
runner = gl.user.runners.create({
    "runner_type": "group_type",
    "group_id": 12345678,
    "description": "My brand new runner",
    "paused": True,
    "locked": False,
    "run_untagged": True,
    "tag_list": ["linux", "docker", "testing"],
```

(continues on next page)

(continued from previous page)

```
    "access_level": "not_protected"
})
```

Create a project runner:

```
runner = gl.user.runners.create({
    "runner_type": "project_type",
    "project_id": 987564321,
    "description": "My brand new runner",
    "paused": True,
    "locked": False,
    "run_untagged": True,
    "tag_list": ["linux", "docker", "testing"],
    "access_level": "not_protected"
})
```

7.6.70 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/api/instance_level_ci_variables

Examples

List all instance variables:

```
variables = gl.variables.list(get_all=True)
```

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/api/instance_level_ci_variables
  - https://docs.gitlab.com/api/project_level_variables
  - https://docs.gitlab.com/api/group_level_variables

Examples

List variables:

```
p_variables = project.variables.list(get_all=True)
g_variables = group.variables.list(get_all=True)
```

Get a variable:

```
p_var = project.variables.get('key_name')
g_var = group.variables.get('key_name')
```

Note

If there are multiple variables with the same key, use `filter` to select the correct `environment_scope`. See the GitLab API docs for more information.

Create a variable:

```
var = project.variables.create({'key': 'key1', 'value': 'value1'})
var = group.variables.create({'key': 'key1', 'value': 'value1'})
```

Note

If a variable with the same key already exists, the new variable must have a different `environment_scope`. Otherwise, GitLab returns a message similar to: `VARIABLE_NAME has already been taken`. See the GitLab API docs for more information.

Update a variable value:

```
var.value = 'new_value'
var.save()
# or
project.variables.update("key1", {"value": "new_value"})
```

Note

If there are multiple variables with the same key, use `filter` to select the correct `environment_scope`. See the GitLab API docs for more information.

Remove a variable:

```
project.variables.delete('key_name')
group.variables.delete('key_name')
# or
var.delete()
```

Note

If there are multiple variables with the same key, use `filter` to select the correct `environment_scope`. See the GitLab API docs for more information.

7.6.71 Sidekiq metrics

Reference

- v4 API:
  - *gitlab.v4.objects.SidekiqManager*
  - *gitlab.Gitlab.sidekiq*
- GitLab API: https://docs.gitlab.com/api/sidekiq_metrics

Examples

```
gl.sidekiq.queue_metrics()
gl.sidekiq.process_metrics()
gl.sidekiq.job_stats()
gl.sidekiq.compound_metrics()
```

7.6.72 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/api/wikis
- GitLab API for Groups: https://docs.gitlab.com/api/group_wikis

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_all=True)
```

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_all=True)
```

Get a single wiki page for a project:

```
page = project.wikis.get(page_slug)
```

Get a single wiki page for a group:

```
page = group.wikis.get(page_slug)
```

Get the contents of a wiki page:

```
print(page.content)
```

Create a wiki page on a project level:

```
page = project.wikis.create({'title': 'Wiki Page 1',
                             'content': open(a_file).read()})
```

Update a wiki page:

```
page.content = 'My new content'
page.save()
```

Delete a wiki page:

```
page.delete()
```

File uploads

Reference

- v4 API:
  - `gitlab.v4.objects.ProjectWiki.upload`
  - `gitlab.v4.objects.GrouptWiki.upload`
- Gitlab API for Projects: https://docs.gitlab.com/api/wikis#upload-an-attachment-to-the-wiki-repository
- Gitlab API for Groups: https://docs.gitlab.com/api/group_wikis#upload-an-attachment-to-the-wiki-repository

Examples

Upload a file into a project wiki using a filesystem path:

```
page = project.wikis.get(page_slug)
page.upload("filename.txt", filepath="/some/path/filename.txt")
```

Upload a file into a project wiki with raw data:

```
page.upload("filename.txt", filedata="Raw data")
```

Upload a file into a group wiki using a filesystem path:

```
page = group.wikis.get(page_slug)
page.upload("filename.txt", filepath="/some/path/filename.txt")
```

Upload a file into a group wiki using raw data:

```
page.upload("filename.txt", filedata="Raw data")
```

7.6.73 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/api/project_clusters
- GitLab API: https://docs.gitlab.com/api/group_clusters

Examples

List clusters for a project:

```
clusters = project.clusters.list(get_all=True)
```

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(get_all=True)
```

Create an cluster for a group:

```
cluster = group.clusters.create(
{
    "name": "cluster1",
    "platform_kubernetes_attributes": {
        "api_url": "http://url",
        "token": "tokenval",
    },
})
```

Retrieve a specific cluster for a group:

```
cluster = group.clusters.get(cluster_id)
```

Update an cluster for a group:

```
cluster.platform_kubernetes_attributes = {"api_url": "http://newurl"}
cluster.save()
```

Delete an cluster for a group:

```
cluster = group.clusters.delete(cluster_id)
# or
cluster.delete()
```

7.7 API reference (gitlab package)

7.7.1 Module contents

Wrapper for the GitLab API.

class gitlab.**AsyncGraphQL**(*url: str | None = None*, ** (Keyword-only parameters separator (PEP 3102))*, *token: str | None = None*, *ssl_verify: bool | str = True*, *client: AsyncClient | None = None*, *timeout: float | None = None*, *user_agent: str = 'python-gitlab/8.4.0'*, *fetch_schema_from_transport: bool = False*, *max_retries: int = 10*, *obey_rate_limit: bool = True*, *retry_transient_errors: bool = False*)

Bases: _BaseGraphQL

Parameters

- **url** (*str | None*)
- **token** (*str | None*)
- **ssl_verify** (*bool | str*)
- **client** (*httpx.AsyncClient | None*)
- **timeout** (*float | None*)
- **user_agent** (*str*)
- **fetch_schema_from_transport** (*bool*)
- **max_retries** (*int*)
- **obey_rate_limit** (*bool*)
- **retry_transient_errors** (*bool*)

async execute(*request: str*, **args:* Any, ***kwargs:* Any) → *Any*

Parameters

- **request** (*str*)
- **args** (Any)
- **kwargs** (Any)

Return type
Any

class gitlab.**Gitlab**(*url: str | None = None*, *private_token: str | None = None*, *oauth_token: str | None = None*, *job_token: str | None = None*, *ssl_verify: bool | str = True*, *http_username: str | None = None*, *http_password: str | None = None*, *timeout: float | None = None*, *api_version: str = '4'*, *per_page: int | None = None*, *pagination: str | None = None*, *order_by: str | None = None*, *user_agent: str = 'python-gitlab/8.4.0'*, *retry_transient_errors: bool = False*, *keep_base_url: bool = False*, ***kwargs:* Any)

Bases: object

Represents a GitLab server connection.

Parameters

- **url** (*str | None*) – The URL of the GitLab server (defaults to https://gitlab.com).
- **private_token** (*str | None*) – The user private token
- **oauth_token** (*str | None*) – An oauth token
- **job_token** (*str | None*) – A CI job token
- **ssl_verify** (*bool | str*) – Whether SSL certificates should be validated. If the value is a string, it is the path to a CA file used for certificate validation.
- **timeout** (*float | None*) – Timeout to use for requests to the GitLab server.
- **http_username** (*str | None*) – Username for HTTP authentication
- **http_password** (*str | None*) – Password for HTTP authentication
- **api_version** (*str*) – Gitlab API version to use (support for 4 only)
- **pagination** (*str | None*) – Can be set to 'keyset' to use keyset pagination
- **order_by** (*str | None*) – Set order_by globally
- **user_agent** (*str*) – A custom user agent to use for making HTTP requests.
- **retry_transient_errors** (*bool*) – Whether to retry after 500, 502, 503, 504 or 52x responses. Defaults to False.
- **keep_base_url** (*bool*) – keep user-provided base URL for pagination if it differs from response headers
- **session** (*requests.Session*) – HTTP Requests Session
- **backend** (*RequestsBackend*) – Backend that will be used to make http requests
- **per_page** (*int | None*)
- **kwargs** (*Any*)

property api_url: str

The computed API base URL.

property api_version: str

The API version used (4 only).

appearance

See *ApplicationAppearanceManager*

applications

See *ApplicationManager*

audit_events

See *AuditEventManager*

auth() → None

Performs an authentication using private token. Warns the user if a potentially misconfigured URL is detected on the client or server side.

The *user* attribute will hold a *gitlab.objects.CurrentUser* object on success.

Return type
None

broadcastmessages

See *BroadcastMessageManager*

bulk_import_entities

See *BulkImportAllEntityManager*

bulk_imports

See *BulkImportManager*

ci_lint

See *CiLintManager*

deploykeys

See *DeployKeyManager*

deploytokens

See *DeployTokenManager*

dockerfiles

See *DockerfileManager*

enable_debug(*mask_credentials: bool = True*) → None

Parameters
mask_credentials (*bool*)

Return type
None

events

See *EventManager*

features

See *FeatureManager*

classmethod from_config(*gitlab_id: str | None = None*, *config_files: list[str] | None = None*, ***kwargs: Any*) → *Gitlab*

Create a Gitlab connection from configuration files.

Parameters

- **gitlab_id** (*str | None*) – ID of the configuration section.
- **list[str]** (*config_files*) – List of paths to configuration files.
- **config_files** (*list[str] | None*)
- **kwargs** (*Any*)

Return type
Gitlab

kwargs:
session requests.Session: Custom requests Session

Returns
A Gitlab connection.

Raises
gitlab.config.GitlabDataError – If the configuration is not correct.

Parameters

- **gitlab_id** (*str | None*)
- **config_files** (*list[str] | None*)
- **kwargs** (Any)

Return type
Gitlab

geonodes

See *GeoNodeManager*

get_license(***kwargs: Any*) → dict[str, str | dict[str, str]]

Retrieve information about the current license.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

Returns
The current license information

Return type
dict[str, str | dict[str, str]]

gitignores

See *GitignoreManager*

gitlabciymls

See *GitlabciymlManager*

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/projecs')
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns
The requests object.

Raises
GitlabHttpError – When the return code is not 2xx

Return type
Response

http_get(*path: str*, *query_data: dict[str, Any] | None = None*, *streamed: bool = False*, *raw: bool = False*, ***kwargs: Any*) → dict[str, *Any*] | Response

Make a GET request to the Gitlab server.

Parameters

- **path** (*str*) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projecs')
- **query_data** (*dict[str, Any] | None*) – Data to send as query parameters
- **streamed** (*bool*) – Whether the data should be streamed
- **raw** (*bool*) – If True do not try to parse the output as json
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

Returns

A requests result object is streamed is True or the content type is not json. The parsed json data otherwise.

Raises

- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

Return type

dict[str, Any] | *Response*

http_head(*path: str*, *query_data: dict[str, Any] | None = None*, ***kwargs: Any*) → CaseInsensitiveDict[*Any*]

Make a HEAD request to the Gitlab server.

Parameters

- **path** (*str*) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projecs')
- **query_data** (*dict[str, Any] | None*) – Data to send as query parameters
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo, page, per_page)

Returns

A requests.header object

Raises

GitlabHttpError – When the return code is not 2xx

Return type

CaseInsensitiveDict[Any]

http_list(*path: str*, *query_data: dict[str, Any] | None = None*, ***, *iterator: bool | None = None*, *message_details: WarnMessageData | None = None*, ***kwargs: Any*) → *GitlabList* | list[dict[str, *Any*]]

Make a GET request to the Gitlab server for list-oriented queries.

Parameters

- **path** (*str*) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projects')
- **query_data** (*dict[str, Any] | None*) – Data to send as query parameters
- **iterator** (*bool | None*) – Indicate if should return a generator (True)
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo, page, per_page)
- **message_details** (*WarnMessageData | None*)

- ****kwargs**

Returns
A list of the objects returned by the server. If *iterator* is True and no pagination-related arguments (*page*, *per_page*, *get_all*) are defined then a GitlabList object (generator) is returned instead. This object will make API calls when needed to fetch the next items from the server.

Raises

- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

Return type
GitlabList | list[dict[str, Any]]

http_patch(*path: str*, ***, *query_data: dict[str, Any] | None = None*, *post_data: dict[str, Any] | bytes | None = None*, *raw: bool = False*, ***kwargs: Any*) → dict[str, *Any*] | Response

Make a PATCH request to the Gitlab server.

Parameters

- **path** (*str*) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projecs')
- **query_data** (*dict[str, Any] | None*) – Data to send as query parameters
- **post_data** (*dict[str, Any] | bytes | None*) – Data to send in the body (will be converted to json by default)
- **raw** (*bool*) – If True, do not convert post_data to json
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

Returns
The parsed json returned by the server.

Raises

- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

Return type
dict[str, Any] | *Response*

http_post(*path: str*, *query_data: dict[str, Any] | None = None*, *post_data: dict[str, Any] | None = None*, *raw: bool = False*, *files: dict[str, Any] | None = None*, ***kwargs: Any*) → dict[str, *Any*] | Response

Make a POST request to the Gitlab server.

Parameters

- **path** (*str*) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projecs')
- **query_data** (*dict[str, Any] | None*) – Data to send as query parameters
- **post_data** (*dict[str, Any] | None*) – Data to send in the body (will be converted to json by default)
- **raw** (*bool*) – If True, do not convert post_data to json
- **files** (*dict[str, Any] | None*) – The files to send to the server
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

Returns
The parsed json returned by the server if json is return, else the raw content

Raises

- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

Return type
dict[str, Any] | *Response*

http_put(*path: str*, *query_data: dict[str, Any] | None = None*, *post_data: dict[str, Any] | bytes | BinaryIO | None = None*, *raw: bool = False*, *files: dict[str, Any] | None = None*, ***kwargs: Any*) → dict[str, *Any*] | Response

Make a PUT request to the Gitlab server.

Parameters

- **path** (*str*) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projecs')
- **query_data** (*dict[str, Any] | None*) – Data to send as query parameters
- **post_data** (*dict[str, Any] | bytes | BinaryIO | None*) – Data to send in the body (will be converted to json by default)
- **raw** (*bool*) – If True, do not convert post_data to json
- **files** (*dict[str, Any] | None*) – The files to send to the server
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns
The parsed json returned by the server.

Raises

- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

Return type
dict[str, Any] | *Response*

http_request(*verb: str*, *path: str*, *query_data: dict[str, Any] | None = None*, *post_data: dict[str, Any] | bytes | BinaryIO | None = None*, *raw: bool = False*, *streamed: bool = False*, *files: dict[str, Any] | None = None*, *timeout: float | None = None*, *obey_rate_limit: bool = True*, *retry_transient_errors: bool | None = None*, *max_retries: int = 10*, *extra_headers: dict[str, Any] | None = None*, ***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/projecs')
- **query_data** (*dict[str, Any] | None*) – Data to send as query parameters
- **post_data** (*dict[str, Any] | bytes | BinaryIO | None*) – Data to send in the body (will be converted to json by default)
- **raw** (*bool*) – If True, do not convert post_data to json
- **streamed** (*bool*) – Whether the data should be streamed

- **files** (*dict[str, Any] | None*) – The files to send to the server
- **timeout** (*float | None*) – The timeout, in seconds, for the request
- **obey_rate_limit** (*bool*) – Whether to obey 429 Too Many Request responses. Defaults to True.
- **retry_transient_errors** (*bool | None*) – Whether to retry after 500, 502, 503, 504 or 52x responses. Defaults to False.
- **max_retries** (*int*) – Max retries after 429 or transient errors, set to -1 to retry forever. Defaults to 10.
- **extra_headers** (*dict[str, Any] | None*) – Add and override HTTP headers for the request.
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns
A requests result object.

Raises
GitlabHttpError – When the return code is not 2xx

Return type
Response

issues

See *IssueManager*

issues_statistics

See *IssuesStatisticsManager*

keys

See *KeyManager*

ldapgroups

See *LDAPGroupManager*

licenses

See *LicenseManager*

markdown(*text: str*, *gfm: bool = False*, *project: str | None = None*, ***kwargs: Any*) → str

Render an arbitrary Markdown document.

Parameters

- **text** (*str*) – The markdown text to render
- **gfm** (*bool*) – Render text using GitLab Flavored Markdown. Default is False
- **project** (*str | None*) – Full path of a project used a context when *gfm* is True
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabMarkdownError** – If the server cannot perform the request

Returns
The HTML rendering of the markdown text.

Return type
str

member_roles

See *MergeRequestManager*

classmethod merge_config(*options: dict[str, Any]*, *gitlab_id: str | None = None*, *config_files: list[str] | None = None*) → *Gitlab*

Create a Gitlab connection by merging configuration with the following precedence:

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

Parameters

- **options** (*dict[str, Any]*) – A dictionary of explicitly provided key-value options.
- **gitlab_id** (*str | None*) – ID of the configuration section.
- **config_files** (*list[str] | None*) – List of paths to configuration files.

Returns
A Gitlab connection.

Return type
(*gitlab.Gitlab*)

Raises
gitlab.config.GitlabDataError – If the configuration is not correct.

mergerequests

See *MergeRequestManager*

namespaces

See *NamespaceManager*

notificationsettings

See *NotificationSettingsManager*

pagesdomains

See *PagesDomainManager*

personal_access_tokens

See *PersonalAccessTokenManager*

projects

See *ProjectManager*

registry_repositories

See *RegistryRepositoryManager*

runners

See *RunnerManager*

runners_all

See *RunnerManager*

search(*scope: str*, *search: str*, ***kwargs:* Any) → *GitlabList* | list[dict[str, *Any*]]

Search GitLab resources matching the provided string.'

Parameters

- **scope** (*str*) – Scope of the search
- **search** (*str*) – Search string
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns

A list of dicts describing the resources found.

Return type

GitlabList | list[dict[str, Any]]

service_accounts

See *ServiceAccountManager*

set_license(*license: str*, ***kwargs:* Any) → dict[str, *Any*]

Add a new license.

Parameters

- **license** (*str*) – The license string
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabPostError** – If the server cannot perform the request

Returns

The new license information

Return type

dict[str, Any]

settings

See *ApplicationSettingsManager*

sidekiq

See *SidekiqManager*

snippets

See *SnippetManager*

ssl_verify

Whether SSL certificates should be validated

statistics

See *ApplicationStatisticsManager*

timeout

Timeout to use for requests to gitlab server

todos

See *TodoManager*

topics

See *TopicManager*

property url: str

The user-provided server URL.

user_activities

See *UserActivitiesManager*

users

See *UserManager*

variables

See *VariableManager*

version() → tuple[str, str]

Returns the version and revision of the gitlab server.

Note that self.version and self.revision will be set on the gitlab object.

Returns

The server version and server revision.
('unknown', 'unknown') if the server doesn't perform as expected.

Return type
tuple[str, str]

exception gitlab.**GitlabActivateError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabAttachFileError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)

- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabAuthenticationError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabBanError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabBlockError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabBuildCancelError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabCancelError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabBuildEraseError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: GitlabRetryError

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabBuildPlayError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: GitlabRetryError

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabBuildRetryError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: GitlabRetryError

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabCancelError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: GitlabOperationError

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabCherryPickError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: GitlabOperationError

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabCiLintError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabConnectionError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabCreateError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabDeactivateError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabDeleteError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabDeploymentApprovalError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: Exception

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabFollowError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabGetError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: GitlabOperationError

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabGroupTransferError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: GitlabOperationError

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabHeadError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: GitlabOperationError

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabHookTestError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: GitlabOperationError

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabHousekeepingError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: GitlabOperationError

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabHttpError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabImportError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabInvitationError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabJobCancelError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabCancelError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabJobEraseError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabRetryError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabJobPlayError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabRetryError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabJobRetryError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabRetryError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabLicenseError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

class gitlab.GitlabList(*gl: Gitlab*, *url: str*, *query_data: dict[str, Any]*, *get_next: bool = True*, ***kwargs: Any*)

Bases: object

Generator representing a list of remote objects.

The object handles the links returned by a query to the API, and will call the API again when needed.

Parameters

- **gl** (*Gitlab*)
- **url** (*str*)
- **query_data** (*dict[str, Any]*)
- **get_next** (*bool*)
- **kwargs** (*Any*)

property current_page: int

The current page number.

next() → dict[str, *Any*]

Return type
dict[str, Any]

property next_page: int | None

The next page number.

If None, the current page is the last.

property per_page: int | None

The number of items per page.

property prev_page: int | None

The previous page number.

If None, the current page is the first.

property total: int | None

The total number of items.

property total_pages: int | None

The total number of pages.

exception gitlab.GitlabListError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabMRApprovalError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabMRClosedError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabMRForbiddenError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabMROnBuildSuccessError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabMRRebaseError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabMRResetApprovalError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabMarkdownError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabOperationError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabOwnershipError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabParsingError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabPipelineCancelError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabCancelError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabPipelinePlayError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabRetryError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabPipelineRetryError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabRetryError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabProjectDeployKeyError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabPromoteError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabProtectError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabRenderError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabRepairError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabRestoreError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabRetryError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabRevertError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabRotateError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabSearchError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabSetError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabStopError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabSubscribeError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabTimeTrackingError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: GitlabOperationError

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabTodoError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: GitlabOperationError

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabTopicMergeError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: GitlabOperationError

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabTransferProjectError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: GitlabOperationError

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.GitlabUnbanError(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: GitlabOperationError

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabUnblockError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabUnfollowError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabUnsubscribeError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabUpdateError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabUploadError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabUserApproveError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabUserRejectError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.**GitlabVerifyError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

class gitlab.**GraphQL**(*url: str | None = None*, ***, *token: str | None = None*, *ssl_verify: bool | str = True*, *client: Client | None = None*, *timeout: float | None = None*, *user_agent: str = 'python-gitlab/8.4.0'*, *fetch_schema_from_transport: bool = False*, *max_retries: int = 10*, *obey_rate_limit: bool = True*, *retry_transient_errors: bool = False*)

Bases: _BaseGraphQL

Parameters

- **url** (*str | None*)
- **token** (*str | None*)
- **ssl_verify** (*bool | str*)
- **client** (*httpx.Client | None*)
- **timeout** (*float | None*)
- **user_agent** (*str*)
- **fetch_schema_from_transport** (*bool*)
- **max_retries** (*int*)
- **obey_rate_limit** (*bool*)
- **retry_transient_errors** (*bool*)

execute(*request: str*, **args:* Any, ***kwargs:* Any) → *Any*

Parameters

- **request** (*str*)
- **args** (Any)
- **kwargs** (Any)

Return type
Any

exception gitlab.**RedirectError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

class gitlab.**Gitlab**(*url: str | None = None*, *private_token: str | None = None*, *oauth_token: str | None = None*, *job_token: str | None = None*, *ssl_verify: bool | str = True*, *http_username: str | None = None*, *http_password: str | None = None*, *timeout: float | None = None*, *api_version: str = '4'*, *per_page: int | None = None*, *pagination: str | None = None*, *order_by: str | None = None*, *user_agent: str = 'python-gitlab/8.4.0'*, *retry_transient_errors: bool = False*, *keep_base_url: bool = False*, ***kwargs:* Any)

Bases: object

Represents a GitLab server connection.

Parameters

- **url** (*str | None*) – The URL of the GitLab server (defaults to https://gitlab.com).
- **private_token** (*str | None*) – The user private token
- **oauth_token** (*str | None*) – An oauth token
- **job_token** (*str | None*) – A CI job token
- **ssl_verify** (*bool | str*) – Whether SSL certificates should be validated. If the value is a string, it is the path to a CA file used for certificate validation.
- **timeout** (*float | None*) – Timeout to use for requests to the GitLab server.
- **http_username** (*str | None*) – Username for HTTP authentication
- **http_password** (*str | None*) – Password for HTTP authentication
- **api_version** (*str*) – Gitlab API version to use (support for 4 only)
- **pagination** (*str | None*) – Can be set to 'keyset' to use keyset pagination
- **order_by** (*str | None*) – Set order_by globally
- **user_agent** (*str*) – A custom user agent to use for making HTTP requests.
- **retry_transient_errors** (*bool*) – Whether to retry after 500, 502, 503, 504 or 52x responses. Defaults to False.
- **keep_base_url** (*bool*) – keep user-provided base URL for pagination if it differs from response headers
- **session** (*requests.Session*) – HTTP Requests Session
- **backend** (*RequestsBackend*) – Backend that will be used to make http requests
- **per_page** (*int | None*)
- **kwargs** (*Any*)

property api_url: str

The computed API base URL.

property api_version: str

The API version used (4 only).

appearance

See *ApplicationAppearanceManager*

applications

See *ApplicationManager*

audit_events

See *AuditEventManager*

auth() → None

Performs an authentication using private token. Warns the user if a potentially misconfigured URL is detected on the client or server side.

The *user* attribute will hold a *gitlab.objects.CurrentUser* object on success.

Return type
None

broadcastmessages

See *BroadcastMessageManager*

bulk_import_entities

See *BulkImportAllEntityManager*

bulk_imports

See *BulkImportManager*

ci_lint

See *CiLintManager*

deploykeys

See *DeployKeyManager*

deploytokens

See *DeployTokenManager*

dockerfiles

See *DockerfileManager*

enable_debug(*mask_credentials: bool = True*) → None

Parameters
mask_credentials (*bool*)

Return type
None

events

See *EventManager*

features

See *FeatureManager*

classmethod from_config(*gitlab_id: str | None = None*, *config_files: list[str] | None = None*, ***kwargs: Any*) → *Gitlab*

Create a Gitlab connection from configuration files.

Parameters

- **gitlab_id** (*str | None*) – ID of the configuration section.
- **list[str]** (*config_files*) – List of paths to configuration files.
- **config_files** (*list[str] | None*)
- **kwargs** (*Any*)

Return type
Gitlab

kwargs:
session requests.Session: Custom requests Session

Returns
A Gitlab connection.

Raises
gitlab.config.GitlabDataError – If the configuration is not correct.

Parameters

- **gitlab_id** (*str | None*)
- **config_files** (*list[str] | None*)
- **kwargs** (Any)

Return type
Gitlab

geonodes

See *GeoNodeManager*

get_license(***kwargs: Any*) → dict[str, str | dict[str, str]]

Retrieve information about the current license.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

Returns
The current license information

Return type
dict[str, str | dict[str, str]]

gitignores

See *GitignoreManager*

gitlabciymls

See *GitlabciymlManager*

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/projecs')
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns
The requests object.

Raises
GitlabHttpError – When the return code is not 2xx

Return type
Response

http_get(*path: str*, *query_data: dict[str, Any] | None = None*, *streamed: bool = False*, *raw: bool = False*, ***kwargs: Any*) → dict[str, *Any*] | Response

Make a GET request to the Gitlab server.

Parameters

- **path** (*str*) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projecs')
- **query_data** (*dict[str, Any] | None*) – Data to send as query parameters
- **streamed** (*bool*) – Whether the data should be streamed
- **raw** (*bool*) – If True do not try to parse the output as json
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

Returns
A requests result object is streamed is True or the content type is not json. The parsed json data otherwise.

Raises

- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

Return type
dict[str, Any] | *Response*

http_head(*path: str*, *query_data: dict[str, Any] | None = None*, ***kwargs: Any*) → CaseInsensitiveDict[*Any*]

Make a HEAD request to the Gitlab server.

Parameters

- **path** (*str*) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projecs')
- **query_data** (*dict[str, Any] | None*) – Data to send as query parameters
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo, page, per_page)

Returns
A requests.header object

Raises
GitlabHttpError – When the return code is not 2xx

Return type
CaseInsensitiveDict[Any]

http_list(*path: str*, *query_data: dict[str, Any] | None = None*, ***, *iterator: bool | None = None*, *message_details: WarnMessageData | None = None*, ***kwargs: Any*) → *GitlabList* | list[dict[str, *Any*]]

Make a GET request to the Gitlab server for list-oriented queries.

Parameters

- **path** (*str*) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projects')

- **query_data** (*dict[str, Any] | None*) – Data to send as query parameters
- **iterator** (*bool | None*) – Indicate if should return a generator (True)
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo, page, per_page)
- **message_details** (*WarnMessageData | None*)
- ****kwargs**

Returns
A list of the objects returned by the server. If *iterator* is True and no pagination-related arguments (*page*, *per_page*, *get_all*) are defined then a GitlabList object (generator) is returned instead. This object will make API calls when needed to fetch the next items from the server.

Raises
- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

Return type
GitlabList | list[dict[str, Any]]

http_patch(*path: str*, ***, *query_data: dict[str, Any] | None = None*, *post_data: dict[str, Any] | bytes | None = None*, *raw: bool = False*, ***kwargs: Any*) → dict[str, *Any*] | Response

Make a PATCH request to the Gitlab server.

Parameters
- **path** (*str*) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projecs')
- **query_data** (*dict[str, Any] | None*) – Data to send as query parameters
- **post_data** (*dict[str, Any] | bytes | None*) – Data to send in the body (will be converted to json by default)
- **raw** (*bool*) – If True, do not convert post_data to json
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

Returns
The parsed json returned by the server.

Raises
- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

Return type
dict[str, Any] | *Response*

http_post(*path: str*, *query_data: dict[str, Any] | None = None*, *post_data: dict[str, Any] | None = None*, *raw: bool = False*, *files: dict[str, Any] | None = None*, ***kwargs: Any*) → dict[str, *Any*] | Response

Make a POST request to the Gitlab server.

Parameters
- **path** (*str*) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projecs')
- **query_data** (*dict[str, Any] | None*) – Data to send as query parameters

- **post_data** (*dict[str, Any] | None*) – Data to send in the body (will be converted to json by default)
- **raw** (*bool*) – If True, do not convert post_data to json
- **files** (*dict[str, Any] | None*) – The files to send to the server
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns
The parsed json returned by the server if json is return, else the raw content

Raises
- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

Return type
dict[str, Any] | *Response*

http_put(*path: str*, *query_data: dict[str, Any] | None = None*, *post_data: dict[str, Any] | bytes | BinaryIO | None = None*, *raw: bool = False*, *files: dict[str, Any] | None = None*, ***kwargs: Any*) → dict[str, *Any*] | Response

Make a PUT request to the Gitlab server.

Parameters
- **path** (*str*) – Path or full URL to query ('/projects' or 'http://whatever/v4/api/projecs')
- **query_data** (*dict[str, Any] | None*) – Data to send as query parameters
- **post_data** (*dict[str, Any] | bytes | BinaryIO | None*) – Data to send in the body (will be converted to json by default)
- **raw** (*bool*) – If True, do not convert post_data to json
- **files** (*dict[str, Any] | None*) – The files to send to the server
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns
The parsed json returned by the server.

Raises
- **GitlabHttpError** – When the return code is not 2xx
- **GitlabParsingError** – If the json data could not be parsed

Return type
dict[str, Any] | *Response*

http_request(*verb: str*, *path: str*, *query_data: dict[str, Any] | None = None*, *post_data: dict[str, Any] | bytes | BinaryIO | None = None*, *raw: bool = False*, *streamed: bool = False*, *files: dict[str, Any] | None = None*, *timeout: float | None = None*, *obey_rate_limit: bool = True*, *retry_transient_errors: bool | None = None*, *max_retries: int = 10*, *extra_headers: dict[str, Any] | None = None*, ***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/projecs')
- **query_data** (*dict[str, Any] | None*) – Data to send as query parameters
- **post_data** (*dict[str, Any] | bytes | BinaryIO | None*) – Data to send in the body (will be converted to json by default)
- **raw** (*bool*) – If True, do not convert post_data to json
- **streamed** (*bool*) – Whether the data should be streamed
- **files** (*dict[str, Any] | None*) – The files to send to the server
- **timeout** (*float | None*) – The timeout, in seconds, for the request
- **obey_rate_limit** (*bool*) – Whether to obey 429 Too Many Request responses. Defaults to True.
- **retry_transient_errors** (*bool | None*) – Whether to retry after 500, 502, 503, 504 or 52x responses. Defaults to False.
- **max_retries** (*int*) – Max retries after 429 or transient errors, set to -1 to retry forever. Defaults to 10.
- **extra_headers** (*dict[str, Any] | None*) – Add and override HTTP headers for the request.
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns
A requests result object.

Raises
GitlabHttpError – When the return code is not 2xx

Return type
Response

issues

See *IssueManager*

issues_statistics

See *IssuesStatisticsManager*

keys

See *KeyManager*

ldapgroups

See *LDAPGroupManager*

licenses

See *LicenseManager*

markdown(*text: str*, *gfm: bool = False*, *project: str | None = None*, ***kwargs:* Any) → str

Render an arbitrary Markdown document.

Parameters

- **text** (*str*) – The markdown text to render
- **gfm** (*bool*) – Render text using GitLab Flavored Markdown. Default is False
- **project** (*str | None*) – Full path of a project used a context when *gfm* is True

- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabMarkdownError** – If the server cannot perform the request

Returns
The HTML rendering of the markdown text.

Return type
str

member_roles

See *MergeRequestManager*

classmethod merge_config(*options: dict[str, Any]*, *gitlab_id: str | None = None*, *config_files: list[str] | None = None*) → *Gitlab*

Create a Gitlab connection by merging configuration with the following precedence:

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

Parameters

- **options** (*dict[str, Any]*) – A dictionary of explicitly provided key-value options.
- **gitlab_id** (*str | None*) – ID of the configuration section.
- **config_files** (*list[str] | None*) – List of paths to configuration files.

Returns
A Gitlab connection.

Return type
(*gitlab.Gitlab*)

Raises
gitlab.config.GitlabDataError – If the configuration is not correct.

mergerequests

See *MergeRequestManager*

namespaces

See *NamespaceManager*

notificationsettings

See *NotificationSettingsManager*

pagesdomains

See *PagesDomainManager*

personal_access_tokens

See *PersonalAccessTokenManager*

projects

See *ProjectManager*

registry_repositories

See *RegistryRepositoryManager*

runners

See *RunnerManager*

runners_all

See *RunnerManager*

search(*scope: str*, *search: str*, ***kwargs:* Any) → *GitlabList* | list[dict[str, *Any*]]

Search GitLab resources matching the provided string.'

Parameters

- **scope** (*str*) – Scope of the search
- **search** (*str*) – Search string
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns

A list of dicts describing the resources found.

Return type

GitlabList | list[dict[str, Any]]

service_accounts

See *ServiceAccountManager*

set_license(*license: str*, ***kwargs:* Any) → dict[str, *Any*]

Add a new license.

Parameters

- **license** (*str*) – The license string
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabPostError** – If the server cannot perform the request

Returns

The new license information

Return type

dict[str, Any]

settings

See *ApplicationSettingsManager*

sidekiq

See *SidekiqManager*

snippets

See *SnippetManager*

ssl_verify

Whether SSL certificates should be validated

statistics

See *ApplicationStatisticsManager*

timeout

Timeout to use for requests to gitlab server

todos

See *TodoManager*

topics

See *TopicManager*

property url: str

The user-provided server URL.

user: objects.CurrentUser | None

user_activities

See *UserActivitiesManager*

users

See *UserManager*

variables

See *VariableManager*

version() → tuple[str, str]

Returns the version and revision of the gitlab server.

Note that self.version and self.revision will be set on the gitlab object.

Returns

The server version and server revision.
('unknown', 'unknown') if the server doesn't perform as expected.

Return type
tuple[str, str]

class gitlab.**GitlabList**(*gl: Gitlab*, *url: str*, *query_data: dict[str, Any]*, *get_next: bool = True*, ***kwargs: Any*)

Bases: object

Generator representing a list of remote objects.

The object handles the links returned by a query to the API, and will call the API again when needed.

Parameters

- **gl** (Gitlab)
- **url** (*str*)
- **query_data** (*dict[str, Any]*)
- **get_next** (*bool*)
- **kwargs** (*Any*)

property current_page: int

The current page number.

next() → dict[str, *Any*]

Return type
dict[str, Any]

property next_page: int | None

The next page number.

If None, the current page is the last.

property per_page: int | None

The number of items per page.

property prev_page: int | None

The previous page number.

If None, the current page is the first.

property total: int | None

The total number of items.

property total_pages: int | None

The total number of pages.

7.7.2 Subpackages

gitlab.v4 package

Submodules

gitlab.v4.objects module

class gitlab.v4.objects.**Any**(**args*, ***kwargs*)

Bases: object

Special type indicating an unconstrained type.

- Any is compatible with every type.
- Any assumed to have all methods.
- All values assumed to be instances of Any.

Note that all the above statements are true from the point of view of static type checkers. At runtime, Any should not be used with instance checks.

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

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

Bases: *SaveMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ApplicationAppearanceManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*ApplicationAppearance*], *UpdateMixin*[*ApplicationAppearance*]

Object Creation

Object update

Optional attributes for object update:

- title
- description
- logo
- header_logo
- favicon
- new_project_guidelines
- header_message
- footer_message
- message_background_color
- message_font_color
- email_header_and_footer_enabled

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

update(*id: str | int | None = None*, *new_data: dict[str, Any] | None = None*, ***kwargs: Any*) → dict[str, *Any*]

Update an object on the server.

Parameters

- **id** (*str | int | None*) – ID of the object to update (can be None if not required)

- **new_data** (*dict[str, Any] | None*) – the update data for the object
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns
The new object data (*not* a RESTObject)

Raises
- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUpdateError** – If the server cannot perform the request

Return type
dict[str, Any]

class gitlab.v4.objects.**ApplicationManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *ListMixin*[*Application*], *CreateMixin*[*Application*], *DeleteMixin*[*Application*]

Object Creation

Required attributes for object create:
- name
- redirect_uri
- scopes

Optional attributes for object create:
- confidential

Object update

Parameters
- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *RESTObject*

Parameters
- **manager** (RESTManager*[*Any*]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ApplicationSettingsManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *GetWithoutIdMixin*[*ApplicationSettings*], *UpdateMixin*[*ApplicationSettings*]

Object Creation

Object update

Optional attributes for object update:
- id
- default_projects_limit
- signup_enabled

- silent_mode_enabled
- password_authentication_enabled_for_web
- gravatar_enabled
- sign_in_text
- created_at
- updated_at
- home_page_url
- default_branch_protection
- restricted_visibility_levels
- max_attachment_size
- session_expire_delay
- default_project_visibility
- default_snippet_visibility
- default_group_visibility
- outbound_local_requests_whitelist
- disabled_oauth_sign_in_sources
- domain_whitelist
- domain_blacklist_enabled
- domain_blacklist
- domain_allowlist
- domain_denylist_enabled
- domain_denylist
- external_authorization_service_enabled
- external_authorization_service_url
- external_authorization_service_default_label
- external_authorization_service_timeout
- import_sources
- user_oauth_applications
- after_sign_out_path
- container_registry_token_expire_delay
- repository_storages
- plantuml_enabled
- plantuml_url
- terminal_max_session_time
- polling_interval_multiplier
- rsa_key_restriction

- dsa_key_restriction
- ecdsa_key_restriction
- ed25519_key_restriction
- first_day_of_week
- enforce_terms
- terms
- performance_bar_allowed_group_id
- instance_statistics_visibility_private
- user_show_add_ssh_key_message
- file_template_project_id
- local_markdown_version
- asset_proxy_enabled
- asset_proxy_url
- asset_proxy_whitelist
- asset_proxy_allowlist
- geo_node_allowed_ips
- allow_local_requests_from_hooks_and_services
- allow_local_requests_from_web_hooks_and_services
- allow_local_requests_from_system_hooks

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

update(*id: str | int | None = None*, *new_data: dict[str, Any] | None = None*, ***kwargs: Any*) → dict[str, *Any*]

Update an object on the server.

Parameters

- **id** (*str | int | None*) – ID of the object to update (can be None if not required)
- **new_data** (*dict[str, Any] | None*) – the update data for the object
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns

The new object data (*not* a RESTObject)

Raises

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

Return type

dict[str, Any]

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ApplicationStatisticsManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*ApplicationStatistics*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

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

Bases: *RetrieveMixin*[*AuditEvent*]

Object listing filters

- created_after
- created_before
- entity_type
- entity_id

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**BroadcastMessageManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*BroadcastMessage*]

Object Creation

Required attributes for object create:

- message

Optional attributes for object create:

- starts_at
- ends_at
- color
- font
- target_access_levels

Object update

Optional attributes for object update:

- message
- starts_at
- ends_at
- color
- font
- target_access_levels

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RefreshMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)

- **lazy** (*bool*)

entities: *BulkImportEntityManager*

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**BulkImportAllEntityManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*BulkImportAllEntity*]

Object listing filters

- sort
- status

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RefreshMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**BulkImportEntityManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*BulkImportEntity*]

Object listing filters

- sort
- status

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)

- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**BulkImportManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CreateMixin*[*BulkImport*], *RetrieveMixin*[*BulkImport*]

Object listing filters

- sort
- status

Object Creation

Required attributes for object create:

- configuration
- entities

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str,* Any])
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**CiLintManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CreateMixin*[*CiLint*]

Object Creation

Required attributes for object create:

- content

Optional attributes for object create:

- include_merged_yaml
- include_jobs

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

validate(**args:* Any, ***kwargs:* Any) → None

Raise an error if the CI Lint results are not valid.

This is a custom python-gitlab method to wrap lint endpoints.

Parameters

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

Return type
None

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

Bases: *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

emails: *CurrentUserEmailManager*

gpgkeys: *CurrentUserGPGKeyManager*

keys: *CurrentUserKeyManager*

runners: *CurrentUserRunnerManager*

status: *CurrentUserStatusManager*

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**CurrentUserEmailManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*CurrentUserEmail*], *CreateMixin*[*CurrentUserEmail*], *DeleteMixin*[*CurrentUserEmail*]

Object Creation

Required attributes for object create:

- email

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

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

Bases: *RetrieveMixin*[*CurrentUserGPGKey*], *CreateMixin*[*CurrentUserGPGKey*], *DeleteMixin*[*CurrentUserGPGKey*]

Object Creation

Required attributes for object create:

- key

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**CurrentUserKeyManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*CurrentUserKey*], *CreateMixin*[*CurrentUserKey*], *DeleteMixin*[*CurrentUserKey*]

Object Creation

Required attributes for object create:

- title
- key

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**CurrentUserManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *GetWithoutIdMixin*[*CurrentUser*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str,* Any])
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**CurrentUserRunnerManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CreateMixin*[*CurrentUserRunner*]

Object Creation

Required attributes for object create:

- runner_type

Optional attributes for object create:

- group_id
- project_id
- description
- paused
- locked
- run_untagged
- tag_list
- access_level
- maximum_timeout
- maintenance_note

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**CurrentUserStatusManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*CurrentUserStatus*], *UpdateMixin*[*CurrentUserStatus*]

Object Creation

Object update

Optional attributes for object update:

- emoji
- message

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**DeployKeyManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CreateMixin*[*DeployKey*], *ListMixin*[*DeployKey*]

Object Creation

Required attributes for object create:

- title
- key

Optional attributes for object create:

- expires_at

Object update

Parameters

- **gl** (*Gitlab*)

- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**DeployTokenManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *ListMixin*[*DeployToken*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**DockerfileManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*Dockerfile*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)

- **lazy** (*bool*)

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

Bases: *ListMixin*[*Event*]

Object listing filters

- action
- target_type
- before
- after
- sort
- scope

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**FeatureManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*Feature*], *DeleteMixin*[*Feature*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

set(*name: str*, *value: bool | int*, *feature_group: str | None = None*, *user: str | None = None*, *group: str | None = None*, *project: str | None = None*, ***kwargs: Any*) → *Feature*

Create or update the object.

Parameters

- **name** (*str*) – The value to set for the object
- **value** (*bool | int*) – The value to set for the object
- **feature_group** (*str | None*) – A feature group name

- **user** (*str | None*) – A GitLab username
- **group** (*str | None*) – A GitLab group
- **project** (*str | None*) – A GitLab project in form group/project
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabSetError** – If an error occurred

Returns
The created/updated attribute

Return type
Feature

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GenericPackageManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RESTManager*[*GenericPackage*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

download(*package_name: str*, *package_version: str*, *file_name: str*, *streamed: Literal[False] = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → bytes

download(*package_name: str*, *package_version: str*, *file_name: str*, *streamed: bool = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[True] = True*, ***kwargs: Any*) → Iterator[*Any*]

download(*package_name: str*, *package_version: str*, *file_name: str*, *streamed: Literal[True] = True*, *action: Callable[[bytes], Any] | None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → None

Download a generic package.

Parameters

- **package_name** (*str*) – The package name.
- **package_version** (*str*) – The package version.
- **file_name** (*str*) – The name of the file in the registry

- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable[[bytes], Any] | None*) – Callable responsible of dealing with chunk of data
- **chunk_size** (*int*) – Size of each chunk
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
The package content if streamed is False, None otherwise

Return type
bytes | *Iterator*[Any] | None

upload(*package_name: str*, *package_version: str*, *file_name: str*, *path: str | Path | None = None*, *select: str | None = None*, *data: bytes | BinaryIO | None = None*, ***kwargs:* Any) → *GenericPackage*

Upload a file as a generic package.

Parameters

- **package_name** (*str*) – The package name. Must follow generic package name regex rules
- **package_version** (*str*) – The package version. Must follow semantic version regex rules
- **file_name** (*str*) – The name of the file as uploaded in the registry
- **path** (*str | Path | None*) – The path to a local file to upload
- **select** (*str | None*) – GitLab API accepts a value of 'package_file'
- **data** (*bytes | BinaryIO | None*)
- **kwargs** (Any)

Raises

- **GitlabConnectionError** – If the server cannot be reached
- **GitlabUploadError** – If the file upload fails
- **GitlabUploadError** – If `path` cannot be read
- **GitlabUploadError** – If both `path` and `data` are passed

Returns
An object storing the metadata of the uploaded package.

Return type
GenericPackage

https://docs.gitlab.com/ee/user/packages/generic_packages/

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

repair(***kwargs: Any*) → None

Repair the OAuth authentication of the geo node.

Parameters
****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

Raises

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

Return type
None

status(***kwargs: Any*) → Dict[str, *Any*]

Get the status of the geo node.

Parameters
****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
The status of the geo node

Return type
Dict[str, Any]

class gitlab.v4.objects.**GeoNodeManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*GeoNode*], *UpdateMixin*[*GeoNode*], *DeleteMixin*[*GeoNode*]

Object Creation

Object update

Optional attributes for object update:

- `enabled`
- `url`
- `files_max_capacity`
- `repos_max_capacity`

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

current_failures(***kwargs: Any*) → List[Dict[str, *Any*]]

Get the list of failures on the current geo node.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

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

Returns
The list of failures

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

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[Any]*, *attrs: dict[str, Any]*, ***, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *RESTObject*

Parameters
- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

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

Bases: *RetrieveMixin*[*Gitignore*]

Object Creation

Object update

Parameters
- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GitlabciymlManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*Gitlabciyml*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

access_tokens: *GroupAccessTokenManager*

accessrequests: *GroupAccessRequestManager*

approval_rules: *GroupApprovalRuleManager*

audit_events: *GroupAuditEventManager*

badges: *GroupBadgeManager*

billable_members: *GroupBillableMemberManager*

boards: *GroupBoardManager*

clusters: *GroupClusterManager*

customattributes: *GroupCustomAttributeManager*

deploytokens: *GroupDeployTokenManager*

descendant_groups: *GroupDescendantGroupManager*

epics: *GroupEpicManager*

exports: *GroupExportManager*

hooks: *GroupHookManager*

imports: *GroupImportManager*

invitations: *GroupInvitationManager*

issues: *GroupIssueManager*

issues_statistics: *GroupIssuesStatisticsManager*

iterations: *GroupIterationManager*

labels: *GroupLabelManager*

ldap_group_links: *GroupLDAPGroupLinkManager*

ldap_sync(***kwargs: Any*) → None

Sync LDAP groups.

Parameters

****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

Return type

None

member_roles: *GroupMemberRoleManager*

members: *GroupMemberManager*

members_all: *GroupMemberAllManager*

mergerequests: *GroupMergeRequestManager*

milestones: *GroupMilestoneManager*

notificationsettings: *GroupNotificationSettingsManager*

packages: *GroupPackageManager*

projects: *GroupProjectManager*

protectedbranches: GroupProtectedBranchManager

pushrules: *GroupPushRulesManager*

registry_repositories: *GroupRegistryRepositoryManager*

restore(***kwargs: Any*) → None

Restore a group marked for deletion..

Parameters

****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Return type
None

runners: *GroupRunnerManager*

saml_group_links: *GroupSAMLGroupLinkManager*

search(*scope: str*, *search: str*, ***kwargs:* Any) → *GitlabList* | list[dict[str, *Any*]]

Search the group resources matching the provided string.

Parameters

- **scope** (*str*) – Scope of the search
- **search** (*str*) – Search string
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
A list of dicts describing the resources found.

Return type
GitlabList | list[dict[str, Any]]

service_accounts: *GroupServiceAccountManager*

share(*group_id: int*, *group_access: int*, *expires_at: str | None = None*, ***kwargs:* Any) → None

Share the group with a group.

Parameters

- **group_id** (*int*) – ID of the group.
- **group_access** (*int*) – Access level for the group.
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)
- **expires_at** (*str | None*)
- ****kwargs**

Raises

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

Returns
Group

Return type
None

shared_projects: *SharedProjectManager*

subgroups: *GroupSubgroupManager*

transfer(*group_id: int | None = None*, ***kwargs: Any*) → None

Transfer the group to a new parent group or make it a top-level group.

Requires GitLab 14.6.

Parameters

- **group_id** (*int | None*) – ID of the new parent group. When not specified, the group to transfer is instead turned into a top-level group.
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGroupTransferError** – If the group could not be transferred

Return type
None

transfer_project(*project_id: int*, ***kwargs: Any*) → None

Transfer a project to this group.

Parameters

- **to_project_id** – ID of the project to transfer
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)
- **project_id** (*int*)
- ****kwargs**

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabTransferProjectError** – If the project could not be transferred

Return type
None

unshare(*group_id: int*, ***kwargs: Any*) → None

Delete a shared group link within a group.

Parameters

- **group_id** (*int*) – ID of the group.
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Return type
None

variables: *GroupVariableManager*

wikis: *GroupWikiManager*

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

Bases: *AccessRequestMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupAccessRequestManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*GroupAccessRequest*], *CreateMixin*[*GroupAccessRequest*], *DeleteMixin*[*GroupAccessRequest*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, ObjectRotateMixin, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupAccessTokenManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CreateMixin*[*GroupAccessToken*], *DeleteMixin*[*GroupAccessToken*], *RetrieveMixin*[*GroupAccessToken*], RotateMixin[*GroupAccessToken*]

Object Creation

Required attributes for object create:

- name
- scopes

Optional attributes for object create:

- access_level
- expires_at

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupApprovalRuleManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*GroupApprovalRule*], *CreateMixin*[*GroupApprovalRule*], *UpdateMixin*[*GroupApprovalRule*]

Object Creation

Required attributes for object create:

- name
- approvals_required

Optional attributes for object create:

- user_ids
- group_ids
- rule_type

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupAuditEventManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*GroupAuditEvent*]

Object listing filters

- created_after
- created_before

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupBadgeManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *BadgeRenderMixin*[*GroupBadge*], *CRUDMixin*[*GroupBadge*]

Object Creation

Required attributes for object create:

- `link_url`
- `image_url`

Object update

Optional attributes for object update:

- `link_url`
- `image_url`

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

memberships: *GroupBillableMemberMembershipManager*

class gitlab.v4.objects.**GroupBillableMemberManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*GroupBillableMember*], *DeleteMixin*[*GroupBillableMember*]

Object listing filters

- `search`

- sort

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupBillableMemberMembershipManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *ListMixin*[*GroupBillableMemberMembership*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

lists: *GroupBoardListManager*

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)

- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupBoardListManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*GroupBoardList*]

Object Creation

Mutually exclusive attributes for object create:

- `label_id`
- `assignee_id`
- `milestone_id`
- `iteration_id`

Object update

Required attributes for object update:

- `position`

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *CRUDMixin*[*GroupBoard*]

Object Creation

Required attributes for object create:

- `name`

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupClusterManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CRUDMixin*[*GroupCluster*]

Object Creation

Required attributes for object create:

- `name`
- `platform_kubernetes_attributes`

Optional attributes for object create:

- `domain`
- `enabled`
- `managed`
- `environment_scope`

Object update

Optional attributes for object update:

- `name`
- `domain`
- `management_project_id`
- `platform_kubernetes_attributes`
- `environment_scope`

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

create(*data: dict[str,* Any*]* | *None* = *None*, ***kwargs:* Any) → *GroupCluster*

Create a new object.

Parameters

- **data** (*dict[str,* Any*] | None*) – Parameters to send to the server to create the resource
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo or 'ref_name', 'stage', 'name', 'all')

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

Returns

A new instance of the manage object class build with the data sent by the server

Return type

GroupCluster

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupCustomAttributeManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*GroupCustomAttribute*], *SetMixin*[*GroupCustomAttribute*], *DeleteMixin*[*GroupCustomAttribute*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupDeployTokenManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*GroupDeployToken*], *CreateMixin*[*GroupDeployToken*], *DeleteMixin*[*GroupDeployToken*]

Object listing filters

- `scopes`

Object Creation

Required attributes for object create:

- `name`
- `scopes`

Optional attributes for object create:

- `expires_at`
- `username`

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupDescendantGroupManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: SubgroupBaseManager[*GroupDescendantGroup*]

This manager inherits from GroupSubgroupManager as descendant groups share all attributes with subgroups, except the path and object class.

Object listing filters

- skip_groups
- all_available
- search
- order_by
- sort
- statistics
- owned
- with_custom_attributes
- min_access_level

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *SaveMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

issues: *GroupEpicIssueManager*

notes: *GroupEpicNoteManager*

resourcelabelevents: *GroupEpicResourceLabelEventManager*

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupEpicAwardEmojiManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *NoUpdateMixin*[*GroupEpicAwardEmoji*]

Object Creation

Required attributes for object create:

- name

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupEpicDiscussionNoteManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetMixin*[*GroupEpicDiscussionNote*], *CreateMixin*[*GroupEpicDiscussionNote*], *UpdateMixin*[*GroupEpicDiscussionNote*], *DeleteMixin*[*GroupEpicDiscussionNote*]

Object Creation

Required attributes for object create:

- body

Optional attributes for object create:

- created_at

Object update

Required attributes for object update:

- body

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *SaveMixin*, *RESTObject*

Parameters

- **manager** (GroupEpicIssueManager)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

manager: *GroupEpicIssueManager*

save(***kwargs: Any*) → None

Save the changes made to the object to the server.

The object is updated to match what the server returns.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Return type
None

class gitlab.v4.objects.**GroupEpicIssueManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*GroupEpicIssue*], *CreateMixin*[*GroupEpicIssue*], *UpdateMixin*[*GroupEpicIssue*], *DeleteMixin*[*GroupEpicIssue*]

Object Creation

Required attributes for object create:

- issue_id

Object update

Optional attributes for object update:

- move_before_id
- move_after_id

Parameters

- **gl** (*Gitlab*)

- **parent** (*RESTObject | None*)

create(*data: dict[str, Any] | None = None*, ***kwargs: Any*) → *GroupEpicIssue*

Create a new object.

Parameters

- **data** (*dict[str, Any] | None*) – Parameters to send to the server to create the resource
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

Returns

A new instance of the manage object class build with
the data sent by the server

Return type
GroupEpicIssue

class gitlab.v4.objects.**GroupEpicManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*GroupEpic*]

Object listing filters

- author_id
- labels
- order_by
- sort
- search

Object Creation

Required attributes for object create:

- title

Optional attributes for object create:

- labels
- description
- start_date
- end_date

Object update

Optional attributes for object update:

- title
- labels
- description
- start_date

- end_date

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

awardemojis: *GroupEpicNoteAwardEmojiManager*

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupEpicNoteAwardEmojiManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *NoUpdateMixin*[*GroupEpicNoteAwardEmoji*]

Object Creation

Required attributes for object create:

- name

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**GroupEpicNoteManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*GroupEpicNote*]

Object Creation

Required attributes for object create:

- body

Optional attributes for object create:

- `created_at`

Object update

Required attributes for object update:

- `body`

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupEpicResourceLabelEventManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*GroupEpicResourceLabelEvent*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *DownloadMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupExportManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*GroupExport*], *CreateMixin*[*GroupExport*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

test(*trigger: str*) → None

Test a Group Hook

Parameters
trigger (*str*) – Type of trigger event to test

Raises
GitlabHookTestError – If the hook test attempt failed

Return type
None

class gitlab.v4.objects.**GroupHookManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*GroupHook*]

Object Creation

Required attributes for object create:

- `url`

Optional attributes for object create:

- `push_events`
- `issues_events`
- `confidential_issues_events`
- `merge_requests_events`
- `tag_push_events`
- `note_events`
- `confidential_note_events`
- `job_events`
- `pipeline_events`
- `wiki_page_events`
- `deployment_events`
- `releases_events`
- `subgroup_events`

- enable_ssl_verification
- token

Object update

Required attributes for object update:

- url

Optional attributes for object update:

- push_events
- issues_events
- confidential_issues_events
- merge_requests_events
- tag_push_events
- note_events
- confidential_note_events
- job_events
- pipeline_events
- wiki_page_events
- deployment_events
- releases_events
- subgroup_events
- enable_ssl_verification
- token

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupImportManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*GroupImport*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupInvitationManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: InvitationMixin[*GroupInvitation*]

Object listing filters

- query

Object Creation

Required attributes for object create:

- access_level

Optional attributes for object create:

- expires_at
- invite_source
- tasks_to_be_done
- tasks_project_id

Mutually exclusive attributes for object create:

- email
- user_id

Object update

Optional attributes for object update:

- access_level
- expires_at

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupIssueManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*GroupIssue*]

Object listing filters

- state
- labels
- milestone
- order_by
- sort
- iids
- author_id
- iteration_id
- assignee_id
- my_reaction_emoji
- search
- created_after
- created_before
- updated_after
- updated_before

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RefreshMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupIssuesStatisticsManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*GroupIssuesStatistics*]

Object listing filters

- iids

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupIterationManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*GroupIteration*]

Object listing filters

- include_ancestors
- include_descendants
- in
- search
- state
- updated_after
- updated_before

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)

- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

delete(***kwargs: Any*) → None

Delete the LDAP group link from the server.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises
- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the server cannot perform the request

Return type
None

class gitlab.v4.objects.**GroupLDAPGroupLinkManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*GroupLDAPGroupLink*], *CreateMixin*[*GroupLDAPGroupLink*], *DeleteMixin*[*GroupLDAPGroupLink*]

Object Creation

Required attributes for object create:

- `provider`
- `group_access`

Mutually exclusive attributes for object create:

- `cn`
- `filter`

Object update

Parameters
- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SubscribableMixin*, *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters
- **manager** (GroupLabelManager)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

manager: *GroupLabelManager*

save(***kwargs: Any*) → None

Saves the changes made to the object to the server.

The object is updated to match what the server returns.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises
- **GitlabAuthenticationError** – If authentication is not correct.
- **GitlabUpdateError** – If the server cannot perform the request.

Return type
None

class gitlab.v4.objects.**GroupLabelManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *RetrieveMixin*[*GroupLabel*], *CreateMixin*[*GroupLabel*], *UpdateMixin*[*GroupLabel*], *DeleteMixin*[*GroupLabel*]

Object Creation

Required attributes for object create:
- `name`
- `color`

Optional attributes for object create:
- `description`
- `priority`

Object update

Required attributes for object update:
- `name`

Optional attributes for object update:
- `new_name`
- `color`
- `description`
- `priority`

Parameters
- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

update(*name: str | None*, *new_data: dict[str,* Any*] | None* = *None*, ***kwargs:* Any) → dict[str, *Any*]

Update a Label on the server.

Parameters
- **name** (*str | None*) – The name of the label
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)
- **new_data** (*dict[str, Any] | None*)
- ****kwargs**

Return type
dict[str, Any]

class gitlab.v4.objects.**GroupManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CRUDMixin*[*Group*]

Object listing filters

- skip_groups
- all_available
- search
- order_by
- sort
- statistics
- owned
- with_custom_attributes
- min_access_level
- top_level_only

Object Creation

Required attributes for object create:

- name
- path

Optional attributes for object create:

- description
- membership_lock
- visibility
- share_with_group_lock
- require_two_factor_authentication
- two_factor_grace_period
- project_creation_level
- auto_devops_enabled
- subgroup_creation_level
- emails_disabled
- avatar
- mentions_disabled
- lfs_enabled
- request_access_enabled
- parent_id
- default_branch_protection
- shared_runners_minutes_limit
- extra_shared_runners_minutes_limit

Object update

Optional attributes for object update:

- name
- path
- description
- membership_lock
- share_with_group_lock
- visibility
- require_two_factor_authentication
- two_factor_grace_period
- project_creation_level
- auto_devops_enabled
- subgroup_creation_level
- emails_disabled
- avatar
- mentions_disabled
- lfs_enabled
- request_access_enabled
- default_branch_protection
- file_template_project_id
- shared_runners_minutes_limit
- extra_shared_runners_minutes_limit
- prevent_forking_outside_group
- shared_runners_setting

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

import_group(*file: BinaryIO*, *path: str*, *name: str*, *parent_id: int | str | None = None*, ***kwargs:* Any) → dict[str, *Any*] | Response

Import a group from an archive file.

Parameters

- **file** (*BinaryIO*) – Data or file object containing the group
- **path** (*str*) – The path for the new group to be imported.
- **name** (*str*) – The name for the new group.
- **parent_id** (*int | str | None*) – ID of a parent group that the group will be imported into.

- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
A representation of the import status.

Return type
dict[str, Any] | *Response*

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupMemberAllManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*GroupMemberAll*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**GroupMemberManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*GroupMember*]

Object Creation

Required attributes for object create:

- `access_level`

Optional attributes for object create:

- `expires_at`

- tasks_to_be_done

Mutually exclusive attributes for object create:

- username
- user_id

Object update

Required attributes for object update:

- access_level

Optional attributes for object update:

- expires_at

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupMemberRoleManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*GroupMemberRole*], *CreateMixin*[*GroupMemberRole*], *DeleteMixin*[*GroupMemberRole*]

Object Creation

Required attributes for object create:

- name
- base_access_level

Optional attributes for object create:

- description
- admin_cicd_variables
- admin_compliance_framework
- admin_group_member
- admin_group_member
- admin_merge_request
- admin_push_rules
- admin_terraform_state

- admin_vulnerability
- admin_web_hook
- archive_project
- manage_deploy_tokens
- manage_group_access_tokens
- manage_merge_request_settings
- manage_project_access_tokens
- manage_security_policy_link
- read_code
- read_runners
- read_dependency
- read_vulnerability
- remove_group
- remove_project

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupMergeRequestManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*GroupMergeRequest*]

Object listing filters

- state
- order_by
- sort
- milestone
- view
- labels
- created_after
- created_before

- updated_after
- updated_before
- scope
- author_id
- assignee_id
- approver_ids
- approved_by_ids
- my_reaction_emoji
- source_branch
- target_branch
- search
- wip

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

issues(***kwargs: Any*) → *RESTObjectList*[*GroupIssue*]

List issues related to this milestone.

Parameters

- **get_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[GroupIssue]

merge_requests(***kwargs: Any*) → *RESTObjectList*[*GroupMergeRequest*]

List the merge requests related to this milestone.

Parameters

- **get_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[GroupMergeRequest]

class gitlab.v4.objects.**GroupMilestoneManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*GroupMilestone*]

Object listing filters

- iids
- state
- search

Object Creation

Required attributes for object create:

- title

Optional attributes for object create:

- description
- due_date
- start_date

Object update

Optional attributes for object update:

- title
- description
- due_date
- start_date
- state_event

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *NotificationSettings*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupNotificationSettingsManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *NotificationSettingsManager*

Object Creation

Object update

Optional attributes for object update:

- level
- notification_email
- new_note
- new_issue
- reopen_issue
- close_issue
- reassign_issue
- new_merge_request
- reopen_merge_request
- close_merge_request
- reassign_merge_request
- merge_merge_request

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)

- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupPackageManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*GroupPackage*]

Object listing filters

- exclude_subgroups
- order_by
- sort
- package_type
- package_name
- package_version
- include_versionless
- status

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupProjectManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*GroupProject*]

Object listing filters

- archived
- visibility
- order_by
- sort
- search
- simple
- owned

- starred
- with_custom_attributes
- include_subgroups
- with_issues_enabled
- with_merge_requests_enabled
- with_shared
- min_access_level
- with_security_reports

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupPushRulesManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*GroupPushRules*], *CreateMixin*[*GroupPushRules*], *UpdateMixin*[*GroupPushRules*], *DeleteMixin*[*GroupPushRules*]

Object Creation

Optional attributes for object create:

- deny_delete_tag
- member_check
- prevent_secrets
- commit_message_regex
- commit_message_negative_regex
- branch_name_regex
- author_email_regex
- file_name_regex
- max_file_size
- commit_committer_check
- reject_unsigned_commits

Object update

Optional attributes for object update:

- deny_delete_tag
- member_check
- prevent_secrets
- commit_message_regex
- commit_message_negative_regex
- branch_name_regex
- author_email_regex
- file_name_regex
- max_file_size
- commit_committer_check
- reject_unsigned_commits

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**GroupRegistryRepositoryManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectRegistryRepository*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupRunnerManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*GroupRunner*]

Object listing filters

- scope
- tag_list

Object Creation

Required attributes for object create:

- `runner_id`

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupSAMLGroupLinkManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *NoUpdateMixin*[*GroupSAMLGroupLink*]

Object Creation

Required attributes for object create:

- `saml_group_name`
- `access_level`

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

access_tokens: *GroupServiceAccountAccessTokenManager*

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

Bases: *ObjectDeleteMixin*, `ObjectRotateMixin`, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupServiceAccountAccessTokenManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CreateMixin*[*GroupServiceAccountAccessToken*], *DeleteMixin*[*GroupServiceAccountAccessToken*], *ListMixin*[*GroupServiceAccountAccessToken*], RotateMixin[*GroupServiceAccountAccessToken*]

Object listing filters

- created_after
- created_before
- expires_after
- expires_before
- last_used_after
- last_used_before
- revoked
- search
- sort
- state

Object Creation

Required attributes for object create:

- name
- scopes

Optional attributes for object create:

- description
- expires_at

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**GroupServiceAccountManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CreateMixin*[*GroupServiceAccount*], *DeleteMixin*[*GroupServiceAccount*], *ListMixin*[*GroupServiceAccount*], *UpdateMixin*[*GroupServiceAccount*]

Object listing filters

- order_by
- sort

Object Creation

Optional attributes for object create:

- name
- username
- email

Object update

Optional attributes for object update:

- name
- username
- email

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupSubgroupManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: SubgroupBaseManager[*GroupSubgroup*]

Object listing filters

- skip_groups
- all_available
- search
- order_by
- sort
- statistics
- owned
- with_custom_attributes
- min_access_level

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupVariableManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*GroupVariable*]

Object Creation

Required attributes for object create:

- key
- value

Optional attributes for object create:

- description
- environment_scope
- masked
- masked_and_hidden
- protected
- raw
- variable_type

Object update

Required attributes for object update:

- key

Optional attributes for object update:

- description
- environment_scope
- masked
- protected
- raw
- value
- variable_type

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, UploadMixin, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**GroupWikiManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*GroupWiki*]

Object listing filters

- with_content

Object Creation

Required attributes for object create:

- title
- content

Optional attributes for object create:

- format

Object update

Optional attributes for object update:

- title
- content
- format

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**HookManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *NoUpdateMixin*[*Hook*]

Object Creation

Required attributes for object create:

- url

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str,* Any])
- **created_from_list** (*bool*)
- **lazy** (*bool*)

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

Bases: *RetrieveMixin*[*Issue*]

Object listing filters

- state
- labels
- milestone
- scope
- author_id
- iteration_id
- assignee_id
- my_reaction_emoji
- iids
- order_by
- sort
- search
- created_after
- created_before
- updated_after
- updated_before

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RefreshMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**IssuesStatisticsManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*IssuesStatistics*]

Object listing filters

- iids

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

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

Bases: *GetMixin*[*Key*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

get(*id: int | str | None = None*, *lazy: bool = False*, ***kwargs: Any*) → *Key*

Retrieve a single object.

Parameters

- **id** (*int | str | None*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

Returns

The generated RESTObject.

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

Return type

Key

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

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

Bases: *RESTManager*[*LDAPGroup*]

Object listing filters

- search
- provider

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

list(***, *iterator: Literal[False] = False*, ***kwargs: Any*) → *list*[*LDAPGroup*]

list(***, *iterator: Literal[True] = True*, ***kwargs: Any*) → RESTObjectList[*LDAPGroup*]

list(***, *iterator: bool = False*, ***kwargs: Any*) → *list*[*LDAPGroup*] | RESTObjectList[*LDAPGroup*]

Retrieve a list of objects.

Parameters

- **get_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** (*bool*) – If set to True and no pagination option is defined, return a generator instead of a list
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns
The list of objects, or a generator if *iterator* is True

Raises
- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the server cannot perform the request

Return type
list[LDAPGroup] | RESTObjectList[LDAPGroup]

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

Bases: *RESTObject*

Parameters
- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**LicenseManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*License*]

Object listing filters
- popular

Object Creation

Object update

Parameters
- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters
- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**MemberRoleManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *ListMixin*[*MemberRole*], *CreateMixin*[*MemberRole*], *DeleteMixin*[*MemberRole*]

Object Creation

Required attributes for object create:

- `name`
- `base_access_level`

Optional attributes for object create:

- `description`
- `admin_cicd_variables`
- `admin_compliance_framework`
- `admin_group_member`
- `admin_group_member`
- `admin_merge_request`
- `admin_push_rules`
- `admin_terraform_state`
- `admin_vulnerability`
- `admin_web_hook`
- `archive_project`
- `manage_deploy_tokens`
- `manage_group_access_tokens`
- `manage_merge_request_settings`
- `manage_project_access_tokens`
- `manage_security_policy_link`
- `read_code`
- `read_runners`
- `read_dependency`
- `read_vulnerability`
- `remove_group`
- `remove_project`

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (dict[str, Any])
- **created_from_list** (bool)
- **lazy** (bool)

class gitlab.v4.objects.**MergeRequestManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: ListMixin[MergeRequest]

Object listing filters

- state
- order_by
- sort
- milestone
- view
- labels
- with_labels_details
- with_merge_status_recheck
- created_after
- created_before
- updated_after
- updated_before
- scope
- author_id
- author_username
- assignee_id
- approver_ids
- approved_by_ids
- reviewer_id
- reviewer_username
- my_reaction_emoji
- source_branch
- target_branch
- search
- in
- wip
- not
- environment
- deployed_before

- deployed_after

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**NamespaceManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*Namespace*]

Object listing filters

- search

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

exists(*namespace: str*, ***kwargs: Any*) → *Namespace*

Get existence of a namespace by path.

Parameters

- **namespace** (*str*) – The path to the namespace.
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns

Data on namespace existence returned from the server.

Return type

Namespace

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

Bases: *SaveMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**NotificationSettingsManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*NotificationSettings*], *UpdateMixin*[*NotificationSettings*]

Object Creation

Object update

Optional attributes for object update:

- level
- notification_email
- new_note
- new_issue
- reopen_issue
- close_issue
- reassign_issue
- new_merge_request
- reopen_merge_request
- close_merge_request
- reassign_merge_request
- merge_merge_request

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**PagesDomainManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*PagesDomain*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, ObjectRotateMixin, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**PersonalAccessTokenManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *DeleteMixin*[*PersonalAccessToken*], *RetrieveMixin*[*PersonalAccessToken*], RotateMixin[*PersonalAccessToken*]

Object listing filters

- user_id

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RefreshMixin*, *SaveMixin*, *ObjectDeleteMixin*, *RepositoryMixin*, UploadMixin, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

access_tokens: *ProjectAccessTokenManager*

accessrequests: *ProjectAccessRequestManager*

additionalstatistics: *ProjectAdditionalStatisticsManager*

approvalrules: *ProjectApprovalRuleManager*

approvals: *ProjectApprovalManager*

archive(***kwargs: Any*) → None

Archive a project.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

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

Return type
None

artifacts: *ProjectArtifactManager*

audit_events: *ProjectAuditEventManager*

badges: *ProjectBadgeManager*

boards: *ProjectBoardManager*

branches: *ProjectBranchManager*

ci_lint: *ProjectCiLintManager*

cluster_agents: *ProjectClusterAgentManager*

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*

dockerfile_templates: *ProjectDockerfileTemplateManager*

environments: *ProjectEnvironmentManager*

events: *ProjectEventManager*

exports: *ProjectExportManager*

external_status_checks: *ProjectExternalStatusCheckManager*

feature_flags: *ProjectFeatureFlagManager*

feature_flags_user_lists: *ProjectFeatureFlagUserListManager*

files: *ProjectFileManager*

forks: *ProjectForkManager*

generic_packages: *GenericPackageManager*

gitignore_templates: *ProjectGitignoreTemplateManager*

gitlabciyml_templates: *ProjectGitlabciymlTemplateManager*

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*

issue_templates: *ProjectIssueTemplateManager*

issues: *ProjectIssueManager*

issues_statistics: *ProjectIssuesStatisticsManager*

iterations: *ProjectIterationManager*

job_token_scope: *ProjectJobTokenScopeManager*

jobs: *ProjectJobManager*

keys: *ProjectKeyManager*

labels: *ProjectLabelManager*

languages(***kwargs:* Any) → dict[str, *Any*] | Response

Get languages used in the project with percentage value.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

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

Return type
dict[str, Any] | *Response*

license_templates: *ProjectLicenseTemplateManager*

members: *ProjectMemberManager*

members_all: *ProjectMemberAllManager*

merge_request_templates: *ProjectMergeRequestTemplateManager*

merge_trains: *ProjectMergeTrainManager*

mergerequests: *ProjectMergeRequestManager*

milestones: *ProjectMilestoneManager*

mirror_pull(***kwargs:* Any) → None

Start the pull mirroring process for the project.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

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

Return type
None

mirror_pull_details(***kwargs:* Any) → dict[str, *Any*]

Get a project's pull mirror details.

Introduced in GitLab 15.5.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
dict of the parsed json returned by the server

Return type
dict[str, Any]

notes: *ProjectNoteManager*

notificationsettings: *ProjectNotificationSettingsManager*

package_protection_rules: *ProjectPackageProtectionRuleManager*

packages: *ProjectPackageManager*

pages: *ProjectPagesManager*

pagesdomains: *ProjectPagesDomainManager*

path_with_namespace: **str**

pipelines: *ProjectPipelineManager*

pipelineschedules: *ProjectPipelineScheduleManager*

protected_environments: *ProjectProtectedEnvironmentManager*

protectedbranches: *ProjectProtectedBranchManager*

protectedtags: *ProjectProtectedTagManager*

pull_mirror: *ProjectPullMirrorManager*

pushrules: *ProjectPushRulesManager*

registry_protection_repository_rules:
ProjectRegistryRepositoryProtectionRuleManager

registry_protection_rules: *ProjectRegistryProtectionRuleManager*

releases: *ProjectReleaseManager*

remote_mirrors: *ProjectRemoteMirrorManager*

repositories: *ProjectRegistryRepositoryManager*

resource_groups: *ProjectResourceGroupManager*

restore(***kwargs: Any*) → None

Restore a project marked for deletion.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct

- **GitlabRestoreError** – If the server failed to perform the request

Return type
None

runners: *ProjectRunnerManager*

search(*scope: str*, *search: str*, ***kwargs: Any*) → *GitlabList* | list[dict[str, *Any*]]

Search the project resources matching the provided string.'

Parameters

- **scope** (*str*) – Scope of the search
- **search** (*str*) – Search string
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
A list of dicts describing the resources found.

Return type
GitlabList | list[dict[str, Any]]

secure_files: *ProjectSecureFileManager*

service_accounts: *ProjectServiceAccountManager*

services: *ProjectServiceManager*

share(*group_id: int*, *group_access: int*, *expires_at: str | None = None*, ***kwargs: Any*) → None

Share the project with a group.

Parameters

- **group_id** (*int*) – ID of the group.
- **group_access** (*int*) – Access level for the group.
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)
- **expires_at** (*str | None*)
- ****kwargs**

Raises

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

Return type
None

snapshot(*wiki: bool = False*, *streamed: Literal[False] = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → bytes

snapshot(*wiki: bool = False*, *streamed: bool = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[True] = True*, ***kwargs: Any*) → Iterator[*Any*]

snapshot(*wiki: bool = False*, *streamed: Literal[True] = True*, *action: Callable[[bytes], Any] | None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → None

Return a snapshot of the repository.

Parameters

- **wiki** (*bool*) – If True return the wiki repository
- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk_size* and each chunk is passed to *action* for treatment.
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable[[bytes], Any] | None*) – Callable responsible of dealing with chunk of data
- **chunk_size** (*int*) – Size of each chunk
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns

The uncompressed tar archive of the repository

Return type

bytes | *Iterator*[Any] | None

snippets: *ProjectSnippetManager*

star(***kwargs: Any*) → None

Star a project.

Parameters

****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Return type

None

storage: *ProjectStorageManager*

tags: *ProjectTagManager*

transfer(*to_namespace: int | str*, ***kwargs: Any*) → None

Transfer a project to the given namespace ID

Parameters

- **to_namespace** (*int | str*) – ID or path of the namespace to transfer the
- **to** (*project*)
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabTransferProjectError** – If the project could not be transferred

Return type
None

trigger_pipeline(*ref: str*, *token: str*, *variables: dict[str, Any] | None = None*, *inputs: dict[str, Any] | None = None*, ***kwargs: Any*) → *ProjectPipeline*

Trigger a CI build.

See https://gitlab.com/help/ci/triggers/README.md#trigger-a-build

Parameters

- **ref** (*str*) – Commit to build; can be a branch name or a tag
- **token** (*str*) – The trigger token
- **variables** (*dict[str, Any] | None*) – Variables passed to the build script
- **inputs** (*dict[str, Any] | None*) – Inputs passed to the build script
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Return type
ProjectPipeline

triggers: *ProjectTriggerManager*

unarchive(***kwargs: Any*) → None

Unarchive a project.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Return type
None

unshare(*group_id: int*, ***kwargs: Any*) → None

Delete a shared project link within a group.

Parameters

- **group_id** (*int*) – ID of the group.
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Return type
None

unstar(***kwargs: Any*) → None

Unstar a project.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

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

Return type
None

users: *ProjectUserManager*

variables: *ProjectVariableManager*

wikis: *ProjectWikiManager*

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

Bases: *AccessRequestMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters
- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectAccessRequestManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectAccessRequest*], *CreateMixin*[*ProjectAccessRequest*], *DeleteMixin*[*ProjectAccessRequest*]

Object Creation

Object update

Parameters
- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, ObjectRotateMixin, *RESTObject*

Parameters
- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectAccessTokenManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CreateMixin*[*ProjectAccessToken*], *DeleteMixin*[*ProjectAccessToken*], *RetrieveMixin*[*ProjectAccessToken*], RotateMixin[*ProjectAccessToken*]

Object Creation

Required attributes for object create:

- name
- scopes

Optional attributes for object create:

- access_level
- expires_at

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RefreshMixin*, *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str,* Any*]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectAdditionalStatisticsManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *GetWithoutIdMixin*[*ProjectAdditionalStatistics*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str,* Any*]*)
- **created_from_list** (*bool*)

- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectApprovalManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *GetWithoutIdMixin*[*ProjectApproval*], *UpdateMixin*[*ProjectApproval*]

Object Creation

Object update

Optional attributes for object update:

- approvals_before_merge
- reset_approvals_on_push
- disable_overriding_approvers_per_merge_request
- merge_requests_author_approval
- merge_requests_disable_committers_approval

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str,* Any]*)*
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectApprovalRuleManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *RetrieveMixin*[*ProjectApprovalRule*], *CreateMixin*[*ProjectApprovalRule*], *UpdateMixin*[*ProjectApprovalRule*], *DeleteMixin*[*ProjectApprovalRule*]

Object Creation

Required attributes for object create:

- name
- approvals_required

Optional attributes for object create:

- user_ids
- group_ids
- protected_branch_ids
- usernames
- applies_to_all_protected_branches

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**ProjectArtifact**(*manager: RESTManager[Any]*, *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[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectArtifactManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RESTManager*[*ProjectArtifact*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

delete(***kwargs: Any*) → None

Delete the project's artifacts on the server.

Parameters

****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the server cannot perform the request

Return type

None

download(*ref_name: str*, *job: str*, *streamed: Literal[False] = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → bytes

download(*ref_name: str*, *job: str*, *streamed: bool = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[True] = True*, ***kwargs: Any*) → Iterator[*Any*]

download(*ref_name: str*, *job: str*, *streamed: Literal[True] = True*, *action: Callable[[bytes], Any] | None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → None

Get the job artifacts archive from a specific tag or branch.

Parameters

- **ref_name** (*str*) – Branch or tag name in repository. HEAD or SHA references are not supported.
- **job** (*str*) – The name of the job.
- **job_token** – Job token for multi-project pipeline triggers.

- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable[[bytes], Any] | None*) – Callable responsible of dealing with chunk of data
- **chunk_size** (*int*) – Size of each chunk
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
The artifacts if *streamed* is False, None otherwise.

Return type
bytes | *Iterator*[Any] | None

raw(*ref_name: str*, *artifact_path: str*, *job: str*, *streamed: Literal[False] = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs:* Any) → bytes

raw(*ref_name: str*, *artifact_path: str*, *job: str*, *streamed: bool = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[True] = True*, ***kwargs:* Any) → Iterator[*Any*]

raw(*ref_name: str*, *artifact_path: str*, *job: str*, *streamed: Literal[True] = True*, *action: Callable[[bytes], Any] | None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs:* Any) → None

Download a single artifact file from a specific tag or branch from within the job's artifacts archive.

Parameters

- **ref_name** (*str*) – Branch or tag name in repository. HEAD or SHA references are not supported.
- **artifact_path** (*str*) – Path to a file inside the artifacts archive.
- **job** (*str*) – The name of the job.
- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable[[bytes], Any] | None*) – Callable responsible of dealing with chunk of data
- **chunk_size** (*int*) – Size of each chunk
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
The artifact if *streamed* is False, None otherwise.

Return type
bytes | *Iterator*[Any] | None

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

Bases: *ProjectAuditEvent*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

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

Bases: *RetrieveMixin*[*ProjectAuditEvent*]

Object listing filters

- created_after
- created_before

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ProjectAuditEventManager*

Object listing filters

- created_after
- created_before

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectBadgeManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *BadgeRenderMixin*[*ProjectBadge*], *CRUDMixin*[*ProjectBadge*]

Object Creation

Required attributes for object create:

- link_url
- image_url

Object update

Optional attributes for object update:

- link_url
- image_url

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

lists: *ProjectBoardListManager*

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)

- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectBoardListManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*ProjectBoardList*]

Object Creation

Mutually exclusive attributes for object create:

- `label_id`
- `assignee_id`
- `milestone_id`
- `iteration_id`

Object update

Required attributes for object update:

- `position`

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *CRUDMixin*[*ProjectBoard*]

Object Creation

Required attributes for object create:

- `name`

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectBranchManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *NoUpdateMixin*[*ProjectBranch*]

Object Creation

Required attributes for object create:

- branch
- ref

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectCiLintManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*ProjectCiLint*], *CreateMixin*[*ProjectCiLint*]

Object Creation

Required attributes for object create:

- content

Optional attributes for object create:

- dry_run
- include_jobs
- ref

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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[Any]*, *attrs: dict[str, Any]*, ***, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectClusterAgentManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *NoUpdateMixin*[*ProjectClusterAgent*]

Object Creation

Required attributes for object create:

- `name`

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**ProjectClusterManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*ProjectCluster*]

Object Creation

Required attributes for object create:

- `name`
- `platform_kubernetes_attributes`

Optional attributes for object create:

- `domain`
- `enabled`
- `managed`
- `environment_scope`

Object update

Optional attributes for object update:

- `name`
- `domain`
- `management_project_id`
- `platform_kubernetes_attributes`
- `environment_scope`

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

create(*data: dict[str, Any] | None = None*, ***kwargs: Any*) → *ProjectCluster*

Create a new object.

Parameters

- **data** (*dict[str, Any] | None*) – Parameters to send to the server to create the resource
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo or 'ref_name', 'stage', 'name', 'all')

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

Returns

A new instance of the manage object class build with
the data sent by the server

Return type
ProjectCluster

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

cherry_pick(*branch: str*, ***kwargs: Any*) → dict[str, *Any*] | Response

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

Returns
The new commit data (*not* a RESTObject)

Return type
dict[str, Any] | *Response*

comments: *ProjectCommitCommentManager*

diff(***kwargs: Any*) → *GitlabList* | list[dict[str, *Any*]]

Generate the commit diff.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
The changes done in this commit

Return type
GitlabList | list[dict[str, Any]]

discussions: *ProjectCommitDiscussionManager*

merge_requests(***kwargs: Any*) → *GitlabList* | list[dict[str, *Any*]]

List the merge requests related to the commit.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
The merge requests related to the commit.

Return type
GitlabList | list[dict[str, Any]]

refs(*type: str = 'all'*, ***kwargs: Any*) → *GitlabList* | list[dict[str, *Any*]]

List the references the commit is pushed to.

Parameters

- **type** (*str*) – The scope of references ('branch', 'tag' or 'all')
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
The references the commit is pushed to.

Return type
GitlabList | list[dict[str, Any]]

revert(*branch: str*, ***kwargs: Any*) → dict[str, *Any*] | Response

Revert a commit on a given branch.

Parameters

- **branch** (*str*) – Name of target branch
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabRevertError** – If the revert could not be performed

Returns
The new commit data (*not* a RESTObject)

Return type
dict[str, Any] | *Response*

sequence(***kwargs: Any*) → dict[str, *Any*] | Response

Get the sequence number 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 sequence number could not be retrieved

Returns
The commit's sequence number

Return type
dict[str, Any] | *Response*

signature(***kwargs: Any*) → dict[str, *Any*] | Response

Get the signature of the commit.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
The commit's signature data

Return type
dict[str, Any] | *Response*

statuses: *ProjectCommitStatusManager*

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectCommitCommentManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectCommitComment*], *CreateMixin*[*ProjectCommitComment*]

Object Creation

Required attributes for object create:

- `note`

Optional attributes for object create:

- `path`
- `line`
- `line_type`

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

notes: *ProjectCommitDiscussionNoteManager*

class gitlab.v4.objects.**ProjectCommitDiscussionManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectCommitDiscussion*], *CreateMixin*[*ProjectCommitDiscussion*]

Object Creation

Required attributes for object create:

- `body`

Optional attributes for object create:

- created_at

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectCommitDiscussionNoteManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *GetMixin*[*ProjectCommitDiscussionNote*], *CreateMixin*[*ProjectCommitDiscussionNote*], *UpdateMixin*[*ProjectCommitDiscussionNote*], *DeleteMixin*[*ProjectCommitDiscussionNote*]

Object Creation

Required attributes for object create:

- body

Optional attributes for object create:

- created_at
- position

Object update

Required attributes for object update:

- body

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RetrieveMixin*[*ProjectCommit*], *CreateMixin*[*ProjectCommit*]

Object listing filters

- all
- ref_name
- since
- until

- path
- with_stats
- first_parent
- order
- trailers

Object Creation

Required attributes for object create:

- branch
- commit_message
- actions

Optional attributes for object create:

- author_email
- author_name

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RefreshMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectCommitStatusManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectCommitStatus*], *CreateMixin*[*ProjectCommitStatus*]

Object Creation

Required attributes for object create:

- state

Optional attributes for object create:

- description
- name
- context
- ref
- target_url
- coverage

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

create(*data: dict[str, Any] | None = None*, ***kwargs: Any*) → *ProjectCommitStatus*

Create a new object.

Parameters

- **data** (*dict[str, Any] | None*) – Parameters to send to the server to create the resource
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo or 'ref_name', 'stage', 'name', 'all')

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

Returns

A new instance of the manage object class build with
the data sent by the server

Return type
ProjectCommitStatus

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectCustomAttributeManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectCustomAttribute*], *SetMixin*[*ProjectCustomAttribute*], *DeleteMixin*[*ProjectCustomAttribute*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (dict[str, Any])
- **created_from_list** (bool)
- **lazy** (bool)

class gitlab.v4.objects.**ProjectDeployTokenManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *RetrieveMixin*[*ProjectDeployToken*], *CreateMixin*[*ProjectDeployToken*], *DeleteMixin*[*ProjectDeployToken*]

Object listing filters

- scopes

Object Creation

Required attributes for object create:

- name
- scopes

Optional attributes for object create:

- expires_at
- username

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (dict[str, Any])
- **created_from_list** (bool)
- **lazy** (bool)

approval(*status: str*, *comment: str | None = None*, *represented_as: str | None = None*, ***kwargs:* Any) → dict[str, *Any*]

Approve or reject a blocked deployment.

Parameters

- **status** (*str*) – Either “approved” or “rejected”
- **comment** (*str | None*) – A comment to go with the approval
- **represented_as** (*str | None*) – The name of the User/Group/Role to use for the approval, when the user belongs to multiple approval rules.
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabMRApprovalError** – If the approval failed

Returns
A dict containing the result.

Return type
dict[str, Any]

https://docs.gitlab.com/ee/api/deployments.html#approve-or-reject-a-blocked-deployment

mergerequests: *ProjectDeploymentMergeRequestManager*

class gitlab.v4.objects.**ProjectDeploymentManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *RetrieveMixin*[*ProjectDeployment*], *CreateMixin*[*ProjectDeployment*], *UpdateMixin*[*ProjectDeployment*]

Object listing filters

- order_by
- sort
- updated_after
- updated_before
- environment
- status

Object Creation

Required attributes for object create:

- sha
- ref
- tag
- status
- environment

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *MergeRequest*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)

- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectDeploymentMergeRequestManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *MergeRequestManager*

Object listing filters

- state
- order_by
- sort
- milestone
- view
- labels
- with_labels_details
- with_merge_status_recheck
- created_after
- created_before
- updated_after
- updated_before
- scope
- author_id
- author_username
- assignee_id
- approver_ids
- approved_by_ids
- reviewer_id
- reviewer_username
- my_reaction_emoji
- source_branch
- target_branch
- search
- in
- wip
- not
- environment
- deployed_before
- deployed_after

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectDockerfileTemplateManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectDockerfileTemplate*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

stop(***kwargs: Any*) → dict[str, *Any*] | Response

Stop the environment.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabStopError** – If the operation failed

Returns
A dict of the result.

Return type
dict[str, Any] | *Response*

class gitlab.v4.objects.**ProjectEnvironmentManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *RetrieveMixin*[*ProjectEnvironment*], *CreateMixin*[*ProjectEnvironment*], *UpdateMixin*[*ProjectEnvironment*], *DeleteMixin*[*ProjectEnvironment*]

Object listing filters

- name
- search
- states

Object Creation

Required attributes for object create:

- name

Optional attributes for object create:

- external_url

Object update

Optional attributes for object update:

- name
- external_url

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *Event*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str,* Any*]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectEventManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *EventManager*

Object listing filters

- action
- target_type
- before
- after
- sort
- scope

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *DownloadMixin*, *RefreshMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectExportManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*ProjectExport*], *CreateMixin*[*ProjectExport*]

Object Creation

Optional attributes for object create:

- description

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectExternalStatusCheckManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectExternalStatusCheck*], *CreateMixin*[*ProjectExternalStatusCheck*], *UpdateMixin*[*ProjectExternalStatusCheck*], *DeleteMixin*[*ProjectExternalStatusCheck*]

Object Creation

Required attributes for object create:

- name

- external_url

Optional attributes for object create:

- shared_secret
- protected_branch_ids

Object update

Optional attributes for object update:

- name
- external_url
- shared_secret
- protected_branch_ids

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (ProjectFeatureFlagManager)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

manager: *ProjectFeatureFlagManager*

save(***kwargs: Any*) → dict[str, *Any*] | None

Save the changes made to the object to the server.

This is the standard method to use when updating a feature flag object that you have already retrieved.

It is overridden here to correctly handle renaming. When *name* is changed, the API requires the *original* name in the URL, and this method provides it.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns
The new object data (*not* a RESTObject)

Raises

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

Return type
dict[str, Any] | None

class gitlab.v4.objects.**ProjectFeatureFlagManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CRUDMixin*[*ProjectFeatureFlag*]

Object listing filters

- `scope`

Object Creation

Required attributes for object create:

- `name`

Optional attributes for object create:

- `version`
- `description`
- `active`
- `strategies`

Object update

Optional attributes for object update:

- `name`
- `new_name`
- `description`
- `active`
- `strategies`

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

update(*id: str | int | None = None*, *new_data: dict[str,* Any*] | None = None*, ***kwargs:* Any) → dict[str, *Any*]

Update a Project Feature Flag.

This is a lower-level method called by *ProjectFeatureFlag.save()* and is also used directly by the CLI.

The *new_name* parameter is a special case to support renaming via the CLI (*–new-name*). It is converted to the *name* parameter that the GitLab API expects in the request body.

Parameters

- **id** (*str | int | None*) – The current name of the feature flag.
- **new_data** (*dict[str,* Any*] | None*) – The dictionary of attributes to update.
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Return type

dict[str, Any]

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectFeatureFlagUserListManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*ProjectFeatureFlagUserList*]

Object listing filters

- search

Object Creation

Required attributes for object create:

- name
- user_xids

Object update

Optional attributes for object update:

- name
- user_xids

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**ProjectFile**(*manager: RESTManager[Any]*, *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

content: 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*, *start_branch: str | None = None*, ***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
- **start_branch** (*str | None*) – Name of the branch to start the new branch from
- ****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

start_branch: str | None = None

class gitlab.v4.objects.**ProjectFileManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CreateMixin*[*ProjectFile*], *UpdateMixin*[*ProjectFile*], *DeleteMixin*[*ProjectFile*]

Object Creation

Required attributes for object create:

- file_path
- branch
- content
- commit_message

Optional attributes for object create:

- encoding
- author_email
- author_name
- execute_filemode
- start_branch

Object update

Required attributes for object update:

- file_path
- branch
- content
- commit_message

Optional attributes for object update:

- encoding
- author_email
- author_name
- execute_filemode
- start_branch
- last_commit_id

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

blame(*file_path: str*, *ref: str*, ***kwargs:* Any) → list[dict[str, *Any*]]

Return the content of a file for a commit.

Parameters

- **file_path** (*str*) – Path of the file to retrieve
- **ref** (*str*) – Name of the branch, tag or commit
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns

A list of commits/lines matching the file

Return type

list[dict[str, Any]]

create(*data: dict[str, Any] | None = None*, ***kwargs: Any*) → *ProjectFile*

Create a new object.

Parameters

- **data** (*dict[str, Any] | None*) – parameters to send to the server to create the resource
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns

a new instance of the managed object class built with
the data sent by the server

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

Return type
ProjectFile

delete(*file_path: str*, *branch: str*, *commit_message: str*, ***kwargs: Any*) → None

Delete a file on the server.

Parameters

- **file_path** (*str*) – Path of the file to remove
- **branch** (*str*) – Branch from which the file will be removed
- **commit_message** (*str*) – Commit message for the deletion
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the server cannot perform the request

Return type
None

get(*file_path: str*, *ref: str*, ***kwargs: Any*) → *ProjectFile*

Retrieve a single file.

Parameters

- **file_path** (*str*) – Path of the file to retrieve
- **ref** (*str*) – Name of the branch, tag or commit
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
The generated RESTObject

Return type
ProjectFile

head(*file_path: str*, *ref: str*, ***kwargs: Any*) → CaseInsensitiveDict[*Any*]

Retrieve just metadata for 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 response headers as a dictionary

Return type
CaseInsensitiveDict[Any]

raw(*file_path: str*, *ref: str | None = None*, *streamed: Literal[False] = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → bytes

raw(*file_path: str*, *ref: str | None = None*, *streamed: bool = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[True] = True*, ***kwargs: Any*) → Iterator[*Any*]

raw(*file_path: str*, *ref: str | None = None*, *streamed: Literal[True] = True*, *action: Callable[[bytes], Any] | None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → None

Return the content of a file for a commit.

Parameters

- **file_path** (*str*) – Path of the file to return
- **ref** (*str | None*) – ID of the commit
- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk_size* and each chunk is passed to *action* for treatment
- **action** (*Callable[[...], Any] | None*) – Callable responsible for dealing with each chunk of data
- **chunk_size** (*int*) – Size of each chunk
- **iterator** (*bool*) – If True directly return the underlying response iterator
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
The file content

Return type
bytes | *Iterator*[Any] | None

update(*file_path: str*, *new_data: dict[str, Any] | None = None*, ***kwargs: Any*) → dict[str, *Any*]

Update an object on the server.

Parameters

- **id** – ID of the object to update (can be None if not required)
- **new_data** (*dict[str, Any] | None*) – the update data for the object
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)
- **file_path** (*str*)

Returns
The new object data (*not* a RESTObject)

Raises

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

Return type
dict[str, Any]

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

Bases: *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectForkManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CreateMixin*[*ProjectFork*], *ListMixin*[*ProjectFork*]

Object listing filters

- archived
- visibility
- order_by
- sort
- search
- simple
- owned
- membership
- starred
- statistics
- with_custom_attributes
- with_issues_enabled
- with_merge_requests_enabled

Object Creation

Optional attributes for object create:

- namespace

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

create(*data: dict[str, Any] | None = None*, ***kwargs: Any*) → *ProjectFork*

Creates a new object.

Parameters

- **data** (*dict[str, Any] | None*) – Parameters to send to the server to create the resource
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

Returns

A new instance of the managed object class build with the data sent by the server

Return type
ProjectFork

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

Bases: *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectGitignoreTemplateManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectGitignoreTemplate*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectGitlabciymlTemplateManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectGitlabciymlTemplate*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

test(*trigger: str*) → None

Test a Project Hook

Parameters
trigger (*str*) – Type of trigger event to test

Raises
GitlabHookTestError – If the hook test attempt failed

Return type
None

class gitlab.v4.objects.**ProjectHookManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*ProjectHook*]

Object Creation

Required attributes for object create:

- `url`

Optional attributes for object create:

- push_events
- issues_events
- confidential_issues_events
- merge_requests_events
- tag_push_events
- note_events
- job_events
- pipeline_events
- wiki_page_events
- enable_ssl_verification
- token

Object update

Required attributes for object update:

- url

Optional attributes for object update:

- push_events
- issues_events
- confidential_issues_events
- merge_requests_events
- tag_push_events
- note_events
- job_events
- pipeline_events
- wiki_events
- enable_ssl_verification
- token

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RefreshMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)

- **lazy** (*bool*)

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

Bases: *GetWithoutIdMixin*[*ProjectImport*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

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

Bases: *GetMixin*[*ProjectIntegration*], *UpdateMixin*[*ProjectIntegration*], *DeleteMixin*[*ProjectIntegration*], *ListMixin*[*ProjectIntegration*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

available() → List[str]

List the services known by python-gitlab.

Returns
The list of service code names.

Return type
List[str]

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectInvitationManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: InvitationMixin[*ProjectInvitation*]

Object listing filters

- query

Object Creation

Required attributes for object create:

- access_level

Optional attributes for object create:

- expires_at
- invite_source
- tasks_to_be_done
- tasks_project_id

Mutually exclusive attributes for object create:

- email
- user_id

Object update

Optional attributes for object update:

- access_level
- expires_at

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *UserAgentDetailMixin*, *SubscribableMixin*, *TodoMixin*, *TimeTrackingMixin*, *ParticipantsMixin*, *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str,* Any])
- **created_from_list** (*bool*)
- **lazy** (*bool*)

awardemojis: *ProjectIssueAwardEmojiManager*

closed_by(***kwargs:* Any) → *GitlabList* | list[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
- **GitlabGetErrot** – If the merge requests could not be retrieved

Returns
The list of merge requests.

Return type
GitlabList | list[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*) → *GitlabList* | list[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
- **GitlabGetErrot** – If the merge requests could not be retrieved

Returns
The list of merge requests.

Return type
GitlabList | list[dict[str, Any]]

reorder(*move_after_id: int | None = None*, *move_before_id: int | None = None*, ***kwargs: Any*) → None
Reorder an issue on a board.

Parameters

- **move_after_id** (*int | None*) – ID of an issue that should be placed after this issue
- **move_before_id** (*int | None*) – ID of an issue that should be placed before this issue
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct

- **GitlabUpdateError** – If the issue could not be reordered

Return type
None

resource_iteration_events: *ProjectIssueResourceIterationEventManager*

resource_weight_events: *ProjectIssueResourceWeightEventManager*

resourcelabelevents: *ProjectIssueResourceLabelEventManager*

resourcemilestoneevents: *ProjectIssueResourceMilestoneEventManager*

resourcestateevents: *ProjectIssueResourceStateEventManager*

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectIssueAwardEmojiManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *NoUpdateMixin*[*ProjectIssueAwardEmoji*]

Object Creation

Required attributes for object create:

- name

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

notes: *ProjectIssueDiscussionNoteManager*

class gitlab.v4.objects.**ProjectIssueDiscussionManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *RetrieveMixin*[*ProjectIssueDiscussion*], *CreateMixin*[*ProjectIssueDiscussion*]

Object Creation

Required attributes for object create:

- body

Optional attributes for object create:

- created_at

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectIssueDiscussionNoteManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CRUDMixin*[*ProjectIssueDiscussionNote*]

Object Creation

Required attributes for object create:

- body

Optional attributes for object create:

- created_at

Object update

Required attributes for object update:

- body

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectIssueLinkManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectIssueLink*], *CreateMixin*[*ProjectIssueLink*], *DeleteMixin*[*ProjectIssueLink*]

Object Creation

Required attributes for object create:

- target_project_id
- target_issue_iid

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

create(*data: dict[str, Any]*, ***kwargs: Any*) → tuple[*ProjectIssue*, *ProjectIssue*]

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[ProjectIssue, ProjectIssue]

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

Bases: *CRUDMixin*[*ProjectIssue*]

Object listing filters

- iids
- state
- labels
- milestone

- scope
- author_id
- iteration_id
- assignee_id
- my_reaction_emoji
- order_by
- sort
- search
- created_after
- created_before
- updated_after
- updated_before

Object Creation

Required attributes for object create:

- title

Optional attributes for object create:

- description
- confidential
- assignee_ids
- assignee_id
- milestone_id
- labels
- created_at
- due_date
- merge_request_to_resolve_discussions_of
- discussion_to_resolve

Object update

Optional attributes for object update:

- title
- description
- confidential
- assignee_ids
- assignee_id
- milestone_id
- labels
- state_event

- updated_at
- due_date
- discussion_locked

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

awardemojis: *ProjectIssueNoteAwardEmojiManager*

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectIssueNoteAwardEmojiManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *NoUpdateMixin*[*ProjectIssueNoteAwardEmoji*]

Object Creation

Required attributes for object create:

- name

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**ProjectIssueNoteManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CRUDMixin*[*ProjectIssueNote*]

Object Creation

Required attributes for object create:

- body

Optional attributes for object create:

- created_at

Object update

Required attributes for object update:

- body

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectIssueResourceIterationEventManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *RetrieveMixin*[*ProjectIssueResourceIterationEvent*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)

- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectIssueResourceLabelEventManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectIssueResourceLabelEvent*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectIssueResourceMilestoneEventManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectIssueResourceMilestoneEvent*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectIssueResourceStateEventManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectIssueResourceStateEvent*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectIssueResourceWeightEventManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectIssueResourceWeightEvent*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectIssueTemplateManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectIssueTemplate*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RefreshMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectIssuesStatisticsManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*ProjectIssuesStatistics*]

Object listing filters

- iids

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**ProjectIterationManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*GroupIteration*]

Object listing filters

- include_ancestors
- include_descendants
- in
- search
- state
- updated_after
- updated_before

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RefreshMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

artifact(*path: str*, *streamed: Literal[False] = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → bytes

artifact(*path: str*, *streamed: bool = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[True] = True*, ***kwargs: Any*) → Iterator[*Any*]

artifact(*path: str*, *streamed: Literal[True] = True*, *action: Callable[[bytes], Any] | None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → None

Get a single artifact file from within the job's artifacts archive.

Parameters

- **path** (*str*) – Path of the artifact
- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable[[...], Any] | None*) – Callable responsible of dealing with chunk of data
- **chunk_size** (*int*) – Size of each chunk
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns

The artifacts if *streamed* is False, None otherwise.

Return type

bytes | *Iterator*[Any] | None

artifacts(*streamed: Literal[False] = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → bytes

artifacts(*streamed: bool = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[True] = True*, ***kwargs: Any*) → Iterator[*Any*]

artifacts(*streamed: Literal[True] = True*, *action: Callable[[bytes], Any] | None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → None

Get the job artifacts.

Parameters

- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable[[...], Any] | None*) – Callable responsible of dealing with chunk of data
- **chunk_size** (*int*) – Size of each chunk

- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
The artifacts if *streamed* is False, None otherwise.

Return type
bytes | *Iterator*[Any] | None

cancel(***kwargs: Any*) → dict[str, *Any*]

Cancel the job.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabJobCancelError** – If the job could not be canceled

Return type
dict[str, Any]

delete_artifacts(***kwargs: Any*) → None

Delete artifacts of a job.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the request could not be performed

Return type
None

erase(***kwargs: Any*) → None

Erase the job (remove job artifacts and trace).

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabJobEraseError** – If the job could not be erased

Return type
None

keep_artifacts(***kwargs: Any*) → None

Prevent artifacts from being deleted when expiration is set.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the request could not be performed

Return type
None

play(***kwargs: Any*) → None

Trigger a job explicitly.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabJobPlayError** – If the job could not be triggered

Return type
None

retry(***kwargs: Any*) → dict[str, *Any*]

Retry the job.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabJobRetryError** – If the job could not be retried

Return type
dict[str, Any]

trace(*streamed: Literal[False] = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → bytes

trace(*streamed: bool = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[True] = True*, ***kwargs: Any*) → Iterator[*Any*]

trace(*streamed: Literal[True] = True*, *action: Callable[[bytes], Any] | None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → None

Get the job trace.

Parameters

- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable[[...], Any] | None*) – Callable responsible of dealing with chunk of data
- **chunk_size** (*int*) – Size of each chunk
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
The trace

Return type
bytes | *Iterator*[Any] | None

class gitlab.v4.objects.**ProjectJobManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *RetrieveMixin*[*ProjectJob*]

Object listing filters

- scope
- order_by
- sort

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RefreshMixin*, *SaveMixin*, *RESTObject*

Parameters

- **manager** (RESTManager*[*Any*]*)
- **attrs** (*dict[str,* Any*]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

allowlist: AllowlistProjectManager

groups_allowlist: AllowlistGroupManager

class gitlab.v4.objects.**ProjectJobTokenScopeManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *GetWithoutIdMixin*[*ProjectJobTokenScope*], *UpdateMixin*[*ProjectJobTokenScope*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (RESTManager*[*Any*]*)
- **attrs** (*dict[str,* Any*]*)

- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectKeyManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*ProjectKey*]

Object Creation

Required attributes for object create:

- title
- key

Optional attributes for object create:

- can_push
- expires_at

Object update

Optional attributes for object update:

- title
- can_push
- expires_at

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

enable(*key_id: int*, ***kwargs: Any*) → dict[str, *Any*] | Response

Enable a deploy key for a project.

Parameters

- **key_id** (*int*) – The ID of the key to enable
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabProjectDeployKeyError** – If the key could not be enabled

Returns

A dict of the result.

Return type

dict[str, Any] | *Response*

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

Bases: PromoteMixin, *SubscribableMixin*, *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (ProjectLabelManager)
- **attrs** (*dict[str, Any]*)

- **created_from_list** (*bool*)
- **lazy** (*bool*)

manager: *ProjectLabelManager*

save(***kwargs: Any*) → None

Saves the changes made to the object to the server.

The object is updated to match what the server returns.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct.
- **GitlabUpdateError** – If the server cannot perform the request.

Return type
None

class gitlab.v4.objects.**ProjectLabelManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectLabel*], *CreateMixin*[*ProjectLabel*], *UpdateMixin*[*ProjectLabel*], *DeleteMixin*[*ProjectLabel*]

Object Creation

Required attributes for object create:

- name
- color

Optional attributes for object create:

- description
- priority

Object update

Required attributes for object update:

- name

Optional attributes for object update:

- new_name
- color
- description
- priority

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

update(*name: str | None*, *new_data: dict[str, Any] | None = None*, ***kwargs: Any*) → dict[str, *Any*]

Update a Label on the server.

Parameters

- **name** (*str | None*) – The name of the label
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)
- **new_data** (*dict[str, Any] | None*)
- ****kwargs**

Return type

dict[str, Any]

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

Bases: *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectLicenseTemplateManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectLicenseTemplate*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**ProjectManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*Project*]

Object listing filters

- archived
- id_after
- id_before
- last_activity_after
- last_activity_before
- membership
- min_access_level
- order_by
- owned
- repository_checksum_failed

- repository_storage
- search_namespaces
- search
- simple
- sort
- starred
- statistics
- topic
- visibility
- wiki_checksum_failed
- with_custom_attributes
- with_issues_enabled
- with_merge_requests_enabled
- with_programming_language

Object Creation

Optional attributes for object create:

- name
- path
- allow_merge_on_skipped_pipeline
- only_allow_merge_if_all_status_checks_passed
- analytics_access_level
- approvals_before_merge
- auto_cancel_pending_pipelines
- auto_devops_deploy_strategy
- auto_devops_enabled
- autoclose_referenced_issues
- avatar
- build_coverage_regex
- build_git_strategy
- build_timeout
- builds_access_level
- ci_config_path
- container_expiration_policy_attributes
- container_registry_access_level
- container_registry_enabled
- default_branch

- description
- emails_disabled
- external_authorization_classification_label
- forking_access_level
- group_with_project_templates_id
- import_url
- initialize_with_readme
- issues_access_level
- issues_enabled
- jobs_enabled
- lfs_enabled
- merge_method
- merge_pipelines_enabled
- merge_requests_access_level
- merge_requests_enabled
- mirror_trigger_builds
- mirror
- namespace_id
- operations_access_level
- only_allow_merge_if_all_discussions_are_resolved
- only_allow_merge_if_pipeline_succeeds
- packages_enabled
- pages_access_level
- requirements_access_level
- printing_merge_request_link_enabled
- public_builds
- releases_access_level
- environments_access_level
- feature_flags_access_level
- infrastructure_access_level
- monitor_access_level
- remove_source_branch_after_merge
- repository_access_level
- repository_storage
- request_access_enabled
- resolve_outdated_diff_discussions

- security_and_compliance_access_level
- shared_runners_enabled
- show_default_award_emojis
- snippets_access_level
- snippets_enabled
- squash_option
- tag_list
- topics
- template_name
- template_project_id
- use_custom_template
- visibility
- wiki_access_level
- wiki_enabled

Object update

Optional attributes for object update:

- allow_merge_on_skipped_pipeline
- only_allow_merge_if_all_status_checks_passed
- analytics_access_level
- approvals_before_merge
- auto_cancel_pending_pipelines
- auto_devops_deploy_strategy
- auto_devops_enabled
- autoclose_referenced_issues
- avatar
- build_coverage_regex
- build_git_strategy
- build_timeout
- builds_access_level
- ci_config_path
- ci_default_git_depth
- ci_forward_deployment_enabled
- ci_allow_fork_pipelines_to_run_in_parent_project
- ci_separated_caches
- container_expiration_policy_attributes
- container_registry_access_level

- container_registry_enabled
- default_branch
- description
- emails_disabled
- enforce_auth_checks_on_uploads
- external_authorization_classification_label
- forking_access_level
- import_url
- issues_access_level
- issues_enabled
- issues_template
- jobs_enabled
- keep_latest_artifact
- lfs_enabled
- merge_commit_template
- merge_method
- merge_pipelines_enabled
- merge_requests_access_level
- merge_requests_enabled
- merge_requests_template
- merge_trains_enabled
- mirror_overwrites_diverged_branches
- mirror_trigger_builds
- mirror_user_id
- mirror
- mr_default_target_self
- name
- operations_access_level
- only_allow_merge_if_all_discussions_are_resolved
- only_allow_merge_if_pipeline_succeeds
- only_mirror_protected_branches
- packages_enabled
- pages_access_level
- requirements_access_level
- restrict_user_defined_variables
- path

- public_builds
- releases_access_level
- environments_access_level
- feature_flags_access_level
- infrastructure_access_level
- monitor_access_level
- remove_source_branch_after_merge
- repository_access_level
- repository_storage
- request_access_enabled
- resolve_outdated_diff_discussions
- security_and_compliance_access_level
- service_desk_enabled
- shared_runners_enabled
- show_default_award_emojis
- snippets_access_level
- snippets_enabled
- issue_branch_template
- squash_commit_template
- squash_option
- suggestion_commit_message
- tag_list
- topics
- visibility
- wiki_access_level
- wiki_enabled

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

import_bitbucket_server(*bitbucket_server_url: str*, *bitbucket_server_username: str*, *personal_access_token: str*, *bitbucket_server_project: str*, *bitbucket_server_repo: str*, *new_name: str | None = None*, *target_namespace: str | None = None*, ***kwargs:* Any) → dict[str, *Any*] | Response

Import a project from BitBucket Server to Gitlab (schedule the import)

This method will return when an import operation has been safely queued, or an error has occurred. After triggering an import, check the `import_status` of the newly created project to detect when the import operation has completed.

> **Note**
>
> This request may take longer than most other API requests. So this method will specify a 60 second default timeout if none is specified. A timeout can be specified via kwargs to override this functionality.

Parameters

- **bitbucket_server_url** (*str*) – Bitbucket Server URL
- **bitbucket_server_username** (*str*) – Bitbucket Server Username
- **personal_access_token** (*str*) – Bitbucket Server personal access token/password
- **bitbucket_server_project** (*str*) – Bitbucket Project Key
- **bitbucket_server_repo** (*str*) – Bitbucket Repository Name
- **new_name** (*str | None*) – New repository name (Optional)
- **target_namespace** (*str | None*) – Namespace to import repository into. Supports subgroups like /namespace/subgroup (Optional)
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns

A representation of the import status.

Return type

dict[str, Any] | *Response*

Example:

```
gl = gitlab.Gitlab_from_config()
print("Triggering import")
result = gl.projects.import_bitbucket_server(
    bitbucket_server_url="https://some.server.url",
    bitbucket_server_username="some_bitbucket_user",
    personal_access_token="my_password_or_access_token",
    bitbucket_server_project="my_project",
    bitbucket_server_repo="my_repo",
    new_name="gl_project_name",
    target_namespace="gl_project_path"
)
project = gl.projects.get(ret['id'])
print("Waiting for import to complete")
while project.import_status == u'started':
    time.sleep(1.0)
    project = gl.projects.get(project.id)
print("BitBucket import complete")
```

import_github(*personal_access_token: str*, *repo_id: int*, *target_namespace: str*, *new_name: str | None = None*, *github_hostname: str | None = None*, *optional_stages: dict[str, bool] | None = None*, ***kwargs: Any*) → dict[str, *Any*] | Response

Import a project from Github to Gitlab (schedule the import)

This method will return when an import operation has been safely queued, or an error has occurred. After triggering an import, check the `import_status` of the newly created project to detect when the import operation has completed.

> **Note**
>
> This request may take longer than most other API requests. So this method will specify a 60 second default timeout if none is specified. A timeout can be specified via kwargs to override this functionality.

Parameters

- **personal_access_token** (*str*) – GitHub personal access token
- **repo_id** (*int*) – Github repository ID
- **target_namespace** (*str*) – Namespace to import repo into
- **new_name** (*str | None*) – New repo name (Optional)
- **github_hostname** (*str | None*) – Custom GitHub Enterprise hostname. Do not set for GitHub.com. (Optional)
- **optional_stages** (*dict[str, bool] | None*) – Additional items to import. (Optional)
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns

A representation of the import status.

Return type

dict[str, Any] | *Response*

Example:

```
gl = gitlab.Gitlab_from_config()
print("Triggering import")
result = gl.projects.import_github(ACCESS_TOKEN,
                                   123456,
                                   "my-group/my-subgroup")
project = gl.projects.get(ret['id'])
print("Waiting for import to complete")
while project.import_status == u'started':
    time.sleep(1.0)
    project = gl.projects.get(project.id)
print("Github import complete")
```

import_project(*file: BufferedReader*, *path: str*, *name: str | None = None*, *namespace: str | None = None*, *overwrite: bool = False*, *override_params: dict[str, Any] | None = None*, ***kwargs: Any*) → dict[str, *Any*] | Response

Import a project from an archive file.

Parameters

- **file** (*BufferedReader*) – Data or file object containing the project
- **path** (*str*) – Name and path for the new project
- **name** (*str | None*) – The name of the project to import. If not provided, defaults to the path of the project.
- **namespace** (*str | None*) – The ID or path of the namespace that the project will be imported to
- **overwrite** (*bool*) – If True overwrite an existing project with the same path
- **override_params** (*dict[str, Any] | None*) – Set the specific settings for the project
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
A representation of the import status.

Return type
dict[str, Any] | *Response*

remote_import(*url: str*, *path: str*, *name: str | None = None*, *namespace: str | None = None*, *overwrite: bool = False*, *override_params: dict[str, Any] | None = None*, ***kwargs: Any*) → dict[str, *Any*] | Response

Import a project from an archive file stored on a remote URL.

Parameters

- **url** (*str*) – URL for the file containing the project data to import
- **path** (*str*) – Name and path for the new project
- **name** (*str | None*) – The name of the project to import. If not provided, defaults to the path of the project.
- **namespace** (*str | None*) – The ID or path of the namespace that the project will be imported to
- **overwrite** (*bool*) – If True overwrite an existing project with the same path
- **override_params** (*dict[str, Any] | None*) – Set the specific settings for the project
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
A representation of the import status.

Return type
dict[str, Any] | *Response*

remote_import_s3(*path: str*, *region: str*, *bucket_name: str*, *file_key: str*, *access_key_id: str*, *secret_access_key: str*, *name: str | None = None*, *namespace: str | None = None*, *overwrite: bool = False*, *override_params: dict[str, Any] | None = None*, ***kwargs: Any*) → dict[str, *Any*] | Response

Import a project from an archive file stored on AWS S3.

Parameters

- **region** (*str*) – AWS S3 region name where the file is stored
- **bucket_name** (*str*) – AWS S3 bucket name where the file is stored
- **file_key** (*str*) – AWS S3 file key to identify the file.
- **access_key_id** (*str*) – AWS S3 access key ID.
- **secret_access_key** (*str*) – AWS S3 secret access key.
- **path** (*str*) – Name and path for the new project
- **name** (*str | None*) – The name of the project to import. If not provided, defaults to the path of the project.
- **namespace** (*str | None*) – The ID or path of the namespace that the project will be imported to
- **overwrite** (*bool*) – If True overwrite an existing project with the same path
- **override_params** (*dict[str, Any] | None*) – Set the specific settings for the project
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
A representation of the import status.

Return type
dict[str, Any] | *Response*

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

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

Bases: *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectMemberAllManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *RetrieveMixin*[*ProjectMemberAll*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**ProjectMemberManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CRUDMixin*[*ProjectMember*]

Object Creation

Required attributes for object create:

- access_level

Optional attributes for object create:

- expires_at
- tasks_to_be_done

Mutually exclusive attributes for object create:

- username
- user_id

Object update

Required attributes for object update:

- access_level

Optional attributes for object update:

- expires_at

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SubscribableMixin*, *TodoMixin*, *TimeTrackingMixin*, *ParticipantsMixin*, *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (RESTManager[Any])

- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

approval_rules: *ProjectMergeRequestApprovalRuleManager*

approval_state: *ProjectMergeRequestApprovalStateManager*

approvals: *ProjectMergeRequestApprovalManager*

approve(*sha: str | None = None*, ***kwargs:* Any) → dict[str, *Any*]

Approve the merge request.

Parameters

- **sha** (*str | None*) – Head SHA of MR
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabMRApprovalError** – If the approval failed

Returns
A dict containing the result.

Return type
dict[str, Any]

https://docs.gitlab.com/ee/api/merge_request_approvals.html#approve-merge-request

awardemojis: *ProjectMergeRequestAwardEmojiManager*

cancel_merge_when_pipeline_succeeds(***kwargs:* Any) → dict[str, str]

Cancel merge when the pipeline succeeds.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabMROnBuildSuccessError** – If the server could not handle the request

Returns
dict of the parsed json returned by the server

Return type
dict[str, str]

changes(***kwargs:* Any) → dict[str, *Any*] | Response

List the merge request changes.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the list could not be retrieved

Returns
List of changes

Return type
dict[str, Any] | *Response*

closes_issues(***kwargs: Any*) → *RESTObjectList*[*ProjectIssue*]

List issues that will close on merge."

Parameters

- **get_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[ProjectIssue]

commits(***kwargs: Any*) → *RESTObjectList*[*ProjectCommit*]

List the merge request commits.

Parameters

- **get_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[ProjectCommit]

diffs: *ProjectMergeRequestDiffManager*

discussions: *ProjectMergeRequestDiscussionManager*

draft_notes: *ProjectMergeRequestDraftNoteManager*

merge(*merge_commit_message: str | None = None*, *should_remove_source_branch: bool | None = None*, *merge_when_pipeline_succeeds: bool | None = None*, ***kwargs: Any*) → dict[str, *Any*]

Accept the merge request.

Parameters

- **merge_commit_message** (*str | None*) – Commit message
- **should_remove_source_branch** (*bool | None*) – If True, removes the source branch
- **merge_when_pipeline_succeeds** (*bool | None*) – Wait for the build to succeed, then merge
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabMRClosedError** – If the merge failed

Return type
dict[str, Any]

merge_ref(***kwargs:* Any) → dict[str, *Any*] | Response

Attempt to merge changes between source and target branches into *refs/merge-requests/:iid/merge*.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises
GitlabGetError – If cannot be merged

Return type
dict[str, Any] | *Response*

notes: *ProjectMergeRequestNoteManager*

pipelines: *ProjectMergeRequestPipelineManager*

rebase(***kwargs:* Any) → dict[str, *Any*] | Response

Attempt to rebase the source branch onto the target branch

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabMRRebaseError** – If rebasing failed

Return type
dict[str, Any] | *Response*

related_issues(***kwargs:* Any) → *RESTObjectList*[*ProjectIssue*]

List issues related to this merge request."

Parameters

- **get_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[ProjectIssue]

reset_approvals(***kwargs: Any*) → dict[str, *Any*] | Response

Clear all approvals of the merge request.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabMRResetApprovalError** – If reset approval failed

Return type
dict[str, Any] | *Response*

resourcelabelevents: *ProjectMergeRequestResourceLabelEventManager*

resourcemilestoneevents: *ProjectMergeRequestResourceMilestoneEventManager*

resourcestateevents: *ProjectMergeRequestResourceStateEventManager*

reviewer_details: *ProjectMergeRequestReviewerDetailManager*

status_checks: *ProjectMergeRequestStatusCheckManager*

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

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

Bases: *SaveMixin*, *RESTObject*

Parameters

- **manager** (RESTManager[Any])

- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectMergeRequestApprovalManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*ProjectMergeRequestApproval*], *UpdateMixin*[*ProjectMergeRequestApproval*]

Object Creation

Object update

Required attributes for object update:

- approvals_required

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

set_approvers(*approvals_required: int*, *approver_ids: list[int] | None = None*, *approver_group_ids: list[int] | None = None*, *approval_rule_name: str = 'name'*, ***, *approver_usernames: list[str] | None = None*, ***kwargs: Any*) → *RESTObject*

Change MR-level allowed approvers and approver groups.

Parameters

- **approvals_required** (*int*) – The number of required approvals for this rule
- **approver_ids** (*list[int] | None*) – User IDs that can approve MRs
- **approver_group_ids** (*list[int] | None*) – Group IDs whose members can approve MRs
- **approval_rule_name** (*str*)
- **approver_usernames** (*list[str] | None*)
- **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[Any]*, *attrs: dict[str, Any]*, ***, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)

- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectMergeRequestApprovalRuleManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*ProjectMergeRequestApprovalRule*]

Object Creation

Required attributes for object create:

- name
- approvals_required

Optional attributes for object create:

- approval_project_rule_id
- user_ids
- group_ids
- usernames

Object update

Required attributes for object update:

- id
- merge_request_iid
- name
- approvals_required

Optional attributes for object update:

- user_ids
- group_ids
- usernames

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectMergeRequestApprovalStateManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*ProjectMergeRequestApprovalState*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectMergeRequestAwardEmojiManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *NoUpdateMixin*[*ProjectMergeRequestAwardEmoji*]

Object Creation

Required attributes for object create:

- name

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectMergeRequestDiffManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectMergeRequestDiff*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

notes: *ProjectMergeRequestDiscussionNoteManager*

class gitlab.v4.objects.**ProjectMergeRequestDiscussionManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectMergeRequestDiscussion*], *CreateMixin*[*ProjectMergeRequestDiscussion*], *UpdateMixin*[*ProjectMergeRequestDiscussion*]

Object Creation

Required attributes for object create:

- `body`

Optional attributes for object create:

- `created_at`
- `position`

Object update

Required attributes for object update:

- `resolved`

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (dict[str, Any])
- **created_from_list** (bool)
- **lazy** (bool)

class gitlab.v4.objects.**ProjectMergeRequestDiscussionNoteManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: CRUDMixin[ProjectMergeRequestDiscussionNote]

Object Creation

Required attributes for object create:

- body

Optional attributes for object create:

- created_at

Object update

Required attributes for object update:

- body

Parameters

- **gl** (Gitlab)
- **parent** (RESTObject | None)

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

Bases: ObjectDeleteMixin, SaveMixin, RESTObject

Parameters

- **manager** (RESTManager[Any])
- **attrs** (dict[str, Any])
- **created_from_list** (bool)
- **lazy** (bool)

publish(***kwargs: Any*) → None

Parameters
kwargs (Any)

Return type
None

class gitlab.v4.objects.**ProjectMergeRequestDraftNoteManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: CRUDMixin[ProjectMergeRequestDraftNote]

Object Creation

Required attributes for object create:

- note

Optional attributes for object create:

- commit_id
- in_reply_to_discussion_id
- position
- resolve_discussion

Object update

Optional attributes for object update:

- position

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

bulk_publish(***kwargs:* Any) → None

Parameters
kwargs (Any)

Return type
None

class gitlab.v4.objects.**ProjectMergeRequestManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CRUDMixin*[*ProjectMergeRequest*]

Object listing filters

- state
- order_by
- sort
- milestone
- view
- labels
- created_after
- created_before
- updated_after
- updated_before
- scope
- iids
- author_id
- assignee_id
- approver_ids
- approved_by_ids

- my_reaction_emoji
- source_branch
- target_branch
- search
- wip

Object Creation

Required attributes for object create:

- source_branch
- target_branch
- title

Optional attributes for object create:

- allow_collaboration
- allow_maintainer_to_push
- approvals_before_merge
- assignee_id
- assignee_ids
- description
- labels
- milestone_id
- remove_source_branch
- reviewer_ids
- squash
- target_project_id

Object update

Optional attributes for object update:

- target_branch
- assignee_id
- title
- description
- state_event
- labels
- milestone_id
- remove_source_branch
- discussion_locked
- allow_maintainer_to_push
- squash

- reviewer_ids

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

awardemojis: *ProjectMergeRequestNoteAwardEmojiManager*

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectMergeRequestNoteAwardEmojiManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *NoUpdateMixin*[*ProjectMergeRequestNoteAwardEmoji*]

Object Creation

Required attributes for object create:

- name

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**ProjectMergeRequestNoteManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*ProjectMergeRequestNote*]

Object Creation

Required attributes for object create:

- body

Object update

Required attributes for object update:

- body

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectMergeRequestPipelineManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *CreateMixin*[*ProjectMergeRequestPipeline*], *ListMixin*[*ProjectMergeRequestPipeline*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectMergeRequestResourceLabelEventManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectMergeRequestResourceLabelEvent*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectMergeRequestResourceMilestoneEventManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectMergeRequestResourceMilestoneEvent*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectMergeRequestResourceStateEventManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectMergeRequestResourceStateEvent*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectMergeRequestReviewerDetailManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectMergeRequestReviewerDetail*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectMergeRequestStatusCheckManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectMergeRequestStatusCheck*]

Object Creation

Object update

Required attributes for object update:

- sha
- external_status_check_id
- status

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectMergeRequestTemplateManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectMergeRequestTemplate*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectMergeTrainManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectMergeTrain*]

Object listing filters

- scope

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: PromoteMixin, *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

issues(***kwargs: Any*) → *RESTObjectList*[*ProjectIssue*]

List issues related to this milestone.

Parameters

- **get_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[ProjectIssue]

merge_requests(***kwargs: Any*) → *RESTObjectList*[*ProjectMergeRequest*]

List the merge requests related to this milestone.

Parameters

- **get_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[ProjectMergeRequest]

class gitlab.v4.objects.**ProjectMilestoneManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CRUDMixin*[*ProjectMilestone*]

Object listing filters

- `iids`
- `state`
- `search`

Object Creation

Required attributes for object create:

- `title`

Optional attributes for object create:

- `description`
- `due_date`
- `start_date`
- `state_event`

Object update

Optional attributes for object update:

- `title`
- `description`
- `due_date`
- `start_date`
- `state_event`

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str,* Any]*)*
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectNoteManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *RetrieveMixin*[*ProjectNote*]

Object Creation

Required attributes for object create:

- body

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *NotificationSettings*

Parameters

- **manager** (RESTManager*[*Any*]*)
- **attrs** (*dict[str,* Any*]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectNotificationSettingsManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *NotificationSettingsManager*

Object Creation

Object update

Optional attributes for object update:

- level
- notification_email
- new_note
- new_issue
- reopen_issue
- close_issue
- reassign_issue
- new_merge_request
- reopen_merge_request
- close_merge_request
- reassign_merge_request
- merge_merge_request

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

package_files: *ProjectPackageFileManager*

pipelines: *ProjectPackagePipelineManager*

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectPackageFileManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *DeleteMixin*[*ProjectPackageFile*], *ListMixin*[*ProjectPackageFile*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**ProjectPackageManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectPackage*], *GetMixin*[*ProjectPackage*], *DeleteMixin*[*ProjectPackage*]

Object listing filters

- order_by
- sort
- package_type
- package_name
- package_version
- include_versionless
- status

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectPackagePipelineManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectPackagePipeline*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *SaveMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectPackageProtectionRuleManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectPackageProtectionRule*], *CreateMixin*[*ProjectPackageProtectionRule*], *DeleteMixin*[*ProjectPackageProtectionRule*], *UpdateMixin*[*ProjectPackageProtectionRule*]

Object Creation

Required attributes for object create:

- `package_name_pattern`
- `package_type`
- `minimum_access_level_for_push`

Object update

Optional attributes for object update:

- package_name_pattern
- package_type
- minimum_access_level_for_push

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RefreshMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectPagesDomainManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*ProjectPagesDomain*]

Object Creation

Required attributes for object create:

- domain

Optional attributes for object create:

- certificate
- key

Object update

Optional attributes for object update:

- certificate
- key

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**ProjectPagesManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *DeleteMixin*[*ProjectPages*], *UpdateMixin*[*ProjectPages*], *GetWithoutIdMixin*[*ProjectPages*]

Object Creation

Object update

Optional attributes for object update:

- `pages_unique_domain_enabled`
- `pages_https_only`

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RefreshMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (RESTManager*[*Any*]*)
- **attrs** (*dict[str,* Any*]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

bridges: *ProjectPipelineBridgeManager*

cancel(***kwargs:* Any) → dict[str, *Any*] | Response

Cancel the job.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabPipelineCancelError** – If the request failed

Return type
dict[str, Any] | *Response*

jobs: *ProjectPipelineJobManager*

retry(***kwargs:* Any) → dict[str, *Any*] | Response

Retry the job.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabPipelineRetryError** – If the request failed

Return type
dict[str, Any] | *Response*

test_report: *ProjectPipelineTestReportManager*

test_report_summary: *ProjectPipelineTestReportSummaryManager*

variables: *ProjectPipelineVariableManager*

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectPipelineBridgeManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectPipelineBridge*]

Object listing filters

- scope

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectPipelineJobManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectPipelineJob*]

Object listing filters

- scope
- include_retried

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**ProjectPipelineManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *RetrieveMixin*[*ProjectPipeline*], *CreateMixin*[*ProjectPipeline*], *DeleteMixin*[*ProjectPipeline*]

Object listing filters

- scope
- status
- source
- ref
- sha
- yaml_errors
- name
- username
- order_by
- sort

Object Creation

Required attributes for object create:

- ref

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

create(*data: dict[str,* Any*]* | *None* = *None*, ***kwargs:* Any) → *ProjectPipeline*

Creates a new object.

Parameters

- **data** (*dict[str,* Any*] | None*) – Parameters to send to the server to create the resource
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

Returns

A new instance of the managed object class build with
the data sent by the server

Return type
ProjectPipeline

`latest`(*ref: str | None = None*, *lazy: bool = False*) → *ProjectPipeline*

Get the latest pipeline for the most recent commit
on a specific ref in a project

Parameters

- **ref** (*str | None*) – The branch or tag to check for the latest pipeline. Defaults to the default branch when not specified.
- **lazy** (*bool*)

Returns
A Pipeline instance

Return type
ProjectPipeline

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

pipelines: *ProjectPipelineSchedulePipelineManager*

`play`(***kwargs: Any*) → dict[str, *Any*]

Trigger a new scheduled pipeline, which runs immediately. The next scheduled run of this pipeline is not affected.

Parameters
****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabPipelinePlayError** – If the request failed

Return type
dict[str, Any]

`take_ownership`(***kwargs: Any*) → None

Update the owner of a pipeline schedule.

Parameters
****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabOwnershipError** – If the request failed

Return type
None

variables: *ProjectPipelineScheduleVariableManager*

class gitlab.v4.objects.**ProjectPipelineScheduleManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*ProjectPipelineSchedule*]

Object Creation

Required attributes for object create:

- description
- ref
- cron

Optional attributes for object create:

- cron_timezone
- active

Object update

Optional attributes for object update:

- description
- ref
- cron
- cron_timezone
- active

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectPipelineSchedulePipelineManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectPipelineSchedulePipeline*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectPipelineScheduleVariableManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CreateMixin*[*ProjectPipelineScheduleVariable*], *UpdateMixin*[*ProjectPipelineScheduleVariable*], *DeleteMixin*[*ProjectPipelineScheduleVariable*]

Object Creation

Required attributes for object create:

- `key`
- `value`

Object update

Required attributes for object update:

- `key`
- `value`

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)

- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectPipelineTestReportManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*ProjectPipelineTestReport*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectPipelineTestReportSummaryManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*ProjectPipelineTestReportSummary*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectPipelineVariableManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectPipelineVariable*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectProtectedBranchManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*ProjectProtectedBranch*]

Object Creation

Required attributes for object create:

- name

Optional attributes for object create:

- push_access_level
- merge_access_level
- unprotect_access_level
- allow_force_push
- allowed_to_push
- allowed_to_merge
- allowed_to_unprotect
- code_owner_approval_required

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectProtectedEnvironmentManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectProtectedEnvironment*], *CreateMixin*[*ProjectProtectedEnvironment*], *DeleteMixin*[*ProjectProtectedEnvironment*]

Object Creation

Required attributes for object create:

- `name`
- `deploy_access_levels`

Optional attributes for object create:

- `required_approval_count`
- `approval_rules`

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectProtectedTagManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *NoUpdateMixin*[*ProjectProtectedTag*]

Object Creation

Required attributes for object create:

- `name`

Optional attributes for object create:

- create_access_level

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectPullMirrorManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*ProjectPullMirror*], *UpdateMixin*[*ProjectPullMirror*]

Object Creation

Object update

Optional attributes for object update:

- url

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

create(*data: dict[str, Any], **kwargs: Any*) → *ProjectPullMirror*

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

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
ProjectPullMirror

start(***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

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectPushRulesManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*ProjectPushRules*], *CreateMixin*[*ProjectPushRules*], *UpdateMixin*[*ProjectPushRules*], *DeleteMixin*[*ProjectPushRules*]

Object Creation

Optional attributes for object create:

- author_email_regex
- branch_name_regex
- commit_committer_check
- commit_message_negative_regex
- commit_message_regex
- deny_delete_tag
- file_name_regex
- max_file_size
- member_check
- prevent_secrets
- reject_unsigned_commits

Object update

Optional attributes for object update:

- author_email_regex
- branch_name_regex
- commit_committer_check

- commit_message_negative_regex
- commit_message_regex
- deny_delete_tag
- file_name_regex
- max_file_size
- member_check
- prevent_secrets
- reject_unsigned_commits

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectRegistryProtectionRuleManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectRegistryProtectionRule*], *CreateMixin*[*ProjectRegistryProtectionRule*], *UpdateMixin*[*ProjectRegistryProtectionRule*]

Object Creation

Required attributes for object create:

- repository_path_pattern

Optional attributes for object create:

- minimum_access_level_for_push
- minimum_access_level_for_delete

Object update

Optional attributes for object update:

- repository_path_pattern
- minimum_access_level_for_push
- minimum_access_level_for_delete

Parameters

- **gl** (*Gitlab*)

- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

tags: *ProjectRegistryTagManager*

class gitlab.v4.objects.**ProjectRegistryRepositoryManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *DeleteMixin*[*ProjectRegistryRepository*], *ListMixin*[*ProjectRegistryRepository*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectRegistryRepositoryProtectionRuleManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectRegistryRepositoryProtectionRule*], *CreateMixin*[*ProjectRegistryRepositoryProtec*
UpdateMixin[*ProjectRegistryRepositoryProtectionRule*], *DeleteMixin*[*ProjectRegistryRepositoryProtectio*

Object Creation

Required attributes for object create:

- `repository_path_pattern`

Optional attributes for object create:

- `minimum_access_level_for_push`

- minimum_access_level_for_delete

Object update

Optional attributes for object update:

- repository_path_pattern
- minimum_access_level_for_push
- minimum_access_level_for_delete

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectRegistryTagManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *DeleteMixin*[*ProjectRegistryTag*], *RetrieveMixin*[*ProjectRegistryTag*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

delete_in_bulk(*name_regex_delete: str*, ***kwargs: Any*) → None

Delete Tag in bulk

Parameters

- **name_regex_delete** (*str*) – The regex of the name to delete. To delete all tags specify .*.
- **keep_n** – The amount of latest tags of given name to keep.
- **name_regex_keep** – The regex of the name to keep. This value overrides any matches from name_regex.
- **older_than** – Tags to delete that are older than the given time, written in human readable form 1h, 1d, 1month.
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct

- **GitlabDeleteError** – If the server cannot perform the request

Return type
None

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

Bases: *SaveMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

links: *ProjectReleaseLinkManager*

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

Bases: *ObjectDeleteMixin*, *SaveMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectReleaseLinkManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*ProjectReleaseLink*]

Object Creation

Required attributes for object create:

- name
- url

Optional attributes for object create:

- filepath
- direct_asset_path
- link_type

Object update

Optional attributes for object update:

- name
- url
- filepath
- direct_asset_path
- link_type

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**ProjectReleaseManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*ProjectRelease*]

Object listing filters

- order_by
- sort
- include_html_description

Object Creation

Required attributes for object create:

- tag_name

Optional attributes for object create:

- name
- description
- ref
- assets

Object update

Optional attributes for object update:

- name
- description
- milestones
- released_at

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *SaveMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

sync(***kwargs: Any*) → dict[str, *Any*] | Response

Force push mirror update.

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

dict[str, Any] | *Response*

class gitlab.v4.objects.**ProjectRemoteMirrorManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectRemoteMirror*], *CreateMixin*[*ProjectRemoteMirror*], *UpdateMixin*[*ProjectRemoteMirror*], *DeleteMixin*[*ProjectRemoteMirror*]

Object Creation

Required attributes for object create:

- `url`

Optional attributes for object create:

- `enabled`
- `only_protected_branches`

Object update

Optional attributes for object update:

- `enabled`
- `only_protected_branches`

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

upcoming_jobs: *ProjectResourceGroupUpcomingJobManager*

class gitlab.v4.objects.**ProjectResourceGroupManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectResourceGroup*], *UpdateMixin*[*ProjectResourceGroup*]

Object listing filters

- order_by
- sort
- include_html_description

Object Creation

Object update

Optional attributes for object update:

- process_mode

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectResourceGroupUpcomingJobManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectResourceGroupUpcomingJob*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectRunnerManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CreateMixin*[*ProjectRunner*], *DeleteMixin*[*ProjectRunner*], *ListMixin*[*ProjectRunner*]

Object listing filters

- scope
- tag_list

Object Creation

Required attributes for object create:

- runner_id

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str,* Any])
- **created_from_list** (*bool*)
- **lazy** (*bool*)

download(*streamed: Literal[False] = False, action: None = None, chunk_size: int = 1024, *, iterator: Literal[False] = False, **kwargs:* Any) → bytes

download(*streamed: bool = False, action: None = None, chunk_size: int = 1024, *, iterator: Literal[True] = True, **kwargs:* Any) → Iterator[*Any*]

download(*streamed: Literal[True] = True, action: Callable[[bytes],* Any] | *None = None, chunk_size: int = 1024, *, iterator: Literal[False] = False, **kwargs:* Any) → None

Download the secure file.

Parameters

- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable[[bytes],* Any] | *None*) – Callable responsible of dealing with chunk of data
- **chunk_size** (*int*) – Size of each chunk
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
The artifacts if *streamed* is False, None otherwise.

Return type
bytes | *Iterator*[Any] | None

class gitlab.v4.objects.**ProjectSecureFileManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *NoUpdateMixin*[*ProjectSecureFile*]

Object Creation

Required attributes for object create:

- `name`
- `file`

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ProjectIntegration*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str,* Any])
- **created_from_list** (*bool*)
- **lazy** (*bool*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str,* Any])
- **created_from_list** (*bool*)
- **lazy** (*bool*)

access_tokens: *ProjectServiceAccountAccessTokenManager*

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

Bases: *ObjectDeleteMixin*, ObjectRotateMixin, *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str,* Any])

- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectServiceAccountAccessTokenManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CreateMixin*[*ProjectServiceAccountAccessToken*], *DeleteMixin*[*ProjectServiceAccountAccessToken*], *ListMixin*[*ProjectServiceAccountAccessToken*], RotateMixin[*ProjectServiceAccountAccessToken*]

Object listing filters

- created_after
- created_before
- expires_after
- expires_before
- last_used_after
- last_used_before
- revoked
- search
- sort
- state

Object Creation

Required attributes for object create:

- name
- scopes

Optional attributes for object create:

- description
- expires_at

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**ProjectServiceAccountManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CreateMixin*[*ProjectServiceAccount*], *DeleteMixin*[*ProjectServiceAccount*], *ListMixin*[*ProjectServiceAccount*], *UpdateMixin*[*ProjectServiceAccount*]

Object listing filters

- order_by
- sort

Object Creation

Optional attributes for object create:

- name

- username
- email

Object update

Optional attributes for object update:

- name
- username
- email

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**ProjectServiceManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ProjectIntegrationManager*

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *UserAgentDetailMixin*, *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

awardemojis: *ProjectSnippetAwardEmojiManager*

content(*streamed: Literal[False] = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → bytes

content(*streamed: bool = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[True] = True*, ***kwargs: Any*) → Iterator[*Any*]

content(*streamed: Literal[True] = True*, *action: Callable[[bytes], Any] | None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → None

Return the content of a snippet.

Parameters

- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk_size* and each chunk is passed to *action* for treatment.
- **iterator** (*bool*) – If True directly return the underlying response iterator

- **action** (*Callable[[...], Any] | None*) – Callable responsible of dealing with chunk of data
- **chunk_size** (*int*) – Size of each chunk
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
The snippet content

Return type
bytes | *Iterator*[Any] | None

discussions: *ProjectSnippetDiscussionManager*

notes: *ProjectSnippetNoteManager*

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectSnippetAwardEmojiManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *NoUpdateMixin*[*ProjectSnippetAwardEmoji*]

Object Creation

Required attributes for object create:

- name

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)

- **created_from_list** (*bool*)
- **lazy** (*bool*)

notes: *ProjectSnippetDiscussionNoteManager*

class gitlab.v4.objects.**ProjectSnippetDiscussionManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*ProjectSnippetDiscussion*], *CreateMixin*[*ProjectSnippetDiscussion*]

Object Creation

Required attributes for object create:

- body

Optional attributes for object create:

- created_at

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectSnippetDiscussionNoteManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetMixin*[*ProjectSnippetDiscussionNote*], *CreateMixin*[*ProjectSnippetDiscussionNote*], *UpdateMixin*[*ProjectSnippetDiscussionNote*], *DeleteMixin*[*ProjectSnippetDiscussionNote*]

Object Creation

Required attributes for object create:

- body

Optional attributes for object create:

- created_at

Object update

Required attributes for object update:

- body

Parameters

- **gl** (*Gitlab*)

- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**ProjectSnippetManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*ProjectSnippet*]

Object Creation

Required attributes for object create:

- `title`
- `visibility`

Optional attributes for object create:

- `description`
- `content`

Mutually exclusive attributes for object create:

- `files`
- `file_name`

Object update

Optional attributes for object update:

- `title`
- `files`
- `file_name`
- `content`
- `visibility`
- `description`

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

awardemojis: *ProjectSnippetNoteAwardEmojiManager*

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectSnippetNoteAwardEmojiManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *NoUpdateMixin*[*ProjectSnippetNoteAwardEmoji*]

Object Creation

Required attributes for object create:

- name

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**ProjectSnippetNoteManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*ProjectSnippetNote*]

Object Creation

Required attributes for object create:

- body

Object update

Required attributes for object update:

- body

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RefreshMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)

- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectStorageManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*ProjectStorage*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectTagManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *NoUpdateMixin*[*ProjectTag*]

Object listing filters

- `order_by`
- `sort`
- `search`

Object Creation

Required attributes for object create:

- `tag_name`
- `ref`

Optional attributes for object create:

- `message`

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)

- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectTriggerManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*ProjectTrigger*]

Object Creation

Required attributes for object create:

- `description`

Object update

Required attributes for object update:

- `description`

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectUserManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*ProjectUser*]

Object listing filters

- `search`
- `skip_users`

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)

- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectVariableManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*ProjectVariable*]

Object Creation

Required attributes for object create:

- key
- value

Optional attributes for object create:

- description
- environment_scope
- masked
- masked_and_hidden
- protected
- raw
- variable_type

Object update

Required attributes for object update:

- key

Optional attributes for object update:

- description
- environment_scope
- masked
- protected
- raw
- value
- variable_type

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, UploadMixin, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)

- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ProjectWikiManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*ProjectWiki*]

Object listing filters

- `with_content`

Object Creation

Required attributes for object create:

- `title`
- `content`

Optional attributes for object create:

- `format`

Object update

Optional attributes for object update:

- `title`
- `content`
- `format`

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**RegistryRepositoryManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetMixin*[*RegistryRepository*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**RepositoryMixin**

Bases: object

delete_merged_branches(***kwargs: Any*) → None

Delete merged branches.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

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

Return type
None

repository_archive(*sha: str | None = None*, *streamed: Literal[False] = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → bytes

repository_archive(*sha: str | None = None*, *streamed: bool = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[True] = True*, ***kwargs: Any*) → Iterator[*Any*]

repository_archive(*sha: str | None = None*, *streamed: Literal[True] = True*, *action: Callable[[bytes], Any] | None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → None

Return an archive of the repository.

Parameters
- **sha** (*str | None*) – ID of the commit (default branch by default)
- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable[[...], Any] | None*) – Callable responsible of dealing with chunk of data
- **chunk_size** (*int*) – Size of each chunk
- **format** (*str | None*) – file format (tar.gz by default)
- **path** (*str | None*) – The subpath of the repository to download (all files by default)
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

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

Returns
The binary data of the archive

Return type
bytes | *Iterator*[Any] | None

repository_blob(*sha: str*, ***kwargs: Any*) → dict[str, *Any*] | Response

Return a file by blob SHA.

Parameters

- **sha** (*str*) – ID of the blob
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
The blob content and metadata

Return type
dict[str, Any] | *Response*

repository_compare(*from_: str*, *to: str*, ***kwargs: Any*) → dict[str, *Any*] | Response

Return a diff between two branches/commits.

Parameters

- **from** – Source branch/SHA
- **to** (*str*) – Destination branch/SHA
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)
- **from_** (*str*)

Raises

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

Returns
The diff

Return type
dict[str, Any] | *Response*

repository_contributors(***kwargs: Any*) → *GitlabList* | list[dict[str, *Any*]]

Return a list of contributors for the project.

Parameters

- **get_all** – If True, return all the items, without pagination
- **per_page** – Number of items to retrieve per request
- **page** – ID of the page to return (starts with page 1)
- **iterator** – If set to True and no pagination option is defined, return a generator instead of a list
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
The contributors

Return type
GitlabList | list[dict[str, Any]]

repository_merge_base(*refs: list[str]*, ***kwargs: Any*) → dict[str, *Any*] | Response

Return a diff between two branches/commits.

Parameters

- **refs** (*list[str]*) – The refs to find the common ancestor of. Multiple refs can be passed.
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns

The common ancestor commit (*not* a RESTObject)

Return type

dict[str, Any] | *Response*

repository_raw_blob(*sha: str*, *streamed: Literal[False] = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → bytes

repository_raw_blob(*sha: str*, *streamed: bool = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[True] = True*, ***kwargs: Any*) → Iterator[*Any*]

repository_raw_blob(*sha: str*, *streamed: Literal[True] = True*, *action: Callable[[bytes], Any] | None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → None

Return the raw file contents for a blob.

Parameters

- **sha** (*str*) – ID of the blob
- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable[[...], Any] | None*) – Callable responsible of dealing with chunk of data
- **chunk_size** (*int*) – Size of each chunk
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns

The blob content if streamed is False, None otherwise

Return type

bytes | *Iterator*[Any] | None

repository_tree(*path: str = ''*, *ref: str = ''*, *recursive: bool = False*, ***kwargs: Any*) → *GitlabList* | list[dict[str, *Any*]]

Return a list of files in the repository.

Parameters

- **path** (*str*) – Path of the top folder (/ by default)
- **ref** (*str*) – Reference to a commit or branch
- **recursive** (*bool*) – Whether to get the tree recursively
- **get_all** – If True, return all the items, without pagination
- **per_page** – Number of items to retrieve per request
- **page** – ID of the page to return (starts with page 1)
- **iterator** – If set to True and no pagination option is defined, return a generator instead of a list
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

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

Returns
The representation of the tree

Return type
GitlabList | list[dict[str, Any]]

update_submodule(*submodule: str*, *branch: str*, *commit_sha: str*, ***kwargs:* Any) → dict[str, *Any*] | Response

Update a project submodule

Parameters

- **submodule** (*str*) – Full path to the submodule
- **branch** (*str*) – Name of the branch to commit into
- **commit_sha** (*str*) – Full commit SHA to update the submodule to
- **commit_message** – Commit message. If no message is provided, a default one will be set (optional)
- **kwargs** (Any)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabPutError** – If the submodule could not be updated

Return type
dict[str, Any] | *Response*

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

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (RESTManager*[*Any*]*)
- **attrs** (*dict[str,* Any*]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

jobs: *RunnerJobManager*

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**RunnerAllManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*RunnerAll*]

Object listing filters

- scope
- type
- status
- paused
- tag_list

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**RunnerJobManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*RunnerJob*]

Object listing filters

- status
- order_by
- sort

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**RunnerManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CRUDMixin*[*Runner*]

Object listing filters

- scope
- type
- status
- paused
- tag_list

Object Creation

Required attributes for object create:

- token

Optional attributes for object create:

- description
- info
- active
- locked
- run_untagged
- tag_list
- access_level
- maximum_timeout

Object update

Optional attributes for object update:

- description
- active
- tag_list
- run_untagged
- locked
- access_level
- maximum_timeout

Parameters

- **gl** (*Gitlab*)

- **parent** (*RESTObject | None*)

all(*scope: str | None = None*, ***kwargs: Any*) → list[*Runner*]

List all the runners.

Parameters

- **scope** (*str | None*) – The scope of runners to show, one of: specific, shared, active, paused, online
- **get_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]

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.**ServiceAccount**(*manager: RESTManager[Any]*, *attrs: dict[str, Any]*, ***, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *SaveMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**ServiceAccountManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CreateMixin*[*ServiceAccount*], *ListMixin*[*ServiceAccount*], *UpdateMixin*[*ServiceAccount*]

Object listing filters

- order_by
- sort

Object Creation

Optional attributes for object create:

- name
- username
- email

Object update

Optional attributes for object update:

- name
- username
- email

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**SharedProject**(*manager:* RESTManager[Any], *attrs: dict[str,* Any], *, *created_from_list: bool* = *False*, *lazy: bool* = *False*)

Bases: *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str,* Any]*)*
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**SharedProjectManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *ListMixin*[*SharedProject*]

Object listing filters

- archived
- visibility
- order_by
- sort
- search
- simple
- starred

- with_issues_enabled
- with_merge_requests_enabled
- min_access_level
- with_custom_attributes

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**SidekiqManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *RESTManager*[*RESTObject*]

Manager for the Sidekiq methods.

This manager doesn't actually manage objects but provides helper function for the sidekiq metrics API.

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

compound_metrics(***kwargs:* Any) → dict[str, *Any*] | Response

Return all available metrics and statistics.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the information couldn't be retrieved

Returns
All available Sidekiq metrics and statistics

Return type
dict[str, Any] | *Response*

job_stats(***kwargs:* Any) → dict[str, *Any*] | Response

Return statistics about the jobs performed.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the information couldn't be retrieved

Returns
Statistics about the Sidekiq jobs performed

Return type
dict[str, Any] | *Response*

process_metrics(***kwargs: Any*) → dict[str, *Any*] | Response

Return the registered sidekiq workers.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises
- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the information couldn't be retrieved

Returns
Information about the register Sidekiq worker

Return type
dict[str, Any] | *Response*

queue_metrics(***kwargs: Any*) → dict[str, *Any*] | Response

Return the registered queues information.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises
- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the information couldn't be retrieved

Returns
Information about the Sidekiq queues

Return type
dict[str, Any] | *Response*

class gitlab.v4.objects.**Snippet**(*manager: RESTManager[Any], attrs: dict[str, Any], *, created_from_list: bool = False, lazy: bool = False*)

Bases: *UserAgentDetailMixin*, *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters
- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

content(*streamed: Literal[False] = False, action: None = None, chunk_size: int = 1024, *, iterator: Literal[False] = False, **kwargs: Any*) → bytes

content(*streamed: bool = False, action: None = None, chunk_size: int = 1024, *, iterator: Literal[True] = True, **kwargs: Any*) → Iterator[*Any*]

content(*streamed: Literal[True] = True, action: Callable[[bytes], Any] | None = None, chunk_size: int = 1024, *, iterator: Literal[False] = False, **kwargs: Any*) → None

Return the content of a snippet.

Parameters

- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk_size* and each chunk is passed to *action* for treatment.
- **iterator** (*bool*) – If True directly return the underlying response iterator
- **action** (*Callable[[...], Any] | None*) – Callable responsible of dealing with chunk of data
- **chunk_size** (*int*) – Size of each chunk
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the content could not be retrieved

Returns
The snippet content

Return type
bytes | *Iterator*[Any] | None

class gitlab.v4.objects.**SnippetManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CRUDMixin*[*Snippet*]

Object Creation

Required attributes for object create:

- `title`

Optional attributes for object create:

- `description`
- `content`
- `visibility`

Mutually exclusive attributes for object create:

- `files`
- `file_name`

Object update

Optional attributes for object update:

- `title`
- `files`
- `file_name`
- `content`
- `visibility`
- `description`

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

list_all(*, *iterator: Literal[False] = False*, ***kwargs:* Any) → list[*Snippet*]

list_all(*, *iterator: Literal[True] = True*, ***kwargs:* Any) → RESTObjectList[*Snippet*]

list_all(*, *iterator: bool = False*, ***kwargs:* Any) → RESTObjectList[*Snippet*] | list[*Snippet*]

List all snippets.

Parameters

- **get_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** (*bool*) – 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

GitlabListError – If the list could not be retrieved

Returns

A generator for the snippets list

Return type

RESTObjectList[Snippet] | list[Snippet]

list_public(*, *iterator: Literal[False] = False*, ***kwargs:* Any) → list[*Snippet*]

list_public(*, *iterator: Literal[True] = True*, ***kwargs:* Any) → RESTObjectList[*Snippet*]

list_public(*, *iterator: bool = False*, ***kwargs:* Any) → RESTObjectList[*Snippet*] | list[*Snippet*]

List all public snippets.

Parameters

- **get_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** (*bool*) – 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

GitlabListError – If the list could not be retrieved

Returns

The list of snippets, or a generator if *iterator* is True

Return type

RESTObjectList[Snippet] | list[Snippet]

public(*, *iterator: Literal[False] = False*, *page: int | None = None*, ***kwargs:* Any) → list[*Snippet*]

public(*, *iterator: Literal[True] = True*, ***kwargs:* Any) → *RESTObjectList*[*Snippet*]

public(*, *iterator: bool = False*, ***kwargs:* Any) → *RESTObjectList*[*Snippet*] | list[*Snippet*]

List all public snippets.

Parameters

- **get_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** (*bool*) – 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
GitlabListError – If the list could not be retrieved

Returns
The list of snippets, or a generator if *iterator* is True

Return type
RESTObjectList[Snippet] | list[Snippet]

class gitlab.v4.objects.**StarredProject**(*manager: RESTManager[Any]*, *attrs: dict[str, Any]*, *, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**StarredProjectManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *ListMixin*[*StarredProject*]

Object listing filters

- archived
- membership
- min_access_level
- order_by
- owned
- search
- simple
- sort
- starred
- statistics
- visibility
- with_custom_attributes
- with_issues_enabled
- with_merge_requests_enabled

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**Todo**(*manager: RESTManager[Any], attrs: dict[str, Any], *, created_from_list: bool = False, lazy: bool = False*)

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

mark_as_done(***kwargs: Any*) → Dict[str, *Any*]

Mark the todo as done.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabTodoError** – If the server failed to perform the request

Returns
A dict with the result

Return type
Dict[str, Any]

class gitlab.v4.objects.**TodoManager**(*gl: Gitlab, parent: RESTObject | None = None*)

Bases: *ListMixin*[*Todo*], *DeleteMixin*[*Todo*]

Object listing filters

- action
- author_id
- project_id
- state
- type

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

mark_all_as_done(***kwargs: Any*) → None

Mark all the todos as done.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises
- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabTodoError** – If the server failed to perform the request

Returns
The number of todos marked done

Return type
None

class gitlab.v4.objects.**Topic**(*manager: RESTManager[Any]*, *attrs: dict[str, Any]*, ***, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters
- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**TopicManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*Topic*]

Object Creation

Required attributes for object create:
- name

Optional attributes for object create:
- avatar
- description
- title

Object update

Optional attributes for object update:
- avatar
- description
- name

Parameters
- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

merge(*source_topic_id: int | str*, *target_topic_id: int | str*, ***kwargs: Any*) → dict[str, *Any*]

Merge two topics, assigning all projects to the target topic.

Parameters

- **source_topic_id** (*int | str*) – ID of source project topic
- **target_topic_id** (*int | str*) – ID of target project topic
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabTopicMergeError** – If the merge failed

Returns
The merged topic data (*not* a RESTObject)

Return type
dict[str, Any]

class gitlab.v4.objects.**User**(*manager: RESTManager[Any]*, *attrs: dict[str, Any]*, ***, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

activate(***kwargs: Any*) → dict[str, *Any*] | Response

Activate the user.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabActivateError** – If the user could not be activated

Returns
Whether the user status has been changed

Return type
dict[str, Any] | *Response*

approve(***kwargs: Any*) → dict[str, *Any*] | Response

Approve a user creation request.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUserApproveError** – If the user could not be activated

Returns
The new object data (*not* a RESTObject)

Return type
dict[str, Any] | *Response*

ban(***kwargs: Any*) → dict[str, *Any*] | Response

Ban the user.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises
- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabBanError** – If the user could not be banned

Returns
Whether the user has been banned

Return type
dict[str, Any] | *Response*

block(***kwargs: Any*) → bool | None

Block the user.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises
- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabBlockError** – If the user could not be blocked

Returns
Whether the user status has been changed

Return type
bool | None

contributed_projects: *UserContributedProjectManager*

customattributes: *UserCustomAttributeManager*

deactivate(***kwargs: Any*) → dict[str, *Any*] | Response

Deactivate the user.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises
- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeactivateError** – If the user could not be deactivated

Returns
Whether the user status has been changed

Return type
dict[str, Any] | *Response*

emails: *UserEmailManager*

events: *UserEventManager*

follow(***kwargs: Any*) → dict[str, *Any*] | Response

Follow the user.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises
- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabFollowError** – If the user could not be followed

Returns
The new object data (*not* a RESTObject)

Return type
dict[str, Any] | *Response*

followers_users: **UserFollowersManager**

following_users: **UserFollowingManager**

gpgkeys: *UserGPGKeyManager*

identityproviders: *UserIdentityProviderManager*

impersonationtokens: *UserImpersonationTokenManager*

keys: *UserKeyManager*

memberships: *UserMembershipManager*

personal_access_tokens: *UserPersonalAccessTokenManager*

projects: *UserProjectManager*

reject(***kwargs: Any*) → dict[str, *Any*] | Response

Reject a user creation request.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises
- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUserRejectError** – If the user could not be rejected

Returns
The new object data (*not* a RESTObject)

Return type
dict[str, Any] | *Response*

starred_projects: *StarredProjectManager*

status: *UserStatusManager*

unban(***kwargs: Any*) → dict[str, *Any*] | Response

Unban the user.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUnbanError** – If the user could not be unbanned

Returns
Whether the user has been unbanned

Return type
dict[str, Any] | *Response*

unblock(***kwargs: Any*) → bool | None

Unblock the user.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUnblockError** – If the user could not be unblocked

Returns
Whether the user status has been changed

Return type
bool | None

unfollow(***kwargs: Any*) → dict[str, *Any*] | Response

Unfollow the user.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUnfollowError** – If the user could not be followed

Returns
The new object data (*not* a RESTObject)

Return type
dict[str, Any] | *Response*

class gitlab.v4.objects.**UserActivities**(*manager: RESTManager[Any]*, *attrs: dict[str, Any]*, ***, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**UserActivitiesManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *ListMixin*[*UserActivities*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**UserContributedProject**(*manager:* RESTManager[Any], *attrs: dict[str,* Any], **, created_from_list: bool = False, lazy: bool = False*)

Bases: *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str,* Any])
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**UserContributedProjectManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *ListMixin*[*UserContributedProject*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**UserCustomAttribute**(*manager:* RESTManager[Any], *attrs: dict[str,* Any], **, created_from_list: bool = False, lazy: bool = False*)

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str,* Any])
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**UserCustomAttributeManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *RetrieveMixin*[*UserCustomAttribute*], *SetMixin*[*UserCustomAttribute*], *DeleteMixin*[*UserCustomAttribute*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)

- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**UserEmail**(*manager: RESTManager[Any]*, *attrs: dict[str, Any]*, ***, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**UserEmailManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*UserEmail*], *CreateMixin*[*UserEmail*], *DeleteMixin*[*UserEmail*]

Object Creation

Required attributes for object create:

- email

Optional attributes for object create:

- skip_confirmation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**UserEvent**(*manager: RESTManager[Any]*, *attrs: dict[str, Any]*, ***, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *Event*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**UserEventManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *EventManager*

Object listing filters

- action
- target_type
- before
- after
- sort
- scope

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**UserGPGKey**(*manager: RESTManager[Any]*, *attrs: dict[str, Any]*, ***, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**UserGPGKeyManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*UserGPGKey*], *CreateMixin*[*UserGPGKey*], *DeleteMixin*[*UserGPGKey*]

Object Creation

Required attributes for object create:

- key

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**UserIdentityProviderManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *DeleteMixin*[*User*]

Manager for user identities.

This manager does not actually manage objects but enables functionality for deletion of user identities by provider.

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**UserImpersonationToken**(*manager: RESTManager[Any]*, *attrs: dict[str, Any]*, ***, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)

- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**UserImpersonationTokenManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *NoUpdateMixin*[*UserImpersonationToken*]

Object listing filters

- `state`

Object Creation

Required attributes for object create:

- `name`
- `scopes`

Optional attributes for object create:

- `expires_at`

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**UserKey**(*manager: RESTManager[Any]*, *attrs: dict[str, Any]*, ***, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**UserKeyManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*UserKey*], *CreateMixin*[*UserKey*], *DeleteMixin*[*UserKey*]

Object Creation

Required attributes for object create:

- `title`
- `key`

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**UserManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CRUDMixin*[*User*]

Object listing filters

- username
- public_email
- search
- active
- external
- blocked
- humans
- created_after
- created_before
- exclude_active
- exclude_external
- exclude_humans
- exclude_internal
- without_project_bots
- extern_uid
- provider
- two_factor
- without_projects
- admins
- auditors
- skip_ldap
- custom_attributes
- status

Object Creation

Optional attributes for object create:

- email
- username
- name
- password
- reset_password
- skype
- linkedin
- twitter

- projects_limit
- extern_uid
- provider
- bio
- admin
- can_create_group
- website_url
- skip_confirmation
- external
- organization
- location
- avatar
- public_email
- private_profile
- color_scheme_id
- theme_id

Object update

Required attributes for object update:

- email
- username
- name

Optional attributes for object update:

- password
- skype
- linkedin
- twitter
- projects_limit
- extern_uid
- provider
- bio
- admin
- can_create_group
- website_url
- skip_reconfirmation
- external
- organization

- location
- avatar
- public_email
- private_profile
- color_scheme_id
- theme_id

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**UserMembership**(*manager: RESTManager[Any]*, *attrs: dict[str, Any]*, ***, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**UserMembershipManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RetrieveMixin*[*UserMembership*]

Object listing filters

- type

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**UserPersonalAccessToken**(*manager: RESTManager[Any]*, *attrs: dict[str, Any]*, ***, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**UserPersonalAccessTokenManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *CreateMixin*[*UserPersonalAccessToken*]

Object Creation

Required attributes for object create:

- name
- scopes

Optional attributes for object create:

- expires_at

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**UserProject**(*manager:* RESTManager[Any], *attrs: dict[str,* Any], *, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *RESTObject*

Parameters

- **manager** (RESTManager[Any])
- **attrs** (*dict[str,* Any]*)*
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**UserProjectManager**(*gl:* Gitlab, *parent:* RESTObject | *None = None*)

Bases: *ListMixin*[*UserProject*], *CreateMixin*[*UserProject*]

Object listing filters

- archived
- visibility
- order_by
- sort
- search
- simple
- owned
- membership
- starred
- statistics
- with_issues_enabled
- with_merge_requests_enabled
- with_custom_attributes

- with_programming_language
- wiki_checksum_failed
- repository_checksum_failed
- min_access_level
- id_after
- id_before

Object Creation

Required attributes for object create:

- name

Optional attributes for object create:

- default_branch
- issues_enabled
- wall_enabled
- merge_requests_enabled
- wiki_enabled
- snippets_enabled
- squash_option
- public
- visibility
- description
- builds_enabled
- public_builds
- import_url
- only_allow_merge_if_build_succeeds

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

list(*, *iterator: Literal[False] = False*, ***kwargs:* Any) → *list*[*UserProject*]
list(*, *iterator: Literal[True] = True*, ***kwargs:* Any) → *RESTObjectList*[*UserProject*]
list(*, *iterator: bool = False*, ***kwargs:* Any) → *RESTObjectList*[*UserProject*] | *list*[*UserProject*]

Retrieve a list of objects.

Parameters

- **get_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** (*bool*) – If set to True and no pagination option is defined, return a generator instead of a list
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns
The list of objects, or a generator if *iterator* is True

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the server cannot perform the request

Return type
RESTObjectList[UserProject] | *list*[UserProject]

class gitlab.v4.objects.**UserStatus**(*manager: RESTManager[Any]*, *attrs: dict[str, Any]*, ***, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**UserStatusManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetWithoutIdMixin*[*UserStatus*]

Object Creation

Object update

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.v4.objects.**Variable**(*manager: RESTManager[Any]*, *attrs: dict[str, Any]*, ***, *created_from_list: bool = False*, *lazy: bool = False*)

Bases: *SaveMixin*, *ObjectDeleteMixin*, *RESTObject*

Parameters

- **manager** (*RESTManager[Any]*)
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

class gitlab.v4.objects.**VariableManager**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *CRUDMixin*[*Variable*]

Object Creation

Required attributes for object create:

- key

- value

Optional attributes for object create:

- description
- masked
- protected
- raw
- variable_type

Object update

Required attributes for object update:

- key

Optional attributes for object update:

- description
- masked
- protected
- raw
- value
- variable_type

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

gitlab.v4.objects.**overload**(*func*)

Decorator for overloaded functions/methods.

In a stub file, place two or more stub definitions for the same function in a row, each decorated with @overload.

For example:

```
@overload
def utf8(value: None) -> None: ...
@overload
def utf8(value: bytes) -> bytes: ...
@overload
def utf8(value: str) -> bytes: ...
```

In a non-stub file (i.e. a regular .py file), do the same but follow it with an implementation. The implementation should *not* be decorated with @overload:

```
@overload
def utf8(value: None) -> None: ...
@overload
def utf8(value: bytes) -> bytes: ...
@overload
def utf8(value: str) -> bytes: ...
```

(continues on next page)

(continued from previous page)

```
def utf8(value):
    ...  # implementation goes here
```

The overloads for a function can be retrieved at runtime using the get_overloads() function.

Module contents

7.7.3 Submodules

7.7.4 gitlab.base module

class gitlab.base.**RESTManager**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: Generic[TObjCls]

Base class for CRUD operations on objects.

Derived class must define _path and _obj_cls.

_path: Base URL path on which requests will be sent (e.g. '/projects') _obj_cls: The class of objects that will be created

Parameters

- **gl** (*Gitlab*)
- **parent** (RESTObject | *None*)

gitlab: *Gitlab*

property parent_attrs: dict[str, *Any*] | None

property path: str

class gitlab.base.**RESTObject**(*manager:* RESTManager[Any], *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[Any])
- **attrs** (*dict[str, Any]*)
- **created_from_list** (*bool*)
- **lazy** (*bool*)

asdict(*, *with_parent_attrs: bool* = *False*) → dict[str, *Any*]

Parameters
with_parent_attrs (*bool*)

Return type
dict[str, Any]

property attributes: dict[str, *Any*]

property encoded_id: int | str | None

Ensure that the ID is url-encoded so that it can be safely used in a URL path

get_id() → int | str | None

Returns the id of the resource.

Return type
int | str | None

manager: *RESTManager*[*Any*]

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[TObjCls]*, *obj_cls: type[TObjCls]*, *_list: GitlabList*)

Bases: Generic[TObjCls]

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[TObjCls]*) – Manager to attach to the created objects
- **obj_cls** (*type[TObjCls]*) – Type of objects to create from the json data
- **_list** (*GitlabList*) – A GitlabList object

property current_page: int

The current page number.

next() → TObjCls

Return type
TObjCls

property next_page: int | None

The next page number.

If None, the current page is the last.

property per_page: int | None

The number of items per page.

property prev_page: int | None

The previous page number.

If None, the current page is the first.

property total: int | None

The total number of items.

property total_pages: int | None

The total number of pages.

7.7.5 gitlab.cli module

class gitlab.cli.**CustomAction**(*required: 'tuple[str, ...]'*, *optional: 'tuple[str, ...]'*, *in_object: 'bool'*, *requires_id: 'bool'*, *help: 'str | None'*)

Bases: object

Parameters

- **required** (*tuple[str, ...]*)
- **optional** (*tuple[str, ...]*)
- **in_object** (*bool*)
- **requires_id** (*bool*)
- **help** (*str | None*)

help: str | None

in_object: bool

optional: tuple[str, ...]

required: tuple[str, ...]

requires_id: bool

gitlab.cli.**cls_to_gitlab_resource**(*cls: type[RESTObject]*) → str

Parameters

cls (*type[RESTObject]*)

Return type

str

gitlab.cli.**die**(*msg: str*, *e: Exception | None = None*) → NoReturn

Parameters

- **msg** (*str*)
- **e** (*Exception | None*)

Return type
NoReturn

gitlab.cli.**docs**() → ArgumentParser

Provide a statically generated parser for sphinx only, so we don't need to provide dummy gitlab config for readthedocs.

Return type
ArgumentParser

gitlab.cli.**gitlab_resource_to_cls**(*gitlab_resource: str*, *namespace: ModuleType*) → type[*RESTObject*]

Parameters

- **gitlab_resource** (*str*)
- **namespace** (*ModuleType*)

Return type
type[RESTObject]

gitlab.cli.**main**() → None

Return type
None

gitlab.cli.**register_custom_action**(*, *cls_names: str | tuple[str, ...]*, *required: tuple[str, ...] = ()*, *optional: tuple[str, ...] = ()*, *custom_action: str | None = None*, *requires_id: bool = True*, *help: str | None = None*) → Callable[[__F], __F]

Parameters

- **cls_names** (*str | tuple[str, ...]*)
- **required** (*tuple[str, ...]*)
- **optional** (*tuple[str, ...]*)
- **custom_action** (*str | None*)
- **requires_id** (*bool*)
- **help** (*str | None*)

Return type
Callable[[__F], __F]

7.7.6 gitlab.config module

exception gitlab.config.**ConfigError**

Bases: Exception

exception gitlab.config.**GitlabConfigHelperError**

Bases: *ConfigError*

exception gitlab.config.**GitlabConfigMissingError**

Bases: *ConfigError*

class gitlab.config.**GitlabConfigParser**(*gitlab_id: str | None = None*, *config_files: list[str] | None = None*)

Bases: object

Parameters

- **gitlab_id** (*str | None*)
- **config_files** (*list[str] | None*)

exception gitlab.config.**GitlabDataError**

Bases: *ConfigError*

exception gitlab.config.**GitlabIDError**

Bases: *ConfigError*

7.7.7 gitlab.const module

class gitlab.const.**AccessLevel**(**values*)

Bases: IntEnum

ADMIN = 60

DEVELOPER = 30

GUEST = 10

MAINTAINER = 40

MINIMAL_ACCESS = 5

NO_ACCESS = 0

OWNER = 50

PLANNER = 15

REPORTER = 20

SECURITY_MANAGER = 25

class gitlab.const.**NotificationLevel**(**values*)

Bases: GitlabEnum

CUSTOM = 'custom'

DISABLED = 'disabled'

GLOBAL = 'global'

MENTION = 'mention'

PARTICIPATING = 'participating'

WATCH = 'watch'

class gitlab.const.**SearchScope**(**values*)

Bases: GitlabEnum

BLOBS = 'blobs'

COMMITS = 'commits'

GLOBAL_SNIPPET_TITLES = 'snippet_titles'

ISSUES = 'issues'

MERGE_REQUESTS = 'merge_requests'

MILESTONES = 'milestones'

PROJECTS = 'projects'

PROJECT_NOTES = 'notes'

USERS = 'users'

WIKI_BLOBS = 'wiki_blobs'

class gitlab.const.**Visibility**(**values*)

Bases: GitlabEnum

INTERNAL = 'internal'

PRIVATE = 'private'

PUBLIC = 'public'

7.7.8 gitlab.exceptions module

exception gitlab.exceptions.**GitlabActivateError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabAttachFileError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabAuthenticationError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabError*

Parameters

- **error_message** (*str | bytes*)

- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabBanError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabBlockError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabBuildCancelError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabCancelError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabBuildEraseError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabRetryError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabBuildPlayError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabRetryError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabBuildRetryError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabRetryError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabCancelError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabCherryPickError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabCiLintError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabConnectionError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabCreateError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabDeactivateError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabDeleteError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabDeploymentApprovalError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: Exception

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabFollowError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabGetError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabGroupTransferError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabHeadError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabHookTestError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabHousekeepingError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)

- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabHttpError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabImportError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabInvitationError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabJobCancelError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabCancelError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabJobEraseError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabRetryError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabJobPlayError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabRetryError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabJobRetryError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabRetryError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabLicenseError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabListError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabMRApprovalError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabMRClosedError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabMRForbiddenError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabMROnBuildSuccessError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabMRRebaseError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabMRResetApprovalError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabMarkdownError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabOperationError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabOwnershipError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabParsingError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabPipelineCancelError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabCancelError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabPipelinePlayError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabRetryError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabPipelineRetryError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabRetryError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabProjectDeployKeyError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabPromoteError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabProtectError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabRenderError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabRepairError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabRestoreError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabRetryError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)

- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabRevertError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabRotateError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabSearchError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabSetError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabStopError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabSubscribeError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabTimeTrackingError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabTodoError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabTopicMergeError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabTransferProjectError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabUnbanError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabUnblockError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabUnfollowError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabUnsubscribeError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabUpdateError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabUploadError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabUserApproveError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)

- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabUserRejectError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**GitlabVerifyError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabOperationError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

exception gitlab.exceptions.**RedirectError**(*error_message: str | bytes = ''*, *response_code: int | None = None*, *response_body: bytes | None = None*)

Bases: *GitlabError*

Parameters

- **error_message** (*str | bytes*)
- **response_code** (*int | None*)
- **response_body** (*bytes | None*)

Return type
None

7.7.9 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*[*Any*]

class gitlab.mixins.**BadgeRenderMixin**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *RESTManager*[TObjCls]

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *GetMixin*[TObjCls], *ListMixin*[TObjCls], *CreateMixin*[TObjCls], *UpdateMixin*[TObjCls], *DeleteMixin*[TObjCls]

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.mixins.**CreateMixin**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *RESTManager*[TObjCls]

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

create(*data: dict[str,* Any*]* | *None* = *None*, ***kwargs:* Any) → TObjCls

Create a new object.

Parameters

- **data** (*dict[str, Any] | None*) – parameters to send to the server to create the resource
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

Returns

A new instance of the managed object class built with
the data sent by the server

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabCreateError** – If the server cannot perform the request

Return type
TObjCls

class gitlab.mixins.**DeleteMixin**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RESTManager*[TObjCls]

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

delete(*id: str | int | None = None*, ***kwargs: Any*) → None

Delete an object on the server.

Parameters

- **id** (*str | int | None*) – ID of the object to delete
- ****kwargs** (*Any*) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabDeleteError** – If the server cannot perform the request

Return type
None

class gitlab.mixins.**DownloadMixin**

Bases: object

download(*streamed: Literal[False] = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → bytes

download(*streamed: bool = False*, *action: None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[True] = True*, ***kwargs: Any*) → Iterator[Any]

download(*streamed: Literal[True] = True*, *action: Callable[[bytes], Any] | None = None*, *chunk_size: int = 1024*, ***, *iterator: Literal[False] = False*, ***kwargs: Any*) → None

Download the archive of a resource export.

Parameters

- **streamed** (*bool*) – If True the data will be processed by chunks of *chunk_size* and each chunk is passed to *action* for treatment
- **iterator** (*bool*) – If True directly return the underlying response iterator

- **action** (*Callable[[bytes], Any] | None*) – Callable responsible of dealing with chunk of data
- **chunk_size** (*int*) – Size of each chunk
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server failed to perform the request

Returns
The blob content if streamed is False, None otherwise

Return type
bytes | *Iterator*[Any] | None

manager: *RESTManager*[*Any*]

class gitlab.mixins.**GetMixin**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: HeadMixin[TObjCls]

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

get(*id: str | int*, *lazy: bool* = *False*, ***kwargs:* Any) → TObjCls

Retrieve a single object.

Parameters

- **id** (*str | int*) – ID of the object to retrieve
- **lazy** (*bool*) – If True, don't request the server, but create a shallow object giving access to the managers. This is useful if you want to avoid useless calls to the API.
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns
The generated RESTObject.

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

Return type
TObjCls

class gitlab.mixins.**GetWithoutIdMixin**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: HeadMixin[TObjCls]

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

get(***kwargs: Any*) → TObjCls

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
TObjCls

class gitlab.mixins.**ListMixin**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: HeadMixin[TObjCls]

Parameters
- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

list(*, *iterator: Literal[False] = False*, ***kwargs: Any*) → *list*[base.TObjCls]

list(*, *iterator: Literal[True] = True*, ***kwargs: Any*) → base.RESTObjectList[base.TObjCls]

list(*, *iterator: bool = False*, ***kwargs: Any*) → base.RESTObjectList[base.TObjCls] | *list*[base.TObjCls]

Retrieve a list of objects.

Parameters
- **get_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** (*bool*) – If set to True and no pagination option is defined, return a generator instead of a list
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns
The list of objects, or a generator if *iterator* is True

Raises
- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabListError** – If the server cannot perform the request

Return type
RESTObjectList[*TObjCls*] | *list*[*TObjCls*]

class gitlab.mixins.**NoUpdateMixin**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *GetMixin*[TObjCls], *ListMixin*[TObjCls], *CreateMixin*[TObjCls], *DeleteMixin*[TObjCls]

Parameters
- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

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*[*Any*]

class gitlab.mixins.**ParticipantsMixin**

Bases: object

manager: *RESTManager*[*Any*]

participants(***kwargs: Any*) → *GitlabList* | list[dict[str, *Any*]]

List the participants.

Parameters
- **get_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
GitlabList | list[dict[str, Any]]

class gitlab.mixins.**RefreshMixin**

Bases: object

manager: *RESTManager*[*Any*]

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**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *ListMixin*[TObjCls], *GetMixin*[TObjCls]

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

class gitlab.mixins.**SaveMixin**

Bases: object

Mixin for RESTObject's that can be updated.

manager: *RESTManager*[*Any*]

save(***kwargs:* Any) → dict[str, *Any*] | None

Save the changes made to the object to the server.

The object is updated to match what the server returns.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns
The new object data (*not* a RESTObject)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUpdateError** – If the server cannot perform the request

Return type
dict[str, Any] | None

class gitlab.mixins.**SetMixin**(*gl:* Gitlab, *parent:* RESTObject | *None* = *None*)

Bases: *RESTManager*[TObjCls]

Parameters

- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

set(*key: str*, *value: str*, ***kwargs:* Any) → TObjCls

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
TObjCls

class gitlab.mixins.**SubscribableMixin**

Bases: object

manager: *RESTManager*[*Any*]

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*[*Any*]

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*[*Any*]

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**(*gl: Gitlab*, *parent: RESTObject | None = None*)

Bases: *RESTManager*[TObjCls]

Parameters
- **gl** (*Gitlab*)
- **parent** (*RESTObject | None*)

update(*id: str | int | None = None*, *new_data: dict[str, Any] | None = None*, ***kwargs: Any*) → dict[str, *Any*]

Update an object on the server.

Parameters
- **id** (*str | int | None*) – ID of the object to update (can be None if not required)
- **new_data** (*dict[str, Any] | None*) – the update data for the object
- ****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Returns
The new object data (*not* a RESTObject)

Raises
- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabUpdateError** – If the server cannot perform the request

Return type
dict[str, Any]

class gitlab.mixins.**UserAgentDetailMixin**

Bases: object

manager: *RESTManager*[*Any*]

user_agent_detail(***kwargs: Any*) → dict[str, *Any*]

Get the user agent detail.

Parameters
****kwargs** (Any) – Extra options to send to the server (e.g. sudo)

Raises

- **GitlabAuthenticationError** – If authentication is not correct
- **GitlabGetError** – If the server cannot perform the request

Return type
dict[str, Any]

7.7.10 gitlab.utils module

class gitlab.utils.**EncodedId**(*value: str | int | EncodedId*)

Bases: str

A custom *str* class that will return the URL-encoded value of the string.

- Using it recursively will only url-encode the value once.
- Can accept either *str* or *int* as input value.
- Can be used in an f-string and output the URL-encoded string.

Reference to documentation on why this is necessary.

See:

```
https://docs.gitlab.com/ee/api/index.html#namespaced-path-encoding
https://docs.gitlab.com/ee/api/index.html#path-parameters
```

Parameters
value (*str | int | EncodedId*)

Return type
EncodedId

class gitlab.utils.**MaskingFormatter**(*fmt: str | None = '%(levelname)s:%(name)s:%(message)s'*, *datefmt: str | None = None*, *style: Literal['%', '{', '$'] = '%'*, *validate: bool = True*, *masked: str | None = None*)

Bases: Formatter

A logging formatter that can mask credentials

Parameters

- **fmt** (*str | None*)
- **datefmt** (*str | None*)
- **style** (*Literal['%', '{', '$']*)
- **validate** (*bool*)
- **masked** (*str | None*)

format(*record: LogRecord*) → str

Format the specified record as text.

The record's attribute dictionary is used as the operand to a string formatting operation which yields the returned string. Before formatting the dictionary, a couple of preparatory steps are carried out. The message attribute of the record is computed using LogRecord.getMessage(). If the formatting string uses the time (as determined by a call to usesTime(), formatTime() is called to format the event time. If there is exception information, it is formatted using formatException() and appended to the message.

Parameters
record (*LogRecord*)

Return type
str

class gitlab.utils.**Retry**(*max_retries: int*, *obey_rate_limit: bool | None = True*, *retry_transient_errors: bool | None = False*)

Bases: object

Parameters

- **max_retries** (*int*)
- **obey_rate_limit** (*bool | None*)
- **retry_transient_errors** (*bool | None*)

handle_retry() → bool

Return type
bool

handle_retry_on_status(*status_code: int | None*, *headers: MutableMapping[str, str] | None = None*, *reason: str = ''*) → bool

Parameters

- **status_code** (*int | None*)
- **headers** (*MutableMapping[str, str] | None*)
- **reason** (*str*)

Return type
bool

class gitlab.utils.**WarnMessageData**(*message: 'str'*, *show_caller: 'bool'*)

Bases: object

Parameters

- **message** (*str*)
- **show_caller** (*bool*)

message: str

show_caller: bool

gitlab.utils.**copy_dict**(*, *src: dict[str, Any]*, *dest: dict[str, Any]*) → None

Parameters

- **src** (*dict[str, Any]*)
- **dest** (*dict[str, Any]*)

Return type
None

gitlab.utils.**get_base_url**(*url: str | None = None*) → str

Return the base URL with the trailing slash stripped. If the URL is a Falsy value, return the default URL. :returns: The base URL

Parameters
url (*str | None*)

Return type
str

gitlab.utils.**get_content_type**(*content_type: str | None*) → str

Parameters
content_type (*str | None*)

Return type
str

gitlab.utils.**remove_none_from_dict**(*data: dict[str, Any]*) → dict[str, *Any*]

Parameters
data (*dict[str, Any]*)

Return type
dict[str, Any]

gitlab.utils.**response_content**(*response: Response*, *streamed: bool*, *action: Callable[[bytes], Any] | None*, *chunk_size: int*, *, *iterator: bool*) → bytes | Iterator[*Any*] | None

Parameters

- **response** (*Response*)
- **streamed** (*bool*)
- **action** (*Callable[[bytes], Any] | None*)
- **chunk_size** (*int*)
- **iterator** (*bool*)

Return type
bytes | *Iterator*[Any] | None

gitlab.utils.**warn**(*message: str*, *, *category: type[Warning] | None = None*, *source: Any | None = None*, *show_caller: bool = True*) → None

This *warnings.warn* wrapper function attempts to show the location causing the warning in the user code that called the library.

It does this by walking up the stack trace to find the first frame located outside the *gitlab/* directory. This is helpful to users as it shows them their code that is causing the warning.

Parameters

- **message** (*str*)
- **category** (*type[Warning] | None*)
- **source** (*Any | None*)
- **show_caller** (*bool*)

Return type
None

7.8 CLI reference (`gitlab` command)

> **Warning**
>
> The following is a complete, auto-generated list of subcommands available via the **`gitlab`** command-line tool. Some of the actions may currently not work as expected or lack functionality available via the API.
>
> Please see the existing list of CLI related issues, or open a new one if it is not already listed there.

7.8.1 gitlab

GitLab API Command Line Interface

```
usage: gitlab [-h] [--version] [-v] [-d] [-c CONFIG_FILE] [-g GITLAB]
              [-o {json,legacy,yaml}] [-f FIELDS] [--server-url SERVER_URL]
              [--ssl-verify SSL_VERIFY | --no-ssl-verify] [--timeout TIMEOUT]
              [--api-version API_VERSION] [--per-page PER_PAGE]
              [--pagination PAGINATION] [--order-by ORDER_BY]
              [--user-agent USER_AGENT] [--private-token PRIVATE_TOKEN |
              --oauth-token OAUTH_TOKEN | --job-token JOB_TOKEN]
              [--skip-login] [--no-mask-credentials]
              {application,application-appearance,application-settings,application-
↪statistics,audit-event,broadcast-message,bulk-import,bulk-import-all-entity,bulk-
↪import-entity,ci-lint,current-user,current-user-email,current-user-gpg-key,current-
↪user-key,current-user-runner,current-user-status,deploy-key,deploy-token,dockerfile,
↪event,feature,generic-package,geo-node,gitignore,gitlabciyml,group,group-access-
↪request,group-access-token,group-approval-rule,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-ldap-group-link,group-label,group-member,group-member-all,group-member-role,
↪group-merge-request,group-milestone,group-notification-settings,group-package,group-
↪project,group-push-rules,group-runner,group-saml-group-link,group-service-account,
↪group-service-account-access-token,group-subgroup,group-variable,group-wiki,hook,issue,
↪issues-statistics,key,ldap-group,license,member-role,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-cluster-agent,
↪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-dockerfile-template,
↪project-environment,project-event,project-export,project-external-status-check,project-
↪feature-flag,project-feature-flag-user-list,project-file,project-fork,project-
↪gitignore-template,project-gitlabciyml-template,project-hook,project-import,project-
↪integration,project-invitation,project-issue,project-issue-award-emoji,project-issue-
↪discussion,project-issue-discussion-note,project-issue-link,project-issue-note,project-
↪issue-note-award-emoji,project-issue-resource-iteration-event,project-issue-resource-
↪label-event,project-issue-resource-milestone-event,project-issue-resource-state-event,
↪project-issue-resource-weight-event,project-issue-template,project-issues-statistics,
```

(continues on next page)

(continued from previous page)

```
↪project-job,project-job-token-scope,project-key,project-label,project-license-template,
↪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-draft-note,project-merge-
↪request-note,project-merge-request-note-award-emoji,project-merge-request-pipeline,
↪project-merge-request-resource-label-event,project-merge-request-resource-milestone-
↪event,project-merge-request-resource-state-event,project-merge-request-reviewer-detail,
↪project-merge-request-status-check,project-merge-request-template,project-merge-train,
↪project-milestone,project-note,project-notification-settings,project-package,project-
↪package-file,project-package-pipeline,project-package-protection-rule,project-pages,
↪project-pages-domain,project-pipeline,project-pipeline-bridge,project-pipeline-job,
↪project-pipeline-schedule,project-pipeline-schedule-pipeline,project-pipeline-schedule-
↪variable,project-pipeline-test-report,project-pipeline-test-report-summary,project-
↪pipeline-variable,project-protected-branch,project-protected-environment,project-
↪protected-tag,project-pull-mirror,project-push-rules,project-registry-protection-rule,
↪project-registry-repository,project-registry-repository-protection-rule,project-
↪registry-tag,project-release,project-release-link,project-remote-mirror,project-
↪resource-group,project-resource-group-upcoming-job,project-runner,project-secure-file,
↪project-service,project-service-account,project-service-account-access-token,project-
↪snippet,project-snippet-award-emoji,project-snippet-discussion,project-snippet-
↪discussion-note,project-snippet-note,project-snippet-note-award-emoji,project-storage,
↪project-tag,project-trigger,project-user,project-variable,project-wiki,registry-
↪repository,runner,runner-all,runner-job,service-account,shared-project,snippet,starred-
↪project,todo,topic,user,user-activities,user-contributed-project,user-custom-attribute,
↪user-email,user-event,user-gpg-key,user-impersonation-token,user-key,user-membership,
↪user-personal-access-token,user-project,user-status,variable} ...
```

-h, --help
: show this help message and exit

--version
: Display the version.

-v, --verbose, --fancy
: Verbose mode (legacy format only) [env var: GITLAB_VERBOSE]

-d, --debug
: Debug mode (display HTTP requests) [env var: GITLAB_DEBUG]

-c <config_file>, **--config-file** <config_file>
: Configuration file to use. Can be used multiple times. [env var: PYTHON_GITLAB_CFG]

-g <gitlab>, **--gitlab** <gitlab>
: Which configuration section should be used. If not defined, the default selection will be used.

-o {json,legacy,yaml}, **--output** {json,legacy,yaml}
: Output format (v4 only): json|legacy|yaml

-f <fields>, **--fields** <fields>
: Fields to display in the output (comma separated). Not used with legacy output

--server-url <server_url>
: GitLab server URL [env var: GITLAB_URL]

--ssl-verify <ssl_verify>
Path to a CA_BUNDLE file or directory with certificates of trusted CAs. [env var: GITLAB_SSL_VERIFY]

--no-ssl-verify
Disable SSL verification

--timeout <timeout>
Timeout to use for requests to the GitLab server. [env var: GITLAB_TIMEOUT]

--api-version <api_version>
GitLab API version [env var: GITLAB_API_VERSION]

--per-page <per_page>
Number of entries to return per page in the response. [env var: GITLAB_PER_PAGE]

--pagination <pagination>
Whether to use keyset or offset pagination [env var: GITLAB_PAGINATION]

--order-by <order_by>
Set order_by globally [env var: GITLAB_ORDER_BY]

--user-agent <user_agent>
The user agent to send to GitLab with the HTTP request. [env var: GITLAB_USER_AGENT]

--private-token <private_token>
GitLab private access token [env var: GITLAB_PRIVATE_TOKEN]

--oauth-token <oauth_token>
GitLab OAuth token [env var: GITLAB_OAUTH_TOKEN]

--job-token <job_token>
GitLab CI job token [env var: CI_JOB_TOKEN]

--skip-login
Skip initial authenticated API call to the current user endpoint. This may be useful when invoking the CLI in scripts. [env var: GITLAB_SKIP_LOGIN]

--no-mask-credentials
Don't mask credentials in debug mode

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 |
                               --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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]
                                          [--silent-mode-enabled SILENT_MODE_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_
```

(continues on next page)

(continued from previous page)

```
↪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>

--silent-mode-enabled <silent_mode_enabled>

```
--password-authentication-enabled-for-web <password_authentication_enabled_for_web>

--gravatar-enabled <gravatar_enabled>

--sign-in-text <sign_in_text>

--created-at <created_at>

--updated-at <updated_at>

--home-page-url <home_page_url>

--default-branch-protection <default_branch_protection>

--restricted-visibility-levels <restricted_visibility_levels>

--max-attachment-size <max_attachment_size>

--session-expire-delay <session_expire_delay>

--default-project-visibility <default_project_visibility>

--default-snippet-visibility <default_snippet_visibility>

--default-group-visibility <default_group_visibility>

--outbound-local-requests-whitelist <outbound_local_requests_whitelist>

--disabled-oauth-sign-in-sources <disabled_oauth_sign_in_sources>

--domain-whitelist <domain_whitelist>

--domain-blacklist-enabled <domain_blacklist_enabled>

--domain-blacklist <domain_blacklist>

--domain-allowlist <domain_allowlist>

--domain-denylist-enabled <domain_denylist_enabled>

--domain-denylist <domain_denylist>

--external-authorization-service-enabled <external_authorization_service_enabled>

--external-authorization-service-url <external_authorization_service_url>

--external-authorization-service-default-label␣
<external_authorization_service_default_label>

--external-authorization-service-timeout <external_authorization_service_timeout>

--import-sources <import_sources>

--user-oauth-applications <user_oauth_applications>

--after-sign-out-path <after_sign_out_path>

--container-registry-token-expire-delay <container_registry_token_expire_delay>

--repository-storages <repository_storages>
```

--plantuml-enabled <plantuml_enabled>

--plantuml-url <plantuml_url>

--terminal-max-session-time <terminal_max_session_time>

--polling-interval-multiplier <polling_interval_multiplier>

--rsa-key-restriction <rsa_key_restriction>

--dsa-key-restriction <dsa_key_restriction>

--ecdsa-key-restriction <ecdsa_key_restriction>

--ed25519-key-restriction <ed25519_key_restriction>

--first-day-of-week <first_day_of_week>

--enforce-terms <enforce_terms>

--terms <terms>

--performance-bar-allowed-group-id <performance_bar_allowed_group_id>

--instance-statistics-visibility-private <instance_statistics_visibility_private>

--user-show-add-ssh-key-message <user_show_add_ssh_key_message>

--file-template-project-id <file_template_project_id>

--local-markdown-version <local_markdown_version>

--asset-proxy-enabled <asset_proxy_enabled>

--asset-proxy-url <asset_proxy_url>

--asset-proxy-whitelist <asset_proxy_whitelist>

--asset-proxy-allowlist <asset_proxy_allowlist>

--geo-node-allowed-ips <geo_node_allowed_ips>

--allow-local-requests-from-hooks-and-services
<allow_local_requests_from_hooks_and_services>

--allow-local-requests-from-web-hooks-and-services
<allow_local_requests_from_web_hooks_and_services>

--allow-local-requests-from-system-hooks <allow_local_requests_from_system_hooks>

gitlab application-statistics

```
usage: gitlab application-statistics [-h] {get} ...
```

-h, --help
show this help message and exit

gitlab application-statistics get

```
usage: gitlab application-statistics get [-h] [--sudo SUDO]
```

-h, --help
show this help message and exit

--sudo <sudo>

gitlab audit-event

```
usage: gitlab audit-event [-h] {list,get} ...
```

-h, --help
show this help message and exit

gitlab audit-event get

```
usage: gitlab audit-event get [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab audit-event list

```
usage: gitlab audit-event list [-h] [--sudo SUDO]
                               [--created-after CREATED_AFTER]
                               [--created-before CREATED_BEFORE]
                               [--entity-type ENTITY_TYPE]
                               [--entity-id ENTITY_ID] [--page PAGE]
                               [--per-page PER_PAGE] [--get-all |
                               --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--created-after <created_after>

--created-before <created_before>

--entity-type <entity_type>

--entity-id <entity_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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 |
                                     --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab broadcast-message update

```
usage: gitlab broadcast-message update [-h] [--sudo SUDO] --id ID
                                       [--message MESSAGE]
                                       [--starts-at STARTS_AT]
                                       [--ends-at ENDS_AT] [--color COLOR]
                                       [--font FONT]
                                       [--target-access-levels TARGET_ACCESS_LEVELS]
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

--message <message>

--starts-at <starts_at>

--ends-at <ends_at>

--color <color>

--font <font>

--target-access-levels <target_access_levels>

gitlab bulk-import

```
usage: gitlab bulk-import [-h] {list,get,create} ...
```

-h, --help
show this help message and exit

gitlab bulk-import create

```
usage: gitlab bulk-import create [-h] [--sudo SUDO]
                                 --configuration CONFIGURATION
                                 --entities ENTITIES
```

-h, --help
show this help message and exit

--sudo <sudo>

--configuration <configuration>

--entities <entities>

gitlab bulk-import get

```
usage: gitlab bulk-import get [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab bulk-import list

```
usage: gitlab bulk-import list [-h] [--sudo SUDO] [--sort SORT]
                               [--status STATUS] [--page PAGE]
                               [--per-page PER_PAGE] [--get-all |
                               --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--sort <sort>

--status <status>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab bulk-import-all-entity

```
usage: gitlab bulk-import-all-entity [-h] {list} ...
```

-h, --help
show this help message and exit

gitlab bulk-import-all-entity list

```
usage: gitlab bulk-import-all-entity list [-h] [--sudo SUDO] [--sort SORT]
                                          [--status STATUS] [--page PAGE]
                                          [--per-page PER_PAGE] [--get-all |
                                          --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--sort <sort>

--status <status>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab bulk-import-entity

```
usage: gitlab bulk-import-entity [-h] {list,get} ...
```

-h, --help
show this help message and exit

gitlab bulk-import-entity get

```
usage: gitlab bulk-import-entity get [-h] [--sudo SUDO]
                                     --bulk-import-id BULK_IMPORT_ID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--bulk-import-id <bulk_import_id>

--id <id>

gitlab bulk-import-entity list

```
usage: gitlab bulk-import-entity list [-h] [--sudo SUDO]
                                      --bulk-import-id BULK_IMPORT_ID
                                      [--sort SORT] [--status STATUS]
                                      [--page PAGE] [--per-page PER_PAGE]
                                      [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--bulk-import-id <bulk_import_id>

--sort <sort>

--status <status>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                      --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                        --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                    --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab current-user-runner

```
usage: gitlab current-user-runner [-h] {create} ...
```

-h, --help
show this help message and exit

gitlab current-user-runner create

```
usage: gitlab current-user-runner create [-h] [--sudo SUDO]
                                         --runner-type RUNNER_TYPE
                                         [--group-id GROUP_ID]
                                         [--project-id PROJECT_ID]
                                         [--description DESCRIPTION]
                                         [--paused PAUSED] [--locked LOCKED]
                                         [--run-untagged RUN_UNTAGGED]
                                         [--tag-list TAG_LIST]
                                         [--access-level ACCESS_LEVEL]
                                         [--maximum-timeout MAXIMUM_TIMEOUT]
                                         [--maintenance-note MAINTENANCE_NOTE]
```

-h, --help
show this help message and exit

--sudo <sudo>

--runner-type <runner_type>

--group-id <group_id>

--project-id <project_id>

--description <description>

--paused <paused>

--locked <locked>

--run-untagged <run_untagged>

--tag-list <tag_list>

--access-level <access_level>

--maximum-timeout <maximum_timeout>

--maintenance-note <maintenance_note>

gitlab current-user-status

```
usage: gitlab current-user-status [-h] {get,update} ...
```

-h, --help
show this help message and exit

gitlab current-user-status get

```
usage: gitlab current-user-status get [-h] [--sudo SUDO]
```

-h, --help
show this help message and exit

--sudo <sudo>

gitlab current-user-status update

```
usage: gitlab current-user-status update [-h] [--sudo SUDO] [--emoji EMOJI]
                                         [--message MESSAGE]
```

-h, --help
show this help message and exit

--sudo <sudo>

--emoji <emoji>

--message <message>

gitlab deploy-key

```
usage: gitlab deploy-key [-h] {list,create} ...
```

-h, --help
show this help message and exit

gitlab deploy-key create

```
usage: gitlab deploy-key create [-h] [--sudo SUDO] --title TITLE --key KEY
                                [--expires-at EXPIRES_AT]
```

-h, --help
show this help message and exit

--sudo <sudo>

--title <title>

--key <key>

--expires-at <expires_at>

gitlab deploy-key list

```
usage: gitlab deploy-key list [-h] [--sudo SUDO] [--page PAGE]
                              [--per-page PER_PAGE] [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab event

```
usage: gitlab event [-h] {list} ...
```

-h, --help
show this help message and exit

gitlab event list

```
usage: gitlab event list [-h] [--sudo SUDO] [--action ACTION]
                         [--target-type TARGET_TYPE] [--before BEFORE]
                         [--after AFTER] [--sort SORT] [--scope SCOPE]
                         [--page PAGE] [--per-page PER_PAGE] [--get-all |
                         --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--action <action>

--target-type <target_type>

--before <before>

--after <after>

--sort <sort>

--scope <scope>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab gitlabciyml

```
usage: gitlab gitlabciyml [-h] {list,get} ...
```

-h, --help
show this help message and exit

gitlab gitlabciyml get

```
usage: gitlab gitlabciyml get [-h] [--sudo SUDO] --name NAME
```

-h, --help
show this help message and exit

--sudo <sudo>

--name <name>

gitlab gitlabciyml list

```
usage: gitlab gitlabciyml list [-h] [--sudo SUDO] [--page PAGE]
                               [--per-page PER_PAGE] [--get-all |
                               --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab group

```
usage: gitlab group [-h]
                    {list,get,create,update,delete,transfer-project,transfer,search,ldap-
↪sync,share,unshare,restore} ...
```

-h, --help
show this help message and exit

gitlab group create

```
usage: gitlab group create [-h] [--sudo SUDO] --name NAME --path PATH
                           [--description DESCRIPTION]
                           [--membership-lock MEMBERSHIP_LOCK]
                           [--visibility VISIBILITY]
                           [--share-with-group-lock SHARE_WITH_GROUP_LOCK]
                           [--require-two-factor-authentication REQUIRE_TWO_FACTOR_
↪AUTHENTICATION]
                           [--two-factor-grace-period TWO_FACTOR_GRACE_PERIOD]
                           [--project-creation-level PROJECT_CREATION_LEVEL]
                           [--auto-devops-enabled AUTO_DEVOPS_ENABLED]
                           [--subgroup-creation-level SUBGROUP_CREATION_LEVEL]
                           [--emails-disabled EMAILS_DISABLED]
                           [--avatar AVATAR]
                           [--mentions-disabled MENTIONS_DISABLED]
                           [--lfs-enabled LFS_ENABLED]
                           [--request-access-enabled REQUEST_ACCESS_ENABLED]
                           [--parent-id PARENT_ID]
                           [--default-branch-protection DEFAULT_BRANCH_PROTECTION]
                           [--shared-runners-minutes-limit SHARED_RUNNERS_MINUTES_LIMIT]
                           [--extra-shared-runners-minutes-limit EXTRA_SHARED_RUNNERS_
↪MINUTES_LIMIT]
```

-h, --help
show this help message and exit

--sudo <sudo>

--name <name>

--path <path>

--description <description>

--membership-lock <membership_lock>

--visibility <visibility>

--share-with-group-lock <share_with_group_lock>

--require-two-factor-authentication <require_two_factor_authentication>

--two-factor-grace-period <two_factor_grace_period>

--project-creation-level <project_creation_level>

--auto-devops-enabled <auto_devops_enabled>

--subgroup-creation-level <subgroup_creation_level>

--emails-disabled <emails_disabled>

--avatar <avatar>

--mentions-disabled <mentions_disabled>

--lfs-enabled <lfs_enabled>

--request-access-enabled <request_access_enabled>

--parent-id <parent_id>

--default-branch-protection <default_branch_protection>

--shared-runners-minutes-limit <shared_runners_minutes_limit>

--extra-shared-runners-minutes-limit <extra_shared_runners_minutes_limit>

gitlab group delete

```
usage: gitlab group delete [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab group get

```
usage: gitlab group get [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab group ldap-sync

```
usage: gitlab group ldap-sync [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab group list

```
usage: gitlab group list [-h] [--sudo SUDO] [--skip-groups SKIP_GROUPS]
                         [--all-available ALL_AVAILABLE] [--search SEARCH]
                         [--order-by ORDER_BY] [--sort SORT]
                         [--statistics STATISTICS] [--owned OWNED]
                         [--with-custom-attributes WITH_CUSTOM_ATTRIBUTES]
                         [--min-access-level MIN_ACCESS_LEVEL]
                         [--top-level-only TOP_LEVEL_ONLY] [--page PAGE]
                         [--per-page PER_PAGE] [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--skip-groups <skip_groups>

--all-available <all_available>

--search <search>

--order-by <order_by>

--sort <sort>

--statistics <statistics>

--owned <owned>

--with-custom-attributes <with_custom_attributes>

--min-access-level <min_access_level>

--top-level-only <top_level_only>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab group restore

```
usage: gitlab group restore [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab group search

```
usage: gitlab group search [-h] --id ID --scope SCOPE --search SEARCH
```

-h, --help
show this help message and exit

--id <id>

--scope <scope>

--search <search>

gitlab group share

```
usage: gitlab group share [-h] --id ID --group-id GROUP_ID
                          --group-access GROUP_ACCESS
                          [--expires-at EXPIRES_AT]
```

-h, --help
show this help message and exit

--id <id>

--group-id <group_id>

--group-access <group_access>

--expires-at <expires_at>

gitlab group transfer

```
usage: gitlab group transfer [-h] --id ID [--group-id GROUP_ID]
```

-h, --help
show this help message and exit

--id <id>

--group-id <group_id>

gitlab group transfer-project

```
usage: gitlab group transfer-project [-h] --id ID --project-id PROJECT_ID
```

-h, --help
show this help message and exit

--id <id>

--project-id <project_id>

gitlab group unshare

```
usage: gitlab group unshare [-h] --id ID --group-id GROUP_ID
```

-h, --help
show this help message and exit

--id <id>

--group-id <group_id>

gitlab group update

```
usage: gitlab group update [-h] [--sudo SUDO] --id ID [--name NAME]
                           [--path PATH] [--description DESCRIPTION]
                           [--membership-lock MEMBERSHIP_LOCK]
                           [--share-with-group-lock SHARE_WITH_GROUP_LOCK]
                           [--visibility VISIBILITY]
                           [--require-two-factor-authentication REQUIRE_TWO_FACTOR_
↪AUTHENTICATION]
                           [--two-factor-grace-period TWO_FACTOR_GRACE_PERIOD]
                           [--project-creation-level PROJECT_CREATION_LEVEL]
                           [--auto-devops-enabled AUTO_DEVOPS_ENABLED]
                           [--subgroup-creation-level SUBGROUP_CREATION_LEVEL]
                           [--emails-disabled EMAILS_DISABLED]
                           [--avatar AVATAR]
                           [--mentions-disabled MENTIONS_DISABLED]
                           [--lfs-enabled LFS_ENABLED]
                           [--request-access-enabled REQUEST_ACCESS_ENABLED]
                           [--default-branch-protection DEFAULT_BRANCH_PROTECTION]
                           [--file-template-project-id FILE_TEMPLATE_PROJECT_ID]
                           [--shared-runners-minutes-limit SHARED_RUNNERS_MINUTES_LIMIT]
                           [--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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab group-access-token

```
usage: gitlab group-access-token [-h] {list,get,create,delete,rotate} ...
```

-h, --help
show this help message and exit

gitlab group-access-token create

```
usage: gitlab group-access-token create [-h] [--sudo SUDO] --group-id GROUP_ID
                                        --name NAME --scopes SCOPES
                                        [--access-level ACCESS_LEVEL]
                                        [--expires-at EXPIRES_AT]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--name <name>

--scopes <scopes>

--access-level <access_level>

--expires-at <expires_at>

gitlab group-access-token delete

```
usage: gitlab group-access-token delete [-h] [--sudo SUDO] --group-id GROUP_ID
                                        --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--id <id>

gitlab group-access-token get

```
usage: gitlab group-access-token get [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--id <id>

gitlab group-access-token list

```
usage: gitlab group-access-token list [-h] [--sudo SUDO] --group-id GROUP_ID
                                      [--page PAGE] [--per-page PER_PAGE]
                                      [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

gitlab group-access-token rotate

```
usage: gitlab group-access-token rotate [-h] --group-id GROUP_ID [--sudo SUDO]
                                        --id ID [--expires-at EXPIRES_AT]
```

-h, --help

show this help message and exit

--group-id <group_id>

--sudo <sudo>

--id <id>

--expires-at <expires_at>

gitlab group-approval-rule

```
usage: gitlab group-approval-rule [-h] {list,get,create,update} ...
```

-h, --help

show this help message and exit

gitlab group-approval-rule create

```
usage: gitlab group-approval-rule create [-h] [--sudo SUDO]
                                         --group-id GROUP_ID --name NAME
                                         --approvals-required APPROVALS_REQUIRED
                                         [--user-ids USER_IDS]
                                         [--group-ids GROUP_IDS]
                                         [--rule-type RULE_TYPE]
```

-h, --help

show this help message and exit

--sudo <sudo>

--group-id <group_id>

--name <name>

--approvals-required <approvals_required>

--user-ids <user_ids>

--group-ids <group_ids>

--rule-type <rule_type>

gitlab group-approval-rule get

```
usage: gitlab group-approval-rule 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-approval-rule list

```
usage: gitlab group-approval-rule list [-h] [--sudo SUDO] --group-id GROUP_ID
                                       [--page PAGE] [--per-page PER_PAGE]
                                       [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab group-approval-rule update

```
usage: gitlab group-approval-rule 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-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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--created-after <created_after>

--created-before <created_before>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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 |
                                         --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--search <search>

--sort <sort>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--user-id <user_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab group-board update

```
usage: gitlab group-board update [-h] [--sudo SUDO] --group-id GROUP_ID
                                 --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--id <id>

gitlab group-board-list

```
usage: gitlab group-board-list [-h] {list,get,create,update,delete} ...
```

-h, --help
show this help message and exit

gitlab group-board-list create

```
usage: gitlab group-board-list create [-h] [--sudo SUDO] --group-id GROUP_ID
                                      --board-id BOARD_ID
                                      [--label-id LABEL_ID |
                                      --assignee-id ASSIGNEE_ID |
                                      --milestone-id MILESTONE_ID |
                                      --iteration-id ITERATION_ID]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--board-id <board_id>

--label-id <label_id>

--assignee-id <assignee_id>

--milestone-id <milestone_id>

--iteration-id <iteration_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 |
                                    --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--board-id <board_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                          --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                      --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--scopes <scopes>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--skip-groups <skip_groups>

--all-available <all_available>

--search <search>

--order-by <order_by>

--sort <sort>

--statistics <statistics>

--owned <owned>

--with-custom-attributes <with_custom_attributes>

--min-access-level <min_access_level>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--author-id <author_id>

--labels <labels>

--order-by <order_by>

--sort <sort>

--search <search>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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 |
                                          --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-iid <epic_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab group-epic-discussion-note

```
usage: gitlab group-epic-discussion-note [-h] {get,create,update,delete} ...
```

-h, --help
show this help message and exit

gitlab group-epic-discussion-note create

```
usage: gitlab group-epic-discussion-note create [-h] [--sudo SUDO]
                                                --group-id GROUP_ID
                                                --epic-id EPIC_ID
                                                --discussion-id DISCUSSION_ID
                                                --body BODY
                                                [--created-at CREATED_AT]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-id <epic_id>

--discussion-id <discussion_id>

--body <body>

--created-at <created_at>

gitlab group-epic-discussion-note delete

```
usage: gitlab group-epic-discussion-note delete [-h] [--sudo SUDO]
                                                --group-id GROUP_ID
                                                --epic-id EPIC_ID
                                                --discussion-id DISCUSSION_ID
                                                --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-id <epic_id>

--discussion-id <discussion_id>

--id <id>

gitlab group-epic-discussion-note get

```
usage: gitlab group-epic-discussion-note get [-h] [--sudo SUDO]
                                             --group-id GROUP_ID
                                             --epic-id EPIC_ID
                                             --discussion-id DISCUSSION_ID
                                             --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-id <epic_id>

--discussion-id <discussion_id>

--id <id>

gitlab group-epic-discussion-note update

```
usage: gitlab group-epic-discussion-note update [-h] [--sudo SUDO]
                                                --group-id GROUP_ID
                                                --epic-id EPIC_ID
                                                --discussion-id DISCUSSION_ID
                                                --id ID --body BODY
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-id <epic_id>

--discussion-id <discussion_id>

--id <id>

--body <body>

gitlab group-epic-issue

```
usage: gitlab group-epic-issue [-h] {list,create,update,delete} ...
```

-h, --help
show this help message and exit

gitlab group-epic-issue create

```
usage: gitlab group-epic-issue create [-h] [--sudo SUDO] --group-id GROUP_ID
                                      --epic-iid EPIC_IID --issue-id ISSUE_ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-iid <epic_iid>

--issue-id <issue_id>

gitlab group-epic-issue delete

```
usage: gitlab group-epic-issue delete [-h] [--sudo SUDO] --group-id GROUP_ID
                                      --epic-iid EPIC_IID
                                      --epic-issue-id EPIC_ISSUE_ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-iid <epic_iid>

--epic-issue-id <epic_issue_id>

gitlab group-epic-issue list

```
usage: gitlab group-epic-issue list [-h] [--sudo SUDO] --group-id GROUP_ID
                                    --epic-iid EPIC_IID [--page PAGE]
                                    [--per-page PER_PAGE] [--get-all |
                                    --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-iid <epic_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab group-epic-issue update

```
usage: gitlab group-epic-issue update [-h] [--sudo SUDO] --group-id GROUP_ID
                                      --epic-iid EPIC_IID
                                      --epic-issue-id EPIC_ISSUE_ID
                                      [--move-before-id MOVE_BEFORE_ID]
                                      [--move-after-id MOVE_AFTER_ID]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-iid <epic_iid>

--epic-issue-id <epic_issue_id>

--move-before-id <move_before_id>

--move-after-id <move_after_id>

gitlab group-epic-note

```
usage: gitlab group-epic-note [-h] {list,get,create,update,delete} ...
```

-h, --help
show this help message and exit

gitlab group-epic-note create

```
usage: gitlab group-epic-note create [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --epic-id EPIC_ID --body BODY
                                     [--created-at CREATED_AT]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-id <epic_id>

--body <body>

--created-at <created_at>

gitlab group-epic-note delete

```
usage: gitlab group-epic-note delete [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --epic-id EPIC_ID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-id <epic_id>

--id <id>

gitlab group-epic-note get

```
usage: gitlab group-epic-note get [-h] [--sudo SUDO] --group-id GROUP_ID
                                  --epic-id EPIC_ID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-id <epic_id>

--id <id>

gitlab group-epic-note list

```
usage: gitlab group-epic-note list [-h] [--sudo SUDO] --group-id GROUP_ID
                                   --epic-id EPIC_ID [--page PAGE]
                                   [--per-page PER_PAGE] [--get-all |
                                   --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-id <epic_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab group-epic-note update

```
usage: gitlab group-epic-note update [-h] [--sudo SUDO] --group-id GROUP_ID
                                     --epic-id EPIC_ID --id ID --body BODY
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-id <epic_id>

--id <id>

--body <body>

gitlab group-epic-note-award-emoji

```
usage: gitlab group-epic-note-award-emoji [-h] {list,get,create,delete} ...
```

-h, --help
show this help message and exit

gitlab group-epic-note-award-emoji create

```
usage: gitlab group-epic-note-award-emoji create [-h] [--sudo SUDO]
                                                 --group-id GROUP_ID
                                                 --epic-iid EPIC_IID
                                                 --note-id NOTE_ID --name NAME
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-iid <epic_iid>

--note-id <note_id>

--name <name>

gitlab group-epic-note-award-emoji delete

```
usage: gitlab group-epic-note-award-emoji delete [-h] [--sudo SUDO]
                                                 --group-id GROUP_ID
                                                 --epic-iid EPIC_IID
                                                 --note-id NOTE_ID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-iid <epic_iid>

--note-id <note_id>

--id <id>

gitlab group-epic-note-award-emoji get

```
usage: gitlab group-epic-note-award-emoji get [-h] [--sudo SUDO]
                                              --group-id GROUP_ID
                                              --epic-iid EPIC_IID
                                              --note-id NOTE_ID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-iid <epic_iid>

--note-id <note_id>

--id <id>

gitlab group-epic-note-award-emoji list

```
usage: gitlab group-epic-note-award-emoji list [-h] [--sudo SUDO]
                                               --group-id GROUP_ID
                                               --epic-iid EPIC_IID
                                               --note-id NOTE_ID [--page PAGE]
                                               [--per-page PER_PAGE]
                                               [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-iid <epic_iid>

--note-id <note_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--epic-id <epic_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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]
```

(continues on next page)

(continued from previous page)

```
                                [--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 |
                              --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab group-hook update

```
usage: gitlab group-hook update [-h] [--sudo SUDO] --group-id GROUP_ID --id ID
                                --url URL [--push-events PUSH_EVENTS]
                                [--issues-events ISSUES_EVENTS]
                                [--confidential-issues-events CONFIDENTIAL_ISSUES_EVENTS]
                                [--merge-requests-events MERGE_REQUESTS_EVENTS]
                                [--tag-push-events TAG_PUSH_EVENTS]
                                [--note-events NOTE_EVENTS]
                                [--confidential-note-events CONFIDENTIAL_NOTE_EVENTS]
                                [--job-events JOB_EVENTS]
                                [--pipeline-events PIPELINE_EVENTS]
                                [--wiki-page-events WIKI_PAGE_EVENTS]
                                [--deployment-events DEPLOYMENT_EVENTS]
                                [--releases-events RELEASES_EVENTS]
                                [--subgroup-events SUBGROUP_EVENTS]
                                [--enable-ssl-verification ENABLE_SSL_VERIFICATION]
                                [--token TOKEN]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--id <id>

--url <url>

--push-events <push_events>

--issues-events <issues_events>

--confidential-issues-events <confidential_issues_events>

--merge-requests-events <merge_requests_events>

--tag-push-events <tag_push_events>

--note-events <note_events>

--confidential-note-events <confidential_note_events>

--job-events <job_events>

--pipeline-events <pipeline_events>

--wiki-page-events <wiki_page_events>

--deployment-events <deployment_events>

--releases-events <releases_events>

--subgroup-events <subgroup_events>

--enable-ssl-verification <enable_ssl_verification>

--token <token>

gitlab group-import

```
usage: gitlab group-import [-h] {get} ...
```

-h, --help
show this help message and exit

gitlab group-import get

```
usage: gitlab group-import get [-h] [--sudo SUDO] --group-id GROUP_ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

gitlab group-invitation

```
usage: gitlab group-invitation [-h] {list,get,create,update,delete} ...
```

-h, --help
show this help message and exit

gitlab group-invitation create

```
usage: gitlab group-invitation create [-h] [--sudo SUDO] --group-id GROUP_ID
                                      --access-level ACCESS_LEVEL
                                      [--expires-at EXPIRES_AT]
                                      [--invite-source INVITE_SOURCE]
                                      [--tasks-to-be-done TASKS_TO_BE_DONE]
                                      [--tasks-project-id TASKS_PROJECT_ID]
                                      [--email EMAIL | --user-id USER_ID]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--access-level <access_level>

--expires-at <expires_at>

--invite-source <invite_source>

--tasks-to-be-done <tasks_to_be_done>

--tasks-project-id <tasks_project_id>

--email <email>

--user-id <user_id>

gitlab group-invitation delete

```
usage: gitlab group-invitation delete [-h] [--sudo SUDO] --group-id GROUP_ID
                                      --email EMAIL
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--email <email>

gitlab group-invitation get

```
usage: gitlab group-invitation get [-h] [--sudo SUDO] --group-id GROUP_ID
                                   --email EMAIL
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--email <email>

gitlab group-invitation list

```
usage: gitlab group-invitation list [-h] [--sudo SUDO] --group-id GROUP_ID
                                    [--query QUERY] [--page PAGE]
                                    [--per-page PER_PAGE] [--get-all |
                                    --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--query <query>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab group-invitation update

```
usage: gitlab group-invitation update [-h] [--sudo SUDO] --group-id GROUP_ID
                                      --email EMAIL
                                      [--access-level ACCESS_LEVEL]
                                      [--expires-at EXPIRES_AT]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--email <email>

--access-level <access_level>

--expires-at <expires_at>

gitlab group-issue

```
usage: gitlab group-issue [-h] {list} ...
```

-h, --help
show this help message and exit

gitlab group-issue list

```
usage: gitlab group-issue list [-h] [--sudo SUDO] --group-id GROUP_ID
                               [--state STATE] [--labels LABELS]
                               [--milestone MILESTONE] [--order-by ORDER_BY]
                               [--sort SORT] [--iids IIDS]
                               [--author-id AUTHOR_ID]
                               [--iteration-id ITERATION_ID]
                               [--assignee-id ASSIGNEE_ID]
                               [--my-reaction-emoji MY_REACTION_EMOJI]
                               [--search SEARCH]
                               [--created-after CREATED_AFTER]
                               [--created-before CREATED_BEFORE]
                               [--updated-after UPDATED_AFTER]
                               [--updated-before UPDATED_BEFORE] [--page PAGE]
                               [--per-page PER_PAGE] [--get-all |
                               --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--state <state>

--labels <labels>

--milestone <milestone>

--order-by <order_by>

--sort <sort>

--iids <iids>

--author-id <author_id>

--iteration-id <iteration_id>

--assignee-id <assignee_id>

--my-reaction-emoji <my_reaction_emoji>

--search <search>

--created-after <created_after>

--created-before <created_before>

--updated-after <updated_after>

--updated-before <updated_before>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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
                                   [--include-ancestors INCLUDE_ANCESTORS]
                                   [--include-descendants INCLUDE_DESCENDANTS]
                                   [--in IN] [--search SEARCH] [--state STATE]
                                   [--updated-after UPDATED_AFTER]
                                   [--updated-before UPDATED_BEFORE]
                                   [--page PAGE] [--per-page PER_PAGE]
                                   [--get-all | --no-get-all]
```

-h, --help

show this help message and exit

--sudo <sudo>

--group-id <group_id>

--include-ancestors <include_ancestors>

--include-descendants <include_descendants>

--in <in>

--search <search>

--state <state>

--updated-after <updated_after>

--updated-before <updated_before>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab group-label subscribe

```
usage: gitlab group-label subscribe [-h] --group-id GROUP_ID [--sudo SUDO]
                                    --name NAME
```

-h, --help
show this help message and exit

--group-id <group_id>

--sudo <sudo>

--name <name>

gitlab group-label unsubscribe

```
usage: gitlab group-label unsubscribe [-h] --group-id GROUP_ID [--sudo SUDO]
                                      --name NAME
```

-h, --help
show this help message and exit

--group-id <group_id>

--sudo <sudo>

--name <name>

gitlab group-label update

```
usage: gitlab group-label update [-h] [--sudo SUDO] --group-id GROUP_ID
                                 --name NAME [--new-name NEW_NAME]
                                 [--color COLOR] [--description DESCRIPTION]
                                 [--priority PRIORITY]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--name <name>

--new-name <new_name>

--color <color>

--description <description>

--priority <priority>

gitlab group-ldap-group-link

```
usage: gitlab group-ldap-group-link [-h] {list,create,delete} ...
```

-h, --help
show this help message and exit

gitlab group-ldap-group-link create

```
usage: gitlab group-ldap-group-link create [-h] [--sudo SUDO]
                                           --group-id GROUP_ID
                                           --provider PROVIDER
                                           --group-access GROUP_ACCESS
                                           [--cn CN | --filter FILTER]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--provider <provider>

--group-access <group_access>

--cn <cn>

--filter <filter>

gitlab group-ldap-group-link delete

```
usage: gitlab group-ldap-group-link delete [-h] [--sudo SUDO]
                                           --group-id GROUP_ID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--id <id>

gitlab group-ldap-group-link list

```
usage: gitlab group-ldap-group-link list [-h] [--sudo SUDO]
                                         --group-id GROUP_ID [--page PAGE]
                                         [--per-page PER_PAGE] [--get-all |
                                         --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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
                                  [--expires-at EXPIRES_AT]
                                  [--tasks-to-be-done TASKS_TO_BE_DONE]
                                  [--username USERNAME | --user-id USER_ID]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--access-level <access_level>

--expires-at <expires_at>

--tasks-to-be-done <tasks_to_be_done>

--username <username>

--user-id <user_id>

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab group-member-role

```
usage: gitlab group-member-role [-h] {list,create,delete} ...
```

-h, --help
show this help message and exit

gitlab group-member-role create

```
usage: gitlab group-member-role create [-h] [--sudo SUDO] --group-id GROUP_ID
                                       --name NAME
                                       --base-access-level BASE_ACCESS_LEVEL
                                       [--description DESCRIPTION]
                                       [--admin-cicd-variables ADMIN_CICD_VARIABLES]
                                       [--admin-compliance-framework ADMIN_COMPLIANCE_
↪FRAMEWORK]
                                       [--admin-group-member ADMIN_GROUP_MEMBER]
                                       [--admin-merge-request ADMIN_MERGE_REQUEST]
```

(continues on next page)

(continued from previous page)

```
                                    [--admin-push-rules ADMIN_PUSH_RULES]
                                    [--admin-terraform-state ADMIN_TERRAFORM_STATE]
                                    [--admin-vulnerability ADMIN_VULNERABILITY]
                                    [--admin-web-hook ADMIN_WEB_HOOK]
                                    [--archive-project ARCHIVE_PROJECT]
                                    [--manage-deploy-tokens MANAGE_DEPLOY_TOKENS]
                                    [--manage-group-access-tokens MANAGE_GROUP_ACCESS_
↪TOKENS]
                                    [--manage-merge-request-settings MANAGE_MERGE_
↪REQUEST_SETTINGS]
                                    [--manage-project-access-tokens MANAGE_PROJECT_
↪ACCESS_TOKENS]
                                    [--manage-security-policy-link MANAGE_SECURITY_
↪POLICY_LINK]
                                    [--read-code READ_CODE]
                                    [--read-runners READ_RUNNERS]
                                    [--read-dependency READ_DEPENDENCY]
                                    [--read-vulnerability READ_VULNERABILITY]
                                    [--remove-group REMOVE_GROUP]
                                    [--remove-project REMOVE_PROJECT]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--name <name>

--base-access-level <base_access_level>

--description <description>

--admin-cicd-variables <admin_cicd_variables>

--admin-compliance-framework <admin_compliance_framework>

--admin-group-member <admin_group_member>

--admin-merge-request <admin_merge_request>

--admin-push-rules <admin_push_rules>

--admin-terraform-state <admin_terraform_state>

--admin-vulnerability <admin_vulnerability>

--admin-web-hook <admin_web_hook>

--archive-project <archive_project>

--manage-deploy-tokens <manage_deploy_tokens>

--manage-group-access-tokens <manage_group_access_tokens>

--manage-merge-request-settings <manage_merge_request_settings>

--manage-project-access-tokens <manage_project_access_tokens>

--manage-security-policy-link <manage_security_policy_link>

--read-code <read_code>

--read-runners <read_runners>

--read-dependency <read_dependency>

--read-vulnerability <read_vulnerability>

--remove-group <remove_group>

--remove-project <remove_project>

gitlab group-member-role delete

```
usage: gitlab group-member-role 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-role list

```
usage: gitlab group-member-role list [-h] [--sudo SUDO] --group-id GROUP_ID
                                     [--page PAGE] [--per-page PER_PAGE]
                                     [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help

show this help message and exit

--sudo <sudo>

--group-id <group_id>

--state <state>

--order-by <order_by>

--sort <sort>

--milestone <milestone>

--view <view>

--labels <labels>

--created-after <created_after>

--created-before <created_before>

--updated-after <updated_after>

--updated-before <updated_before>

--scope <scope>

--author-id <author_id>

--assignee-id <assignee_id>

--approver-ids <approver_ids>

--approved-by-ids <approved_by_ids>

--my-reaction-emoji <my_reaction_emoji>

--source-branch <source_branch>

--target-branch <target_branch>

--search <search>

--wip <wip>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                   --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--iids <iids>

--state <state>

--search <search>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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]
                                 [--package-version PACKAGE_VERSION]
                                 [--include-versionless INCLUDE_VERSIONLESS]
                                 [--status STATUS] [--page PAGE]
                                 [--per-page PER_PAGE] [--get-all |
                                 --no-get-all]
```

-h, **--help**
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--exclude-subgroups <exclude_subgroups>

--order-by <order_by>

--sort <sort>

--package-type <package_type>

--package-name <package_name>

--package-version <package_version>

--include-versionless <include_versionless>

--status <status>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--archived <archived>

--visibility <visibility>

--order-by <order_by>

--sort <sort>

--search <search>

--simple <simple>

--owned <owned>

--starred <starred>

--with-custom-attributes <with_custom_attributes>

--include-subgroups <include_subgroups>

--with-issues-enabled <with_issues_enabled>

--with-merge-requests-enabled <with_merge_requests_enabled>

--with-shared <with_shared>

--min-access-level <min_access_level>

--with-security-reports <with_security_reports>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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]
```

(continues on next page)

(continued from previous page)

```
                                    [--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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--scope <scope>

--tag-list <tag_list>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab group-saml-group-link

```
usage: gitlab group-saml-group-link [-h] {list,get,create,delete} ...
```

-h, --help
show this help message and exit

gitlab group-saml-group-link create

```
usage: gitlab group-saml-group-link create [-h] [--sudo SUDO]
                                           --group-id GROUP_ID
                                           --saml-group-name SAML_GROUP_NAME
                                           --access-level ACCESS_LEVEL
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--saml-group-name <saml_group_name>

--access-level <access_level>

gitlab group-saml-group-link delete

```
usage: gitlab group-saml-group-link delete [-h] [--sudo SUDO]
                                           --group-id GROUP_ID --name NAME
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--name <name>

gitlab group-saml-group-link get

```
usage: gitlab group-saml-group-link get [-h] [--sudo SUDO] --group-id GROUP_ID
                                        --name NAME
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--name <name>

gitlab group-saml-group-link list

```
usage: gitlab group-saml-group-link list [-h] [--sudo SUDO]
                                         --group-id GROUP_ID [--page PAGE]
                                         [--per-page PER_PAGE] [--get-all |
                                         --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab group-service-account

```
usage: gitlab group-service-account [-h] {list,create,update,delete} ...
```

-h, --help
show this help message and exit

gitlab group-service-account create

```
usage: gitlab group-service-account create [-h] [--sudo SUDO]
                                           --group-id GROUP_ID [--name NAME]
                                           [--username USERNAME]
                                           [--email EMAIL]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--name <name>

--username <username>

--email <email>

gitlab group-service-account delete

```
usage: gitlab group-service-account 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-service-account list

```
usage: gitlab group-service-account list [-h] [--sudo SUDO]
                                         --group-id GROUP_ID
                                         [--order-by ORDER_BY] [--sort SORT]
                                         [--page PAGE] [--per-page PER_PAGE]
                                         [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--order-by <order_by>

--sort <sort>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab group-service-account update

```
usage: gitlab group-service-account update [-h] [--sudo SUDO]
                                           --group-id GROUP_ID --id ID
                                           [--name NAME] [--username USERNAME]
                                           [--email EMAIL]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--id <id>

--name <name>

--username <username>

--email <email>

gitlab group-service-account-access-token

```
usage: gitlab group-service-account-access-token [-h]
                                                 {list,create,delete,rotate} ...
```

-h, --help
show this help message and exit

gitlab group-service-account-access-token create

```
usage: gitlab group-service-account-access-token create [-h] [--sudo SUDO]
                                                        --group-id GROUP_ID
                                                        --user-id USER_ID
                                                        --name NAME
                                                        --scopes SCOPES
                                                        [--description DESCRIPTION]
                                                        [--expires-at EXPIRES_AT]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--user-id <user_id>

--name <name>

--scopes <scopes>

--description <description>

--expires-at <expires_at>

gitlab group-service-account-access-token delete

```
usage: gitlab group-service-account-access-token delete [-h] [--sudo SUDO]
                                                        --group-id GROUP_ID
                                                        --user-id USER_ID
                                                        --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--user-id <user_id>

--id <id>

gitlab group-service-account-access-token list

```
usage: gitlab group-service-account-access-token list [-h] [--sudo SUDO]
                                                      --group-id GROUP_ID
                                                      --user-id USER_ID
                                                      [--created-after CREATED_AFTER]
                                                      [--created-before CREATED_BEFORE]
                                                      [--expires-after EXPIRES_AFTER]
                                                      [--expires-before EXPIRES_BEFORE]
                                                      [--last-used-after LAST_USED_AFTER]
                                                      [--last-used-before LAST_USED_
↪BEFORE]
                                                      [--revoked REVOKED]
                                                      [--search SEARCH]
                                                      [--sort SORT]
                                                      [--state STATE]
                                                      [--page PAGE]
                                                      [--per-page PER_PAGE]
                                                      [--get-all |
                                                      --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--user-id <user_id>

--created-after <created_after>

--created-before <created_before>

--expires-after <expires_after>

--expires-before <expires_before>

--last-used-after <last_used_after>

--last-used-before <last_used_before>

--revoked <revoked>

--search <search>

--sort <sort>

--state <state>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab group-service-account-access-token rotate

```
usage: gitlab group-service-account-access-token rotate [-h]
                                                        --group-id GROUP_ID
                                                        --user-id USER_ID
                                                        [--sudo SUDO] --id ID
                                                        [--expires-at EXPIRES_AT]
```

-h, --help
show this help message and exit

--group-id <group_id>

--user-id <user_id>

--sudo <sudo>

--id <id>

--expires-at <expires_at>

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]
```

(continues on next page)

(continued from previous page)

```
                                    [--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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--skip-groups <skip_groups>

--all-available <all_available>

--search <search>

--order-by <order_by>

--sort <sort>

--statistics <statistics>

--owned <owned>

--with-custom-attributes <with_custom_attributes>

--min-access-level <min_access_level>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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
                                    [--description DESCRIPTION]
                                    [--environment-scope ENVIRONMENT_SCOPE]
```

(continues on next page)

(continued from previous page)

```
                                    [--masked MASKED]
                                    [--masked-and-hidden MASKED_AND_HIDDEN]
                                    [--protected PROTECTED] [--raw RAW]
                                    [--variable-type VARIABLE_TYPE]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--key <key>

--value <value>

--description <description>

--environment-scope <environment_scope>

--masked <masked>

--masked-and-hidden <masked_and_hidden>

--protected <protected>

--raw <raw>

--variable-type <variable_type>

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab group-variable update

```
usage: gitlab group-variable update [-h] [--sudo SUDO] --group-id GROUP_ID
                                    --key KEY [--description DESCRIPTION]
                                    [--environment-scope ENVIRONMENT_SCOPE]
                                    [--masked MASKED] [--protected PROTECTED]
                                    [--raw RAW] [--value VALUE]
                                    [--variable-type VARIABLE_TYPE]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--key <key>

--description <description>

--environment-scope <environment_scope>

--masked <masked>

--protected <protected>

--raw <raw>

--value <value>

--variable-type <variable_type>

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--with-content <with_content>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab issue

```
usage: gitlab issue [-h] {list,get} ...
```

-h, --help
show this help message and exit

gitlab issue get

```
usage: gitlab issue get [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab issue list

```
usage: gitlab issue list [-h] [--sudo SUDO] [--state STATE] [--labels LABELS]
                         [--milestone MILESTONE] [--scope SCOPE]
                         [--author-id AUTHOR_ID] [--iteration-id ITERATION_ID]
                         [--assignee-id ASSIGNEE_ID]
                         [--my-reaction-emoji MY_REACTION_EMOJI] [--iids IIDS]
                         [--order-by ORDER_BY] [--sort SORT] [--search SEARCH]
                         [--created-after CREATED_AFTER]
                         [--created-before CREATED_BEFORE]
                         [--updated-after UPDATED_AFTER]
                         [--updated-before UPDATED_BEFORE] [--page PAGE]
                         [--per-page PER_PAGE] [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--state <state>

--labels <labels>

--milestone <milestone>

--scope <scope>

--author-id <author_id>

--iteration-id <iteration_id>

--assignee-id <assignee_id>

--my-reaction-emoji <my_reaction_emoji>

--iids <iids>

--order-by <order_by>

--sort <sort>

--search <search>

--created-after <created_after>

--created-before <created_before>

--updated-after <updated_after>

--updated-before <updated_before>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--search <search>

--provider <provider>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                           --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--popular <popular>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab member-role

```
usage: gitlab member-role [-h] {list,create,delete} ...
```

-h, --help
show this help message and exit

gitlab member-role create

```
usage: gitlab member-role create [-h] [--sudo SUDO] --name NAME
                                 --base-access-level BASE_ACCESS_LEVEL
                                 [--description DESCRIPTION]
                                 [--admin-cicd-variables ADMIN_CICD_VARIABLES]
                                 [--admin-compliance-framework ADMIN_COMPLIANCE_
↪FRAMEWORK]
                                 [--admin-group-member ADMIN_GROUP_MEMBER]
                                 [--admin-merge-request ADMIN_MERGE_REQUEST]
                                 [--admin-push-rules ADMIN_PUSH_RULES]
                                 [--admin-terraform-state ADMIN_TERRAFORM_STATE]
                                 [--admin-vulnerability ADMIN_VULNERABILITY]
                                 [--admin-web-hook ADMIN_WEB_HOOK]
                                 [--archive-project ARCHIVE_PROJECT]
                                 [--manage-deploy-tokens MANAGE_DEPLOY_TOKENS]
                                 [--manage-group-access-tokens MANAGE_GROUP_ACCESS_
↪TOKENS]
                                 [--manage-merge-request-settings MANAGE_MERGE_REQUEST_
↪SETTINGS]
                                 [--manage-project-access-tokens MANAGE_PROJECT_ACCESS_
↪TOKENS]
                                 [--manage-security-policy-link MANAGE_SECURITY_POLICY_
↪LINK]
```

(continues on next page)

(continued from previous page)

```
                                    [--read-code READ_CODE]
                                    [--read-runners READ_RUNNERS]
                                    [--read-dependency READ_DEPENDENCY]
                                    [--read-vulnerability READ_VULNERABILITY]
                                    [--remove-group REMOVE_GROUP]
                                    [--remove-project REMOVE_PROJECT]
```

-h, --help
show this help message and exit

--sudo <sudo>

--name <name>

--base-access-level <base_access_level>

--description <description>

--admin-cicd-variables <admin_cicd_variables>

--admin-compliance-framework <admin_compliance_framework>

--admin-group-member <admin_group_member>

--admin-merge-request <admin_merge_request>

--admin-push-rules <admin_push_rules>

--admin-terraform-state <admin_terraform_state>

--admin-vulnerability <admin_vulnerability>

--admin-web-hook <admin_web_hook>

--archive-project <archive_project>

--manage-deploy-tokens <manage_deploy_tokens>

--manage-group-access-tokens <manage_group_access_tokens>

--manage-merge-request-settings <manage_merge_request_settings>

--manage-project-access-tokens <manage_project_access_tokens>

--manage-security-policy-link <manage_security_policy_link>

--read-code <read_code>

--read-runners <read_runners>

--read-dependency <read_dependency>

--read-vulnerability <read_vulnerability>

--remove-group <remove_group>

--remove-project <remove_project>

gitlab member-role delete

```
usage: gitlab member-role delete [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab member-role list

```
usage: gitlab member-role list [-h] [--sudo SUDO] [--page PAGE]
                               [--per-page PER_PAGE] [--get-all |
                               --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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]
```

(continues on next page)

(continued from previous page)

```
                                    [--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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--state <state>

--order-by <order_by>

--sort <sort>

--milestone <milestone>

--view <view>

--labels <labels>

--with-labels-details <with_labels_details>

--with-merge-status-recheck <with_merge_status_recheck>

--created-after <created_after>

--created-before <created_before>

--updated-after <updated_after>

--updated-before <updated_before>

--scope <scope>

--author-id <author_id>

--author-username <author_username>

--assignee-id <assignee_id>

--approver-ids <approver_ids>

--approved-by-ids <approved_by_ids>

--reviewer-id <reviewer_id>

--reviewer-username <reviewer_username>

--my-reaction-emoji <my_reaction_emoji>

--source-branch <source_branch>

--target-branch <target_branch>

--search <search>

--in <in>

--wip <wip>

--not <not>

--environment <environment>

--deployed-before <deployed_before>

--deployed-after <deployed_after>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                             --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--search <search>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab notification-settings

```
usage: gitlab notification-settings [-h] {get,update} ...
```

-h, --help
show this help message and exit

gitlab notification-settings get

```
usage: gitlab notification-settings get [-h] [--sudo SUDO]
```

-h, --help
show this help message and exit

--sudo <sudo>

gitlab notification-settings update

```
usage: gitlab notification-settings update [-h] [--sudo SUDO] [--level LEVEL]
                                           [--notification-email NOTIFICATION_EMAIL]
                                           [--new-note NEW_NOTE]
                                           [--new-issue NEW_ISSUE]
                                           [--reopen-issue REOPEN_ISSUE]
                                           [--close-issue CLOSE_ISSUE]
                                           [--reassign-issue REASSIGN_ISSUE]
                                           [--new-merge-request NEW_MERGE_REQUEST]
                                           [--reopen-merge-request REOPEN_MERGE_REQUEST]
                                           [--close-merge-request CLOSE_MERGE_REQUEST]
                                           [--reassign-merge-request REASSIGN_MERGE_
↪REQUEST]
                                           [--merge-merge-request MERGE_MERGE_REQUEST]
```

-h, --help
show this help message and exit

--sudo <sudo>

--level <level>

--notification-email <notification_email>

--new-note <new_note>

--new-issue <new_issue>

--reopen-issue <reopen_issue>

--close-issue <close_issue>

--reassign-issue <reassign_issue>

--new-merge-request <new_merge_request>

--reopen-merge-request <reopen_merge_request>

--close-merge-request <close_merge_request>

--reassign-merge-request <reassign_merge_request>

--merge-merge-request <merge_merge_request>

gitlab pages-domain

```
usage: gitlab pages-domain [-h] {list} ...
```

-h, --help
show this help message and exit

gitlab pages-domain list

```
usage: gitlab pages-domain list [-h] [--sudo SUDO] [--page PAGE]
                                [--per-page PER_PAGE] [--get-all |
                                --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab personal-access-token

```
usage: gitlab personal-access-token [-h] {list,get,delete,rotate} ...
```

-h, --help
show this help message and exit

gitlab personal-access-token delete

```
usage: gitlab personal-access-token delete [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab personal-access-token get

```
usage: gitlab personal-access-token get [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab personal-access-token list

```
usage: gitlab personal-access-token list [-h] [--sudo SUDO]
                                         [--user-id USER_ID] [--page PAGE]
                                         [--per-page PER_PAGE] [--get-all |
                                         --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--user-id <user_id>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

gitlab personal-access-token rotate

```
usage: gitlab personal-access-token rotate [-h] --id ID
                                           [--expires-at EXPIRES_AT]
```

-h, --help

show this help message and exit

--id <id>

--expires-at <expires_at>

gitlab project

```
usage: gitlab project [-h]
                      {list,get,create,update,delete,upload,update-submodule,repository-
↪tree,repository-blob,repository-raw-blob,repository-compare,repository-contributors,
↪repository-archive,repository-merge-base,delete-merged-branches,create-fork-relation,
↪delete-fork-relation,languages,star,unstar,archive,unarchive,share,unshare,trigger-
↪pipeline,housekeeping,restore,snapshot,search,mirror-pull,mirror-pull-details,transfer}
↪ ...
```

-h, --help

show this help message and exit

gitlab project archive

```
usage: gitlab project archive [-h] --id ID
```

-h, --help

show this help message and exit

--id <id>

gitlab project create

```
usage: gitlab project create [-h] [--sudo SUDO] [--name NAME] [--path PATH]
                             [--allow-merge-on-skipped-pipeline ALLOW_MERGE_ON_SKIPPED_
↪PIPELINE]
                             [--only-allow-merge-if-all-status-checks-passed ONLY_ALLOW_
↪MERGE_IF_ALL_STATUS_CHECKS_PASSED]
                             [--analytics-access-level ANALYTICS_ACCESS_LEVEL]
```

(continues on next page)

(continued from previous page)

```
                              [--approvals-before-merge APPROVALS_BEFORE_MERGE]
                              [--auto-cancel-pending-pipelines AUTO_CANCEL_PENDING_
↪PIPELINES]
                              [--auto-devops-deploy-strategy AUTO_DEVOPS_DEPLOY_STRATEGY]
                              [--auto-devops-enabled AUTO_DEVOPS_ENABLED]
                              [--autoclose-referenced-issues AUTOCLOSE_REFERENCED_ISSUES]
                              [--avatar AVATAR]
                              [--build-coverage-regex BUILD_COVERAGE_REGEX]
                              [--build-git-strategy BUILD_GIT_STRATEGY]
                              [--build-timeout BUILD_TIMEOUT]
                              [--builds-access-level BUILDS_ACCESS_LEVEL]
                              [--ci-config-path CI_CONFIG_PATH]
                              [--container-expiration-policy-attributes CONTAINER_
↪EXPIRATION_POLICY_ATTRIBUTES]
                              [--container-registry-access-level CONTAINER_REGISTRY_
↪ACCESS_LEVEL]
                              [--container-registry-enabled CONTAINER_REGISTRY_ENABLED]
                              [--default-branch DEFAULT_BRANCH]
                              [--description DESCRIPTION]
                              [--emails-disabled EMAILS_DISABLED]
                              [--external-authorization-classification-label EXTERNAL_
↪AUTHORIZATION_CLASSIFICATION_LABEL]
                              [--forking-access-level FORKING_ACCESS_LEVEL]
                              [--group-with-project-templates-id GROUP_WITH_PROJECT_
↪TEMPLATES_ID]
                              [--import-url IMPORT_URL]
                              [--initialize-with-readme INITIALIZE_WITH_README]
                              [--issues-access-level ISSUES_ACCESS_LEVEL]
                              [--issues-enabled ISSUES_ENABLED]
                              [--jobs-enabled JOBS_ENABLED]
                              [--lfs-enabled LFS_ENABLED]
                              [--merge-method MERGE_METHOD]
                              [--merge-pipelines-enabled MERGE_PIPELINES_ENABLED]
                              [--merge-requests-access-level MERGE_REQUESTS_ACCESS_LEVEL]
                              [--merge-requests-enabled MERGE_REQUESTS_ENABLED]
                              [--mirror-trigger-builds MIRROR_TRIGGER_BUILDS]
                              [--mirror MIRROR] [--namespace-id NAMESPACE_ID]
                              [--operations-access-level OPERATIONS_ACCESS_LEVEL]
                              [--only-allow-merge-if-all-discussions-are-resolved ONLY_
↪ALLOW_MERGE_IF_ALL_DISCUSSIONS_ARE_RESOLVED]
                              [--only-allow-merge-if-pipeline-succeeds ONLY_ALLOW_MERGE_
↪IF_PIPELINE_SUCCEEDS]
                              [--packages-enabled PACKAGES_ENABLED]
                              [--pages-access-level PAGES_ACCESS_LEVEL]
                              [--requirements-access-level REQUIREMENTS_ACCESS_LEVEL]
                              [--printing-merge-request-link-enabled PRINTING_MERGE_
↪REQUEST_LINK_ENABLED]
                              [--public-builds PUBLIC_BUILDS]
                              [--releases-access-level RELEASES_ACCESS_LEVEL]
                              [--environments-access-level ENVIRONMENTS_ACCESS_LEVEL]
                              [--feature-flags-access-level FEATURE_FLAGS_ACCESS_LEVEL]
                              [--infrastructure-access-level INFRASTRUCTURE_ACCESS_LEVEL]
```

(continues on next page)

(continued from previous page)

```
                              [--monitor-access-level MONITOR_ACCESS_LEVEL]
                              [--remove-source-branch-after-merge REMOVE_SOURCE_BRANCH_
↪AFTER_MERGE]
                              [--repository-access-level REPOSITORY_ACCESS_LEVEL]
                              [--repository-storage REPOSITORY_STORAGE]
                              [--request-access-enabled REQUEST_ACCESS_ENABLED]
                              [--resolve-outdated-diff-discussions RESOLVE_OUTDATED_DIFF_
↪DISCUSSIONS]
                              [--security-and-compliance-access-level SECURITY_AND_
↪COMPLIANCE_ACCESS_LEVEL]
                              [--shared-runners-enabled SHARED_RUNNERS_ENABLED]
                              [--show-default-award-emojis SHOW_DEFAULT_AWARD_EMOJIS]
                              [--snippets-access-level SNIPPETS_ACCESS_LEVEL]
                              [--snippets-enabled SNIPPETS_ENABLED]
                              [--squash-option SQUASH_OPTION]
                              [--tag-list TAG_LIST] [--topics TOPICS]
                              [--template-name TEMPLATE_NAME]
                              [--template-project-id TEMPLATE_PROJECT_ID]
                              [--use-custom-template USE_CUSTOM_TEMPLATE]
                              [--visibility VISIBILITY]
                              [--wiki-access-level WIKI_ACCESS_LEVEL]
                              [--wiki-enabled WIKI_ENABLED]
```

-h, --help

show this help message and exit

--sudo <sudo>

--name <name>

--path <path>

--allow-merge-on-skipped-pipeline <allow_merge_on_skipped_pipeline>

--only-allow-merge-if-all-status-checks-passed
<only_allow_merge_if_all_status_checks_passed>

--analytics-access-level <analytics_access_level>

--approvals-before-merge <approvals_before_merge>

--auto-cancel-pending-pipelines <auto_cancel_pending_pipelines>

--auto-devops-deploy-strategy <auto_devops_deploy_strategy>

--auto-devops-enabled <auto_devops_enabled>

--autoclose-referenced-issues <autoclose_referenced_issues>

--avatar <avatar>

--build-coverage-regex <build_coverage_regex>

--build-git-strategy <build_git_strategy>

--build-timeout <build_timeout>

--builds-access-level <builds_access_level>

--ci-config-path <ci_config_path>

--container-expiration-policy-attributes <container_expiration_policy_attributes>

--container-registry-access-level <container_registry_access_level>

--container-registry-enabled <container_registry_enabled>

--default-branch <default_branch>

--description <description>

--emails-disabled <emails_disabled>

--external-authorization-classification-label <external_authorization_classification_label>

--forking-access-level <forking_access_level>

--group-with-project-templates-id <group_with_project_templates_id>

--import-url <import_url>

--initialize-with-readme <initialize_with_readme>

--issues-access-level <issues_access_level>

--issues-enabled <issues_enabled>

--jobs-enabled <jobs_enabled>

--lfs-enabled <lfs_enabled>

--merge-method <merge_method>

--merge-pipelines-enabled <merge_pipelines_enabled>

--merge-requests-access-level <merge_requests_access_level>

--merge-requests-enabled <merge_requests_enabled>

--mirror-trigger-builds <mirror_trigger_builds>

--mirror <mirror>

--namespace-id <namespace_id>

--operations-access-level <operations_access_level>

--only-allow-merge-if-all-discussions-are-resolved <only_allow_merge_if_all_discussions_are_resolved>

--only-allow-merge-if-pipeline-succeeds <only_allow_merge_if_pipeline_succeeds>

--packages-enabled <packages_enabled>

--pages-access-level <pages_access_level>

--requirements-access-level <requirements_access_level>

--printing-merge-request-link-enabled <printing_merge_request_link_enabled>

--public-builds <public_builds>

--releases-access-level <releases_access_level>

--environments-access-level <environments_access_level>

--feature-flags-access-level <feature_flags_access_level>

--infrastructure-access-level <infrastructure_access_level>

--monitor-access-level <monitor_access_level>

--remove-source-branch-after-merge <remove_source_branch_after_merge>

--repository-access-level <repository_access_level>

--repository-storage <repository_storage>

--request-access-enabled <request_access_enabled>

--resolve-outdated-diff-discussions <resolve_outdated_diff_discussions>

--security-and-compliance-access-level <security_and_compliance_access_level>

--shared-runners-enabled <shared_runners_enabled>

--show-default-award-emojis <show_default_award_emojis>

--snippets-access-level <snippets_access_level>

--snippets-enabled <snippets_enabled>

--squash-option <squash_option>

--tag-list <tag_list>

--topics <topics>

--template-name <template_name>

--template-project-id <template_project_id>

--use-custom-template <use_custom_template>

--visibility <visibility>

--wiki-access-level <wiki_access_level>

--wiki-enabled <wiki_enabled>

gitlab project create-fork-relation

```
usage: gitlab project create-fork-relation [-h] --id ID
                                           --forked-from-id FORKED_FROM_ID
```

-h, --help
show this help message and exit

--id <id>

--forked-from-id <forked_from_id>

gitlab project delete

```
usage: gitlab project delete [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab project delete-fork-relation

```
usage: gitlab project delete-fork-relation [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab project delete-merged-branches

```
usage: gitlab project delete-merged-branches [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab project get

```
usage: gitlab project get [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab project housekeeping

```
usage: gitlab project housekeeping [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab project languages

```
usage: gitlab project languages [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab project list

```
usage: gitlab project list [-h] [--sudo SUDO] [--archived ARCHIVED]
                           [--id-after ID_AFTER] [--id-before ID_BEFORE]
                           [--last-activity-after LAST_ACTIVITY_AFTER]
                           [--last-activity-before LAST_ACTIVITY_BEFORE]
                           [--membership MEMBERSHIP]
                           [--min-access-level MIN_ACCESS_LEVEL]
                           [--order-by ORDER_BY] [--owned OWNED]
                           [--repository-checksum-failed REPOSITORY_CHECKSUM_FAILED]
                           [--repository-storage REPOSITORY_STORAGE]
                           [--search-namespaces SEARCH_NAMESPACES]
                           [--search SEARCH] [--simple SIMPLE] [--sort SORT]
                           [--starred STARRED] [--statistics STATISTICS]
                           [--topic TOPIC] [--visibility VISIBILITY]
                           [--wiki-checksum-failed WIKI_CHECKSUM_FAILED]
                           [--with-custom-attributes WITH_CUSTOM_ATTRIBUTES]
                           [--with-issues-enabled WITH_ISSUES_ENABLED]
                           [--with-merge-requests-enabled WITH_MERGE_REQUESTS_ENABLED]
                           [--with-programming-language WITH_PROGRAMMING_LANGUAGE]
                           [--page PAGE] [--per-page PER_PAGE] [--get-all |
                           --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--archived <archived>

--id-after <id_after>

--id-before <id_before>

--last-activity-after <last_activity_after>

--last-activity-before <last_activity_before>

--membership <membership>

--min-access-level <min_access_level>

--order-by <order_by>

--owned <owned>

--repository-checksum-failed <repository_checksum_failed>

--repository-storage <repository_storage>

--search-namespaces <search_namespaces>

--search <search>

--simple <simple>

--sort <sort>

--starred <starred>

--statistics <statistics>

--topic <topic>

--visibility <visibility>

--wiki-checksum-failed <wiki_checksum_failed>

--with-custom-attributes <with_custom_attributes>

--with-issues-enabled <with_issues_enabled>

--with-merge-requests-enabled <with_merge_requests_enabled>

--with-programming-language <with_programming_language>

--page <page>

--per-page <per_page>

--get-all
: Return all items from the server, without pagination.

--no-get-all
: Don't return all items from the server.

gitlab project mirror-pull

```
usage: gitlab project mirror-pull [-h] --id ID
```

-h, --help
: show this help message and exit

--id <id>

gitlab project mirror-pull-details

```
usage: gitlab project mirror-pull-details [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab project repository-archive

```
usage: gitlab project repository-archive [-h] --id ID [--sha SHA]
                                         [--format FORMAT]
```

-h, --help
show this help message and exit

--id <id>

--sha <sha>

--format <format>

gitlab project repository-blob

```
usage: gitlab project repository-blob [-h] --id ID --sha SHA
```

-h, --help
show this help message and exit

--id <id>

--sha <sha>

gitlab project repository-compare

```
usage: gitlab project repository-compare [-h] --id ID --from- FROM_ --to TO
```

-h, --help
show this help message and exit

--id <id>

--from- <from_>

--to <to>

gitlab project repository-contributors

```
usage: gitlab project repository-contributors [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab project repository-merge-base

```
usage: gitlab project repository-merge-base [-h] --id ID --refs REFS
```

-h, --help
show this help message and exit

--id <id>

--refs <refs>

gitlab project repository-raw-blob

```
usage: gitlab project repository-raw-blob [-h] --id ID --sha SHA
```

-h, --help
show this help message and exit

--id <id>

--sha <sha>

gitlab project repository-tree

```
usage: gitlab project repository-tree [-h] --id ID [--path PATH] [--ref REF]
                                      [--recursive RECURSIVE]
```

-h, --help
show this help message and exit

--id <id>

--path <path>

--ref <ref>

--recursive <recursive>

gitlab project restore

```
usage: gitlab project restore [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab project search

```
usage: gitlab project search [-h] --id ID --scope SCOPE --search SEARCH
```

-h, --help
show this help message and exit

--id <id>

--scope <scope>

--search <search>

gitlab project share

```
usage: gitlab project share [-h] --id ID --group-id GROUP_ID
                            --group-access GROUP_ACCESS
                            [--expires-at EXPIRES_AT]
```

-h, --help
show this help message and exit

--id <id>

--group-id <group_id>

--group-access <group_access>

--expires-at <expires_at>

gitlab project snapshot

```
usage: gitlab project snapshot [-h] --id ID [--wiki WIKI]
```

-h, --help
show this help message and exit

--id <id>

--wiki <wiki>

gitlab project star

```
usage: gitlab project star [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab project transfer

```
usage: gitlab project transfer [-h] --id ID --to-namespace TO_NAMESPACE
```

-h, --help
show this help message and exit

--id <id>

--to-namespace <to_namespace>

gitlab project trigger-pipeline

```
usage: gitlab project trigger-pipeline [-h] --id ID --ref REF --token TOKEN
```

-h, --help
show this help message and exit

--id <id>

--ref <ref>

--token <token>

gitlab project unarchive

```
usage: gitlab project unarchive [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab project unshare

```
usage: gitlab project unshare [-h] --id ID --group-id GROUP_ID
```

-h, --help
show this help message and exit

--id <id>

--group-id <group_id>

gitlab project unstar

```
usage: gitlab project unstar [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab project update

```
usage: gitlab project update [-h] [--sudo SUDO] --id ID
                             [--allow-merge-on-skipped-pipeline ALLOW_MERGE_ON_SKIPPED_
↪PIPELINE]
                             [--only-allow-merge-if-all-status-checks-passed ONLY_ALLOW_
↪MERGE_IF_ALL_STATUS_CHECKS_PASSED]
                             [--analytics-access-level ANALYTICS_ACCESS_LEVEL]
                             [--approvals-before-merge APPROVALS_BEFORE_MERGE]
                             [--auto-cancel-pending-pipelines AUTO_CANCEL_PENDING_
↪PIPELINES]
                             [--auto-devops-deploy-strategy AUTO_DEVOPS_DEPLOY_STRATEGY]
```

(continues on next page)

(continued from previous page)

```
                              [--auto-devops-enabled AUTO_DEVOPS_ENABLED]
                              [--autoclose-referenced-issues AUTOCLOSE_REFERENCED_ISSUES]
                              [--avatar AVATAR]
                              [--build-coverage-regex BUILD_COVERAGE_REGEX]
                              [--build-git-strategy BUILD_GIT_STRATEGY]
                              [--build-timeout BUILD_TIMEOUT]
                              [--builds-access-level BUILDS_ACCESS_LEVEL]
                              [--ci-config-path CI_CONFIG_PATH]
                              [--ci-default-git-depth CI_DEFAULT_GIT_DEPTH]
                              [--ci-forward-deployment-enabled CI_FORWARD_DEPLOYMENT_
↪ENABLED]
                              [--ci-allow-fork-pipelines-to-run-in-parent-project CI_
↪ALLOW_FORK_PIPELINES_TO_RUN_IN_PARENT_PROJECT]
                              [--ci-separated-caches CI_SEPARATED_CACHES]
                              [--container-expiration-policy-attributes CONTAINER_
↪EXPIRATION_POLICY_ATTRIBUTES]
                              [--container-registry-access-level CONTAINER_REGISTRY_
↪ACCESS_LEVEL]
                              [--container-registry-enabled CONTAINER_REGISTRY_ENABLED]
                              [--default-branch DEFAULT_BRANCH]
                              [--description DESCRIPTION]
                              [--emails-disabled EMAILS_DISABLED]
                              [--enforce-auth-checks-on-uploads ENFORCE_AUTH_CHECKS_ON_
↪UPLOADS]
                              [--external-authorization-classification-label EXTERNAL_
↪AUTHORIZATION_CLASSIFICATION_LABEL]
                              [--forking-access-level FORKING_ACCESS_LEVEL]
                              [--import-url IMPORT_URL]
                              [--issues-access-level ISSUES_ACCESS_LEVEL]
                              [--issues-enabled ISSUES_ENABLED]
                              [--issues-template ISSUES_TEMPLATE]
                              [--jobs-enabled JOBS_ENABLED]
                              [--keep-latest-artifact KEEP_LATEST_ARTIFACT]
                              [--lfs-enabled LFS_ENABLED]
                              [--merge-commit-template MERGE_COMMIT_TEMPLATE]
                              [--merge-method MERGE_METHOD]
                              [--merge-pipelines-enabled MERGE_PIPELINES_ENABLED]
                              [--merge-requests-access-level MERGE_REQUESTS_ACCESS_LEVEL]
                              [--merge-requests-enabled MERGE_REQUESTS_ENABLED]
                              [--merge-requests-template MERGE_REQUESTS_TEMPLATE]
                              [--merge-trains-enabled MERGE_TRAINS_ENABLED]
                              [--mirror-overwrites-diverged-branches MIRROR_OVERWRITES_
↪DIVERGED_BRANCHES]
                              [--mirror-trigger-builds MIRROR_TRIGGER_BUILDS]
                              [--mirror-user-id MIRROR_USER_ID]
                              [--mirror MIRROR]
                              [--mr-default-target-self MR_DEFAULT_TARGET_SELF]
                              [--name NAME]
                              [--operations-access-level OPERATIONS_ACCESS_LEVEL]
                              [--only-allow-merge-if-all-discussions-are-resolved ONLY_
↪ALLOW_MERGE_IF_ALL_DISCUSSIONS_ARE_RESOLVED]
                              [--only-allow-merge-if-pipeline-succeeds ONLY_ALLOW_MERGE_
```

(continues on next page)

(continued from previous page)

```
↪IF_PIPELINE_SUCCEEDS]
                            [--only-mirror-protected-branches ONLY_MIRROR_PROTECTED_
↪BRANCHES]
                            [--packages-enabled PACKAGES_ENABLED]
                            [--pages-access-level PAGES_ACCESS_LEVEL]
                            [--requirements-access-level REQUIREMENTS_ACCESS_LEVEL]
                            [--restrict-user-defined-variables RESTRICT_USER_DEFINED_
↪VARIABLES]
                            [--path PATH] [--public-builds PUBLIC_BUILDS]
                            [--releases-access-level RELEASES_ACCESS_LEVEL]
                            [--environments-access-level ENVIRONMENTS_ACCESS_LEVEL]
                            [--feature-flags-access-level FEATURE_FLAGS_ACCESS_LEVEL]
                            [--infrastructure-access-level INFRASTRUCTURE_ACCESS_LEVEL]
                            [--monitor-access-level MONITOR_ACCESS_LEVEL]
                            [--remove-source-branch-after-merge REMOVE_SOURCE_BRANCH_
↪AFTER_MERGE]
                            [--repository-access-level REPOSITORY_ACCESS_LEVEL]
                            [--repository-storage REPOSITORY_STORAGE]
                            [--request-access-enabled REQUEST_ACCESS_ENABLED]
                            [--resolve-outdated-diff-discussions RESOLVE_OUTDATED_DIFF_
↪DISCUSSIONS]
                            [--security-and-compliance-access-level SECURITY_AND_
↪COMPLIANCE_ACCESS_LEVEL]
                            [--service-desk-enabled SERVICE_DESK_ENABLED]
                            [--shared-runners-enabled SHARED_RUNNERS_ENABLED]
                            [--show-default-award-emojis SHOW_DEFAULT_AWARD_EMOJIS]
                            [--snippets-access-level SNIPPETS_ACCESS_LEVEL]
                            [--snippets-enabled SNIPPETS_ENABLED]
                            [--issue-branch-template ISSUE_BRANCH_TEMPLATE]
                            [--squash-commit-template SQUASH_COMMIT_TEMPLATE]
                            [--squash-option SQUASH_OPTION]
                            [--suggestion-commit-message SUGGESTION_COMMIT_MESSAGE]
                            [--tag-list TAG_LIST] [--topics TOPICS]
                            [--visibility VISIBILITY]
                            [--wiki-access-level WIKI_ACCESS_LEVEL]
                            [--wiki-enabled WIKI_ENABLED]
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

--allow-merge-on-skipped-pipeline <allow_merge_on_skipped_pipeline>

--only-allow-merge-if-all-status-checks-passed
<only_allow_merge_if_all_status_checks_passed>

--analytics-access-level <analytics_access_level>

--approvals-before-merge <approvals_before_merge>

--auto-cancel-pending-pipelines <auto_cancel_pending_pipelines>

--auto-devops-deploy-strategy <auto_devops_deploy_strategy>

--auto-devops-enabled <auto_devops_enabled>

--autoclose-referenced-issues <autoclose_referenced_issues>

--avatar <avatar>

--build-coverage-regex <build_coverage_regex>

--build-git-strategy <build_git_strategy>

--build-timeout <build_timeout>

--builds-access-level <builds_access_level>

--ci-config-path <ci_config_path>

--ci-default-git-depth <ci_default_git_depth>

--ci-forward-deployment-enabled <ci_forward_deployment_enabled>

--ci-allow-fork-pipelines-to-run-in-parent-project <ci_allow_fork_pipelines_to_run_in_parent_project>

--ci-separated-caches <ci_separated_caches>

--container-expiration-policy-attributes <container_expiration_policy_attributes>

--container-registry-access-level <container_registry_access_level>

--container-registry-enabled <container_registry_enabled>

--default-branch <default_branch>

--description <description>

--emails-disabled <emails_disabled>

--enforce-auth-checks-on-uploads <enforce_auth_checks_on_uploads>

--external-authorization-classification-label <external_authorization_classification_label>

--forking-access-level <forking_access_level>

--import-url <import_url>

--issues-access-level <issues_access_level>

--issues-enabled <issues_enabled>

--issues-template <issues_template>

--jobs-enabled <jobs_enabled>

--keep-latest-artifact <keep_latest_artifact>

--lfs-enabled <lfs_enabled>

--merge-commit-template <merge_commit_template>

--merge-method <merge_method>

--merge-pipelines-enabled <merge_pipelines_enabled>

--merge-requests-access-level <merge_requests_access_level>

--merge-requests-enabled <merge_requests_enabled>

--merge-requests-template <merge_requests_template>

--merge-trains-enabled <merge_trains_enabled>

--mirror-overwrites-diverged-branches <mirror_overwrites_diverged_branches>

--mirror-trigger-builds <mirror_trigger_builds>

--mirror-user-id <mirror_user_id>

--mirror <mirror>

--mr-default-target-self <mr_default_target_self>

--name <name>

--operations-access-level <operations_access_level>

--only-allow-merge-if-all-discussions-are-resolved
<only_allow_merge_if_all_discussions_are_resolved>

--only-allow-merge-if-pipeline-succeeds <only_allow_merge_if_pipeline_succeeds>

--only-mirror-protected-branches <only_mirror_protected_branches>

--packages-enabled <packages_enabled>

--pages-access-level <pages_access_level>

--requirements-access-level <requirements_access_level>

--restrict-user-defined-variables <restrict_user_defined_variables>

--path <path>

--public-builds <public_builds>

--releases-access-level <releases_access_level>

--environments-access-level <environments_access_level>

--feature-flags-access-level <feature_flags_access_level>

--infrastructure-access-level <infrastructure_access_level>

--monitor-access-level <monitor_access_level>

--remove-source-branch-after-merge <remove_source_branch_after_merge>

--repository-access-level <repository_access_level>

--repository-storage <repository_storage>

--request-access-enabled <request_access_enabled>

--resolve-outdated-diff-discussions <resolve_outdated_diff_discussions>

--security-and-compliance-access-level <security_and_compliance_access_level>

--service-desk-enabled <service_desk_enabled>

--shared-runners-enabled <shared_runners_enabled>

--show-default-award-emojis <show_default_award_emojis>

--snippets-access-level <snippets_access_level>

--snippets-enabled <snippets_enabled>

--issue-branch-template <issue_branch_template>

--squash-commit-template <squash_commit_template>

--squash-option <squash_option>

--suggestion-commit-message <suggestion_commit_message>

--tag-list <tag_list>

--topics <topics>

--visibility <visibility>

--wiki-access-level <wiki_access_level>

--wiki-enabled <wiki_enabled>

gitlab project update-submodule

```
usage: gitlab project update-submodule [-h] --id ID --submodule SUBMODULE
                                       --branch BRANCH --commit-sha COMMIT_SHA
```

-h, **--help**
show this help message and exit

--id <id>

--submodule <submodule>

--branch <branch>

--commit-sha <commit_sha>

gitlab project upload

```
usage: gitlab project upload [-h] --id ID --filename FILENAME
                             --filepath FILEPATH
```

-h, --help
show this help message and exit

--id <id>

--filename <filename>

--filepath <filepath>

gitlab project-access-request

```
usage: gitlab project-access-request [-h] {list,create,delete,approve} ...
```

-h, --help
show this help message and exit

gitlab project-access-request approve

```
usage: gitlab project-access-request approve [-h] --project-id PROJECT_ID
                                             [--sudo SUDO] --id ID
                                             [--access-level ACCESS_LEVEL]
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--id <id>

--access-level <access_level>

gitlab project-access-request create

```
usage: gitlab project-access-request create [-h] [--sudo SUDO]
                                            --project-id PROJECT_ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

gitlab project-access-request delete

```
usage: gitlab project-access-request delete [-h] [--sudo SUDO]
                                            --project-id PROJECT_ID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

gitlab project-access-request list

```
usage: gitlab project-access-request list [-h] [--sudo SUDO]
                                          --project-id PROJECT_ID
                                          [--page PAGE] [--per-page PER_PAGE]
                                          [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-access-token

```
usage: gitlab project-access-token [-h] {list,get,create,delete,rotate} ...
```

-h, --help
show this help message and exit

gitlab project-access-token create

```
usage: gitlab project-access-token create [-h] [--sudo SUDO]
                                          --project-id PROJECT_ID --name NAME
                                          --scopes SCOPES
                                          [--access-level ACCESS_LEVEL]
                                          [--expires-at EXPIRES_AT]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--name <name>

--scopes <scopes>

--access-level <access_level>

--expires-at <expires_at>

gitlab project-access-token delete

```
usage: gitlab project-access-token delete [-h] [--sudo SUDO]
                                          --project-id PROJECT_ID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

gitlab project-access-token get

```
usage: gitlab project-access-token get [-h] [--sudo SUDO]
                                       --project-id PROJECT_ID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

gitlab project-access-token list

```
usage: gitlab project-access-token list [-h] [--sudo SUDO]
                                        --project-id PROJECT_ID [--page PAGE]
                                        [--per-page PER_PAGE] [--get-all |
                                        --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-access-token rotate

```
usage: gitlab project-access-token rotate [-h] --project-id PROJECT_ID
                                          [--sudo SUDO] --id ID
                                          [--expires-at EXPIRES_AT]
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--id <id>

--expires-at <expires_at>

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,get,create,update,delete} ...
```

-h, --help
show this help message and exit

gitlab project-approval-rule create

```
usage: gitlab project-approval-rule create [-h] [--sudo SUDO]
                                           --project-id PROJECT_ID --name NAME
                                           --approvals-required APPROVALS_REQUIRED
                                           [--user-ids USER_IDS]
                                           [--group-ids GROUP_IDS]
                                           [--protected-branch-ids PROTECTED_BRANCH_IDS]
                                           [--usernames USERNAMES]
                                           [--applies-to-all-protected-branches APPLIES_
↪TO_ALL_PROTECTED_BRANCHES]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--name <name>

--approvals-required <approvals_required>

--user-ids <user_ids>

--group-ids <group_ids>

--protected-branch-ids <protected_branch_ids>

--usernames <usernames>

--applies-to-all-protected-branches <applies_to_all_protected_branches>

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 get

```
usage: gitlab project-approval-rule 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-approval-rule list

```
usage: gitlab project-approval-rule list [-h] [--sudo SUDO]
                                         --project-id PROJECT_ID [--page PAGE]
                                         [--per-page PER_PAGE] [--get-all |
                                         --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--created-after <created_after>

--created-before <created_before>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-board update

```
usage: gitlab project-board update [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

gitlab project-board-list

```
usage: gitlab project-board-list [-h] {list,get,create,update,delete} ...
```

-h, --help
show this help message and exit

gitlab project-board-list create

```
usage: gitlab project-board-list create [-h] [--sudo SUDO]
                                        --project-id PROJECT_ID
                                        --board-id BOARD_ID
                                        [--label-id LABEL_ID |
                                        --assignee-id ASSIGNEE_ID |
                                        --milestone-id MILESTONE_ID |
                                        --iteration-id ITERATION_ID]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--board-id <board_id>

--label-id <label_id>

--assignee-id <assignee_id>

--milestone-id <milestone_id>

--iteration-id <iteration_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 |
                                      --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--board-id <board_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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
                                  [--content-ref CONTENT_REF]
                                  [--dry-run DRY_RUN]
                                  [--dry-run-ref DRY_RUN_REF]
                                  [--include-jobs INCLUDE_JOBS] [--ref REF]
```

-h, **--help**
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--content-ref <content_ref>

--dry-run <dry_run>

--dry-run-ref <dry_run_ref>

--include-jobs <include_jobs>

--ref <ref>

gitlab project-ci-lint validate

```
usage: gitlab project-ci-lint validate [-h] --project-id PROJECT_ID
                                       [--sudo SUDO] --content CONTENT
                                       [--dry-run DRY_RUN]
                                       [--include-jobs INCLUDE_JOBS]
                                       [--ref REF]
```

-h, **--help**
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--content <content>

--dry-run <dry_run>

--include-jobs <include_jobs>

--ref <ref>

gitlab project-cluster

```
usage: gitlab project-cluster [-h] {list,get,create,update,delete} ...
```

-h, **--help**
show this help message and exit

gitlab project-cluster create

```
usage: gitlab project-cluster create [-h] [--sudo SUDO]
                                     --project-id PROJECT_ID --name NAME
                                     --platform-kubernetes-attributes PLATFORM_
↪KUBERNETES_ATTRIBUTES
                                     [--domain DOMAIN] [--enabled ENABLED]
                                     [--managed MANAGED]
                                     [--environment-scope ENVIRONMENT_SCOPE]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--name <name>

--platform-kubernetes-attributes <platform_kubernetes_attributes>

--domain <domain>

--enabled <enabled>

--managed <managed>

--environment-scope <environment_scope>

gitlab project-cluster delete

```
usage: gitlab project-cluster delete [-h] [--sudo SUDO]
                                     --project-id PROJECT_ID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

gitlab project-cluster get

```
usage: gitlab project-cluster get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                  --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

gitlab project-cluster list

```
usage: gitlab project-cluster list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   [--page PAGE] [--per-page PER_PAGE]
                                   [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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-cluster-agent

```
usage: gitlab project-cluster-agent [-h] {list,get,create,delete} ...
```

-h, --help
show this help message and exit

gitlab project-cluster-agent create

```
usage: gitlab project-cluster-agent 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-cluster-agent delete

```
usage: gitlab project-cluster-agent 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-agent get

```
usage: gitlab project-cluster-agent 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-agent list

```
usage: gitlab project-cluster-agent list [-h] [--sudo SUDO]
                                         --project-id PROJECT_ID [--page PAGE]
                                         [--per-page PER_PAGE] [--get-all |
                                         --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-commit

```
usage: gitlab project-commit [-h]
                             {list,get,create,diff,cherry-pick,refs,merge-requests,
↪revert,sequence,signature} ...
```

-h, --help
show this help message and exit

gitlab project-commit cherry-pick

```
usage: gitlab project-commit cherry-pick [-h] --project-id PROJECT_ID
                                         [--sudo SUDO] --id ID --branch BRANCH
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--id <id>

--branch <branch>

gitlab project-commit create

```
usage: gitlab project-commit create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                    --branch BRANCH
                                    --commit-message COMMIT_MESSAGE
                                    --actions ACTIONS
                                    [--author-email AUTHOR_EMAIL]
                                    [--author-name AUTHOR_NAME]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--branch <branch>

--commit-message <commit_message>

--actions <actions>

--author-email <author_email>

--author-name <author_name>

gitlab project-commit diff

```
usage: gitlab project-commit diff [-h] --project-id PROJECT_ID [--sudo SUDO]
                                  --id ID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--id <id>

gitlab project-commit get

```
usage: gitlab project-commit get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                 --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

gitlab project-commit list

```
usage: gitlab project-commit list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                  [--all ALL] [--ref-name REF_NAME]
                                  [--since SINCE] [--until UNTIL]
                                  [--path PATH] [--with-stats WITH_STATS]
                                  [--first-parent FIRST_PARENT]
                                  [--order ORDER] [--trailers TRAILERS]
                                  [--page PAGE] [--per-page PER_PAGE]
                                  [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--all <all>

--ref-name <ref_name>

--since <since>

--until <until>

--path <path>

--with-stats <with_stats>

--first-parent <first_parent>

--order <order>

--trailers <trailers>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 sequence

```
usage: gitlab project-commit sequence [-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 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 |
                                          --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--commit-id <commit_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--commit-id <commit_id>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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 |
                                         --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--commit-id <commit_id>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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 |
                                            --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                        --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--scopes <scopes>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-deployment

```
usage: gitlab project-deployment [-h] {list,get,create,update,approval} ...
```

-h, --help
show this help message and exit

gitlab project-deployment approval

```
usage: gitlab project-deployment approval [-h] --project-id PROJECT_ID
                                          [--sudo SUDO] --id ID
                                          --status STATUS [--comment COMMENT]
                                          [--represented-as REPRESENTED_AS]
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--id <id>

--status <status>

--comment <comment>

--represented-as <represented_as>

gitlab project-deployment create

```
usage: gitlab project-deployment create [-h] [--sudo SUDO]
                                        --project-id PROJECT_ID --sha SHA
                                        --ref REF --tag TAG --status STATUS
                                        --environment ENVIRONMENT
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--sha <sha>

--ref <ref>

--tag <tag>

--status <status>

--environment <environment>

gitlab project-deployment get

```
usage: gitlab project-deployment get [-h] [--sudo SUDO]
                                     --project-id PROJECT_ID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

gitlab project-deployment list

```
usage: gitlab project-deployment list [-h] [--sudo SUDO]
                                      --project-id PROJECT_ID
                                      [--order-by ORDER_BY] [--sort SORT]
                                      [--updated-after UPDATED_AFTER]
                                      [--updated-before UPDATED_BEFORE]
                                      [--environment ENVIRONMENT]
                                      [--status STATUS] [--page PAGE]
                                      [--per-page PER_PAGE] [--get-all |
                                      --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--order-by <order_by>

--sort <sort>

--updated-after <updated_after>

--updated-before <updated_before>

--environment <environment>

--status <status>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--deployment-id <deployment_id>

--project-id <project_id>

--state <state>

--order-by <order_by>

--sort <sort>

--milestone <milestone>

--view <view>

--labels <labels>

--with-labels-details <with_labels_details>

--with-merge-status-recheck <with_merge_status_recheck>

--created-after <created_after>

--created-before <created_before>

--updated-after <updated_after>

--updated-before <updated_before>

--scope <scope>

--author-id <author_id>

--author-username <author_username>

--assignee-id <assignee_id>

--approver-ids <approver_ids>

--approved-by-ids <approved_by_ids>

--reviewer-id <reviewer_id>

--reviewer-username <reviewer_username>

--my-reaction-emoji <my_reaction_emoji>

--source-branch <source_branch>

--target-branch <target_branch>

--search <search>

--in <in>

--wip <wip>

--not <not>

--environment <environment>

--deployed-before <deployed_before>

--deployed-after <deployed_after>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-dockerfile-template

```
usage: gitlab project-dockerfile-template [-h] {list,get} ...
```

-h, --help
show this help message and exit

gitlab project-dockerfile-template get

```
usage: gitlab project-dockerfile-template 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-dockerfile-template list

```
usage: gitlab project-dockerfile-template list [-h] [--sudo SUDO]
                                               --project-id PROJECT_ID
                                               [--page PAGE]
                                               [--per-page PER_PAGE]
                                               [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--name <name>

--search <search>

--states <states>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-environment stop

```
usage: gitlab project-environment stop [-h] --project-id PROJECT_ID
                                       [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--id <id>

gitlab project-environment update

```
usage: gitlab project-environment update [-h] [--sudo SUDO]
                                         --project-id PROJECT_ID --id ID
                                         [--name NAME]
                                         [--external-url EXTERNAL_URL]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

--name <name>

--external-url <external_url>

gitlab project-event

```
usage: gitlab project-event [-h] {list} ...
```

-h, **--help**
show this help message and exit

gitlab project-event list

```
usage: gitlab project-event list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                 [--action ACTION] [--target-type TARGET_TYPE]
                                 [--before BEFORE] [--after AFTER]
                                 [--sort SORT] [--scope SCOPE] [--page PAGE]
                                 [--per-page PER_PAGE] [--get-all |
                                 --no-get-all]
```

-h, **--help**
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--action <action>

--target-type <target_type>

--before <before>

--after <after>

--sort <sort>

--scope <scope>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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-external-status-check

```
usage: gitlab project-external-status-check [-h]
                                            {list,create,update,delete} ...
```

-h, --help
show this help message and exit

gitlab project-external-status-check create

```
usage: gitlab project-external-status-check create [-h] [--sudo SUDO]
                                                   --project-id PROJECT_ID
                                                   --name NAME
                                                   --external-url EXTERNAL_URL
                                                   [--shared-secret SHARED_SECRET]
                                                   [--protected-branch-ids PROTECTED_
↪BRANCH_IDS]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--name <name>

--external-url <external_url>

--shared-secret <shared_secret>

--protected-branch-ids <protected_branch_ids>

gitlab project-external-status-check delete

```
usage: gitlab project-external-status-check 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-external-status-check list

```
usage: gitlab project-external-status-check list [-h] [--sudo SUDO]
                                                 --project-id PROJECT_ID
                                                 [--page PAGE]
                                                 [--per-page PER_PAGE]
                                                 [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

gitlab project-external-status-check update

```
usage: gitlab project-external-status-check update [-h] [--sudo SUDO]
                                                   --project-id PROJECT_ID
                                                   --id ID [--name NAME]
                                                   [--external-url EXTERNAL_URL]
                                                   [--shared-secret SHARED_SECRET]
                                                   [--protected-branch-ids PROTECTED_
↪BRANCH_IDS]
```

-h, --help

show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

--name <name>

--external-url <external_url>

--shared-secret <shared_secret>

--protected-branch-ids <protected_branch_ids>

gitlab project-feature-flag

```
usage: gitlab project-feature-flag [-h] {list,get,create,update,delete} ...
```

-h, --help

show this help message and exit

gitlab project-feature-flag create

```
usage: gitlab project-feature-flag create [-h] [--sudo SUDO]
                                          --project-id PROJECT_ID --name NAME
                                          [--version VERSION]
                                          [--description DESCRIPTION]
                                          [--active ACTIVE]
                                          [--strategies STRATEGIES]
```

-h, --help

show this help message and exit

--sudo <sudo>

--project-id <project_id>

--name <name>

--version <version>

--description <description>

--active <active>

--strategies <strategies>

gitlab project-feature-flag delete

```
usage: gitlab project-feature-flag 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-feature-flag get

```
usage: gitlab project-feature-flag 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-feature-flag list

```
usage: gitlab project-feature-flag list [-h] [--sudo SUDO]
                                        --project-id PROJECT_ID
                                        [--scope SCOPE] [--page PAGE]
                                        [--per-page PER_PAGE] [--get-all |
                                        --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--scope <scope>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

gitlab project-feature-flag update

```
usage: gitlab project-feature-flag update [-h] [--sudo SUDO]
                                          --project-id PROJECT_ID --name NAME
                                          [--new-name NEW_NAME]
                                          [--description DESCRIPTION]
                                          [--active ACTIVE]
                                          [--strategies STRATEGIES]
```

-h, --help

show this help message and exit

--sudo <sudo>

--project-id <project_id>

--name <name>

--new-name <new_name>

--description <description>

--active <active>

--strategies <strategies>

gitlab project-feature-flag-user-list

```
usage: gitlab project-feature-flag-user-list [-h]
                                             {list,get,create,update,delete} ...
```

-h, --help

show this help message and exit

gitlab project-feature-flag-user-list create

```
usage: gitlab project-feature-flag-user-list create [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --name NAME
                                                    --user-xids USER_XIDS
```

-h, --help

show this help message and exit

--sudo <sudo>

--project-id <project_id>

--name <name>

--user-xids <user_xids>

gitlab project-feature-flag-user-list delete

```
usage: gitlab project-feature-flag-user-list 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-feature-flag-user-list get

```
usage: gitlab project-feature-flag-user-list 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-feature-flag-user-list list

```
usage: gitlab project-feature-flag-user-list list [-h] [--sudo SUDO]
                                                  --project-id PROJECT_ID
                                                  [--search SEARCH]
                                                  [--page PAGE]
                                                  [--per-page PER_PAGE]
                                                  [--get-all | --no-get-all]
```

-h, **--help**
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--search <search>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

gitlab project-feature-flag-user-list update

```
usage: gitlab project-feature-flag-user-list update [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --iid IID [--name NAME]
                                                    [--user-xids USER_XIDS]
```

-h, --help

show this help message and exit

--sudo <sudo>

--project-id <project_id>

--iid <iid>

--name <name>

--user-xids <user_xids>

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]
                                  [--execute-filemode EXECUTE_FILEMODE]
                                  [--start-branch START_BRANCH]
```

-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>

--execute-filemode <execute_filemode>

--start-branch <start_branch>

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]
                                  [--execute-filemode EXECUTE_FILEMODE]
                                  [--start-branch START_BRANCH]
                                  [--last-commit-id LAST_COMMIT_ID]
```

-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>

--execute-filemode <execute_filemode>

--start-branch <start_branch>

--last-commit-id <last_commit_id>

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--archived <archived>

--visibility <visibility>

--order-by <order_by>

--sort <sort>

--search <search>

--simple <simple>

--owned <owned>

--membership <membership>

--starred <starred>

--statistics <statistics>

--with-custom-attributes <with_custom_attributes>

--with-issues-enabled <with_issues_enabled>

--with-merge-requests-enabled <with_merge_requests_enabled>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-gitignore-template

```
usage: gitlab project-gitignore-template [-h] {list,get} ...
```

-h, --help
show this help message and exit

gitlab project-gitignore-template get

```
usage: gitlab project-gitignore-template 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-gitignore-template list

```
usage: gitlab project-gitignore-template list [-h] [--sudo SUDO]
                                              --project-id PROJECT_ID
                                              [--page PAGE]
                                              [--per-page PER_PAGE]
                                              [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-gitlabciyml-template

```
usage: gitlab project-gitlabciyml-template [-h] {list,get} ...
```

-h, --help
show this help message and exit

gitlab project-gitlabciyml-template get

```
usage: gitlab project-gitlabciyml-template 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-gitlabciyml-template list

```
usage: gitlab project-gitlabciyml-template list [-h] [--sudo SUDO]
                                                --project-id PROJECT_ID
                                                [--page PAGE]
                                                [--per-page PER_PAGE]
                                                [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-hook

```
usage: gitlab project-hook [-h] {list,get,create,update,delete} ...
```

-h, --help
show this help message and exit

gitlab project-hook create

```
usage: gitlab project-hook create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                  --url URL [--push-events PUSH_EVENTS]
                                  [--issues-events ISSUES_EVENTS]
                                  [--confidential-issues-events CONFIDENTIAL_ISSUES_
↪EVENTS]
                                  [--merge-requests-events MERGE_REQUESTS_EVENTS]
                                  [--tag-push-events TAG_PUSH_EVENTS]
                                  [--note-events NOTE_EVENTS]
                                  [--job-events JOB_EVENTS]
                                  [--pipeline-events PIPELINE_EVENTS]
                                  [--wiki-page-events WIKI_PAGE_EVENTS]
                                  [--enable-ssl-verification ENABLE_SSL_VERIFICATION]
                                  [--token TOKEN]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--url <url>

--push-events <push_events>

--issues-events <issues_events>

--confidential-issues-events <confidential_issues_events>

--merge-requests-events <merge_requests_events>

--tag-push-events <tag_push_events>

--note-events <note_events>

--job-events <job_events>

--pipeline-events <pipeline_events>

--wiki-page-events <wiki_page_events>

--enable-ssl-verification <enable_ssl_verification>

--token <token>

gitlab project-hook delete

```
usage: gitlab project-hook delete [-h] [--sudo SUDO] --project-id PROJECT_ID
                                  --id ID
```

-h, **--help**
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

gitlab project-hook get

```
usage: gitlab project-hook get [-h] [--sudo SUDO] --project-id PROJECT_ID
                               --id ID
```

-h, **--help**
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

gitlab project-hook list

```
usage: gitlab project-hook list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                [--page PAGE] [--per-page PER_PAGE]
                                [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                       --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-integration update

```
usage: gitlab project-integration update [-h] [--sudo SUDO]
                                         --project-id PROJECT_ID --slug SLUG
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--slug <slug>

gitlab project-invitation

```
usage: gitlab project-invitation [-h] {list,get,create,update,delete} ...
```

-h, **--help**
show this help message and exit

gitlab project-invitation create

```
usage: gitlab project-invitation create [-h] [--sudo SUDO]
                                        --project-id PROJECT_ID
                                        --access-level ACCESS_LEVEL
                                        [--expires-at EXPIRES_AT]
                                        [--invite-source INVITE_SOURCE]
                                        [--tasks-to-be-done TASKS_TO_BE_DONE]
                                        [--tasks-project-id TASKS_PROJECT_ID]
                                        [--email EMAIL | --user-id USER_ID]
```

-h, **--help**
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--access-level <access_level>

--expires-at <expires_at>

--invite-source <invite_source>

--tasks-to-be-done <tasks_to_be_done>

--tasks-project-id <tasks_project_id>

--email <email>

--user-id <user_id>

gitlab project-invitation delete

```
usage: gitlab project-invitation delete [-h] [--sudo SUDO]
                                        --project-id PROJECT_ID --email EMAIL
```

-h, **--help**
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--email <email>

gitlab project-invitation get

```
usage: gitlab project-invitation get [-h] [--sudo SUDO]
                                     --project-id PROJECT_ID --email EMAIL
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--email <email>

gitlab project-invitation list

```
usage: gitlab project-invitation list [-h] [--sudo SUDO]
                                      --project-id PROJECT_ID [--query QUERY]
                                      [--page PAGE] [--per-page PER_PAGE]
                                      [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--query <query>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-invitation update

```
usage: gitlab project-invitation update [-h] [--sudo SUDO]
                                        --project-id PROJECT_ID --email EMAIL
                                        [--access-level ACCESS_LEVEL]
                                        [--expires-at EXPIRES_AT]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--email <email>

--access-level <access_level>

--expires-at <expires_at>

gitlab project-issue

```
usage: gitlab project-issue [-h]
                            {list,get,create,update,delete,user-agent-detail,subscribe,
↪unsubscribe,todo,time-stats,time-estimate,reset-time-estimate,add-spent-time,reset-
↪spent-time,participants,move,reorder,related-merge-requests,closed-by} ...
```

-h, --help
show this help message and exit

gitlab project-issue add-spent-time

```
usage: gitlab project-issue add-spent-time [-h] --project-id PROJECT_ID
                                           [--sudo SUDO] --iid IID
                                           --duration DURATION
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--iid <iid>

--duration <duration>

gitlab project-issue closed-by

```
usage: gitlab project-issue closed-by [-h] --project-id PROJECT_ID
                                      [--sudo SUDO] --iid IID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--iid <iid>

gitlab project-issue create

```
usage: gitlab project-issue create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --title TITLE [--description DESCRIPTION]
                                   [--confidential CONFIDENTIAL]
                                   [--assignee-ids ASSIGNEE_IDS]
                                   [--assignee-id ASSIGNEE_ID]
                                   [--milestone-id MILESTONE_ID]
                                   [--labels LABELS] [--created-at CREATED_AT]
                                   [--due-date DUE_DATE]
                                   [--merge-request-to-resolve-discussions-of MERGE_
↪REQUEST_TO_RESOLVE_DISCUSSIONS_OF]
                                   [--discussion-to-resolve DISCUSSION_TO_RESOLVE]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--title <title>

--description <description>

--confidential <confidential>

--assignee-ids <assignee_ids>

--assignee-id <assignee_id>

--milestone-id <milestone_id>

--labels <labels>

--created-at <created_at>

--due-date <due_date>

--merge-request-to-resolve-discussions-of <merge_request_to_resolve_discussions_of>

--discussion-to-resolve <discussion_to_resolve>

gitlab project-issue delete

```
usage: gitlab project-issue delete [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --iid IID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--iid <iid>

gitlab project-issue get

```
usage: gitlab project-issue get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                --iid IID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--iid <iid>

gitlab project-issue list

```
usage: gitlab project-issue list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                 [--iids IIDS] [--state STATE]
                                 [--labels LABELS] [--milestone MILESTONE]
                                 [--scope SCOPE] [--author-id AUTHOR_ID]
                                 [--iteration-id ITERATION_ID]
                                 [--assignee-id ASSIGNEE_ID]
                                 [--my-reaction-emoji MY_REACTION_EMOJI]
                                 [--order-by ORDER_BY] [--sort SORT]
                                 [--search SEARCH]
                                 [--created-after CREATED_AFTER]
                                 [--created-before CREATED_BEFORE]
                                 [--updated-after UPDATED_AFTER]
                                 [--updated-before UPDATED_BEFORE]
                                 [--page PAGE] [--per-page PER_PAGE]
                                 [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--iids <iids>

--state <state>

--labels <labels>

--milestone <milestone>

--scope <scope>

--author-id <author_id>

--iteration-id <iteration_id>

--assignee-id <assignee_id>

--my-reaction-emoji <my_reaction_emoji>

--order-by <order_by>

--sort <sort>

--search <search>

--created-after <created_after>

--created-before <created_before>

--updated-after <updated_after>

--updated-before <updated_before>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--issue-iid <issue_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                            --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--issue-iid <issue_iid>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

gitlab project-issue-discussion-note

```
usage: gitlab project-issue-discussion-note [-h]
                                            {list,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 list

```
usage: gitlab project-issue-discussion-note list [-h] [--sudo SUDO]
                                                 --project-id PROJECT_ID
                                                 --issue-iid ISSUE_IID
                                                 --discussion-id DISCUSSION_ID
                                                 [--page PAGE]
                                                 [--per-page PER_PAGE]
                                                 [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--issue-iid <issue_iid>

--discussion-id <discussion_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                      --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--issue-iid <issue_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                      --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--issue-iid <issue_iid>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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 | --no-get-all]
```

-h, **--help**
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--issue-iid <issue_iid>

--note-id <note_id>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

gitlab project-issue-resource-iteration-event

```
usage: gitlab project-issue-resource-iteration-event [-h] {list,get} ...
```

-h, --help

show this help message and exit

gitlab project-issue-resource-iteration-event get

```
usage: gitlab project-issue-resource-iteration-event get [-h] [--sudo SUDO]
                                                         --project-id PROJECT_ID
                                                         --issue-iid ISSUE_IID
                                                         --id ID
```

-h, --help

show this help message and exit

--sudo <sudo>

--project-id <project_id>

--issue-iid <issue_iid>

--id <id>

gitlab project-issue-resource-iteration-event list

```
usage: gitlab project-issue-resource-iteration-event list [-h] [--sudo SUDO]
                                                          --project-id PROJECT_ID
                                                          --issue-iid ISSUE_IID
                                                          [--page PAGE]
                                                          [--per-page PER_PAGE]
                                                          [--get-all |
                                                          --no-get-all]
```

-h, --help

show this help message and exit

--sudo <sudo>

--project-id <project_id>

--issue-iid <issue_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                                       --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--issue-iid <issue_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                                           --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--issue-iid <issue_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                                       --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--issue-iid <issue_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-issue-resource-weight-event

```
usage: gitlab project-issue-resource-weight-event [-h] {list,get} ...
```

-h, --help
show this help message and exit

gitlab project-issue-resource-weight-event get

```
usage: gitlab project-issue-resource-weight-event get [-h] [--sudo SUDO]
                                                      --project-id PROJECT_ID
                                                      --issue-iid ISSUE_IID
                                                      --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--issue-iid <issue_iid>

--id <id>

gitlab project-issue-resource-weight-event list

```
usage: gitlab project-issue-resource-weight-event list [-h] [--sudo SUDO]
                                                       --project-id PROJECT_ID
                                                       --issue-iid ISSUE_IID
                                                       [--page PAGE]
                                                       [--per-page PER_PAGE]
                                                       [--get-all |
                                                       --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--issue-iid <issue_iid>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

gitlab project-issue-template

```
usage: gitlab project-issue-template [-h] {list,get} ...
```

-h, --help

show this help message and exit

gitlab project-issue-template get

```
usage: gitlab project-issue-template 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-issue-template list

```
usage: gitlab project-issue-template list [-h] [--sudo SUDO]
                                          --project-id PROJECT_ID
                                          [--page PAGE] [--per-page PER_PAGE]
                                          [--get-all | --no-get-all]
```

-h, --help

show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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] [--order-by ORDER_BY]
                               [--sort SORT] [--page PAGE]
                               [--per-page PER_PAGE] [--get-all |
                               --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--scope <scope>

--order-by <order_by>

--sort <sort>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-job play

```
usage: gitlab project-job play [-h] --project-id PROJECT_ID [--sudo SUDO]
                               --id ID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--id <id>

gitlab project-job retry

```
usage: gitlab project-job retry [-h] --project-id PROJECT_ID [--sudo SUDO]
                                --id ID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--id <id>

gitlab project-job trace

```
usage: gitlab project-job trace [-h] --project-id PROJECT_ID [--sudo SUDO]
                                --id ID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--id <id>

gitlab project-job-token-scope

```
usage: gitlab project-job-token-scope [-h] {get,update} ...
```

-h, --help
show this help message and exit

gitlab project-job-token-scope get

```
usage: gitlab project-job-token-scope get [-h] [--sudo SUDO]
                                          --project-id PROJECT_ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

gitlab project-job-token-scope update

```
usage: gitlab project-job-token-scope update [-h] [--sudo SUDO]
                                             --project-id PROJECT_ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

gitlab project-key

```
usage: gitlab project-key [-h] {list,get,create,update,delete,enable} ...
```

-h, --help
show this help message and exit

gitlab project-key create

```
usage: gitlab project-key create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                 --title TITLE --key KEY [--can-push CAN_PUSH]
                                 [--expires-at EXPIRES_AT]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--title <title>

--key <key>

--can-push <can_push>

--expires-at <expires_at>

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]
                                 --key-id KEY_ID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-key update

```
usage: gitlab project-key update [-h] [--sudo SUDO] --project-id PROJECT_ID
                                 --id ID [--title TITLE] [--can-push CAN_PUSH]
                                 [--expires-at EXPIRES_AT]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

--title <title>

--can-push <can_push>

--expires-at <expires_at>

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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-license-template

```
usage: gitlab project-license-template [-h] {list,get} ...
```

-h, --help
show this help message and exit

gitlab project-license-template get

```
usage: gitlab project-license-template 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-license-template list

```
usage: gitlab project-license-template list [-h] [--sudo SUDO]
                                            --project-id PROJECT_ID
                                            [--page PAGE]
                                            [--per-page PER_PAGE] [--get-all |
                                            --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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
                                    [--expires-at EXPIRES_AT]
                                    [--tasks-to-be-done TASKS_TO_BE_DONE]
                                    [--username USERNAME | --user-id USER_ID]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--access-level <access_level>

--expires-at <expires_at>

--tasks-to-be-done <tasks_to_be_done>

--username <username>

--user-id <user_id>

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                      --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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,related-issues,closes-issues,commits,
↪changes,approve,unapprove,rebase,reset-approvals,merge-ref,merge} ...
```

-h, --help
show this help message and exit

gitlab project-merge-request add-spent-time

```
usage: gitlab project-merge-request add-spent-time [-h]
                                                   --project-id PROJECT_ID
                                                   [--sudo SUDO] --iid IID
                                                   --duration DURATION
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--iid <iid>

--duration <duration>

gitlab project-merge-request approve

```
usage: gitlab project-merge-request approve [-h] --project-id PROJECT_ID
                                            [--sudo SUDO] --iid IID
                                            [--sha SHA]
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--iid <iid>

--sha <sha>

gitlab project-merge-request cancel-merge-when-pipeline-succeeds

```
usage: gitlab project-merge-request cancel-merge-when-pipeline-succeeds
       [-h] --project-id PROJECT_ID [--sudo SUDO] --iid IID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--iid <iid>

gitlab project-merge-request changes

```
usage: gitlab project-merge-request changes [-h] --project-id PROJECT_ID
                                            [--sudo SUDO] --iid IID
                                            [--access-raw-diffs ACCESS_RAW_DIFFS]
```

-h, **--help**
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--iid <iid>

--access-raw-diffs <access_raw_diffs>

gitlab project-merge-request closes-issues

```
usage: gitlab project-merge-request closes-issues [-h] --project-id PROJECT_ID
                                                  [--sudo SUDO] --iid IID
```

-h, **--help**
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--iid <iid>

gitlab project-merge-request commits

```
usage: gitlab project-merge-request commits [-h] --project-id PROJECT_ID
                                            [--sudo SUDO] --iid IID
```

-h, **--help**
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--iid <iid>

gitlab project-merge-request create

```
usage: gitlab project-merge-request create [-h] [--sudo SUDO]
                                           --project-id PROJECT_ID
                                           --source-branch SOURCE_BRANCH
                                           --target-branch TARGET_BRANCH
```

(continues on next page)

(continued from previous page)

```
                                            --title TITLE
                                            [--allow-collaboration ALLOW_COLLABORATION]
                                            [--allow-maintainer-to-push ALLOW_MAINTAINER_
↪TO_PUSH]
                                            [--approvals-before-merge APPROVALS_BEFORE_
↪MERGE]
                                            [--assignee-id ASSIGNEE_ID]
                                            [--assignee-ids ASSIGNEE_IDS]
                                            [--description DESCRIPTION]
                                            [--labels LABELS]
                                            [--milestone-id MILESTONE_ID]
                                            [--remove-source-branch REMOVE_SOURCE_BRANCH]
                                            [--reviewer-ids REVIEWER_IDS]
                                            [--squash SQUASH]
                                            [--target-project-id TARGET_PROJECT_ID]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--source-branch <source_branch>

--target-branch <target_branch>

--title <title>

--allow-collaboration <allow_collaboration>

--allow-maintainer-to-push <allow_maintainer_to_push>

--approvals-before-merge <approvals_before_merge>

--assignee-id <assignee_id>

--assignee-ids <assignee_ids>

--description <description>

--labels <labels>

--milestone-id <milestone_id>

--remove-source-branch <remove_source_branch>

--reviewer-ids <reviewer_ids>

--squash <squash>

--target-project-id <target_project_id>

gitlab project-merge-request delete

```
usage: gitlab project-merge-request delete [-h] [--sudo SUDO]
                                           --project-id PROJECT_ID --iid IID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--iid <iid>

gitlab project-merge-request get

```
usage: gitlab project-merge-request get [-h] [--sudo SUDO]
                                        --project-id PROJECT_ID --iid IID
                                        [--render-html RENDER_HTML]
                                        [--include-diverged-commits-count INCLUDE_
↪DIVERGED_COMMITS_COUNT]
                                        [--include-rebase-in-progress INCLUDE_REBASE_IN_
↪PROGRESS]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--iid <iid>

--render-html <render_html>

--include-diverged-commits-count <include_diverged_commits_count>

--include-rebase-in-progress <include_rebase_in_progress>

gitlab project-merge-request list

```
usage: gitlab project-merge-request list [-h] [--sudo SUDO]
                                         --project-id PROJECT_ID
                                         [--state STATE] [--order-by ORDER_BY]
                                         [--sort SORT] [--milestone MILESTONE]
                                         [--view VIEW] [--labels LABELS]
                                         [--created-after CREATED_AFTER]
                                         [--created-before CREATED_BEFORE]
                                         [--updated-after UPDATED_AFTER]
                                         [--updated-before UPDATED_BEFORE]
                                         [--scope SCOPE] [--iids IIDS]
                                         [--author-id AUTHOR_ID]
                                         [--assignee-id ASSIGNEE_ID]
                                         [--approver-ids APPROVER_IDS]
                                         [--approved-by-ids APPROVED_BY_IDS]
```

(continues on next page)

(continued from previous page)

```
                                                [--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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--state <state>

--order-by <order_by>

--sort <sort>

--milestone <milestone>

--view <view>

--labels <labels>

--created-after <created_after>

--created-before <created_before>

--updated-after <updated_after>

--updated-before <updated_before>

--scope <scope>

--iids <iids>

--author-id <author_id>

--assignee-id <assignee_id>

--approver-ids <approver_ids>

--approved-by-ids <approved_by_ids>

--my-reaction-emoji <my_reaction_emoji>

--source-branch <source_branch>

--target-branch <target_branch>

--search <search>

--wip <wip>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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 related-issues

```
usage: gitlab project-merge-request related-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 reset-approvals

```
usage: gitlab project-merge-request reset-approvals [-h]
                                                    --project-id PROJECT_ID
                                                    [--sudo SUDO] --iid IID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--iid <iid>

gitlab project-merge-request reset-spent-time

```
usage: gitlab project-merge-request reset-spent-time [-h]
                                                     --project-id PROJECT_ID
                                                     [--sudo SUDO] --iid IID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--iid <iid>

gitlab project-merge-request reset-time-estimate

```
usage: gitlab project-merge-request reset-time-estimate [-h]
                                                        --project-id PROJECT_ID
                                                        [--sudo SUDO]
                                                        --iid IID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--iid <iid>

gitlab project-merge-request subscribe

```
usage: gitlab project-merge-request subscribe [-h] --project-id PROJECT_ID
                                              [--sudo SUDO] --iid IID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--iid <iid>

gitlab project-merge-request time-estimate

```
usage: gitlab project-merge-request time-estimate [-h] --project-id PROJECT_ID
                                                  [--sudo SUDO] --iid IID
                                                  --duration DURATION
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--iid <iid>

--duration <duration>

gitlab project-merge-request time-stats

```
usage: gitlab project-merge-request time-stats [-h] --project-id PROJECT_ID
                                               [--sudo SUDO] --iid IID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--iid <iid>

gitlab project-merge-request todo

```
usage: gitlab project-merge-request todo [-h] --project-id PROJECT_ID
                                         [--sudo SUDO] --iid IID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--iid <iid>

gitlab project-merge-request unapprove

```
usage: gitlab project-merge-request unapprove [-h] --project-id PROJECT_ID
                                              [--sudo SUDO] --iid IID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--iid <iid>

gitlab project-merge-request unsubscribe

```
usage: gitlab project-merge-request unsubscribe [-h] --project-id PROJECT_ID
                                                [--sudo SUDO] --iid IID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--iid <iid>

gitlab project-merge-request update

```
usage: gitlab project-merge-request update [-h] [--sudo SUDO]
                                           --project-id PROJECT_ID --iid IID
                                           [--target-branch TARGET_BRANCH]
                                           [--assignee-id ASSIGNEE_ID]
                                           [--title TITLE]
                                           [--description DESCRIPTION]
                                           [--state-event STATE_EVENT]
                                           [--labels LABELS]
                                           [--milestone-id MILESTONE_ID]
                                           [--remove-source-branch REMOVE_SOURCE_BRANCH]
                                           [--discussion-locked DISCUSSION_LOCKED]
                                           [--allow-maintainer-to-push ALLOW_MAINTAINER_
↪TO_PUSH]
                                           [--squash SQUASH]
                                           [--reviewer-ids REVIEWER_IDS]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--iid <iid>

--target-branch <target_branch>

--assignee-id <assignee_id>

--title <title>

--description <description>

--state-event <state_event>

--labels <labels>

--milestone-id <milestone_id>

--remove-source-branch <remove_source_branch>

--discussion-locked <discussion_locked>

--allow-maintainer-to-push <allow_maintainer_to_push>

--squash <squash>

--reviewer-ids <reviewer_ids>

gitlab project-merge-request-approval

```
usage: gitlab project-merge-request-approval [-h] {get,update} ...
```

-h, --help
show this help message and exit

gitlab project-merge-request-approval get

```
usage: gitlab project-merge-request-approval get [-h] [--sudo SUDO]
                                                 --project-id PROJECT_ID
                                                 --mr-iid MR_IID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

gitlab project-merge-request-approval update

```
usage: gitlab project-merge-request-approval update [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --mr-iid MR_IID
                                                    --approvals-required APPROVALS_
↪REQUIRED
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--approvals-required <approvals_required>

gitlab project-merge-request-approval-rule

```
usage: gitlab project-merge-request-approval-rule [-h]
                                                  {list,get,create,update,delete} ...
```

-h, --help
show this help message and exit

gitlab project-merge-request-approval-rule create

```
usage: gitlab project-merge-request-approval-rule create [-h] [--sudo SUDO]
                                                         --project-id PROJECT_ID
                                                         --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]
                                                         [--usernames USERNAMES]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_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>

--usernames <usernames>

gitlab project-merge-request-approval-rule delete

```
usage: gitlab project-merge-request-approval-rule delete [-h] [--sudo SUDO]
                                                          --project-id PROJECT_ID
                                                          --merge-request-iid MERGE_
↪REQUEST_IID
                                                          --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--merge-request-iid <merge_request_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
                                                       --merge-request-iid MERGE_REQUEST_
↪IID
                                                       --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--merge-request-iid <merge_request_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
                                                       --merge-request-iid MERGE_REQUEST_
↪IID
                                                       [--page PAGE]
                                                       [--per-page PER_PAGE]
                                                       [--get-all |
                                                       --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--merge-request-iid <merge_request_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-merge-request-approval-rule update

```
usage: gitlab project-merge-request-approval-rule update [-h] [--sudo SUDO]
                                                         --project-id PROJECT_ID
                                                         --id ID
                                                         --merge-request-iid MERGE_
↪REQUEST_IID
                                                         --name NAME
                                                         --approvals-required APPROVALS_
↪REQUIRED
                                                         [--user-ids USER_IDS]
                                                         [--group-ids GROUP_IDS]
                                                         [--usernames USERNAMES]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

--merge-request-iid <merge_request_iid>

--name <name>

--approvals-required <approvals_required>

--user-ids <user_ids>

--group-ids <group_ids>

--usernames <usernames>

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 |
                                                     --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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 | --no-get-all]
```

-h, --help

show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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]
                                                     {list,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 list

```
usage: gitlab project-merge-request-discussion-note list [-h] [--sudo SUDO]
                                                         --project-id PROJECT_ID
                                                         --mr-iid MR_IID
                                                         --discussion-id DISCUSSION_ID
                                                         [--page PAGE]
                                                         [--per-page PER_PAGE]
                                                         [--get-all |
                                                         --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--discussion-id <discussion_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-merge-request-discussion-note update

```
usage: gitlab project-merge-request-discussion-note update [-h] [--sudo SUDO]
                                                           --project-id PROJECT_ID
                                                           --mr-iid MR_IID
                                                           --discussion-id DISCUSSION_ID
                                                           --id ID --body BODY
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--discussion-id <discussion_id>

--id <id>

--body <body>

gitlab project-merge-request-draft-note

```
usage: gitlab project-merge-request-draft-note [-h]
                                               {list,get,create,update,delete} ...
```

-h, --help
show this help message and exit

gitlab project-merge-request-draft-note create

```
usage: gitlab project-merge-request-draft-note create [-h] [--sudo SUDO]
                                                      --project-id PROJECT_ID
                                                      --mr-iid MR_IID
                                                      --note NOTE
                                                      [--commit-id COMMIT_ID]
                                                      [--in-reply-to-discussion-id IN_
↪REPLY_TO_DISCUSSION_ID]
                                                      [--position POSITION]
                                                      [--resolve-discussion RESOLVE_
↪DISCUSSION]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--note <note>

--commit-id <commit_id>

--in-reply-to-discussion-id <in_reply_to_discussion_id>

--position <position>

--resolve-discussion <resolve_discussion>

gitlab project-merge-request-draft-note delete

```
usage: gitlab project-merge-request-draft-note delete [-h] [--sudo SUDO]
                                                      --project-id PROJECT_ID
                                                      --mr-iid MR_IID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--id <id>

gitlab project-merge-request-draft-note get

```
usage: gitlab project-merge-request-draft-note get [-h] [--sudo SUDO]
                                                   --project-id PROJECT_ID
                                                   --mr-iid MR_IID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--id <id>

gitlab project-merge-request-draft-note list

```
usage: gitlab project-merge-request-draft-note list [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    --mr-iid MR_IID
                                                    [--page PAGE]
                                                    [--per-page PER_PAGE]
                                                    [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-merge-request-draft-note update

```
usage: gitlab project-merge-request-draft-note update [-h] [--sudo SUDO]
                                                      --project-id PROJECT_ID
                                                      --mr-iid MR_IID --id ID
                                                      [--position POSITION]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--id <id>

--position <position>

gitlab project-merge-request-note

```
usage: gitlab project-merge-request-note [-h]
                                         {list,get,create,update,delete} ...
```

-h, **--help**
show this help message and exit

gitlab project-merge-request-note create

```
usage: gitlab project-merge-request-note create [-h] [--sudo SUDO]
                                                --project-id PROJECT_ID
                                                --mr-iid MR_IID --body BODY
```

-h, **--help**
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--body <body>

gitlab project-merge-request-note delete

```
usage: gitlab project-merge-request-note delete [-h] [--sudo SUDO]
                                                --project-id PROJECT_ID
                                                --mr-iid MR_IID --id ID
```

-h, **--help**
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--id <id>

gitlab project-merge-request-note get

```
usage: gitlab project-merge-request-note get [-h] [--sudo SUDO]
                                             --project-id PROJECT_ID
                                             --mr-iid MR_IID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--id <id>

gitlab project-merge-request-note list

```
usage: gitlab project-merge-request-note list [-h] [--sudo SUDO]
                                              --project-id PROJECT_ID
                                              --mr-iid MR_IID [--page PAGE]
                                              [--per-page PER_PAGE]
                                              [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                                          --no-get-all]
```

-h, **--help**
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--note-id <note_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-merge-request-reviewer-detail

```
usage: gitlab project-merge-request-reviewer-detail [-h] {list} ...
```

-h, --help
show this help message and exit

gitlab project-merge-request-reviewer-detail list

```
usage: gitlab project-merge-request-reviewer-detail list [-h] [--sudo SUDO]
                                                         --project-id PROJECT_ID
                                                         --mr-iid MR_IID
                                                         [--page PAGE]
                                                         [--per-page PER_PAGE]
                                                         [--get-all |
                                                         --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--mr-iid <mr_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-merge-request-status-check

```
usage: gitlab project-merge-request-status-check [-h] {list} ...
```

-h, --help
show this help message and exit

gitlab project-merge-request-status-check list

```
usage: gitlab project-merge-request-status-check list [-h] [--sudo SUDO]
                                                      --project-id PROJECT_ID
                                                      --merge-request-iid MERGE_REQUEST_
↪IID
                                                      [--page PAGE]
                                                      [--per-page PER_PAGE]
                                                      [--get-all |
                                                      --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--merge-request-iid <merge_request_iid>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-merge-request-template

```
usage: gitlab project-merge-request-template [-h] {list,get} ...
```

-h, --help
show this help message and exit

gitlab project-merge-request-template get

```
usage: gitlab project-merge-request-template 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-merge-request-template list

```
usage: gitlab project-merge-request-template list [-h] [--sudo SUDO]
                                                  --project-id PROJECT_ID
                                                  [--page PAGE]
                                                  [--per-page PER_PAGE]
                                                  [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--scope <scope>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--iids <iids>

--state <state>

--search <search>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-notification-settings

```
usage: gitlab project-notification-settings [-h] {get,update} ...
```

-h, --help
show this help message and exit

gitlab project-notification-settings get

```
usage: gitlab project-notification-settings get [-h] [--sudo SUDO]
                                                --project-id PROJECT_ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

gitlab project-notification-settings update

```
usage: gitlab project-notification-settings update [-h] [--sudo SUDO]
                                                   --project-id PROJECT_ID
                                                   [--level LEVEL]
                                                   [--notification-email NOTIFICATION_
↪EMAIL]
                                                   [--new-note NEW_NOTE]
                                                   [--new-issue NEW_ISSUE]
                                                   [--reopen-issue REOPEN_ISSUE]
                                                   [--close-issue CLOSE_ISSUE]
                                                   [--reassign-issue REASSIGN_ISSUE]
                                                   [--new-merge-request NEW_MERGE_
```

(continues on next page)

(continued from previous page)

```
↪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>

--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]
                                   [--package-version PACKAGE_VERSION]
                                   [--include-versionless INCLUDE_VERSIONLESS]
                                   [--status STATUS] [--page PAGE]
                                   [--per-page PER_PAGE] [--get-all |
                                   --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--order-by <order_by>

--sort <sort>

--package-type <package_type>

--package-name <package_name>

--package-version <package_version>

--include-versionless <include_versionless>

--status <status>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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 |
                                        --no-get-all]
```

-h, --help

show this help message and exit

--sudo <sudo>

--project-id <project_id>

--package-id <package_id>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

gitlab project-package-pipeline

```
usage: gitlab project-package-pipeline [-h] {list} ...
```

-h, --help
show this help message and exit

gitlab project-package-pipeline list

```
usage: gitlab project-package-pipeline list [-h] [--sudo SUDO]
                                            --project-id PROJECT_ID
                                            --package-id PACKAGE_ID
                                            [--page PAGE]
                                            [--per-page PER_PAGE] [--get-all |
                                            --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--package-id <package_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-package-protection-rule

```
usage: gitlab project-package-protection-rule [-h]
                                              {list,create,update,delete} ...
```

-h, --help
show this help message and exit

gitlab project-package-protection-rule create

```
usage: gitlab project-package-protection-rule create [-h] [--sudo SUDO]
                                                     --project-id PROJECT_ID
                                                     --package-name-pattern PACKAGE_NAME_
↪PATTERN
                                                     --package-type PACKAGE_TYPE
                                                     --minimum-access-level-for-push␣
↪MINIMUM_ACCESS_LEVEL_FOR_PUSH
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--package-name-pattern <package_name_pattern>

--package-type <package_type>

--minimum-access-level-for-push <minimum_access_level_for_push>

gitlab project-package-protection-rule delete

```
usage: gitlab project-package-protection-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-package-protection-rule list

```
usage: gitlab project-package-protection-rule list [-h] [--sudo SUDO]
                                                   --project-id PROJECT_ID
                                                   [--page PAGE]
                                                   [--per-page PER_PAGE]
                                                   [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-package-protection-rule update

```
usage: gitlab project-package-protection-rule update [-h] [--sudo SUDO]
                                                     --project-id PROJECT_ID
                                                     --id ID
                                                     [--package-name-pattern PACKAGE_
```

(continues on next page)

(continued from previous page)

```
↪NAME_PATTERN]
                                                    [--package-type PACKAGE_TYPE]
                                                    [--minimum-access-level-for-push␣
↪MINIMUM_ACCESS_LEVEL_FOR_PUSH]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

--package-name-pattern <package_name_pattern>

--package-type <package_type>

--minimum-access-level-for-push <minimum_access_level_for_push>

gitlab project-pages

```
usage: gitlab project-pages [-h] {get,update,delete} ...
```

-h, --help
show this help message and exit

gitlab project-pages delete

```
usage: gitlab project-pages delete [-h] [--sudo SUDO] --project-id PROJECT_ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

gitlab project-pages get

```
usage: gitlab project-pages get [-h] [--sudo SUDO] --project-id PROJECT_ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

gitlab project-pages update

```
usage: gitlab project-pages update [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   [--pages-unique-domain-enabled PAGES_UNIQUE_DOMAIN_
↪ENABLED]
                                   [--pages-https-only PAGES_HTTPS_ONLY]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--pages-unique-domain-enabled <pages_unique_domain_enabled>

--pages-https-only <pages_https_only>

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 |
                                        --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-pages-domain update

```
usage: gitlab project-pages-domain update [-h] [--sudo SUDO]
                                          --project-id PROJECT_ID
                                          --domain DOMAIN
                                          [--certificate CERTIFICATE]
                                          [--key KEY]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--domain <domain>

--certificate <certificate>

--key <key>

gitlab project-pipeline

```
usage: gitlab project-pipeline [-h] {list,get,create,delete,cancel,retry} ...
```

-h, --help
show this help message and exit

gitlab project-pipeline cancel

```
usage: gitlab project-pipeline cancel [-h] --project-id PROJECT_ID
                                      [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--id <id>

gitlab project-pipeline create

```
usage: gitlab project-pipeline create [-h] [--sudo SUDO]
                                      --project-id PROJECT_ID --ref REF
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--ref <ref>

gitlab project-pipeline delete

```
usage: gitlab project-pipeline delete [-h] [--sudo SUDO]
                                      --project-id PROJECT_ID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

gitlab project-pipeline get

```
usage: gitlab project-pipeline get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

gitlab project-pipeline list

```
usage: gitlab project-pipeline list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                    [--scope SCOPE] [--status STATUS]
                                    [--source SOURCE] [--ref REF] [--sha SHA]
                                    [--yaml-errors YAML_ERRORS] [--name NAME]
                                    [--username USERNAME]
                                    [--order-by ORDER_BY] [--sort SORT]
                                    [--page PAGE] [--per-page PER_PAGE]
                                    [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--scope <scope>

--status <status>

--source <source>

--ref <ref>

--sha <sha>

--yaml-errors <yaml_errors>

--name <name>

--username <username>

--order-by <order_by>

--sort <sort>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                           --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--pipeline-id <pipeline_id>

--scope <scope>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--pipeline-id <pipeline_id>

--scope <scope>

--include-retried <include_retried>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, **--help**
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

gitlab project-pipeline-schedule play

```
usage: gitlab project-pipeline-schedule play [-h] --project-id PROJECT_ID
                                             [--sudo SUDO] --id ID
```

-h, --help

show this help message and exit

--project-id <project_id>

--sudo <sudo>

--id <id>

gitlab project-pipeline-schedule take-ownership

```
usage: gitlab project-pipeline-schedule take-ownership [-h]
                                                       --project-id PROJECT_ID
                                                       [--sudo SUDO] --id ID
```

-h, --help

show this help message and exit

--project-id <project_id>

--sudo <sudo>

--id <id>

gitlab project-pipeline-schedule update

```
usage: gitlab project-pipeline-schedule update [-h] [--sudo SUDO]
                                               --project-id PROJECT_ID --id ID
                                               [--description DESCRIPTION]
                                               [--ref REF] [--cron CRON]
                                               [--cron-timezone CRON_TIMEZONE]
                                               [--active ACTIVE]
```

-h, --help

show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

--description <description>

--ref <ref>

--cron <cron>

--cron-timezone <cron_timezone>

--active <active>

gitlab project-pipeline-schedule-pipeline

```
usage: gitlab project-pipeline-schedule-pipeline [-h] {list} ...
```

-h, --help
show this help message and exit

gitlab project-pipeline-schedule-pipeline list

```
usage: gitlab project-pipeline-schedule-pipeline list [-h] [--sudo SUDO]
                                                       --project-id PROJECT_ID
                                                       --pipeline-schedule-id PIPELINE_
↪SCHEDULE_ID
                                                       [--page PAGE]
                                                       [--per-page PER_PAGE]
                                                       [--get-all |
                                                       --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--pipeline-schedule-id <pipeline_schedule_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--pipeline-id <pipeline_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-protected-branch

```
usage: gitlab project-protected-branch [-h]
                                       {list,get,create,update,delete} ...
```

-h, --help
show this help message and exit

gitlab project-protected-branch create

```
usage: gitlab project-protected-branch create [-h] [--sudo SUDO]
                                              --project-id PROJECT_ID
                                              --name NAME
                                              [--push-access-level PUSH_ACCESS_LEVEL]
                                              [--merge-access-level MERGE_ACCESS_LEVEL]
                                              [--unprotect-access-level UNPROTECT_ACCESS_
↪LEVEL]
                                              [--allow-force-push ALLOW_FORCE_PUSH]
                                              [--allowed-to-push ALLOWED_TO_PUSH]
                                              [--allowed-to-merge ALLOWED_TO_MERGE]
                                              [--allowed-to-unprotect ALLOWED_TO_
↪UNPROTECT]
                                              [--code-owner-approval-required CODE_OWNER_
↪APPROVAL_REQUIRED]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--name <name>

--push-access-level <push_access_level>

--merge-access-level <merge_access_level>

--unprotect-access-level <unprotect_access_level>

--allow-force-push <allow_force_push>

--allowed-to-push <allowed_to_push>

--allowed-to-merge <allowed_to_merge>

--allowed-to-unprotect <allowed_to_unprotect>

--code-owner-approval-required <code_owner_approval_required>

gitlab project-protected-branch delete

```
usage: gitlab project-protected-branch delete [-h] [--sudo SUDO]
                                              --project-id PROJECT_ID
                                              --name NAME
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--name <name>

gitlab project-protected-branch get

```
usage: gitlab project-protected-branch get [-h] [--sudo SUDO]
                                           --project-id PROJECT_ID --name NAME
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--name <name>

gitlab project-protected-branch list

```
usage: gitlab project-protected-branch list [-h] [--sudo SUDO]
                                            --project-id PROJECT_ID
                                            [--page PAGE]
                                            [--per-page PER_PAGE] [--get-all |
                                            --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-protected-branch update

```
usage: gitlab project-protected-branch update [-h] [--sudo SUDO]
                                              --project-id PROJECT_ID
                                              --name NAME
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--name <name>

gitlab project-protected-environment

```
usage: gitlab project-protected-environment [-h] {list,get,create,delete} ...
```

-h, --help
show this help message and exit

gitlab project-protected-environment create

```
usage: gitlab project-protected-environment create [-h] [--sudo SUDO]
                                                   --project-id PROJECT_ID
                                                   --name NAME
                                                   --deploy-access-levels DEPLOY_ACCESS_
↪LEVELS
                                                   [--required-approval-count REQUIRED_
↪APPROVAL_COUNT]
                                                   [--approval-rules APPROVAL_RULES]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--name <name>

--deploy-access-levels <deploy_access_levels>

--required-approval-count <required_approval_count>

--approval-rules <approval_rules>

gitlab project-protected-environment delete

```
usage: gitlab project-protected-environment delete [-h] [--sudo SUDO]
                                                   --project-id PROJECT_ID
                                                   --name NAME
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--name <name>

gitlab project-protected-environment get

```
usage: gitlab project-protected-environment get [-h] [--sudo SUDO]
                                                --project-id PROJECT_ID
                                                --name NAME
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--name <name>

gitlab project-protected-environment list

```
usage: gitlab project-protected-environment list [-h] [--sudo SUDO]
                                                 --project-id PROJECT_ID
                                                 [--page PAGE]
                                                 [--per-page PER_PAGE]
                                                 [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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 |
                                         --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-pull-mirror

```
usage: gitlab project-pull-mirror [-h] {get,create,update,start} ...
```

-h, --help
show this help message and exit

gitlab project-pull-mirror create

```
usage: gitlab project-pull-mirror create [-h] [--sudo SUDO]
                                         --project-id PROJECT_ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

gitlab project-pull-mirror get

```
usage: gitlab project-pull-mirror get [-h] [--sudo SUDO]
                                      --project-id PROJECT_ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

gitlab project-pull-mirror start

```
usage: gitlab project-pull-mirror start [-h] --project-id PROJECT_ID
                                        [--sudo SUDO]
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

gitlab project-pull-mirror update

```
usage: gitlab project-pull-mirror update [-h] [--sudo SUDO]
                                         --project-id PROJECT_ID [--url URL]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--url <url>

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_
↪COMMITS]
```

-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_
↪COMMITS]
```

-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-protection-rule

```
usage: gitlab project-registry-protection-rule [-h] {list,create,update} ...
```

-h, --help
show this help message and exit

gitlab project-registry-protection-rule create

```
usage: gitlab project-registry-protection-rule create [-h] [--sudo SUDO]
                                                      --project-id PROJECT_ID
                                                      --repository-path-pattern␣
↪REPOSITORY_PATH_PATTERN
                                                      [--minimum-access-level-for-push␣
↪MINIMUM_ACCESS_LEVEL_FOR_PUSH]
                                                      [--minimum-access-level-for-delete␣
↪MINIMUM_ACCESS_LEVEL_FOR_DELETE]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--repository-path-pattern <repository_path_pattern>

--minimum-access-level-for-push <minimum_access_level_for_push>

--minimum-access-level-for-delete <minimum_access_level_for_delete>

gitlab project-registry-protection-rule list

```
usage: gitlab project-registry-protection-rule list [-h] [--sudo SUDO]
                                                    --project-id PROJECT_ID
                                                    [--page PAGE]
                                                    [--per-page PER_PAGE]
                                                    [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-registry-protection-rule update

```
usage: gitlab project-registry-protection-rule update [-h] [--sudo SUDO]
                                                      --project-id PROJECT_ID
                                                      --id ID
                                                      [--repository-path-pattern␣
↪REPOSITORY_PATH_PATTERN]
                                                      [--minimum-access-level-for-push␣
```

(continues on next page)

(continued from previous page)

```
↪MINIMUM_ACCESS_LEVEL_FOR_PUSH]
                                                    [--minimum-access-level-for-delete␣
↪MINIMUM_ACCESS_LEVEL_FOR_DELETE]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

--repository-path-pattern <repository_path_pattern>

--minimum-access-level-for-push <minimum_access_level_for_push>

--minimum-access-level-for-delete <minimum_access_level_for_delete>

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

gitlab project-registry-repository-protection-rule

```
usage: gitlab project-registry-repository-protection-rule [-h]
                                                          {list,create,update,delete} ...
```

-h, --help

show this help message and exit

gitlab project-registry-repository-protection-rule create

```
usage: gitlab project-registry-repository-protection-rule create
       [-h] [--sudo SUDO] --project-id PROJECT_ID
       --repository-path-pattern REPOSITORY_PATH_PATTERN
       [--minimum-access-level-for-push MINIMUM_ACCESS_LEVEL_FOR_PUSH]
       [--minimum-access-level-for-delete MINIMUM_ACCESS_LEVEL_FOR_DELETE]
```

-h, --help

show this help message and exit

--sudo <sudo>

--project-id <project_id>

--repository-path-pattern <repository_path_pattern>

--minimum-access-level-for-push <minimum_access_level_for_push>

--minimum-access-level-for-delete <minimum_access_level_for_delete>

gitlab project-registry-repository-protection-rule delete

```
usage: gitlab project-registry-repository-protection-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-registry-repository-protection-rule list

```
usage: gitlab project-registry-repository-protection-rule list
       [-h] [--sudo SUDO] --project-id PROJECT_ID [--page PAGE]
       [--per-page PER_PAGE] [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-registry-repository-protection-rule update

```
usage: gitlab project-registry-repository-protection-rule update
       [-h] [--sudo SUDO] --project-id PROJECT_ID --id ID
       [--repository-path-pattern REPOSITORY_PATH_PATTERN]
       [--minimum-access-level-for-push MINIMUM_ACCESS_LEVEL_FOR_PUSH]
       [--minimum-access-level-for-delete MINIMUM_ACCESS_LEVEL_FOR_DELETE]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

--repository-path-pattern <repository_path_pattern>

--minimum-access-level-for-push <minimum_access_level_for_push>

--minimum-access-level-for-delete <minimum_access_level_for_delete>

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--repository-id <repository_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-release

```
usage: gitlab project-release [-h] {list,get,create,update,delete} ...
```

-h, --help
show this help message and exit

gitlab project-release create

```
usage: gitlab project-release create [-h] [--sudo SUDO]
                                     --project-id PROJECT_ID
                                     --tag-name TAG_NAME [--name NAME]
                                     [--description DESCRIPTION] [--ref REF]
                                     [--assets ASSETS]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--tag-name <tag_name>

--name <name>

--description <description>

--ref <ref>

--assets <assets>

gitlab project-release delete

```
usage: gitlab project-release delete [-h] [--sudo SUDO]
                                     --project-id PROJECT_ID
                                     --tag-name TAG_NAME
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--tag-name <tag_name>

gitlab project-release get

```
usage: gitlab project-release get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                  --tag-name TAG_NAME
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--tag-name <tag_name>

gitlab project-release list

```
usage: gitlab project-release list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   [--order-by ORDER_BY] [--sort SORT]
                                   [--include-html-description INCLUDE_HTML_DESCRIPTION]
                                   [--page PAGE] [--per-page PER_PAGE]
                                   [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--order-by <order_by>

--sort <sort>

--include-html-description <include_html_description>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-release update

```
usage: gitlab project-release update [-h] [--sudo SUDO]
                                     --project-id PROJECT_ID
                                     --tag-name TAG_NAME [--name NAME]
                                     [--description DESCRIPTION]
                                     [--milestones MILESTONES]
                                     [--released-at RELEASED_AT]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--tag-name <tag_name>

--name <name>

--description <description>

--milestones <milestones>

--released-at <released_at>

gitlab project-release-link

```
usage: gitlab project-release-link [-h] {list,get,create,update,delete} ...
```

-h, --help
show this help message and exit

gitlab project-release-link create

```
usage: gitlab project-release-link create [-h] [--sudo SUDO]
                                          --project-id PROJECT_ID
                                          --tag-name TAG_NAME --name NAME
                                          --url URL [--filepath FILEPATH]
                                          [--direct-asset-path DIRECT_ASSET_PATH]
                                          [--link-type LINK_TYPE]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--tag-name <tag_name>

--name <name>

--url <url>

--filepath <filepath>

--direct-asset-path <direct_asset_path>

--link-type <link_type>

gitlab project-release-link delete

```
usage: gitlab project-release-link delete [-h] [--sudo SUDO]
                                          --project-id PROJECT_ID
                                          --tag-name TAG_NAME --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--tag-name <tag_name>

--id <id>

gitlab project-release-link get

```
usage: gitlab project-release-link get [-h] [--sudo SUDO]
                                       --project-id PROJECT_ID
                                       --tag-name TAG_NAME --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--tag-name <tag_name>

--id <id>

gitlab project-release-link list

```
usage: gitlab project-release-link list [-h] [--sudo SUDO]
                                        --project-id PROJECT_ID
                                        --tag-name TAG_NAME [--page PAGE]
                                        [--per-page PER_PAGE] [--get-all |
                                        --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--tag-name <tag_name>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-release-link update

```
usage: gitlab project-release-link update [-h] [--sudo SUDO]
                                          --project-id PROJECT_ID
                                          --tag-name TAG_NAME --id ID
                                          [--name NAME] [--url URL]
                                          [--filepath FILEPATH]
                                          [--direct-asset-path DIRECT_ASSET_PATH]
                                          [--link-type LINK_TYPE]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--tag-name <tag_name>

--id <id>

--name <name>

--url <url>

--filepath <filepath>

--direct-asset-path <direct_asset_path>

--link-type <link_type>

gitlab project-remote-mirror

```
usage: gitlab project-remote-mirror [-h] {list,create,update,delete,sync} ...
```

-h, --help
show this help message and exit

gitlab project-remote-mirror create

```
usage: gitlab project-remote-mirror create [-h] [--sudo SUDO]
                                           --project-id PROJECT_ID --url URL
                                           [--enabled ENABLED]
                                           [--only-protected-branches ONLY_PROTECTED_
↪BRANCHES]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--url <url>

--enabled <enabled>

--only-protected-branches <only_protected_branches>

gitlab project-remote-mirror delete

```
usage: gitlab project-remote-mirror delete [-h] [--sudo SUDO]
                                           --project-id PROJECT_ID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

gitlab project-remote-mirror list

```
usage: gitlab project-remote-mirror list [-h] [--sudo SUDO]
                                         --project-id PROJECT_ID [--page PAGE]
                                         [--per-page PER_PAGE] [--get-all |
                                         --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

gitlab project-remote-mirror sync

```
usage: gitlab project-remote-mirror sync [-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-remote-mirror update

```
usage: gitlab project-remote-mirror update [-h] [--sudo SUDO]
                                           --project-id PROJECT_ID --id ID
                                           [--enabled ENABLED]
                                           [--only-protected-branches ONLY_PROTECTED_
↪BRANCHES]
```

-h, --help

show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

--enabled <enabled>

--only-protected-branches <only_protected_branches>

gitlab project-resource-group

```
usage: gitlab project-resource-group [-h] {list,get,update} ...
```

-h, --help

show this help message and exit

gitlab project-resource-group get

```
usage: gitlab project-resource-group get [-h] [--sudo SUDO]
                                         --project-id PROJECT_ID --key KEY
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--key <key>

gitlab project-resource-group list

```
usage: gitlab project-resource-group list [-h] [--sudo SUDO]
                                          --project-id PROJECT_ID
                                          [--order-by ORDER_BY] [--sort SORT]
                                          [--include-html-description INCLUDE_HTML_
↪DESCRIPTION]
                                          [--page PAGE] [--per-page PER_PAGE]
                                          [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--order-by <order_by>

--sort <sort>

--include-html-description <include_html_description>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-resource-group update

```
usage: gitlab project-resource-group update [-h] [--sudo SUDO]
                                            --project-id PROJECT_ID --key KEY
                                            [--process-mode PROCESS_MODE]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--key <key>

--process-mode <process_mode>

gitlab project-resource-group-upcoming-job

```
usage: gitlab project-resource-group-upcoming-job [-h] {list} ...
```

-h, --help
show this help message and exit

gitlab project-resource-group-upcoming-job list

```
usage: gitlab project-resource-group-upcoming-job list [-h] [--sudo SUDO]
                                                       --project-id PROJECT_ID
                                                       --resource-group-key RESOURCE_
↪GROUP_KEY
                                                       [--page PAGE]
                                                       [--per-page PER_PAGE]
                                                       [--get-all |
                                                       --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--resource-group-key <resource_group_key>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--scope <scope>

--tag-list <tag_list>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-secure-file

```
usage: gitlab project-secure-file [-h] {list,get,create,delete,download} ...
```

-h, --help
show this help message and exit

gitlab project-secure-file create

```
usage: gitlab project-secure-file create [-h] [--sudo SUDO]
                                         --project-id PROJECT_ID --name NAME
                                         --file FILE
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--name <name>

--file <file>

gitlab project-secure-file delete

```
usage: gitlab project-secure-file delete [-h] [--sudo SUDO]
                                         --project-id PROJECT_ID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

gitlab project-secure-file download

```
usage: gitlab project-secure-file download [-h] --project-id PROJECT_ID
                                           [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--id <id>

gitlab project-secure-file get

```
usage: gitlab project-secure-file get [-h] [--sudo SUDO]
                                      --project-id PROJECT_ID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

gitlab project-secure-file list

```
usage: gitlab project-secure-file list [-h] [--sudo SUDO]
                                       --project-id PROJECT_ID [--page PAGE]
                                       [--per-page PER_PAGE] [--get-all |
                                       --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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-service-account

```
usage: gitlab project-service-account [-h] {list,create,update,delete} ...
```

-h, --help
show this help message and exit

gitlab project-service-account create

```
usage: gitlab project-service-account create [-h] [--sudo SUDO]
                                             --project-id PROJECT_ID
                                             [--name NAME]
                                             [--username USERNAME]
                                             [--email EMAIL]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--name <name>

--username <username>

--email <email>

gitlab project-service-account delete

```
usage: gitlab project-service-account 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-service-account list

```
usage: gitlab project-service-account list [-h] [--sudo SUDO]
                                           --project-id PROJECT_ID
                                           [--order-by ORDER_BY] [--sort SORT]
                                           [--page PAGE] [--per-page PER_PAGE]
                                           [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--order-by <order_by>

--sort <sort>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-service-account update

```
usage: gitlab project-service-account update [-h] [--sudo SUDO]
                                             --project-id PROJECT_ID --id ID
                                             [--name NAME]
                                             [--username USERNAME]
                                             [--email EMAIL]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

--name <name>

--username <username>

--email <email>

gitlab project-service-account-access-token

```
usage: gitlab project-service-account-access-token [-h]
                                                   {list,create,delete,rotate} ...
```

-h, --help
show this help message and exit

gitlab project-service-account-access-token create

```
usage: gitlab project-service-account-access-token create [-h] [--sudo SUDO]
                                                          --project-id PROJECT_ID
                                                          --user-id USER_ID
                                                          --name NAME
                                                          --scopes SCOPES
                                                          [--description DESCRIPTION]
                                                          [--expires-at EXPIRES_AT]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--user-id <user_id>

--name <name>

--scopes <scopes>

--description <description>

--expires-at <expires_at>

gitlab project-service-account-access-token delete

```
usage: gitlab project-service-account-access-token delete [-h] [--sudo SUDO]
                                                          --project-id PROJECT_ID
                                                          --user-id USER_ID
                                                          --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--user-id <user_id>

--id <id>

gitlab project-service-account-access-token list

```
usage: gitlab project-service-account-access-token list [-h] [--sudo SUDO]
                                                        --project-id PROJECT_ID
                                                        --user-id USER_ID
                                                        [--created-after CREATED_AFTER]
                                                        [--created-before CREATED_BEFORE]
                                                        [--expires-after EXPIRES_AFTER]
                                                        [--expires-before EXPIRES_BEFORE]
                                                        [--last-used-after LAST_USED_
↪AFTER]
```

(continues on next page)

(continued from previous page)

```
                                                        [--last-used-before LAST_USED_
↪BEFORE]
                                                        [--revoked REVOKED]
                                                        [--search SEARCH]
                                                        [--sort SORT]
                                                        [--state STATE]
                                                        [--page PAGE]
                                                        [--per-page PER_PAGE]
                                                        [--get-all |
                                                        --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--user-id <user_id>

--created-after <created_after>

--created-before <created_before>

--expires-after <expires_after>

--expires-before <expires_before>

--last-used-after <last_used_after>

--last-used-before <last_used_before>

--revoked <revoked>

--search <search>

--sort <sort>

--state <state>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-service-account-access-token rotate

```
usage: gitlab project-service-account-access-token rotate [-h]
                                                          --project-id PROJECT_ID
                                                          --user-id USER_ID
                                                          [--sudo SUDO]
```

(continues on next page)

(continued from previous page)

```
                                                                --id ID
                                                                [--expires-at EXPIRES_AT]
```

-h, --help
show this help message and exit

--project-id <project_id>

--user-id <user_id>

--sudo <sudo>

--id <id>

--expires-at <expires_at>

gitlab project-snippet

```
usage: gitlab project-snippet [-h]
                              {list,get,create,update,delete,user-agent-detail,content} .
↪..
```

-h, --help
show this help message and exit

gitlab project-snippet content

```
usage: gitlab project-snippet content [-h] --project-id PROJECT_ID
                                      [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--project-id <project_id>

--sudo <sudo>

--id <id>

gitlab project-snippet create

```
usage: gitlab project-snippet create [-h] [--sudo SUDO]
                                     --project-id PROJECT_ID --title TITLE
                                     --visibility VISIBILITY
                                     [--description DESCRIPTION]
                                     [--content CONTENT] [--files FILES |
                                     --file-name FILE_NAME]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--title <title>

--visibility <visibility>

--description <description>

--content <content>

--files <files>

--file-name <file_name>

gitlab project-snippet delete

```
usage: gitlab project-snippet delete [-h] [--sudo SUDO]
                                     --project-id PROJECT_ID --id ID
```

-h, **--help**
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

gitlab project-snippet get

```
usage: gitlab project-snippet get [-h] [--sudo SUDO] --project-id PROJECT_ID
                                  --id ID
```

-h, **--help**
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

gitlab project-snippet list

```
usage: gitlab project-snippet list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                   [--page PAGE] [--per-page PER_PAGE]
                                   [--get-all | --no-get-all]
```

-h, **--help**
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

gitlab project-snippet update

```
usage: gitlab project-snippet update [-h] [--sudo SUDO]
                                     --project-id PROJECT_ID --id ID
                                     [--title TITLE] [--files FILES]
                                     [--file-name FILE_NAME]
                                     [--content CONTENT]
                                     [--visibility VISIBILITY]
                                     [--description DESCRIPTION]
```

-h, --help

show this help message and exit

--sudo <sudo>

--project-id <project_id>

--id <id>

--title <title>

--files <files>

--file-name <file_name>

--content <content>

--visibility <visibility>

--description <description>

gitlab project-snippet user-agent-detail

```
usage: gitlab project-snippet user-agent-detail [-h] --project-id PROJECT_ID
                                                [--sudo SUDO] --id ID
```

-h, --help

show this help message and exit

--project-id <project_id>

--sudo <sudo>

--id <id>

gitlab project-snippet-award-emoji

```
usage: gitlab project-snippet-award-emoji [-h] {list,get,create,delete} ...
```

-h, --help

show this help message and exit

gitlab project-snippet-award-emoji create

```
usage: gitlab project-snippet-award-emoji create [-h] [--sudo SUDO]
                                                 --project-id PROJECT_ID
                                                 --snippet-id SNIPPET_ID
                                                 --name NAME
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--snippet-id <snippet_id>

--name <name>

gitlab project-snippet-award-emoji delete

```
usage: gitlab project-snippet-award-emoji delete [-h] [--sudo SUDO]
                                                 --project-id PROJECT_ID
                                                 --snippet-id SNIPPET_ID
                                                 --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--snippet-id <snippet_id>

--id <id>

gitlab project-snippet-award-emoji get

```
usage: gitlab project-snippet-award-emoji get [-h] [--sudo SUDO]
                                              --project-id PROJECT_ID
                                              --snippet-id SNIPPET_ID --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--snippet-id <snippet_id>

--id <id>

gitlab project-snippet-award-emoji list

```
usage: gitlab project-snippet-award-emoji list [-h] [--sudo SUDO]
                                               --project-id PROJECT_ID
                                               --snippet-id SNIPPET_ID
                                               [--page PAGE]
                                               [--per-page PER_PAGE]
                                               [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--snippet-id <snippet_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--snippet-id <snippet_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                        --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--snippet-id <snippet_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--snippet-id <snippet_id>

--note-id <note_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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
                               [--order-by ORDER_BY] [--sort SORT]
                               [--search SEARCH] [--page PAGE]
                               [--per-page PER_PAGE] [--get-all |
                               --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--order-by <order_by>

--sort <sort>

--search <search>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--search <search>

--skip-users <skip_users>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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
                                      [--description DESCRIPTION]
                                      [--environment-scope ENVIRONMENT_SCOPE]
                                      [--masked MASKED]
                                      [--masked-and-hidden MASKED_AND_HIDDEN]
                                      [--protected PROTECTED] [--raw RAW]
                                      [--variable-type VARIABLE_TYPE]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--key <key>

--value <value>

--description <description>

--environment-scope <environment_scope>

--masked <masked>

--masked-and-hidden <masked_and_hidden>

--protected <protected>

--raw <raw>

--variable-type <variable_type>

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 | --no-get-all]
```

-h, **--help**
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab project-variable update

```
usage: gitlab project-variable update [-h] [--sudo SUDO]
                                      --project-id PROJECT_ID --key KEY
                                      [--description DESCRIPTION]
                                      [--environment-scope ENVIRONMENT_SCOPE]
                                      [--masked MASKED]
                                      [--protected PROTECTED] [--raw RAW]
                                      [--value VALUE]
                                      [--variable-type VARIABLE_TYPE]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--key <key>

--description <description>

--environment-scope <environment_scope>

--masked <masked>

--protected <protected>

--raw <raw>

--value <value>

--variable-type <variable_type>

gitlab project-wiki

```
usage: gitlab project-wiki [-h] {list,get,create,update,delete,upload} ...
```

-h, --help
show this help message and exit

gitlab project-wiki create

```
usage: gitlab project-wiki create [-h] [--sudo SUDO] --project-id PROJECT_ID
                                  --title TITLE --content CONTENT
                                  [--format FORMAT]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--title <title>

--content <content>

--format <format>

gitlab project-wiki delete

```
usage: gitlab project-wiki delete [-h] [--sudo SUDO] --project-id PROJECT_ID
                                  --slug SLUG
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--slug <slug>

gitlab project-wiki get

```
usage: gitlab project-wiki get [-h] [--sudo SUDO] --project-id PROJECT_ID
                               --slug SLUG
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--slug <slug>

gitlab project-wiki list

```
usage: gitlab project-wiki list [-h] [--sudo SUDO] --project-id PROJECT_ID
                                [--with-content WITH_CONTENT] [--page PAGE]
                                [--per-page PER_PAGE] [--get-all |
                                --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--project-id <project_id>

--with-content <with_content>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

gitlab project-wiki update

```
usage: gitlab project-wiki update [-h] [--sudo SUDO] --project-id PROJECT_ID
                                  --slug SLUG [--title TITLE]
                                  [--content CONTENT] [--format FORMAT]
```

-h, --help

show this help message and exit

--sudo <sudo>

--project-id <project_id>

--slug <slug>

--title <title>

--content <content>

--format <format>

gitlab project-wiki upload

```
usage: gitlab project-wiki upload [-h] --project-id PROJECT_ID [--sudo SUDO]
                                  --slug SLUG --filename FILENAME
                                  --filepath FILEPATH
```

-h, --help

show this help message and exit

--project-id <project_id>

--sudo <sudo>

--slug <slug>

--filename <filename>

--filepath <filepath>

gitlab registry-repository

```
usage: gitlab registry-repository [-h] {get} ...
```

-h, --help

show this help message and exit

gitlab registry-repository get

```
usage: gitlab registry-repository get [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab runner

```
usage: gitlab runner [-h] {list,get,create,update,delete,all,verify} ...
```

-h, --help
show this help message and exit

gitlab runner all

```
usage: gitlab runner all [-h] --id ID [--scope SCOPE]
```

-h, --help
show this help message and exit

--id <id>

--scope <scope>

gitlab runner create

```
usage: gitlab runner create [-h] [--sudo SUDO] --token TOKEN
                            [--description DESCRIPTION] [--info INFO]
                            [--active ACTIVE] [--locked LOCKED]
                            [--run-untagged RUN_UNTAGGED]
                            [--tag-list TAG_LIST]
                            [--access-level ACCESS_LEVEL]
                            [--maximum-timeout MAXIMUM_TIMEOUT]
```

-h, --help
show this help message and exit

--sudo <sudo>

--token <token>

--description <description>

--info <info>

--active <active>

--locked <locked>

--run-untagged <run_untagged>

--tag-list <tag_list>

--access-level <access_level>

--maximum-timeout <maximum_timeout>

gitlab runner delete

```
usage: gitlab runner delete [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab runner get

```
usage: gitlab runner get [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab runner list

```
usage: gitlab runner list [-h] [--sudo SUDO] [--scope SCOPE] [--type TYPE]
                          [--status STATUS] [--paused PAUSED]
                          [--tag-list TAG_LIST] [--page PAGE]
                          [--per-page PER_PAGE] [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--scope <scope>

--type <type>

--status <status>

--paused <paused>

--tag-list <tag_list>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--scope <scope>

--type <type>

--status <status>

--paused <paused>

--tag-list <tag_list>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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] [--order-by ORDER_BY]
                              [--sort SORT] [--page PAGE]
                              [--per-page PER_PAGE] [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--runner-id <runner_id>

--status <status>

--order-by <order_by>

--sort <sort>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab service-account

```
usage: gitlab service-account [-h] {list,create,update} ...
```

-h, --help
show this help message and exit

gitlab service-account create

```
usage: gitlab service-account create [-h] [--sudo SUDO] [--name NAME]
                                     [--username USERNAME] [--email EMAIL]
```

-h, --help
show this help message and exit

--sudo <sudo>

--name <name>

--username <username>

--email <email>

gitlab service-account list

```
usage: gitlab service-account list [-h] [--sudo SUDO] [--order-by ORDER_BY]
                                   [--sort SORT] [--page PAGE]
                                   [--per-page PER_PAGE] [--get-all |
                                   --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--order-by <order_by>

--sort <sort>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab service-account update

```
usage: gitlab service-account update [-h] [--sudo SUDO] --id ID [--name NAME]
                                     [--username USERNAME] [--email EMAIL]
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

--name <name>

--username <username>

--email <email>

gitlab shared-project

```
usage: gitlab shared-project [-h] {list} ...
```

-h, --help
show this help message and exit

gitlab shared-project list

```
usage: gitlab shared-project list [-h] [--sudo SUDO] --group-id GROUP_ID
                                  [--archived ARCHIVED]
                                  [--visibility VISIBILITY]
                                  [--order-by ORDER_BY] [--sort SORT]
                                  [--search SEARCH] [--simple SIMPLE]
                                  [--starred STARRED]
                                  [--with-issues-enabled WITH_ISSUES_ENABLED]
                                  [--with-merge-requests-enabled WITH_MERGE_REQUESTS_
↪ENABLED]
                                  [--min-access-level MIN_ACCESS_LEVEL]
                                  [--with-custom-attributes WITH_CUSTOM_ATTRIBUTES]
                                  [--page PAGE] [--per-page PER_PAGE]
                                  [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--group-id <group_id>

--archived <archived>

--visibility <visibility>

--order-by <order_by>

--sort <sort>

--search <search>

--simple <simple>

--starred <starred>

--with-issues-enabled <with_issues_enabled>

--with-merge-requests-enabled <with_merge_requests_enabled>

--min-access-level <min_access_level>

--with-custom-attributes <with_custom_attributes>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab snippet

```
usage: gitlab snippet [-h]
                      {list,get,create,update,delete,user-agent-detail,content,list-
↪public,list-all} ...
```

-h, --help
show this help message and exit

gitlab snippet content

```
usage: gitlab snippet content [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab snippet create

```
usage: gitlab snippet create [-h] [--sudo SUDO] --title TITLE
                             [--description DESCRIPTION] [--content CONTENT]
                             [--visibility VISIBILITY] [--files FILES |
                             --file-name FILE_NAME]
```

-h, --help
show this help message and exit

--sudo <sudo>

--title <title>

--description <description>

--content <content>

--visibility <visibility>

--files <files>

--file-name <file_name>

gitlab snippet delete

```
usage: gitlab snippet delete [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab snippet get

```
usage: gitlab snippet get [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab snippet list

```
usage: gitlab snippet list [-h] [--sudo SUDO] [--page PAGE]
                           [--per-page PER_PAGE] [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab snippet list-all

```
usage: gitlab snippet list-all [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab snippet list-public

```
usage: gitlab snippet list-public [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab snippet update

```
usage: gitlab snippet update [-h] [--sudo SUDO] --id ID [--title TITLE]
                             [--files FILES] [--file-name FILE_NAME]
                             [--content CONTENT] [--visibility VISIBILITY]
                             [--description DESCRIPTION]
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

--title <title>

--files <files>

--file-name <file_name>

--content <content>

--visibility <visibility>

--description <description>

gitlab snippet user-agent-detail

```
usage: gitlab snippet user-agent-detail [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab starred-project

```
usage: gitlab starred-project [-h] {list} ...
```

-h, --help
show this help message and exit

gitlab starred-project list

```
usage: gitlab starred-project list [-h] [--sudo SUDO] --user-id USER_ID
                                   [--archived ARCHIVED]
                                   [--membership MEMBERSHIP]
                                   [--min-access-level MIN_ACCESS_LEVEL]
                                   [--order-by ORDER_BY] [--owned OWNED]
                                   [--search SEARCH] [--simple SIMPLE]
                                   [--sort SORT] [--starred STARRED]
                                   [--statistics STATISTICS]
                                   [--visibility VISIBILITY]
                                   [--with-custom-attributes WITH_CUSTOM_ATTRIBUTES]
                                   [--with-issues-enabled WITH_ISSUES_ENABLED]
                                   [--with-merge-requests-enabled WITH_MERGE_REQUESTS_
↪ENABLED]
                                   [--page PAGE] [--per-page PER_PAGE]
                                   [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--user-id <user_id>

--archived <archived>

--membership <membership>

--min-access-level <min_access_level>

--order-by <order_by>

--owned <owned>

--search <search>

--simple <simple>

--sort <sort>

--starred <starred>

--statistics <statistics>

--visibility <visibility>

--with-custom-attributes <with_custom_attributes>

--with-issues-enabled <with_issues_enabled>

--with-merge-requests-enabled <with_merge_requests_enabled>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

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 | --no-get-all]
```

-h, --help

show this help message and exit

--sudo <sudo>

--action <action>

--author-id <author_id>

--project-id <project_id>

--state <state>

--type <type>

--page <page>

--per-page <per_page>

--get-all

Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

gitlab todo mark-all-as-done

```
usage: gitlab todo mark-all-as-done [-h] --id ID
```

-h, --help

show this help message and exit

--id <id>

gitlab todo mark-as-done

```
usage: gitlab todo mark-as-done [-h] --id ID
```

-h, --help

show this help message and exit

--id <id>

gitlab topic

```
usage: gitlab topic [-h] {list,get,create,update,delete,merge} ...
```

-h, --help

show this help message and exit

gitlab topic create

```
usage: gitlab topic create [-h] [--sudo SUDO] --name NAME [--avatar AVATAR]
                           [--description DESCRIPTION] [--title TITLE]
```

-h, --help

show this help message and exit

--sudo <sudo>

--name <name>

--avatar <avatar>

--description <description>

--title <title>

gitlab topic delete

```
usage: gitlab topic delete [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab topic get

```
usage: gitlab topic get [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab topic list

```
usage: gitlab topic list [-h] [--sudo SUDO] [--page PAGE]
                         [--per-page PER_PAGE] [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab topic merge

```
usage: gitlab topic merge [-h] --id ID --source-topic-id SOURCE_TOPIC_ID
                          --target-topic-id TARGET_TOPIC_ID
```

-h, --help
show this help message and exit

--id <id>

--source-topic-id <source_topic_id>

--target-topic-id <target_topic_id>

gitlab topic update

```
usage: gitlab topic update [-h] [--sudo SUDO] --id ID [--avatar AVATAR]
                           [--description DESCRIPTION] [--name NAME]
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

--avatar <avatar>

--description <description>

--name <name>

gitlab user

```
usage: gitlab user [-h]
                   {list,get,create,update,delete,block,follow,unfollow,unblock,
↪deactivate,activate,approve,reject,ban,unban} ...
```

-h, --help
show this help message and exit

gitlab user activate

```
usage: gitlab user activate [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab user approve

```
usage: gitlab user approve [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab user ban

```
usage: gitlab user ban [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab user block

```
usage: gitlab user block [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab user create

```
usage: gitlab user create [-h] [--sudo SUDO] [--email EMAIL]
                          [--username USERNAME] [--name NAME]
                          [--password PASSWORD]
                          [--reset-password RESET_PASSWORD] [--skype SKYPE]
                          [--linkedin LINKEDIN] [--twitter TWITTER]
                          [--projects-limit PROJECTS_LIMIT]
                          [--extern-uid EXTERN_UID] [--provider PROVIDER]
                          [--bio BIO] [--admin ADMIN]
                          [--can-create-group CAN_CREATE_GROUP]
                          [--website-url WEBSITE_URL]
                          [--skip-confirmation SKIP_CONFIRMATION]
                          [--external EXTERNAL] [--organization ORGANIZATION]
                          [--location LOCATION] [--avatar AVATAR]
                          [--public-email PUBLIC_EMAIL]
                          [--private-profile PRIVATE_PROFILE]
                          [--color-scheme-id COLOR_SCHEME_ID]
                          [--theme-id THEME_ID]
```

-h, --help
show this help message and exit

--sudo <sudo>

--email <email>

--username <username>

--name <name>

--password <password>

--reset-password <reset_password>

--skype <skype>

--linkedin <linkedin>

--twitter <twitter>

--projects-limit <projects_limit>

--extern-uid <extern_uid>

--provider <provider>

--bio <bio>

--admin <admin>

--can-create-group <can_create_group>

--website-url <website_url>

--skip-confirmation <skip_confirmation>

--external <external>

--organization <organization>

--location <location>

--avatar <avatar>

--public-email <public_email>

--private-profile <private_profile>

--color-scheme-id <color_scheme_id>

--theme-id <theme_id>

gitlab user deactivate

```
usage: gitlab user deactivate [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab user delete

```
usage: gitlab user delete [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab user follow

```
usage: gitlab user follow [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab user get

```
usage: gitlab user get [-h] [--sudo SUDO] --id ID
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

gitlab user list

```
usage: gitlab user list [-h] [--sudo SUDO] [--username USERNAME]
                        [--public-email PUBLIC_EMAIL] [--search SEARCH]
                        [--active ACTIVE] [--external EXTERNAL]
                        [--blocked BLOCKED] [--humans HUMANS]
                        [--created-after CREATED_AFTER]
                        [--created-before CREATED_BEFORE]
                        [--exclude-active EXCLUDE_ACTIVE]
                        [--exclude-external EXCLUDE_EXTERNAL]
                        [--exclude-humans EXCLUDE_HUMANS]
                        [--exclude-internal EXCLUDE_INTERNAL]
                        [--without-project-bots WITHOUT_PROJECT_BOTS]
                        [--extern-uid EXTERN_UID] [--provider PROVIDER]
                        [--two-factor TWO_FACTOR]
                        [--without-projects WITHOUT_PROJECTS]
                        [--admins ADMINS] [--auditors AUDITORS]
                        [--skip-ldap SKIP_LDAP]
                        [--custom-attributes CUSTOM_ATTRIBUTES]
                        [--status STATUS] [--page PAGE] [--per-page PER_PAGE]
                        [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--username <username>

--public-email <public_email>

--search <search>

--active <active>

--external <external>

--blocked <blocked>

--humans <humans>

--created-after <created_after>

--created-before <created_before>

--exclude-active <exclude_active>

--exclude-external <exclude_external>

--exclude-humans <exclude_humans>

--exclude-internal <exclude_internal>

--without-project-bots <without_project_bots>

--extern-uid <extern_uid>

--provider <provider>

--two-factor <two_factor>

--without-projects <without_projects>

--admins <admins>

--auditors <auditors>

--skip-ldap <skip_ldap>

--custom-attributes <custom_attributes>

--status <status>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab user reject

```
usage: gitlab user reject [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab user unban

```
usage: gitlab user unban [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab user unblock

```
usage: gitlab user unblock [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab user unfollow

```
usage: gitlab user unfollow [-h] --id ID
```

-h, --help
show this help message and exit

--id <id>

gitlab user update

```
usage: gitlab user update [-h] [--sudo SUDO] --id ID --email EMAIL
                          --username USERNAME --name NAME
                          [--password PASSWORD] [--skype SKYPE]
                          [--linkedin LINKEDIN] [--twitter TWITTER]
                          [--projects-limit PROJECTS_LIMIT]
                          [--extern-uid EXTERN_UID] [--provider PROVIDER]
                          [--bio BIO] [--admin ADMIN]
                          [--can-create-group CAN_CREATE_GROUP]
                          [--website-url WEBSITE_URL]
                          [--skip-reconfirmation SKIP_RECONFIRMATION]
                          [--external EXTERNAL] [--organization ORGANIZATION]
                          [--location LOCATION] [--avatar AVATAR]
                          [--public-email PUBLIC_EMAIL]
                          [--private-profile PRIVATE_PROFILE]
                          [--color-scheme-id COLOR_SCHEME_ID]
                          [--theme-id THEME_ID]
```

-h, --help
show this help message and exit

--sudo <sudo>

--id <id>

--email <email>

--username <username>

--name <name>

--password <password>

--skype <skype>

--linkedin <linkedin>

--twitter <twitter>

--projects-limit <projects_limit>

--extern-uid <extern_uid>

--provider <provider>

--bio <bio>

--admin <admin>

--can-create-group <can_create_group>

--website-url <website_url>

--skip-reconfirmation <skip_reconfirmation>

--external <external>

--organization <organization>

--location <location>

--avatar <avatar>

--public-email <public_email>

--private-profile <private_profile>

--color-scheme-id <color_scheme_id>

--theme-id <theme_id>

gitlab user-activities

```
usage: gitlab user-activities [-h] {list} ...
```

-h, --help
show this help message and exit

gitlab user-activities list

```
usage: gitlab user-activities list [-h] [--sudo SUDO] [--page PAGE]
                                   [--per-page PER_PAGE] [--get-all |
                                   --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab user-contributed-project

```
usage: gitlab user-contributed-project [-h] {list} ...
```

-h, --help
show this help message and exit

gitlab user-contributed-project list

```
usage: gitlab user-contributed-project list [-h] [--sudo SUDO]
                                            --user-id USER_ID [--page PAGE]
                                            [--per-page PER_PAGE] [--get-all |
                                            --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--user-id <user_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--user-id <user_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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
                                [--skip-confirmation SKIP_CONFIRMATION]
```

-h, --help
show this help message and exit

--sudo <sudo>

--user-id <user_id>

--email <email>

--skip-confirmation <skip_confirmation>

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 |
                              --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--user-id <user_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

gitlab user-event

```
usage: gitlab user-event [-h] {list} ...
```

-h, --help
show this help message and exit

gitlab user-event list

```
usage: gitlab user-event list [-h] [--sudo SUDO] --user-id USER_ID
                              [--action ACTION] [--target-type TARGET_TYPE]
                              [--before BEFORE] [--after AFTER] [--sort SORT]
                              [--scope SCOPE] [--page PAGE]
                              [--per-page PER_PAGE] [--get-all | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--user-id <user_id>

--action <action>

--target-type <target_type>

--before <before>

--after <after>

--sort <sort>

--scope <scope>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--user-id <user_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                            --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--user-id <user_id>

--state <state>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--user-id <user_id>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 |
                                   --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--user-id <user_id>

--type <type>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--user-id <user_id>

--archived <archived>

--visibility <visibility>

--order-by <order_by>

--sort <sort>

--search <search>

--simple <simple>

--owned <owned>

--membership <membership>

--starred <starred>

--statistics <statistics>

--with-issues-enabled <with_issues_enabled>

--with-merge-requests-enabled <with_merge_requests_enabled>

--with-custom-attributes <with_custom_attributes>

--with-programming-language <with_programming_language>

--wiki-checksum-failed <wiki_checksum_failed>

--repository-checksum-failed <repository_checksum_failed>

--min-access-level <min_access_level>

--id-after <id_after>

--id-before <id_before>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all
Don't return all items from the server.

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
                              [--description DESCRIPTION] [--masked MASKED]
                              [--protected PROTECTED] [--raw RAW]
                              [--variable-type VARIABLE_TYPE]
```

-h, --help
show this help message and exit

--sudo <sudo>

--key <key>

--value <value>

--description <description>

--masked <masked>

--protected <protected>

--raw <raw>

--variable-type <variable_type>

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 | --no-get-all]
```

-h, --help
show this help message and exit

--sudo <sudo>

--page <page>

--per-page <per_page>

--get-all
Return all items from the server, without pagination.

--no-get-all

Don't return all items from the server.

gitlab variable update

```
usage: gitlab variable update [-h] [--sudo SUDO] --key KEY
                              [--description DESCRIPTION] [--masked MASKED]
                              [--protected PROTECTED] [--raw RAW]
                              [--value VALUE] [--variable-type VARIABLE_TYPE]
```

-h, --help

show this help message and exit

--sudo <sudo>

--key <key>

--description <description>

--masked <masked>

--protected <protected>

--raw <raw>

--value <value>

--variable-type <variable_type>

7.9 Lower-level APIs

python-gitlab's API levels provide different degrees of convenience, control and stability.

7.9.1 Main interface - Gitlab, managers and objects

As shown in previous sections and examples, the high-level API interface wraps GitLab's API endpoints and makes them available from the `Gitlab` instance via managers that create objects you can manipulate.

This is what most users will want to use, as it covers most of GitLab's API endpoints, and allows you to write idiomatic Python code when interacting with the API.

7.9.2 Lower-level API - HTTP methods

Danger

At this point, python-gitlab will no longer take care of URL-encoding and other transformations needed to correctly pass API parameter types. You have to construct these correctly yourself.

However, you still benefit from many of the client's *Features* such as authentication, requests and retry handling.

Important

If you've found yourself at this section because of an endpoint not yet implemented in the library - please consider opening a pull request implementing the resource or at least filing an issue so we can track progress.

High-quality pull requests for standard endpoints that pass CI and include unit tests and documentation are easy to review, and can land quickly with monthly releases. If you ask, we can also trigger a new release, so you and everyone benefits from the contribution right away!

Managers and objects call specific HTTP methods to fetch or send data to the server. These methods can be invoked directly to access endpoints not currently implemented by the client. This essentially gives you some level of usability for any endpoint the moment it is available on your GitLab instance.

These methods can be accessed directly via the `Gitlab` instance (e.g. `gl.http_get()`), or via an object's manager (e.g. `project.manager.gitlab.http_get()`), if the `Gitlab` instance is not available in the current context.

For example, if you'd like to access GitLab's undocumented latest pipeline endpoint, you can do so by calling `http_get()` with the path to the endpoint:

```
>>> gl = gitlab.Gitlab(private_token=private_token)
>>>
>>> pipeline = gl.http_get("/projects/gitlab-org%2Fgitlab/pipelines/latest")
>>> pipeline["id"]
449070256
```

The available methods are:

- `http_get()`
- `http_post()`
- `http_put()`
- `http_patch()`
- `http_delete()`
- `http_list()` (a wrapper around `http_get` handling pagination, including with lazy generators)
- `http_head()` (only returns the header dictionary)

7.9.3 Lower-lower-level API - HTTP requests

Important

This is mostly intended for internal use in python-gitlab and may have a less stable interface than higher-level APIs. To lessen the chances of a change to the interface impacting your code, we recommend using keyword arguments when calling the interfaces.

At the lowest level, HTTP methods call `http_request()`, which performs the actual request and takes care of details such as timeouts, retries, and handling rate-limits.

This method can be invoked directly to or customize this behavior for a single request, or to call custom HTTP methods not currently implemented in the library - while still making use of all of the client's options and authentication methods.

For example, if for whatever reason you want to fetch allowed methods for an endpoint at runtime:

```
>>> gl = gitlab.Gitlab(private_token=private_token)
>>>
>>> response = gl.http_request(verb="OPTIONS", path="/projects")
>>> response.headers["Allow"]
'OPTIONS, GET, POST, HEAD'
```

Or get the total number of a user's events with a customized HEAD request:

```
>>> response = gl.http_request(
        verb="HEAD",
        path="/events",
        query_params={"sudo": "some-user"},
        timeout=10
    )
>>> response.headers["X-Total"]
'123'
```

7.10 CHANGELOG

All versions below are listed in reverse chronological order.

7.10.1 v8.4.0 (2026-05-28)

Features

- **const**: Add new Security Manager role (3738bb2)

Testing

- **const**: Add tests for AccessLevel (2ab6d9f)

7.10.2 v8.3.0 (2026-04-28)

Bug Fixes

- **api**: Update CI/CD variable create and update attributes (849301b)
- **mixins**: Register service account token classes in RotateMixin CLI actions (2cf430b)

Features

- **api**: Add full service account support at instance, group, and project level (feb6ced)

7.10.3 v8.2.0 (2026-03-28)

Documentation

- **testing**: Document passing pytest options during local development (e6669f9)

Features

- **api**: Add support for project feature flags and feature flag user lists (be68285)
- **projects**: Add optional parameter to set approval rule on all protected branches. (8d76028)

7.10.4 v8.1.0 (2026-02-28)

Bug Fixes

- **types**: Add explicit submodule import for pyright compatibility (84ad3fd)

Features

- **api**: Add support for order_by filter in runner jobs (d5dc73d)
- **api**: Add support for sort filter in runner jobs (b117ee3)

7.10.5 v8.0.0 (2026-01-28)

Bug Fixes

- Actually define project repr_attr (4187a69)
- File save start_branch as a body attribute (1001d93)

Chores

- **black**: Run black v26 on code (4a8d82b)
- **precommit**: Update dependency black to v26 (ad43b76)

Features

- **graphql**: Update to gql 4.0.0 (6f0da67)

Breaking Changes

- **graphql**: GraphQL.execute() no longer accepts graphql.Source

7.10.6 v7.1.0 (2025-12-28)

Bug Fixes

- **utils**: Prevent negative sleep time in rate limit retry (4221195)

Continuous Integration

- **release**: Use the correct token for publish to GitHub (614a74c)

Features

- **registry-protection**: Add support for registry protection rule deletion (9dd62c3)

Testing

- **functional**: Update to GitLab 18.6 and resolve issues found (c7c139b)

7.10.7 v7.0.0 (2025-10-29)

Features

- Drop Python 3.9 support and add Python 3.14 (22941ac)

Breaking Changes

- As of python-gitlab 7.0.0, Python 3.9 is no longer supported. Python 3.10 or higher is required.

7.10.8 v6.5.0 (2025-10-17)

Bug Fixes

- **semantic-release**: Enable CHANGELOG.md generation (`fb9693b`)

Continuous Integration

- **stale**: Fix permission for stale action and allow manual run (`9357a37`)

Documentation

- **pull_mirror**: Fix incorrect start() method usage example (`2acac19`)

Features

- **api**: Add content_ref and dry_run_ref parameters to ProjectCiLintManager (`e8d2538`)
- **users**: Implement 'skip_confirmation' in users 'emails' creation (`2981730`)

7.10.9 v6.4.0 (2025-09-28)

Features

- **users**: Implement missing arguments in users 'list' (`99923d4`)
- **users**: Sort 'user list' arguments against documentation (`99923d4`)

7.10.10 v6.3.0 (2025-08-28)

Features

- Add sync method to force remote mirror updates (`f3c6678`)
- **api**: Add missing ProjectJob list filters (`5fe0e71`)
- **api**: Add missing ProjectPackageManager list filters (`b1696be`)
- **users**: Implement support for 'admins' in administrators 'list' (`aaed51c`)

7.10.11 v6.2.0 (2025-07-28)

Build System

- **release**: Use correct python-semantic-release/publish-action (`2f20634`)

Continuous Integration

- **stale**: Improve formatting of stale message (`0ef20d1`)
- **stale**: Increase `operations-per-run` to 500 (`326e1a4`)

Features

- **api**: Add ListMixin to ProjectIssueDiscussionNoteManager (f908f0e)
- **api**: Add ListMixin to ProjectMergeRequestDiscussionNoteManager (865339a)

7.10.12 v6.1.0 (2025-06-28)

Chores

- Update to mypy 1.16.0 and resolve issues found (f734c58)

Documentation

- Update CONTRIBUTING.rst with policy on issue management (45dda50)

Features

- **api**: Add listing user contributed projects (98c1307)
- **api**: Add support for project tag list filters (378a836)
- **api**: Pipeline inputs support (#3194, 306c4b1)
- **const**: Add PLANNER_ACCESS constant (ba6f174)
- **groups**: Add protectedbranches to group class (#3164, bfd31a8)

7.10.13 v6.0.0 (2025-06-04)

Chores

- Add reformat code commit to .git-blame-ignore-revs (a6ac939)
- Reformat code with skip_magic_trailing_comma = true (2100aa4)
- Remove trivial get methods in preparation for generic Get mixin (edd01a5)
- Upgrade to sphinx 8.2.1 and resolve issues (d0b5ae3)
- **ci**: Replace docs artifact with readthedocs previews (193c5de)

Documentation

- Use get_all keyword arg instead of all in docstrings (f62dda7)
- Use list(get_all=True) in documentation examples (f36614f)
- **api-usage**: Fix GitLab API links to the publicly accessible URLs (f55fa15)
- **api-usage-graphql**: Fix the example graphql query string (8dbdd7e)
- **job_token_scope**: Fix typo/inconsistency (203bd92)

Features

- Adds member role methods (055557e)
- **api**: Add iteration_id as boards create attribute (#3191, 938b0d9)
- **api**: Add support for adding instance deploy keys (22be96c)
- **api**: Add support for avatar removal (5edd2e6)
- **api**: Add support for token self-rotation (da40e09)

- **api**: ListMixin.list typing overload (6eee494)
- **api**: Make RESTManager generic on RESTObject class (91c4f18)
- **api**: Make RESTObjectList typing generic (befba35)
- **settings**: Implement support for 'silent_mode_enabled' (a9163a9)

Refactoring

- Use more python3.9 syntax (4e90c11)

Testing

- **functional**: Switch to new runner registration API (cbc613d)

7.10.14 v5.6.0 (2025-01-28)

Features

- **group**: Add support for group level MR approval rules (304bdd0)

7.10.15 v5.5.0 (2025-01-28)

Chores

- Add deprecation warning for mirror_pull functions (7f6fd5c)
- Relax typing constraints for response action (f430078)
- **tests**: Catch deprecation warnings (0c1af08)

Documentation

- Add usage of pull mirror (9b374b2)
- Remove old pull mirror implementation (9e18672)

Features

- **functional**: Add pull mirror test (3b31ade)
- **projects**: Add pull mirror class (2411bff)
- **unit**: Add pull mirror tests (5c11203)

7.10.16 v5.4.0 (2025-01-28)

Bug Fixes

- **api**: Make type ignores more specific where possible (e3cb806)

Instead of using absolute ignore `# type: ignore` use a more specific ignores like `# type: ignore[override]`. This might help in the future where a new bug might be introduced and get ignored by a general ignore comment but not a more specific one.

Signed-off-by: Igor Ponomarev igor.ponomarev@collabora.com

- **api**: Return the new commit when calling cherry_pick (de29503)
- **files**: Add optional ref parameter for cli project-file raw (python-gitlab#3032) (22f03bd)

The ef parameter was removed in python-gitlab v4.8.0. This will add ef back as an optional parameter for the project-file raw cli command.

Chores

- Fix missing space in deprecation message (ba75c31)
- Fix pytest deprecation (95db680)

pytest has changed the function argument name to `start_path`

- Fix warning being generated (0eb5eb0)

The CI shows a warning. Use `get_all=False` to resolve issue.

- Resolve DeprecationWarning message in CI run (accd5aa)

Catch the DeprecationWarning in our test, as we expect it.

- **ci**: Set a 30 minute timeout for 'functional' tests (e8d6953)

Currently the functional API test takes around 17 minutes to run. And the functional CLI test takes around 12 minutes to run.

Occasionally a job gets stuck and will sit until the default 360 minutes job timeout occurs.

Now have a 30 minute timeout for the 'functional' tests.

- **deps**: Update all non-major dependencies (939505b)
- **deps**: Update all non-major dependencies (cbd4263)
- **deps**: Update gitlab (#3088, 9214b83)

Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com>

- **deps**: Update gitlab/gitlab-ee docker tag to v17.7.1-ee.0 (#3082, 1e95944)

Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com>

- **deps**: Update mypy to 1.14 and resolve issues (671e711)

mypy 1.14 has a change to Enum Membership Semantics: https://mypy.readthedocs.io/en/latest/changelog.html

Resolve the issues with Enum and typing, and update mypy to 1.14

- **test**: Prevent 'job_with_artifact' fixture running forever (e4673d8)

Previously the 'job_with_artifact' fixture could run forever. Now give it up to 60 seconds to complete before failing.

Continuous Integration

- Use gitlab-runner:v17.7.1 for the CI (2dda9dc)

The `latest` gitlab-runner image does not have the `gitlab-runner` user and it causes our tests to fail.

Closes: #3091

Features

- **api**: Add argument that appends extra HTTP headers to a request (fb07b5c)

Currently the only way to manipulate the headers for a request is to use `Gitlab.headers` attribute. However, this makes it very concurrently unsafe because the `Gitlab` object can be shared between multiple requests at the same time.

Instead add a new keyword argument `extra_headers` which will update the headers dictionary with new values just before the request is sent.

For example, this can be used to download a part of a artifacts file using the `Range` header: https://developer.mozilla.org/en-US/docs/Web/HTTP/Range_requests

Signed-off-by: Igor Ponomarev igor.ponomarev@collabora.com

- **api**: Add support for external status check (`175b355`)
- **api**: Narrow down return type of download methods using typing.overload (`44fd9dc`)

Currently the download methods such as `ProjectJob.artifacts` have return type set to `Optional[Union[bytes, Iterator[Any]]]` which means they return either `None` or `bytes` or `Iterator[Any]`.

However, the actual return type is determined by the passed `streamed` and `iterator` arguments. Using `@typing.overload` decorator it is possible to return a single type based on the passed arguments.

Add overloads in the following order to all download methods:

1. If `streamed=False` and `iterator=False` return `bytes`. This is the default argument values therefore it should be first as it will be used to lookup default arguments. 2. If `iterator=True` return `Iterator[Any]`. This can be combined with both `streamed=True` and `streamed=False`. 3. If `streamed=True` and `iterator=False` return `None`. In this case `action` argument can be set to a callable that accepts `bytes`.

Signed-off-by: Igor Ponomarev igor.ponomarev@collabora.com

- **api**: Narrow down return type of ProjectFileManager.raw using typing.overload (`36d9b24`)

This is equivalent to the changes in 44fd9dc1176a2c5529c45cc3186c0e775026175e but for `ProjectFileManager.raw` method that I must have missed in the original commit.

Signed-off-by: Igor Ponomarev igor.ponomarev@collabora.com

7.10.17 v5.3.1 (2025-01-07)

Bug Fixes

- **api**: Allow configuration of keep_base_url from file (`f4f7d7a`)
- **registry-protection**: Fix api url (`8c1aaa3`)

See: https://docs.gitlab.com/ee/api/container_repository_protection_rules.html#list-container-repository-protection-rules

Chores

- Bump to 5.3.1 (`912e1a0`)
- **deps**: Update dependency jinja2 to v3.1.5 [security] (`01d4194`)

7.10.18 v5.3.0 (2024-12-28)

Chores

- **deps**: Update gitlab/gitlab-ee docker tag to v17.7.0-ee.0 (#3070, `62b7eb7`)

Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com>

- **renovate**: Update httpx and respx again (`aa07449`)

Features

- **api**: Support the new registry protection rule endpoint (40af1c8)

7.10.19 v5.2.0 (2024-12-17)

Chores

- **deps**: Update all non-major dependencies (1e02f23)
- **deps**: Update all non-major dependencies (6532e8c)
- **deps**: Update all non-major dependencies (8046387)
- **deps**: Update codecov/codecov-action action to v5 (735efff)
- **deps**: Update dependency commitizen to v4 (9306362)
- **deps**: Update gitlab/gitlab-ee docker tag to v17.6.1-ee.0 (#3053, f2992ae)

Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com>

- **deps**: Update gitlab/gitlab-ee docker tag to v17.6.2-ee.0 (#3065, db0db26)

Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com>

- **deps**: Update pre-commit hook commitizen-tools/commitizen to v4 (a8518f1)
- **docs**: Fix CHANGELOG tracebacks codeblocks (9fe372a)

With v5.1.0 CHANGELOG.md was updated that mangled v1.10.0 triple backtick codeblock Traceback output that made sphinx fail [1] with a non-zero return code.

The resulting docs appears to be processes as text after the failing line [2]. While reviewing other backtick codeblocks fix v1.8.0 [3] to the original traceback.

[1] https://github.com/python-gitlab/python-gitlab/actions/runs/12060608158/job/33631303063#step:5:204 [2] https://python-gitlab.readthedocs.io/en/v5.1.0/changelog.html#v1-10-0-2019-07-22 [3] https://python-gitlab.readthedocs.io/en/v5.0.0/changelog.html#id258

- **renovate**: Pin httpx until respx is fixed (b70830d)

Documentation

- **api-usage**: Fix link to Gitlab REST API Authentication Docs (#3059, f460d95)

Features

- **api**: Add project templates (#3057, 0d41da3)
- feat(api): Added project template classes to templates.py * feat(api): Added project template managers to Project in project.py * docs(merge_requests): Add example of creating mr with description template * test(templates): Added unit tests for templates * docs(templates): added section for project templates
- **graphql**: Add async client (288f39c)

7.10.20 v5.1.0 (2024-11-28)

Chores

- **deps**: Update all non-major dependencies (9061647)
- **deps**: Update all non-major dependencies (62da12a)
- **deps**: Update all non-major dependencies (7e62136)

- **deps**: Update all non-major dependencies (d4b52e7)
- **deps**: Update all non-major dependencies (541a7e3)
- **deps**: Update dependency pytest-cov to v6 (ffa88b3)
- **deps**: Update gitlab/gitlab-ee docker tag to v17.5.1-ee.0 (8111f49)
- **deps**: Update gitlab/gitlab-ee docker tag to v17.5.2-ee.0 (#3041, d39129b)

Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com>

- **deps**: Update gitlab/gitlab-ee docker tag to v17.6.0-ee.0 (#3044, 79113d9)

Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com>

- **deps**: Update pre-commit hook maxbrunet/pre-commit-renovate to v39 (11458e0)

Features

- **api**: Get single project approval rule (029695d)
- **api**: Support list and delete for group service accounts (#2963, 499243b)
- **cli**: Enable token rotation via CLI (0cb8171)
- **const**: Add new Planner role to access levels (bdc8852)
- **files**: Add support for more optional flags (f51cd52)

GitLab's Repository Files API supports additional flags that weren't implemented before. Notably, the "start_branch" flag is particularly useful, as previously one had to use the "project-branch" command alongside "project-file" to add a file on a separate branch.

[1] https://docs.gitlab.com/ee/api/repository_files.html

7.10.21 v5.0.0 (2024-10-28)

Bug Fixes

- **api**: Set _repr_attr for project approval rules to name attr (#3011, 1a68f1c)

Co-authored-by: Patrick Evans patrick.evans@gehealthcare.com

Chores

- Add Python 3.13 as supported (#3012, b565e78)

Mark that Python 3.13 is supported.

Use Python 3.13 for the Mac and Windows tests.

Also remove the 'py38' tox environment. We no longer support Python 3.8.

- Add testing of Python 3.14 (14d2a82)

Also fix **annotations** not working in Python 3.14 by using the annotation on the 'class' instead of on the 'instance'

Closes: #3013

- Remove "v3" question from issue template (#3017, 482f2fe)

python-gitlab hasn't supported the GitLab v3 API since 2018. The last version of python-gitlab to support it was v1.4

Support was removed in:

commit fe89b949922c028830dd49095432ba627d330186 Author: Gauvain Pocentek gauvain@pocentek.net

Date: Sat May 19 17:10:08 2018 +0200

Drop API v3 support

Drop the code, the tests, and update the documentation.

- **deps**: Update all non-major dependencies (1e4326b)
- **deps**: Update all non-major dependencies (b3834dc)
- **deps**: Update dependency ubuntu to v24 (6fda15d)
- **deps**: Update gitlab/gitlab-ee docker tag to v17.4.2-ee.0 (1cdfe40)
- **deps**: Update gitlab/gitlab-ee docker tag to v17.5.0-ee.0 (c02a392)

Documentation

- **users**: Update Gitlab docs links (#3022, 3739b5d)

Features

- Remove support for Python 3.8, require 3.9 or higher (#3005, 9734ad4)

Python 3.8 is End-of-Life (EOL) as of 2024-10 as stated in https://devguide.python.org/versions/ and https://peps.python.org/pep-0569/#lifespan

By dropping support for Python 3.8 and requiring Python 3.9 or higher it allows python-gitlab to take advantage of new features in Python 3.9, which are documented at: https://docs.python.org/3/whatsnew/3.9.html

Closes: #2968

BREAKING CHANGE: As of python-gitlab 5.0.0, Python 3.8 is no longer supported. Python 3.9 or higher is required.

Testing

- Add test for `to_json()` method (f4bfe19)

This should get us to 100% test coverage on `gitlab/base.py`

BREAKING CHANGES

- As of python-gitlab 5.0.0, Python 3.8 is no longer supported. Python 3.9 or higher is required.

7.10.22 v4.13.0 (2024-10-08)

Chores

- **deps**: Update all non-major dependencies (c3efb37)
- **deps**: Update dependency pre-commit to v4 (#3008, 5c27546)

Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com>

Features

- **api**: Add support for project Pages API (0ee0e02)

7.10.23 v4.12.2 (2024-10-01)

Bug Fixes

- Raise GitlabHeadError in `project.files.head()` method (#3006, 9bf26df)

When an error occurs, raise `GitlabHeadError` in `project.files.head()` method.

Closes: #3004

7.10.24 v4.12.1 (2024-09-30)

Bug Fixes

- **ci**: Do not rely on GitLab.com runner arch variables (#3003, c848d12)
- **files**: Correctly raise GitlabGetError in get method (190ec89)

Chores

- **deps**: Update all non-major dependencies (#3000, d3da326)

Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com>

- **deps**: Update gitlab/gitlab-ee docker tag to v17.4.1-ee.0 (64eed5d)

7.10.25 v4.12.0 (2024-09-28)

Bug Fixes

- **api**: Head requests for projectfilemanager (#2977, 96a18b0)
- fix(api): head requests for projectfilemanager

Co-authored-by: Patrick Evans patrick.evans@gehealthcare.com

Co-authored-by: Nejc Habjan hab.nejc@gmail.com

Chores

- Update pylint to 3.3.1 and resolve issues (#2997, a0729b8)

pylint 3.3.1 appears to have added "too-many-positional-arguments" check with a value of 5.

I don't disagree with this, but we have many functions which exceed this value. We might think about converting some of positional arguments over to keyword arguments in the future. But that is for another time.

For now disable the check across the project.

- **deps**: Update all non-major dependencies (ae132e7)
- **deps**: Update all non-major dependencies (10ee58a)
- **deps**: Update dependency types-setuptools to v75 (a2ab54c)
- **deps**: Update gitlab/gitlab-ee docker tag to v17.3.2-ee.0 (5cd1ab2)
- **deps**: Update gitlab/gitlab-ee docker tag to v17.4.0-ee.0 (8601808)

Features

- Introduce related_issues to merge requests (#2996, 174d992)
- **build**: Build multi-arch images (#2987, 29f617d)

7.10.26 v4.11.1 (2024-09-13)

Bug Fixes

- **client**: Ensure type evaluations are postponed (b41b2de)

7.10.27 v4.11.0 (2024-09-13)

Chores

- **deps**: Update all non-major dependencies (fac8bf9)
- **deps**: Update all non-major dependencies (88c7529)
- **deps**: Update dependency types-setuptools to v74 (bdfaddb)
- **pre-commit**: Add deps (fe5e608)

Documentation

- **objects**: Fix typo in get latest pipeline (b9f5c12)

Features

- Add a minimal GraphQL client (d6b1b0a)
- **api**: Add exclusive GET attrs for /groups/:id/members (d44ddd2)
- **api**: Add exclusive GET attrs for /projects/:id/members (e637808)
- **client**: Add retry handling to GraphQL client (8898c38)
- **client**: Make retries configurable in GraphQL (145870e)

Refactoring

- **client**: Move retry logic into utility (3235c48)

7.10.28 v4.10.0 (2024-08-28)

Chores

- **deps**: Update all non-major dependencies (2ade0d9)
- **deps**: Update all non-major dependencies (0578bf0)
- **deps**: Update all non-major dependencies (31786a6)
- **deps**: Update dependency myst-parser to v4 (930d4a2)
- **deps**: Update dependency sphinx to v8 (cb65ffb)
- **deps**: Update dependency types-setuptools to v73 (d55c045)
- **deps**: Update gitlab/gitlab-ee docker tag to v17.2.2-ee.0 (b2275f7)
- **deps**: Update gitlab/gitlab-ee docker tag to v17.3.0-ee.0 (e5a46f5)
- **deps**: Update gitlab/gitlab-ee docker tag to v17.3.1-ee.0 (3fdd130)

- **deps**: Update python-semantic-release/upload-to-gh-release digest to 17c75b7 (12caaa4)
- **release**: Track tags for renovate (d600444)

Documentation

- **faq**: Correct the attribute fetching example (43a16ac)

There is an example about object attributes in the FAQ. It shows how to properly fetch all attributes of all projects, by using list() followed by a get(id) call.

Unfortunately this example used a wrong variable name, which caused it not to work and which could have made it slightly confusing to readers. This commit fixes that, by changing the variable name.

Now the example uses one variable for two Python objects. As they correspond to the same GitLab object and the intended behavior is to obtain that very object, just with all attributes, this is fine and is probably what readers will find most useful in this context.

Features

- **api**: Project/group hook test triggering (9353f54)

Add the ability to trigger tests of project and group hooks.

Fixes #2924

Testing

- **cli**: Allow up to 30 seconds for a project export (bdc155b)

Before we allowed a maximum of around 15 seconds for the project-export. Often times the CI was failing with this value.

Change it to a maximum of around 30 seconds.

7.10.29 v4.9.0 (2024-08-06)

Chores

- **ci**: Make pre-commit check happy (67370d8)

pre-commit incorrectly wants double back-quotes inside the code section. Rather than fight it, just use single quotes.

- **deps**: Update all non-major dependencies (f95ca26)
- **deps**: Update all non-major dependencies (7adc86b)
- **deps**: Update all non-major dependencies (e820db0)
- **deps**: Update dependency types-setuptools to v71 (d6a7dba)
- **deps**: Update gitlab/gitlab-ee docker tag to v17.2.1-ee.0 (d13a656)
- **deps**: Update pre-commit hook maxbrunet/pre-commit-renovate to v38 (f13968b)
- **deps**: Update python-semantic-release/upload-to-gh-release digest to 0dcddac (eb5c6f7)
- **deps**: Update python-semantic-release/upload-to-gh-release digest to e2355e1 (eb18552)

Features

- **snippets**: Add support for listing all instance snippets (64ae61e)

7.10.30 v4.8.0 (2024-07-16)

Bug Fixes

- Have `participants()` method use `http_list()` (d065275)

Previously it was using `http_get()` but the `participants` API returns a list of participants. Also by using this then we will warn if only a subset of the participants are returned.

Closes: #2913

- Issues `closed_by()/related_merge_requests()` use `http_list` (de2e4dd)

The `closed_by()` and `related_merge_requests()` API calls return lists. So use the `http_list()` method.

This will also warn the user if only a subset of the data is returned.

- **cli**: Generate UserWarning if `list` does not return all entries (e5a4379)

Previously in the CLI, calls to `list()` would have `get_all=False` by default. Therefore hiding the fact that not all items are being returned if there were more than 20 items.

Added `--no-get-all` option to `list` actions. Along with the already existing `--get-all`.

Closes: #2900

- **files**: Cr: add explicit comparison to `None` (51d8f88)

Co-authored-by: Nejc Habjan hab.nejc@gmail.com

- **files**: Make `ref` parameter optional in get raw file api (00640ac)

The `ref` parameter was made optional in gitlab v13.11.0.

Chores

- Add `show_caller` argument to `utils.warn()` (7d04315)

This allows us to not add the caller's location to the UserWarning message.

- Use correct type-hint for `die()` (9358640)
- **ci**: Specify name of "stale" label (44f62c4)

Saw the following error in the log: [#2618] Removing the label "Stale" from this issue… ##[error][#2618] Error when removing the label: "Label does not exist"

My theory is that the case doesn't match ("Stale" != "stale") and that is why it failed. Our label is "stale" so update this to match. Thought of changing the label name on GitHub but then would also require a change here to the "any-of-labels". So it seemed simpler to just change it here.

It is confusing though that it detected the label "stale", but then couldn't delete it.

- **ci**: Stale: allow issues/PRs that have stale label to be closed (2ab88b2)

If a `stale` label is manually applied, allow the issue or PR to be closed by the stale job.

Previously it would require the `stale` label and to also have one of 'need info' or 'Waiting for response' labels added.

- **ci**: Use codecov token when available (b74a6fb)
- **deps**: Update all non-major dependencies (4a2b213)
- **deps**: Update all non-major dependencies (0f59069)

- **deps**: Update all non-major dependencies (cf87226)
- **deps**: Update gitlab/gitlab-ee docker tag to v17.1.1-ee.0 (5e98510)
- **deps**: Update gitlab/gitlab-ee docker tag to v17.1.2-ee.0 (6fedfa5)
- **deps**: Update python-semantic-release/upload-to-gh-release digest to c7c3b69 (23393fa)
- **deps**: Update python-semantic-release/upload-to-gh-release digest to fe6cc89 (3f3ad80)

Documentation

- Document how to use `sudo` if modifying an object (d509da6)

Add a warning about using `sudo` when saving.

Give an example of how to `get` an object, modify it, and then `save` it using `sudo`

Closes: #532

- Variables: add note about `filter` for updating (c378817)

Add a note about using `filter` when updating a variable.

Closes: #2835

Closes: #1387

Closes: #1125

Features

- **api**: Add support for commit sequence (1f97be2)
- **api**: Add support for container registry protection rules (6d31649)
- **api**: Add support for package protection rules (6b37811)
- **api**: Add support for project cluster agents (32dbc6f)

Refactoring

- **package_protection_rules**: Add missing attributes (c307dd2)

Testing

- **files**: Omit optional `ref` parameter in test case (9cb3396)
- **files**: Test with and without `ref` parameter in test case (f316b46)
- **fixtures**: Remove deprecated config option (2156949)
- **registry**: Disable functional tests for unavailable endpoints (ee393a1)

7.10.31 v4.7.0 (2024-06-28)

Bug Fixes

- Add ability to add help to custom_actions (9acd2d2)

Now when registering a custom_action can add help text if desired.

Also delete the VerticalHelpFormatter as no longer needed. When the help value is set to `None` or some other value, the actions will get printed vertically. Before when the help value was not set the actions would all get put onto one line.

Chores

- Add a help message for `gitlab project-key enable` (1291dbb)

Add some help text for `gitlab project-key enable`. This both adds help text and shows how to use the new `help` feature.

Example:

$ gitlab project-key –help usage: gitlab project-key [-h] {list,get,create,update,delete,enable} ...

options: -h, –help show this help message and exit

action: {list,get,create,update,delete,enable} Action to execute on the GitLab resource. list List the GitLab resources get Get a GitLab resource create Create a GitLab resource update Update a GitLab resource delete Delete a GitLab resource enable Enable a deploy key for the project

- Sort CLI behavior-related args to remove (9b4b0ef)

Sort the list of CLI behavior-related args that are to be removed.

- **deps**: Update all non-major dependencies (88de2f0)
- **deps**: Update all non-major dependencies (a510f43)
- **deps**: Update all non-major dependencies (d4fdf90)
- **deps**: Update all non-major dependencies (d5de288)
- **deps**: Update dependency types-setuptools to v70 (7767514)
- **deps**: Update gitlab/gitlab-ee docker tag to v17.0.1-ee.0 (df0ff4c)
- **deps**: Update gitlab/gitlab-ee docker tag to v17.0.2-ee.0 (51779c6)
- **deps**: Update python-semantic-release/upload-to-gh-release digest to 477a404 (02a551d)
- **deps**: Update python-semantic-release/upload-to-gh-release digest to 6b7558f (fd0f0b0)

Features

- Add `--no-mask-credentials` CLI argument (18aa1fc)

This gives the ability to not mask credentials when using the `--debug` argument.

- **api**: Add support for latest pipeline (635f5a7)

7.10.32 v4.6.0 (2024-05-28)

Bug Fixes

- Don't raise `RedirectError` for redirected `HEAD` requests (8fc13b9)
- Handle large number of approval rules (ef8f0e1)

Use `iterator=True` when going through the list of current approval rules. This allows it to handle more than the default of 20 approval rules.

Closes: #2825

- **cli**: Don't require `--id` when enabling a deploy key (98fc578)

No longer require `--id` when doing: gitlab project-key enable

Now only the –project-id and –key-id are required.

- **deps**: Update minimum dependency versions in pyproject.toml (37b5a70)

Update the minimum versions of the dependencies in the pyproject.toml file.

This is related to PR #2878

- **projects**: Fix 'import_project' file argument type for typings (33fbc14)

Signed-off-by: Adrian DC radian.dc@gmail.com

Chores

- Add an initial .git-blame-ignore-revs (74db84c)

This adds the `.git-blame-ignore-revs` file which allows ignoring certain commits when doing a `git blame --ignore-revs`

Ignore the commit that requires keyword arguments for `register_custom_action()`

https://docs.github.com/en/repositories/working-with-files/using-files/viewing-a-file#ignore-commits-in-the-blame-view

- Add type info for ProjectFile.content (62fa271)

Closes: #2821

- Correct type-hint for `job.trace()` (840572e)

Closes: #2808

- Create a CustomAction dataclass (61d8679)
- Remove typing-extensions from requirements.txt (d569128)

We no longer support Python versions before 3.8. So it isn't needed anymore.

- Require keyword arguments for register_custom_action (7270523)

This makes it more obvious when reading the code what each argument is for.

- Update commit reference in git-blame-ignore-revs (d0fd5ad)
- **cli**: Add ability to not add `_id_attr` as an argument (2037352)

In some cases we don't want to have `_id_attr` as an argument.

Add ability to have it not be added as an argument.

- **cli**: Add some simple help for the standard operations (5a4a940)

Add help for the following standard operations: * list: List the GitLab resources * get: Get a GitLab resource * create: Create a GitLab resource * update: Update a GitLab resource * delete: Delete a GitLab resource

For example: $ gitlab project-key –help usage: gitlab project-key [-h] {list,get,create,update,delete,enable} ...

options: -h, –help show this help message and exit

action: list get create update delete enable Action to execute on the GitLab resource. list List the GitLab resources get Get a GitLab resource create Create a GitLab resource update Update a GitLab resource delete Delete a GitLab resource

- **cli**: On the CLI help show the API endpoint of resources (f1ef565)

This makes it easier for people to map CLI command names to the API.

Looks like this: $ gitlab –help The GitLab resource to manipulate. application API endpoint: /applications application-appearance API endpoint: /application/appearance application-settings API endpoint: /application/settings application-statistics API endpoint: /application/statistics

- **deps**: Update all non-major dependencies (4c7014c)

- **deps**: Update all non-major dependencies (ba1eec4)
- **deps**: Update dependency requests to v2.32.0 [security] (1bc788c)
- **deps**: Update gitlab/gitlab-ee docker tag to v17 (5070d07)
- **deps**: Update python-semantic-release/upload-to-gh-release digest to 673709c (1b550ac)

Features

- More usernames support for MR approvals (12d195a)

I don't think commit a2b8c8ccfb5d went far enough to enable usernames support. We create and edit a lot of approval rules based on an external service (similar to CODE_OWNERS), but only have the usernames available, and currently, have to look up each user to get their user ID to populate user_ids for .set_approvers() calls. Would very much like to skip the lookup and just send the usernames, which this change should allow.

See: https://docs.gitlab.com/ee/api/merge_request_approvals.html#create-project-level-rule

Signed-off-by: Jarod Wilson jarod@redhat.com

- **api**: Add additional parameter to project/group iteration search (#2796, 623dac9)

Co-authored-by: Cristiano Casella cristiano.casella@seacom.it

Co-authored-by: Nejc Habjan hab.nejc@gmail.com

- **api**: Add support for gitlab service account (#2851, b187dea)

Co-authored-by: Nejc Habjan hab.nejc@siemens.com

7.10.33 v4.5.0 (2024-05-13)

Bug Fixes

- Consider `scope` an ArrayAttribute in PipelineJobManager (c5d0404)

List query params like 'scope' were not being handled correctly for pipeline/jobs endpoint. This change ensures multiple values are appended with '[]', resulting in the correct URL structure.

Signed-off-by: Guilherme Gallo guilherme.gallo@collabora.com

Background: If one queries for pipeline jobs with `scope=["failed", "success"]`

One gets: GET /api/v4/projects/176/pipelines/1113028/jobs?scope=success&scope=failed

But it is supposed to get: GET /api/v4/projects/176/pipelines/1113028/jobs?scope[]=success&scope[]=failed

The current version only considers the last element of the list argument.

- User.warn() to show correct filename of issue (529f1fa)

Previously would only go to the 2nd level of the stack for determining the offending filename and line number. When it should be showing the first filename outside of the python-gitlab source code. As we want it to show the warning for the user of the libraries code.

Update test to show it works as expected.

- **api**: Fix saving merge request approval rules (b8b3849)

Closes #2548

- **api**: Update manual job status when playing it (9440a32)
- **cli**: Allow exclusive arguments as optional (#2770, 7ec3189)

- fix(cli): allow exclusive arguments as optional

The CLI takes its arguments from the RequiredOptional, which has three fields: required, optional, and exclusive. In practice, the exclusive options are not defined as either required or optional, and would not be allowed in the CLI. This changes that, so that exclusive options are also added to the argument parser.

- fix(cli): inform argument parser that options are mutually exclusive
- fix(cli): use correct exclusive options, add unit test

Closes #2769

- **test**: Use different ids for merge request, approval rule, project (`c23e6bd`)

The original bug was that the merge request identifier was used instead of the approval rule identifier. The test didn't notice that because it used 1 for all identifiers. Make these identifiers different so that a mixup will become apparent.

Build System

- Add "–no-cache-dir" to pip commands in Dockerfile (`4ef94c8`)

This would not leave cache files in the built docker image.

Additionally, also only build the wheel in the build phase.

On my machine, before this PR, size is 74845395; after this PR, size is 72617713.

Chores

- Adapt style for black v24 (`4e68d32`)
- Add py312 & py313 to tox environment list (`679ddc7`)

Even though there isn't a Python 3.13 at this time, this is done for the future. tox is already configured to just warn about missing Python versions, but not fail if they don't exist.

- Add tox `labels` to enable running groups of environments (`d7235c7`)

tox now has a feature of `labels` which allows running groups of environments using the command `tox -m LABEL_NAME`. For example `tox -m lint` which has been setup to run the linters.

Bumped the minimum required version of tox to be 4.0, which was released over a year ago.

- Update `mypy` to 1.9.0 and resolve one issue (`dd00bfc`)

mypy 1.9.0 flagged one issue in the code. Resolve the issue. Current unit tests already check that a `None` value returns `text/plain`. So function is still working as expected.

- Update version of `black` for `pre-commit` (`3501716`)

The version of `black` needs to be updated to be in sync with what is in `requirements-lint.txt`

- **deps**: Update all non-major dependencies (`4f338ae`)
- **deps**: Update all non-major dependencies (`65d0e65`)
- **deps**: Update all non-major dependencies (`1f0343c`)
- **deps**: Update all non-major dependencies (`0e9f4da`)
- **deps**: Update all non-major dependencies (`d5b5fb0`)
- **deps**: Update all non-major dependencies (`14a3ffe`)
- **deps**: Update all non-major dependencies (`3c4dcca`)
- **deps**: Update all non-major dependencies (`04c569a`)

- **deps**: Update all non-major dependencies (3c4b27e)
- **deps**: Update all non-major dependencies (7dc2fa6)
- **deps**: Update all non-major dependencies (48726fd)
- **deps**: Update codecov/codecov-action action to v4 (d2be1f7)
- **deps**: Update dependency black to v24 (f59aee3)
- **deps**: Update dependency black to v24.3.0 [security] (f6e8692)
- **deps**: Update dependency furo to v2024 (f6fd02d)
- **deps**: Update dependency jinja2 to v3.1.4 [security] (8ea10c3)
- **deps**: Update dependency myst-parser to v3 (9289189)
- **deps**: Update dependency pytest to v8 (253babb)
- **deps**: Update dependency pytest-cov to v5 (db32000)
- **deps**: Update dependency pytest-docker to v3 (35d2aec)
- **deps**: Update gitlab/gitlab-ee docker tag to v16 (ea8c4c2)
- **deps**: Update gitlab/gitlab-ee docker tag to v16.11.1-ee.0 (1ed8d6c)
- **deps**: Update gitlab/gitlab-ee docker tag to v16.11.2-ee.0 (9be48f0)
- **deps**: Update python-semantic-release/python-semantic-release action to v9 (e11d889)

Documentation

- Add FAQ about conflicting parameters (683ce72)

We have received multiple issues lately about this. Add it to the FAQ.

- Correct rotate token example (c53e695)

Rotate token returns a dict. Change example to print the entire dict.

Closes: #2836

- How to run smoke tests (2d1f487)

Signed-off-by: Tim Knight tim.knight1@engineering.digital.dwp.gov.uk

- Note how to use the Docker image from within GitLab CI (6d4bffb)

Ref: #2823

- **artifacts**: Fix argument indentation (c631eeb)
- **objects**: Minor rst formatting typo (57dfd17)

To correctly format a code block have to use `::`

- **README**: Tweak GitLab CI usage docs (d9aaa99)

Features

- **api**: Allow updating protected branches (#2771, a867c48)
- feat(api): allow updating protected branches

Closes #2390

- **cli**: Allow skipping initial auth calls (001e596)

- **job_token_scope**: Support Groups in job token allowlist API (#2816, 2d1b749)
- feat(job_token_scope): support job token access allowlist API

Signed-off-by: Tim Knight tim.knight1@engineering.digital.dwp.gov.uk

l.dwp.gov.uk> Co-authored-by: Nejc Habjan nejc.habjan@siemens.com

Testing

- Don't use weak passwords (c64d126)

Newer versions of GitLab will refuse to create a user with a weak password. In order for us to move to a newer GitLab version in testing use a stronger password for the tests that create a user.

- Remove approve step (48a6705)

Signed-off-by: Tim Knight tim.knight1@engineering.digital.dwp.gov.uk

- Tidy up functional tests (06266ea)

Signed-off-by: Tim Knight tim.knight1@engineering.digital.dwp.gov.uk

- Update api tests for GL 16.10 (4bef473)
- Make sure we're testing python-gitlab functionality, make sure we're not awaiting on Gitlab Async functions - Decouple and improve test stability

Signed-off-by: Tim Knight tim.knight1@engineering.digital.dwp.gov.uk

- Update tests for gitlab 16.8 functionality (f8283ae)
- use programmatic dates for expires_at in tokens tests - set PAT for 16.8 into tests

Signed-off-by: Tim Knight tim.knight1@engineering.digital.dwp.gov.uk

- **functional**: Enable bulk import feature flag before test (b81da2e)
- **smoke**: Normalize all dist titles for smoke tests (ee013fe)

7.10.34 v4.4.0 (2024-01-15)

Bug Fixes

- **cli**: Support binary files with `@` notation (57749d4)

Support binary files being used in the CLI with arguments using the `@` notation. For example `--avatar @/path/to/avatar.png`

Also explicitly catch the common OSError exception, which is the parent exception for things like: FileNotFoundError, PermissionError and more exceptions.

Remove the bare exception handling. We would rather have the full traceback of any exceptions that we don't know about and add them later if needed.

Closes: #2752

Chores

- **ci**: Add Python 3.13 development CI job (ff0c11b)

Add a job to test the development versions of Python 3.13.

- **ci**: Align upload and download action versions (dcca59d)
- **deps**: Update actions/upload-artifact action to v4 (7114af3)

- **deps**: Update all non-major dependencies (550f935)
- **deps**: Update all non-major dependencies (cbc13a6)
- **deps**: Update all non-major dependencies (369a595)
- **deps**: Update dependency flake8 to v7 (20243c5)
- **deps**: Update dependency jinja2 to v3.1.3 [security] (880913b)
- **deps**: Update pre-commit hook pycqa/flake8 to v7 (9a199b6)

Features

- **api**: Add reviewer_details manager for mergrequest to get reviewers of merge request (adbd90c)

Those changes implements 'GET /projects/:id/merge_requests/:merge_request_iid/reviewers' gitlab API call. Naming for call is not reviewers because reviewers atribute already presen in merge request response

- **api**: Support access token rotate API (b13971d)
- **api**: Support single resource access token get API (dae9e52)

7.10.35 v4.3.0 (2023-12-28)

Bug Fixes

- **cli**: Add ability to disable SSL verification (3fe9fa6)

Add a `--no-ssl-verify` option to disable SSL verification

Closes: #2714

Chores

- **deps**: Update actions/setup-python action to v5 (fad1441)
- **deps**: Update actions/stale action to v9 (c01988b)
- **deps**: Update all non-major dependencies (d7bdb02)
- **deps**: Update all non-major dependencies (9e067e5)
- **deps**: Update all non-major dependencies (bb2af7b)
- **deps**: Update all non-major dependencies (5ef1b4a)
- **deps**: Update dependency types-setuptools to v69 (de11192)

Documentation

- Fix rst link typo in CONTRIBUTING.rst (2b6da6e)

Features

- **api**: Add support for the Draft notes API (#2728, ebf9d82)
- feat(api): add support for the Draft notes API
- fix(client): handle empty 204 reponses in PUT requests

7.10.36 v4.2.0 (2023-11-28)

Chores

- **deps**: Update all non-major dependencies (8aeb853)
- **deps**: Update all non-major dependencies (9fe2335)
- **deps**: Update all non-major dependencies (91e66e9)
- **deps**: Update all non-major dependencies (d0546e0)
- **deps**: Update dessant/lock-threads action to v5 (f4ce867)

Features

- Add pipeline status as Enum (4954bbc)

https://docs.gitlab.com/ee/api/pipelines.html

- **api**: Add support for wiki attachments (#2722, 7b864b8)

Added UploadMixin in mixin module Added UploadMixin dependency for Project, ProjectWiki, GroupWiki Added api tests for wiki upload Added unit test for mixin Added docs sections to wikis.rst

7.10.37 v4.1.1 (2023-11-03)

Bug Fixes

- **build**: Include py.typed in dists (b928639)

Chores

- **ci**: Add release id to workflow step (9270e10)
- **deps**: Update all non-major dependencies (32954fb)

Documentation

- **users**: Add missing comma in v4 API create runner examples (b1b2edf)

The examples which show usage of new runner registration api endpoint are missing commas. This change adds the missing commas.

7.10.38 v4.1.0 (2023-10-28)

Bug Fixes

- Remove depricated MergeStatus (c6c012b)

Chores

- Add source label to container image (7b19278)
- **CHANGELOG**: Re-add v4.0.0 changes using old format (258a751)
- **CHANGELOG**: Revert python-semantic-release format change (b5517e0)
- **deps**: Update all non-major dependencies (bf68485)
- **rtd**: Revert to python 3.11 (#2694, 1113742)

Continuous Integration

- Remove unneeded GitLab auth (fd7bbfc)

Features

- Add Merge Request merge_status and detailed_merge_status values as constants (e18a424)

7.10.39 v4.0.0 (2023-10-17)

Bug Fixes

- **cli**: Add _from_parent_attrs to user-project manager (#2558, 016d90c)
- **cli**: Fix action display in –help when there are few actions (b22d662)

fixes #2656

- **cli**: Remove deprecated `--all` option in favor of `--get-all` (e9d48cf)

BREAKING CHANGE: The `--all` option is no longer available in the CLI. Use `--get-all` instead.

- **client**: Support empty 204 responses in http_patch (e15349c)
- **snippets**: Allow passing list of files (31c3c5e)

Chores

- Add package pipelines API link (2a2404f)
- Change `_update_uses` to `_update_method` and use an Enum (7073a2d)

Change the name of the `_update_uses` attribute to `_update_method` and store an Enum in the attribute to indicate which type of HTTP method to use. At the moment it supports `POST` and `PUT`. But can in the future support `PATCH`.

- Fix test names (f1654b8)
- Make linters happy (3b83d5d)
- Switch to docker-compose v2 (713b5ca)

Closes: #2625

- Update PyYAML to 6.0.1 (3b8939d)

Fixes issue with CI having error: `AttributeError: cython_sources`

Closes: #2624

- **ci**: Adapt release workflow and config for v8 (827fefe)
- **ci**: Fix pre-commit deps and python version (1e7f257)
- **ci**: Follow upstream config for release build_command (3e20a76)
- **ci**: Remove Python 3.13 dev job (e8c50f2)
- **ci**: Update release build for python-semantic-release v8 (#2692, bf050d1)
- **deps**: Bring furo up to date with sphinx (a15c927)
- **deps**: Bring myst-parser up to date with sphinx 7 (da03e9c)
- **deps**: Pin pytest-console-scripts for 3.7 (6d06630)
- **deps**: Update actions/checkout action to v3 (e2af1e8)
- **deps**: Update actions/checkout action to v4 (af13914)

- **deps**: Update actions/setup-python action to v4 (e0d6783)
- **deps**: Update actions/upload-artifact action to v3 (b78d6bf)
- **deps**: Update all non-major dependencies (1348a04)
- **deps**: Update all non-major dependencies (ff45124)
- **deps**: Update all non-major dependencies (0d49164)
- **deps**: Update all non-major dependencies (6093dbc)
- **deps**: Update all non-major dependencies (bb728b1)
- **deps**: Update all non-major dependencies (9083787)
- **deps**: Update all non-major dependencies (b6a3db1)
- **deps**: Update all non-major dependencies (16f2d34)
- **deps**: Update all non-major dependencies (5b33ade)
- **deps**: Update all non-major dependencies (3732841)
- **deps**: Update all non-major dependencies (511f45c)
- **deps**: Update all non-major dependencies (d4a7410)
- **deps**: Update all non-major dependencies (12846cf)
- **deps**: Update all non-major dependencies (33d2aa2)
- **deps**: Update all non-major dependencies (5ff56d8)
- **deps**: Update all non-major dependencies (7586a5c)
- **deps**: Update all non-major dependencies to v23.9.1 (a16b732)
- **deps**: Update dependency build to v1 (2e856f2)
- **deps**: Update dependency commitizen to v3.10.0 (becd8e2)
- **deps**: Update dependency pylint to v3 (491350c)
- **deps**: Update dependency pytest-docker to v2 (b87bb0d)
- **deps**: Update dependency setuptools to v68 (0f06082)
- **deps**: Update dependency sphinx to v7 (2918dfd)
- **deps**: Update dependency types-setuptools to v68 (bdd4eb6)
- **deps**: Update dependency ubuntu to v22 (8865552)
- **deps**: Update pre-commit hook commitizen-tools/commitizen to v3.10.0 (626c2f8)
- **deps**: Update pre-commit hook maxbrunet/pre-commit-renovate to v36 (db58cca)
- **deps**: Update pre-commit hook maxbrunet/pre-commit-renovate to v37 (b4951cd)
- **deps**: Update pre-commit hook pycqa/pylint to v3 (0f4a346)
- **deps**: Update relekang/python-semantic-release action to v8 (c57c85d)
- **helpers**: Fix previously undetected flake8 issue (bf8bd73)
- **rtd**: Fix docs build on readthedocs.io (#2654, 3d7139b)
- **rtd**: Use readthedocs v2 syntax (6ce2149)

Documentation

- Correct error with back-ticks (#2653, 0b98dd3)

New linting package update detected the issue.

- **access_token**: Adopt token docs to 16.1 (fe7a971)

expires_at is now required Upstream MR: https://gitlab.com/gitlab-org/gitlab/-/merge_requests/124964

- **advanced**: Document new netrc behavior (45b8930)

BREAKING CHANGE: python-gitlab now explicitly passes auth to requests, meaning it will only read netrc credentials if no token is provided, fixing a bug where netrc credentials took precedence over OAuth tokens. This also affects the CLI, where all environment variables now take precedence over netrc files.

- **files**: Fix minor typo in variable declaration (118ce42)

Features

- Added iteration to issue and group filters (8d2d297)
- Officially support Python 3.12 (2a69c0e)
- Remove support for Python 3.7, require 3.8 or higher (058d5a5)

Python 3.8 is End-of-Life (EOL) as of 2023-06-27 as stated in https://devguide.python.org/versions/ and https://peps.python.org/pep-0537/

By dropping support for Python 3.7 and requiring Python 3.8 or higher it allows python-gitlab to take advantage of new features in Python 3.8, which are documented at: https://docs.python.org/3/whatsnew/3.8.html

BREAKING CHANGE: As of python-gitlab 4.0.0, Python 3.7 is no longer supported. Python 3.8 or higher is required.

- Use requests AuthBase classes (5f46cfd)
- **api**: Add optional GET attrs for /projects/:id/ci/lint (40a102d)
- **api**: Add ProjectPackagePipeline (5b4addd)

Add ProjectPackagePipeline, which is scheduled to be included in GitLab 16.0

- **api**: Add support for job token scope settings (59d6a88)
- **api**: Add support for new runner creation API (#2635, 4abcd17)

Co-authored-by: Nejc Habjan hab.nejc@gmail.com

- **api**: Support project remote mirror deletion (d900910)
- **client**: Mask tokens by default when logging (1611d78)
- **packages**: Allow uploading bytes and files (61e0fae)

This commit adds a keyword argument to GenericPackageManager.upload() to allow uploading bytes and file-like objects to the generic package registry. That necessitates changing file path to be a keyword argument as well, which then cascades into a whole slew of checks to not allow passing both and to not allow uploading file-like objects as JSON data.

Closes https://github.com/python-gitlab/python-gitlab/issues/1815

- **releases**: Add support for direct_asset_path (d054917)

This commit adds support for the "new" alias for `filepath`: `direct_asset_path` (added in 15.10) in release links API.

Refactoring

- **artifacts**: Remove deprecated `artifact()`in favor of `artifacts.raw()` (`90134c9`)

BREAKING CHANGE: The deprecated `project.artifact()` method is no longer available. Use `project.artifacts.raw()` instead.

- **artifacts**: Remove deprecated `artifacts()`in favor of `artifacts.download()` (`42639f3`)

BREAKING CHANGE: The deprecated `project.artifacts()` method is no longer available. Use `project.artifacts.download()` instead.

- **build**: Build project using PEP 621 (`71fca8c`)

BREAKING CHANGE: python-gitlab now stores metadata in pyproject.toml as per PEP 621, with setup.py removed. pip version v21.1 or higher is required if you want to perform an editable install.

- **const**: Remove deprecated global constant import (`e4a1f6e`)

BREAKING CHANGE: Constants defined in `gitlab.const` can no longer be imported globally from `gitlab`. Import them from `gitlab.const` instead.

- **groups**: Remove deprecated LDAP group link add/delete methods (`5c8b7c1`)

BREAKING CHANGE: The deprecated `group.add_ldap_group_link()` and `group.delete_ldap_group_link()` methods are no longer available. Use `group.ldap_group_links.create()` and `group.ldap_group_links.delete()` instead.

- **lint**: Remove deprecated `lint()`in favor of `ci_lint.create()` (`0b17a2d`)

BREAKING CHANGE: The deprecated `lint()` method is no longer available. Use `ci_lint.create()` instead.

- **list**: `as_list` support is removed. (`9b6d89e`)

In `list()` calls support for the `as_list` argument has been removed. `as_list` was previously deprecated and now the use of `iterator` will be required if wanting to have same functionality as using `as_list`

BREAKING CHANGE: Support for the deprecated `as_list` argument in `list()` calls has been removed. Use `iterator` instead.

- **projects**: Remove deprecated `project.transfer_project()` in favor of `project.transfer()` (`27ed490`)

BREAKING CHANGE: The deprecated `project.transfer_project()` method is no longer available. Use `project.transfer()` instead.

Testing

- Add tests for token masking (`163bfcf`)
- Correct calls to `script_runner.run()` (`cd04315`)

Warnings were being raised. Resolve those warnings.

- Fix failing tests that use 204 (No Content) plus content (`3074f52`)

urllib3>=2 now checks for expected content length. Also codes 204 and 304 are set to expect a content length of 0 [1]

So in the unit tests stop setting content to return in these situations.

[1] https://github.com/urllib3/urllib3/blob/88a707290b655394aade060a8b7eaee83152dc8b/src/urllib3/response.py#L691-L693

- **cli**: Add test for user-project list (`a788cff`)

BREAKING CHANGES

- **advanced**: Python-gitlab now explicitly passes auth to requests, meaning it will only read netrc credentials if no token is provided, fixing a bug where netrc credentials took precedence over OAuth tokens. This also affects the CLI, where all environment variables now take precedence over netrc files.
- **build**: Python-gitlab now stores metadata in pyproject.toml as per PEP 621, with setup.py removed. pip version v21.1 or higher is required if you want to perform an editable install.

7.10.40 v3.15.0 (2023-06-09)

Chores

- Update copyright year to include 2023 (`511c6e5`)
- Update sphinx from 5.3.0 to 6.2.1 (`c44a290`)
- **ci**: Use OIDC trusted publishing for pypi.org (#2559, `7be09e5`)
- chore(ci): use OIDC trusted publishing for pypi.org
- chore(ci): explicitly install setuptools in tests
- **deps**: Update all non-major dependencies (`e3de6ba`)
- **deps**: Update dependency commitizen to v3 (`784d59e`)
- **deps**: Update dependency myst-parser to v1 (`9c39848`)
- **deps**: Update dependency requests-toolbelt to v1 (`86eba06`)
- **deps**: Update dependency types-setuptools to v67 (`c562424`)
- **deps**: Update pre-commit hook commitizen-tools/commitizen to v3 (`1591e33`)
- **deps**: Update pre-commit hook maxbrunet/pre-commit-renovate to v35 (`8202e3f`)

Documentation

- Remove exclusive EE about issue links (`e0f6f18`)

Features

- Add support for `select="package_file"` in package upload (`3a49f09`)

Add ability to use `select="package_file"` when uploading a generic package as described in: https://docs.gitlab.com/ee/user/packages/generic_packages/index.html

Closes: #2557

- Usernames support for MR approvals (`a2b8c8c`)

This can be used instead of 'user_ids'

See: https://docs.gitlab.com/ee/api/merge_request_approvals.html#create-project-level-rule

- **api**: Add support for events scope parameter (`348f56e`)

7.10.41 v3.14.0 (2023-04-11)

Bug Fixes

- Support int for `parent_id` in `import_group` (`90f96ac`)

This will also fix other use cases where an integer is passed in to MultipartEncoder.

Added unit tests to show it works.

Closes: #2506

- **cli**: Add ability to escape at-prefixed parameter (#2513, 4f7c784)
- fix(cli): Add ability to escape at-prefixed parameter (#2511)

Co-authored-by: Nejc Habjan hab.nejc@gmail.com

- **cli**: Display items when iterator is returned (33a04e7)
- **cli**: Warn user when no fields are displayed (8bf53c8)
- **client**: Properly parse content-type when charset is present (76063c3)

Chores

- Add Contributor Covenant 2.1 as Code of Conduct (fe334c9)

See https://www.contributor-covenant.org/version/2/1/code_of_conduct/

- Add Python 3.12 testing (0867564)

Add a unit test for Python 3.12. This will use the latest version of Python 3.12 that is available from https://github.com/actions/python-versions/

At this time it is 3.12.0-alpha.4 but will move forward over time until the final 3.12 release and updates. So 3.12.0, 3.12.1, … will be matched.

- Add SECURITY.md (572ca3b)
- Remove `pre-commit` as a default `tox` environment (#2470, fde2495)

For users who use `tox` having `pre-commit` as part of the default environment list is redundant as it will run the same tests again that are being run in other environments. For example: black, flake8, pylint, and more.

- Use a dataclass to return values from `prepare_send_data` (f2b5e4f)

I found the tuple of three values confusing. So instead use a dataclass to return the three values. It is still confusing but a little bit less so.

Also add some unit tests

- **.github**: Actually make PR template the default (7a8a862)
- **ci**: Wait for all coverage reports in CI status (511764d)
- **contributing**: Refresh development docs (d387d91)
- **deps**: Update actions/stale action to v8 (7ac4b86)
- **deps**: Update all non-major dependencies (8b692e8)
- **deps**: Update all non-major dependencies (2f06999)
- **deps**: Update all non-major dependencies (#2493, 07d03dc)
- chore(deps): update all non-major dependencies * chore(fixtures): downgrade GitLab for now * chore(deps): ungroup typing deps, group gitlab instead * chore(deps): downgrade argcomplete for now

---

Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com>

Co-authored-by: Nejc Habjan nejc.habjan@siemens.com

- **deps**: Update black (23.1.0) and commitizen (2.40.0) (#2479, 44786ef)

Update the dependency versions: black: 23.1.0

commitizen: 2.40.0

They needed to be updated together as just updating `black` caused a dependency conflict.

Updated files by running `black` and committing the changes.

- **deps**: Update dependency coverage to v7 (#2501, aee73d0)

Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com>

- **deps**: Update dependency flake8 to v6 (#2502, 3d4596e)

Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com>

- **deps**: Update dependency furo to v2023 (7a1545d)
- **deps**: Update dependency pre-commit to v3 (#2508, 7d779c8)

Co-authored-by: renovate[bot] <29139614+renovate[bot]@users.noreply.github.com>

- **deps**: Update mypy (1.0.0) and responses (0.22.0) (9c24657)

Update the `requirements-*` files.

In order to update mypy==1.0.0 we need to also update responses==0.22.0

Fix one issue found by `mypy`

Leaving updates for `precommit` to be done in a separate commit by someone.

- **deps**: Update pre-commit hook psf/black to v23 (217a787)
- **github**: Add default pull request template (bf46c67)
- **pre-commit**: Bumping versions (e973729)
- **renovate**: Bring back custom requirements pattern (ae0b21c)
- **renovate**: Do not ignore tests dir (5b8744e)
- **renovate**: Swith to gitlab-ee (8da48ee)
- **setup**: Depend on typing-extensions for 3.7 until EOL (3abc557)

Documentation

- Fix update badge behaviour (3d7ca1c)

docs: fix update badge behaviour

Earlier: badge.image_link = new_link

Now: badge.image_url = new_image_url badge.link_url = new_link_url

- **advanced**: Clarify netrc, proxy behavior with requests (1da7c53)
- **advanced**: Fix typo in Gitlab examples (1992790)
- **objects**: Fix typo in pipeline schedules (3057f45)

Features

- Add resource_weight_event for ProjectIssue (6e5ef55)
- **backends**: Use PEP544 protocols for structural subtyping (#2442, 4afeaff)

The purpose of this change is to track API changes described in https://github.com/python-gitlab/python-gitlab/blob/main/docs/api-levels.rst, for example, for package versioning and breaking change announcements in case of protocol changes.

This is MVP implementation to be used by #2435.

- **cli**: Add setting of `allow_force_push` for protected branch (929e07d)

For the CLI: add `allow_force_push` as an optional argument for creating a protected branch.

API reference: https://docs.gitlab.com/ee/api/protected_branches.html#protect-repository-branches

Closes: #2466

- **client**: Add http_patch method (#2471, f711d9e)

In order to support some new API calls we need to support the HTTP `PATCH` method.

Closes: #2469

- **objects**: Support fetching PATs via id or `self` endpoint (19b38bd)
- **projects**: Allow importing additional items from GitHub (ce84f2e)

Refactoring

- **client**: Let mypy know http_password is set (2dd177b)

Testing

- **functional**: Clarify MR fixture factory name (d8fd1a8)
- **meta**: Move meta suite into unit tests (847004b)

They're always run with it anyway, so it makes no difference.

- **unit**: Consistently use inline fixtures (1bc56d1)
- **unit**: Increase V4 CLI coverage (5748d37)
- **unit**: Remove redundant package (4a9e3ee)
- **unit**: Split the last remaining unittest-based classes into modules" (14e0f65)

7.10.42 v3.13.0 (2023-01-30)

Bug Fixes

- Change return value to "None" in case getattr returns None to prevent error (3f86d36)
- Typo fixed in docs (ee5f444)
- Use the ProjectIterationManager within the Project object (44f05dc)

The Project object was previously using the GroupIterationManager resulting in the incorrect API endpoint being used. Utilize the correct ProjectIterationManager instead.

Resolves #2403

- **api**: Make description optional for releases (5579750)

- **client**: Regression - do not automatically get_next if page=# and (585e3a8)
- **deps**: Bump requests-toolbelt to fix deprecation warning (faf842e)

Chores

- Add a UserWarning if both `iterator=True` and `page=X` are used (#2462, 8e85791)

If a caller calls a `list()` method with both `iterator=True` (or `as_list=False`) and `page=X` then emit a `UserWarning` as the options are mutually exclusive.

- Add docs for schedule pipelines (9a9a6a9)
- Add test, docs, and helper for 409 retries (3e1c625)
- Make backends private (1e629af)
- Remove tox `envdir` values (3c7c7fc)

tox > 4 no longer will re-use the tox directory :(What this means is that with the previous config if you ran: $ tox -e mypy; tox -e isort; tox -e mypy It would recreate the tox environment each time :(

By removing the `envdir` values it will have the tox environments in separate directories and not recreate them.

The have an FAQ entry about this: https://tox.wiki/en/latest/upgrading.html#re-use-of-environments

- Update attributes for create and update projects (aa44f2a)
- Use SPDX license expression in project metadata (acb3a4a)
- **ci**: Complete all unit tests even if one has failed (#2438, 069c6c3)
- **deps**: Update actions/download-artifact action to v3 (64ca597)
- **deps**: Update actions/stale action to v7 (76eb024)
- **deps**: Update all non-major dependencies (ea7010b)
- **deps**: Update all non-major dependencies (122988c)
- **deps**: Update all non-major dependencies (49c0233)
- **deps**: Update all non-major dependencies (10c4f31)
- **deps**: Update all non-major dependencies (bbd01e8)
- **deps**: Update all non-major dependencies (6682808)
- **deps**: Update all non-major dependencies (1816107)
- **deps**: Update all non-major dependencies (21e767d)
- **deps**: Update dessant/lock-threads action to v4 (337b25c)
- **deps**: Update pre-commit hook maxbrunet/pre-commit-renovate to v34.48.4 (985b971)
- **deps**: Update pre-commit hook pycqa/flake8 to v6 (82c61e1)
- **tox**: Ensure test envs have all dependencies (63cf4e4)

Documentation

- **faq**: Describe and group common errors (4c9a072)

Features

- Add keep_base_url when getting configuration from file (50a0301)
- Add resource iteration events (see https://docs.gitlab.com/ee/api/resource_iteration_events.html) (ef5feb4)
- Allow filtering pipelines by source (b6c0872)

See: https://docs.gitlab.com/ee/api/pipelines.html#list-project-pipelines Added in GitLab 14.3

- Allow passing kwargs to Gitlab class when instantiating with `from_config` (#2392, e88d34e)
- **api**: Add support for bulk imports API (043de2d)
- **api**: Add support for resource groups (5f8b8f5)
- **api**: Support listing pipelines triggered by pipeline schedules (865fa41)
- **client**: Automatically retry on HTTP 409 Resource lock (dced76a)

Fixes: #2325

- **client**: Bootstrap the http backends concept (#2391, 91a665f)
- **group**: Add support for group restore API (9322db6)

Refactoring

- Add reason property to RequestsResponse (#2439, b59b7bd)
- Migrate MultipartEncoder to RequestsBackend (#2421, 43b369f)
- Move Response object to backends (#2420, 7d9ce0d)
- Move the request call to the backend (#2413, 283e7cc)
- Moving RETRYABLE_TRANSIENT_ERROR_CODES to const (887852d)
- Remove unneeded requests.utils import (#2426, 6fca651)

Testing

- **functional**: Do not require config file (43c2dda)
- **unit**: Expand tests for pipeline schedules (c7cf0d1)

7.10.43 v3.12.0 (2022-11-28)

Bug Fixes

- Use POST method and return dict in `cancel_merge_when_pipeline_succeeds()` (#2350, bd82d74)
- Call was incorrectly using a `PUT` method when should have used a `POST` method. * Changed return type to a `dict` as GitLab only returns {'status': 'success'} on success. Since the function didn't work previously, this should not impact anyone. * Updated the test fixture `merge_request` to add ability to create a pipeline. * Added functional test for `mr.cancel_merge_when_pipeline_succeeds()`

Fixes: #2349

- **cli**: Enable debug before doing auth (65abb85)

Authentication issues are currently hard to debug since `--debug` only has effect after `gl.auth()` has been called.

For example, a 401 error is printed without any details about the actual HTTP request being sent:

$ gitlab –debug –server-url https://gitlab.com current-user get 401: 401 Unauthorized

By moving the call to `gl.enable_debug()` the usual debug logs get printed before the final error message.

Signed-off-by: Emanuele Aina emanuele.aina@collabora.com

- **cli**: Expose missing mr_default_target_self project attribute (`12aea32`)

Example::

gitlab project update –id 616 –mr-default-target-self 1

References:

- https://gitlab.com/gitlab-org/gitlab/-/merge_requests/58093 * https://gitlab.com/gitlab-org/gitlab/-/blob/v13.11.0-ee/doc/user/project/merge_requests/creating_merge_requests.md#new-merge-request-from-a-fork
  - https://gitlab.com/gitlab-org/gitlab/-/blob/v14.7.0-ee/doc/api/projects.md#get-single-project

Chores

- Correct website for pylint (`fcd72fe`)

Use https://github.com/PyCQA/pylint as the website for pylint.

- Validate httpx package is not installed by default (`0ecf3bb`)
- **deps**: Update all non-major dependencies (`d8a657b`)
- **deps**: Update all non-major dependencies (`b2c6d77`)
- **deps**: Update pre-commit hook maxbrunet/pre-commit-renovate to v34 (`623e768`)
- **deps**: Update pre-commit hook maxbrunet/pre-commit-renovate to v34.20.0 (`e6f1bd6`)
- **deps**: Update pre-commit hook maxbrunet/pre-commit-renovate to v34.24.0 (`a0553c2`)

Documentation

- Use the term “log file” for getting a job log file (`9d2b1ad`)

The GitLab docs refer to it as a log file: https://docs.gitlab.com/ee/api/jobs.html#get-a-log-file

“trace” is the endpoint name but not a common term people will think of for a “log file”

- **api**: Pushrules remove saying `None` is returned when not found (`c3600b4`)

In `groups.pushrules.get()`, GitLab does not return `None` when no rules are found. GitLab returns a 404.

Update docs to not say it will return `None`

Also update docs in `project.pushrules.get()` to be consistent. Not 100% sure if it returns `None` or returns a 404, but we don’t need to document that.

Closes: #2368

- **groups**: Describe GitLab.com group creation limitation (`9bd433a`)

Features

- Add support for SAML group links (#2367, `1020ce9`)
- Implement secure files API (`d0a0348`)
- **api**: Add application statistics (`6fcf3b6`)
- **api**: Add support for getting a project’s pull mirror details (`060cfe1`)

Add the ability to get a project's pull mirror details. This was added in GitLab 15.5 and is a PREMIUM feature.

https://docs.gitlab.com/ee/api/projects.html#get-a-projects-pull-mirror-details

- **api**: Add support for remote project import (#2348, e5dc72d)
- **api**: Add support for remote project import from AWS S3 (#2357, 892281e)
- **ci**: Re-run Tests on PR Comment workflow (034cde3)
- **groups**: Add LDAP link manager and deprecate old API endpoints (3a61f60)
- **groups**: Add support for listing ldap_group_links (#2371, ad7c8fa)

Refactoring

- Explicitly use ProjectSecureFile (0c98b2d)

Testing

- **api**: Fix flaky test `test_cancel_merge_when_pipeline_succeeds` (6525c17)

This is an attempt to fix the flaky test `test_cancel_merge_when_pipeline_succeeds`. Were seeing a: 405 Method Not Allowed error when setting the MR to merge_when_pipeline_succeeds.

Closes: #2383

7.10.44 v3.11.0 (2022-10-28)

Bug Fixes

- Intermittent failure in test_merge_request_reset_approvals (3dde36e)

Have been seeing intermittent failures in the test: tests/functional/api/test_merge_requests.py::test_merge_request_reset_approvals

Also saw a failure in: tests/functional/cli/test_cli_v4.py::test_accept_request_merge[subprocess]

Add a call to `wait_for_sidekiq()` to hopefully resolve the issues.

- Remove `project.approvals.set_approvals()` method (91f08f0)

The `project.approvals.set_approvals()` method used the `/projects/:id/approvers` end point. That end point was removed from GitLab in the 13.11 release, on 2-Apr-2021 in commit 27dc2f2fe81249bbdc25f7bd8fe799752aac05e6 via merge commit e482597a8cf1bae8e27abd6774b684fb90491835. It was deprecated on 19-Aug-2019.

See merge request: https://gitlab.com/gitlab-org/gitlab/-/merge_requests/57473

- Use epic id instead of iid for epic notes (97cae38)
- **cli**: Handle list response for json/yaml output (9b88132)

Handle the case with the CLI where a list response is returned from GitLab and json/yaml output is requested.

Add a functional CLI test to validate it works.

Closes: #2287

Chores

- Add `not-callable` to pylint ignore list (f0c02a5)

The `not-callable` error started showing up. Ignore this error as it is invalid. Also `mypy` tests for these issues.

Closes: #2334

- Add basic type checks to functional/api tests (5b642a5)
- Add basic type checks to meta tests (545d6d6)
- Add basic typing to functional tests (ee143c9)
- Add basic typing to smoke tests (64e8c31)
- Add basic typing to test root (0b2f6bc)
- Add responses to pre-commit deps (4b8ddc7)
- Fix flaky test (fdd4114)
- Narrow type hints for license API (50731c1)
- Renovate and precommit cleanup (153d373)
- Revert compose upgrade (dd04e8e)

This reverts commit f825d70e25feae8cd9da84e768ec6075edbc2200.

- Simplify `wait_for_sidekiq` usage (196538b)

Simplify usage of `wait_for_sidekiq` by putting the assert if it timed out inside the function rather than after calling it.

- Topic functional tests (d542eba)
- Update the issue templates (c15bd33)
- Have an option to go to the discussions * Have an option to go to the Gitter chat * Move the bug/issue template into the .github/ISSUE_TEMPLATE/ directory
- Use kwargs for http_request docs (124abab)
- **deps**: Group non-major upgrades to reduce noise (37d14bd)
- **deps**: Pin and clean up test dependencies (60b9197)
- **deps**: Pin dependencies (953f38d)
- **deps**: Pin GitHub Actions (8dbaa5c)
- **deps**: Update all non-major dependencies (dde3642)
- **deps**: Update all non-major dependencies (2966234)
- **deps**: Update black to v22.10.0 (531ee05)
- **deps**: Update dependency commitizen to v2.35.0 (4ce9559)
- **deps**: Update dependency mypy to v0.981 (da48849)
- **deps**: Update dependency pylint to v2.15.3 (6627a60)
- **deps**: Update dependency types-requests to v2.28.11.2 (d47c0f0)
- **deps**: Update pre-commit hook maxbrunet/pre-commit-renovate to v33 (932bbde)
- **deps**: Update typing dependencies (81285fa)

Documentation

- Add minimal docs about the `enable_debug()` method (b4e9ab7)

Add some minimal documentation about the `enable_debug()` method.

- **advanced**: Add hint on type narrowing (a404152)

- **api**: Describe the list() and all() runners' functions (b6cc3f2)
- **api**: Describe use of lower-level methods (b7a6874)
- **api**: Update `merge_requests.rst`: `mr_id` to `mr_iid` (b32234d)

Typo: Author probably meant `mr_iid` (i.e. project-specific MR ID)

and **not** `mr_id` (i.e. server-wide MR ID)

Closes: https://github.com/python-gitlab/python-gitlab/issues/2295

Signed-off-by: Stavros Ntentos 133706+stdedos@users.noreply.github.com

- **commits**: Fix commit create example for binary content (bcc1eb4)
- **readme**: Add a basic feature list (b4d53f1)

Features

- **api**: Add support for topics merge API (9a6d197)
- **build**: Officially support Python 3.11 (74f66c7)

Refactoring

- Migrate legacy EE tests to pytest (88c2505)
- Pre-commit trigger from tox (6e59c12)
- Pytest-docker fixtures (3e4781a)
- **deps**: Drop compose v1 dependency in favor of v2 (f825d70)

Testing

- Enable skipping tests per GitLab plan (01d5f68)
- Fix `test_project_push_rules` test (8779cf6)

Make the `test_project_push_rules` test work.

- Use false instead of /usr/bin/false (51964b3)

On Debian systems false is located at /bin/false (coreutils package). This fixes unit test failure on Debian system:

FileNotFoundError: [Errno 2] No such file or directory: '/usr/bin/false'

/usr/lib/python3.10/subprocess.py:1845: FileNotFoundError

7.10.45 v3.10.0 (2022-09-28)

Bug Fixes

- **cli**: Add missing attribute for MR changes (20c46a0)
- **cli**: Add missing attributes for creating MRs (1714d0a)

Chores

- Bump GitLab docker image to 15.4.0-ee.0 (b87a2bc)
- Use `settings.delayed_group_deletion=False` as that is the recommended method to turn off the delayed group deletion now. * Change test to look for `default` as `pages` is not mentioned in the docs[1]

[1] https://docs.gitlab.com/ee/api/sidekiq_metrics.html#get-the-current-queue-metrics

- **deps**: Update black to v22.8.0 (86b0e40)
- **deps**: Update dependency commitizen to v2.32.2 (31aea28)
- **deps**: Update dependency commitizen to v2.32.5 (e180f14)
- **deps**: Update dependency pytest to v7.1.3 (ec7f26c)
- **deps**: Update dependency types-requests to v2.28.10 (5dde7d4)
- **deps**: Update pre-commit hook commitizen-tools/commitizen to v2.32.2 (31ba64f)

Features

- Add reset_approvals api (88693ff)

Added the newly added reset_approvals merge request api.

Signed-off-by: Lucas Zampieri lzampier@redhat.com

- Add support for deployment approval endpoint (9c9eeb9)

Add support for the deployment approval endpoint[1]

[1] https://docs.gitlab.com/ee/api/deployments.html#approve-or-reject-a-blocked-deployment Closes: #2253

7.10.46 v3.9.0 (2022-08-28)

Chores

- Fix issue if only run test_gitlab.py func test (98f1956)

Make it so can run just the test_gitlab.py functional test.

For example: $ tox -e api_func_v4 – -k test_gitlab.py

- Only check for our UserWarning (bd4dfb4)

The GitHub CI is showing a ResourceWarning, causing our test to fail.

Update test to only look for our UserWarning which should not appear.

What was seen when debugging the GitHub CI: {message: ResourceWarning(“unclosed <socket.socket fd=12, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=6, laddr=(‘127.0.0.1’, 50862), raddr=(‘127.0.0.1’, 8080)>”), category: ‘ResourceWarning’, filename: ‘/home/runner/work/python-gitlab/python-gitlab/.tox/api_func_v4/lib/python3.10/site-packages/urllib3/poolmanager.py’, lineno: 271, line: None }

- **ci**: Make pytest annotations work (f67514e)
- **deps**: Update dependency commitizen to v2.31.0 (4ff0894)
- **deps**: Update dependency commitizen to v2.32.1 (9787c5c)
- **deps**: Update dependency types-requests to v2.28.8 (8e5b86f)
- **deps**: Update dependency types-requests to v2.28.9 (be932f6)
- **deps**: Update dependency types-setuptools to v64 (4c97f26)
- **deps**: Update pre-commit hook commitizen-tools/commitizen to v2.31.0 (71d37d9)
- **deps**: Update pre-commit hook commitizen-tools/commitizen to v2.32.1 (cdd6efe)
- **deps**: Update pre-commit hook pycqa/flake8 to v5 (835d884)

Features

- Add support for merge_base API (dd4fbd5)

7.10.47 v3.8.1 (2022-08-10)

Bug Fixes

- **client**: Do not assume user attrs returned for auth() (a07547c)

This is mostly relevant for people mocking the API in tests.

Chores

- Add license badge to readme (9aecc9e)
- Consolidate license and authors (366665e)
- Remove broad Exception catching from `config.py` (0abc90b)

Change "except Exception:" catching to more granular exceptions.

A step in enabling the "broad-except" check in pylint.

- **deps**: Update dependency commitizen to v2.29.5 (181390a)
- **deps**: Update dependency flake8 to v5.0.4 (50a4fec)
- **deps**: Update dependency sphinx to v5 (3f3396e)

7.10.48 v3.8.0 (2022-08-04)

Bug Fixes

- Optionally keep user-provided base URL for pagination (#2149, e2ea8b8)
- **client**: Ensure encoded query params are never duplicated (1398426)

Chores

- Change `_repr_attr` for Project to be `path_with_namespace` (7cccefe)

Previously `_repr_attr` was `path` but that only gives the basename of the path. So https://gitlab.com/gitlab-org/gitlab would only show "gitlab". Using `path_with_namespace` it will now show "gitlab-org/gitlab"

- Enable mypy check `disallow_any_generics` (24d17b4)
- Enable mypy check `no_implicit_optional` (64b208e)
- Enable mypy check `warn_return_any` (76ec4b4)

Update code so that the `warn_return_any` check passes.

- Make code PEP597 compliant (433dba0)

Use `encoding="utf-8"` in `open()` and open-like functions.

https://peps.python.org/pep-0597/

- Use `urlunparse` instead of string replace (6d1b62d)

Use the `urlunparse()` function to reconstruct the URL without the query parameters.

- **ci**: Bump semantic-release for fixed commit parser (1e063ae)
- **clusters**: Deprecate clusters support (b46b379)

Cluster support was deprecated in GitLab 14.5 [1]. And disabled by default in GitLab 15.0 [2]

- Update docs to mark clusters as deprecated * Remove testing of clusters

[1] https://docs.gitlab.com/ee/api/project_clusters.html [2] https://gitlab.com/groups/gitlab-org/configure/-/epics/8

- **deps**: Update dependency commitizen to v2.29.2 (`30274ea`)
- **deps**: Update dependency flake8 to v5 (`cdc384b`)
- **deps**: Update dependency types-requests to v2.28.6 (`54dd4c3`)
- **deps**: Update pre-commit hook commitizen-tools/commitizen to v2.29.2 (`4988c02`)
- **topics**: 'title' is required when creating a topic (`271f688`)

In GitLab >= 15.0 `title` is required when creating a topic.

Documentation

- Describe self-revoking personal access tokens (`5ea48fc`)

Features

- Support downloading archive subpaths (`cadb0e5`)
- **client**: Warn user on misconfigured URL in `auth()` (`0040b43`)

Refactoring

- **client**: Factor out URL check into a helper (`af21a18`)
- **client**: Remove handling for incorrect link header (`77c04b1`)

This was a quirk only present in GitLab 13.0 and fixed with 13.1. See https://gitlab.com/gitlab-org/gitlab/-/merge_requests/33714 and https://gitlab.com/gitlab-org/gitlab/-/issues/218504 for more context.

Testing

- Attempt to make functional test startup more reliable (`67508e8`)

The functional tests have been erratic. Current theory is that we are starting the tests before the GitLab container is fully up and running.

- Add checking of the Health Check[1] endpoints. * Add a 20 second delay after we believe it is up and running. * Increase timeout from 300 to 400 seconds

[1] https://docs.gitlab.com/ee/user/admin_area/monitoring/health_check.html

- **functional**: Bump GitLab docker image to 15.2.0-ee.0 (`69014e9`)

Use the GitLab docker image 15.2.0-ee.0 in the functional testing.

- **unit**: Reproduce duplicate encoded query params (`6f71c66`)

7.10.49 v3.7.0 (2022-07-28)

Bug Fixes

- Add `get_all` param (and `--get-all`) to allow passing `all` to API (`7c71d5d`)
- Enable epic notes (`5fc3216`)

Add the notes attribute to GroupEpic

- Ensure path elements are escaped (`5d9c198`)

Ensure the path elements that are passed to the server are escaped. For example a "/" will be changed to "%2F"

Closes: #2116

- Results returned by `attributes` property to show updates (`e5affc8`)

Previously the `attributes` method would show the original values in a Gitlab Object even if they had been updated. Correct this so that the updated value will be returned.

Also use copy.deepcopy() to ensure that modifying the dictionary returned can not also modify the object.

- Support array types for most resources (`d9126cd`)
- Use the [] after key names for array variables in `params` (`1af44ce`)

1. If a value is of type ArrayAttribute then append '[]' to the name of the value for query parameters (`params`).

This is step 3 in a series of steps of our goal to add full support for the GitLab API data types[1]: * array * hash * array of hashes

Step one was: commit 5127b1594c00c7364e9af15e42d2e2f2d909449b Step two was: commit a57334f1930752c70ea15847a39324fa94042460

Fixes: #1698

[1] https://docs.gitlab.com/ee/api/#encoding-api-parameters-of-array-and-hash-types

- **cli**: Remove irrelevant MR approval rule list filters (`0daec5f`)
- **config**: Raise error when gitlab id provided but no config file found (`ac46c1c`)
- **config**: Raise error when gitlab id provided but no config section found (`1ef7018`)
- **runners**: Fix listing for /runners/all (`c6dd57c`)

Chores

- Add a `lazy` boolean attribute to `RESTObject` (`a7e8cfb`)

This can be used to tell if a `RESTObject` was created using `lazy=True`.

Add a message to the `AttributeError` if attribute access fails for an instance created with `lazy=True`.

- Change name of API functional test to `api_func_v4` (`8cf5cd9`)

The CLI test is `cli_func_v4` and using `api_func_v4` matches with that naming convention.

- Enable mypy check `strict_equality` (`a29cd6c`)

Enable the `mypy strict_equality` check.

- Enable using GitLab EE in functional tests (`17c01ea`)

Enable using GitLab Enterprise Edition (EE) in the functional tests. This will allow us to add functional tests for EE only features in the functional tests.

- Fix misspelling (`2d08fc8`)
- Fixtures: after delete() wait to verify deleted (`1f73b6b`)

In our fixtures that create: - groups - project merge requests - projects - users

They delete the created objects after use. Now wait to ensure the objects are deleted before continuing as having unexpected objects existing can impact some of our tests.

- Make reset_gitlab() better (`d87d6b1`)

Saw issues in the CI where reset_gitlab() would fail. It would fail to delete the group that is created when GitLab starts up. Extending the timeout didn't fix the issue.

Changed the code to use the new `helpers.safe_delete()` function. Which will delete the resource and then make sure it is deleted before returning.

Also added some logging functionality that can be seen if logging is turned on in pytest.

- Revert "test(functional): simplify token creation" (`4b798fc`)

This reverts commit 67ab24fe5ae10a9f8cc9122b1a08848e8927635d.

- Simplify multi-nested try blocks (`e734470`)

Instead of have a multi-nested series of try blocks. Convert it to a more readable series of `if` statements.

- **authors**: Fix email and do the ABC (`9833632`)
- **ci_lint**: Add create attributes (`6e1342f`)
- **deps**: Update black to v22.6.0 (`82bd596`)
- **deps**: Update dependency commitizen to v2.28.0 (`8703dd3`)
- **deps**: Update dependency commitizen to v2.29.0 (`c365be1`)
- **deps**: Update dependency mypy to v0.971 (`7481d27`)
- **deps**: Update dependency pylint to v2.14.4 (`2cee2d4`)
- **deps**: Update dependency pylint to v2.14.5 (`e153636`)
- **deps**: Update dependency requests to v2.28.1 (`be33245`)
- **deps**: Update pre-commit hook commitizen-tools/commitizen to v2.28.0 (`d238e1b`)
- **deps**: Update pre-commit hook commitizen-tools/commitizen to v2.29.0 (`ad8d62a`)
- **deps**: Update pre-commit hook pycqa/pylint to v2.14.4 (`5cd39be`)
- **deps**: Update pre-commit hook pycqa/pylint to v2.14.5 (`c75a1d8`)
- **deps**: Update typing dependencies (`f2209a0`)
- **deps**: Update typing dependencies (`e772248`)
- **docs**: Convert tabs to spaces (`9ea5520`)

Some tabs snuck into the documentation. Convert them to 4-spaces.

Documentation

- Describe fetching existing export status (`9c5b8d5`)
- Describe ROPC flow in place of password authentication (`91c17b7`)
- Document CI Lint usage (`d5de4b1`)
- Update return type of pushrules (`53cbecc`)

Update the return type of pushrules to surround None with back-ticks to make it code-formatted.

- **authors**: Add John (`e2afb84`)
- **cli**: Showcase use of token scopes (`4a6f8d6`)
- **projects**: Document export with upload to URL (`03f5484`)
- **readme**: Remove redundant `-v` that breaks the command (`c523e18`)

- **users**: Add docs about listing a user's projects (065a1a5)

Add docs about listing a user's projects.

Update docs on the membership API to update the URL to the upstream docs and also add a note that it requires Administrator access to use.

Features

- Add 'merge_pipelines_enabled' project attribute (fc33c93)

Boolean. Enable or disable merge pipelines.

See: https://docs.gitlab.com/ee/api/projects.html#edit-project https://docs.gitlab.com/ee/ci/pipelines/merged_results_pipelines.html

- Add `asdict()` and `to_json()` methods to Gitlab Objects (08ac071)

Add an `asdict()` method that returns a dictionary representation copy of the Gitlab Object. This is a copy and changes made to it will have no impact on the Gitlab Object.

The `asdict()` method name was chosen as both the `dataclasses` and `attrs` libraries have an `asdict()` function which has the similar purpose of creating a dictionary represenation of an object.

Also add a `to_json()` method that returns a JSON string representation of the object.

Closes: #1116

- Add support for filtering jobs by scope (0e1c0dd)

See: 'scope' here:

https://docs.gitlab.com/ee/api/jobs.html#list-project-jobs

- Add support for group and project invitations API (7afd340)
- Add support for group push rules (b5cdc09)

Add the GroupPushRules and GroupPushRulesManager classes.

Closes: #1259

- Add support for iterations API (194ee01)
- Allow sort/ordering for project releases (b1dd284)

See: https://docs.gitlab.com/ee/api/releases/#list-releases

- Support validating CI lint results (3b1ede4)
- **api**: Add support for `get` for a MR approval rule (89c18c6)

In GitLab 14.10 they added support to get a single merge request approval rule [1]

Add support for it to ProjectMergeRequestApprovalRuleManager

[1] https://docs.gitlab.com/ee/api/merge_request_approvals.html#get-a-single-merge-request-level-rule

- **api**: Add support for instance-level registry repositories (284d739)
- **cli**: Add a custom help formatter (005ba93)

Add a custom argparse help formatter that overrides the output format to list items vertically.

The formatter is derived from argparse.HelpFormatter with minimal changes.

Co-authored-by: John Villalovos john@sodarock.com

Co-authored-by: Nejc Habjan nejc.habjan@siemens.com

- **cli**: Add support for global CI lint (3f67c4b)

- **groups**: Add support for group-level registry repositories (70148c6)
- **groups**: Add support for shared projects API (66461ba)
- **issues**: Add support for issue reorder API (8703324)
- **namespaces**: Add support for namespace existence API (4882cb2)
- **objects**: Add Project CI Lint support (b213dd3)

Add support for validating a project's CI configuration [1]

[1] https://docs.gitlab.com/ee/api/lint.html

- **projects**: Add support for project restore API (4794ecc)

Refactoring

- Migrate services to integrations (a428051)
- **objects**: Move ci lint to separate file (6491f1b)
- **test-projects**: Apply suggestions and use fixtures (a51f848)
- **test-projects**: Remove test_restore_project (9be0875)

Testing

- Add more tests for container registries (f6b6e18)
- Add test to show issue fixed (75bec7d)

https://github.com/python-gitlab/python-gitlab/issues/1698 has been fixed. Add test to show that.

- Allow `podman` users to run functional tests (ff215b7)

Users of `podman` will likely have `DOCKER_HOST` set to something like `unix:///run/user/1000/podman/podman.sock`

Pass this environment variable so that it will be used during the functional tests.

- Always ensure clean config environment (8d4f13b)
- Fix broken test if user had config files (864fc12)

Use `monkeypatch` to ensure that no config files are reported for the test.

Closes: #2172

- **api_func_v4**: Catch deprecation warning for `gl.lint()` (95fe924)

Catch the deprecation warning for the call to `gl.lint()`, so it won't show up in the log.

- **cli**: Add tests for token scopes (263fe3d)
- **ee**: Add an EE specific test (10987b3)
- **functional**: Replace len() calls with list membership checks (97e0eb9)
- **functional**: Simplify token creation (67ab24f)
- **functional**: Use both get_all and all in list() tests (201298d)
- **projects**: Add unit tests for projects (67942f0)

7.10.50 v3.6.0 (2022-06-28)

Bug Fixes

- **base**: Do not fail repr() on lazy objects (1efb123)
- **cli**: Fix project export download for CLI (5d14867)

Since ac1c619cae6481833f5df91862624bf0380fef67 we delete parent arg keys from the args dict so this has been trying to access the wrong attribute.

- **cli**: Project-merge-request-approval-rule (15a242c)

Using the CLI the command: gitlab project-merge-request-approval-rule list –mr-iid 1 –project-id foo/bar

Would raise an exception. This was due to the fact that `_id_attr` and `_repr_attr` were set for keys which are not returned in the response.

Add a unit test which shows the `repr` function now works. Before it did not.

This is an EE feature so we can't functional test it.

Closes: #2065

Chores

- Add link to Commitizen in Github workflow (d08d07d)

Add a link to the Commitizen website in the Github workflow. Hopefully this will help people when their job fails.

- Bump mypy pre-commit hook (0bbcad7)
- Correct ModuleNotFoundError() arguments (0b7933c)

Previously in commit 233b79ed442aac66faf9eb4b0087ea126d6dffc5 I had used the `name` argument for `ModuleNotFoundError()`. This basically is the equivalent of not passing any message to `ModuleNotFoundError()`. So when the exception was raised it wasn't very helpful.

Correct that and add a unit-test that shows we get the message we expect.

- Enable 'consider-using-sys-exit' pylint check (0afcc3e)

Enable the 'consider-using-sys-exit' pylint check and fix errors raised.

- Enable pylint check "raise-missing-from" (1a2781e)

Enable the pylint check "raise-missing-from" and fix errors detected.

- Enable pylint check: "attribute-defined-outside-init" (d6870a9)

Enable the pylint check: "attribute-defined-outside-init" and fix errors detected.

- Enable pylint check: "no-else-return" (d0b0811)

Enable the pylint check "no-else-return" and fix the errors detected.

- Enable pylint check: "no-self-use" (80aadaf)

Enable the pylint check "no-self-use" and fix the errors detected.

- Enable pylint check: "redefined-outer-name", (1324ce1)

Enable the pylint check "redefined-outer-name" and fix the errors detected.

- Enable pylint checks (1e89164)

Enable the pylint checks: * unnecessary-pass * unspecified-encoding

Update code to resolve errors found

- Enable pylint checks which require no changes (50fdbc4)

Enabled the pylint checks that don't require any code changes. Previously these checks were disabled.

- Fix issue found with pylint==2.14.3 (eeab035)

A new error was reported when running pylint==2.14.3: gitlab/client.py:488:0: W1404: Implicit string concatenation found in call (implicit-str-concat)

Fixed this issue.

- Have `EncodedId` creation always return `EncodedId` (a1a246f)

There is no reason to return an `int` as we can always return a `str` version of the `int`

Change `EncodedId` to always return an `EncodedId`. This removes the need to have `mypy` ignore the error raised.

- Move `RequiredOptional` to the `gitlab.types` module (7d26530)

By having `RequiredOptional` in the `gitlab.base` module it makes it difficult with circular imports. Move it to the `gitlab.types` module which has no dependencies on any other gitlab module.

- Move `utils._validate_attrs` inside `types.RequiredOptional` (9d629bb)

Move the `validate_attrs` function to be inside the `RequiredOptional` class. It makes sense for it to be part of the class as it is working on data related to the class.

- Patch sphinx for explicit re-exports (06871ee)
- Remove use of '%' string formatter in `gitlab/utils.py` (0c5a121)

Replace usage with f-string

- Rename `__call__()` to `run()` in GitlabCLI (6189437)

Less confusing to have it be a normal method.

- Rename `whaction` and `action` to `resource_action` in CLI (fb3f28a)

Rename the variables `whaction` and `action` to `resource_action` to improve code-readability.

- Rename `what` to `gitlab_resource` (c86e471)

Naming a variable `what` makes it difficult to understand what it is used for.

Rename it to `gitlab_resource` as that is what is being stored.

The Gitlab documentation talks about them being resources: https://docs.gitlab.com/ee/api/api_resources.html

This will improve code readability.

- Require f-strings (96e994d)

We previously converted all string formatting to use f-strings. Enable pylint check to enforce this.

- Update type-hints return signature for GetWithoutIdMixin methods (aa972d4)

Commit f0152dc3cc9a42aa4dc3c0014b4c29381e9b39d6 removed situation where `get()` in a `GetWithoutIdMixin` based class could return `None`

Update the type-hints to no longer return `Optional` AKA `None`

- Use multiple processors when running PyLint (7f2240f)

Use multiple processors when running PyLint. On my system it took about 10.3 seconds to run PyLint before this change. After this change it takes about 5.8 seconds to run PyLint.

- **ci**: Increase timeout for docker container to come online (bda020b)

Have been seeing timeout issues more and more. Increase timeout from 200 seconds to 300 seconds (5 minutes).

- **ci**: Pin 3.11 to beta.1 (7119f2d)
- **cli**: Ignore coverage on exceptions triggering cli.die (98ccc3c)
- **cli**: Rename "object" to "GitLab resource" (62e64a6)

Make the parser name more user friendly by renaming from generic "object" to "GitLab resource"

- **deps**: Ignore python-semantic-release updates (f185b17)
- **deps**: Update actions/setup-python action to v4 (77c1f03)
- **deps**: Update dependency commitizen to v2.27.1 (456f9f1)
- **deps**: Update dependency mypy to v0.960 (8c016c7)
- **deps**: Update dependency mypy to v0.961 (f117b2f)
- **deps**: Update dependency pylint to v2.14.3 (9a16bb1)
- **deps**: Update dependency requests to v2.28.0 (d361f4b)
- **deps**: Update pre-commit hook commitizen-tools/commitizen to v2.27.1 (22c5db4)
- **deps**: Update pre-commit hook pycqa/pylint to v2.14.3 (d1fe838)
- **deps**: Update typing dependencies (acc5c39)
- **deps**: Update typing dependencies (aebf9c8)
- **deps**: Update typing dependencies (f3f79c1)
- **docs**: Ignore nitpicky warnings (1c3efb5)
- **gitlab**: Fix implicit re-exports for mpypy (981b844)
- **mixins**: Remove None check as http_get always returns value (f0152dc)
- **workflows**: Explicitly use python-version (eb14475)

Documentation

- Documentation updates to reflect addition of mutually exclusive attributes (24b720e)
- Drop deprecated setuptools build_sphinx (048d66a)
- Use `as_list=False` or `all=True` in Getting started (de8c6e8)

In the "Getting started with the API" section of the documentation, use either `as_list=False` or `all=True` in the example usages of the `list()` method.

Also add a warning about the fact that `list()` by default does not return all items.

- **api**: Add separate section for advanced usage (22ae101)
- **api**: Document usage of head() methods (f555bfb)
- **api**: Fix incorrect docs for merge_request_approvals (#2094, 5583eaa)
- docs(api): fix incorrect docs for merge_request_approvals

The `set_approvers()` method is on the `ProjectApprovalManager` class. It is not part of the `ProjectApproval` class.

The docs were previously showing to call `set_approvers` using a `ProjectApproval` instance, which would fail. Correct the documentation.

This was pointed out by a question on the Gitter channel.

Co-authored-by: Nejc Habjan nejc.habjan@siemens.com

- **api**: Stop linking to python-requests.org (49c7e83)
- **api-usage**: Add import os in example (2194a44)
- **ext**: Fix rendering for RequiredOptional dataclass (4d431e5)
- **projects**: Document 404 gotcha with unactivated integrations (522ecff)
- **projects**: Provide more detailed import examples (8f8611a)
- **usage**: Refer to upsteam docs instead of custom attributes (ae7d3b0)
- **variables**: Instruct users to follow GitLab rules for values (194b6be)

Features

- Add support for Protected Environments (1dc9d0f)
- https://docs.gitlab.com/ee/api/protected_environments.html - https://github.com/python-gitlab/python-gitlab/issues/1130

no write operation are implemented yet as I have no use case right now and am not sure how it should be done

- Support mutually exclusive attributes and consolidate validation to fix board lists (#2037, 3fa330c)

add exclusive tuple to RequiredOptional data class to support for mutually exclusive attributes

consolidate _check_missing_create_attrs and _check_missing_update_attrs from mixins.py into _validate_attrs in utils.py

change _create_attrs in board list manager classes from required=('label_ld',) to exclusive=('label_id','asignee_id','milestone_id')

closes https://github.com/python-gitlab/python-gitlab/issues/1897

- **api**: Convert gitlab.const to Enums (c3c6086)

This allows accessing the elements by value, i.e.:

import gitlab.const gitlab.const.AccessLevel(20)

- **api**: Implement HEAD method (90635a7)
- **api**: Support head() method for get and list endpoints (ce9216c)
- **client**: Introduce `iterator=True` and deprecate `as_list=False` in `list()` (cdc6605)

`as_list=False` is confusing as it doesn't explain what is being returned. Replace it with `iterator=True` which more clearly explains to the user that an iterator/generator will be returned.

This maintains backward compatibility with `as_list` but does issue a DeprecationWarning if `as_list` is set.

- **docker**: Provide a Debian-based slim image (384031c)
- **downloads**: Allow streaming downloads access to response iterator (#1956, b644721)
- feat(downloads): allow streaming downloads access to response iterator

Allow access to the underlying response iterator when downloading in streaming mode by specifying `iterator=True`.

Update type annotations to support this change.

- docs(api-docs): add iterator example to artifact download

Document the usage of the `iterator=True` option when downloading artifacts

- test(packages): add tests for streaming downloads
- **users**: Add approve and reject methods to User (f57139d)

As requested in #1604.

Co-authored-by: John Villalovos john@sodarock.com

- **users**: Add ban and unban methods (0d44b11)

Refactoring

- Avoid possible breaking change in iterator (#2107, 212ddfc)

Commit b6447211754e126f64e12fc735ad74fe557b7fb4 inadvertently introduced a possible breaking change as it added a new argument `iterator` and added it in between existing (potentially positional) arguments.

This moves the `iterator` argument to the end of the argument list and requires it to be a keyword-only argument.

- Do not recommend plain gitlab.const constants (d652133)
- Remove no-op id argument in GetWithoutIdMixin (0f2a602)
- **mixins**: Extract custom type transforms into utils (09b3b22)

Testing

- Add more tests for RequiredOptional (ce40fde)
- Add tests and clean up usage for new enums (323ab3c)
- Increase client coverage (00aec96)
- Move back to using latest Python 3.11 version (8c34781)
- **api**: Add tests for HEAD method (b0f02fa)
- **cli**: Improve coverage for custom actions (7327f78)
- **gitlab**: Increase unit test coverage (df072e1)
- **pylint**: Enable pylint "unused-argument" check (23feae9)

Enable the pylint "unused-argument" check and resolve issues it found.

- Quite a few functions were accepting `**kwargs` but not then passing them on through to the next level. Now pass `**kwargs` to next level. * Other functions had no reason to accept `**kwargs`, so remove it * And a few other fixes.

7.10.51 v3.5.0 (2022-05-28)

Bug Fixes

- Duplicate subparsers being added to argparse (f553fd3)

Python 3.11 added an additional check in the argparse libary which detected duplicate subparsers being added. We had duplicate subparsers being added.

Make sure we don't add duplicate subparsers.

Closes: #2015

- **cli**: Changed default `allow_abbrev` value to fix arguments collision problem (#2013, d68cacf)

fix(cli): change default `allow_abbrev` value to fix argument collision

Chores

- Add `cz` to default tox environment list and skip_missing_interpreters (`ba8c052`)

Add the `cz` (`comittizen`) check by default.

Set skip_missing_interpreters = True so that when a user runs tox and doesn't have a specific version of Python it doesn't mark it as an error.

- Exclude `build/` directory from mypy check (`989a12b`)

The `build/` directory is created by the tox environment `twine-check`. When the `build/` directory exists `mypy` will have an error.

- Rename the test which runs `flake8` to be `flake8` (`78b4f99`)

Previously the test was called `pep8`. The test only runs `flake8` so call it `flake8` to be more precise.

- Run the `pylint` check by default in tox (`55ace1d`)

Since we require `pylint` to pass in the CI. Let's run it by default in tox.

- **ci**: Fix prefix for action version (`1c02189`)
- **ci**: Pin semantic-release version (`0ea61cc`)
- **ci**: Replace commitlint with commitizen (`b8d15fe`)
- **deps**: Update dependency pylint to v2.13.8 (`b235bb0`)
- **deps**: Update dependency pylint to v2.13.9 (`4224950`)
- **deps**: Update dependency types-requests to v2.27.23 (`a6fed8b`)
- **deps**: Update dependency types-requests to v2.27.24 (`f88e3a6`)
- **deps**: Update dependency types-requests to v2.27.25 (`d6ea47a`)
- **deps**: Update pre-commit hook pycqa/pylint to v2.13.8 (`1835593`)
- **deps**: Update pre-commit hook pycqa/pylint to v2.13.9 (`1e22790`)
- **renovate**: Set schedule to reduce noise (`882fe7a`)

Documentation

- Add missing Admin access const value (`3e0d4d9`)

As shown here, Admin access is set to 60: https://docs.gitlab.com/ee/api/protected_branches.html#protected-branches-api

- Update issue example and extend API usage docs (`aad71d2`)
- **CONTRIBUTING.rst**: Fix link to conventional-changelog commit format documentation (`2373a4f`)
- **merge_requests**: Add new possible merge request state and link to the upstream docs (`e660fa8`)

The actual documentation do not mention the locked state for a merge request

Features

- Display human-readable attribute in `repr()` if present (`6b47c26`)
- **objects**: Support get project storage endpoint (`8867ee5`)
- **ux**: Display project.name_with_namespace on project repr (`e598762`)

This change the repr from:

$ gitlab.projects.get(id=some_id)

To:

$ gitlab.projects.get(id=some_id) <Project id:some_id name_with_namespace:"group_name / project_name">

This is especially useful when working on random projects or listing of projects since users generally don't remember projects ids.

Testing

- **projects**: Add tests for list project methods (fa47829)

7.10.52 v3.4.0 (2022-04-28)

Bug Fixes

- Add 52x range to retry transient failures and tests (c3ef1b5)
- Add ChunkedEncodingError to list of retryable exceptions (7beb20f)
- Also retry HTTP-based transient errors (3b49e4d)
- Avoid passing redundant arguments to API (3431887)
- **cli**: Add missing filters for project commit list (149d244)

Chores

- **client**: Remove duplicate code (5cbbf26)
- **deps**: Update black to v22.3.0 (8d48224)
- **deps**: Update codecov/codecov-action action to v3 (292e91b)
- **deps**: Update dependency mypy to v0.950 (241e626)
- **deps**: Update dependency pylint to v2.13.3 (0ae3d20)
- **deps**: Update dependency pylint to v2.13.4 (a9a9392)
- **deps**: Update dependency pylint to v2.13.5 (5709675)
- **deps**: Update dependency pylint to v2.13.7 (5fb2234)
- **deps**: Update dependency pytest to v7.1.2 (fd3fa23)
- **deps**: Update dependency types-requests to v2.27.16 (ad799fc)
- **deps**: Update dependency types-requests to v2.27.21 (0fb0955)
- **deps**: Update dependency types-requests to v2.27.22 (22263e2)
- **deps**: Update dependency types-setuptools to v57.4.12 (6551353)
- **deps**: Update pre-commit hook pycqa/pylint to v2.13.3 (8f0a3af)
- **deps**: Update pre-commit hook pycqa/pylint to v2.13.4 (9d0b252)
- **deps**: Update pre-commit hook pycqa/pylint to v2.13.5 (17d5c6c)
- **deps**: Update pre-commit hook pycqa/pylint to v2.13.7 (1396221)
- **deps**: Update typing dependencies (c12466a)
- **deps**: Update typing dependencies (d27cc6a)

- **deps**: Upgrade gitlab-ce to 14.9.2-ce.0 (`d508b18`)

Documentation

- **api-docs**: Docs fix for application scopes (`e1ad93d`)

Features

- Emit a warning when using a `list()` method returns max (`1339d64`)

A common cause of issues filed and questions raised is that a user will call a `list()` method and only get 20 items. As this is the default maximum of items that will be returned from a `list()` method.

To help with this we now emit a warning when the result from a `list()` method is greater-than or equal to 20 (or the specified `per_page` value) and the user is not using either `all=True`, `all=False`, `as_list=False`, or `page=X`.

- **api**: Re-add topic delete endpoint (`d1d96bd`)

This reverts commit e3035a799a484f8d6c460f57e57d4b59217cd6de.

- **objects**: Support getting project/group deploy tokens by id (`fcd37fe`)
- **user**: Support getting user SSH key by id (`6f93c05`)

7.10.53 v3.3.0 (2022-03-28)

Bug Fixes

- Support RateLimit-Reset header (`4060146`)

Some endpoints are not returning the `Retry-After` header when rate-limiting occurrs. In those cases use the `RateLimit-Reset` [1] header, if available.

Closes: #1889

[1] https://docs.gitlab.com/ee/user/admin_area/settings/user_and_ip_rate_limits.html#response-headers

Chores

- **deps**: Update actions/checkout action to v3 (`7333cbb`)
- **deps**: Update actions/setup-python action to v3 (`7f845f7`)
- **deps**: Update actions/stale action to v5 (`d841185`)
- **deps**: Update actions/upload-artifact action to v3 (`18a0eae`)
- **deps**: Update black to v22 (`3f84f1b`)
- **deps**: Update dependency mypy to v0.931 (`33646c1`)
- **deps**: Update dependency mypy to v0.940 (`dd11084`)
- **deps**: Update dependency mypy to v0.941 (`3a9d4f1`)
- **deps**: Update dependency mypy to v0.942 (`8ba0f8c`)
- **deps**: Update dependency pylint to v2.13.0 (`5fa403b`)
- **deps**: Update dependency pylint to v2.13.1 (`eefd724`)
- **deps**: Update dependency pylint to v2.13.2 (`10f15a6`)
- **deps**: Update dependency pytest to v7 (`ae8d70d`)
- **deps**: Update dependency pytest to v7.1.0 (`27c7e33`)

- **deps**: Update dependency pytest to v7.1.1 (e31f2ef)
- **deps**: Update dependency pytest-console-scripts to v1.3 (9c202dd)
- **deps**: Update dependency pytest-console-scripts to v1.3.1 (da392e3)
- **deps**: Update dependency requests to v2.27.1 (95dad55)
- **deps**: Update dependency sphinx to v4.4.0 (425d161)
- **deps**: Update dependency sphinx to v4.5.0 (36ab769)
- **deps**: Update dependency types-requests to v2.27.12 (8cd668e)
- **deps**: Update dependency types-requests to v2.27.14 (be6b54c)
- **deps**: Update dependency types-requests to v2.27.15 (2e8ecf5)
- **deps**: Update dependency types-setuptools to v57.4.10 (b37fc41)
- **deps**: Update pre-commit hook alessandrojcm/commitlint-pre-commit-hook to v8 (5440780)
- **deps**: Update pre-commit hook pycqa/pylint to v2.13.0 (9fe60f7)
- **deps**: Update pre-commit hook pycqa/pylint to v2.13.1 (1d0c6d4)
- **deps**: Update pre-commit hook pycqa/pylint to v2.13.2 (14d367d)
- **deps**: Update typing dependencies (21e7c37)
- **deps**: Update typing dependencies (37a7c40)

Code Style

- Reformat for black v22 (93d4403)

Documentation

- Add pipeline test report summary support (d78afb3)
- Fix typo and incorrect style (2828b10)
- **chore**: Include docs .js files in sdist (3010b40)

Features

- **object**: Add pipeline test report summary support (a97e0cf)

7.10.54 v3.2.0 (2022-02-28)

Bug Fixes

- Remove custom `delete` method for labels (0841a2a)

The usage of deleting was incorrect according to the current API. Remove custom `delete()` method as not needed.

Add tests to show it works with labels needing to be encoded.

Also enable the test_group_labels() test function. Previously it was disabled.

Add ability to do a `get()` for group labels.

Closes: #1867

- **services**: Use slug for id_attr instead of custom methods (e30f39d)

Chores

- Correct type-hints for per_page attrbute (e825653)

There are occasions where a GitLab `list()` call does not return the `x-per-page` header. For example the listing of custom attributes.

Update the type-hints to reflect that.

- Create a custom `warnings.warn` wrapper (6ca9aa2)

Create a custom `warnings.warn` wrapper that will walk the stack trace to find the first frame outside of the `gitlab/` path to print the warning against. This will make it easier for users to find where in their code the error is generated from

- Create new ArrayAttribute class (a57334f)

Create a new ArrayAttribute class. This is to indicate types which are sent to the GitLab server as arrays https://docs.gitlab.com/ee/api/#array

At this stage it is identical to the CommaSeparatedListAttribute class but will be used later to support the array types sent to GitLab.

This is the second step in a series of steps of our goal to add full support for the GitLab API data types[1]: * array * hash * array of hashes

Step one was: commit 5127b1594c00c7364e9af15e42d2e2f2d909449b

[1] https://docs.gitlab.com/ee/api/#encoding-api-parameters-of-array-and-hash-types

Related: #1698

- Require kwargs for `utils.copy_dict()` (7cf35b2)

The non-keyword arguments were a tiny bit confusing as the destination was first and the source was second.

Change the order and require key-word only arguments to ensure we don't silently break anyone.

- **ci**: Do not run release workflow in forks (2b6edb9)

Code Style

- **objects**: Add spacing to docstrings (700d25d)

Documentation

- Add delete methods for runners and project artifacts (5e711fd)
- Add retry_transient infos (bb1f054)

Co-authored-by: Nejc Habjan hab.nejc@gmail.com

- Add transient errors retry info (b7a1266)
- Enable gitter chat directly in docs (bd1ecdd)
- Revert "chore: add temporary banner for v3" (#1864, 7a13b9b)

This reverts commit a349793307e3a975bb51f864b48e5e9825f70182.

Co-authored-by: Wadim Klincov wadim.klincov@siemens.com

- **artifacts**: Deprecate artifacts() and artifact() methods (64d01ef)

Features

- **artifacts**: Add support for project artifacts delete API (c01c034)
- **merge_request_approvals**: Add support for deleting MR approval rules (85a734f)
- **mixins**: Allow deleting resources without IDs (0717517)
- **objects**: Add a complete artifacts manager (c8c2fa7)

Testing

- **functional**: Fix GitLab configuration to support pagination (5b7d00d)

When pagination occurs python-gitlab uses the URL provided by the GitLab server to use for the next request.

We had previously set the GitLab server configuraiton to say its URL was `http://gitlab.test` which is not in DNS. Set the hostname in the URL to `http://127.0.0.1:8080` which is the correct URL for the GitLab server to be accessed while doing functional tests.

Closes: #1877

- **objects**: Add tests for project artifacts (8ce0336)
- **runners**: Add test for deleting runners by auth token (14b88a1)
- **services**: Add functional tests for services (2fea2e6)
- **unit**: Clean up MR approvals fixtures (0eb4f7f)

7.10.55 v3.1.1 (2022-01-28)

Bug Fixes

- **cli**: Allow custom methods in managers (8dfed0c)
- **cli**: Make 'per_page' and 'page' type explicit (d493a5e)
- **cli**: Make 'timeout' type explicit (bbb7df5)
- **objects**: Make resource access tokens and repos available in CLI (e0a3a41)

Chores

- Always use context manager for file IO (e8031f4)
- Consistently use open() encoding and file descriptor (dc32d54)
- Create return type-hints for `get_id()` & `encoded_id` (0c3a1d1)

Create return type-hints for `RESTObject.get_id()` and `RESTObject.encoded_id`. Previously was saying they return Any. Be more precise in saying they can return either: None, str, or int.

- Don't explicitly pass args to super() (618267c)
- Remove old-style classes (ae2a015)
- Remove redundant list comprehension (271cfd3)
- Rename `gitlab/__version__.py` -> `gitlab/_version.py` (b981ce7)

It is confusing to have a `gitlab/__version__.py` because we also create a variable `gitlab.__version__` which can conflict with `gitlab/__version__.py`.

For example in `gitlab/const.py` we have to know that `gitlab.__version__` is a module and not the variable due to the ordering of imports. But in most other usage `gitlab.__version__` is a version string.

To reduce confusion make the name of the version file `gitlab/_version.py`.

- Rename `types.ListAttribute` to `types.CommaSeparatedListAttribute` (5127b15)

This name more accurately describes what the type is. Also this is the first step in a series of steps of our goal to add full support for the GitLab API data types[1]: * array * hash * array of hashes

[1] https://docs.gitlab.com/ee/api/#encoding-api-parameters-of-array-and-hash-types

- Use dataclass for RequiredOptional (30117a3)
- **tests**: Use method `projects.transfer()` (e5af2a7)

When doing the functional tests use the new function `projects.transfer` instead of the deprecated function `projects.transfer_project()`

Code Style

- Use f-strings where applicable (cfed622)
- Use literals to declare data structures (019a40f)

Documentation

- Enhance release docs for CI_JOB_TOKEN usage (5d973de)
- **changelog**: Add missing changelog items (01755fb)

Testing

- Add a meta test to make sure that v4/objects/ files are imported (9c8c804)

Add a test to make sure that all of the `gitlab/v4/objects/` files are imported in `gitlab/v4/objects/__init__.py`

- Convert usage of `match_querystring` to `match` (d16e41b)

In the `responses` library the usage of `match_querystring` is deprecated. Convert to using `match`

- Remove usage of httpmock library (5254f19)

Convert all usage of the `httpmock` library to using the `responses` library.

- Use 'responses' in test_mixins_methods.py (208da04)

Convert from httmock to responses in test_mixins_methods.py

This leaves only one file left to convert

7.10.56 v3.1.0 (2022-01-14)

Bug Fixes

- Broken URL for FAQ about attribute-error-list (1863f30)

The URL was missing a 'v' before the version number and thus the page did not exist.

Previously the URL for python-gitlab 3.0.0 was: https://python-gitlab.readthedocs.io/en/3.0.0/faq.html#attribute-error-list

Which does not exist.

Change it to: https://python-gitlab.readthedocs.io/en/v3.0.0/faq.html#attribute-error-list add the 'v' ——————————^

- Change to `http_list` for some ProjectCommit methods (497e860)

Fix the type-hints and use `http_list()` for the ProjectCommits methods: - diff() - merge_requests()

- refs()

This will enable using the pagination support we have for lists.

Closes: #1805

Closes: #1231

- Remove custom URL encoding (3d49e5e)

We were using `str.replace()` calls to take care of URL encoding issues.

Switch them to use our `utils._url_encode()` function which itself uses `urllib.parse.quote()`

Closes: #1356

- Remove default arguments for mergerequests.merge() (8e589c4)

The arguments `should_remove_source_branch` and `merge_when_pipeline_succeeds` are optional arguments. We should not be setting any default value for them.

https://docs.gitlab.com/ee/api/merge_requests.html#accept-mr

Closes: #1750

- Use url-encoded ID in all paths (12435d7)

Make sure all usage of the ID in the URL path is encoded. Normally it isn't an issue as most IDs are integers or strings which don't contain a slash ('/'). But when the ID is a string with a slash character it will break things.

Add a test case that shows this fixes wikis issue with subpages which use the slash character.

Closes: #1079

- **api**: Services: add missing `lazy` parameter (888f332)

Commit 8da0b758c589f608a6ae4eeb74b3f306609ba36d added the `lazy` parameter to the services `get()` method but missed then using the `lazy` parameter when it called `super(...).get(...)`

Closes: #1828

- **cli**: Add missing list filters for environments (6f64d40)
- **cli**: Url-encode path components of the URL (ac1c619)

In the CLI we need to make sure the components put into the path portion of the URL are url-encoded. Otherwise they will be interpreted as part of the path. For example can specify the project ID as a path, but in the URL it must be url-encoded or it doesn't work.

Also stop adding the components of the path as query parameters in the URL.

Closes: #783

Closes: #1498

- **members**: Use new *All objects for *AllManager managers (755e0a3)

Change it so that:

GroupMemberAllManager uses GroupMemberAll object ProjectMemberAllManager uses ProjectMemberAll object

Create GroupMemberAll and ProjectMemberAll objects that do not support any Mixin type methods. Previously we were using GroupMember and ProjectMember which support the `save()` and `delete()` methods but those methods will not work with objects retrieved using the `/members/all/` API calls.

`list()` API calls: [1] GET /groups/:id/members/all GET /projects/:id/members/all

`get()` API calls: [2] GET /groups/:id/members/all/:user_id GET /projects/:id/members/all/:user_id

Closes: #1825

Closes: #848

[1] https://docs.gitlab.com/ee/api/members.html#list-all-members-of-a-group-or-project-including-inherited-and-invited-members [2] https://docs.gitlab.com/ee/api/members.html#get-a-member-of-a-group-or-project-including-inherited-and-invited-members

Chores

- Add `pprint()` and `pformat()` methods to RESTObject (`d69ba04`)

This is useful in debugging and testing. As can easily print out the values from an instance in a more human-readable form.

- Add a stale workflow (`2c036a9`)

Use the stale action to close issues and pull-requests with no activity.

Issues: It will mark them as stale after 60 days and then close

them once they have been stale for 15 days.

Pull-Requests: It will mark pull-requests as stale after 90 days and then close

https://github.com/actions/stale

Closes: #1649

- Add EncodedId string class to use to hold URL-encoded paths (`a2e7c38`)

Add EncodedId string class. This class returns a URL-encoded string but ensures it will only URL-encode it once even if recursively called.

Also added some functional tests of 'lazy' objects to make sure they work.

- Add functional test of mergerequest.get() (`a92b55b`)

Add a functional test of test mergerequest.get() and mergerequest.get(..., lazy=True)

Closes: #1425

- Add logging to `tests/functional/conftest.py` (`a1ac9ae`)

I have found trying to debug issues in the functional tests can be difficult. Especially when trying to figure out failures in the CI running on Github.

Add logging to `tests/functional/conftest.py` to have a better understanding of what is happening during a test run which is useful when trying to troubleshoot issues in the CI.

- Add temporary banner for v3 (`a349793`)
- Fix functional test failure if config present (`c9ed3dd`)

Previously c8256a5933d745f70c7eea0a7d6230b51bac0fbc was done to fix this but it missed two other failures.

- Fix missing comma (`7c59fac`)

There was a missing comma which meant the strings were concatenated instead of being two separate strings.

- Ignore intermediate coverage artifacts (`110ae91`)
- Replace usage of utils._url_encode() with utils.EncodedId() (`b07eece`)

utils.EncodedId() has basically the same functionalityy of using utils._url_encode(). So remove utils._url_encode() as we don't need it.

- **dist**: Add docs *.md files to sdist (d9457d8)

build_sphinx to fail due to setup.cfg warning-is-error

- **docs**: Use admonitions consistently (55c67d1)
- **groups**: Use encoded_id for group path (868f243)
- **objects**: Use `self.encoded_id` where applicable (75758bf)

Updated a few remaining usages of `self.id` to use `self.encoded_id` as for the most part we shouldn't be using `self.id`

There are now only a few (4 lines of code) remaining uses of `self.id`, most of which seem that they should stay that way.

- **objects**: Use `self.encoded_id` where could be a string (c3c3a91)

Updated a few remaining usages of `self.id` to use `self.encoded_id` where it could be a string value.

- **projects**: Fix typing for transfer method (0788fe6)

Co-authored-by: John Villalovos john@sodarock.com

Continuous Integration

- Don't fail CI if unable to upload the code coverage data (d5b3744)

If a CI job can't upload coverage results to codecov.com it causes the CI to fail and code can't be merged.

Documentation

- Update project access token API reference link (73ae955)
- **cli**: Make examples more easily navigable by generating TOC (f33c523)

Features

- Add support for Group Access Token API (c01b7c4)

See https://docs.gitlab.com/ee/api/group_access_tokens.html

- Add support for Groups API method `transfer()` (0007006)
- **api**: Add `project.transfer()` and deprecate `transfer_project()` (259668a)
- **api**: Return result from `SaveMixin.save()` (e6258a4)

Return the new object data when calling `SaveMixin.save()`.

Also remove check for `None` value when calling `self.manager.update()` as that method only returns a dictionary.

Closes: #1081

Testing

- **groups**: Enable group transfer tests (57bb67a)

7.10.57 v3.0.0 (2022-01-05)

Bug Fixes

- Handle situation where GitLab does not return values (cb824a4)

If a query returns more than 10,000 records than the following values are NOT returned: x.total_pages x.total

Modify the code to allow no value to be set for these values. If there is not a value returned the functions will now return None.

Update unit test so no longer `xfail`

https://docs.gitlab.com/ee/user/gitlab_com/index.html#pagination-response-headers

Closes #1686

- Raise error if there is a 301/302 redirection (`d56a434`)

Before we raised an error if there was a 301, 302 redirect but only from an http URL to an https URL. But we didn't raise an error for any other redirects.

This caused two problems:

1. PUT requests that are redirected get changed to GET requests which don't perform the desired action but raise no error. This is because the GET response succeeds but since it wasn't a PUT it doesn't update. See issue: https://github.com/python-gitlab/python-gitlab/issues/1432 2. POST requests that are redirected also got changed to GET requests. They also caused hard to debug tracebacks for the user. See issue: https://github.com/python-gitlab/python-gitlab/issues/1477

Correct this by always raising a RedirectError exception and improve the exception message to let them know what was redirected.

Closes: #1485

Closes: #1432

Closes: #1477

- Stop encoding '.' to '%2E' (`702e41d`)

Forcing the encoding of '.' to '%2E' causes issues. It also goes against the RFC: https://datatracker.ietf.org/doc/html/rfc3986.html#section-2.3

From the RFC: For consistency, percent-encoded octets in the ranges of ALPHA (%41-%5A and %61-%7A), DIGIT (%30-%39), hyphen (%2D), period (%2E), underscore (%5F), or tilde (%7E) should not be created by URI producers...

Closes #1006 Related #1356 Related #1561

BREAKING CHANGE: stop encoding '.' to '%2E'. This could potentially be a breaking change for users who have incorrectly configured GitLab servers which don't handle period '.' characters correctly.

- **api**: Delete invalid 'project-runner get' command (#1628, `905781b`)
- fix(api): delete 'group-runner get' and 'group-runner delete' commands

Co-authored-by: Léo GATELLIER git@leogatellier.fr

- **api**: Replace deprecated attribute in delete_in_bulk() (#1536, `c59fbdb`)

BREAKING CHANGE: The deprecated `name_regex` attribute has been removed in favor of `name_regex_delete`. (see https://gitlab.com/gitlab-org/gitlab/-/commit/ce99813cf54)

- **build**: Do not include docs in wheel package (`68a97ce`)
- **build**: Do not package tests in wheel (`969dccc`)
- **objects**: Rename confusing `to_project_id` argument (`ce4bc0d`)

BREAKING CHANGE: rename confusing `to_project_id` argument in transfer_project to `project_id` (`--project-id` in CLI). This is used for the source project, not for the target namespace.

Chores

- Add .env as a file that search tools should not ignore (c9318a9)

The `.env` file was not set as a file that should not be ignored by search tools. We want to have the search tools search any `.env` files.

- Add and document optional parameters for get MR (bfa3dbe)

Add and document (some of the) optional parameters that can be done for a `project.merge_requests.get()`

Closes #1775

- Add get() methods for GetWithoutIdMixin based classes (d27c50a)

Add the get() methods for the GetWithoutIdMixin based classes.

Update the tests/meta/test_ensure_type_hints.py tests to check to ensure that the get methods are defined with the correct return type.

- Add initial pylint check (041091f)

Initial pylint check is added. A LONG list of disabled checks is also added. In the future we should work through the list and resolve the errors or disable them on a more granular level.

- Add Python 3.11 testing (b5ec192)

Add a unit test for Python 3.11. This will use the latest version of Python 3.11 that is available from https://github.com/actions/python-versions/

At this time it is 3.11.0-alpha.2 but will move forward over time until the final 3.11 release and updates. So 3.11.0, 3.11.1, … will be matched.

- Add running unit tests on windows/macos (ad5d60c)

Add running the unit tests on windows-latest and macos-latest with Python 3.10.

- Add test case to show branch name with period works (ea97d7a)

Add a test case to show that a branch name with a period can be fetched with a `get()`

Closes: #1715

- Add type hints for gitlab/v4/objects/commits.py (dc096a2)
- Add type-hints to gitlab/v4/objects/epics.py (d4adf8d)
- Add type-hints to gitlab/v4/objects/files.py (0c22bd9)
- Add type-hints to gitlab/v4/objects/geo_nodes.py (13243b7)
- Add type-hints to gitlab/v4/objects/groups.py (94dcb06)
- Add type-hints to gitlab/v4/objects/groups.py * Have share() function update object attributes. * Add 'get()' method so that type-checkers will understand that getting a group is of type Group.
- Add type-hints to gitlab/v4/objects/issues.py (93e39a2)
- Add type-hints to gitlab/v4/objects/jobs.py (e8884f2)
- Add type-hints to gitlab/v4/objects/labels.py (d04e557)
- Add type-hints to gitlab/v4/objects/merge_request_approvals.py (cf3a99a)
- Add type-hints to gitlab/v4/objects/merge_requests.py (f9c0ad9)

- Add type-hints to gitlab/v4/objects/merge_requests.py * Add return value to cancel_merge_when_pipeline_succeeds() function as GitLab docs show it returns a value. * Add return value to approve() function as GitLab docs show it returns a value. * Add 'get()' method so that type-checkers will understand that getting a project merge request is of type ProjectMergeRequest.
- Add type-hints to gitlab/v4/objects/milestones.py (8b6078f)
- Add type-hints to gitlab/v4/objects/pipelines.py (cb3ad6c)
- Add type-hints to gitlab/v4/objects/repositories.py (00d7b20)
- Add type-hints to gitlab/v4/objects/services.py (8da0b75)
- Add type-hints to gitlab/v4/objects/sidekiq.py (a91a303)
- Add type-hints to gitlab/v4/objects/snippets.py (f256d4f)
- Add type-hints to gitlab/v4/objects/users.py (88988e3)

Adding type-hints to gitlab/v4/objects/users.py

- Add type-hints to multiple files in gitlab/v4/objects/ (8b75a77)

Add and/or check type-hints for the following files gitlab.v4.objects.access_requests gitlab.v4.objects.applications gitlab.v4.objects.broadcast_messages gitlab.v4.objects.deployments gitlab.v4.objects.keys gitlab.v4.objects.merge_trains gitlab.v4.objects.namespaces gitlab.v4.objects.pages gitlab.v4.objects.personal_access_tokens gitlab.v4.objects.project_access_tokens gitlab.v4.objects.tags gitlab.v4.objects.templates gitlab.v4.objects.triggers

Add a 'get' method with the correct type for Managers derived from GetMixin.

- Add type-hints to setup.py and check with mypy (06184da)
- Attempt to be more informative for missing attributes (1839c9e)

A commonly reported issue from users on Gitter is that they get an AttributeError for an attribute that should be present. This is often caused due to the fact that they used the `list()` method to retrieve the object and objects retrieved this way often only have a subset of the full data.

Add more details in the AttributeError message that explains the situation to users. This will hopefully allow them to resolve the issue.

Update the FAQ in the docs to add a section discussing the issue.

Closes #1138

- Attempt to fix flaky functional test (487b9a8)

Add an additional check to attempt to solve the flakiness of the test_merge_request_should_remove_source_branch() test.

- Check setup.py with mypy (77cb7a8)

Prior commit 06184daafd5010ba40bb39a0768540b7e98bd171 fixed the type-hints for setup.py. But missed removing 'setup' from the exclude list in pyproject.toml for mypy checks.

Remove 'setup' from the exclude list in pyproject.toml from mypy checks.

- Clean up install docs (a5d8b7f)
- Convert to using type-annotations for managers (d8de4dc)

Convert our manager usage to be done via type annotations.

Now to define a manager to be used in a RESTObject subclass can simply do: class ExampleClass(CRUDMixin, RESTObject): my_manager: MyManager

Any type-annotation that annotates it to be of type *Manager (with the exception of RESTManager) will cause the manager to be created on the object.

- Correct test_groups.py test (9c878a4)

The test was checking twice if the same group3 was not in the returned list. Should have been checking for group3 and group4.

Also added a test that only skipped one group and checked that the group was not in the returned list and a non-skipped group was in the list.

- Create a 'tests/meta/' directory and put test_mro.py in it (94feb8a)

The 'test_mro.py' file is not really a unit test but more of a 'meta' check on the validity of the code base.

- Enable 'warn_redundant_casts' for mypy (f40e9b3)

Enable 'warn_redundant_casts'for mypy and resolve one issue.

- Enable mypy for tests/meta/* (ba7707f)
- Enable subset of the 'mypy –strict' options that work (a86d049)

Enable the subset of the 'mypy –strict' options that work with no changes to the code.

- Enforce type-hints on most files in gitlab/v4/objects/ (7828ba2)
- Add type-hints to some of the files in gitlab/v4/objects/ * Fix issues detected when adding type-hints * Changed mypy exclusion to explicitly list the 13 files that have not yet had type-hints added.
- Ensure get() methods have correct type-hints (46773a8)

Fix classes which don't have correct 'get()' methods for classes derived from GetMixin.

Add a unit test which verifies that classes have the correct return type in their 'get()' method.

- Ensure reset_gitlab() succeeds (0aa0b27)

Ensure reset_gitlab() succeeds by waiting to make sure everything has been deleted as expected. If the timeout is exceeded fail the test.

Not using `wait_for_sidekiq` as it didn't work. During testing I didn't see any sidekiq processes as being busy even though not everything was deleted.

- Fix functional test failure if config present (c8256a5)

Fix functional test failure if config present and configured with token.

Closes: #1791

- Fix issue with adding type-hints to 'manager' attribute (9a451a8)

When attempting to add type-hints to the the 'manager' attribute into a RESTObject derived class it would break things.

This was because our auto-manager creation code would automatically add the specified annotated manager to the 'manager' attribute. This breaks things.

Now check in our auto-manager creation if our attribute is called 'manager'. If so we ignore it.

- Fix pylint error "expression-not-assigned" (a90eb23)

Fix pylint error "expression-not-assigned" and remove check from the disabled list.

And I personally think it is much more readable now and is less lines of code.

- Fix renovate setup for gitlab docker image (49af15b)
- Fix type-check issue shown by new requests-types (0ee9aa4)

types-requests==2.25.9 changed a type-hint. Update code to handle this change.

- Fix typo in MR documentation (2254222)
- Fix unit test if config file exists locally (c80b3b7)

Closes #1764

- Generate artifacts for the docs build in the CI (85b43ae)

When building the docs store the created documentation as an artifact so that it can be viewed.

This will create a html-docs.zip file which can be downloaded containing the contents of the `build/sphinx/html/` directory. It can be downloaded, extracted, and then viewed. This can be useful in reviewing changes to the documentation.

See https://github.com/actions/upload-artifact for more information on how this works.

- Github workflow: cancel prior running jobs on new push (fd81569)

If new new push is done to a pull-request, then cancel any already running github workflow jobs in order to conserve resources.

- Have renovate upgrade black version (#1700, 21228cd)

renovate is not upgrading the `black` package. There is an open issue[1] about this.

Also change .commitlintrc.json to allow 200 character footer lines in the commit message. Otherwise would be forced to split the URL across multiple lines making it un-clickable :(

Use suggested work-arounds from: https://github.com/renovatebot/renovate/issues/7167#issuecomment-904106838 https://github.com/scop/bash-completion/blob/e7497f6ee8232065ec11450a52a1f244f345e2c6/renovate.json#L34-L38

[1] https://github.com/renovatebot/renovate/issues/7167

- Improve type-hinting for managers (c9b5d3b)

The 'managers' are dynamically created. This unfortunately means that we don't have any type-hints for them and so editors which understand type-hints won't know that they are valid attributes.

- Add the type-hints for the managers we define. * Add a unit test that makes sure that the type-hints and the '_managers' attribute are kept in sync with each other. * Add unit test that makes sure specified managers in '_managers' have a name ending in 'Managers' to keep with current convention. * Make RESTObject._managers always present with a default value of None. * Fix a type-issue revealed now that mypy knows what the type is
- Remove '# type: ignore' for new mypy version (34a5f22)

mypy 0.920 now understands the type of 'http.client.HTTPConnection.debuglevel' so we remove the 'type: ignore' comment to make mypy pass

- Remove duplicate/no-op tests from meta/test_ensure_type_hints (a2f59f4)

Before we were generating 725 tests for the meta/test_ensure_type_hints.py tests. Which isn't a huge concern as it was fairly fast. But when we had a failure we would usually get two failures for each problem as the same test was being run multiple times.

Changed it so that: 1. Don't add tests that are not for *Manager classes 2. Use a set so that we don't have duplicate tests.

After doing that our generated test count in meta/test_ensure_type_hints.py went from 725 to 178 tests.

Additionally removed the parsing of `pyproject.toml` to generate files to ignore as we have finished adding type-hints to all files in gitlab/v4/objects/. This also means we no longer use the toml library so remove installation of `types-toml`.

To determine the test count the following command was run: $ tox -e py39 – -k test_ensure_type_hints

- Remove pytest-console-scripts specific config (e80dcb1)

Remove the pytest-console-scripts specific config from the global '[pytest]' config section.

Use the command line option `--script-launch-mode=subprocess`

Closes #1713

- Rename `master` branch to `main` (545f8ed)

BREAKING CHANGE: As of python-gitlab 3.0.0, the default branch for development has changed from `master` to `main`.

- Run pre-commit on changes to the config file (5f10b3b)

If .pre-commit-config.yaml or .github/workflows/pre_commit.yml are updated then run pre-commit.

- Set pre-commit mypy args to empty list (b67a6ad)

https://github.com/pre-commit/mirrors-mypy/blob/master/.pre-commit-hooks.yaml

Sets some default args which seem to be interfering with things. Plus we set all of our args in the `pyproject.toml` file.

- Skip a functional test if not using >= py3.9 (ac9b595)

One of the tests requires Python 3.9 or higher to run. Mark the test to be skipped if running Python less than 3.9.

- Update version in docker-compose.yml (79321aa)

When running with docker-compose on Ubuntu 20.04 I got the error:

$ docker-compose up ERROR: The Compose file './docker-compose.yml' is invalid because:

networks.gitlab-network value Additional properties are not allowed ('name' was unexpected)

Changing the version in the docker-compose.yml file fro '3' to '3.5' resolved the issue.

- Use constants from gitlab.const module (6b8067e)

Have code use constants from the gitlab.const module instead of from the top-level gitlab module.

- **api**: Temporarily remove topic delete endpoint (e3035a7)

It is not yet available upstream.

- **ci**: Add workflow to lock old issues (a7d64fe)
- **ci**: Enable renovate for pre-commit (1ac4329)
- **ci**: Wait for all coverage jobs before posting comment (c7fdad4)
- **deps**: Update dependency argcomplete to v2 (c6d7e9a)
- **deps**: Update dependency black to v21 (5bca87c)
- **deps**: Update dependency black to v21.12b0 (ab841b8)
- **deps**: Update dependency flake8 to v4 (79785f0)
- **deps**: Update dependency isort to v5.10.0 (ae62468)
- **deps**: Update dependency isort to v5.10.1 (2012975)
- **deps**: Update dependency mypy to v0.920 (a519b2f)
- **deps**: Update dependency mypy to v0.930 (ccf8190)
- **deps**: Update dependency requests to v2.27.0 (f8c3d00)

- **deps**: Update dependency sphinx to v4 (73745f7)
- **deps**: Update dependency sphinx to v4.3.0 (57283fc)
- **deps**: Update dependency sphinx to v4.3.1 (93a3893)
- **deps**: Update dependency sphinx to v4.3.2 (2210e56)
- **deps**: Update dependency types-pyyaml to v5.4.10 (bdb6cb9)
- **deps**: Update dependency types-pyyaml to v6 (0b53c0a)
- **deps**: Update dependency types-pyyaml to v6.0.1 (a544cd5)
- **deps**: Update dependency types-requests to v2.25.12 (205ad5f)
- **deps**: Update dependency types-requests to v2.25.9 (e3912ca)
- **deps**: Update dependency types-requests to v2.26.0 (7528d84)
- **deps**: Update dependency types-requests to v2.26.2 (ac7e329)
- **deps**: Update dependency types-setuptools to v57.4.3 (ec2c68b)
- **deps**: Update pre-commit hook alessandrojcm/commitlint-pre-commit-hook to v6 (fb9110b)
- **deps**: Update pre-commit hook psf/black to v21 (b86e819)
- **deps**: Update pre-commit hook pycqa/flake8 to v4 (98a5592)
- **deps**: Update pre-commit hook pycqa/isort to v5.10.1 (8ac4f4a)
- **deps**: Update python docker tag to v3.10 (b3d6d91)
- **deps**: Update typing dependencies (1f95613)
- **deps**: Update typing dependencies (8d4c953)
- **deps**: Update typing dependencies (4170dbe)
- **deps**: Update typing dependencies (4eb8ec8)
- **deps**: Upgrade gitlab-ce to 14.3.2-ce.0 (5a1678f)
- **deps**: Upgrade mypy pre-commit hook (e19e4d7)
- **docs**: Link to main, not master (af0cb4d)
- **docs**: Load autodoc-typehints module (bd366ab)
- **docs**: Use builtin autodoc hints (5e9c943)
- **objects**: Remove non-existing trigger ownership method (8dc7f40)
- **tests**: Apply review suggestions (381c748)

Documentation

- Add links to the GitLab API docs (e3b5d27)

Add links to the GitLab API docs for merge_requests.py as it contains code which spans two different API documentation pages.

- Consolidate changelogs and remove v3 API docs (90da8ba)
- Correct documentation for updating discussion note (ee66f4a)

Closes #1777

- Correct documented return type (acabf63)

repository_archive() returns 'bytes' not 'str'

https://docs.gitlab.com/ee/api/repositories.html#get-file-archive

Fixes: #1584

- Fix a few typos (7ea4ddc)

There are small typos in: - docs/gl_objects/deploy_tokens.rst - gitlab/base.py - gitlab/mixins.py - gitlab/v4/objects/features.py - gitlab/v4/objects/groups.py - gitlab/v4/objects/packages.py - gitlab/v4/objects/projects.py - gitlab/v4/objects/sidekiq.py - gitlab/v4/objects/todos.py

Fixes: - Should read `treatment` rather than `reatment`. - Should read `transferred` rather than `transfered`. - Should read `registered` rather than `registred`. - Should read `occurred` rather than `occured`. - Should read `overridden` rather than `overriden`. - Should read `marked` rather than `maked`. - Should read `instantiate` rather than `instanciate`. - Should read `function` rather than `fonction`.

- Fix API delete key example (b31bb05)
- Only use type annotations for documentation (b7dde0d)
- Rename documentation files to match names of code files (ee3f865)

Rename the merge request related documentation files to match the code files. This will make it easier to find the documentation quickly.

Rename: `docs/gl_objects/mrs.rst` -> docs/gl_objects/merge_requests.rst docs/gl_objects/mr_approvals.rst -> `docs/gl_objects/merge_request_approvals.rst`

- Switch to Furo and refresh introduction pages (ee6b024)
- Update docs to use gitlab.const for constants (b3b0b5f)

Update the docs to use gitlab.const to access constants.

- Use annotations for return types (79e785e)
- **api**: Clarify job token usage with auth() (3f423ef)

See issue #1620

- **api**: Document the update method for project variables (7992911)
- **pipelines**: Document take_ownership method (69461f6)
- **project**: Remove redundant encoding parameter (fed613f)

Features

- Add delete on package_file object (124667b)
- Add support for `projects.groups.list()` (68ff595)

Add support for `projects.groups.list()` endpoint.

Closes #1717

- Add support for `squash_option` in Projects (a246ce8)

There is an optional `squash_option` parameter which can be used when creating Projects and UserProjects.

Closes #1744

- Allow global retry_transient_errors setup (3b1d3a4)

`retry_transient_errors` can now be set through the Gitlab instance and global configuration

Documentation for API usage has been updated and missing tests have been added.

- Default to gitlab.com if no URL given (8236281)

BREAKING CHANGE: python-gitlab will now default to gitlab.com if no URL is given

- Remove support for Python 3.6, require 3.7 or higher (414009d)

Python 3.6 is End-of-Life (EOL) as of 2021-12 as stated in https://www.python.org/dev/peps/pep-0494/

By dropping support for Python 3.6 and requiring Python 3.7 or higher it allows python-gitlab to take advantage of new features in Python 3.7, which are documented at: https://docs.python.org/3/whatsnew/3.7.html

Some of these new features that may be useful to python-gitlab are: * PEP 563, postponed evaluation of type annotations. * dataclasses: PEP 557 – Data Classes * importlib.resources * PEP 562, customization of access to module attributes. * PEP 560, core support for typing module and generic types. * PEP 565, improved DeprecationWarning handling

BREAKING CHANGE: As of python-gitlab 3.0.0, Python 3.6 is no longer supported. Python 3.7 or higher is required.

- **api**: Add merge request approval state (f41b093)

Add support for merge request approval state

- **api**: Add merge trains (fd73a73)

Add support for merge trains

- **api**: Add project label promotion (6d7c88a)

Adds a mixin that allows the /promote endpoint to be called.

Signed-off-by: Raimund Hook raimund.hook@exfo.com

- **api**: Add project milestone promotion (f068520)

Adds promotion to Project Milestones

Signed-off-by: Raimund Hook raimund.hook@exfo.com

- **api**: Add support for epic notes (7f4edb5)

Added support for notes on group epics

Signed-off-by: Raimund Hook raimund.hook@exfo.com

- **api**: Add support for Topics API (e7559bf)
- **api**: Support file format for repository archive (83dcabf)
- **build**: Officially support and test python 3.10 (c042ddc)
- **cli**: Allow options from args and environment variables (ca58008)

BREAKING-CHANGE: The gitlab CLI will now accept CLI arguments

and environment variables for its global options in addition to configuration file options. This may change behavior for some workflows such as running inside GitLab CI and with certain environment variables configured.

- **cli**: Do not require config file to run CLI (92a893b)

BREAKING CHANGE: A config file is no longer needed to run the CLI. python-gitlab will default to https://gitlab.com with no authentication if there is no config file provided. python-gitlab will now also only look for configuration in the provided PYTHON_GITLAB_CFG path, instead of merging it with user- and system-wide config files. If the environment variable is defined and the file cannot be opened, python-gitlab will now explicitly fail.

- **docker**: Remove custom entrypoint from image (80754a1)

This is no longer needed as all of the configuration is handled by the CLI and can be passed as arguments.

- **objects**: List starred projects of a user (47a5606)

- **objects**: Support Create and Revoke personal access token API (e19314d)
- **objects**: Support delete package files API (4518046)

Refactoring

- Deprecate accessing constants from top-level namespace (c0aa0e1)

We are planning on adding enumerated constants into gitlab/const.py, but if we do that than they will end up being added to the top-level gitlab namespace. We really want to get users to start using `gitlab.const.` to access the constant values in the future.

Add the currently defined constants to a list that should not change. Use a module level **getattr** function so that we can deprecate access to the top-level constants.

Add a unit test which verifies we generate a warning when accessing the top-level constants.

- Use f-strings for string formatting (7925c90)
- Use new-style formatting for named placeholders (c0d8810)
- **objects**: Remove deprecated branch protect methods (9656a16)

BREAKING CHANGE: remove deprecated branch protect methods in favor of the more complete protected branches API.

- **objects**: Remove deprecated constants defined in objects (3f320af)

BREAKING CHANGE: remove deprecated constants defined in gitlab.v4.objects, and use only gitlab.const module

- **objects**: Remove deprecated members.all() method (4d7b848)

BREAKING CHANGE: remove deprecated members.all() method in favor of members_all.list()

- **objects**: Remove deprecated pipelines() method (c4f5ec6)

BREAKING CHANGE: remove deprecated pipelines() methods in favor of pipelines.list()

- **objects**: Remove deprecated project.issuesstatistics (ca7777e)

BREAKING CHANGE: remove deprecated project.issuesstatistics in favor of project.issues_statistics

- **objects**: Remove deprecated tag release API (2b8a94a)

BREAKING CHANGE: remove deprecated tag release API. This was removed in GitLab 14.0

Testing

- Drop httmock dependency in test_gitlab.py (c764bee)
- Reproduce missing pagination headers in tests (501f9a1)
- **api**: Fix current user mail count in newer gitlab (af33aff)
- **build**: Add smoke tests for sdist & wheel package (b8a47ba)
- **cli**: Improve basic CLI coverage (6b892e3)

7.10.58 v2.10.1 (2021-08-28)

Bug Fixes

- **deps**: Upgrade requests to 2.25.0 (see CVE-2021-33503) (ce995b2)
- **mixins**: Improve deprecation warning (57e0187)

Also note what should be changed

Chores

- Define root dir in mypy, not tox (7a64e67)
- Fix mypy pre-commit hook (bd50df6)
- **deps**: Group typing requirements with mypy additional_dependencies (38597e7)
- **deps**: Update codecov/codecov-action action to v2 (44f4fb7)
- **deps**: Update dependency isort to v5.9.3 (ab46e31)
- **deps**: Update dependency types-pyyaml to v5.4.7 (ec8be67)
- **deps**: Update dependency types-pyyaml to v5.4.8 (2ae1dd7)
- **deps**: Update dependency types-requests to v2.25.1 (a2d133a)
- **deps**: Update dependency types-requests to v2.25.2 (4782678)
- **deps**: Update precommit hook pycqa/isort to v5.9.3 (e1954f3)
- **deps**: Update typing dependencies (34fc210)
- **deps**: Update wagoid/commitlint-github-action action to v4 (ae97196)

Documentation

- **mergequests**: Gl.mergequests.list documentation was missleading (5b5a7bc)

7.10.59 v2.10.0 (2021-07-28)

Bug Fixes

- **api**: Do not require Release name for creation (98cd03b)

Stop requiring a `name` attribute for creating a Release, since a release name has not been required since GitLab 12.5.

Chores

- **deps**: Update dependency isort to v5.9.2 (d5dcf1c)
- **deps**: Update dependency requests to v2.26.0 (d3ea203)
- **deps**: Update precommit hook pycqa/isort to v5.9.2 (521cddd)

Documentation

- Add example for mr.merge_ref (b30b8ac)

Signed-off-by: Matej Focko mfocko@redhat.com

- **project**: Add example on getting a single project using name with namespace (ef16a97)
- **readme**: Move contributing docs to CONTRIBUTING.rst (edf49a3)

Move the Contributing section of README.rst to CONTRIBUTING.rst, so it is recognized by GitHub and shown when new contributors make pull requests.

Features

- **api**: Add `name_regex_keep` attribute in `delete_in_bulk()` (e49ff3f)
- **api**: Add merge_ref for merge requests (1e24ab2)

Support merge_ref on merge requests that returns commit of attempted merge of the MR.

Signed-off-by: Matej Focko mfocko@redhat.com

Testing

- **functional**: Add mr.merge_ref tests (a9924f4)
- Add test for using merge_ref on non-merged MR - Add test for using merge_ref on MR with conflicts

Signed-off-by: Matej Focko mfocko@redhat.com

7.10.60 v2.9.0 (2021-06-28)

Chores

- Add new required type packages for mypy (a7371e1)

New version of mypy flagged errors for missing types. Install the recommended type-* packages that resolve the issues.

- Add type-hints to gitlab/v4/objects/projects.py (872dd6d)

Adding type-hints to gitlab/v4/objects/projects.py

- Skip EE test case in functional tests (953f207)
- **deps**: Update dependency isort to v5.9.1 (0479dba)
- **deps**: Update dependency mypy to v0.902 (19c9736)
- **deps**: Update dependency mypy to v0.910 (02a56f3)
- **deps**: Update dependency types-pyyaml to v0.1.8 (e566767)
- **deps**: Update dependency types-pyyaml to v0.1.9 (1f5b3c0)
- **deps**: Update dependency types-pyyaml to v5 (5c22634)
- **deps**: Update dependency types-requests to v0.1.11 (6ba629c)
- **deps**: Update dependency types-requests to v0.1.12 (f84c2a8)
- **deps**: Update dependency types-requests to v0.1.13 (c3ddae2)
- **deps**: Update dependency types-requests to v2 (a81a926)
- **deps**: Update precommit hook pycqa/isort to v5.9.1 (c57ffe3)

Documentation

- Make Gitlab class usable for intersphinx (8753add)
- **release**: Add update example (6254a5f)
- **tags**: Remove deprecated functions (1b1a827)

Features

- **api**: Add group hooks (4a7e9b8)
- **api**: Add MR pipeline manager in favor of pipelines() method (954357c)
- **api**: Add support for creating/editing reviewers in project merge requests (676d1f6)
- **api**: Remove responsibility for API inconsistencies for MR reviewers (3d985ee)
- **release**: Allow to update release (b4c4787)

Release API now supports PUT.

Testing

- **releases**: Add unit-tests for release update (5b68a5a)
- **releases**: Integration for release PUT (13bf61d)

7.10.61 v2.8.0 (2021-06-10)

Bug Fixes

- Add a check to ensure the MRO is correct (565d548)

Add a check to ensure the MRO (Method Resolution Order) is correct for classes in gitlab.v4.objects when doing type-checking.

An example of an incorrect definition: class ProjectPipeline(RESTObject, RefreshMixin, ObjectDeleteMixin): ^^^^^^^^^^ This should be at the end.

Correct way would be: class ProjectPipeline(RefreshMixin, ObjectDeleteMixin, RESTObject): Correctly at the end ^^^^^^^^^^

Also fix classes which have the issue.

- Catch invalid type used to initialize RESTObject (c7bcc25)

Sometimes we have errors where we don't get a dictionary passed to RESTObject.**init**() method. This breaks things but in confusing ways.

Check in the **init**() method and raise an exception if it occurs.

- Change mr.merge() to use 'post_data' (cb6a3c6)

MR https://github.com/python-gitlab/python-gitlab/pull/1121 changed mr.merge() to use 'query_data'. This appears to have been wrong.

From the Gitlab docs they state it should be sent in a payload body https://docs.gitlab.com/ee/api/README.html#request-payload since mr.merge() is a PUT request.

> Request Payload
>
> API Requests can use parameters sent as query strings or as a > payload body. GET requests usually send a query string, while PUT > or POST requests usually send the payload body

Fixes: #1452

Related to: #1120

- Ensure kwargs are passed appropriately for ObjectDeleteMixin (4e690c2)
- Functional project service test (#1500, 093db9d)

chore: fix functional project service test

- Iids not working as a list in projects.issues.list() (45f806c)

Set the 'iids' values as type ListAttribute so it will pass the list as a comma-separated string, instead of a list.

Add a functional test.

Closes: #1407

- **cli**: Add missing list filter for jobs (b3d1c26)
- **cli**: Fix parsing CLI objects to classnames (4252070)

- **objects**: Add missing group attributes (d20ff4f)
- **objects**: Allow lists for filters for in all objects (603a351)
- **objects**: Return server data in cancel/retry methods (9fed061)

Chores

- Add a functional test for issue #1120 (7d66115)

Going to switch to putting parameters from in the query string to having them in the 'data' body section. Add a functional test to make sure that we don't break anything.

https://github.com/python-gitlab/python-gitlab/issues/1120

- Add a merge_request() pytest fixture and use it (8be2838)

Added a pytest.fixture for merge_request(). Use this fixture in tools/functional/api/test_merge_requests.py

- Add an isort tox environment and run isort in CI (dda646e)
- Add an isort tox environment * Run the isort tox environment using –check in the Github CI

https://pycqa.github.io/isort/

- Add functional test mr.merge() with long commit message (cd5993c)

Functional test to show that https://github.com/python-gitlab/python-gitlab/issues/1452 is fixed.

Added a functional test to ensure that we can use large commit message (10_000+ bytes) in mr.merge()

Related to: #1452

- Add missing linters to pre-commit and pin versions (85bbd1a)
- Add missing optional create parameter for approval_rules (06a6001)

Add missing optional create parameter ('protected_branch_ids') to the project approvalrules.

https://docs.gitlab.com/ee/api/merge_request_approvals.html#create-project-level-rule

- Add type-hints to gitlab/v4/cli.py (2673af0)
- Add type-hints to gitlab/v4/cli.py * Add required type-hints to other files based on adding type-hints to gitlab/v4/cli.py
- Apply suggestions (fe7d19d)
- Apply typing suggestions (a11623b)

Co-authored-by: John Villalovos john@sodarock.com

- Clean up tox, pre-commit and requirements (237b97c)
- Correct a type-hint (046607c)
- Fix import ordering using isort (f3afd34)

Fix the import ordering using isort.

https://pycqa.github.io/isort/

- Have black run at the top-level (429d6c5)

This will ensure everything is formatted with black, including setup.py.

- Have flake8 check the entire project (ab343ef)

Have flake8 run at the top-level of the projects instead of just the gitlab directory.

- Make certain dotfiles searchable by ripgrep (e4ce078)

By explicitly NOT excluding the dotfiles we care about to the .gitignore file we make those files searchable by tools like ripgrep.

By default dotfiles are ignored by ripgrep and other search tools (not grep)

- Make Get.*Mixin._optional_get_attrs always present (3c1a0b3)

Always create GetMixin/GetWithoutIdMixin._optional_get_attrs attribute with a default value of tuple()

This way we don't need to use hasattr() and we will know the type of the attribute.

- Move 'gitlab/tests/' dir to 'tests/unit/' (1ac0722)

Move the 'gitlab/tests/' directory to 'tests/unit/' so we have all the tests located under the 'tests/' directory.

- Mypy: Disallow untyped definitions (6aef2da)

Be more strict and don't allow untyped definitions on the files we check.

Also this adds type-hints for two of the decorators so that now functions/methods decorated by them will have their types be revealed correctly.

- Remove commented-out print (0357c37)
- Rename 'tools/functional/' to 'tests/functional/' (502715d)

Rename the 'tools/functional/' directory to 'tests/functional/'

This makes more sense as these are functional tests and not tools.

This was dicussed in: https://github.com/python-gitlab/python-gitlab/discussions/1468

- Simplify functional tests (df9b5f9)

Add a helper function to have less code duplication in the functional testing.

- Sync create and update attributes for Projects (0044bd2)

Sync the create attributes with: https://docs.gitlab.com/ee/api/projects.html#create-project

Sync the update attributes with documentation at: https://docs.gitlab.com/ee/api/projects.html#edit-project

As a note the ordering of the attributes was done to match the ordering of the attributes in the documentation.

Closes: #1497

- Use built-in function issubclass() instead of getmro() (81f6386)

Code was using inspect.getmro() to replicate the functionality of the built-in function issubclass()

Switch to using issubclass()

- **ci**: Automate releases (0ef497e)
- **ci**: Ignore .python-version from pyenv (149953d)
- **ci**: Ignore debug and type_checking in coverage (885b608)
- **ci**: Use admin PAT for release workflow (d175d41)
- **deps**: Update dependency docker-compose to v1.29.2 (fc241e1)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.11.2-ce.0 (434d15d)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.11.3-ce.0 (f0b52d8)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.11.4-ce.0 (4223269)
- **deps**: Update precommit hook alessandrojcm/commitlint-pre-commit-hook to v5 (9ff349d)

- **docs**: Fix import order for readthedocs build (c3de1fb)

Code Style

- Clean up test run config (dfa40c1)

Documentation

- Fail on warnings during sphinx build (cbd4d52)

This is useful when docs aren't included in the toctree and don't show up on RTD.

- Fix typo in http_delete docstring (5226f09)
- **api**: Add behavior in local attributes when updating objects (38f65e8)

Features

- Add code owner approval as attribute (fdc46ba)

The python API was missing the field code_owner_approval_required as implemented in the GitLab REST API.

- Add feature to get inherited member for project/group (e444b39)
- Add keys endpoint (a81525a)
- Add support for lists of integers to ListAttribute (115938b)

Previously ListAttribute only support lists of integers. Now be more flexible and support lists of items which can be coerced into strings, for example integers.

This will help us fix issue #1407 by using ListAttribute for the 'iids' field.

- Indicate that we are a typed package (e4421ca)

By adding the file: py.typed it indicates that python-gitlab is a typed package and contains type-hints.

https://www.python.org/dev/peps/pep-0561/

- **api**: Add deployment mergerequests interface (fbbc0d4)
- **objects**: Add pipeline test report support (ee9f96e)
- **objects**: Add support for billable members (fb0b083)
- **objects**: Add support for descendant groups API (1b70580)
- **objects**: Add support for generic packages API (79d88bd)
- **objects**: Add support for Group wikis (#1484, 74f5e62)

feat(objects): add support for Group wikis

- **objects**: Support all issues statistics endpoints (f731707)

Testing

- **api**: Fix issues test (8e5b0de)

Was incorrectly using the issue 'id' vs 'iid'.

- **cli**: Add more real class scenarios (8cf5031)
- **cli**: Replace assignment expression (11ae11b)

This is a feature added in 3.8, removing it allows for the test to run with lower python versions.

- **functional**: Add test for skip_groups list filter (a014774)

- **functional**: Explicitly remove deploy tokens on reset (19a55d8)

Deploy tokens would remain in the instance if the respective project or group was deleted without explicitly revoking the deploy tokens first.

- **functional**: Force delete users on reset (8f81456)

Timing issues between requesting group deletion and GitLab enacting that deletion resulted in errors while attempting to delete a user which was the sole owner of said group (see: test_groups). Pass the 'hard_delete' parameter to ensure user deletion.

- **functional**: Optionally keep containers running post-tests (4c475ab)

Additionally updates token creation to make use of `first_or_create()`, to avoid errors from the script caused by GitLab constraints preventing duplicate tokens with the same value.

- **functional**: Start tracking functional test coverage (f875786)

7.10.62 v2.7.1 (2021-04-26)

Bug Fixes

- **files**: Do not url-encode file paths twice (8e25cec)

7.10.63 v2.7.0 (2021-04-25)

Bug Fixes

- Argument type was not a tuple as expected (062f8f6)

While adding type-hints mypy flagged this as an issue. The third argument to register_custom_action is supposed to be a tuple. It was being passed as a string rather than a tuple of strings.

- Better real life token lookup example (9ef8311)
- Checking if RESTManager._from_parent_attrs is set (8224b40)

Prior to commit 3727cbd21fc40b312573ca8da56e0f6cf9577d08 RESTManager._from_parent_attrs did not exist unless it was explicitly set. But commit 3727cbd21fc40b312573ca8da56e0f6cf9577d08 set it to a default value of {}.

So the checks using hasattr() were no longer valid.

Update the checks to check if RESTManager._from_parent_attrs has a value.

- Correct ProjectFile.decode() documentation (b180baf)

ProjectFile.decode() returns 'bytes' and not 'str'.

Update the method's doc-string and add a type-hint.

ProjectFile.decode() returns the result of a call to base64.b64decode()

The docs for that function state it returns 'bytes': https://docs.python.org/3/library/base64.html#base64.b64decode

Fixes: #1403

- Correct some type-hints in gitlab/mixins.py (8bd3124)

Commit baea7215bbbe07c06b2ca0f97a1d3d482668d887 introduced type-hints for gitlab/mixins.py.

After starting to add type-hints to gitlab/v4/objects/users.py discovered a few errors.

Main error was using '=' instead of ':'. For example: _parent = Optional[...] should be _parent: Optional[...]

Resolved those issues.

- Extend wait timeout for test_delete_user() (19fde8e)

Have been seeing intermittent failures of the test_delete_user() functional test. Have made the following changes to hopefully resolve the issue and if it still fails to know better why the failure occurred.

- Extend the wait timeout for test_delete_user() from 30 to 60 tries of 0.5 seconds each.
- Modify wait_for_sidekiq() to return True if sidekiq process terminated. Return False if the timeout expired.
- Modify wait_for_sidekiq() to loop through all processes instead of assuming there is only one process. If all processes are not busy then return.
- Modify wait_for_sidekiq() to sleep at least once before checking for processes being busy.
- Check for True being returned in test_delete_user() call to wait_for_sidekiq()
- Handle tags like debian/2%2.6-21 as identifiers (b4dac5c)

Git refnames are relatively free-form and can contain all sort for special characters, not just / and #, see http://git-scm.com/docs/git-check-ref-format

In particular, Debian's DEP-14 standard for storing packaging in git repositories mandates the use of the % character in tags in some cases like `debian/2%2.6-21`.

Unfortunately python-gitlab currently only escapes / to `%2F` and in some cases # to `%23`. This means that when using the commit API to retrieve information about the `debian/2%2.6-21` tag only the slash is escaped before being inserted in the URL path and the % is left untouched, resulting in something like `/api/v4/projects/123/repository/commits/debian%2F2%2.6-21`. When urllib3 seees that it detects the invalid % escape and then urlencodes the whole string, resulting in `/api/v4/projects/123/repository/commits/debian%252F2%252.6-21`, where the original / got escaped twice and produced `%252F`.

To avoid the issue, fully urlencode identifiers and parameters to avoid the urllib3 auto-escaping in all cases.

Signed-off-by: Emanuele Aina emanuele.aina@collabora.com

- Handling config value in _get_values_from_helper (9dfb4cd)
- Honor parameter value passed (c2f8f0e)

Gitlab allows setting the defaults for MR to delete the source. Also the inline help of the CLI suggest that a boolean is expected, but no matter what value you set, it will always delete.

- Let the homedir be expanded in path of helper (fc7387a)
- Linting issues and test (b04dd2c)
- Make secret helper more user friendly (fc2798f)
- Only add query_parameters to GitlabList once (ca2c3c9)

Fixes #1386

- Only append kwargs as query parameters (b9ecc9a)

Some arguments to `http_request` were being read from kwargs, but kwargs is where this function creates query parameters from, by default. In the absence of a `query_parameters` param, the function would construct URLs with query parameters such as `retry_transient_errors=True` despite those parameters having no meaning to the API to which the request was sent.

This change names those arguments that are specific to `http_request` so that they do not end up as query parameters read from kwargs.

- Remove duplicate class definitions in v4/objects/users.py (7c4e625)

The classes UserStatus and UserStatusManager were each declared twice. Remove the duplicate declarations.

- Test_update_group() dependency on ordering (e78a8d6)

Since there are two groups we can't depend on the one we changed to always be the first one returned.

Instead fetch the group we want and then test our assertion against that group.

- Tox pep8 target, so that it can run (f518e87)

Previously running the pep8 target would fail as flake8 was not installed.

Now install flake8 for the pep8 target.

NOTE: Running the pep8 target fails as there are many warnings/errors.

But it does allow us to run it and possibly work on reducing these warnings/errors in the future.

In addition, add two checks to the ignore list as black takes care of formatting. The two checks added to the ignore list are: * E501: line too long * W503: line break before binary operator

- Undefined name errors (48ec9e0)

Discovered that there were some undefined names.

- Update doc for token helper (3ac6fa1)
- Update user's bool data and avatar (3ba27ff)

If we want to update email, avatar and do not send email confirmation change (`skip_reconfirmation = True`), `MultipartEncoder` will try to encode everything except None and bytes. So it tries to encode bools. Casting bool's values to their stringified int representation fix it.

- Wrong variable name (15ec41c)

Discovered this when I ran flake8 on the file. Unfortunately I was the one who introduced this wrong variable name :(

- **objects**: Add single get endpoint for instance audit events (c3f0a6f)
- **types**: Prevent **dir** from producing duplicates (5bf7525)

Chores

- Add _create_attrs & _update_attrs to RESTManager (147f05d)

Add the attributes: _create_attrs and _update_attrs to the RESTManager class. This is so that we stop using getattr() if we don't need to.

This also helps with type-hints being available for these attributes.

- Add additional type-hints for gitlab/base.py (ad72ef3)

Add type-hints for the variables which are set via self.**dict**

mypy doesn't see them when they are assigned via self.**dict**. So declare them in the class definition.

- Add and fix some type-hints in gitlab/client.py (8837207)

Was able to figure out better type-hints for gitlab/client.py

- Add test (f8cf1e1)
- Add type hints to gitlab/base.py (3727cbd)
- Add type hints to gitlab/base.py:RESTManager (9c55593)

Add some additional type hints to gitlab/base.py

- Add type hints to gitlab/utils.py (acd9294)
- Add type-hints for gitlab/mixins.py (baea721)

- Added type-hints for gitlab/mixins.py * Changed use of filter with a lambda expression to list-comprehension. mypy was not able to understand the previous code. Also list-comprehension is better :)
- Add type-hints to gitlab/cli.py (10b7b83)
- Add type-hints to gitlab/client.py (c9e5b4f)

Adding some initial type-hints to gitlab/client.py

- Add type-hints to gitlab/config.py (213e563)
- Add type-hints to gitlab/const.py (a10a777)
- Bump version to 2.7.0 (34c4052)
- Del 'import *' in gitlab/v4/objects/project_access_tokens.py (9efbe12)

Remove usage of 'import *' in gitlab/v4/objects/project_access_tokens.py.

- Disallow incomplete type defs (907634f)

Don't allow a partially annotated function definition. Either none of the function is annotated or all of it must be.

Update code to ensure no-more partially annotated functions.

Update gitlab/cli.py with better type-hints. Changed Tuple[Any, ...] to Tuple[str, ...]

- Explicitly import gitlab.v4.objects/cli (233b79e)

As we only support the v4 Gitlab API, explicitly import gitlab.v4.objects and gitlab.v4.clie instead of dynamically importing it depending on the API version.

This has the added benefit of mypy being able to type check the Gitlab **init**() function as currently it will fail if we enable type checking of **init**() it will fail.

Also, this also helps by not confusing tools like pyinstaller/cx_freeze with dynamic imports so you don't need hooks for standalone executables. And according to https://docs.gitlab.com/ee/api/,

"GraphQL co-exists with the current v4 REST API. If we have a v5 API, this should be a compatibility layer on top of GraphQL."

- Fix E711 error reported by flake8 (630901b)

E711: Comparison to none should be 'if cond is none:'

https://www.flake8rules.com/rules/E711.html

- Fix E712 errors reported by flake8 (83670a4)

E712: Comparison to true should be 'if cond is true:' or 'if cond:'

https://www.flake8rules.com/rules/E712.html

- Fix E741/E742 errors reported by flake8 (380f227)

Fixes to resolve errors for: https://www.flake8rules.com/rules/E741.html Do not use variables named 'I', 'O', or 'l' (E741)

https://www.flake8rules.com/rules/E742.html Do not define classes named 'I', 'O', or 'l' (E742)

- Fix F401 errors reported by flake8 (ff21eb6)

F401: Module imported but unused

https://www.flake8rules.com/rules/F401.html

- Fix F841 errors reported by flake8 (40f4ab2)

Local variable name is assigned to but never used

https://www.flake8rules.com/rules/F841.html

- Fix package file test naming (8c80268)
- Fix typo in mr events (c5e6fb3)
- Have _create_attrs & _update_attrs be a namedtuple (aee1f49)

Convert _create_attrs and _update_attrs to use a NamedTuple (RequiredOptional) to help with code readability. Update all code to use the NamedTuple.

- Import audit events in objects (35a190c)
- Improve type-hints for gitlab/base.py (cbd43d0)

Determined the base class for obj_cls and adding type-hints for it.

- Make _types always present in RESTManager (924f83e)

We now create _types = {} in RESTManager class.

By making _types always present in RESTManager it makes the code simpler. We no longer have to do: types = getattr(self, "_types", {})

And the type checker now understands the type.

- Make lint happy (7a7c9fd)
- Make lint happy (b5f43c8)
- Make lint happy (732e49c)
- Make ListMixin._list_filters always present (8933113)

Always create ListMixin._list_filters attribute with a default value of tuple().

This way we don't need to use hasattr() and we will know the type of the attribute.

- Make RESTObject._short_print_attrs always present (6d55120)

Always create RESTObject._short_print_attrs with a default value of None.

This way we don't need to use hasattr() and we will know the type of the attribute.

- Put assert statements inside 'if TYPE_CHECKING:' (b562458)

To be safe that we don't assert while running, put the assert statements, which are used by mypy to check that types are correct, inside an 'if TYPE_CHECKING:' block.

Also, instead of asserting that the item is a dict, instead assert that it is not a requests.Response object. Theoretically the JSON could return as a list or dict, though at this time we are assuming a dict.

- Remove import of gitlab.utils from **init**.py (39b9183)

Initially when extracting out the gitlab/client.py code we tried to remove this but functional tests failed.

Later we fixed the functional test that was failing, so now remove the unneeded import.

- Remove Python 2 code (b5d4e40)

httplib is a Python 2 library. It was renamed to http.client in Python 3.

https://docs.python.org/2.7/library/httplib.html

- Remove unused ALLOWED_KEYSET_ENDPOINTS variable (3d5d5d8)

The variable ALLOWED_KEYSET_ENDPOINTS was added in commit f86ef3bbdb5bffa1348a802e62b281d3f31d33ad.

Then most of that commit was removed in commit e71fe16b47835aa4db2834e98c7ffc6bdec36723, but ALLOWED_KEYSET_ENDPOINTS was missed.

- Remove unused function _construct_url() (009d369)

The function _construct_url() was used by the v3 API. All usage of the function was removed in commit fe89b949922c028830dd49095432ba627d330186

- Remove unused function sanitize_parameters() (443b934)

The function sanitize_parameters() was used when the v3 API was in use. Since v3 API support has been removed there are no more users of this function.

- Remove usage of 'from … import *' (c83eaf4)

In gitlab/v4/objects/*.py remove usage of: * from gitlab.base import * * from gitlab.mixins import *

Change them to: * from gitlab.base import CLASS_NAME * from gitlab.mixins import CLASS_NAME

Programmatically update code to explicitly import needed classes only.

After the change the output of: $ flake8 gitlab/v4/objects/*py | grep 'REST|Mixin'

Is empty. Before many messages about unable to determine if it was a valid name.

- Remove usage of 'from … import *' in client.py (bf0c8c5)

In gitlab/client.py remove usage of: * from gitlab.const import * * from gitlab.exceptions import *

Change them to: * import gitlab.const * import gitlab.exceptions

Update code to explicitly reference things in gitlab.const and gitlab.exceptions

A flake8 run no longer lists any undefined variables. Before it listed possible undefined variables.

- Remove usage of getattr() (2afd18a)

Remove usage of getattr(self, "_update_uses_post", False)

Instead add it to class and set default value to False.

Add a tests that shows it is set to True for the ProjectMergeRequestApprovalManager and ProjectApprovalManager classes.

- **api**: Move repository endpoints into separate module (1ed154c)
- **ci**: Deduplicate PR jobs (63918c3)
- **config**: Allow simple commands without external script (91ffb8e)
- **deps**: Update dependency docker-compose to v1.28.3 (2358d48)
- **deps**: Update dependency docker-compose to v1.28.4 (8938484)
- **deps**: Update dependency docker-compose to v1.28.5 (f4ab558)
- **deps**: Update dependency docker-compose to v1.28.6 (46b05d5)
- **deps**: Update dependency docker-compose to v1.29.1 (a89ec43)
- **deps**: Update dependency sphinx to v3.4.3 (37c992c)
- **deps**: Update dependency sphinx to v3.5.0 (188c5b6)
- **deps**: Update dependency sphinx to v3.5.1 (f916f09)
- **deps**: Update dependency sphinx to v3.5.2 (9dee5c4)

- **deps**: Update dependency sphinx to v3.5.4 (a886d28)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.10.0-ce.0 (5221e33)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.10.1-ce.0 (1995361)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.10.3-ce.0 (eabe091)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.11.0-ce.0 (711896f)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.11.1-ce.0 (3088714)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.8.2-ce.0 (7c12038)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.8.3-ce.0 (e6c20f1)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.8.4-ce.0 (832cb88)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.9.0-ce.0 (3aef19c)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.9.1-ce.0 (f6fd995)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.9.2-ce.0 (933ba52)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.9.3-ce.0 (2ddf45f)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.9.4-ce.0 (939f769)
- **deps**: Update precommit hook alessandrojcm/commitlint-pre-commit-hook to v4 (505a8b8)
- **deps**: Update wagoid/commitlint-github-action action to v3 (b3274cf)
- **objects**: Make Project refreshable (958a6aa)

Helps getting the real state of the project from the server.

- **objects**: Remove noisy deprecation warning for audit events (2953642)

It's mostly an internal thing anyway and can be removed in 3.0.0

- **tests**: Remove unused URL segment (66f0b6c)

Documentation

- Add docs and examples for custom user agent (a69a214)
- Add information about the gitter community (6ff67e7)

Add a section in the README.rst about the gitter community. The badge already exists and is useful but very easy to miss.

- Change travis-ci badge to githubactions (2ba5ba2)
- **api**: Add examples for resource state events (4d00c12)
- **api**: Add release links API docs (36d65f0)

Features

- Add an initial mypy test to tox.ini (fdec039)

Add an initial mypy test to test gitlab/base.py and gitlab/**init**.py

- Add personal access token API (2bb16fa)

See: https://docs.gitlab.com/ee/api/personal_access_tokens.html

- Add project audit endpoint (6660dbe)
- Add ProjectPackageFile (b9d469b)

Add ProjectPackageFile and the ability to list project package package_files.

Fixes #1372

- Import from bitbucket server (ff3013a)

I'd like to use this libary to automate importing Bitbucket Server repositories into GitLab. There is a GitLab API endpoint to do this, but it is not exposed through this library.

- Add an `import_bitbucket_server` method to the `ProjectManager`. This method calls this GitLab API endpoint: https://docs.gitlab.com/ee/api/import.html#import-repository-from-bitbucket-server * Modify `import_gitlab` method docstring for python3 compatibility * Add a skipped stub test for the existing `import_github` method
- Option to add a helper to lookup token (8ecf559)
- **api,cli**: Make user agent configurable (4bb201b)
- **issues**: Add missing get verb to IssueManager (f78ebe0)
- **objects**: Add Release Links API support (28d7518)
- **objects**: Add support for group audit events API (2a0fbdf)
- **objects**: Add support for resource state events API (d4799c4)
- **projects**: Add project access token api (1becef0)
- **users**: Add follow/unfollow API (e456869)

Refactoring

- Move Gitlab and GitlabList to gitlab/client.py (53a7645)

Move the classes Gitlab and GitlabList from gitlab/**init**.py to the newly created gitlab/client.py file.

Update one test case that was depending on requests being defined in gitlab/**init**.py

- **api**: Explicitly export classes for star imports (f05c287)
- **objects**: Move instance audit events where they belong (48ba88f)
- **v4**: Split objects and managers per API resource (a5a48ad)

Testing

- Don't add duplicate fixture (5d94846)

Co-authored-by: Nejc Habjan hab.nejc@gmail.com

- **api**: Add functional test for release links API (ab2a1c8)
- **api,cli**: Add tests for custom user agent (c5a37e7)
- **object**: Add test for **dir** duplicates (a8e591f)
- **objects**: Add tests for resource state events (10225cf)
- **objects**: Add unit test for instance audit events (84e3247)

7.10.64 v2.6.0 (2021-01-29)

Bug Fixes

- Docs changed using the consts (650b65c)
- Typo (9baa905)

- **api**: Add missing runner access_level param (92669f2)
- **api**: Use RetrieveMixin for ProjectLabelManager (1a14395)

Allows to get a single label from a project, which was missing before even though the GitLab API has the ability to.

- **base**: Really refresh object (e1e0d8c)

This fixes and error, where deleted attributes would not show up

Fixes #1155

- **cli**: Add missing args for project lists (c73e237)
- **cli**: Write binary data to stdout buffer (0733ec6)

Chores

- Added constants for search API (8ef53d6)
- Added docs for search scopes constants (7565bf0)
- Allow overriding docker-compose env vars for tag (27109ca)
- Apply suggestions (65ce026)
- Move .env into docker-compose dir (55cbd1c)
- Offically support and test 3.9 (62dd07d)
- Remove unnecessary random function (d4ee0a6)
- Simplified search scope constants (16fc048)
- Use helper fixtures for test directories (40ec2f5)
- **ci**: Add .readthedocs.yml (0ad441e)
- **ci**: Add coverage and docs jobs (2de64cf)
- **ci**: Add pytest PR annotations (8f92230)
- **ci**: Fix copy/paste oopsie (c6241e7)
- **ci**: Fix typo in matrix (5e1547a)
- **ci**: Force colors in pytest runs (1502079)
- **ci**: Pin docker-compose install for tests (1f7a2ab)

This ensures python-dotenv with expected behavior for .env processing

- **ci**: Pin os version (cfa27ac)
- **ci**: Reduce renovate PR noise (f4d7a55)
- **ci**: Replace travis with Actions (8bb73a3)
- **cli**: Remove python2 code (1030e0a)
- **deps**: Pin dependencies (14d8f77)
- **deps**: Pin dependency requests-toolbelt to ==0.9.1 (4d25f20)
- **deps**: Update dependency requests to v2.25.1 (9c2789e)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.3.3-ce.0 (667bf01)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.3.4-ce.0 (e94c4c6)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.3.5-ce.0 (c88d870)

- **deps**: Update gitlab/gitlab-ce docker tag to v13.3.6-ce.0 (57b5782)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.4.3-ce.0 (bc17889)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.5.0-ce.0 (fc205cc)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.5.1-ce.0 (348e860)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.5.2-ce.0 (4a6831c)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.5.3-ce.0 (d1b0b08)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.5.4-ce.0 (265dbbd)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.8.1-ce.0 (9854d6d)
- **deps**: Update python docker tag to v3.9 (1fc65e0)
- **docs**: Always edit the file directly on master (35e43c5)

There is no way to edit the raw commit

- **test**: Remove hacking dependencies (9384493)

Documentation

- Add Project Merge Request approval rule documentation (449fc26)
- Clean up grammar and formatting in documentation (aff9bc7)
- **cli**: Add auto-generated CLI reference (6c21fc8)
- **cli**: Add example for job artifacts download (375b29d)
- **cli**: Use inline anonymous references for external links (f2cf467)

There doesn't seem to be an obvious way to use an alias for identical text labels that link to different targets. With inline links we can work around this shortcoming. Until we know better.

- **cli-usage**: Fixed term (d282a99)
- **groups**: Add example for creating subgroups (92eb4e3)
- **issues**: Add admin, project owner hint (609c03b)

Closes #1101

- **projects**: Correct fork docs (54921db)

Closes #1126

- **readme**: Also add hint to delete gitlab-runner-test (8894f2d)

Otherwise the whole testsuite will refuse to run

- **readme**: Update supported Python versions (20b1e79)

Features

- Add MINIMAL_ACCESS constant (49eb3ca)

A "minimal access" access level was introduced in GitLab 13.5.

- Added support for pipeline bridges (05cbdc2)
- Adds support for project merge request approval rules (#1199, c6fbf39)
- Support multipart uploads (2fa3004)
- Unit tests added (f37ebf5)

- **api**: Add support for user identity provider deletion (e78e121)
- **api**: Added wip filter param for merge requests (d6078f8)
- **api**: Added wip filter param for merge requests (aa6e80d)
- **tests**: Test label getter (a41af90)

Refactoring

- **tests**: Split functional tests (61e43eb)

Testing

- Add test_project_merge_request_approvals.py (9f6335f)
- Add unit tests for badges API (2720b73)
- Add unit tests for resource label events API (e9a211c)
- Ignore failing test for now (4b4e253)
- **cli**: Add test for job artifacts download (f4e7950)
- **env**: Replace custom scripts with pytest and docker-compose (79489c7)

7.10.65 v2.5.0 (2020-09-01)

Bug Fixes

- Implement Gitlab's behavior change for owned=True (9977799)
- Tests fail when using REUSE_CONTAINER option (0078f89)

Fixes #1146

- Wrong reconfirmation parameter when updating user's email (b5c267e)

Since version 10.3 (and later), param to not send (re)confirmation when updating an user is `skip_reconfirmation` (and not `skip_confirmation`).

See:

- https://gitlab.com/gitlab-org/gitlab-foss/-/merge_requests/15175?tab= * https://docs.gitlab.com/11.11/ee/api/users.html#user-modification * https://docs.gitlab.com/ee/api/users.html#user-modification

Chores

- Bump python-gitlab to 2.5.0 (56fef01)
- Make latest black happy with existing code (6961479)
- Make latest black happy with existing code (4039c8c)
- Make latest black happy with existing code (d299753)
- Remove remnants of python2 imports (402566a)
- Remove unnecessary import (f337b7a)
- Run unittest2pytest on all unit tests (11383e7)
- Update tools dir for latest black version (c2806d8)
- Update tools dir for latest black version (f245ffb)
- **ci**: Pin gitlab-ce version for renovate (cb79fb7)

- **ci**: Use fixed black version (9565684)
- **deps**: Update gitlab/gitlab-ce docker tag to v13.3.2-ce.0 (9fd778b)
- **deps**: Update python docker tag to v3.8 (a8070f2)
- **env**: Add pre-commit and commit-msg hooks (82070b2)
- **test**: Use pathlib for paths (5a56b6b)

Documentation

- Additional project file delete example (9e94b75)

Showing how to delete without having to pull the file

- **api**: Add example for latest pipeline job artifacts (d20f022)
- **cli**: Add examples for group-project list (af86dcd)
- **packages**: Add examples for Packages API and cli usage (a47dfcd)
- **variables**: Add docs for instance-level variables (ad4b87c)

Features

- Add share/unshare group with group (7c6e541)
- Add support to resource milestone events (88f8cc7)

Fixes #1154

- **api**: Add endpoint for latest ref artifacts (b7a07fc)
- **api**: Add support for instance variables (4492fc4)
- **api**: Add support for Packages API (71495d1)

Refactoring

- Rewrite unit tests for objects with responses (204782a)
- Split unit tests by GitLab API resources (76b2cad)
- Turn objects module into a package (da8af6f)

Testing

- Add unit tests for resource milestone events API (1317f4b)

Fixes #1154

- **api**: Add tests for variables API (66d108d)
- **packages**: Add tests for Packages API (7ea178b)

7.10.66 v2.4.0 (2020-07-09)

Bug Fixes

- Add masked parameter for variables command (b6339bf)
- Do not check if kwargs is none (a349b90)

Co-authored-by: Traian Nedelea tron1point0@pm.me

- Make query kwargs consistent between call in init and next (72ffa01)

- Pass kwargs to subsequent queries in gitlab list (1d011ac)
- **merge**: Parse arguments as query_data (878098b)

Chores

- Bump version to 2.4.0 (1606310)

Documentation

- **pipelines**: Simplify download (9a068e0)

This uses a context instead of inventing your own stream handler which makes the code simpler and should be fine for most use cases.

Signed-off-by: Paul Spooren mail@aparcar.org

Features

- Added NO_ACCESS const (dab4d0a)

This constant is useful for cases where no access is granted, e.g. when creating a protected branch.

The `NO_ACCESS` const corresponds to the definition in https://docs.gitlab.com/ee/api/protected_branches.html

7.10.67 v2.3.1 (2020-06-09)

Bug Fixes

- Disable default keyset pagination (e71fe16)

Instead we set pagination to offset on the other paths

Chores

- Bump version to 2.3.1 (870e7ea)

7.10.68 v2.3.0 (2020-06-08)

Bug Fixes

- Use keyset pagination by default for /projects > 50000 (f86ef3b)

Workaround for https://gitlab.com/gitlab-org/gitlab/-/issues/218504. Remove this in 13.1

- **config**: Fix duplicate code (ee2df6f)

Fixes #1094

- **project**: Add missing project parameters (ad8c67d)

Chores

- Bring commit signatures up to date with 12.10 (dc382fe)
- Bump to 2.3.0 (01ff865)
- Correctly render rst (f674bf2)
- Fix typo in docstring (c20f5f1)
- Remove old builds-email service (c60e2df)
- Use pytest for unit tests and coverage (9787a40)

- **ci**: Add codecov integration to Travis (e230568)
- **services**: Update available service attributes (7afc357)
- **test**: Remove outdated token test (e6c9fe9)

Continuous Integration

- Add a test for creating and triggering pipeline schedule (9f04560)
- Lint fixes (930122b)

Documentation

- Update authors (ac0c84d)
- **readme**: Add codecov badge for master (e21b2c5)
- **readme**: Update test docs (6e2b1ec)
- **remote_mirrors**: Fix create command (bab91fe)
- **remote_mirrors**: Fix create command (1bb4e42)

Features

- Add group runners api (4943991)
- Add play command to project pipeline schedules (07b9988)

fix: remove version from setup

feat: add pipeline schedule play error exception

docs: add documentation for pipeline schedule play

- Allow an environment variable to specify config location (401e702)

It can be useful (especially in scripts) to specify a configuration location via an environment variable. If the "PYTHON_GITLAB_CFG" environment variable is defined, treat its value as the path to a configuration file and include it in the set of default configuration locations.

- **api**: Added support in the GroupManager to upload Group avatars (28eb7ea)
- **services**: Add project service list API (fc52221)

Can be used to list available services It was introduced in GitLab 12.7

- **types**: Add **dir** to RESTObject to expose attributes (cad134c)

Testing

- Disable test until Gitlab 13.1 (63ae77a)
- **cli**: Convert shell tests to pytest test cases (c4ab4f5)
- **runners**: Add all runners unit tests (127fa5a)

7.10.69 v2.2.0 (2020-04-07)

Bug Fixes

- Add missing import_project param (9b16614)
- **types**: Do not split single value string in ListAttribute (a26e585)

Chores

- Bump to 2.2.0 (22d4b46)
- Clean up for black and flake8 (4fede5d)
- Fix typo in allow_failures (265bbdd)
- Flatten test_import_github (b8ea96c)
- Improve and document testing against different images (98d3f77)
- Move test_import_github into TestProjectImport (a881fb7)
- Pass environment variables in tox (e06d33c)
- Remove references to python2 in test env (6e80723)
- Rename ExportMixin to DownloadMixin (847da60)
- Use raise..from for chained exceptions (#939, 79fef26)
- **group**: Update group_manager attributes (#1062, fa34f5e)
- chore(group): update group_manager attributes

Co-Authored-By: Nejc Habjan hab.nejc@gmail.com

- **mixins**: Factor out export download into ExportMixin (6ce5d1f)

Documentation

- Add docs for Group Import/Export API (8c3d744)
- Fix comment of prev_page() (b066b41)

Co-Authored-By: Nejc Habjan hab.nejc@gmail.com

- Fix comment of prev_page() (ac6b2da)

Co-Authored-By: Nejc Habjan hab.nejc@gmail.com

- Fix comment of prev_page() (7993c93)

Features

- Add create from template args to ProjectManager (f493b73)

This commit adds the v4 Create project attributes necessary to create a project from a project, instance, or group level template as documented in https://docs.gitlab.com/ee/api/projects.html#create-project

- Add support for commit GPG signature API (da7a809)
- **api**: Add support for Gitlab Deploy Token API (01de524)
- **api**: Add support for Group Import/Export API (#1037, 6cb9d92)
- **api**: Add support for remote mirrors API (#1056, 4cfaa2f)

Testing

- Add unit tests for Project Export (600dc86)
- Add unit tests for Project Import (f7aad5f)
- Create separate module for commit tests (8c03771)
- Move mocks to top of module (0bff713)

- Prepare base project test class for more tests (915587f)
- **api**: Add tests for group export/import API (e7b2d6c)
- **types**: Reproduce get_for_api splitting strings (#1057, babd298)

7.10.70 v2.1.2 (2020-03-09)

Chores

- Bump version to 2.1.2 (ad7e2bf)

7.10.71 v2.1.1 (2020-03-09)

Bug Fixes

- **docs**: Additional project statistics example (5ae5a06)

Chores

- Bump version to 2.1.1 (6c5458a)
- **user**: Update user attributes to 12.8 (666f880)

7.10.72 v2.1.0 (2020-03-08)

Bug Fixes

- Do not require empty data dict for create() (99d959f)
- Remove null values from features POST data, because it fails (1ec1816)
- Remove trailing slashes from base URL (#913, 2e396e4)
- Return response with commit data (b77b945)
- **docs**: Fix typo in user memberships example (33889bc)
- **docs**: Update to new set approvers call for # of approvers (8e0c526)

to set the # of approvers for an MR you need to use the same function as for setting the approvers id.

- **docs and tests**: Update docs and tests for set_approvers (2cf12c7)

Updated the docs with the new set_approvers arguments, and updated tests with the arg as well.

- **objects**: Add default name data and use http post (70c0cfb)

Updating approvers new api needs a POST call. Also It needs a name of the new rule, defaulting this to 'name'.

- **objects**: Update set_approvers function call (65ecadc)

Added a miss paramter update to the set_approvers function

- **objects**: Update to new gitlab api for path, and args (e512cdd)

Updated the gitlab path for set_approvers to approvers_rules, added default arg for rule type, and added arg for # of approvals required.

- **projects**: Correct copy-paste error (adc9101)

Chores

- Bump version to 2.1.0 (47cb58c)
- Ensure developers use same gitlab image as Travis (fab17fc)
- Fix broken requests links (b392c21)

Another case of the double slash rewrite.

Code Style

- Fix black violations (ad3e833)

Documentation

- Add reference for REQUESTS_CA_BUNDLE (37e8d5d)
- **pagination**: Clear up pagination docs (1609824)

Co-Authored-By: Mitar mitar.git@tnode.com

Features

- Add capability to control GitLab features per project or group (7f192b4)
- Add support for commit revert API (#991, 5298964)
- Add support for user memberships API (#1009, c313c2b)
- Use keyset pagination by default for `all=True` (99b4484)
- **api**: Add support for GitLab OAuth Applications API (4e12356)

Performance Improvements

- Prepare environment when gitlab is reconfigured (3834d9c)

Testing

- Add unit tests for base URLs with trailing slashes (32844c7)
- Add unit tests for revert commit API (d7a3066)
- Remove duplicate resp_get_project (cb43695)
- Use lazy object in unit tests (31c6562)

7.10.73 v2.0.1 (2020-02-05)

Chores

- Bump to 2.1.0 (a6c0660)

There are a few more features in there

- Bump version to 2.0.1 (8287a0d)
- Revert to 2.0.1 (272db26)

I've misread the tag

- **user**: Update user attributes (27375f6)

This also workarounds an GitLab issue, where private_profile, would reset to false if not supplied

Documentation

- **auth**: Remove email/password auth (c9329bb)

7.10.74 v2.0.0 (2020-01-26)

Bug Fixes

- **projects**: Adjust snippets to match the API (e104e21)

Chores

- Add PyYaml as extra require (7ecd518)
- Build_sphinx needs sphinx >= 1.7.6 (528dfab)

Stepping thru Sphinx versions from 1.6.5 to 1.7.5 build_sphinx fails. Once Sphinx == 1.7.6 build_sphinx finished.

- Bump minimum required requests version (3f78aa3)

for security reasons

- Bump to 2.0.0 (c817dcc)

Dropping support for legacy python requires a new major version

- Drop legacy python tests (af8679a)

Support dropped for: 2.7, 3.4, 3.5

- Enforce python version requirements (70176db)

Documentation

- Fix snippet get in project (3a4ff2f)
- **projects**: Add raw file download docs (939e9d3)

Fixes #969

Features

- Add appearance API (4c4ac5c)
- Add autocompletion support (973cb8b)
- Add global order_by option to ease pagination (d187925)
- Support keyset pagination globally (0b71ba4)

Refactoring

- Remove six dependency (9fb4645)
- Support new list filters (bded2de)

This is most likely only useful for the CLI

Testing

- Add project snippet tests (0952c55)
- Adjust functional tests for project snippets (ac0ea91)

7.10.75 v1.15.0 (2019-12-16)

Bug Fixes

- Ignore all parameter, when as_list=True (137d72b)

Closes #962

Chores

- Bump version to 1.15.0 (2a01326)
- **ci**: Use correct crane ci (18913dd)

Code Style

- Format with the latest black version (06a8050)

Documentation

- Added docs for statistics (8c84cbf)
- **projects**: Fix file deletion docs (1c4f1c4)

The function `file.delete()` requires `branch` argument in addition to `commit_message`.

Features

- Access project's issues statistics (482e57b)

Fixes #966

- Add support for /import/github (aa4d41b)

Addresses python-gitlab/python-gitlab#952

This adds a method to the `ProjectManager` called `import_github`, which maps to the `/import/github` API endpoint. Calling `import_github` will trigger an import operation from <repo_id> into <target_namespace>, using <personal_access_token> to authenticate against github. In practice a gitlab server may take many 10's of seconds to respond to this API call, so we also take the liberty of increasing the default timeout (only for this method invocation).

Unfortunately since `import` is a protected keyword in python, I was unable to follow the endpoint structure with the manager namespace. I'm open to suggestions on a more sensible interface.

I'm successfully using this addition to batch-import hundreds of github repositories into gitlab.

- Add variable_type to groups ci variables (0986c93)

This adds the ci variables types for create/update requests.

See https://docs.gitlab.com/ee/api/group_level_variables.html#create-variable

- Add variable_type/protected to projects ci variables (4724c50)

This adds the ci variables types and protected flag for create/update requests.

See https://docs.gitlab.com/ee/api/project_level_variables.html#create-variable

- Adding project stats (db0b00a)

Fixes #967

- Allow cfg timeout to be overrided via kwargs (e9a8289)

On startup, the `timeout` parameter is loaded from config and stored on the base gitlab object instance. This instance parameter is used as the timeout for all API requests (it's passed into the `session` object when making HTTP calls).

This change allows any API method to specify a `timeout` argument to `**kwargs` that will override the global timeout value. This was somewhat needed / helpful for the `import_github` method.

I have also updated the docs accordingly.

- Nicer stacktrace (`697cda2`)
- Retry transient HTTP errors (`59fe271`)

Fixes #970

Testing

- Added tests for statistics (`8760efc`)
- Test that all is ignored, when as_list=False (`b5e88f3`)

7.10.76 v1.14.0 (2019-12-07)

Bug Fixes

- Added missing attributes for project approvals (`460ed63`)

Reference: https://docs.gitlab.com/ee/api/merge_request_approvals.html#change-configuration

Missing attributes: * merge_requests_author_approval * merge_requests_disable_committers_approval

- **labels**: Ensure label.save() works (`727f536`)

Otherwise, we get: File "gitlabracadabra/mixins/labels.py", line 67, in _process_labels current_label.save() File "gitlab/exceptions.py", line 267, in wrapped_f return f(*args, **kwargs) File "gitlab/v4/objects.py", line 896, in save self._update_attrs(server_data) File "gitlab/base.py", line 131, in _update_attrs self.**dict**["_attrs"].update(new_attrs) TypeError: 'NoneType' object is not iterable

Because server_data is None.

- **project-fork**: Copy create fix from ProjectPipelineManager (`516307f`)
- **project-fork**: Correct path computation for project-fork list (`44a7c27`)

Chores

- Bump version to 1.14.0 (`164fa4f`)
- **ci**: Switch to crane docker image (#944, `e0066b6`)

Documentation

- Add project and group cluster examples (`d15801d`)
- Fix typo (`d9871b1`)
- **changelog**: Add notice for release-notes on Github (#938, `de98e57`)
- **pipelines_and_jobs**: Add pipeline custom variables usage example (`b275eb0`)
- **readme**: Fix Docker image reference (`b9a40d8`)

v1.8.0 is not available. ``` Unable to find image 'registry.gitlab.com/python-gitlab/python-gitlab:v1.8.0' locally docker: Error response from daemon: manifest for registry.gitlab.com/python-gitlab/python-gitlab:v1.8.0 not found: manifest unknown: manifest unknown.

```
- **snippets**: Fix snippet docs
  ([`bbaa754`](https://github.com/python-gitlab/python-gitlab/commit/
↪bbaa754673c4a0bffece482fe33e4875ddadc2dc))

Fixes #954

### Features

- Add audit endpoint
  ([`2534020`](https://github.com/python-gitlab/python-gitlab/commit/
↪2534020b1832f28339ef466d6dd3edc21a521260))

- Add project and group clusters
  ([`ebd053e`](https://github.com/python-gitlab/python-gitlab/commit/
↪ebd053e7bb695124c8117a95eab0072db185ddf9))

- Add support for include_subgroups filter
  ([`adbcd83`](https://github.com/python-gitlab/python-gitlab/commit/
↪adbcd83fa172af2f3929ba063a0e780395b102d8))


## v1.13.0 (2019-11-02)

### Bug Fixes

- **projects**: Support `approval_rules` endpoint for projects
  ([`2cef2bb`](https://github.com/python-gitlab/python-gitlab/commit/
↪2cef2bb40b1f37b97bb2ee9894ab3b9970cef231))

The `approvers` API endpoint is deprecated [1]. GitLab instead uses the `approval_rules`␣
↪API
  endpoint to modify approval settings for merge requests. This adds the functionality␣
↪for
  project-level merge request approval settings.

Note that there does not exist an endpoint to 'get' a single approval rule at this␣
↪moment - only
  'list'.

[1] https://docs.gitlab.com/ee/api/merge_request_approvals.html

### Chores

- Bump version to 1.13.0
  ([`d0750bc`](https://github.com/python-gitlab/python-gitlab/commit/
↪d0750bc01ed12952a4d259a13b3917fa404fd435))

- **ci**: Update latest docker image for every tag
  ([`01cbc7a`](https://github.com/python-gitlab/python-gitlab/commit/
↪01cbc7ad04a875bea93a08c0ce563ab5b4fe896b))

- **dist**: Add test data
```

(continues on next page)

(continued from previous page)

```
  ([`3133ed7`](https://github.com/python-gitlab/python-gitlab/commit/
↪3133ed7d1df6f49de380b35331bbcc67b585a61b))

Closes #907

- **setup**: We support 3.8 ([#924](https://github.com/python-gitlab/python-gitlab/pull/
↪924),
  [`6048175`](https://github.com/python-gitlab/python-gitlab/commit/
↪6048175ef2c21fda298754e9b07515b0a56d66bd))

* chore(setup): we support 3.8

* style: format with black

### Documentation

- Projects get requires id
  ([`5bd8947`](https://github.com/python-gitlab/python-gitlab/commit/
↪5bd8947bd16398aed218f07458aef72e67f2d130))

Also, add an example value for project_id to the other projects.get() example.

- **project**: Fix group project example
  ([`e680943`](https://github.com/python-gitlab/python-gitlab/commit/
↪e68094317ff6905049e464a59731fe4ab23521de))

GroupManager.search is removed since 9a66d78, use list(search='keyword') instead

### Features

- Add deployment creation
  ([`ca256a0`](https://github.com/python-gitlab/python-gitlab/commit/
↪ca256a07a2cdaf77a5c20e307d334b82fd0fe861))

Added in GitLab 12.4

Fixes #917

- Add users activate, deactivate functionality
  ([`32ad669`](https://github.com/python-gitlab/python-gitlab/commit/
↪32ad66921e408f6553b9d60b6b4833ed3180f549))

These were introduced in GitLab 12.4

- Send python-gitlab version as user-agent
  ([`c22d49d`](https://github.com/python-gitlab/python-gitlab/commit/
↪c22d49d084d1e03426cfab0d394330f8ab4bd85a))

- **auth**: Remove deprecated session auth
  ([`b751cdf`](https://github.com/python-gitlab/python-gitlab/commit/
↪b751cdf424454d3859f3f038b58212e441faafaf))
```

(continues on next page)

(continued from previous page)

```
- **doc**: Remove refs to api v3 in docs
  ([`6beeaa9`](https://github.com/python-gitlab/python-gitlab/commit/
↪6beeaa993f8931d6b7fe682f1afed2bd4c8a4b73))

- **test**: Unused unittest2, type -> isinstance
  ([`33b1801`](https://github.com/python-gitlab/python-gitlab/commit/
↪33b180120f30515d0f76fcf635cb8c76045b1b42))

### Testing

- Remove warning about open files from test_todo()
  ([`d6419aa`](https://github.com/python-gitlab/python-gitlab/commit/
↪d6419aa86d6ad385e15d685bf47242bb6c67653e))

When running unittests python warns that the json file from test_todo() was still open.␣
↪Use with to
  open, read, and create encoded json data that is used by resp_get_todo().

- **projects**: Support `approval_rules` endpoint for projects
  ([`94bac44`](https://github.com/python-gitlab/python-gitlab/commit/
↪94bac4494353e4f597df0251f0547513c011e6de))


## v1.12.1 (2019-10-07)

### Bug Fixes

- Fix not working without auth
  ([`03b7b5b`](https://github.com/python-gitlab/python-gitlab/commit/
↪03b7b5b07e1fd2872e8968dd6c29bc3161c6c43a))


## v1.12.0 (2019-10-06)

### Bug Fixes

- **cli**: Fix cli command user-project list
  ([`c17d7ce`](https://github.com/python-gitlab/python-gitlab/commit/
↪c17d7ce14f79c21037808894d8c7ba1117779130))

- **labels**: Don't mangle label name on update
  ([`1fb6f73`](https://github.com/python-gitlab/python-gitlab/commit/
↪1fb6f73f4d501c2b6c86c863d40481e1d7a707fe))

- **todo**: Mark_all_as_done doesn't return anything
  ([`5066e68`](https://github.com/python-gitlab/python-gitlab/commit/
↪5066e68b398039beb5e1966ba1ed7684d97a8f74))

### Chores

- Bump to 1.12.0
  ([`4648128`](https://github.com/python-gitlab/python-gitlab/commit/
```

(continues on next page)

(continued from previous page)

```
↪46481283a9985ae1b07fe686ec4a34e4a1219b66))

- **ci**: Build test images on tag
  ([`0256c67`](https://github.com/python-gitlab/python-gitlab/commit/
↪0256c678ea9593c6371ffff60663f83c423ca872))

### Code Style

- Format with black
  ([`fef085d`](https://github.com/python-gitlab/python-gitlab/commit/
↪fef085dca35d6b60013d53a3723b4cbf121ab2ae))

### Documentation

- **project**: Add submodule docs
  ([`b5969a2`](https://github.com/python-gitlab/python-gitlab/commit/
↪b5969a2dcea77fa608cc29be7a5f39062edd3846))

- **projects**: Add note about project list
  ([`44407c0`](https://github.com/python-gitlab/python-gitlab/commit/
↪44407c0f59b9602b17cfb93b5e1fa37a84064766))

Fixes #795

- **repository-tags**: Fix typo
  ([`3024c5d`](https://github.com/python-gitlab/python-gitlab/commit/
↪3024c5dc8794382e281b83a8266be7061069e83e))

Closes #879

- **todo**: Correct todo docs
  ([`d64edcb`](https://github.com/python-gitlab/python-gitlab/commit/
↪d64edcb4851ea62e72e3808daf7d9b4fdaaf548b))

### Features

- Add support for job token
  ([`cef3aa5`](https://github.com/python-gitlab/python-gitlab/commit/
↪cef3aa51a6928338c6755c3e6de78605fae8e59e))

See https://docs.gitlab.com/ee/api/jobs.html#get-job-artifacts for usage

- **ci**: Improve functionnal tests
  ([`eefceac`](https://github.com/python-gitlab/python-gitlab/commit/
↪eefceace2c2094ef41d3da2bf3c46a58a450dcba))

- **project**: Add file blame api
  ([`f5b4a11`](https://github.com/python-gitlab/python-gitlab/commit/
↪f5b4a113a298d33cb72f80c94d85bdfec3c4e149))

https://docs.gitlab.com/ee/api/repository_files.html#get-file-blame-from-repository
```

(continues on next page)

(continued from previous page)

```
- **project**: Implement update_submodule
  ([`4d1e377`](https://github.com/python-gitlab/python-gitlab/commit/
↪4d1e3774706f336e87ebe70e1b373ddb37f34b45))

- **user**: Add status api
  ([`62c9fe6`](https://github.com/python-gitlab/python-gitlab/commit/
↪62c9fe63a47ddde2792a4a5e9cd1c7aa48661492))

### Refactoring

- Remove obsolete test image
  ([`a14c02e`](https://github.com/python-gitlab/python-gitlab/commit/
↪a14c02ef85bd4d273b8c7f0f6bd07680c91955fa))

Follow up of #896

- Remove unused code, simplify string format
  ([`c7ff676`](https://github.com/python-gitlab/python-gitlab/commit/
↪c7ff676c11303a00da3a570bf2893717d0391f20))

### Testing

- Re-enabled py_func_v4 test
  ([`49d84ba`](https://github.com/python-gitlab/python-gitlab/commit/
↪49d84ba7e95fa343e622505380b3080279b83f00))

- **func**: Disable commit test
  ([`c9c76a2`](https://github.com/python-gitlab/python-gitlab/commit/
↪c9c76a257d2ed3b394f499253d890c2dd9a01e24))

GitLab seems to be randomly failing here

- **status**: Add user status test
  ([`fec4f9c`](https://github.com/python-gitlab/python-gitlab/commit/
↪fec4f9c23b8ba33bb49dca05d9c3e45cb727e0af))

- **submodules**: Correct test method
  ([`e59356f`](https://github.com/python-gitlab/python-gitlab/commit/
↪e59356f6f90d5b01abbe54153441b6093834aa11))

- **todo**: Add unittests
  ([`7715567`](https://github.com/python-gitlab/python-gitlab/commit/
↪77155678a5d8dbbf11d00f3586307694042d3227))


## v1.11.0 (2019-08-31)

### Bug Fixes

- Add project and group label update without id to fix cli
  ([`a3d0d7c`](https://github.com/python-gitlab/python-gitlab/commit/
↪a3d0d7c1e7b259a25d9dc84c0b1de5362c80abb8))
```

(continues on next page)

(continued from previous page)

```
- Remove empty dict default arguments
  ([`8fc8e35`](https://github.com/python-gitlab/python-gitlab/commit/
↪8fc8e35c63d7ebd80408ae002693618ca16488a7))

Signed-off-by: Frantisek Lachman <flachman@redhat.com>

- Remove empty list default arguments
  ([`6e204ce`](https://github.com/python-gitlab/python-gitlab/commit/
↪6e204ce819fc8bdd5359325ed7026a48d63f8103))

Signed-off-by: Frantisek Lachman <flachman@redhat.com>

- **projects**: Avatar uploading for projects
  ([`558ace9`](https://github.com/python-gitlab/python-gitlab/commit/
↪558ace9b007ff9917734619c05a7c66008a4c3f0))

### Chores

- Bump package version
  ([`37542cd`](https://github.com/python-gitlab/python-gitlab/commit/
↪37542cd28aa94ba01d5d289d950350ec856745af))

### Features

- Add methods to retrieve an individual project environment
  ([`29de40e`](https://github.com/python-gitlab/python-gitlab/commit/
↪29de40ee6a20382c293d8cdc8d831b52ad56a657))

- Group labels with subscriptable mixin
  ([`4a9ef9f`](https://github.com/python-gitlab/python-gitlab/commit/
↪4a9ef9f0fa26e01fc6c97cf88b2a162e21f61cce))

### Testing

- Add group label cli tests
  ([`f7f24bd`](https://github.com/python-gitlab/python-gitlab/commit/
↪f7f24bd324eaf33aa3d1d5dd12719237e5bf9816))


## v1.10.0 (2019-07-22)

### Bug Fixes

- Convert # to %23 in URLs
  ([`14f5385`](https://github.com/python-gitlab/python-gitlab/commit/
↪14f538501bfb47c92e02e615d0817675158db3cf))

Refactor a bit to handle this change, and add unit tests.

Closes #779
```

(continues on next page)

(continued from previous page)

```
- Docker entry point argument passing
  ([`67ab637`](https://github.com/python-gitlab/python-gitlab/commit/
↪67ab6371e69fbf137b95fd03105902206faabdac))

Fixes the problem of passing spaces in the arguments to the docker entrypoint.

Before this fix, there was virtually no way to pass spaces in arguments such as task␣
↪description.

- Enable use of YAML in the CLI
  ([`ad0b476`](https://github.com/python-gitlab/python-gitlab/commit/
↪ad0b47667f98760d6a802a9d08b2da8f40d13e87))

In order to use the YAML output, PyYaml needs to be installed on the docker image. This␣
↪commit adds
  the installation to the dockerfile as a separate layer.

- Handle empty 'Retry-After' header from GitLab
  ([`7a3724f`](https://github.com/python-gitlab/python-gitlab/commit/
↪7a3724f3fca93b4f55aed5132cf46d3718c4f594))

When requests are throttled (HTTP response code 429), python-gitlab assumed that 'Retry-
↪After'
  existed in the response headers. This is not always the case and so the request fails␣
↪due to a
  KeyError. The change in this commit adds a rudimentary exponential backoff to the
↪'http_request'
  method, which defaults to 10 retries but can be set to -1 to retry without bound.

- Improve pickle support
  ([`b4b5dec`](https://github.com/python-gitlab/python-gitlab/commit/
↪b4b5decb7e49ac16d98d56547a874fb8f9d5492b))

- Pep8 errors
  ([`334f9ef`](https://github.com/python-gitlab/python-gitlab/commit/
↪334f9efb18c95bb5df3271d26fa0a55b7aec1c7a))

Errors have not been detected by broken travis runs.

- Re-add merge request pipelines
  ([`877ddc0`](https://github.com/python-gitlab/python-gitlab/commit/
↪877ddc0dbb664cd86e870bb81d46ca614770b50e))

- Remove decode() on error_message string
  ([`16bda20`](https://github.com/python-gitlab/python-gitlab/commit/
↪16bda20514e036e51bef210b565671174cdeb637))

The integration tests failed because a test called 'decode()' on a string-type variable -
↪ the
  GitLabException class handles byte-to-string conversion already in its __init__. This␣
↪commit
  removes the call to 'decode()' in the test.
```

(continues on next page)

(continued from previous page)

```
``` Traceback (most recent call last): File "./tools/python_test_v4.py", line 801, in
↪<module>
  assert 'Retry later' in error_message.decode() AttributeError: 'str' object has no␣
↪attribute
  'decode'
```

- Use python2 compatible syntax for super (b08efcb)
- **api**: Avoid parameter conflicts with python and gitlab (4bd027a)

Provide another way to send data to gitlab with a new `query_parameters` argument. This parameter can be used to explicitly define the dict of items to send to the server, so that **kwargs are only used to specify python-gitlab specific parameters.

Closes #566 Closes #629

- **api**: Don't try to parse raw downloads (35a6d85)

http_get always tries to interpret the retrieved data if the content-type is json. In some cases (artifact download for instance) this is not the expected behavior.

This patch changes http_get and download methods to always get the raw data without parsing.

Closes #683

- **api**: Make *MemberManager.all() return a list of objects (d74ff50)

Fixes #699

- **api**: Make reset_time_estimate() work again (cb388d6)

Closes #672

- **cli**: Allow –recursive parameter in repository tree (7969a78)

Fixes #718 Fixes #731

- **cli**: Don't fail when the short print attr value is None (8d1552a)

Fixes #717 Fixes #727

- **cli**: Exit on config parse error, instead of crashing (6ad9da0)
- Exit and hint user about possible errors * test: adjust test cases to config missing error
- **cli**: Fix update value for key not working (b766203)
- **cli**: Print help and usage without config file (6bb4d17)

Fixes #560

- **docker**: Use docker image with current sources (06e8ca8)

Chores

- Add a tox job to run black (c27fa48)

Allow lines to be 88 chars long for flake8.

- Bump package version to 1.10.0 (c7c8470)
- Disable failing travis test (515aa9a)
```

- Move checks back to travis (b764525)
- Release tags to PyPI automatically (3133b48)

Fixes #609

- **ci**: Add automatic GitLab image pushes (95c9b6d)
- **ci**: Don't try to publish existing release (b4e818d)
- **ci**: Fix gitlab PyPI publish (3e37df1)
- **ci**: Rebuild test image, when something changed (2fff260)
- **ci**: Update the GitLab version in the test image (c410699)
- **ci**: Use reliable ci system (724a672)
- **setup**: Add 3.7 to supported python versions (b1525c9)
- **tests**: Add rate limit tests (e216f06)

Code Style

- Format with black again (22b5082)

Documentation

- Add a note for python 3.5 for file content update (ca014f8)

The data passed to the JSON serializer must be a string with python 3. Document this in the exemples.

Fix #175

- Add an example of trigger token usage (ea1eefe)

Closes #752

- Add ApplicationSettings API (ab7d794)
- Add builds-related API docs (8e6a944)
- Add deploy keys API (ea089e0)
- Add labales API (31882b8)
- Add licenses API (4540614)
- Add milestones API (7411907)
- Add missing = (391417c)
- Add missing requiredCreateAttrs (b08d74a)
- Add MR API (5614a7c)
- Add MR approvals in index (0b45afb)
- Add pipeline deletion (2bb2571)
- Add project members doc (dcf31a4)
- Commits API (07c5594)
- Crossref improvements (6f9f42b)
- Do not use the :option: markup (368017c)
- Document hooks API (b21dca0)

- Document projects API (967595f)
- Fix "required" attribute (e64d0b9)
- Fix invalid Raise attribute in docstrings (95a3fe6)
- Fork relationship API (21f48b3)
- Groups API documentation (4d871aa)
- Improve the pagination section (29e2efe)
- Issues API (41cbc32)
- Notes API (3e026d2)
- Project repository API (71a2a4f)
- Project search API (e4cd04c)
- Re-order api examples (5d149a2)

`Pipelines and Jobs` and `Protected Branches` are out of order in contents and sometimes hard to find when looking for examples.

- Remove the build warning about _static (764d3ca)
- Remove v3 support (7927663)
- Repository files API (f00340f)
- Snippets API (35b7f75)
- Start a FAQ (c305459)
- System hooks API (5c51bf3)
- Tags API (dd79eda)
- Trigger_pipeline only accept branches and tags as ref (d63748a)

Fixes #430

- **api-usage**: Add rate limit documentation (ad4de20)
- **api-usage**: Fix project group example (40a1bf3)

Fixes #798

- **cli**: Add PyYAML requirement notice (d29a489)

Fixes #606

- **groups**: Fix typo (ac2d65a)

Fixes #635

- **projects**: Add mention about project listings (f604b25)

Having exactly 20 internal and 5 private projects in the group spent some time debugging this issue.

Hopefully that helped: https://github.com/python-gitlab/python-gitlab/issues/93

Imho should be definitely mention about `all=True` parameter.

- **projects**: Fix typo (c6bcfe6)
- **projects**: Fix typo in code sample (b93f2a9)

Fixes #630

- **readme**: Add docs build information (6585c96)
- **readme**: Add more info about commitlint, code-format (286f703)
- **readme**: Fix six url (0bc30f8)

six URL was pointing to 404

- **readme**: Provide commit message guidelines (bed8e1b)

Fixes #660

- **setup**: Use proper readme on PyPI (6898097)
- **snippets**: Fix project-snippets layout (7feb97e)

Fixes #828

Features

- Add endpoint to get the variables of a pipeline (564de48)

It adds a new endpoint which was released in the Gitlab CE 11.11.

Signed-off-by: Agustin Henze tin@redhat.com

- Add mr rebase method (bc4280c)
- Add support for board update (908d79f)

Closes #801

- Add support for issue.related_merge_requests (90a3631)

Closes #794

- Added approve & unapprove method for Mergerequests (53f7de7)

Offical GitLab API supports this for GitLab EE

- Bump version to 1.9.0 (aaed448)
- Get artifact by ref and job (cda1174)
- Implement artifacts deletion (76b6e1f)

Closes #744

- Obey the rate limit (2abf9ab)

done by using the retry-after header

Fixes #166

- **GitLab Update**: Delete ProjectPipeline (#736, 768ce19)
- feat(GitLab Update): delete ProjectPipeline

As of Gitlab 11.6 it is now possible to delete a pipeline - https://docs.gitlab.com/ee/api/pipelines.html#delete-a-pipeline

Refactoring

- Format everything black (318d277)
- Rename MASTER_ACCESS (c38775a)

to MAINTAINER_ACCESS to follow GitLab 11.0 docs

See: https://docs.gitlab.com/ce/user/permissions.html#project-members-permissions

Testing

- Add project releases test (8ff8af0)

Fixes #762

- Always use latest version to test (82b0fc6)
- Increase speed by disabling the rate limit faster (497f56c)
- Minor test fixes (3b523f4)
- Update the tests for GitLab 11.11 (622854f)

Changes in GitLab make the functional tests fail:

- Some actions add new notes and discussions: do not use hardcoded values in related listing asserts
  - The feature flag API is buggy (errors 500): disable the tests for now

7.11 Release notes

Prior to version 2.0.0 and GitHub Releases, a summary of changes was maintained in release notes. They are available below for historical purposes. For the list of current releases, including breaking changes, please see the changelog.

7.11.1 Changes from 1.8 to 1.9

- `ProjectMemberManager.all()` and `GroupMemberManager.all()` now return a list of `ProjectMember` and `GroupMember` objects respectively, instead of a list of dicts.

7.11.2 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` paremeter is not sent to the GitLab anymore.

7.11.3 Changes from 1.5 to 1.6

- When python-gitlab detects HTTP redirections from http to https it will raise a RedirectionError instead of a cryptic error.

  Make sure to use an `https://` protocol in your GitLab URL parameter if the server requires it.

7.11.4 Changes from 1.4 to 1.5

- APIv3 support has been removed. Use the 1.4 release/branch if you need v3 support.
- GitLab EE features are now supported: Geo nodes, issue links, LDAP groups, project/group boards, project mirror pulling, project push rules, EE license configuration, epics.
- The `GetFromListMixin` class has been removed. The `get()` method is not available anymore for the following managers:
  - UserKeyManager
  - DeployKeyManager
  - GroupAccessRequestManager
  - GroupIssueManager
  - GroupProjectManager
  - GroupSubgroupManager
  - IssueManager
  - ProjectCommitStatusManager
  - ProjectEnvironmentManager
  - ProjectLabelManager
  - ProjectPipelineJobManager
  - ProjectAccessRequestManager
  - TodoManager
- `ProjectPipelineJob` do not heritate from `ProjectJob` anymore and thus can only be listed.

7.11.5 Changes from 1.3 to 1.4

- 1.4 is the last release supporting the v3 API, and the related code will be removed in the 1.5 version.

  If you are using a Gitlab server version that does not support the v4 API you can:

  - upgrade the server (recommended)
  - make sure to use version 1.4 of python-gitlab (`pip install python-gitlab==1.4`)

  See also the Switching to GitLab API v4 documentation.
- python-gitlab now handles the server rate limiting feature. It will pause for the required time when reaching the limit (documentation)
- The `GetFromListMixin.get()` method is deprecated and will be removed in the next python-gitlab version. The goal of this mixin/method is to provide a way to get an object by looping through a list for GitLab objects that don't support the GET method. The method is broken and conflicts with the GET method now supported by some GitLab objects.

  You can implement your own method with something like:

```
def get_from_list(self, id):
    for obj in self.list(as_list=False):
        if obj.get_id() == id:
            return obj
```

- The `GroupMemberManager`, `NamespaceManager` and `ProjectBoardManager` managers now use the GET API from GitLab instead of the `GetFromListMixin.get()` method.

7.11.6 Changes from 1.2 to 1.3

- `gitlab.Gitlab` objects can be used as context managers in a `with` block.

7.11.7 Changes from 1.1 to 1.2

- python-gitlab now respects the `*_proxy`, `REQUESTS_CA_BUNDLE` and `CURL_CA_BUNDLE` environment variables (#352)
- The following deprecated methods and objects have been removed:
  - gitlab.v3.object `Key` and `KeyManager` objects: use `DeployKey` and `DeployKeyManager` instead
  - gitlab.v3.objects.Project `archive_` and `unarchive_` methods
  - gitlab.Gitlab `credentials_auth`, `token_auth`, `set_url`, `set_token` and `set_credentials` methods. Once a Gitlab object has been created its URL and authentication information cannot be updated: create a new Gitlab object if you need to use new information
- The `todo()` method raises a `GitlabTodoError` exception on error

7.11.8 Changes from 1.0.2 to 1.1

- The `ProjectUser` class doesn't inherit from `User` anymore, and the `GroupProject` class doesn't inherit from `Project` anymore. The Gitlab API doesn't provide the same set of features for these objects, so python-gitlab objects shouldn't try to workaround that.

  You can create `User` or `Project` objects from `ProjectUser` and `GroupProject` objects using the `id` attribute:

```
for gr_project in group.projects.list():
    # lazy object creation avoids a Gitlab API request
    project = gl.projects.get(gr_project.id, lazy=True)
    project.default_branch = 'develop'
    project.save()
```

7.11.9 Changes from 0.21 to 1.0.0

1.0.0 brings a stable python-gitlab API for the v4 Gitlab API. v3 is still used by default.

v4 is mostly compatible with the v3, but some important changes have been introduced. Make sure to read Switching to GitLab API v4.

The development focus will be v4 from now on. v3 has been deprecated by GitLab and will disappear from python-gitlab at some point.

7.11.10 Changes from 0.20 to 0.21

- Initial support for the v4 API (experimental)

  The support for v4 is stable enough to be tested, but some features might be broken. Please report issues to https://github.com/python-gitlab/python-gitlab/issues/

  Be aware that the python-gitlab API for v4 objects might change in the next releases.

> **Warning**
>
> Consider defining explicitly which API version you want to use in the configuration files or in your `gitlab.Gitlab` instances. The default will change from v3 to v4 soon.

- Several methods have been deprecated in the `gitlab.Gitlab` class:
  - `credentials_auth()` is deprecated and will be removed. Call `auth()`.
  - `token_auth()` is deprecated and will be removed. Call `auth()`.
  - `set_url()` is deprecated, create a new `Gitlab` instance if you need an updated URL.
  - `set_token()` is deprecated, use the `private_token` argument of the `Gitlab` constructor.
  - `set_credentials()` is deprecated, use the `email` and `password` arguments of the `Gitlab` constructor.
- The service listing method (`ProjectServiceManager.list()`) now returns a python list instead of a JSON string.

7.11.11 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

  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

  Related issue: https://github.com/python-gitlab/python-gitlab/issues/212

7.12 FAQ

7.12.1 General

I cannot edit the merge request / issue I've just retrieved.

It is likely that you used a `MergeRequest`, `GroupMergeRequest`, `Issue` or `GroupIssue` object. These objects cannot be edited. But you can create a new `ProjectMergeRequest` or `ProjectIssue` object to apply changes. For example:

```
issue = gl.issues.list(get_all=False)[0]
project = gl.projects.get(issue.project_id, lazy=True)
editable_issue = project.issues.get(issue.iid, lazy=True)
# you can now edit the object
```

See the *merge requests example* and the *issues examples*.

How can I clone the repository of a project?

python-gitlab does not provide an API to clone a project. You have to use a git library or call the `git` command.

The git URI is exposed in the `ssh_url_to_repo` attribute of `Project` objects.

Example:

```
import subprocess

project = gl.projects.create(data)  # or gl.projects.get(project_id)
print(project.attributes)  # displays all the attributes
git_url = project.ssh_url_to_repo
subprocess.call(['git', 'clone', git_url])
```

Not all items are returned from the API

If you've passed `all=True` to the API and still cannot see all items returned, use `get_all=True` (or `--get-all` via the CLI) instead. See *Pagination* for more details.

7.12.2 Common errors

`AttributeError` when accessing object attributes retrieved via `list()`

Fetching a list of objects does not always include all attributes in the objects. To retrieve an object with all attributes, use a `get()` call.

Example with projects:

```
for project in gl.projects.list(iterator=True):
    # Retrieve project object with all attributes
    project = gl.projects.get(project.id)
```

`AttributeError` when accessing attributes after `save()` or `refresh()`

You are most likely trying to access an attribute that was not returned by the server on the second request. Please look at the documentation in *Attributes in updated objects* to see how to avoid this.

`TypeError` when accessing object attributes

When you encounter errors such as `object is not iterable` or `object is not subscriptable` when trying to access object attributes returned from the server, you are most likely trying to access an attribute that is shadowed by python-gitlab's own methods or managers.

You can use the object's `attributes` dictionary to access it directly instead. See the *Gitlab Objects* section for more details on how attributes are exposed.

I cannot use the parameter `path` (or some other parameter) as it conflicts with the library

`path` is used by the python-gitlab library and cannot be used as a parameter if wanting to send it to the GitLab instance. You can use the `query_parameters` argument to send arguments that would conflict with python or python-gitlab when using them as kwargs:

```
## invalid, as ``path`` is interpreted by python-gitlab as the Path or full
## URL to query ('/projects' or 'http://whatever/v4/api/projects')
project.commits.list(path='some_file_path', iterator=True)

project.commits.list(query_parameters={'path': 'some_file_path'}, iterator=True)  # OK
```

See *Conflicting Parameters* for more information.

CHAPTER

EIGHT

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

g

gitlab.base, 428
gitlab.cli, 430
gitlab.config, 431
gitlab.const, 432
gitlab.exceptions, 433
gitlab.mixins, 450
gitlab.utils, 460
gitlab.v4, 428
gitlab.v4.objects, 208

INDEX

Symbols

--access-level
    gitlab-current-user-runner-create command line option, 482
    gitlab-group-access-request-approve command line option, 496
    gitlab-group-access-token-create command line option, 498
    gitlab-group-invitation-create command line option, 530
    gitlab-group-invitation-update command line option, 531
    gitlab-group-member-create command line option, 538
    gitlab-group-member-update command line option, 539
    gitlab-group-saml-group-link-create command line option, 553
    gitlab-project-access-request-approve command line option, 593
    gitlab-project-access-token-create command line option, 594
    gitlab-project-invitation-create command line option, 650
    gitlab-project-invitation-update command line option, 651
    gitlab-project-member-create command line option, 684
    gitlab-project-member-update command line option, 686
    gitlab-runner-create command line option, 792
    gitlab-runner-update command line option, 793
--access-raw-diffs
    gitlab-project-merge-request-changes command line option, 688
--action
    gitlab-event-list command line option, 485
    gitlab-project-event-list command line option, 632
    gitlab-todo-list command line option, 801
    gitlab-user-event-list command line option, 814
--actions
    gitlab-project-commit-create command line option, 613
--active
    gitlab-project-feature-flag-create command line option, 636
    gitlab-project-feature-flag-update command line option, 637
    gitlab-project-pipeline-schedule-create command line option, 735
    gitlab-project-pipeline-schedule-update command line option, 737
    gitlab-runner-create command line option, 791
    gitlab-runner-update command line option, 793
    gitlab-user-list command line option, 807
--admin
    gitlab-user-create command line option, 805
    gitlab-user-update command line option, 810
--admin-cicd-variables
    gitlab-group-member-role-create command line option, 541
    gitlab-member-role-create command line option, 569
--admin-compliance-framework
    gitlab-group-member-role-create command line option, 541
    gitlab-member-role-create command line option, 569
--admin-group-member
    gitlab-group-member-role-create command line option, 541
    gitlab-member-role-create command line option, 569
--admin-merge-request
    gitlab-group-member-role-create command

line option, 541
gitlab-member-role-create command line option, 569
--admin-push-rules
gitlab-group-member-role-create command line option, 541
gitlab-member-role-create command line option, 569
--admin-terraform-state
gitlab-group-member-role-create command line option, 541
gitlab-member-role-create command line option, 569
--admin-vulnerability
gitlab-group-member-role-create command line option, 541
gitlab-member-role-create command line option, 569
--admin-web-hook
gitlab-group-member-role-create command line option, 541
gitlab-member-role-create command line option, 569
--admins
gitlab-user-list command line option, 808
--after
gitlab-event-list command line option, 485
gitlab-project-event-list command line option, 632
gitlab-user-event-list command line option, 814
--after-sign-out-path
gitlab-application-settings-update command line option, 470
--all
gitlab-project-commit-list command line option, 614
--all-available
gitlab-group-descendant-group-list command line option, 513
gitlab-group-list command line option, 493
gitlab-group-subgroup-list command line option, 559
--allow-collaboration
gitlab-project-merge-request-create command line option, 689
--allow-force-push
gitlab-project-protected-branch-create command line option, 741
--allow-local-requests-from-hooks-and-services
gitlab-application-settings-update command line option, 471
--allow-local-requests-from-system-hooks
gitlab-application-settings-update command line option, 471
--allow-local-requests-from-web-hooks-and-services
gitlab-application-settings-update command line option, 471
--allow-maintainer-to-push
gitlab-project-merge-request-create command line option, 689
gitlab-project-merge-request-update command line option, 696
--allow-merge-on-skipped-pipeline
gitlab-project-create command line option, 578
gitlab-project-update command line option, 589
--allowed-to-merge
gitlab-project-protected-branch-create command line option, 741
--allowed-to-push
gitlab-project-protected-branch-create command line option, 741
--allowed-to-unprotect
gitlab-project-protected-branch-create command line option, 741
--analytics-access-level
gitlab-project-create command line option, 578
gitlab-project-update command line option, 589
--api-version
gitlab command line option, 465
--applies-to-all-protected-branches
gitlab-project-approval-rule-create command line option, 598
--approval-project-rule-id
gitlab-project-merge-request-approval-rule-create command line option, 698
--approval-rules
gitlab-project-protected-environment-create command line option, 743
--approvals-before-merge
gitlab-project-approval-update command line option, 597
gitlab-project-create command line option, 578
gitlab-project-merge-request-create command line option, 689
gitlab-project-update command line option, 589
--approvals-required
gitlab-group-approval-rule-create command line option, 499
gitlab-project-approval-rule-create

command line option, 598
gitlab-project-merge-request-approval-rule-create command line option, 698
gitlab-project-merge-request-approval-rule-update command line option, 699
gitlab-project-merge-request-approval-update command line option, 697
--approved-by-ids
gitlab-group-merge-request-list command line option, 544
gitlab-merge-request-list command line option, 571
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-merge-request-list command line option, 691
--approver-ids
gitlab-group-merge-request-list command line option, 544
gitlab-merge-request-list command line option, 571
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-merge-request-list command line option, 691
--archive-project
gitlab-group-member-role-create command line option, 541
gitlab-member-role-create command line option, 569
--archived
gitlab-group-project-list command line option, 549
gitlab-project-fork-list command line option, 643
gitlab-project-list command line option, 582
gitlab-shared-project-list command line option, 796
gitlab-starred-project-list command line option, 800
gitlab-user-project-list command line option, 821
--artifact-path
gitlab-project-artifact-raw command line option, 600
--asset-proxy-allowlist
gitlab-application-settings-update command line option, 471
--asset-proxy-enabled
gitlab-application-settings-update command line option, 471
--asset-proxy-url
gitlab-application-settings-update command line option, 471
--asset-proxy-whitelist
gitlab-application-settings-update command line option, 471
--assets
gitlab-project-release-create command line option, 755
--assignee-id
gitlab-group-board-list-create command line option, 507
gitlab-group-issue-list command line option, 532
gitlab-group-merge-request-list command line option, 544
gitlab-issue-list command line option, 565
gitlab-merge-request-list command line option, 571
gitlab-project-board-list-create command line option, 605
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-issue-create command line option, 653
gitlab-project-issue-list command line option, 654
gitlab-project-issue-update command line option, 658
gitlab-project-merge-request-create command line option, 689
gitlab-project-merge-request-list command line option, 691
gitlab-project-merge-request-update command line option, 696
--assignee-ids
gitlab-project-issue-create command line option, 653
gitlab-project-issue-update command line option, 658
gitlab-project-merge-request-create command line option, 689
--auditors
gitlab-user-list command line option, 808
--author-email
gitlab-project-commit-create command line option, 613
gitlab-project-file-create command line option, 640
gitlab-project-file-update command line option, 642
--author-email-regex
gitlab-group-push-rules-create command line option, 551
gitlab-group-push-rules-update command

line option, 552
gitlab-project-push-rules-create command line option, 747
gitlab-project-push-rules-update command line option, 748
--author-id
gitlab-group-epic-list command line option, 515
gitlab-group-issue-list command line option, 532
gitlab-group-merge-request-list command line option, 543
gitlab-issue-list command line option, 565
gitlab-merge-request-list command line option, 571
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-issue-list command line option, 654
gitlab-project-merge-request-list command line option, 691
gitlab-todo-list command line option, 801
--author-name
gitlab-project-commit-create command line option, 614
gitlab-project-file-create command line option, 640
gitlab-project-file-update command line option, 642
--author-username
gitlab-merge-request-list command line option, 571
gitlab-project-deployment-merge-request-list command line option, 628
--auto-cancel-pending-pipelines
gitlab-project-create command line option, 578
gitlab-project-update command line option, 589
--auto-devops-deploy-strategy
gitlab-project-create command line option, 578
gitlab-project-update command line option, 589
--auto-devops-enabled
gitlab-group-create command line option, 491
gitlab-group-update command line option, 495
gitlab-project-create command line option, 578
gitlab-project-update command line option, 590
--autoclose-referenced-issues
gitlab-project-create command line option, 578
gitlab-project-update command line option, 590
--avatar
gitlab-group-create command line option, 491
gitlab-group-update command line option, 495
gitlab-project-create command line option, 578
gitlab-project-update command line option, 590
gitlab-topic-create command line option, 802
gitlab-topic-update command line option, 804
gitlab-user-create command line option, 806
gitlab-user-update command line option, 810
--base-access-level
gitlab-group-member-role-create command line option, 541
gitlab-member-role-create command line option, 569
--before
gitlab-event-list command line option, 485
gitlab-project-event-list command line option, 632
gitlab-user-event-list command line option, 814
--bio
gitlab-user-create command line option, 805
gitlab-user-update command line option, 810
--blocked
gitlab-user-list command line option, 807
--board-id
gitlab-group-board-list-create command line option, 507
gitlab-group-board-list-delete command line option, 507
gitlab-group-board-list-get command line option, 507
gitlab-group-board-list-list command line option, 508
gitlab-group-board-list-update command line option, 508
gitlab-project-board-list-create command line option, 605

gitlab-project-board-list-delete command line option, 605
gitlab-project-board-list-get command line option, 606
gitlab-project-board-list-list command line option, 606
gitlab-project-board-list-update command line option, 607
--body
gitlab-group-epic-discussion-note-create command line option, 518
gitlab-group-epic-discussion-note-update command line option, 519
gitlab-group-epic-note-create command line option, 521
gitlab-group-epic-note-update command line option, 523
gitlab-project-commit-discussion-create command line option, 618
gitlab-project-commit-discussion-note-create command line option, 619
gitlab-project-commit-discussion-note-update command line option, 620
gitlab-project-issue-discussion-create command line option, 661
gitlab-project-issue-discussion-note-create command line option, 662
gitlab-project-issue-discussion-note-update command line option, 664
gitlab-project-issue-note-create command line option, 666
gitlab-project-issue-note-update command line option, 667
gitlab-project-merge-request-discussion-create command line option, 703
gitlab-project-merge-request-discussion-note-create command line option, 705
gitlab-project-merge-request-discussion-note-update command line option, 706
gitlab-project-merge-request-note-create command line option, 709
gitlab-project-merge-request-note-update command line option, 710
gitlab-project-snippet-discussion-create command line option, 775
gitlab-project-snippet-discussion-note-create command line option, 777
gitlab-project-snippet-discussion-note-update command line option, 778
gitlab-project-snippet-note-create command line option, 778
gitlab-project-snippet-note-update command line option, 780
--branch
gitlab-project-branch-create command line option, 607
gitlab-project-commit-cherry-pick command line option, 613
gitlab-project-commit-create command line option, 613
gitlab-project-commit-revert command line option, 616
gitlab-project-file-create command line option, 640
gitlab-project-file-delete command line option, 640
gitlab-project-file-update command line option, 641
gitlab-project-update-submodule command line option, 592
--branch-name-regex
gitlab-group-push-rules-create command line option, 551
gitlab-group-push-rules-update command line option, 552
gitlab-project-push-rules-create command line option, 747
gitlab-project-push-rules-update command line option, 748
--build-coverage-regex
gitlab-project-create command line option, 578
gitlab-project-update command line option, 590
--build-git-strategy
gitlab-project-create command line option, 578
gitlab-project-update command line option, 590
--build-timeout
gitlab-project-create command line option, 578
gitlab-project-update command line option, 590
--builds-access-level
gitlab-project-create command line option, 578
gitlab-project-update command line option, 590
--builds-enabled
gitlab-user-project-create command line option, 820
--bulk-import-id
gitlab-bulk-import-entity-get command line option, 476
gitlab-bulk-import-entity-list command line option, 477
--can-create-group

gitlab-user-create command line option, 806
gitlab-user-update command line option, 810
--can-push
gitlab-project-key-create command line option, 679
gitlab-project-key-update command line option, 681
--certificate
gitlab-project-pages-domain-create command line option, 729
gitlab-project-pages-domain-update command line option, 730
--ci-allow-fork-pipelines-to-run-in-parent-project
gitlab-project-update command line option, 590
--ci-config-path
gitlab-project-create command line option, 579
gitlab-project-update command line option, 590
--ci-default-git-depth
gitlab-project-update command line option, 590
--ci-forward-deployment-enabled
gitlab-project-update command line option, 590
--ci-separated-caches
gitlab-project-update command line option, 590
--close-issue
gitlab-group-notification-settings-update command line option, 548
gitlab-notification-settings-update command line option, 574
gitlab-project-notification-settings-update command line option, 723
--close-merge-request
gitlab-group-notification-settings-update command line option, 548
gitlab-notification-settings-update command line option, 574
gitlab-project-notification-settings-update command line option, 723
--cn
gitlab-group-ldap-group-link-create command line option, 537
--code-owner-approval-required
gitlab-project-protected-branch-create command line option, 741
--color
gitlab-broadcast-message-create command line option, 473
gitlab-broadcast-message-update command line option, 474
gitlab-group-label-create command line option, 534
gitlab-group-label-update command line option, 536
gitlab-project-label-create command line option, 681
gitlab-project-label-update command line option, 683
--color-scheme-id
gitlab-user-create command line option, 806
gitlab-user-update command line option, 810
--comment
gitlab-project-deployment-approval command line option, 625
--commit-committer-check
gitlab-group-push-rules-create command line option, 551
gitlab-group-push-rules-update command line option, 552
gitlab-project-push-rules-create command line option, 747
gitlab-project-push-rules-update command line option, 748
--commit-id
gitlab-project-commit-comment-create command line option, 617
gitlab-project-commit-comment-list command line option, 617
gitlab-project-commit-discussion-create command line option, 618
gitlab-project-commit-discussion-get command line option, 618
gitlab-project-commit-discussion-list command line option, 618
gitlab-project-commit-discussion-note-create command line option, 619
gitlab-project-commit-discussion-note-delete command line option, 620
gitlab-project-commit-discussion-note-get command line option, 620
gitlab-project-commit-discussion-note-update command line option, 620
gitlab-project-commit-status-create command line option, 621
gitlab-project-commit-status-list command line option, 621
gitlab-project-merge-request-draft-note-create command line option, 707
--commit-message
gitlab-project-commit-create command

line option, 613
gitlab-project-file-create command line option, 640
gitlab-project-file-delete command line option, 640
gitlab-project-file-update command line option, 642
--commit-message-negative-regex
gitlab-group-push-rules-create command line option, 551
gitlab-group-push-rules-update command line option, 552
gitlab-project-push-rules-create command line option, 747
gitlab-project-push-rules-update command line option, 748
--commit-message-regex
gitlab-group-push-rules-create command line option, 551
gitlab-group-push-rules-update command line option, 552
gitlab-project-push-rules-create command line option, 747
gitlab-project-push-rules-update command line option, 748
--commit-sha
gitlab-project-update-submodule command line option, 592
--confidential
gitlab-application-create command line option, 466
gitlab-project-issue-create command line option, 653
gitlab-project-issue-update command line option, 658
--confidential-issues-events
gitlab-group-hook-create command line option, 527
gitlab-group-hook-update command line option, 529
gitlab-project-hook-create command line option, 646
gitlab-project-hook-update command line option, 647
--confidential-note-events
gitlab-group-hook-create command line option, 527
gitlab-group-hook-update command line option, 529
--config-file
gitlab command line option, 464
--configuration
gitlab-bulk-import-create command line option, 475
--container-expiration-policy-attributes
gitlab-project-create command line option, 579
gitlab-project-update command line option, 590
--container-registry-access-level
gitlab-project-create command line option, 579
gitlab-project-update command line option, 590
--container-registry-enabled
gitlab-project-create command line option, 579
gitlab-project-update command line option, 590
--container-registry-token-expire-delay
gitlab-application-settings-update command line option, 470
--content
gitlab-ci-lint-create command line option, 477
gitlab-ci-lint-validate command line option, 477
gitlab-group-wiki-create command line option, 562
gitlab-group-wiki-update command line option, 563
gitlab-project-ci-lint-create command line option, 608
gitlab-project-ci-lint-validate command line option, 609
gitlab-project-file-create command line option, 640
gitlab-project-file-update command line option, 641
gitlab-project-snippet-create command line option, 772
gitlab-project-snippet-update command line option, 773
gitlab-project-wiki-create command line option, 789
gitlab-project-wiki-update command line option, 790
gitlab-snippet-create command line option, 798
gitlab-snippet-update command line option, 799
--content-ref
gitlab-project-ci-lint-get command line option, 609
--context
gitlab-project-commit-status-create command line option, 621
--coverage

gitlab-project-commit-status-create command line option, 621
--create-access-level
gitlab-project-protected-tag-create command line option, 744
--created-after
gitlab-audit-event-list command line option, 472
gitlab-group-audit-event-list command line option, 501
gitlab-group-issue-list command line option, 532
gitlab-group-merge-request-list command line option, 543
gitlab-group-service-account-access-token-list command line option, 557
gitlab-issue-list command line option, 566
gitlab-merge-request-list command line option, 571
gitlab-project-audit-event-list command line option, 601
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-issue-list command line option, 654
gitlab-project-merge-request-list command line option, 691
gitlab-project-service-account-access-token-list command line option, 770
gitlab-user-list command line option, 807
--created-at
gitlab-application-settings-update command line option, 470
gitlab-group-epic-discussion-note-create command line option, 518
gitlab-group-epic-note-create command line option, 521
gitlab-project-commit-discussion-create command line option, 618
gitlab-project-commit-discussion-note-create command line option, 619
gitlab-project-issue-create command line option, 653
gitlab-project-issue-discussion-create command line option, 661
gitlab-project-issue-discussion-note-create command line option, 662
gitlab-project-issue-note-create command line option, 666
gitlab-project-merge-request-discussion-create command line option, 703
gitlab-project-merge-request-discussion-note-create command line option, 705
gitlab-project-snippet-discussion-create command line option, 775
gitlab-project-snippet-discussion-note-create command line option, 777
--created-before
gitlab-audit-event-list command line option, 472
gitlab-group-audit-event-list command line option, 501
gitlab-group-issue-list command line option, 532
gitlab-group-merge-request-list command line option, 543
gitlab-group-service-account-access-token-list command line option, 557
gitlab-issue-list command line option, 566
gitlab-merge-request-list command line option, 571
gitlab-project-audit-event-list command line option, 601
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-issue-list command line option, 654
gitlab-project-merge-request-list command line option, 691
gitlab-project-service-account-access-token-list command line option, 770
gitlab-user-list command line option, 807
--cron
gitlab-project-pipeline-schedule-create command line option, 735
gitlab-project-pipeline-schedule-update command line option, 737
--cron-timezone
gitlab-project-pipeline-schedule-create command line option, 735
gitlab-project-pipeline-schedule-update command line option, 737
--custom-attributes
gitlab-user-list command line option, 808
--debug
gitlab command line option, 464
--default-branch
gitlab-project-create command line option, 579
gitlab-project-update command line option, 590
gitlab-user-project-create command line option, 820
--default-branch-protection
gitlab-application-settings-update command line option, 470

gitlab-group-create command line option, 492
gitlab-group-update command line option, 496
--default-group-visibility
gitlab-application-settings-update command line option, 470
--default-project-visibility
gitlab-application-settings-update command line option, 470
--default-projects-limit
gitlab-application-settings-update command line option, 469
--default-snippet-visibility
gitlab-application-settings-update command line option, 470
--deny-delete-tag
gitlab-group-push-rules-create command line option, 551
gitlab-group-push-rules-update command line option, 552
gitlab-project-push-rules-create command line option, 747
gitlab-project-push-rules-update command line option, 748
--deploy-access-levels
gitlab-project-protected-environment-create command line option, 743
--deployed-after
gitlab-merge-request-list command line option, 572
gitlab-project-deployment-merge-request-list command line option, 628
--deployed-before
gitlab-merge-request-list command line option, 572
gitlab-project-deployment-merge-request-list command line option, 628
--deployment-events
gitlab-group-hook-create command line option, 527
gitlab-group-hook-update command line option, 529
--deployment-id
gitlab-project-deployment-merge-request-list command line option, 627
--description
gitlab-application-appearance-update command line option, 467
gitlab-current-user-runner-create command line option, 482
gitlab-group-create command line option, 491
gitlab-group-epic-create command line option, 514
gitlab-group-epic-update command line option, 516
gitlab-group-label-create command line option, 534
gitlab-group-label-update command line option, 536
gitlab-group-member-role-create command line option, 541
gitlab-group-milestone-create command line option, 544
gitlab-group-milestone-update command line option, 546
gitlab-group-service-account-access-token-create command line option, 556
gitlab-group-update command line option, 495
gitlab-group-variable-create command line option, 560
gitlab-group-variable-update command line option, 561
gitlab-member-role-create command line option, 569
gitlab-project-commit-status-create command line option, 621
gitlab-project-create command line option, 579
gitlab-project-export-create command line option, 633
gitlab-project-feature-flag-create command line option, 636
gitlab-project-feature-flag-update command line option, 637
gitlab-project-issue-create command line option, 653
gitlab-project-issue-update command line option, 658
gitlab-project-label-create command line option, 681
gitlab-project-label-update command line option, 683
gitlab-project-merge-request-create command line option, 689
gitlab-project-merge-request-update command line option, 696
gitlab-project-milestone-create command line option, 719
gitlab-project-milestone-update command line option, 721
gitlab-project-pipeline-schedule-create command line option, 735
gitlab-project-pipeline-schedule-update command line option, 736
gitlab-project-release-create command

line option, 755
gitlab-project-release-update command line option, 756
gitlab-project-service-account-access-token-create command line option, 769
gitlab-project-snippet-create command line option, 772
gitlab-project-snippet-update command line option, 773
gitlab-project-trigger-create command line option, 784
gitlab-project-trigger-update command line option, 785
gitlab-project-update command line option, 590
gitlab-project-variable-create command line option, 787
gitlab-project-variable-update command line option, 788
gitlab-runner-create command line option, 791
gitlab-runner-update command line option, 793
gitlab-snippet-create command line option, 797
gitlab-snippet-update command line option, 799
gitlab-topic-create command line option, 802
gitlab-topic-update command line option, 804
gitlab-user-project-create command line option, 820
gitlab-variable-create command line option, 823
gitlab-variable-update command line option, 824
--direct-asset-path
gitlab-project-release-link-create command line option, 757
gitlab-project-release-link-update command line option, 758
--disable-overriding-approvers-per-merge-request
gitlab-project-approval-update command line option, 597
--disabled-oauth-sign-in-sources
gitlab-application-settings-update command line option, 470
--discussion-id
gitlab-group-epic-discussion-note-create command line option, 518
gitlab-group-epic-discussion-note-delete command line option, 518
gitlab-group-epic-discussion-note-get command line option, 519
gitlab-group-epic-discussion-note-update command line option, 519
gitlab-project-commit-discussion-note-create command line option, 619
gitlab-project-commit-discussion-note-delete command line option, 620
gitlab-project-commit-discussion-note-get command line option, 620
gitlab-project-commit-discussion-note-update command line option, 620
gitlab-project-issue-discussion-note-create command line option, 662
gitlab-project-issue-discussion-note-delete command line option, 663
gitlab-project-issue-discussion-note-get command line option, 663
gitlab-project-issue-discussion-note-list command line option, 663
gitlab-project-issue-discussion-note-update command line option, 664
gitlab-project-merge-request-discussion-note-create command line option, 705
gitlab-project-merge-request-discussion-note-delete command line option, 705
gitlab-project-merge-request-discussion-note-get command line option, 705
gitlab-project-merge-request-discussion-note-list command line option, 706
gitlab-project-merge-request-discussion-note-update command line option, 706
gitlab-project-snippet-discussion-note-create command line option, 777
gitlab-project-snippet-discussion-note-delete command line option, 777
gitlab-project-snippet-discussion-note-get command line option, 778
gitlab-project-snippet-discussion-note-update command line option, 778
--discussion-locked
gitlab-project-issue-update command line option, 658
gitlab-project-merge-request-update command line option, 696
--discussion-to-resolve
gitlab-project-issue-create command line option, 653
--domain
gitlab-group-cluster-create command line option, 509
gitlab-group-cluster-update command line option, 510
gitlab-project-cluster-create command line option, 610

gitlab-project-cluster-update command line option, 611
gitlab-project-pages-domain-create command line option, 729
gitlab-project-pages-domain-delete command line option, 729
gitlab-project-pages-domain-get command line option, 730
gitlab-project-pages-domain-update command line option, 730
--domain-allowlist
gitlab-application-settings-update command line option, 470
--domain-blacklist
gitlab-application-settings-update command line option, 470
--domain-blacklist-enabled
gitlab-application-settings-update command line option, 470
--domain-denylist
gitlab-application-settings-update command line option, 470
--domain-denylist-enabled
gitlab-application-settings-update command line option, 470
--domain-whitelist
gitlab-application-settings-update command line option, 470
--dry-run
gitlab-project-ci-lint-create command line option, 608
gitlab-project-ci-lint-get command line option, 609
gitlab-project-ci-lint-validate command line option, 609
--dry-run-ref
gitlab-project-ci-lint-get command line option, 609
--dsa-key-restriction
gitlab-application-settings-update command line option, 471
--due-date
gitlab-group-milestone-create command line option, 544
gitlab-group-milestone-update command line option, 546
gitlab-project-issue-create command line option, 653
gitlab-project-issue-update command line option, 658
gitlab-project-milestone-create command line option, 719
gitlab-project-milestone-update command line option, 721
--duration
gitlab-project-issue-add-spent-time command line option, 652
gitlab-project-issue-time-estimate command line option, 657
gitlab-project-merge-request-add-spent-time command line option, 687
gitlab-project-merge-request-time-estimate command line option, 694
--ecdsa-key-restriction
gitlab-application-settings-update command line option, 471
--ed25519-key-restriction
gitlab-application-settings-update command line option, 471
--email
gitlab-current-user-email-create command line option, 478
gitlab-group-invitation-create command line option, 530
gitlab-group-invitation-delete command line option, 530
gitlab-group-invitation-get command line option, 531
gitlab-group-invitation-update command line option, 531
gitlab-group-service-account-create command line option, 555
gitlab-group-service-account-update command line option, 556
gitlab-project-invitation-create command line option, 650
gitlab-project-invitation-delete command line option, 650
gitlab-project-invitation-get command line option, 651
gitlab-project-invitation-update command line option, 651
gitlab-project-service-account-create command line option, 767
gitlab-project-service-account-update command line option, 768
gitlab-service-account-create command line option, 795
gitlab-service-account-update command line option, 796
gitlab-user-create command line option, 805
gitlab-user-email-create command line option, 813
gitlab-user-update command line option, 809
--email-header-and-footer-enabled
gitlab-application-appearance-update

command line option, 467
--emails-disabled
gitlab-group-create command line option, 491
gitlab-group-update command line option, 495
gitlab-project-create command line option, 579
gitlab-project-update command line option, 590
--emoji
gitlab-current-user-status-update command line option, 483
--enable-ssl-verification
gitlab-group-hook-create command line option, 527
gitlab-group-hook-update command line option, 529
gitlab-project-hook-create command line option, 646
gitlab-project-hook-update command line option, 648
--enabled
gitlab-geo-node-update command line option, 489
gitlab-group-cluster-create command line option, 509
gitlab-project-cluster-create command line option, 610
gitlab-project-remote-mirror-create command line option, 759
gitlab-project-remote-mirror-update command line option, 760
--encoding
gitlab-project-file-create command line option, 640
gitlab-project-file-update command line option, 642
--end-date
gitlab-group-epic-create command line option, 514
gitlab-group-epic-update command line option, 516
--ends-at
gitlab-broadcast-message-create command line option, 473
gitlab-broadcast-message-update command line option, 474
--enforce-auth-checks-on-uploads
gitlab-project-update command line option, 590
--enforce-terms
gitlab-application-settings-update command line option, 471
--entities
gitlab-bulk-import-create command line option, 475
--entity-id
gitlab-audit-event-list command line option, 472
--entity-type
gitlab-audit-event-list command line option, 472
--environment
gitlab-merge-request-list command line option, 572
gitlab-project-deployment-create command line option, 625
gitlab-project-deployment-list command line option, 626
gitlab-project-deployment-merge-request-list command line option, 628
--environment-scope
gitlab-group-cluster-create command line option, 509
gitlab-group-cluster-update command line option, 510
gitlab-group-variable-create command line option, 560
gitlab-group-variable-update command line option, 561
gitlab-project-cluster-create command line option, 610
gitlab-project-cluster-update command line option, 611
gitlab-project-variable-create command line option, 787
gitlab-project-variable-update command line option, 788
--environments-access-level
gitlab-project-create command line option, 580
gitlab-project-update command line option, 591
--epic-id
gitlab-group-epic-discussion-note-create command line option, 518
gitlab-group-epic-discussion-note-delete command line option, 518
gitlab-group-epic-discussion-note-get command line option, 519
gitlab-group-epic-discussion-note-update command line option, 519
gitlab-group-epic-note-create command line option, 521
gitlab-group-epic-note-delete command line option, 522
gitlab-group-epic-note-get command line

option, 522
gitlab-group-epic-note-list command line option, 522
gitlab-group-epic-note-update command line option, 523
gitlab-group-epic-resource-label-event-get command line option, 525
gitlab-group-epic-resource-label-event-list command line option, 525
--epic-iid
gitlab-group-epic-award-emoji-create command line option, 516
gitlab-group-epic-award-emoji-delete command line option, 517
gitlab-group-epic-award-emoji-get command line option, 517
gitlab-group-epic-award-emoji-list command line option, 517
gitlab-group-epic-issue-create command line option, 520
gitlab-group-epic-issue-delete command line option, 520
gitlab-group-epic-issue-list command line option, 520
gitlab-group-epic-issue-update command line option, 521
gitlab-group-epic-note-award-emoji-create command line option, 523
gitlab-group-epic-note-award-emoji-delete command line option, 523
gitlab-group-epic-note-award-emoji-get command line option, 524
gitlab-group-epic-note-award-emoji-list command line option, 524
--epic-issue-id
gitlab-group-epic-issue-delete command line option, 520
gitlab-group-epic-issue-update command line option, 521
--exclude-active
gitlab-user-list command line option, 807
--exclude-external
gitlab-user-list command line option, 808
--exclude-humans
gitlab-user-list command line option, 808
--exclude-internal
gitlab-user-list command line option, 808
--exclude-subgroups
gitlab-group-package-list command line option, 548
--execute-filemode
gitlab-project-file-create command line option, 640
gitlab-project-file-update command line option, 642
--expires-after
gitlab-group-service-account-access-token-list command line option, 557
gitlab-project-service-account-access-token-list command line option, 770
--expires-at
gitlab-deploy-key-create command line option, 483
gitlab-group-access-token-create command line option, 498
gitlab-group-access-token-rotate command line option, 499
gitlab-group-deploy-token-create command line option, 512
gitlab-group-invitation-create command line option, 530
gitlab-group-invitation-update command line option, 531
gitlab-group-member-create command line option, 538
gitlab-group-member-update command line option, 539
gitlab-group-service-account-access-token-create command line option, 556
gitlab-group-service-account-access-token-rotate command line option, 558
gitlab-group-share command line option, 494
gitlab-personal-access-token-rotate command line option, 576
gitlab-project-access-token-create command line option, 594
gitlab-project-access-token-rotate command line option, 596
gitlab-project-deploy-token-create command line option, 623
gitlab-project-invitation-create command line option, 650
gitlab-project-invitation-update command line option, 651
gitlab-project-key-create command line option, 679
gitlab-project-key-update command line option, 681
gitlab-project-member-create command line option, 684
gitlab-project-member-update command line option, 686
gitlab-project-service-account-access-token-create command line option, 769
gitlab-project-service-account-access-token-rotate command line option, 771
gitlab-project-share command line

option, 586
gitlab-user-impersonation-token-create command line option, 816
gitlab-user-personal-access-token-create command line option, 819
--expires-before
gitlab-group-service-account-access-token-list command line option, 557
gitlab-project-service-account-access-token-list command line option, 770
--extern-uid
gitlab-user-create command line option, 805
gitlab-user-list command line option, 808
gitlab-user-update command line option, 810
--external
gitlab-user-create command line option, 806
gitlab-user-list command line option, 807
gitlab-user-update command line option, 810
--external-authorization-classification-label
gitlab-project-create command line option, 579
gitlab-project-update command line option, 590
--external-authorization-service-default-label
gitlab-application-settings-update command line option, 470
--external-authorization-service-enabled
gitlab-application-settings-update command line option, 470
--external-authorization-service-timeout
gitlab-application-settings-update command line option, 470
--external-authorization-service-url
gitlab-application-settings-update command line option, 470
--external-url
gitlab-project-environment-create command line option, 630
gitlab-project-environment-update command line option, 632
gitlab-project-external-status-check-create command line option, 634
gitlab-project-external-status-check-update command line option, 635
--extra-shared-runners-minutes-limit
gitlab-group-create command line option, 492
gitlab-group-update command line option, 496
--fancy
gitlab command line option, 464
--favicon
gitlab-application-appearance-update command line option, 467
--feature-flags-access-level
gitlab-project-create command line option, 580
gitlab-project-update command line option, 591
--fields
gitlab command line option, 464
--file
gitlab-project-secure-file-create command line option, 764
--file-name
gitlab-generic-package-download command line option, 487
gitlab-generic-package-upload command line option, 487
gitlab-project-snippet-create command line option, 772
gitlab-project-snippet-update command line option, 773
gitlab-snippet-create command line option, 798
gitlab-snippet-update command line option, 799
--file-name-regex
gitlab-group-push-rules-create command line option, 551
gitlab-group-push-rules-update command line option, 552
gitlab-project-push-rules-create command line option, 747
gitlab-project-push-rules-update command line option, 748
--file-path
gitlab-project-file-blame command line option, 639
gitlab-project-file-create command line option, 640
gitlab-project-file-delete command line option, 640
gitlab-project-file-get command line option, 641
gitlab-project-file-raw command line option, 641
gitlab-project-file-update command line option, 641
--file-template-project-id
gitlab-application-settings-update command line option, 471
gitlab-group-update command line option, 496

--filename
    gitlab-project-upload command line option, 593
    gitlab-project-wiki-upload command line option, 790
--filepath
    gitlab-project-release-link-create command line option, 757
    gitlab-project-release-link-update command line option, 758
    gitlab-project-upload command line option, 593
    gitlab-project-wiki-upload command line option, 790
--files
    gitlab-project-snippet-create command line option, 772
    gitlab-project-snippet-update command line option, 773
    gitlab-snippet-create command line option, 798
    gitlab-snippet-update command line option, 799
--files-max-capacity
    gitlab-geo-node-update command line option, 489
--filter
    gitlab-group-ldap-group-link-create command line option, 537
--first-day-of-week
    gitlab-application-settings-update command line option, 471
--first-parent
    gitlab-project-commit-list command line option, 615
--font
    gitlab-broadcast-message-create command line option, 473
    gitlab-broadcast-message-update command line option, 474
--footer-message
    gitlab-application-appearance-update command line option, 467
--forked-from-id
    gitlab-project-create-fork-relation command line option, 581
--forking-access-level
    gitlab-project-create command line option, 579
    gitlab-project-update command line option, 590
--format
    gitlab-group-wiki-create command line option, 562
    gitlab-group-wiki-update command line option, 563
    gitlab-project-repository-archive command line option, 584
    gitlab-project-wiki-create command line option, 789
    gitlab-project-wiki-update command line option, 790
--from-
    gitlab-project-repository-compare command line option, 584
--fullname
    gitlab-license-get command line option, 567
--geo-node-allowed-ips
    gitlab-application-settings-update command line option, 471
--get-all
    gitlab-application-list command line option, 466
    gitlab-audit-event-list command line option, 472
    gitlab-broadcast-message-list command line option, 474
    gitlab-bulk-import-all-entity-list command line option, 476
    gitlab-bulk-import-entity-list command line option, 477
    gitlab-bulk-import-list command line option, 475
    gitlab-current-user-email-list command line option, 479
    gitlab-current-user-gpg-key-list command line option, 480
    gitlab-current-user-key-list command line option, 481
    gitlab-deploy-key-list command line option, 483
    gitlab-deploy-token-list command line option, 484
    gitlab-dockerfile-list command line option, 485
    gitlab-event-list command line option, 485
    gitlab-feature-list command line option, 486
    gitlab-geo-node-list command line option, 488
    gitlab-gitignore-list command line option, 489
    gitlab-gitlabciyml-list command line option, 490
    gitlab-group-access-request-list command line option, 497

gitlab-group-access-token-list command line option, 498
gitlab-group-approval-rule-list command line option, 500
gitlab-group-audit-event-list command line option, 501
gitlab-group-badge-list command line option, 502
gitlab-group-billable-member-list command line option, 504
gitlab-group-billable-member-membership-list command line option, 505
gitlab-group-board-list command line option, 506
gitlab-group-board-list-list command line option, 508
gitlab-group-cluster-list command line option, 510
gitlab-group-custom-attribute-list command line option, 511
gitlab-group-deploy-token-list command line option, 513
gitlab-group-descendant-group-list command line option, 514
gitlab-group-epic-award-emoji-list command line option, 517
gitlab-group-epic-issue-list command line option, 520
gitlab-group-epic-list command line option, 515
gitlab-group-epic-note-award-emoji-list command line option, 524
gitlab-group-epic-note-list command line option, 522
gitlab-group-epic-resource-label-event-list command line option, 525
gitlab-group-hook-list command line option, 528
gitlab-group-invitation-list command line option, 531
gitlab-group-issue-list command line option, 533
gitlab-group-iteration-list command line option, 534
gitlab-group-label-list command line option, 535
gitlab-group-ldap-group-link-list command line option, 537
gitlab-group-list command line option, 493
gitlab-group-member-all-list command line option, 540
gitlab-group-member-list command line option, 539
gitlab-group-member-role-list command line option, 542
gitlab-group-merge-request-list command line option, 544
gitlab-group-milestone-list command line option, 546
gitlab-group-package-list command line option, 549
gitlab-group-project-list command line option, 550
gitlab-group-runner-list command line option, 553
gitlab-group-saml-group-link-list command line option, 554
gitlab-group-service-account-access-token-list command line option, 558
gitlab-group-service-account-list command line option, 555
gitlab-group-subgroup-list command line option, 559
gitlab-group-variable-list command line option, 561
gitlab-group-wiki-list command line option, 563
gitlab-hook-list command line option, 564
gitlab-issue-list command line option, 566
gitlab-ldap-group-list command line option, 567
gitlab-license-list command line option, 568
gitlab-member-role-list command line option, 570
gitlab-merge-request-list command line option, 572
gitlab-namespace-list command line option, 573
gitlab-pages-domain-list command line option, 575
gitlab-personal-access-token-list command line option, 576
gitlab-project-access-request-list command line option, 594
gitlab-project-access-token-list command line option, 595
gitlab-project-approval-rule-list command line option, 599
gitlab-project-audit-event-list command line option, 601
gitlab-project-badge-list command line option, 602
gitlab-project-board-list command line option, 604
gitlab-project-board-list-list command

line option, 606
gitlab-project-branch-list command line option, 608
gitlab-project-cluster-agent-list command line option, 613
gitlab-project-cluster-list command line option, 611
gitlab-project-commit-comment-list command line option, 617
gitlab-project-commit-discussion-list command line option, 619
gitlab-project-commit-list command line option, 615
gitlab-project-commit-status-list command line option, 622
gitlab-project-custom-attribute-list command line option, 623
gitlab-project-deploy-token-list command line option, 624
gitlab-project-deployment-list command line option, 626
gitlab-project-deployment-merge-request-list command line option, 629
gitlab-project-dockerfile-template-list command line option, 629
gitlab-project-environment-list command line option, 631
gitlab-project-event-list command line option, 632
gitlab-project-external-status-check-list command line option, 635
gitlab-project-feature-flag-list command line option, 637
gitlab-project-feature-flag-user-list-list command line option, 639
gitlab-project-fork-list command line option, 643
gitlab-project-gitignore-template-list command line option, 644
gitlab-project-gitlabciyml-template-list command line option, 645
gitlab-project-hook-list command line option, 647
gitlab-project-integration-list command line option, 649
gitlab-project-invitation-list command line option, 651
gitlab-project-issue-award-emoji-list command line option, 660
gitlab-project-issue-discussion-list command line option, 662
gitlab-project-issue-discussion-note-list command line option, 663
gitlab-project-issue-link-list command line option, 665
gitlab-project-issue-list command line option, 655
gitlab-project-issue-note-award-emoji-list command line option, 669
gitlab-project-issue-note-list command line option, 667
gitlab-project-issue-resource-iteration-event-list command line option, 670
gitlab-project-issue-resource-label-event-list command line option, 671
gitlab-project-issue-resource-milestone-event-list command line option, 672
gitlab-project-issue-resource-state-event-list command line option, 673
gitlab-project-issue-resource-weight-event-list command line option, 674
gitlab-project-issue-template-list command line option, 674
gitlab-project-job-list command line option, 677
gitlab-project-key-list command line option, 680
gitlab-project-label-list command line option, 682
gitlab-project-license-template-list command line option, 684
gitlab-project-list command line option, 583
gitlab-project-member-all-list command line option, 686
gitlab-project-member-list command line option, 685
gitlab-project-merge-request-approval-rule-list command line option, 699
gitlab-project-merge-request-award-emoji-list command line option, 701
gitlab-project-merge-request-diff-list command line option, 702
gitlab-project-merge-request-discussion-list command line option, 704
gitlab-project-merge-request-discussion-note-list command line option, 706
gitlab-project-merge-request-draft-note-list command line option, 708
gitlab-project-merge-request-list command line option, 692
gitlab-project-merge-request-note-award-emoji-list command line option, 712
gitlab-project-merge-request-note-list command line option, 710
gitlab-project-merge-request-pipeline-list command line option, 713
gitlab-project-merge-request-resource-label-event-list

command line option, 714
gitlab-project-merge-request-resource-milestone-event-list command line option, 715
gitlab-project-merge-request-resource-state-event-list command line option, 716
gitlab-project-merge-request-reviewer-detail-list command line option, 716
gitlab-project-merge-request-status-check-list command line option, 717
gitlab-project-merge-request-template-list command line option, 718
gitlab-project-merge-train-list command line option, 718
gitlab-project-milestone-list command line option, 720
gitlab-project-note-list command line option, 722
gitlab-project-package-file-list command line option, 725
gitlab-project-package-list command line option, 724
gitlab-project-package-pipeline-list command line option, 726
gitlab-project-package-protection-rule-list command line option, 727
gitlab-project-pages-domain-list command line option, 730
gitlab-project-pipeline-bridge-list command line option, 733
gitlab-project-pipeline-job-list command line option, 734
gitlab-project-pipeline-list command line option, 732
gitlab-project-pipeline-schedule-list command line option, 736
gitlab-project-pipeline-schedule-pipeline-list command line option, 737
gitlab-project-pipeline-variable-list command line option, 740
gitlab-project-protected-branch-list command line option, 742
gitlab-project-protected-environment-list command line option, 744
gitlab-project-protected-tag-list command line option, 745
gitlab-project-registry-protection-rule-list command line option, 749
gitlab-project-registry-repository-list command line option, 751
gitlab-project-registry-repository-protection-rule-list command line option, 752
gitlab-project-registry-tag-list command line option, 754
gitlab-project-release-link-list command line option, 758
gitlab-project-release-list command line option, 756
gitlab-project-remote-mirror-list command line option, 760
gitlab-project-resource-group-list command line option, 761
gitlab-project-resource-group-upcoming-job-list command line option, 762
gitlab-project-runner-list command line option, 763
gitlab-project-secure-file-list command line option, 765
gitlab-project-service-account-access-token-list command line option, 770
gitlab-project-service-account-list command line option, 768
gitlab-project-service-list command line option, 766
gitlab-project-snippet-award-emoji-list command line option, 775
gitlab-project-snippet-discussion-list command line option, 776
gitlab-project-snippet-list command line option, 772
gitlab-project-snippet-note-award-emoji-list command line option, 782
gitlab-project-snippet-note-list command line option, 779
gitlab-project-tag-list command line option, 784
gitlab-project-trigger-list command line option, 785
gitlab-project-user-list command line option, 786
gitlab-project-variable-list command line option, 788
gitlab-project-wiki-list command line option, 790
gitlab-runner-all-list command line option, 794
gitlab-runner-job-list command line option, 794
gitlab-runner-list command line option, 792
gitlab-service-account-list command line option, 795
gitlab-shared-project-list command line option, 797
gitlab-snippet-list command line option, 798
gitlab-starred-project-list command line option, 801
gitlab-todo-list command line option, 801

gitlab-topic-list command line option, 803
gitlab-user-activities-list command line option, 810
gitlab-user-contributed-project-list command line option, 811
gitlab-user-custom-attribute-list command line option, 812
gitlab-user-email-list command line option, 813
gitlab-user-event-list command line option, 814
gitlab-user-gpg-key-list command line option, 815
gitlab-user-impersonation-token-list command line option, 817
gitlab-user-key-list command line option, 818
gitlab-user-list command line option, 808
gitlab-user-membership-list command line option, 819
gitlab-user-project-list command line option, 822
gitlab-variable-list command line option, 823
--gitlab
gitlab command line option, 464
--gravatar-enabled
gitlab-application-settings-update command line option, 470
--group-access
gitlab-group-ldap-group-link-create command line option, 537
gitlab-group-share command line option, 494
gitlab-project-share command line option, 586
--group-id
gitlab-current-user-runner-create command line option, 482
gitlab-group-access-request-approve command line option, 496
gitlab-group-access-request-create command line option, 496
gitlab-group-access-request-delete command line option, 497
gitlab-group-access-request-list command line option, 497
gitlab-group-access-token-create command line option, 497
gitlab-group-access-token-delete command line option, 498
gitlab-group-access-token-get command line option, 498
gitlab-group-access-token-list command line option, 498
gitlab-group-access-token-rotate command line option, 499
gitlab-group-approval-rule-create command line option, 499
gitlab-group-approval-rule-get command line option, 500
gitlab-group-approval-rule-list command line option, 500
gitlab-group-approval-rule-update command line option, 500
gitlab-group-audit-event-get command line option, 501
gitlab-group-audit-event-list command line option, 501
gitlab-group-badge-create command line option, 502
gitlab-group-badge-delete command line option, 502
gitlab-group-badge-get command line option, 502
gitlab-group-badge-list command line option, 502
gitlab-group-badge-render command line option, 503
gitlab-group-badge-update command line option, 503
gitlab-group-billable-member-delete command line option, 504
gitlab-group-billable-member-list command line option, 504
gitlab-group-billable-member-membership-list command line option, 504
gitlab-group-board-create command line option, 505
gitlab-group-board-delete command line option, 505
gitlab-group-board-get command line option, 506
gitlab-group-board-list command line option, 506
gitlab-group-board-list-create command line option, 507
gitlab-group-board-list-delete command line option, 507
gitlab-group-board-list-get command line option, 507
gitlab-group-board-list-list command line option, 508
gitlab-group-board-list-update command line option, 508
gitlab-group-board-update command line option, 506

gitlab-group-cluster-create command line option, 509
gitlab-group-cluster-delete command line option, 509
gitlab-group-cluster-get command line option, 509
gitlab-group-cluster-list command line option, 510
gitlab-group-cluster-update command line option, 510
gitlab-group-custom-attribute-delete command line option, 511
gitlab-group-custom-attribute-get command line option, 511
gitlab-group-custom-attribute-list command line option, 511
gitlab-group-deploy-token-create command line option, 512
gitlab-group-deploy-token-delete command line option, 512
gitlab-group-deploy-token-get command line option, 512
gitlab-group-deploy-token-list command line option, 513
gitlab-group-descendant-group-list command line option, 513
gitlab-group-epic-award-emoji-create command line option, 516
gitlab-group-epic-award-emoji-delete command line option, 517
gitlab-group-epic-award-emoji-get command line option, 517
gitlab-group-epic-award-emoji-list command line option, 517
gitlab-group-epic-create command line option, 514
gitlab-group-epic-delete command line option, 515
gitlab-group-epic-discussion-note-create command line option, 518
gitlab-group-epic-discussion-note-delete command line option, 518
gitlab-group-epic-discussion-note-get command line option, 519
gitlab-group-epic-discussion-note-update command line option, 519
gitlab-group-epic-get command line option, 515
gitlab-group-epic-issue-create command line option, 520
gitlab-group-epic-issue-delete command line option, 520
gitlab-group-epic-issue-list command line option, 520
gitlab-group-epic-issue-update command line option, 521
gitlab-group-epic-list command line option, 515
gitlab-group-epic-note-award-emoji-create command line option, 523
gitlab-group-epic-note-award-emoji-delete command line option, 523
gitlab-group-epic-note-award-emoji-get command line option, 524
gitlab-group-epic-note-award-emoji-list command line option, 524
gitlab-group-epic-note-create command line option, 521
gitlab-group-epic-note-delete command line option, 522
gitlab-group-epic-note-get command line option, 522
gitlab-group-epic-note-list command line option, 522
gitlab-group-epic-note-update command line option, 523
gitlab-group-epic-resource-label-event-get command line option, 525
gitlab-group-epic-resource-label-event-list command line option, 525
gitlab-group-epic-update command line option, 516
gitlab-group-export-create command line option, 526
gitlab-group-export-download command line option, 526
gitlab-group-export-get command line option, 526
gitlab-group-hook-create command line option, 527
gitlab-group-hook-delete command line option, 527
gitlab-group-hook-get command line option, 528
gitlab-group-hook-list command line option, 528
gitlab-group-hook-update command line option, 529
gitlab-group-import-get command line option, 529
gitlab-group-invitation-create command line option, 530
gitlab-group-invitation-delete command line option, 530
gitlab-group-invitation-get command line option, 531
gitlab-group-invitation-list command line option, 531

gitlab-group-invitation-update command line option, 531
gitlab-group-issue-list command line option, 532
gitlab-group-issues-statistics-get command line option, 533
gitlab-group-iteration-list command line option, 534
gitlab-group-label-create command line option, 534
gitlab-group-label-delete command line option, 535
gitlab-group-label-get command line option, 535
gitlab-group-label-list command line option, 535
gitlab-group-label-subscribe command line option, 536
gitlab-group-label-unsubscribe command line option, 536
gitlab-group-label-update command line option, 536
gitlab-group-ldap-group-link-create command line option, 537
gitlab-group-ldap-group-link-delete command line option, 537
gitlab-group-ldap-group-link-list command line option, 537
gitlab-group-member-all-get command line option, 540
gitlab-group-member-all-list command line option, 540
gitlab-group-member-create command line option, 538
gitlab-group-member-delete command line option, 538
gitlab-group-member-get command line option, 539
gitlab-group-member-list command line option, 539
gitlab-group-member-role-create command line option, 541
gitlab-group-member-role-delete command line option, 542
gitlab-group-member-role-list command line option, 542
gitlab-group-member-update command line option, 539
gitlab-group-merge-request-list command line option, 543
gitlab-group-milestone-create command line option, 544
gitlab-group-milestone-delete command line option, 545
gitlab-group-milestone-get command line option, 545
gitlab-group-milestone-issues command line option, 545
gitlab-group-milestone-list command line option, 545
gitlab-group-milestone-merge-requests command line option, 546
gitlab-group-milestone-update command line option, 546
gitlab-group-notification-settings-get command line option, 547
gitlab-group-notification-settings-update command line option, 547
gitlab-group-package-list command line option, 548
gitlab-group-project-list command line option, 549
gitlab-group-push-rules-create command line option, 550
gitlab-group-push-rules-delete command line option, 551
gitlab-group-push-rules-get command line option, 551
gitlab-group-push-rules-update command line option, 552
gitlab-group-runner-list command line option, 552
gitlab-group-saml-group-link-create command line option, 553
gitlab-group-saml-group-link-delete command line option, 553
gitlab-group-saml-group-link-get command line option, 554
gitlab-group-saml-group-link-list command line option, 554
gitlab-group-service-account-access-token-create command line option, 556
gitlab-group-service-account-access-token-delete command line option, 557
gitlab-group-service-account-access-token-list command line option, 557
gitlab-group-service-account-access-token-rotate command line option, 558
gitlab-group-service-account-create command line option, 555
gitlab-group-service-account-delete command line option, 555
gitlab-group-service-account-list command line option, 555
gitlab-group-service-account-update command line option, 556
gitlab-group-share command line option, 494

gitlab-group-subgroup-list command line option, 559
gitlab-group-transfer command line option, 494
gitlab-group-unshare command line option, 494
gitlab-group-variable-create command line option, 560
gitlab-group-variable-delete command line option, 560
gitlab-group-variable-get command line option, 560
gitlab-group-variable-list command line option, 561
gitlab-group-variable-update command line option, 561
gitlab-group-wiki-create command line option, 562
gitlab-group-wiki-delete command line option, 562
gitlab-group-wiki-get command line option, 562
gitlab-group-wiki-list command line option, 563
gitlab-group-wiki-update command line option, 563
gitlab-project-share command line option, 586
gitlab-project-unshare command line option, 587
gitlab-shared-project-list command line option, 796
--group-ids
gitlab-group-approval-rule-create command line option, 499
gitlab-project-approval-rule-create command line option, 598
gitlab-project-merge-request-approval-rule-create command line option, 698
gitlab-project-merge-request-approval-rule-update command line option, 700
--group-with-project-templates-id
gitlab-project-create command line option, 579
--header-logo
gitlab-application-appearance-update command line option, 467
--header-message
gitlab-application-appearance-update command line option, 467
--help
gitlab command line option, 464
gitlab-application command line option, 465
gitlab-application-appearance command line option, 466
gitlab-application-appearance-get command line option, 466
gitlab-application-appearance-update command line option, 467
gitlab-application-create command line option, 465
gitlab-application-delete command line option, 466
gitlab-application-list command line option, 466
gitlab-application-settings command line option, 467
gitlab-application-settings-get command line option, 467
gitlab-application-settings-update command line option, 469
gitlab-application-statistics command line option, 471
gitlab-application-statistics-get command line option, 472
gitlab-audit-event command line option, 472
gitlab-audit-event-get command line option, 472
gitlab-audit-event-list command line option, 472
gitlab-broadcast-message command line option, 473
gitlab-broadcast-message-create command line option, 473
gitlab-broadcast-message-delete command line option, 473
gitlab-broadcast-message-get command line option, 473
gitlab-broadcast-message-list command line option, 474
gitlab-broadcast-message-update command line option, 474
gitlab-bulk-import command line option, 474
gitlab-bulk-import-all-entity command line option, 475
gitlab-bulk-import-all-entity-list command line option, 476
gitlab-bulk-import-create command line option, 475
gitlab-bulk-import-entity command line option, 476
gitlab-bulk-import-entity-get command line option, 476
gitlab-bulk-import-entity-list command line option, 476

gitlab-bulk-import-get command line option, 475
gitlab-bulk-import-list command line option, 475
gitlab-ci-lint command line option, 477
gitlab-ci-lint-create command line option, 477
gitlab-ci-lint-validate command line option, 477
gitlab-current-user command line option, 478
gitlab-current-user-email command line option, 478
gitlab-current-user-email-create command line option, 478
gitlab-current-user-email-delete command line option, 478
gitlab-current-user-email-get command line option, 478
gitlab-current-user-email-list command line option, 479
gitlab-current-user-get command line option, 478
gitlab-current-user-gpg-key command line option, 479
gitlab-current-user-gpg-key-create command line option, 479
gitlab-current-user-gpg-key-delete command line option, 479
gitlab-current-user-gpg-key-get command line option, 480
gitlab-current-user-gpg-key-list command line option, 480
gitlab-current-user-key command line option, 480
gitlab-current-user-key-create command line option, 480
gitlab-current-user-key-delete command line option, 481
gitlab-current-user-key-get command line option, 481
gitlab-current-user-key-list command line option, 481
gitlab-current-user-runner command line option, 481
gitlab-current-user-runner-create command line option, 482
gitlab-current-user-status command line option, 482
gitlab-current-user-status-get command line option, 482
gitlab-current-user-status-update command line option, 483
gitlab-deploy-key command line option, 483
gitlab-deploy-key-create command line option, 483
gitlab-deploy-key-list command line option, 483
gitlab-deploy-token command line option, 484
gitlab-deploy-token-list command line option, 484
gitlab-dockerfile command line option, 484
gitlab-dockerfile-get command line option, 484
gitlab-dockerfile-list command line option, 484
gitlab-event command line option, 485
gitlab-event-list command line option, 485
gitlab-feature command line option, 486
gitlab-feature-delete command line option, 486
gitlab-feature-list command line option, 486
gitlab-generic-package command line option, 486
gitlab-generic-package-download command line option, 486
gitlab-generic-package-upload command line option, 487
gitlab-geo-node command line option, 487
gitlab-geo-node-current-failures command line option, 487
gitlab-geo-node-delete command line option, 487
gitlab-geo-node-get command line option, 488
gitlab-geo-node-list command line option, 488
gitlab-geo-node-repair command line option, 488
gitlab-geo-node-status command line option, 488
gitlab-geo-node-update command line option, 489
gitlab-gitignore command line option, 489
gitlab-gitignore-get command line option, 489
gitlab-gitignore-list command line option, 489
gitlab-gitlabciyml command line option, 490
gitlab-gitlabciyml-get command line option, 490
gitlab-gitlabciyml-list command line

option, 490
gitlab-group command line option, 490
gitlab-group-access-request command line option, 496
gitlab-group-access-request-approve command line option, 496
gitlab-group-access-request-create command line option, 496
gitlab-group-access-request-delete command line option, 497
gitlab-group-access-request-list command line option, 497
gitlab-group-access-token command line option, 497
gitlab-group-access-token-create command line option, 497
gitlab-group-access-token-delete command line option, 498
gitlab-group-access-token-get command line option, 498
gitlab-group-access-token-list command line option, 498
gitlab-group-access-token-rotate command line option, 499
gitlab-group-approval-rule command line option, 499
gitlab-group-approval-rule-create command line option, 499
gitlab-group-approval-rule-get command line option, 500
gitlab-group-approval-rule-list command line option, 500
gitlab-group-approval-rule-update command line option, 500
gitlab-group-audit-event command line option, 500
gitlab-group-audit-event-get command line option, 501
gitlab-group-audit-event-list command line option, 501
gitlab-group-badge command line option, 501
gitlab-group-badge-create command line option, 501
gitlab-group-badge-delete command line option, 502
gitlab-group-badge-get command line option, 502
gitlab-group-badge-list command line option, 502
gitlab-group-badge-render command line option, 503
gitlab-group-badge-update command line option, 503
gitlab-group-billable-member command line option, 503
gitlab-group-billable-member-delete command line option, 503
gitlab-group-billable-member-list command line option, 504
gitlab-group-billable-member-membership command line option, 504
gitlab-group-billable-member-membership-list command line option, 504
gitlab-group-board command line option, 505
gitlab-group-board-create command line option, 505
gitlab-group-board-delete command line option, 505
gitlab-group-board-get command line option, 505
gitlab-group-board-list command line option, 506
gitlab-group-board-list-create command line option, 507
gitlab-group-board-list-delete command line option, 507
gitlab-group-board-list-get command line option, 507
gitlab-group-board-list-list command line option, 508
gitlab-group-board-list-update command line option, 508
gitlab-group-board-update command line option, 506
gitlab-group-cluster command line option, 508
gitlab-group-cluster-create command line option, 509
gitlab-group-cluster-delete command line option, 509
gitlab-group-cluster-get command line option, 509
gitlab-group-cluster-list command line option, 510
gitlab-group-cluster-update command line option, 510
gitlab-group-create command line option, 491
gitlab-group-custom-attribute command line option, 510
gitlab-group-custom-attribute-delete command line option, 511
gitlab-group-custom-attribute-get command line option, 511
gitlab-group-custom-attribute-list command line option, 511

gitlab-group-delete command line option, 492
gitlab-group-deploy-token command line option, 511
gitlab-group-deploy-token-create command line option, 512
gitlab-group-deploy-token-delete command line option, 512
gitlab-group-deploy-token-get command line option, 512
gitlab-group-deploy-token-list command line option, 512
gitlab-group-descendant-group command line option, 513
gitlab-group-descendant-group-list command line option, 513
gitlab-group-epic command line option, 514
gitlab-group-epic-award-emoji command line option, 516
gitlab-group-epic-award-emoji-create command line option, 516
gitlab-group-epic-award-emoji-delete command line option, 517
gitlab-group-epic-award-emoji-get command line option, 517
gitlab-group-epic-award-emoji-list command line option, 517
gitlab-group-epic-create command line option, 514
gitlab-group-epic-delete command line option, 515
gitlab-group-epic-discussion-note command line option, 518
gitlab-group-epic-discussion-note-create command line option, 518
gitlab-group-epic-discussion-note-delete command line option, 518
gitlab-group-epic-discussion-note-get command line option, 519
gitlab-group-epic-discussion-note-update command line option, 519
gitlab-group-epic-get command line option, 515
gitlab-group-epic-issue command line option, 519
gitlab-group-epic-issue-create command line option, 519
gitlab-group-epic-issue-delete command line option, 520
gitlab-group-epic-issue-list command line option, 520
gitlab-group-epic-issue-update command line option, 521
gitlab-group-epic-list command line option, 515
gitlab-group-epic-note command line option, 521
gitlab-group-epic-note-award-emoji command line option, 523
gitlab-group-epic-note-award-emoji-create command line option, 523
gitlab-group-epic-note-award-emoji-delete command line option, 523
gitlab-group-epic-note-award-emoji-get command line option, 524
gitlab-group-epic-note-award-emoji-list command line option, 524
gitlab-group-epic-note-create command line option, 521
gitlab-group-epic-note-delete command line option, 521
gitlab-group-epic-note-get command line option, 522
gitlab-group-epic-note-list command line option, 522
gitlab-group-epic-note-update command line option, 522
gitlab-group-epic-resource-label-event command line option, 525
gitlab-group-epic-resource-label-event-get command line option, 525
gitlab-group-epic-resource-label-event-list command line option, 525
gitlab-group-epic-update command line option, 516
gitlab-group-export command line option, 525
gitlab-group-export-create command line option, 526
gitlab-group-export-download command line option, 526
gitlab-group-export-get command line option, 526
gitlab-group-get command line option, 492
gitlab-group-hook command line option, 526
gitlab-group-hook-create command line option, 527
gitlab-group-hook-delete command line option, 527
gitlab-group-hook-get command line option, 528
gitlab-group-hook-list command line option, 528
gitlab-group-hook-update command line option, 528
gitlab-group-import command line option,

529
gitlab-group-import-get command line option, 529
gitlab-group-invitation command line option, 530
gitlab-group-invitation-create command line option, 530
gitlab-group-invitation-delete command line option, 530
gitlab-group-invitation-get command line option, 531
gitlab-group-invitation-list command line option, 531
gitlab-group-invitation-update command line option, 531
gitlab-group-issue command line option, 532
gitlab-group-issue-list command line option, 532
gitlab-group-issues-statistics command line option, 533
gitlab-group-issues-statistics-get command line option, 533
gitlab-group-iteration command line option, 533
gitlab-group-iteration-list command line option, 533
gitlab-group-label command line option, 534
gitlab-group-label-create command line option, 534
gitlab-group-label-delete command line option, 535
gitlab-group-label-get command line option, 535
gitlab-group-label-list command line option, 535
gitlab-group-label-subscribe command line option, 535
gitlab-group-label-unsubscribe command line option, 536
gitlab-group-label-update command line option, 536
gitlab-group-ldap-group-link command line option, 536
gitlab-group-ldap-group-link-create command line option, 537
gitlab-group-ldap-group-link-delete command line option, 537
gitlab-group-ldap-group-link-list command line option, 537
gitlab-group-ldap-sync command line option, 492
gitlab-group-list command line option, 492
gitlab-group-member command line option, 538
gitlab-group-member-all command line option, 539
gitlab-group-member-all-get command line option, 540
gitlab-group-member-all-list command line option, 540
gitlab-group-member-create command line option, 538
gitlab-group-member-delete command line option, 538
gitlab-group-member-get command line option, 538
gitlab-group-member-list command line option, 539
gitlab-group-member-role command line option, 540
gitlab-group-member-role-create command line option, 541
gitlab-group-member-role-delete command line option, 542
gitlab-group-member-role-list command line option, 542
gitlab-group-member-update command line option, 539
gitlab-group-merge-request command line option, 543
gitlab-group-merge-request-list command line option, 543
gitlab-group-milestone command line option, 544
gitlab-group-milestone-create command line option, 544
gitlab-group-milestone-delete command line option, 545
gitlab-group-milestone-get command line option, 545
gitlab-group-milestone-issues command line option, 545
gitlab-group-milestone-list command line option, 545
gitlab-group-milestone-merge-requests command line option, 546
gitlab-group-milestone-update command line option, 546
gitlab-group-notification-settings command line option, 547
gitlab-group-notification-settings-get command line option, 547
gitlab-group-notification-settings-update command line option, 547
gitlab-group-package command line

option, 548
gitlab-group-package-list command line option, 548
gitlab-group-project command line option, 549
gitlab-group-project-list command line option, 549
gitlab-group-push-rules command line option, 550
gitlab-group-push-rules-create command line option, 550
gitlab-group-push-rules-delete command line option, 551
gitlab-group-push-rules-get command line option, 551
gitlab-group-push-rules-update command line option, 552
gitlab-group-restore command line option, 493
gitlab-group-runner command line option, 552
gitlab-group-runner-list command line option, 552
gitlab-group-saml-group-link command line option, 553
gitlab-group-saml-group-link-create command line option, 553
gitlab-group-saml-group-link-delete command line option, 553
gitlab-group-saml-group-link-get command line option, 554
gitlab-group-saml-group-link-list command line option, 554
gitlab-group-search command line option, 493
gitlab-group-service-account command line option, 554
gitlab-group-service-account-access-token command line option, 556
gitlab-group-service-account-access-token-create command line option, 556
gitlab-group-service-account-access-token-delete command line option, 557
gitlab-group-service-account-access-token-list command line option, 557
gitlab-group-service-account-access-token-rotate command line option, 558
gitlab-group-service-account-create command line option, 554
gitlab-group-service-account-delete command line option, 555
gitlab-group-service-account-list command line option, 555
gitlab-group-service-account-update command line option, 556
gitlab-group-share command line option, 494
gitlab-group-subgroup command line option, 558
gitlab-group-subgroup-list command line option, 559
gitlab-group-transfer command line option, 494
gitlab-group-transfer-project command line option, 494
gitlab-group-unshare command line option, 494
gitlab-group-update command line option, 495
gitlab-group-variable command line option, 559
gitlab-group-variable-create command line option, 560
gitlab-group-variable-delete command line option, 560
gitlab-group-variable-get command line option, 560
gitlab-group-variable-list command line option, 561
gitlab-group-variable-update command line option, 561
gitlab-group-wiki command line option, 562
gitlab-group-wiki-create command line option, 562
gitlab-group-wiki-delete command line option, 562
gitlab-group-wiki-get command line option, 562
gitlab-group-wiki-list command line option, 563
gitlab-group-wiki-update command line option, 563
gitlab-hook command line option, 563
gitlab-hook-create command line option, 564
gitlab-hook-delete command line option, 564
gitlab-hook-get command line option, 564
gitlab-hook-list command line option, 564
gitlab-issue command line option, 565
gitlab-issue-get command line option, 565
gitlab-issue-list command line option, 565
gitlab-issues-statistics command line option, 566
gitlab-issues-statistics-get command line option, 566

gitlab-key command line option, 566
gitlab-key-get command line option, 566
gitlab-ldap-group command line option, 567
gitlab-ldap-group-list command line option, 567
gitlab-license command line option, 567
gitlab-license-get command line option, 567
gitlab-license-list command line option, 568
gitlab-member-role command line option, 568
gitlab-member-role-create command line option, 569
gitlab-member-role-delete command line option, 570
gitlab-member-role-list command line option, 570
gitlab-merge-request command line option, 570
gitlab-merge-request-list command line option, 571
gitlab-namespace command line option, 572
gitlab-namespace-exists command line option, 572
gitlab-namespace-get command line option, 573
gitlab-namespace-list command line option, 573
gitlab-notification-settings command line option, 573
gitlab-notification-settings-get command line option, 573
gitlab-notification-settings-update command line option, 574
gitlab-pages-domain command line option, 574
gitlab-pages-domain-list command line option, 574
gitlab-personal-access-token command line option, 575
gitlab-personal-access-token-delete command line option, 575
gitlab-personal-access-token-get command line option, 575
gitlab-personal-access-token-list command line option, 575
gitlab-personal-access-token-rotate command line option, 576
gitlab-project command line option, 576
gitlab-project-access-request command line option, 593
gitlab-project-access-request-approve command line option, 593
gitlab-project-access-request-create command line option, 593
gitlab-project-access-request-delete command line option, 593
gitlab-project-access-request-list command line option, 594
gitlab-project-access-token command line option, 594
gitlab-project-access-token-create command line option, 594
gitlab-project-access-token-delete command line option, 595
gitlab-project-access-token-get command line option, 595
gitlab-project-access-token-list command line option, 595
gitlab-project-access-token-rotate command line option, 596
gitlab-project-additional-statistics command line option, 596
gitlab-project-additional-statistics-get command line option, 596
gitlab-project-approval command line option, 596
gitlab-project-approval-get command line option, 596
gitlab-project-approval-rule command line option, 597
gitlab-project-approval-rule-create command line option, 597
gitlab-project-approval-rule-delete command line option, 598
gitlab-project-approval-rule-get command line option, 598
gitlab-project-approval-rule-list command line option, 598
gitlab-project-approval-rule-update command line option, 599
gitlab-project-approval-update command line option, 597
gitlab-project-archive command line option, 576
gitlab-project-artifact command line option, 599
gitlab-project-artifact-delete command line option, 599
gitlab-project-artifact-download command line option, 599
gitlab-project-artifact-raw command line option, 600
gitlab-project-audit-event command line option, 600
gitlab-project-audit-event-get command

line option, 600
gitlab-project-audit-event-list command line option, 601
gitlab-project-badge command line option, 601
gitlab-project-badge-create command line option, 601
gitlab-project-badge-delete command line option, 602
gitlab-project-badge-get command line option, 602
gitlab-project-badge-list command line option, 602
gitlab-project-badge-render command line option, 602
gitlab-project-badge-update command line option, 603
gitlab-project-board command line option, 603
gitlab-project-board-create command line option, 603
gitlab-project-board-delete command line option, 604
gitlab-project-board-get command line option, 604
gitlab-project-board-list command line option, 604, 605
gitlab-project-board-list-create command line option, 605
gitlab-project-board-list-delete command line option, 605
gitlab-project-board-list-get command line option, 606
gitlab-project-board-list-list command line option, 606
gitlab-project-board-list-update command line option, 606
gitlab-project-board-update command line option, 604
gitlab-project-branch command line option, 607
gitlab-project-branch-create command line option, 607
gitlab-project-branch-delete command line option, 607
gitlab-project-branch-get command line option, 607
gitlab-project-branch-list command line option, 608
gitlab-project-ci-lint command line option, 608
gitlab-project-ci-lint-create command line option, 608
gitlab-project-ci-lint-get command line option, 609
gitlab-project-ci-lint-validate command line option, 609
gitlab-project-cluster command line option, 609
gitlab-project-cluster-agent command line option, 611
gitlab-project-cluster-agent-create command line option, 612
gitlab-project-cluster-agent-delete command line option, 612
gitlab-project-cluster-agent-get command line option, 612
gitlab-project-cluster-agent-list command line option, 612
gitlab-project-cluster-create command line option, 610
gitlab-project-cluster-delete command line option, 610
gitlab-project-cluster-get command line option, 610
gitlab-project-cluster-list command line option, 611
gitlab-project-cluster-update command line option, 611
gitlab-project-commit command line option, 613
gitlab-project-commit-cherry-pick command line option, 613
gitlab-project-commit-comment command line option, 616
gitlab-project-commit-comment-create command line option, 617
gitlab-project-commit-comment-list command line option, 617
gitlab-project-commit-create command line option, 613
gitlab-project-commit-diff command line option, 614
gitlab-project-commit-discussion command line option, 617
gitlab-project-commit-discussion-create command line option, 618
gitlab-project-commit-discussion-get command line option, 618
gitlab-project-commit-discussion-list command line option, 618
gitlab-project-commit-discussion-note command line option, 619
gitlab-project-commit-discussion-note-create command line option, 619
gitlab-project-commit-discussion-note-delete command line option, 619
gitlab-project-commit-discussion-note-get

command line option, 620
gitlab-project-commit-discussion-note-update command line option, 620
gitlab-project-commit-get command line option, 614
gitlab-project-commit-list command line option, 614
gitlab-project-commit-merge-requests command line option, 615
gitlab-project-commit-refs command line option, 615
gitlab-project-commit-revert command line option, 616
gitlab-project-commit-sequence command line option, 616
gitlab-project-commit-signature command line option, 616
gitlab-project-commit-status command line option, 621
gitlab-project-commit-status-create command line option, 621
gitlab-project-commit-status-list command line option, 621
gitlab-project-create command line option, 578
gitlab-project-create-fork-relation command line option, 581
gitlab-project-custom-attribute command line option, 622
gitlab-project-custom-attribute-delete command line option, 622
gitlab-project-custom-attribute-get command line option, 622
gitlab-project-custom-attribute-list command line option, 622
gitlab-project-delete command line option, 581
gitlab-project-delete-fork-relation command line option, 581
gitlab-project-delete-merged-branches command line option, 581
gitlab-project-deploy-token command line option, 623
gitlab-project-deploy-token-create command line option, 623
gitlab-project-deploy-token-delete command line option, 623
gitlab-project-deploy-token-get command line option, 624
gitlab-project-deploy-token-list command line option, 624
gitlab-project-deployment command line option, 624
gitlab-project-deployment-approval command line option, 625
gitlab-project-deployment-create command line option, 625
gitlab-project-deployment-get command line option, 625
gitlab-project-deployment-list command line option, 626
gitlab-project-deployment-merge-request command line option, 627
gitlab-project-deployment-merge-request-list command line option, 627
gitlab-project-deployment-update command line option, 626
gitlab-project-dockerfile-template command line option, 629
gitlab-project-dockerfile-template-get command line option, 629
gitlab-project-dockerfile-template-list command line option, 629
gitlab-project-environment command line option, 630
gitlab-project-environment-create command line option, 630
gitlab-project-environment-delete command line option, 630
gitlab-project-environment-get command line option, 630
gitlab-project-environment-list command line option, 631
gitlab-project-environment-stop command line option, 631
gitlab-project-environment-update command line option, 631
gitlab-project-event command line option, 632
gitlab-project-event-list command line option, 632
gitlab-project-export command line option, 633
gitlab-project-export-create command line option, 633
gitlab-project-export-download command line option, 633
gitlab-project-export-get command line option, 633
gitlab-project-external-status-check command line option, 633
gitlab-project-external-status-check-create command line option, 634
gitlab-project-external-status-check-delete command line option, 634
gitlab-project-external-status-check-list command line option, 634
gitlab-project-external-status-check-update

command line option, 635
gitlab-project-feature-flag command line option, 635
gitlab-project-feature-flag-create command line option, 635
gitlab-project-feature-flag-delete command line option, 636
gitlab-project-feature-flag-get command line option, 636
gitlab-project-feature-flag-list command line option, 636
gitlab-project-feature-flag-update command line option, 637
gitlab-project-feature-flag-user-list command line option, 637
gitlab-project-feature-flag-user-list-create command line option, 637
gitlab-project-feature-flag-user-list-delete command line option, 638
gitlab-project-feature-flag-user-list-get command line option, 638
gitlab-project-feature-flag-user-list-list command line option, 638
gitlab-project-feature-flag-user-list-update command line option, 639
gitlab-project-file command line option, 639
gitlab-project-file-blame command line option, 639
gitlab-project-file-create command line option, 640
gitlab-project-file-delete command line option, 640
gitlab-project-file-get command line option, 641
gitlab-project-file-raw command line option, 641
gitlab-project-file-update command line option, 641
gitlab-project-fork command line option, 642
gitlab-project-fork-create command line option, 642
gitlab-project-fork-list command line option, 642
gitlab-project-get command line option, 581
gitlab-project-gitignore-template command line option, 643
gitlab-project-gitignore-template-get command line option, 643
gitlab-project-gitignore-template-list command line option, 644
gitlab-project-gitlabciyml-template command line option, 644
gitlab-project-gitlabciyml-template-get command line option, 644
gitlab-project-gitlabciyml-template-list command line option, 645
gitlab-project-hook command line option, 645
gitlab-project-hook-create command line option, 645
gitlab-project-hook-delete command line option, 646
gitlab-project-hook-get command line option, 646
gitlab-project-hook-list command line option, 646
gitlab-project-hook-update command line option, 647
gitlab-project-housekeeping command line option, 582
gitlab-project-import command line option, 648
gitlab-project-import-get command line option, 648
gitlab-project-integration command line option, 648
gitlab-project-integration-available command line option, 648
gitlab-project-integration-delete command line option, 648
gitlab-project-integration-get command line option, 649
gitlab-project-integration-list command line option, 649
gitlab-project-integration-update command line option, 649
gitlab-project-invitation command line option, 650
gitlab-project-invitation-create command line option, 650
gitlab-project-invitation-delete command line option, 650
gitlab-project-invitation-get command line option, 651
gitlab-project-invitation-list command line option, 651
gitlab-project-invitation-update command line option, 651
gitlab-project-issue command line option, 652
gitlab-project-issue-add-spent-time command line option, 652
gitlab-project-issue-award-emoji command line option, 659
gitlab-project-issue-award-emoji-create

command line option, 659
gitlab-project-issue-award-emoji-delete command line option, 659
gitlab-project-issue-award-emoji-get command line option, 660
gitlab-project-issue-award-emoji-list command line option, 660
gitlab-project-issue-closed-by command line option, 652
gitlab-project-issue-create command line option, 652
gitlab-project-issue-delete command line option, 653
gitlab-project-issue-discussion command line option, 660
gitlab-project-issue-discussion-create command line option, 661
gitlab-project-issue-discussion-get command line option, 661
gitlab-project-issue-discussion-list command line option, 661
gitlab-project-issue-discussion-note command line option, 662
gitlab-project-issue-discussion-note-create command line option, 662
gitlab-project-issue-discussion-note-delete command line option, 662
gitlab-project-issue-discussion-note-get command line option, 663
gitlab-project-issue-discussion-note-list command line option, 663
gitlab-project-issue-discussion-note-update command line option, 664
gitlab-project-issue-get command line option, 653
gitlab-project-issue-link command line option, 664
gitlab-project-issue-link-create command line option, 664
gitlab-project-issue-link-delete command line option, 664
gitlab-project-issue-link-list command line option, 665
gitlab-project-issue-list command line option, 654
gitlab-project-issue-move command line option, 655
gitlab-project-issue-note command line option, 665
gitlab-project-issue-note-award-emoji command line option, 667
gitlab-project-issue-note-award-emoji-create command line option, 667
gitlab-project-issue-note-award-emoji-delete command line option, 668
gitlab-project-issue-note-award-emoji-get command line option, 668
gitlab-project-issue-note-award-emoji-list command line option, 668
gitlab-project-issue-note-create command line option, 665
gitlab-project-issue-note-delete command line option, 666
gitlab-project-issue-note-get command line option, 666
gitlab-project-issue-note-list command line option, 666
gitlab-project-issue-note-update command line option, 667
gitlab-project-issue-participants command line option, 655
gitlab-project-issue-related-merge-requests command line option, 655
gitlab-project-issue-reorder command line option, 656
gitlab-project-issue-reset-spent-time command line option, 656
gitlab-project-issue-reset-time-estimate command line option, 656
gitlab-project-issue-resource-iteration-event command line option, 669
gitlab-project-issue-resource-iteration-event-get command line option, 669
gitlab-project-issue-resource-iteration-event-list command line option, 669
gitlab-project-issue-resource-label-event command line option, 670
gitlab-project-issue-resource-label-event-get command line option, 670
gitlab-project-issue-resource-label-event-list command line option, 670
gitlab-project-issue-resource-milestone-event command line option, 671
gitlab-project-issue-resource-milestone-event-get command line option, 671
gitlab-project-issue-resource-milestone-event-list command line option, 671
gitlab-project-issue-resource-state-event command line option, 672
gitlab-project-issue-resource-state-event-get command line option, 672
gitlab-project-issue-resource-state-event-list command line option, 672
gitlab-project-issue-resource-weight-event command line option, 673
gitlab-project-issue-resource-weight-event-get command line option, 673
gitlab-project-issue-resource-weight-event-list

command line option, 673
gitlab-project-issue-subscribe command line option, 656
gitlab-project-issue-template command line option, 674
gitlab-project-issue-template-get command line option, 674
gitlab-project-issue-template-list command line option, 674
gitlab-project-issue-time-estimate command line option, 657
gitlab-project-issue-time-stats command line option, 657
gitlab-project-issue-todo command line option, 657
gitlab-project-issue-unsubscribe command line option, 657
gitlab-project-issue-update command line option, 658
gitlab-project-issue-user-agent-detail command line option, 659
gitlab-project-issues-statistics command line option, 675
gitlab-project-issues-statistics-get command line option, 675
gitlab-project-job command line option, 675
gitlab-project-job-artifact command line option, 675
gitlab-project-job-artifacts command line option, 675
gitlab-project-job-cancel command line option, 676
gitlab-project-job-delete-artifacts command line option, 676
gitlab-project-job-erase command line option, 676
gitlab-project-job-get command line option, 676
gitlab-project-job-keep-artifacts command line option, 677
gitlab-project-job-list command line option, 677
gitlab-project-job-play command line option, 677
gitlab-project-job-retry command line option, 678
gitlab-project-job-token-scope command line option, 678
gitlab-project-job-token-scope-get command line option, 678
gitlab-project-job-token-scope-update command line option, 678
gitlab-project-job-trace command line option, 678
gitlab-project-key command line option, 679
gitlab-project-key-create command line option, 679
gitlab-project-key-delete command line option, 679
gitlab-project-key-enable command line option, 679
gitlab-project-key-get command line option, 680
gitlab-project-key-list command line option, 680
gitlab-project-key-update command line option, 680
gitlab-project-label command line option, 681
gitlab-project-label-create command line option, 681
gitlab-project-label-delete command line option, 681
gitlab-project-label-get command line option, 681
gitlab-project-label-list command line option, 682
gitlab-project-label-subscribe command line option, 682
gitlab-project-label-unsubscribe command line option, 682
gitlab-project-label-update command line option, 683
gitlab-project-languages command line option, 582
gitlab-project-license-template command line option, 683
gitlab-project-license-template-get command line option, 683
gitlab-project-license-template-list command line option, 683
gitlab-project-list command line option, 582
gitlab-project-member command line option, 684
gitlab-project-member-all command line option, 686
gitlab-project-member-all-get command line option, 686
gitlab-project-member-all-list command line option, 686
gitlab-project-member-create command line option, 684
gitlab-project-member-delete command line option, 684
gitlab-project-member-get command line

option, 685
gitlab-project-member-list command line option, 685
gitlab-project-member-update command line option, 685
gitlab-project-merge-request command line option, 687
gitlab-project-merge-request-add-spent-time command line option, 687
gitlab-project-merge-request-approval command line option, 696
gitlab-project-merge-request-approval-get command line option, 697
gitlab-project-merge-request-approval-rule command line option, 697
gitlab-project-merge-request-approval-rule-create command line option, 697
gitlab-project-merge-request-approval-rule-delete command line option, 698
gitlab-project-merge-request-approval-rule-get command line option, 698
gitlab-project-merge-request-approval-rule-list command line option, 699
gitlab-project-merge-request-approval-rule-update command line option, 699
gitlab-project-merge-request-approval-state command line option, 700
gitlab-project-merge-request-approval-state-get command line option, 700
gitlab-project-merge-request-approval-update command line option, 697
gitlab-project-merge-request-approve command line option, 687
gitlab-project-merge-request-award-emoji command line option, 700
gitlab-project-merge-request-award-emoji-create command line option, 700
gitlab-project-merge-request-award-emoji-delete command line option, 701
gitlab-project-merge-request-award-emoji-get command line option, 701
gitlab-project-merge-request-award-emoji-list command line option, 701
gitlab-project-merge-request-cancel-merge-when-pipeline-succeeds command line option, 687
gitlab-project-merge-request-changes command line option, 688
gitlab-project-merge-request-closes-issues command line option, 688
gitlab-project-merge-request-commits command line option, 688
gitlab-project-merge-request-create command line option, 689
gitlab-project-merge-request-delete command line option, 690
gitlab-project-merge-request-diff command line option, 702
gitlab-project-merge-request-diff-get command line option, 702
gitlab-project-merge-request-diff-list command line option, 702
gitlab-project-merge-request-discussion command line option, 703
gitlab-project-merge-request-discussion-create command line option, 703
gitlab-project-merge-request-discussion-get command line option, 703
gitlab-project-merge-request-discussion-list command line option, 703
gitlab-project-merge-request-discussion-note command line option, 704
gitlab-project-merge-request-discussion-note-create command line option, 704
gitlab-project-merge-request-discussion-note-delete command line option, 705
gitlab-project-merge-request-discussion-note-get command line option, 705
gitlab-project-merge-request-discussion-note-list command line option, 706
gitlab-project-merge-request-discussion-note-update command line option, 706
gitlab-project-merge-request-discussion-update command line option, 704
gitlab-project-merge-request-draft-note command line option, 707
gitlab-project-merge-request-draft-note-create command line option, 707
gitlab-project-merge-request-draft-note-delete command line option, 707
gitlab-project-merge-request-draft-note-get command line option, 708
gitlab-project-merge-request-draft-note-list command line option, 708
gitlab-project-merge-request-draft-note-update command line option, 708
gitlab-project-merge-request-get command line option, 690
gitlab-project-merge-request-list command line option, 691
gitlab-project-merge-request-merge command line option, 692
gitlab-project-merge-request-merge-ref command line option, 692
gitlab-project-merge-request-note command line option, 709
gitlab-project-merge-request-note-award-emoji command line option, 711
gitlab-project-merge-request-note-award-emoji-create

command line option, 711
gitlab-project-merge-request-note-award-emoji-delete command line option, 711
gitlab-project-merge-request-note-award-emoji-get command line option, 711
gitlab-project-merge-request-note-award-emoji-list command line option, 712
gitlab-project-merge-request-note-create command line option, 709
gitlab-project-merge-request-note-delete command line option, 709
gitlab-project-merge-request-note-get command line option, 709
gitlab-project-merge-request-note-list command line option, 710
gitlab-project-merge-request-note-update command line option, 710
gitlab-project-merge-request-participants command line option, 692
gitlab-project-merge-request-pipeline command line option, 712
gitlab-project-merge-request-pipeline-create command line option, 712
gitlab-project-merge-request-pipeline-list command line option, 713
gitlab-project-merge-request-rebase command line option, 693
gitlab-project-merge-request-related-issues command line option, 693
gitlab-project-merge-request-reset-approvals command line option, 693
gitlab-project-merge-request-reset-spent-time command line option, 693
gitlab-project-merge-request-reset-time-estimate command line option, 694
gitlab-project-merge-request-resource-label-event command line option, 713
gitlab-project-merge-request-resource-label-event-get command line option, 713
gitlab-project-merge-request-resource-label-event-list command line option, 714
gitlab-project-merge-request-resource-milestone-event command line option, 714
gitlab-project-merge-request-resource-milestone-event-get command line option, 714
gitlab-project-merge-request-resource-milestone-event-list command line option, 715
gitlab-project-merge-request-resource-state-event command line option, 715
gitlab-project-merge-request-resource-state-event-get command line option, 715
gitlab-project-merge-request-resource-state-event-list command line option, 715
gitlab-project-merge-request-reviewer-detail command line option, 716
gitlab-project-merge-request-reviewer-detail-list command line option, 716
gitlab-project-merge-request-status-check command line option, 716
gitlab-project-merge-request-status-check-list command line option, 717
gitlab-project-merge-request-subscribe command line option, 694
gitlab-project-merge-request-template command line option, 717
gitlab-project-merge-request-template-get command line option, 717
gitlab-project-merge-request-template-list command line option, 718
gitlab-project-merge-request-time-estimate command line option, 694
gitlab-project-merge-request-time-stats command line option, 695
gitlab-project-merge-request-todo command line option, 695
gitlab-project-merge-request-unapprove command line option, 695
gitlab-project-merge-request-unsubscribe command line option, 695
gitlab-project-merge-request-update command line option, 696
gitlab-project-merge-train command line option, 718
gitlab-project-merge-train-list command line option, 718
gitlab-project-milestone command line option, 719
gitlab-project-milestone-create command line option, 719
gitlab-project-milestone-delete command line option, 719
gitlab-project-milestone-get command line option, 719
gitlab-project-milestone-issues command line option, 720
gitlab-project-milestone-list command line option, 720
gitlab-project-milestone-merge-requests command line option, 720
gitlab-project-milestone-update command line option, 721
gitlab-project-mirror-pull command line option, 583
gitlab-project-mirror-pull-details command line option, 584
gitlab-project-note command line option, 721
gitlab-project-note-get command line

option, 721
gitlab-project-note-list command line option, 722
gitlab-project-notification-settings command line option, 722
gitlab-project-notification-settings-get command line option, 722
gitlab-project-notification-settings-update command line option, 723
gitlab-project-package command line option, 723
gitlab-project-package-delete command line option, 723
gitlab-project-package-file command line option, 725
gitlab-project-package-file-delete command line option, 725
gitlab-project-package-file-list command line option, 725
gitlab-project-package-get command line option, 724
gitlab-project-package-list command line option, 724
gitlab-project-package-pipeline command line option, 726
gitlab-project-package-pipeline-list command line option, 726
gitlab-project-package-protection-rule command line option, 726
gitlab-project-package-protection-rule-create command line option, 726
gitlab-project-package-protection-rule-delete command line option, 727
gitlab-project-package-protection-rule-list command line option, 727
gitlab-project-package-protection-rule-update command line option, 728
gitlab-project-pages command line option, 728
gitlab-project-pages-delete command line option, 728
gitlab-project-pages-domain command line option, 729
gitlab-project-pages-domain-create command line option, 729
gitlab-project-pages-domain-delete command line option, 729
gitlab-project-pages-domain-get command line option, 730
gitlab-project-pages-domain-list command line option, 730
gitlab-project-pages-domain-update command line option, 730
gitlab-project-pages-get command line option, 728
gitlab-project-pages-update command line option, 728
gitlab-project-pipeline command line option, 731
gitlab-project-pipeline-bridge command line option, 733
gitlab-project-pipeline-bridge-list command line option, 733
gitlab-project-pipeline-cancel command line option, 731
gitlab-project-pipeline-create command line option, 731
gitlab-project-pipeline-delete command line option, 731
gitlab-project-pipeline-get command line option, 731
gitlab-project-pipeline-job command line option, 733
gitlab-project-pipeline-job-list command line option, 734
gitlab-project-pipeline-list command line option, 732
gitlab-project-pipeline-retry command line option, 733
gitlab-project-pipeline-schedule command line option, 734
gitlab-project-pipeline-schedule-create command line option, 734
gitlab-project-pipeline-schedule-delete command line option, 735
gitlab-project-pipeline-schedule-get command line option, 735
gitlab-project-pipeline-schedule-list command line option, 735
gitlab-project-pipeline-schedule-pipeline command line option, 737
gitlab-project-pipeline-schedule-pipeline-list command line option, 737
gitlab-project-pipeline-schedule-play command line option, 736
gitlab-project-pipeline-schedule-take-ownership command line option, 736
gitlab-project-pipeline-schedule-update command line option, 736
gitlab-project-pipeline-schedule-variable command line option, 737
gitlab-project-pipeline-schedule-variable-create command line option, 738
gitlab-project-pipeline-schedule-variable-delete command line option, 738
gitlab-project-pipeline-schedule-variable-update command line option, 738
gitlab-project-pipeline-test-report

command line option, 739
gitlab-project-pipeline-test-report-get command line option, 739
gitlab-project-pipeline-test-report-summary command line option, 739
gitlab-project-pipeline-test-report-summary-get command line option, 739
gitlab-project-pipeline-variable command line option, 739
gitlab-project-pipeline-variable-list command line option, 740
gitlab-project-protected-branch command line option, 740
gitlab-project-protected-branch-create command line option, 740
gitlab-project-protected-branch-delete command line option, 741
gitlab-project-protected-branch-get command line option, 741
gitlab-project-protected-branch-list command line option, 741
gitlab-project-protected-branch-update command line option, 742
gitlab-project-protected-environment command line option, 742
gitlab-project-protected-environment-create command line option, 742
gitlab-project-protected-environment-delete command line option, 743
gitlab-project-protected-environment-get command line option, 743
gitlab-project-protected-environment-list command line option, 743
gitlab-project-protected-tag command line option, 744
gitlab-project-protected-tag-create command line option, 744
gitlab-project-protected-tag-delete command line option, 744
gitlab-project-protected-tag-get command line option, 744
gitlab-project-protected-tag-list command line option, 745
gitlab-project-pull-mirror command line option, 745
gitlab-project-pull-mirror-create command line option, 745
gitlab-project-pull-mirror-get command line option, 745
gitlab-project-pull-mirror-start command line option, 746
gitlab-project-pull-mirror-update command line option, 746
gitlab-project-push-rules command line option, 746
gitlab-project-push-rules-create command line option, 746
gitlab-project-push-rules-delete command line option, 747
gitlab-project-push-rules-get command line option, 747
gitlab-project-push-rules-update command line option, 748
gitlab-project-registry-protection-rule command line option, 748
gitlab-project-registry-protection-rule-create command line option, 749
gitlab-project-registry-protection-rule-list command line option, 749
gitlab-project-registry-protection-rule-update command line option, 750
gitlab-project-registry-repository command line option, 750
gitlab-project-registry-repository-delete command line option, 750
gitlab-project-registry-repository-list command line option, 750
gitlab-project-registry-repository-protection-rule command line option, 751
gitlab-project-registry-repository-protection-rule-cre command line option, 751
gitlab-project-registry-repository-protection-rule-del command line option, 751
gitlab-project-registry-repository-protection-rule-lis command line option, 752
gitlab-project-registry-repository-protection-rule-upd command line option, 752
gitlab-project-registry-tag command line option, 752
gitlab-project-registry-tag-delete command line option, 753
gitlab-project-registry-tag-delete-in-bulk command line option, 753
gitlab-project-registry-tag-get command line option, 753
gitlab-project-registry-tag-list command line option, 754
gitlab-project-release command line option, 754
gitlab-project-release-create command line option, 754
gitlab-project-release-delete command line option, 755
gitlab-project-release-get command line option, 755
gitlab-project-release-link command line option, 756
gitlab-project-release-link-create

command line option, 756
gitlab-project-release-link-delete command line option, 757
gitlab-project-release-link-get command line option, 757
gitlab-project-release-link-list command line option, 758
gitlab-project-release-link-update command line option, 758
gitlab-project-release-list command line option, 755
gitlab-project-release-update command line option, 756
gitlab-project-remote-mirror command line option, 759
gitlab-project-remote-mirror-create command line option, 759
gitlab-project-remote-mirror-delete command line option, 759
gitlab-project-remote-mirror-list command line option, 759
gitlab-project-remote-mirror-sync command line option, 760
gitlab-project-remote-mirror-update command line option, 760
gitlab-project-repository-archive command line option, 584
gitlab-project-repository-blob command line option, 584
gitlab-project-repository-compare command line option, 584
gitlab-project-repository-contributors command line option, 584
gitlab-project-repository-merge-base command line option, 585
gitlab-project-repository-raw-blob command line option, 585
gitlab-project-repository-tree command line option, 585
gitlab-project-resource-group command line option, 760
gitlab-project-resource-group-get command line option, 761
gitlab-project-resource-group-list command line option, 761
gitlab-project-resource-group-upcoming-job command line option, 762
gitlab-project-resource-group-upcoming-job-list command line option, 762
gitlab-project-resource-group-update command line option, 761
gitlab-project-restore command line option, 585
gitlab-project-runner command line option, 762
gitlab-project-runner-create command line option, 763
gitlab-project-runner-delete command line option, 763
gitlab-project-runner-list command line option, 763
gitlab-project-search command line option, 585
gitlab-project-secure-file command line option, 764
gitlab-project-secure-file-create command line option, 764
gitlab-project-secure-file-delete command line option, 764
gitlab-project-secure-file-download command line option, 764
gitlab-project-secure-file-get command line option, 765
gitlab-project-secure-file-list command line option, 765
gitlab-project-service command line option, 765
gitlab-project-service-account command line option, 767
gitlab-project-service-account-access-token command line option, 768
gitlab-project-service-account-access-token-create command line option, 769
gitlab-project-service-account-access-token-delete command line option, 769
gitlab-project-service-account-access-token-list command line option, 770
gitlab-project-service-account-access-token-rotate command line option, 771
gitlab-project-service-account-create command line option, 767
gitlab-project-service-account-delete command line option, 767
gitlab-project-service-account-list command line option, 768
gitlab-project-service-account-update command line option, 768
gitlab-project-service-available command line option, 765
gitlab-project-service-delete command line option, 766
gitlab-project-service-get command line option, 766
gitlab-project-service-list command line option, 766
gitlab-project-service-update command line option, 766
gitlab-project-share command line

option, 586
gitlab-project-snapshot command line option, 586
gitlab-project-snippet command line option, 771
gitlab-project-snippet-award-emoji command line option, 773
gitlab-project-snippet-award-emoji-create command line option, 774
gitlab-project-snippet-award-emoji-delete command line option, 774
gitlab-project-snippet-award-emoji-get command line option, 774
gitlab-project-snippet-award-emoji-list command line option, 775
gitlab-project-snippet-content command line option, 771
gitlab-project-snippet-create command line option, 771
gitlab-project-snippet-delete command line option, 772
gitlab-project-snippet-discussion command line option, 775
gitlab-project-snippet-discussion-create command line option, 775
gitlab-project-snippet-discussion-get command line option, 776
gitlab-project-snippet-discussion-list command line option, 776
gitlab-project-snippet-discussion-note command line option, 776
gitlab-project-snippet-discussion-note-create command line option, 777
gitlab-project-snippet-discussion-note-delete command line option, 777
gitlab-project-snippet-discussion-note-get command line option, 777
gitlab-project-snippet-discussion-note-update command line option, 778
gitlab-project-snippet-get command line option, 772
gitlab-project-snippet-list command line option, 772
gitlab-project-snippet-note command line option, 778
gitlab-project-snippet-note-award-emoji command line option, 780
gitlab-project-snippet-note-award-emoji-create command line option, 780
gitlab-project-snippet-note-award-emoji-delete command line option, 781
gitlab-project-snippet-note-award-emoji-get command line option, 781
gitlab-project-snippet-note-award-emoji-list command line option, 781
gitlab-project-snippet-note-create command line option, 778
gitlab-project-snippet-note-delete command line option, 779
gitlab-project-snippet-note-get command line option, 779
gitlab-project-snippet-note-list command line option, 779
gitlab-project-snippet-note-update command line option, 780
gitlab-project-snippet-update command line option, 773
gitlab-project-snippet-user-agent-detail command line option, 773
gitlab-project-star command line option, 586
gitlab-project-storage command line option, 782
gitlab-project-storage-get command line option, 782
gitlab-project-tag command line option, 782
gitlab-project-tag-create command line option, 782
gitlab-project-tag-delete command line option, 783
gitlab-project-tag-get command line option, 783
gitlab-project-tag-list command line option, 783
gitlab-project-transfer command line option, 586
gitlab-project-trigger command line option, 784
gitlab-project-trigger-create command line option, 784
gitlab-project-trigger-delete command line option, 784
gitlab-project-trigger-get command line option, 784
gitlab-project-trigger-list command line option, 785
gitlab-project-trigger-pipeline command line option, 587
gitlab-project-trigger-update command line option, 785
gitlab-project-unarchive command line option, 587
gitlab-project-unshare command line option, 587
gitlab-project-unstar command line option, 587
gitlab-project-update command line

option, 589
gitlab-project-update-submodule command line option, 592
gitlab-project-upload command line option, 592
gitlab-project-user command line option, 785
gitlab-project-user-list command line option, 786
gitlab-project-variable command line option, 786
gitlab-project-variable-create command line option, 786
gitlab-project-variable-delete command line option, 787
gitlab-project-variable-get command line option, 787
gitlab-project-variable-list command line option, 787
gitlab-project-variable-update command line option, 788
gitlab-project-wiki command line option, 788
gitlab-project-wiki-create command line option, 788
gitlab-project-wiki-delete command line option, 789
gitlab-project-wiki-get command line option, 789
gitlab-project-wiki-list command line option, 789
gitlab-project-wiki-update command line option, 790
gitlab-project-wiki-upload command line option, 790
gitlab-registry-repository command line option, 790
gitlab-registry-repository-get command line option, 791
gitlab-runner command line option, 791
gitlab-runner-all command line option, 791, 793
gitlab-runner-all-list command line option, 793
gitlab-runner-create command line option, 791
gitlab-runner-delete command line option, 792
gitlab-runner-get command line option, 792
gitlab-runner-job command line option, 794
gitlab-runner-job-list command line option, 794
gitlab-runner-list command line option, 792
gitlab-runner-update command line option, 793
gitlab-runner-verify command line option, 793
gitlab-service-account command line option, 795
gitlab-service-account-create command line option, 795
gitlab-service-account-list command line option, 795
gitlab-service-account-update command line option, 796
gitlab-shared-project command line option, 796
gitlab-shared-project-list command line option, 796
gitlab-snippet command line option, 797
gitlab-snippet-content command line option, 797
gitlab-snippet-create command line option, 797
gitlab-snippet-delete command line option, 798
gitlab-snippet-get command line option, 798
gitlab-snippet-list command line option, 798
gitlab-snippet-list-all command line option, 799
gitlab-snippet-list-public command line option, 799
gitlab-snippet-update command line option, 799
gitlab-snippet-user-agent-detail command line option, 799
gitlab-starred-project command line option, 800
gitlab-starred-project-list command line option, 800
gitlab-todo command line option, 801
gitlab-todo-delete command line option, 801
gitlab-todo-list command line option, 801
gitlab-todo-mark-all-as-done command line option, 802
gitlab-todo-mark-as-done command line option, 802
gitlab-topic command line option, 802
gitlab-topic-create command line option, 802
gitlab-topic-delete command line option, 802

gitlab-topic-get command line option, 803
gitlab-topic-list command line option, 803
gitlab-topic-merge command line option, 803
gitlab-topic-update command line option, 804
gitlab-user command line option, 804
gitlab-user-activate command line option, 804
gitlab-user-activities command line option, 810
gitlab-user-activities-list command line option, 810
gitlab-user-approve command line option, 804
gitlab-user-ban command line option, 804
gitlab-user-block command line option, 805
gitlab-user-contributed-project command line option, 811
gitlab-user-contributed-project-list command line option, 811
gitlab-user-create command line option, 805
gitlab-user-custom-attribute command line option, 811
gitlab-user-custom-attribute-delete command line option, 811
gitlab-user-custom-attribute-get command line option, 812
gitlab-user-custom-attribute-list command line option, 812
gitlab-user-deactivate command line option, 806
gitlab-user-delete command line option, 806
gitlab-user-email command line option, 812
gitlab-user-email-create command line option, 812
gitlab-user-email-delete command line option, 813
gitlab-user-email-get command line option, 813
gitlab-user-email-list command line option, 813
gitlab-user-event command line option, 814
gitlab-user-event-list command line option, 814
gitlab-user-follow command line option, 806
gitlab-user-get command line option, 807
gitlab-user-gpg-key command line option, 814
gitlab-user-gpg-key-create command line option, 814
gitlab-user-gpg-key-delete command line option, 815
gitlab-user-gpg-key-get command line option, 815
gitlab-user-gpg-key-list command line option, 815
gitlab-user-impersonation-token command line option, 816
gitlab-user-impersonation-token-create command line option, 816
gitlab-user-impersonation-token-delete command line option, 816
gitlab-user-impersonation-token-get command line option, 816
gitlab-user-impersonation-token-list command line option, 817
gitlab-user-key command line option, 817
gitlab-user-key-create command line option, 817
gitlab-user-key-delete command line option, 817
gitlab-user-key-get command line option, 818
gitlab-user-key-list command line option, 818
gitlab-user-list command line option, 807
gitlab-user-membership command line option, 818
gitlab-user-membership-get command line option, 818
gitlab-user-membership-list command line option, 819
gitlab-user-personal-access-token command line option, 819
gitlab-user-personal-access-token-create command line option, 819
gitlab-user-project command line option, 820
gitlab-user-project-create command line option, 820
gitlab-user-project-list command line option, 821
gitlab-user-reject command line option, 808
gitlab-user-status command line option, 822
gitlab-user-status-get command line option, 822
gitlab-user-unban command line option, 808

gitlab-user-unblock command line option, 809
gitlab-user-unfollow command line option, 809
gitlab-user-update command line option, 809
gitlab-variable command line option, 822
gitlab-variable-create command line option, 822
gitlab-variable-delete command line option, 823
gitlab-variable-get command line option, 823
gitlab-variable-list command line option, 823
gitlab-variable-update command line option, 824
--home-page-url
gitlab-application-settings-update command line option, 470
--humans
gitlab-user-list command line option, 807
--id
gitlab-application-delete command line option, 466
gitlab-application-settings-update command line option, 469
gitlab-audit-event-get command line option, 472
gitlab-broadcast-message-delete command line option, 473
gitlab-broadcast-message-get command line option, 473
gitlab-broadcast-message-update command line option, 474
gitlab-bulk-import-entity-get command line option, 476
gitlab-bulk-import-get command line option, 475
gitlab-current-user-email-delete command line option, 478
gitlab-current-user-email-get command line option, 479
gitlab-current-user-gpg-key-delete command line option, 479
gitlab-current-user-gpg-key-get command line option, 480
gitlab-current-user-key-delete command line option, 481
gitlab-current-user-key-get command line option, 481
gitlab-geo-node-current-failures command line option, 487
gitlab-geo-node-delete command line option, 488
gitlab-geo-node-get command line option, 488
gitlab-geo-node-repair command line option, 488
gitlab-geo-node-status command line option, 488
gitlab-geo-node-update command line option, 489
gitlab-group-access-request-approve command line option, 496
gitlab-group-access-request-delete command line option, 497
gitlab-group-access-token-delete command line option, 498
gitlab-group-access-token-get command line option, 498
gitlab-group-access-token-rotate command line option, 499
gitlab-group-approval-rule-get command line option, 500
gitlab-group-approval-rule-update command line option, 500
gitlab-group-audit-event-get command line option, 501
gitlab-group-badge-delete command line option, 502
gitlab-group-badge-get command line option, 502
gitlab-group-badge-render command line option, 503
gitlab-group-badge-update command line option, 503
gitlab-group-billable-member-delete command line option, 504
gitlab-group-board-delete command line option, 505
gitlab-group-board-get command line option, 506
gitlab-group-board-list-delete command line option, 507
gitlab-group-board-list-get command line option, 507
gitlab-group-board-list-update command line option, 508
gitlab-group-board-update command line option, 506
gitlab-group-cluster-delete command line option, 509
gitlab-group-cluster-get command line option, 509
gitlab-group-cluster-update command line option, 510
gitlab-group-delete command line option,

492
gitlab-group-deploy-token-delete command line option, 512
gitlab-group-deploy-token-get command line option, 512
gitlab-group-epic-award-emoji-delete command line option, 517
gitlab-group-epic-award-emoji-get command line option, 517
gitlab-group-epic-discussion-note-delete command line option, 518
gitlab-group-epic-discussion-note-get command line option, 519
gitlab-group-epic-discussion-note-update command line option, 519
gitlab-group-epic-note-award-emoji-delete command line option, 524
gitlab-group-epic-note-award-emoji-get command line option, 524
gitlab-group-epic-note-delete command line option, 522
gitlab-group-epic-note-get command line option, 522
gitlab-group-epic-note-update command line option, 523
gitlab-group-epic-resource-label-event-get command line option, 525
gitlab-group-get command line option, 492
gitlab-group-hook-delete command line option, 527
gitlab-group-hook-get command line option, 528
gitlab-group-hook-update command line option, 529
gitlab-group-ldap-group-link-delete command line option, 537
gitlab-group-ldap-sync command line option, 492
gitlab-group-member-all-get command line option, 540
gitlab-group-member-delete command line option, 538
gitlab-group-member-get command line option, 539
gitlab-group-member-role-delete command line option, 542
gitlab-group-member-update command line option, 539
gitlab-group-milestone-delete command line option, 545
gitlab-group-milestone-get command line option, 545
gitlab-group-milestone-issues command line option, 545
gitlab-group-milestone-merge-requests command line option, 546
gitlab-group-milestone-update command line option, 546
gitlab-group-restore command line option, 493
gitlab-group-search command line option, 493
gitlab-group-service-account-access-token-delete command line option, 557
gitlab-group-service-account-access-token-rotate command line option, 558
gitlab-group-service-account-delete command line option, 555
gitlab-group-service-account-update command line option, 556
gitlab-group-share command line option, 494
gitlab-group-transfer command line option, 494
gitlab-group-transfer-project command line option, 494
gitlab-group-unshare command line option, 494
gitlab-group-update command line option, 495
gitlab-hook-delete command line option, 564
gitlab-hook-get command line option, 564
gitlab-issue-get command line option, 565
gitlab-key-get command line option, 566
gitlab-member-role-delete command line option, 570
gitlab-namespace-exists command line option, 572
gitlab-namespace-get command line option, 573
gitlab-personal-access-token-delete command line option, 575
gitlab-personal-access-token-get command line option, 575
gitlab-personal-access-token-rotate command line option, 576
gitlab-project-access-request-approve command line option, 593
gitlab-project-access-request-delete command line option, 594
gitlab-project-access-token-delete command line option, 595
gitlab-project-access-token-get command line option, 595
gitlab-project-access-token-rotate command line option, 596
gitlab-project-approval-rule-delete

command line option, 598
gitlab-project-approval-rule-get command line option, 598
gitlab-project-approval-rule-update command line option, 599
gitlab-project-archive command line option, 576
gitlab-project-audit-event-get command line option, 600
gitlab-project-badge-delete command line option, 602
gitlab-project-badge-get command line option, 602
gitlab-project-badge-render command line option, 603
gitlab-project-badge-update command line option, 603
gitlab-project-board-delete command line option, 604
gitlab-project-board-get command line option, 604
gitlab-project-board-list-delete command line option, 606
gitlab-project-board-list-get command line option, 606
gitlab-project-board-list-update command line option, 607
gitlab-project-board-update command line option, 605
gitlab-project-cluster-agent-delete command line option, 612
gitlab-project-cluster-agent-get command line option, 612
gitlab-project-cluster-delete command line option, 610
gitlab-project-cluster-get command line option, 610
gitlab-project-cluster-update command line option, 611
gitlab-project-commit-cherry-pick command line option, 613
gitlab-project-commit-diff command line option, 614
gitlab-project-commit-discussion-get command line option, 618
gitlab-project-commit-discussion-note-delete command line option, 620
gitlab-project-commit-discussion-note-get command line option, 620
gitlab-project-commit-discussion-note-update command line option, 620
gitlab-project-commit-get command line option, 614
gitlab-project-commit-merge-requests command line option, 615
gitlab-project-commit-refs command line option, 615
gitlab-project-commit-revert command line option, 616
gitlab-project-commit-sequence command line option, 616
gitlab-project-commit-signature command line option, 616
gitlab-project-create-fork-relation command line option, 581
gitlab-project-delete command line option, 581
gitlab-project-delete-fork-relation command line option, 581
gitlab-project-delete-merged-branches command line option, 581
gitlab-project-deploy-token-delete command line option, 624
gitlab-project-deploy-token-get command line option, 624
gitlab-project-deployment-approval command line option, 625
gitlab-project-deployment-get command line option, 625
gitlab-project-deployment-update command line option, 626
gitlab-project-environment-delete command line option, 630
gitlab-project-environment-get command line option, 630
gitlab-project-environment-stop command line option, 631
gitlab-project-environment-update command line option, 632
gitlab-project-external-status-check-delete command line option, 634
gitlab-project-external-status-check-update command line option, 635
gitlab-project-get command line option, 581
gitlab-project-hook-delete command line option, 646
gitlab-project-hook-get command line option, 646
gitlab-project-hook-update command line option, 647
gitlab-project-housekeeping command line option, 582
gitlab-project-issue-award-emoji-delete command line option, 659
gitlab-project-issue-award-emoji-get command line option, 660
gitlab-project-issue-discussion-get

command line option, 661
gitlab-project-issue-discussion-note-delete command line option, 663
gitlab-project-issue-discussion-note-get command line option, 663
gitlab-project-issue-discussion-note-update command line option, 664
gitlab-project-issue-note-award-emoji-delete command line option, 668
gitlab-project-issue-note-award-emoji-get command line option, 668
gitlab-project-issue-note-delete command line option, 666
gitlab-project-issue-note-get command line option, 666
gitlab-project-issue-note-update command line option, 667
gitlab-project-issue-resource-iteration-event-get command line option, 669
gitlab-project-issue-resource-label-event-get command line option, 670
gitlab-project-issue-resource-milestone-event-get command line option, 671
gitlab-project-issue-resource-state-event-get command line option, 672
gitlab-project-issue-resource-weight-event-get command line option, 673
gitlab-project-job-artifact command line option, 675
gitlab-project-job-artifacts command line option, 675
gitlab-project-job-cancel command line option, 676
gitlab-project-job-delete-artifacts command line option, 676
gitlab-project-job-erase command line option, 676
gitlab-project-job-get command line option, 676
gitlab-project-job-keep-artifacts command line option, 677
gitlab-project-job-play command line option, 677
gitlab-project-job-retry command line option, 678
gitlab-project-job-trace command line option, 678
gitlab-project-key-delete command line option, 679
gitlab-project-key-get command line option, 680
gitlab-project-key-update command line option, 680
gitlab-project-languages command line option, 582
gitlab-project-member-all-get command line option, 686
gitlab-project-member-delete command line option, 685
gitlab-project-member-get command line option, 685
gitlab-project-member-update command line option, 685
gitlab-project-merge-request-approval-rule-delete command line option, 698
gitlab-project-merge-request-approval-rule-get command line option, 698
gitlab-project-merge-request-approval-rule-update command line option, 699
gitlab-project-merge-request-approval-state-get command line option, 700
gitlab-project-merge-request-award-emoji-delete command line option, 701
gitlab-project-merge-request-award-emoji-get command line option, 701
gitlab-project-merge-request-diff-get command line option, 702
gitlab-project-merge-request-discussion-get command line option, 703
gitlab-project-merge-request-discussion-note-delete command line option, 705
gitlab-project-merge-request-discussion-note-get command line option, 705
gitlab-project-merge-request-discussion-note-update command line option, 706
gitlab-project-merge-request-discussion-update command line option, 704
gitlab-project-merge-request-draft-note-delete command line option, 707
gitlab-project-merge-request-draft-note-get command line option, 708
gitlab-project-merge-request-draft-note-update command line option, 709
gitlab-project-merge-request-note-award-emoji-delete command line option, 711
gitlab-project-merge-request-note-award-emoji-get command line option, 712
gitlab-project-merge-request-note-delete command line option, 709
gitlab-project-merge-request-note-get command line option, 710
gitlab-project-merge-request-note-update command line option, 710
gitlab-project-merge-request-resource-label-event-get command line option, 714
gitlab-project-merge-request-resource-milestone-event- command line option, 714
gitlab-project-merge-request-resource-state-event-get

command line option, 715
gitlab-project-milestone-delete command line option, 719
gitlab-project-milestone-get command line option, 720
gitlab-project-milestone-issues command line option, 720
gitlab-project-milestone-merge-requests command line option, 721
gitlab-project-milestone-update command line option, 721
gitlab-project-mirror-pull command line option, 583
gitlab-project-mirror-pull-details command line option, 584
gitlab-project-note-get command line option, 721
gitlab-project-package-delete command line option, 724
gitlab-project-package-file-delete command line option, 725
gitlab-project-package-get command line option, 724
gitlab-project-package-protection-rule-delete command line option, 727
gitlab-project-package-protection-rule-update command line option, 728
gitlab-project-pipeline-cancel command line option, 731
gitlab-project-pipeline-delete command line option, 731
gitlab-project-pipeline-get command line option, 732
gitlab-project-pipeline-retry command line option, 733
gitlab-project-pipeline-schedule-delete command line option, 735
gitlab-project-pipeline-schedule-get command line option, 735
gitlab-project-pipeline-schedule-play command line option, 736
gitlab-project-pipeline-schedule-take-ownership command line option, 736
gitlab-project-pipeline-schedule-update command line option, 736
gitlab-project-registry-protection-rule-update command line option, 750
gitlab-project-registry-repository-delete command line option, 750
gitlab-project-registry-repository-protection-rule-delete command line option, 751
gitlab-project-registry-repository-protection-rule-update command line option, 752
gitlab-project-release-delete command line option, 757
gitlab-project-release-link-get command line option, 757
gitlab-project-release-link-update command line option, 758
gitlab-project-remote-mirror-delete command line option, 759
gitlab-project-remote-mirror-sync command line option, 760
gitlab-project-remote-mirror-update command line option, 760
gitlab-project-repository-archive command line option, 584
gitlab-project-repository-blob command line option, 584
gitlab-project-repository-compare command line option, 584
gitlab-project-repository-contributors command line option, 584
gitlab-project-repository-merge-base command line option, 585
gitlab-project-repository-raw-blob command line option, 585
gitlab-project-repository-tree command line option, 585
gitlab-project-restore command line option, 585
gitlab-project-runner-delete command line option, 763
gitlab-project-search command line option, 585
gitlab-project-secure-file-delete command line option, 764
gitlab-project-secure-file-download command line option, 764
gitlab-project-secure-file-get command line option, 765
gitlab-project-service-account-access-token-delete command line option, 769
gitlab-project-service-account-access-token-rotate command line option, 771
gitlab-project-service-account-delete command line option, 767
gitlab-project-service-account-update command line option, 768
gitlab-project-share command line option, 586
gitlab-project-snapshot command line option, 586
gitlab-project-snippet-award-emoji-delete command line option, 774
gitlab-project-snippet-award-emoji-get command line option, 774
gitlab-project-snippet-content command

line option, 771
gitlab-project-snippet-delete command line option, 772
gitlab-project-snippet-discussion-get command line option, 776
gitlab-project-snippet-discussion-note-delete command line option, 777
gitlab-project-snippet-discussion-note-get command line option, 778
gitlab-project-snippet-discussion-note-update command line option, 778
gitlab-project-snippet-get command line option, 772
gitlab-project-snippet-note-award-emoji-delete command line option, 781
gitlab-project-snippet-note-award-emoji-get command line option, 781
gitlab-project-snippet-note-delete command line option, 779
gitlab-project-snippet-note-get command line option, 779
gitlab-project-snippet-note-update command line option, 780
gitlab-project-snippet-update command line option, 773
gitlab-project-snippet-user-agent-detail command line option, 773
gitlab-project-star command line option, 586
gitlab-project-storage-get command line option, 782
gitlab-project-transfer command line option, 586
gitlab-project-trigger-delete command line option, 784
gitlab-project-trigger-get command line option, 785
gitlab-project-trigger-pipeline command line option, 587
gitlab-project-trigger-update command line option, 785
gitlab-project-unarchive command line option, 587
gitlab-project-unshare command line option, 587
gitlab-project-unstar command line option, 587
gitlab-project-update command line option, 589
gitlab-project-update-submodule command line option, 592
gitlab-project-upload command line option, 593
gitlab-registry-repository-get command line option, 791
gitlab-runner-all command line option, 791
gitlab-runner-delete command line option, 792
gitlab-runner-get command line option, 792
gitlab-runner-update command line option, 793
gitlab-runner-verify command line option, 793
gitlab-service-account-update command line option, 796
gitlab-snippet-content command line option, 797
gitlab-snippet-delete command line option, 798
gitlab-snippet-get command line option, 798
gitlab-snippet-list-all command line option, 799
gitlab-snippet-list-public command line option, 799
gitlab-snippet-update command line option, 799
gitlab-snippet-user-agent-detail command line option, 799
gitlab-todo-delete command line option, 801
gitlab-todo-mark-all-as-done command line option, 802
gitlab-todo-mark-as-done command line option, 802
gitlab-topic-delete command line option, 803
gitlab-topic-get command line option, 803
gitlab-topic-merge command line option, 803
gitlab-topic-update command line option, 804
gitlab-user-activate command line option, 804
gitlab-user-approve command line option, 804
gitlab-user-ban command line option, 804
gitlab-user-block command line option, 805
gitlab-user-deactivate command line option, 806
gitlab-user-delete command line option, 806
gitlab-user-email-delete command line option, 813
gitlab-user-email-get command line

option, 813
gitlab-user-follow command line option, 806
gitlab-user-get command line option, 807
gitlab-user-gpg-key-delete command line option, 815
gitlab-user-gpg-key-get command line option, 815
gitlab-user-impersonation-token-delete command line option, 816
gitlab-user-impersonation-token-get command line option, 816
gitlab-user-key-delete command line option, 818
gitlab-user-key-get command line option, 818
gitlab-user-reject command line option, 808
gitlab-user-unban command line option, 808
gitlab-user-unblock command line option, 809
gitlab-user-unfollow command line option, 809
gitlab-user-update command line option, 809
--id-after
gitlab-project-list command line option, 582
gitlab-user-project-list command line option, 822
--id-before
gitlab-project-list command line option, 582
gitlab-user-project-list command line option, 822
--iid
gitlab-group-epic-delete command line option, 515
gitlab-group-epic-get command line option, 515
gitlab-group-epic-update command line option, 516
gitlab-project-feature-flag-user-list-delete command line option, 638
gitlab-project-feature-flag-user-list-get command line option, 638
gitlab-project-feature-flag-user-list-update command line option, 639
gitlab-project-issue-add-spent-time command line option, 652
gitlab-project-issue-closed-by command line option, 652
gitlab-project-issue-delete command line option, 653
gitlab-project-issue-get command line option, 653
gitlab-project-issue-move command line option, 655
gitlab-project-issue-participants command line option, 655
gitlab-project-issue-related-merge-requests command line option, 655
gitlab-project-issue-reorder command line option, 656
gitlab-project-issue-reset-spent-time command line option, 656
gitlab-project-issue-reset-time-estimate command line option, 656
gitlab-project-issue-subscribe command line option, 657
gitlab-project-issue-time-estimate command line option, 657
gitlab-project-issue-time-stats command line option, 657
gitlab-project-issue-todo command line option, 657
gitlab-project-issue-unsubscribe command line option, 658
gitlab-project-issue-update command line option, 658
gitlab-project-issue-user-agent-detail command line option, 659
gitlab-project-merge-request-add-spent-time command line option, 687
gitlab-project-merge-request-approve command line option, 687
gitlab-project-merge-request-cancel-merge-when-pipelin command line option, 688
gitlab-project-merge-request-changes command line option, 688
gitlab-project-merge-request-closes-issues command line option, 688
gitlab-project-merge-request-commits command line option, 688
gitlab-project-merge-request-delete command line option, 690
gitlab-project-merge-request-get command line option, 690
gitlab-project-merge-request-merge command line option, 692
gitlab-project-merge-request-merge-ref command line option, 692
gitlab-project-merge-request-participants command line option, 693
gitlab-project-merge-request-rebase command line option, 693
gitlab-project-merge-request-related-issues

command line option, 693
gitlab-project-merge-request-reset-approvals command line option, 693
gitlab-project-merge-request-reset-spent-time command line option, 694
gitlab-project-merge-request-reset-time-estimate command line option, 694
gitlab-project-merge-request-subscribe command line option, 694
gitlab-project-merge-request-time-estimate command line option, 694
gitlab-project-merge-request-time-stats command line option, 695
gitlab-project-merge-request-todo command line option, 695
gitlab-project-merge-request-unapprove command line option, 695
gitlab-project-merge-request-unsubscribe command line option, 695
gitlab-project-merge-request-update command line option, 696
--iids
gitlab-group-issue-list command line option, 532
gitlab-group-milestone-list command line option, 546
gitlab-issue-list command line option, 565
gitlab-project-issue-list command line option, 654
gitlab-project-merge-request-list command line option, 691
gitlab-project-milestone-list command line option, 720
--image-url
gitlab-group-badge-create command line option, 502
gitlab-group-badge-render command line option, 503
gitlab-group-badge-update command line option, 503
gitlab-project-badge-create command line option, 601
gitlab-project-badge-render command line option, 603
gitlab-project-badge-update command line option, 603
--import-sources
gitlab-application-settings-update command line option, 470
--import-url
gitlab-project-create command line option, 579
gitlab-project-update command line option, 590
gitlab-user-project-create command line option, 821
--in
gitlab-group-iteration-list command line option, 534
gitlab-merge-request-list command line option, 572
gitlab-project-deployment-merge-request-list command line option, 628
--in-reply-to-discussion-id
gitlab-project-merge-request-draft-note-create command line option, 707
--include-ancestors
gitlab-group-iteration-list command line option, 534
--include-descendants
gitlab-group-iteration-list command line option, 534
--include-diverged-commits-count
gitlab-project-merge-request-get command line option, 690
--include-html-description
gitlab-project-release-list command line option, 756
gitlab-project-resource-group-list command line option, 761
--include-jobs
gitlab-ci-lint-create command line option, 477
gitlab-ci-lint-validate command line option, 478
gitlab-project-ci-lint-create command line option, 608
gitlab-project-ci-lint-get command line option, 609
gitlab-project-ci-lint-validate command line option, 609
--include-merged-yaml
gitlab-ci-lint-create command line option, 477
gitlab-ci-lint-validate command line option, 477
--include-rebase-in-progress
gitlab-project-merge-request-get command line option, 690
--include-retried
gitlab-project-pipeline-job-list command line option, 734
--include-subgroups
gitlab-group-project-list command line option, 550
--include-versionless
gitlab-group-package-list command line

option, 548
gitlab-project-package-list command line option, 724
--info
gitlab-runner-create command line option, 791
--infrastructure-access-level
gitlab-project-create command line option, 580
gitlab-project-update command line option, 591
--initialize-with-readme
gitlab-project-create command line option, 579
--instance-statistics-visibility-private
gitlab-application-settings-update command line option, 471
--invite-source
gitlab-group-invitation-create command line option, 530
gitlab-project-invitation-create command line option, 650
--issue-branch-template
gitlab-project-update command line option, 592
--issue-id
gitlab-group-epic-issue-create command line option, 520
--issue-iid
gitlab-project-issue-award-emoji-create command line option, 659
gitlab-project-issue-award-emoji-delete command line option, 659
gitlab-project-issue-award-emoji-get command line option, 660
gitlab-project-issue-award-emoji-list command line option, 660
gitlab-project-issue-discussion-create command line option, 661
gitlab-project-issue-discussion-get command line option, 661
gitlab-project-issue-discussion-list command line option, 661
gitlab-project-issue-discussion-note-create-issue-link-id command line option, 662
gitlab-project-issue-discussion-note-delete command line option, 662
gitlab-project-issue-discussion-note-get command line option, 663
gitlab-project-issue-discussion-note-list command line option, 663
gitlab-project-issue-discussion-note-update-issues-enabled command line option, 664
gitlab-project-issue-link-create command line option, 664
gitlab-project-issue-link-delete command line option, 665
gitlab-project-issue-link-list command line option, 665
gitlab-project-issue-note-award-emoji-create command line option, 667
gitlab-project-issue-note-award-emoji-delete command line option, 668
gitlab-project-issue-note-award-emoji-get command line option, 668
gitlab-project-issue-note-award-emoji-list command line option, 669
gitlab-project-issue-note-create command line option, 666
gitlab-project-issue-note-delete command line option, 666
gitlab-project-issue-note-get command line option, 666
gitlab-project-issue-note-list command line option, 666
gitlab-project-issue-note-update command line option, 667
gitlab-project-issue-resource-iteration-event-get command line option, 669
gitlab-project-issue-resource-iteration-event-list command line option, 670
gitlab-project-issue-resource-label-event-get command line option, 670
gitlab-project-issue-resource-label-event-list command line option, 671
gitlab-project-issue-resource-milestone-event-get command line option, 671
gitlab-project-issue-resource-milestone-event-list command line option, 671
gitlab-project-issue-resource-state-event-get command line option, 672
gitlab-project-issue-resource-state-event-list command line option, 672
gitlab-project-issue-resource-weight-event-get command line option, 673
gitlab-project-issue-resource-weight-event-list command line option, 673
gitlab-project-issue-link-delete command line option, 665
--issues-access-level
gitlab-project-create command line option, 579
gitlab-project-update command line option, 590
gitlab-project-create command line option, 579

    gitlab-project-update command line option, 590
    gitlab-user-project-create command line option, 820
--issues-events
    gitlab-group-hook-create command line option, 527
    gitlab-group-hook-update command line option, 529
    gitlab-project-hook-create command line option, 646
    gitlab-project-hook-update command line option, 647
--issues-template
    gitlab-project-update command line option, 590
--iteration-id
    gitlab-group-board-list-create command line option, 507
    gitlab-group-issue-list command line option, 532
    gitlab-issue-list command line option, 565
    gitlab-project-board-list-create command line option, 605
    gitlab-project-issue-list command line option, 654
--job
    gitlab-project-artifact-download command line option, 600
    gitlab-project-artifact-raw command line option, 600
--job-events
    gitlab-group-hook-create command line option, 527
    gitlab-group-hook-update command line option, 529
    gitlab-project-hook-create command line option, 646
    gitlab-project-hook-update command line option, 647
--job-token
    gitlab command line option, 465
    gitlab-project-artifact-download command line option, 600
--jobs-enabled
    gitlab-project-create command line option, 579
    gitlab-project-update command line option, 590
--keep-latest-artifact
    gitlab-project-update command line option, 590
--keep-n
    gitlab-project-registry-tag-delete-in-bulk command line option, 753
--key
    gitlab-current-user-gpg-key-create command line option, 479
    gitlab-current-user-key-create command line option, 480
    gitlab-deploy-key-create command line option, 483
    gitlab-group-custom-attribute-delete command line option, 511
    gitlab-group-custom-attribute-get command line option, 511
    gitlab-group-variable-create command line option, 560
    gitlab-group-variable-delete command line option, 560
    gitlab-group-variable-get command line option, 560
    gitlab-group-variable-update command line option, 561
    gitlab-license-get command line option, 567
    gitlab-project-custom-attribute-delete command line option, 622
    gitlab-project-custom-attribute-get command line option, 622
    gitlab-project-key-create command line option, 679
    gitlab-project-license-template-get command line option, 683
    gitlab-project-pages-domain-create command line option, 729
    gitlab-project-pages-domain-update command line option, 730
    gitlab-project-pipeline-schedule-variable-create command line option, 738
    gitlab-project-pipeline-schedule-variable-delete command line option, 738
    gitlab-project-pipeline-schedule-variable-update command line option, 739
    gitlab-project-resource-group-get command line option, 761
    gitlab-project-resource-group-update command line option, 762
    gitlab-project-variable-create command line option, 786
    gitlab-project-variable-delete command line option, 787
    gitlab-project-variable-get command line option, 787
    gitlab-project-variable-update command line option, 788
    gitlab-user-custom-attribute-delete

command line option, 811
gitlab-user-custom-attribute-get command line option, 812
gitlab-user-gpg-key-create command line option, 815
gitlab-user-key-create command line option, 817
gitlab-variable-create command line option, 823
gitlab-variable-delete command line option, 823
gitlab-variable-get command line option, 823
gitlab-variable-update command line option, 824
--key-id
gitlab-project-key-enable command line option, 680
--label-id
gitlab-group-board-list-create command line option, 507
gitlab-project-board-list-create command line option, 605
--labels
gitlab-group-epic-create command line option, 514
gitlab-group-epic-list command line option, 515
gitlab-group-epic-update command line option, 516
gitlab-group-issue-list command line option, 532
gitlab-group-merge-request-list command line option, 543
gitlab-issue-list command line option, 565
gitlab-merge-request-list command line option, 571
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-issue-create command line option, 653
gitlab-project-issue-list command line option, 654
gitlab-project-issue-update command line option, 658
gitlab-project-merge-request-create command line option, 689
gitlab-project-merge-request-list command line option, 691
gitlab-project-merge-request-update command line option, 696
--last-activity-after
gitlab-project-list command line option, 582
--last-activity-before
gitlab-project-list command line option, 582
--last-commit-id
gitlab-project-file-update command line option, 642
--last-used-after
gitlab-group-service-account-access-token-list command line option, 557
gitlab-project-service-account-access-token-list command line option, 770
--last-used-before
gitlab-group-service-account-access-token-list command line option, 558
gitlab-project-service-account-access-token-list command line option, 770
--level
gitlab-group-notification-settings-update command line option, 547
gitlab-notification-settings-update command line option, 574
gitlab-project-notification-settings-update command line option, 723
--lfs-enabled
gitlab-group-create command line option, 491
gitlab-group-update command line option, 496
gitlab-project-create command line option, 579
gitlab-project-update command line option, 590
--line
gitlab-project-commit-comment-create command line option, 617
--line-type
gitlab-project-commit-comment-create command line option, 617
--link-type
gitlab-project-release-link-create command line option, 757
gitlab-project-release-link-update command line option, 758
--link-url
gitlab-group-badge-create command line option, 502
gitlab-group-badge-render command line option, 503
gitlab-group-badge-update command line option, 503
gitlab-project-badge-create command line option, 601
gitlab-project-badge-render command

line option, 603
gitlab-project-badge-update command line option, 603
--linkedin
gitlab-user-create command line option, 805
gitlab-user-update command line option, 809
--local-markdown-version
gitlab-application-settings-update command line option, 471
--location
gitlab-user-create command line option, 806
gitlab-user-update command line option, 810
--locked
gitlab-current-user-runner-create command line option, 482
gitlab-runner-create command line option, 791
gitlab-runner-update command line option, 793
--logo
gitlab-application-appearance-update command line option, 467
--maintenance-note
gitlab-current-user-runner-create command line option, 482
--manage-deploy-tokens
gitlab-group-member-role-create command line option, 541
gitlab-member-role-create command line option, 569
--manage-group-access-tokens
gitlab-group-member-role-create command line option, 541
gitlab-member-role-create command line option, 569
--manage-merge-request-settings
gitlab-group-member-role-create command line option, 541
gitlab-member-role-create command line option, 569
--manage-project-access-tokens
gitlab-group-member-role-create command line option, 542
gitlab-member-role-create command line option, 569
--manage-security-policy-link
gitlab-group-member-role-create command line option, 542
gitlab-member-role-create command line option, 569
--managed
gitlab-group-cluster-create command line option, 509
gitlab-project-cluster-create command line option, 610
--management-project-id
gitlab-group-cluster-update command line option, 510
gitlab-project-cluster-update command line option, 611
--masked
gitlab-group-variable-create command line option, 560
gitlab-group-variable-update command line option, 561
gitlab-project-variable-create command line option, 787
gitlab-project-variable-update command line option, 788
gitlab-variable-create command line option, 823
gitlab-variable-update command line option, 824
--masked-and-hidden
gitlab-group-variable-create command line option, 560
gitlab-project-variable-create command line option, 787
--max-attachment-size
gitlab-application-settings-update command line option, 470
--max-file-size
gitlab-group-push-rules-create command line option, 551
gitlab-group-push-rules-update command line option, 552
gitlab-project-push-rules-create command line option, 747
gitlab-project-push-rules-update command line option, 748
--maximum-timeout
gitlab-current-user-runner-create command line option, 482
gitlab-runner-create command line option, 792
gitlab-runner-update command line option, 793
--member-check
gitlab-group-push-rules-create command line option, 551
gitlab-group-push-rules-update command line option, 552
gitlab-project-push-rules-create command line option, 747

gitlab-project-push-rules-update command line option, 748
--membership
gitlab-project-fork-list command line option, 643
gitlab-project-list command line option, 582
gitlab-starred-project-list command line option, 800
gitlab-user-project-list command line option, 821
--membership-lock
gitlab-group-create command line option, 491
gitlab-group-update command line option, 495
--mentions-disabled
gitlab-group-create command line option, 491
gitlab-group-update command line option, 496
--merge-access-level
gitlab-project-protected-branch-create command line option, 741
--merge-commit-message
gitlab-project-merge-request-merge command line option, 692
--merge-commit-template
gitlab-project-update command line option, 590
--merge-merge-request
gitlab-group-notification-settings-update command line option, 548
gitlab-notification-settings-update command line option, 574
gitlab-project-notification-settings-update command line option, 723
--merge-method
gitlab-project-create command line option, 579
gitlab-project-update command line option, 591
--merge-pipelines-enabled
gitlab-project-create command line option, 579
gitlab-project-update command line option, 591
--merge-request-iid
gitlab-project-merge-request-approval-rule-create command line option, 698
gitlab-project-merge-request-approval-rule-delete command line option, 698
gitlab-project-merge-request-approval-rule-get command line option, 698
gitlab-project-merge-request-approval-rule-list command line option, 699
gitlab-project-merge-request-approval-rule-update command line option, 699
gitlab-project-merge-request-status-check-list command line option, 717
--merge-request-to-resolve-discussions-of
gitlab-project-issue-create command line option, 653
--merge-requests-access-level
gitlab-project-create command line option, 579
gitlab-project-update command line option, 591
--merge-requests-author-approval
gitlab-project-approval-update command line option, 597
--merge-requests-disable-committers-approval
gitlab-project-approval-update command line option, 597
--merge-requests-enabled
gitlab-project-create command line option, 579
gitlab-project-update command line option, 591
gitlab-user-project-create command line option, 820
--merge-requests-events
gitlab-group-hook-create command line option, 527
gitlab-group-hook-update command line option, 529
gitlab-project-hook-create command line option, 646
gitlab-project-hook-update command line option, 647
--merge-requests-template
gitlab-project-update command line option, 591
--merge-trains-enabled
gitlab-project-update command line option, 591
--merge-when-pipeline-succeeds
gitlab-project-merge-request-merge command line option, 692
--message
gitlab-broadcast-message-create command line option, 473
gitlab-broadcast-message-update command line option, 474
gitlab-current-user-status-update command line option, 483
gitlab-project-tag-create command line option, 783

--message-background-color
    gitlab-application-appearance-update command line option, 467
--message-font-color
    gitlab-application-appearance-update command line option, 467
--milestone
    gitlab-group-issue-list command line option, 532
    gitlab-group-merge-request-list command line option, 543
    gitlab-issue-list command line option, 565
    gitlab-merge-request-list command line option, 571
    gitlab-project-deployment-merge-request-list command line option, 628
    gitlab-project-issue-list command line option, 654
    gitlab-project-merge-request-list command line option, 691
--milestone-id
    gitlab-group-board-list-create command line option, 507
    gitlab-project-board-list-create command line option, 605
    gitlab-project-issue-create command line option, 653
    gitlab-project-issue-update command line option, 658
    gitlab-project-merge-request-create command line option, 689
    gitlab-project-merge-request-update command line option, 696
--milestones
    gitlab-project-release-update command line option, 756
--min-access-level
    gitlab-group-descendant-group-list command line option, 514
    gitlab-group-list command line option, 493
    gitlab-group-project-list command line option, 550
    gitlab-group-subgroup-list command line option, 559
    gitlab-project-list command line option, 583
    gitlab-shared-project-list command line option, 797
    gitlab-starred-project-list command line option, 800
    gitlab-user-project-list command line option, 822
--minimum-access-level-for-delete
    gitlab-project-registry-protection-rule-create command line option, 749
    gitlab-project-registry-protection-rule-update command line option, 750
    gitlab-project-registry-repository-protection-rule-cre command line option, 751
    gitlab-project-registry-repository-protection-rule-upd command line option, 752
--minimum-access-level-for-push
    gitlab-project-package-protection-rule-create command line option, 727
    gitlab-project-package-protection-rule-update command line option, 728
    gitlab-project-registry-protection-rule-create command line option, 749
    gitlab-project-registry-protection-rule-update command line option, 750
    gitlab-project-registry-repository-protection-rule-cre command line option, 751
    gitlab-project-registry-repository-protection-rule-upd command line option, 752
--mirror
    gitlab-project-create command line option, 579
    gitlab-project-update command line option, 591
--mirror-overwrites-diverged-branches
    gitlab-project-update command line option, 591
--mirror-trigger-builds
    gitlab-project-create command line option, 579
    gitlab-project-update command line option, 591
--mirror-user-id
    gitlab-project-update command line option, 591
--monitor-access-level
    gitlab-project-create command line option, 580
    gitlab-project-update command line option, 591
--move-after-id
    gitlab-group-epic-issue-update command line option, 521
    gitlab-project-issue-reorder command line option, 656
--move-before-id
    gitlab-group-epic-issue-update command line option, 521
    gitlab-project-issue-reorder command line option, 656
--mr-default-target-self

gitlab-project-update command line option, 591
--mr-iid
gitlab-project-merge-request-approval-get command line option, 697
gitlab-project-merge-request-approval-state-get command line option, 700
gitlab-project-merge-request-approval-update command line option, 697
gitlab-project-merge-request-award-emoji-create command line option, 700
gitlab-project-merge-request-award-emoji-delete command line option, 701
gitlab-project-merge-request-award-emoji-get command line option, 701
gitlab-project-merge-request-award-emoji-list command line option, 701
gitlab-project-merge-request-diff-get command line option, 702
gitlab-project-merge-request-diff-list command line option, 702
gitlab-project-merge-request-discussion-create command line option, 703
gitlab-project-merge-request-discussion-get command line option, 703
gitlab-project-merge-request-discussion-list command line option, 704
gitlab-project-merge-request-discussion-note-create command line option, 705
gitlab-project-merge-request-discussion-note-delete command line option, 705
gitlab-project-merge-request-discussion-note-get command line option, 705
gitlab-project-merge-request-discussion-note-list command line option, 706
gitlab-project-merge-request-discussion-note-update command line option, 706
gitlab-project-merge-request-discussion-update command line option, 704
gitlab-project-merge-request-draft-note-create command line option, 707
gitlab-project-merge-request-draft-note-delete command line option, 707
gitlab-project-merge-request-draft-note-get command line option, 708
gitlab-project-merge-request-draft-note-list command line option, 708
gitlab-project-merge-request-draft-note-update command line option, 709
gitlab-project-merge-request-note-award-emoji-create command line option, 711
gitlab-project-merge-request-note-award-emoji-delete command line option, 711
gitlab-project-merge-request-note-award-emoji-get command line option, 712
gitlab-project-merge-request-note-award-emoji-list command line option, 712
gitlab-project-merge-request-note-create command line option, 709
gitlab-project-merge-request-note-delete command line option, 709
gitlab-project-merge-request-note-get command line option, 710
gitlab-project-merge-request-note-list command line option, 710
gitlab-project-merge-request-note-update command line option, 710
gitlab-project-merge-request-pipeline-create command line option, 713
gitlab-project-merge-request-pipeline-list command line option, 713
gitlab-project-merge-request-resource-label-event-get command line option, 713
gitlab-project-merge-request-resource-label-event-list command line option, 714
gitlab-project-merge-request-resource-milestone-event- command line option, 714
gitlab-project-merge-request-resource-milestone-event- command line option, 715
gitlab-project-merge-request-resource-state-event-get command line option, 715
gitlab-project-merge-request-resource-state-event-list command line option, 716
gitlab-project-merge-request-reviewer-detail-list command line option, 716
--myreaction-emoji
gitlab-group-issue-list command line option, 532
gitlab-group-merge-request-list command line option, 544
gitlab-issue-list command line option, 565
gitlab-merge-request-list command line option, 572
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-issue-list command line option, 654
gitlab-project-merge-request-list command line option, 691
--name
gitlab-application-create command line option, 466
gitlab-dockerfile-get command line option, 484
gitlab-feature-delete command line option, 486
gitlab-gitignore-get command line

option, 489
gitlab-gitlabciyml-get command line option, 490
gitlab-group-access-token-create command line option, 498
gitlab-group-approval-rule-create command line option, 499
gitlab-group-board-create command line option, 505
gitlab-group-cluster-create command line option, 509
gitlab-group-cluster-update command line option, 510
gitlab-group-create command line option, 491
gitlab-group-deploy-token-create command line option, 512
gitlab-group-epic-award-emoji-create command line option, 516
gitlab-group-epic-note-award-emoji-create command line option, 523
gitlab-group-label-create command line option, 534
gitlab-group-label-delete command line option, 535
gitlab-group-label-get command line option, 535
gitlab-group-label-subscribe command line option, 536
gitlab-group-label-unsubscribe command line option, 536
gitlab-group-label-update command line option, 536
gitlab-group-member-role-create command line option, 541
gitlab-group-saml-group-link-delete command line option, 553
gitlab-group-saml-group-link-get command line option, 554
gitlab-group-service-account-access-token-create command line option, 556
gitlab-group-service-account-create command line option, 555
gitlab-group-service-account-update command line option, 556
gitlab-group-update command line option, 495
gitlab-member-role-create command line option, 569
gitlab-project-access-token-create command line option, 594
gitlab-project-approval-rule-create command line option, 598
gitlab-project-board-create command line option, 603
gitlab-project-branch-delete command line option, 607
gitlab-project-branch-get command line option, 608
gitlab-project-cluster-agent-create command line option, 612
gitlab-project-cluster-create command line option, 610
gitlab-project-cluster-update command line option, 611
gitlab-project-commit-status-create command line option, 621
gitlab-project-create command line option, 578
gitlab-project-deploy-token-create command line option, 623
gitlab-project-dockerfile-template-get command line option, 629
gitlab-project-environment-create command line option, 630
gitlab-project-environment-list command line option, 631
gitlab-project-environment-update command line option, 632
gitlab-project-external-status-check-create command line option, 634
gitlab-project-external-status-check-update command line option, 635
gitlab-project-feature-flag-create command line option, 636
gitlab-project-feature-flag-delete command line option, 636
gitlab-project-feature-flag-get command line option, 636
gitlab-project-feature-flag-update command line option, 637
gitlab-project-feature-flag-user-list-create command line option, 638
gitlab-project-feature-flag-user-list-update command line option, 639
gitlab-project-gitignore-template-get command line option, 644
gitlab-project-gitlabciyml-template-get command line option, 644
gitlab-project-issue-award-emoji-create command line option, 659
gitlab-project-issue-note-award-emoji-create command line option, 668
gitlab-project-issue-template-get command line option, 674
gitlab-project-label-create command line option, 681
gitlab-project-label-delete command

line option, 681
gitlab-project-label-get command line option, 682
gitlab-project-label-subscribe command line option, 682
gitlab-project-label-unsubscribe command line option, 683
gitlab-project-label-update command line option, 683
gitlab-project-merge-request-approval-rule-create command line option, 698
gitlab-project-merge-request-approval-rule-update command line option, 699
gitlab-project-merge-request-award-emoji-create command line option, 700
gitlab-project-merge-request-note-award-emoji-create command line option, 711
gitlab-project-merge-request-template-get command line option, 717
gitlab-project-pipeline-list command line option, 732
gitlab-project-protected-branch-create command line option, 741
gitlab-project-protected-branch-delete command line option, 741
gitlab-project-protected-branch-get command line option, 741
gitlab-project-protected-branch-update command line option, 742
gitlab-project-protected-environment-create command line option, 743
gitlab-project-protected-environment-delete command line option, 743
gitlab-project-protected-environment-get command line option, 743
gitlab-project-protected-tag-create command line option, 744
gitlab-project-protected-tag-delete command line option, 744
gitlab-project-protected-tag-get command line option, 745
gitlab-project-registry-tag-delete command line option, 753
gitlab-project-registry-tag-delete-in-bulk command line option, 753
gitlab-project-registry-tag-get command line option, 754
gitlab-project-release-create command line option, 755
gitlab-project-release-link-create command line option, 757
gitlab-project-release-link-update command line option, 758
gitlab-project-release-update command line option, 756
gitlab-project-secure-file-create command line option, 764
gitlab-project-service-account-access-token-create command line option, 769
gitlab-project-service-account-create command line option, 767
gitlab-project-service-account-update command line option, 768
gitlab-project-snippet-award-emoji-create command line option, 774
gitlab-project-snippet-note-award-emoji-create command line option, 780
gitlab-project-tag-delete command line option, 783
gitlab-project-tag-get command line option, 783
gitlab-project-update command line option, 591
gitlab-service-account-create command line option, 795
gitlab-service-account-update command line option, 796
gitlab-topic-create command line option, 802
gitlab-topic-update command line option, 804
gitlab-user-create command line option, 805
gitlab-user-impersonation-token-create command line option, 816
gitlab-user-personal-access-token-create command line option, 819
gitlab-user-project-create command line option, 820
gitlab-user-update command line option, 809
--name-regex-delete
gitlab-project-registry-tag-delete-in-bulk command line option, 753
--name-regex-keep
gitlab-project-registry-tag-delete-in-bulk command line option, 753
--namespace
gitlab-namespace-exists command line option, 572
gitlab-project-fork-create command line option, 642
--namespace-id
gitlab-project-create command line option, 579
--new-issue
gitlab-group-notification-settings-update command line option, 547

gitlab-notification-settings-update command line option, 574
gitlab-project-notification-settings-update command line option, 723
--new-merge-request
gitlab-group-notification-settings-update command line option, 548
gitlab-notification-settings-update command line option, 574
gitlab-project-notification-settings-update command line option, 723
--new-name
gitlab-group-label-update command line option, 536
gitlab-project-feature-flag-update command line option, 637
gitlab-project-label-update command line option, 683
--new-note
gitlab-group-notification-settings-update command line option, 547
gitlab-notification-settings-update command line option, 574
gitlab-project-notification-settings-update command line option, 723
--new-project-guidelines
gitlab-application-appearance-update command line option, 467
--no-get-all
gitlab-application-list command line option, 466
gitlab-audit-event-list command line option, 472
gitlab-broadcast-message-list command line option, 474
gitlab-bulk-import-all-entity-list command line option, 476
gitlab-bulk-import-entity-list command line option, 477
gitlab-bulk-import-list command line option, 475
gitlab-current-user-email-list command line option, 479
gitlab-current-user-gpg-key-list command line option, 480
gitlab-current-user-key-list command line option, 481
gitlab-deploy-key-list command line option, 483
gitlab-deploy-token-list command line option, 484
gitlab-dockerfile-list command line option, 485
gitlab-event-list command line option, 485
gitlab-feature-list command line option, 486
gitlab-geo-node-list command line option, 488
gitlab-gitignore-list command line option, 489
gitlab-gitlabciyml-list command line option, 490
gitlab-group-access-request-list command line option, 497
gitlab-group-access-token-list command line option, 498
gitlab-group-approval-rule-list command line option, 500
gitlab-group-audit-event-list command line option, 501
gitlab-group-badge-list command line option, 502
gitlab-group-billable-member-list command line option, 504
gitlab-group-billable-member-membership-list command line option, 505
gitlab-group-board-list command line option, 506
gitlab-group-board-list-list command line option, 508
gitlab-group-cluster-list command line option, 510
gitlab-group-custom-attribute-list command line option, 511
gitlab-group-deploy-token-list command line option, 513
gitlab-group-descendant-group-list command line option, 514
gitlab-group-epic-award-emoji-list command line option, 517
gitlab-group-epic-issue-list command line option, 520
gitlab-group-epic-list command line option, 516
gitlab-group-epic-note-award-emoji-list command line option, 524
gitlab-group-epic-note-list command line option, 522
gitlab-group-epic-resource-label-event-list command line option, 525
gitlab-group-hook-list command line option, 528
gitlab-group-invitation-list command line option, 531
gitlab-group-issue-list command line option, 533
gitlab-group-iteration-list command

line option, 534
gitlab-group-label-list command line option, 535
gitlab-group-ldap-group-link-list command line option, 537
gitlab-group-list command line option, 493
gitlab-group-member-all-list command line option, 540
gitlab-group-member-list command line option, 539
gitlab-group-member-role-list command line option, 542
gitlab-group-merge-request-list command line option, 544
gitlab-group-milestone-list command line option, 546
gitlab-group-package-list command line option, 549
gitlab-group-project-list command line option, 550
gitlab-group-runner-list command line option, 553
gitlab-group-saml-group-link-list command line option, 554
gitlab-group-service-account-access-token-list command line option, 558
gitlab-group-service-account-list command line option, 555
gitlab-group-subgroup-list command line option, 559
gitlab-group-variable-list command line option, 561
gitlab-group-wiki-list command line option, 563
gitlab-hook-list command line option, 564
gitlab-issue-list command line option, 566
gitlab-ldap-group-list command line option, 567
gitlab-license-list command line option, 568
gitlab-member-role-list command line option, 570
gitlab-merge-request-list command line option, 572
gitlab-namespace-list command line option, 573
gitlab-pages-domain-list command line option, 575
gitlab-personal-access-token-list command line option, 576
gitlab-project-access-request-list command line option, 594
gitlab-project-access-token-list command line option, 595
gitlab-project-approval-rule-list command line option, 599
gitlab-project-audit-event-list command line option, 601
gitlab-project-badge-list command line option, 602
gitlab-project-board-list command line option, 604
gitlab-project-board-list-list command line option, 606
gitlab-project-branch-list command line option, 608
gitlab-project-cluster-agent-list command line option, 613
gitlab-project-cluster-list command line option, 611
gitlab-project-commit-comment-list command line option, 617
gitlab-project-commit-discussion-list command line option, 619
gitlab-project-commit-list command line option, 615
gitlab-project-commit-status-list command line option, 622
gitlab-project-custom-attribute-list command line option, 623
gitlab-project-deploy-token-list command line option, 624
gitlab-project-deployment-list command line option, 626
gitlab-project-deployment-merge-request-list command line option, 629
gitlab-project-dockerfile-template-list command line option, 629
gitlab-project-environment-list command line option, 631
gitlab-project-event-list command line option, 632
gitlab-project-external-status-check-list command line option, 635
gitlab-project-feature-flag-list command line option, 637
gitlab-project-feature-flag-user-list-list command line option, 639
gitlab-project-fork-list command line option, 643
gitlab-project-gitignore-template-list command line option, 644
gitlab-project-gitlabciyml-template-list command line option, 645
gitlab-project-hook-list command line option, 647

gitlab-project-integration-list command line option, 649
gitlab-project-invitation-list command line option, 651
gitlab-project-issue-award-emoji-list command line option, 660
gitlab-project-issue-discussion-list command line option, 662
gitlab-project-issue-discussion-note-list command line option, 663
gitlab-project-issue-link-list command line option, 665
gitlab-project-issue-list command line option, 655
gitlab-project-issue-note-award-emoji-list command line option, 669
gitlab-project-issue-note-list command line option, 667
gitlab-project-issue-resource-iteration-event-list command line option, 670
gitlab-project-issue-resource-label-event-list command line option, 671
gitlab-project-issue-resource-milestone-event-list command line option, 672
gitlab-project-issue-resource-state-event-list command line option, 673
gitlab-project-issue-resource-weight-event-list command line option, 674
gitlab-project-issue-template-list command line option, 674
gitlab-project-job-list command line option, 677
gitlab-project-key-list command line option, 680
gitlab-project-label-list command line option, 682
gitlab-project-license-template-list command line option, 684
gitlab-project-list command line option, 583
gitlab-project-member-all-list command line option, 686
gitlab-project-member-list command line option, 685
gitlab-project-merge-request-approval-rule-list command line option, 699
gitlab-project-merge-request-award-emoji-list command line option, 702
gitlab-project-merge-request-diff-list command line option, 702
gitlab-project-merge-request-discussion-list command line option, 704
gitlab-project-merge-request-discussion-note-list command line option, 706
gitlab-project-merge-request-draft-note-list command line option, 708
gitlab-project-merge-request-list command line option, 692
gitlab-project-merge-request-note-award-emoji-list command line option, 712
gitlab-project-merge-request-note-list command line option, 710
gitlab-project-merge-request-pipeline-list command line option, 713
gitlab-project-merge-request-resource-label-event-list command line option, 714
gitlab-project-merge-request-resource-milestone-event- command line option, 715
gitlab-project-merge-request-resource-state-event-list command line option, 716
gitlab-project-merge-request-reviewer-detail-list command line option, 716
gitlab-project-merge-request-status-check-list command line option, 717
gitlab-project-merge-request-template-list command line option, 718
gitlab-project-merge-train-list command line option, 718
gitlab-project-milestone-list command line option, 720
gitlab-project-note-list command line option, 722
gitlab-project-package-file-list command line option, 725
gitlab-project-package-list command line option, 725
gitlab-project-package-pipeline-list command line option, 726
gitlab-project-package-protection-rule-list command line option, 727
gitlab-project-pages-domain-list command line option, 730
gitlab-project-pipeline-bridge-list command line option, 733
gitlab-project-pipeline-job-list command line option, 734
gitlab-project-pipeline-list command line option, 732
gitlab-project-pipeline-schedule-list command line option, 736
gitlab-project-pipeline-schedule-pipeline-list command line option, 737
gitlab-project-pipeline-variable-list command line option, 740
gitlab-project-protected-branch-list command line option, 742
gitlab-project-protected-environment-list command line option, 744

gitlab-project-protected-tag-list command line option, 745
gitlab-project-registry-protection-rule-list command line option, 749
gitlab-project-registry-repository-list command line option, 751
gitlab-project-registry-repository-protection-rule-list command line option, 752
gitlab-project-registry-tag-list command line option, 754
gitlab-project-release-link-list command line option, 758
gitlab-project-release-list command line option, 756
gitlab-project-remote-mirror-list command line option, 760
gitlab-project-resource-group-list command line option, 761
gitlab-project-resource-group-upcoming-job-list command line option, 762
gitlab-project-runner-list command line option, 763
gitlab-project-secure-file-list command line option, 765
gitlab-project-service-account-access-token-list command line option, 770
gitlab-project-service-account-list command line option, 768
gitlab-project-service-list command line option, 766
gitlab-project-snippet-award-emoji-list command line option, 775
gitlab-project-snippet-discussion-list command line option, 776
gitlab-project-snippet-list command line option, 773
gitlab-project-snippet-note-award-emoji-list command line option, 782
gitlab-project-snippet-note-list command line option, 779
gitlab-project-tag-list command line option, 784
gitlab-project-trigger-list command line option, 785
gitlab-project-user-list command line option, 786
gitlab-project-variable-list command line option, 788
gitlab-project-wiki-list command line option, 790
gitlab-runner-all-list command line option, 794
gitlab-runner-job-list command line option, 794
gitlab-runner-list command line option, 792
gitlab-service-account-list command line option, 795
gitlab-shared-project-list command line option, 797
gitlab-snippet-list command line option, 798
gitlab-starred-project-list command line option, 801
gitlab-todo-list command line option, 802
gitlab-topic-list command line option, 803
gitlab-user-activities-list command line option, 810
gitlab-user-contributed-project-list command line option, 811
gitlab-user-custom-attribute-list command line option, 812
gitlab-user-email-list command line option, 813
gitlab-user-event-list command line option, 814
gitlab-user-gpg-key-list command line option, 815
gitlab-user-impersonation-token-list command line option, 817
gitlab-user-key-list command line option, 818
gitlab-user-list command line option, 808
gitlab-user-membership-list command line option, 819
gitlab-user-project-list command line option, 822
gitlab-variable-list command line option, 823
--no-mask-credentials
gitlab command line option, 465
--no-ssl-verify
gitlab command line option, 465
--not
gitlab-merge-request-list command line option, 572
gitlab-project-deployment-merge-request-list command line option, 628
--note
gitlab-project-commit-comment-create command line option, 617
gitlab-project-merge-request-draft-note-create command line option, 707
--note-events
gitlab-group-hook-create command line option, 527
gitlab-group-hook-update command line

option, 529
gitlab-project-hook-create command line option, 646
gitlab-project-hook-update command line option, 647
--note-id
gitlab-group-epic-note-award-emoji-create command line option, 523
gitlab-group-epic-note-award-emoji-delete command line option, 523
gitlab-group-epic-note-award-emoji-get command line option, 524
gitlab-group-epic-note-award-emoji-list command line option, 524
gitlab-project-issue-note-award-emoji-create command line option, 667
gitlab-project-issue-note-award-emoji-delete command line option, 668
gitlab-project-issue-note-award-emoji-get command line option, 668
gitlab-project-issue-note-award-emoji-list command line option, 669
gitlab-project-merge-request-note-award-emoji-create command line option, 711
gitlab-project-merge-request-note-award-emoji-delete command line option, 711
gitlab-project-merge-request-note-award-emoji-get command line option, 712
gitlab-project-merge-request-note-award-emoji-list command line option, 712
gitlab-project-snippet-note-award-emoji-create command line option, 780
gitlab-project-snippet-note-award-emoji-delete command line option, 781
gitlab-project-snippet-note-award-emoji-get command line option, 781
gitlab-project-snippet-note-award-emoji-list command line option, 782
--notification-email
gitlab-group-notification-settings-update command line option, 547
gitlab-notification-settings-update command line option, 574
gitlab-project-notification-settings-update command line option, 723
--oauth-token
gitlab command line option, 465
--older-than
gitlab-project-registry-tag-delete-in-bulk command line option, 753
--only-allow-merge-if-all-discussions-are-resolved
gitlab-project-create command line option, 579
gitlab-project-update command line option, 591
--only-allow-merge-if-all-status-checks-passed
gitlab-project-create command line option, 578
gitlab-project-update command line option, 589
--only-allow-merge-if-build-succeeds
gitlab-user-project-create command line option, 821
--only-allow-merge-if-pipeline-succeeds
gitlab-project-create command line option, 579
gitlab-project-update command line option, 591
--only-mirror-protected-branches
gitlab-project-update command line option, 591
--only-protected-branches
gitlab-project-remote-mirror-create command line option, 759
gitlab-project-remote-mirror-update command line option, 760
--operations-access-level
gitlab-project-create command line option, 579
gitlab-project-update command line option, 591
--order
gitlab-project-commit-list command line option, 615
--order-by
gitlab command line option, 465
gitlab-group-descendant-group-list command line option, 513
gitlab-group-epic-list command line option, 515
gitlab-group-issue-list command line option, 532
gitlab-group-list command line option, 493
gitlab-group-merge-request-list command line option, 543
gitlab-group-package-list command line option, 548
gitlab-group-project-list command line option, 549
gitlab-group-service-account-list command line option, 555
gitlab-group-subgroup-list command line option, 559
gitlab-issue-list command line option, 565
gitlab-merge-request-list command line option, 571

gitlab-project-deployment-list command line option, 626
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-fork-list command line option, 643
gitlab-project-issue-list command line option, 654
gitlab-project-job-list command line option, 677
gitlab-project-list command line option, 583
gitlab-project-merge-request-list command line option, 691
gitlab-project-package-list command line option, 724
gitlab-project-pipeline-list command line option, 732
gitlab-project-release-list command line option, 755
gitlab-project-resource-group-list command line option, 761
gitlab-project-service-account-list command line option, 768
gitlab-project-tag-list command line option, 783
gitlab-runner-job-list command line option, 794
gitlab-service-account-list command line option, 795
gitlab-shared-project-list command line option, 796
gitlab-starred-project-list command line option, 800
gitlab-user-project-list command line option, 821
--organization
gitlab-user-create command line option, 806
gitlab-user-update command line option, 810
--outbound-local-requests-whitelist
gitlab-application-settings-update command line option, 470
--output
gitlab command line option, 464
--owned
gitlab-group-descendant-group-list command line option, 514
gitlab-group-list command line option, 493
gitlab-group-project-list command line option, 549
gitlab-group-subgroup-list command line option, 559
gitlab-project-fork-list command line option, 643
gitlab-project-list command line option, 583
gitlab-starred-project-list command line option, 800
gitlab-user-project-list command line option, 821
--package-id
gitlab-project-package-file-delete command line option, 725
gitlab-project-package-file-list command line option, 725
gitlab-project-package-pipeline-list command line option, 726
--package-name
gitlab-generic-package-download command line option, 487
gitlab-generic-package-upload command line option, 487
gitlab-group-package-list command line option, 548
gitlab-project-package-list command line option, 724
--package-name-pattern
gitlab-project-package-protection-rule-create command line option, 727
gitlab-project-package-protection-rule-update command line option, 728
--package-type
gitlab-group-package-list command line option, 548
gitlab-project-package-list command line option, 724
gitlab-project-package-protection-rule-create command line option, 727
gitlab-project-package-protection-rule-update command line option, 728
--package-version
gitlab-generic-package-download command line option, 487
gitlab-generic-package-upload command line option, 487
gitlab-group-package-list command line option, 548
gitlab-project-package-list command line option, 724
--packages-enabled
gitlab-project-create command line option, 579
gitlab-project-update command line option, 591
--page

gitlab-application-list command line option, 466
gitlab-audit-event-list command line option, 472
gitlab-broadcast-message-list command line option, 474
gitlab-bulk-import-all-entity-list command line option, 476
gitlab-bulk-import-entity-list command line option, 477
gitlab-bulk-import-list command line option, 475
gitlab-current-user-email-list command line option, 479
gitlab-current-user-gpg-key-list command line option, 480
gitlab-current-user-key-list command line option, 481
gitlab-deploy-key-list command line option, 483
gitlab-deploy-token-list command line option, 484
gitlab-dockerfile-list command line option, 485
gitlab-event-list command line option, 485
gitlab-feature-list command line option, 486
gitlab-geo-node-list command line option, 488
gitlab-gitignore-list command line option, 489
gitlab-gitlabciyml-list command line option, 490
gitlab-group-access-request-list command line option, 497
gitlab-group-access-token-list command line option, 498
gitlab-group-approval-rule-list command line option, 500
gitlab-group-audit-event-list command line option, 501
gitlab-group-badge-list command line option, 502
gitlab-group-billable-member-list command line option, 504
gitlab-group-billable-member-membership-list command line option, 505
gitlab-group-board-list command line option, 506
gitlab-group-board-list-list command line option, 508
gitlab-group-cluster-list command line option, 510
gitlab-group-custom-attribute-list command line option, 511
gitlab-group-deploy-token-list command line option, 513
gitlab-group-descendant-group-list command line option, 514
gitlab-group-epic-award-emoji-list command line option, 517
gitlab-group-epic-issue-list command line option, 520
gitlab-group-epic-list command line option, 515
gitlab-group-epic-note-award-emoji-list command line option, 524
gitlab-group-epic-note-list command line option, 522
gitlab-group-epic-resource-label-event-list command line option, 525
gitlab-group-hook-list command line option, 528
gitlab-group-invitation-list command line option, 531
gitlab-group-issue-list command line option, 533
gitlab-group-iteration-list command line option, 534
gitlab-group-label-list command line option, 535
gitlab-group-ldap-group-link-list command line option, 537
gitlab-group-list command line option, 493
gitlab-group-member-all-list command line option, 540
gitlab-group-member-list command line option, 539
gitlab-group-member-role-list command line option, 542
gitlab-group-merge-request-list command line option, 544
gitlab-group-milestone-list command line option, 546
gitlab-group-package-list command line option, 549
gitlab-group-project-list command line option, 550
gitlab-group-runner-list command line option, 553
gitlab-group-saml-group-link-list command line option, 554
gitlab-group-service-account-access-token-list command line option, 558
gitlab-group-service-account-list command line option, 555

gitlab-group-subgroup-list command line option, 559
gitlab-group-variable-list command line option, 561
gitlab-group-wiki-list command line option, 563
gitlab-hook-list command line option, 564
gitlab-issue-list command line option, 566
gitlab-ldap-group-list command line option, 567
gitlab-license-list command line option, 568
gitlab-member-role-list command line option, 570
gitlab-merge-request-list command line option, 572
gitlab-namespace-list command line option, 573
gitlab-pages-domain-list command line option, 575
gitlab-personal-access-token-list command line option, 576
gitlab-project-access-request-list command line option, 594
gitlab-project-access-token-list command line option, 595
gitlab-project-approval-rule-list command line option, 599
gitlab-project-audit-event-list command line option, 601
gitlab-project-badge-list command line option, 602
gitlab-project-board-list command line option, 604
gitlab-project-board-list-list command line option, 606
gitlab-project-branch-list command line option, 608
gitlab-project-cluster-agent-list command line option, 612
gitlab-project-cluster-list command line option, 611
gitlab-project-commit-comment-list command line option, 617
gitlab-project-commit-discussion-list command line option, 618
gitlab-project-commit-list command line option, 615
gitlab-project-commit-status-list command line option, 622
gitlab-project-custom-attribute-list command line option, 623
gitlab-project-deploy-token-list command line option, 624
gitlab-project-deployment-list command line option, 626
gitlab-project-deployment-merge-request-list command line option, 629
gitlab-project-dockerfile-template-list command line option, 629
gitlab-project-environment-list command line option, 631
gitlab-project-event-list command line option, 632
gitlab-project-external-status-check-list command line option, 634
gitlab-project-feature-flag-list command line option, 636
gitlab-project-feature-flag-user-list-list command line option, 638
gitlab-project-fork-list command line option, 643
gitlab-project-gitignore-template-list command line option, 644
gitlab-project-gitlabciyml-template-list command line option, 645
gitlab-project-hook-list command line option, 647
gitlab-project-integration-list command line option, 649
gitlab-project-invitation-list command line option, 651
gitlab-project-issue-award-emoji-list command line option, 660
gitlab-project-issue-discussion-list command line option, 661
gitlab-project-issue-discussion-note-list command line option, 663
gitlab-project-issue-link-list command line option, 665
gitlab-project-issue-list command line option, 654
gitlab-project-issue-note-award-emoji-list command line option, 669
gitlab-project-issue-note-list command line option, 667
gitlab-project-issue-resource-iteration-event-list command line option, 670
gitlab-project-issue-resource-label-event-list command line option, 671
gitlab-project-issue-resource-milestone-event-list command line option, 672
gitlab-project-issue-resource-state-event-list command line option, 672
gitlab-project-issue-resource-weight-event-list command line option, 673
gitlab-project-issue-template-list

command line option, 674
gitlab-project-job-list command line option, 677
gitlab-project-key-list command line option, 680
gitlab-project-label-list command line option, 682
gitlab-project-license-template-list command line option, 684
gitlab-project-list command line option, 583
gitlab-project-member-all-list command line option, 686
gitlab-project-member-list command line option, 685
gitlab-project-merge-request-approval-rule-list command line option, 699
gitlab-project-merge-request-award-emoji-list command line option, 701
gitlab-project-merge-request-diff-list command line option, 702
gitlab-project-merge-request-discussion-list command line option, 704
gitlab-project-merge-request-discussion-note-list command line option, 706
gitlab-project-merge-request-draft-note-list command line option, 708
gitlab-project-merge-request-list command line option, 691
gitlab-project-merge-request-note-award-emoji-list command line option, 712
gitlab-project-merge-request-note-list command line option, 710
gitlab-project-merge-request-pipeline-list command line option, 713
gitlab-project-merge-request-resource-label-event-list command line option, 714
gitlab-project-merge-request-resource-milestone-event-list command line option, 715
gitlab-project-merge-request-resource-state-event-list command line option, 716
gitlab-project-merge-request-reviewer-detail-list command line option, 716
gitlab-project-merge-request-status-check-list command line option, 717
gitlab-project-merge-request-template-list command line option, 718
gitlab-project-merge-train-list command line option, 718
gitlab-project-milestone-list command line option, 720
gitlab-project-note-list command line option, 722
gitlab-project-package-file-list command line option, 725
gitlab-project-package-list command line option, 724
gitlab-project-package-pipeline-list command line option, 726
gitlab-project-package-protection-rule-list command line option, 727
gitlab-project-pages-domain-list command line option, 730
gitlab-project-pipeline-bridge-list command line option, 733
gitlab-project-pipeline-job-list command line option, 734
gitlab-project-pipeline-list command line option, 732
gitlab-project-pipeline-schedule-list command line option, 735
gitlab-project-pipeline-schedule-pipeline-list command line option, 737
gitlab-project-pipeline-variable-list command line option, 740
gitlab-project-protected-branch-list command line option, 742
gitlab-project-protected-environment-list command line option, 743
gitlab-project-protected-tag-list command line option, 745
gitlab-project-registry-protection-rule-list command line option, 749
gitlab-project-registry-repository-list command line option, 751
gitlab-project-registry-repository-protection-rule-list command line option, 752
gitlab-project-registry-tag-list command line option, 754
gitlab-project-release-link-list command line option, 758
gitlab-project-release-list command line option, 756
gitlab-project-remote-mirror-list command line option, 759
gitlab-project-resource-group-list command line option, 761
gitlab-project-resource-group-upcoming-job-list command line option, 762
gitlab-project-runner-list command line option, 763
gitlab-project-secure-file-list command line option, 765
gitlab-project-service-account-access-token-list command line option, 770
gitlab-project-service-account-list command line option, 768
gitlab-project-service-list command

line option, 766
gitlab-project-snippet-award-emoji-list command line option, 775
gitlab-project-snippet-discussion-list command line option, 776
gitlab-project-snippet-list command line option, 772
gitlab-project-snippet-note-award-emoji-list command line option, 782
gitlab-project-snippet-note-list command line option, 779
gitlab-project-tag-list command line option, 783
gitlab-project-trigger-list command line option, 785
gitlab-project-user-list command line option, 786
gitlab-project-variable-list command line option, 787
gitlab-project-wiki-list command line option, 789
gitlab-runner-all-list command line option, 794
gitlab-runner-job-list command line option, 794
gitlab-runner-list command line option, 792
gitlab-service-account-list command line option, 795
gitlab-shared-project-list command line option, 797
gitlab-snippet-list command line option, 798
gitlab-starred-project-list command line option, 801
gitlab-todo-list command line option, 801
gitlab-topic-list command line option, 803
gitlab-user-activities-list command line option, 810
gitlab-user-contributed-project-list command line option, 811
gitlab-user-custom-attribute-list command line option, 812
gitlab-user-email-list command line option, 813
gitlab-user-event-list command line option, 814
gitlab-user-gpg-key-list command line option, 815
gitlab-user-impersonation-token-list command line option, 817
gitlab-user-key-list command line option, 818
gitlab-user-list command line option, 808
gitlab-user-membership-list command line option, 819
gitlab-user-project-list command line option, 822
gitlab-variable-list command line option, 823
--pages-access-level
gitlab-project-create command line option, 579
gitlab-project-update command line option, 591
--pages-https-only
gitlab-project-pages-update command line option, 729
--pages-unique-domain-enabled
gitlab-project-pages-update command line option, 729
--pagination
gitlab command line option, 465
--parent-id
gitlab-group-create command line option, 492
gitlab-namespace-exists command line option, 572
--password
gitlab-user-create command line option, 805
gitlab-user-update command line option, 809
--password-authentication-enabled-for-web
gitlab-application-settings-update command line option, 469
--path
gitlab-generic-package-upload command line option, 487
gitlab-group-create command line option, 491
gitlab-group-update command line option, 495
gitlab-project-commit-comment-create command line option, 617
gitlab-project-commit-list command line option, 615
gitlab-project-create command line option, 578
gitlab-project-repository-tree command line option, 585
gitlab-project-update command line option, 591
--paused
gitlab-current-user-runner-create command line option, 482
gitlab-runner-all-list command line

option, 794
gitlab-runner-list command line option, 792
--per-page
gitlab command line option, 465
gitlab-application-list command line option, 466
gitlab-audit-event-list command line option, 472
gitlab-broadcast-message-list command line option, 474
gitlab-bulk-import-all-entity-list command line option, 476
gitlab-bulk-import-entity-list command line option, 477
gitlab-bulk-import-list command line option, 475
gitlab-current-user-email-list command line option, 479
gitlab-current-user-gpg-key-list command line option, 480
gitlab-current-user-key-list command line option, 481
gitlab-deploy-key-list command line option, 483
gitlab-deploy-token-list command line option, 484
gitlab-dockerfile-list command line option, 485
gitlab-event-list command line option, 485
gitlab-feature-list command line option, 486
gitlab-geo-node-list command line option, 488
gitlab-gitignore-list command line option, 489
gitlab-gitlabciyml-list command line option, 490
gitlab-group-access-request-list command line option, 497
gitlab-group-access-token-list command line option, 498
gitlab-group-approval-rule-list command line option, 500
gitlab-group-audit-event-list command line option, 501
gitlab-group-badge-list command line option, 502
gitlab-group-billable-member-list command line option, 504
gitlab-group-billable-member-membership-list command line option, 505
gitlab-group-board-list command line option, 506
gitlab-group-board-list-list command line option, 508
gitlab-group-cluster-list command line option, 510
gitlab-group-custom-attribute-list command line option, 511
gitlab-group-deploy-token-list command line option, 513
gitlab-group-descendant-group-list command line option, 514
gitlab-group-epic-award-emoji-list command line option, 517
gitlab-group-epic-issue-list command line option, 520
gitlab-group-epic-list command line option, 515
gitlab-group-epic-note-award-emoji-list command line option, 524
gitlab-group-epic-note-list command line option, 522
gitlab-group-epic-resource-label-event-list command line option, 525
gitlab-group-hook-list command line option, 528
gitlab-group-invitation-list command line option, 531
gitlab-group-issue-list command line option, 533
gitlab-group-iteration-list command line option, 534
gitlab-group-label-list command line option, 535
gitlab-group-ldap-group-link-list command line option, 537
gitlab-group-list command line option, 493
gitlab-group-member-all-list command line option, 540
gitlab-group-member-list command line option, 539
gitlab-group-member-role-list command line option, 542
gitlab-group-merge-request-list command line option, 544
gitlab-group-milestone-list command line option, 546
gitlab-group-package-list command line option, 549
gitlab-group-project-list command line option, 550
gitlab-group-runner-list command line option, 553
gitlab-group-saml-group-link-list

command line option, 554
gitlab-group-service-account-access-token-list command line option, 558
gitlab-group-service-account-list command line option, 555
gitlab-group-subgroup-list command line option, 559
gitlab-group-variable-list command line option, 561
gitlab-group-wiki-list command line option, 563
gitlab-hook-list command line option, 564
gitlab-issue-list command line option, 566
gitlab-ldap-group-list command line option, 567
gitlab-license-list command line option, 568
gitlab-member-role-list command line option, 570
gitlab-merge-request-list command line option, 572
gitlab-namespace-list command line option, 573
gitlab-pages-domain-list command line option, 575
gitlab-personal-access-token-list command line option, 576
gitlab-project-access-request-list command line option, 594
gitlab-project-access-token-list command line option, 595
gitlab-project-approval-rule-list command line option, 599
gitlab-project-audit-event-list command line option, 601
gitlab-project-badge-list command line option, 602
gitlab-project-board-list command line option, 604
gitlab-project-board-list-list command line option, 606
gitlab-project-branch-list command line option, 608
gitlab-project-cluster-agent-list command line option, 613
gitlab-project-cluster-list command line option, 611
gitlab-project-commit-comment-list command line option, 617
gitlab-project-commit-discussion-list command line option, 619
gitlab-project-commit-list command line option, 615
gitlab-project-commit-status-list command line option, 622
gitlab-project-custom-attribute-list command line option, 623
gitlab-project-deploy-token-list command line option, 624
gitlab-project-deployment-list command line option, 626
gitlab-project-deployment-merge-request-list command line option, 629
gitlab-project-dockerfile-template-list command line option, 629
gitlab-project-environment-list command line option, 631
gitlab-project-event-list command line option, 632
gitlab-project-external-status-check-list command line option, 634
gitlab-project-feature-flag-list command line option, 637
gitlab-project-feature-flag-user-list-list command line option, 638
gitlab-project-fork-list command line option, 643
gitlab-project-gitignore-template-list command line option, 644
gitlab-project-gitlabciyml-template-list command line option, 645
gitlab-project-hook-list command line option, 647
gitlab-project-integration-list command line option, 649
gitlab-project-invitation-list command line option, 651
gitlab-project-issue-award-emoji-list command line option, 660
gitlab-project-issue-discussion-list command line option, 661
gitlab-project-issue-discussion-note-list command line option, 663
gitlab-project-issue-link-list command line option, 665
gitlab-project-issue-list command line option, 654
gitlab-project-issue-note-award-emoji-list command line option, 669
gitlab-project-issue-note-list command line option, 667
gitlab-project-issue-resource-iteration-event-list command line option, 670
gitlab-project-issue-resource-label-event-list command line option, 671
gitlab-project-issue-resource-milestone-event-list command line option, 672

gitlab-project-issue-resource-state-event-list command line option, 673
gitlab-project-issue-resource-weight-event-list command line option, 673
gitlab-project-issue-template-list command line option, 674
gitlab-project-job-list command line option, 677
gitlab-project-key-list command line option, 680
gitlab-project-label-list command line option, 682
gitlab-project-license-template-list command line option, 684
gitlab-project-list command line option, 583
gitlab-project-member-all-list command line option, 686
gitlab-project-member-list command line option, 685
gitlab-project-merge-request-approval-rule-list command line option, 699
gitlab-project-merge-request-award-emoji-list command line option, 701
gitlab-project-merge-request-diff-list command line option, 702
gitlab-project-merge-request-discussion-list command line option, 704
gitlab-project-merge-request-discussion-note-list command line option, 706
gitlab-project-merge-request-draft-note-list command line option, 708
gitlab-project-merge-request-list command line option, 691
gitlab-project-merge-request-note-award-emoji-list command line option, 712
gitlab-project-merge-request-note-list command line option, 710
gitlab-project-merge-request-pipeline-list command line option, 713
gitlab-project-merge-request-resource-label-event-list command line option, 714
gitlab-project-merge-request-resource-milestone-event-list command line option, 715
gitlab-project-merge-request-resource-state-event-list command line option, 716
gitlab-project-merge-request-reviewer-detail-list command line option, 716
gitlab-project-merge-request-status-check-list command line option, 717
gitlab-project-merge-request-template-list command line option, 718
gitlab-project-merge-train-list command line option, 718
gitlab-project-milestone-list command line option, 720
gitlab-project-note-list command line option, 722
gitlab-project-package-file-list command line option, 725
gitlab-project-package-list command line option, 724
gitlab-project-package-pipeline-list command line option, 726
gitlab-project-package-protection-rule-list command line option, 727
gitlab-project-pages-domain-list command line option, 730
gitlab-project-pipeline-bridge-list command line option, 733
gitlab-project-pipeline-job-list command line option, 734
gitlab-project-pipeline-list command line option, 732
gitlab-project-pipeline-schedule-list command line option, 735
gitlab-project-pipeline-schedule-pipeline-list command line option, 737
gitlab-project-pipeline-variable-list command line option, 740
gitlab-project-protected-branch-list command line option, 742
gitlab-project-protected-environment-list command line option, 743
gitlab-project-protected-tag-list command line option, 745
gitlab-project-registry-protection-rule-list command line option, 749
gitlab-project-registry-repository-list command line option, 751
gitlab-project-registry-repository-protection-rule-list command line option, 752
gitlab-project-registry-tag-list command line option, 754
gitlab-project-release-link-list command line option, 758
gitlab-project-release-list command line option, 756
gitlab-project-remote-mirror-list command line option, 760
gitlab-project-resource-group-list command line option, 761
gitlab-project-resource-group-upcoming-job-list command line option, 762
gitlab-project-runner-list command line option, 763
gitlab-project-secure-file-list command line option, 765

gitlab-project-service-account-access-token-list command line option, 770
gitlab-project-service-account-list command line option, 768
gitlab-project-service-list command line option, 766
gitlab-project-snippet-award-emoji-list command line option, 775
gitlab-project-snippet-discussion-list command line option, 776
gitlab-project-snippet-list command line option, 772
gitlab-project-snippet-note-award-emoji-list command line option, 782
gitlab-project-snippet-note-list command line option, 779
gitlab-project-tag-list command line option, 784
gitlab-project-trigger-list command line option, 785
gitlab-project-user-list command line option, 786
gitlab-project-variable-list command line option, 788
gitlab-project-wiki-list command line option, 790
gitlab-runner-all-list command line option, 794
gitlab-runner-job-list command line option, 794
gitlab-runner-list command line option, 792
gitlab-service-account-list command line option, 795
gitlab-shared-project-list command line option, 797
gitlab-snippet-list command line option, 798
gitlab-starred-project-list command line option, 801
gitlab-todo-list command line option, 801
gitlab-topic-list command line option, 803
gitlab-user-activities-list command line option, 810
gitlab-user-contributed-project-list command line option, 811
gitlab-user-custom-attribute-list command line option, 812
gitlab-user-email-list command line option, 813
gitlab-user-event-list command line option, 814
gitlab-user-gpg-key-list command line option, 815
gitlab-user-impersonation-token-list command line option, 817
gitlab-user-key-list command line option, 818
gitlab-user-list command line option, 808
gitlab-user-membership-list command line option, 819
gitlab-user-project-list command line option, 822
gitlab-variable-list command line option, 823
--performance-bar-allowed-group-id
gitlab-application-settings-update command line option, 471
--pipeline-events
gitlab-group-hook-create command line option, 527
gitlab-group-hook-update command line option, 529
gitlab-project-hook-create command line option, 646
gitlab-project-hook-update command line option, 648
--pipeline-id
gitlab-project-pipeline-bridge-list command line option, 733
gitlab-project-pipeline-job-list command line option, 734
gitlab-project-pipeline-test-report-get command line option, 739
gitlab-project-pipeline-test-report-summary-get command line option, 739
gitlab-project-pipeline-variable-list command line option, 740
--pipeline-schedule-id
gitlab-project-pipeline-schedule-pipeline-list command line option, 737
gitlab-project-pipeline-schedule-variable-create command line option, 738
gitlab-project-pipeline-schedule-variable-delete command line option, 738
gitlab-project-pipeline-schedule-variable-update command line option, 738
--plantuml-enabled
gitlab-application-settings-update command line option, 470
--plantuml-url
gitlab-application-settings-update command line option, 471
--platform-kubernetes-attributes
gitlab-group-cluster-create command line option, 509
gitlab-group-cluster-update command

line option, 510
gitlab-project-cluster-create command line option, 610
gitlab-project-cluster-update command line option, 611
--polling-interval-multiplier
gitlab-application-settings-update command line option, 471
--popular
gitlab-license-list command line option, 568
--position
gitlab-group-board-list-update command line option, 508
gitlab-project-board-list-update command line option, 607
gitlab-project-commit-discussion-note-create command line option, 619
gitlab-project-merge-request-discussion-create command line option, 703
gitlab-project-merge-request-draft-note-create command line option, 707
gitlab-project-merge-request-draft-note-update command line option, 709
--prevent-forking-outside-group
gitlab-group-update command line option, 496
--prevent-secrets
gitlab-group-push-rules-create command line option, 551
gitlab-group-push-rules-update command line option, 552
gitlab-project-push-rules-create command line option, 747
gitlab-project-push-rules-update command line option, 748
--printing-merge-request-link-enabled
gitlab-project-create command line option, 580
--priority
gitlab-group-label-create command line option, 534
gitlab-group-label-update command line option, 536
gitlab-project-label-create command line option, 681
gitlab-project-label-update command line option, 683
--private-profile
gitlab-user-create command line option, 806
gitlab-user-update command line option, 810
--private-token
gitlab command line option, 465
--process-mode
gitlab-project-resource-group-update command line option, 762
--project
gitlab-license-get command line option, 567
--project-creation-level
gitlab-group-create command line option, 491
gitlab-group-update command line option, 495
--project-id
gitlab-current-user-runner-create command line option, 482
gitlab-generic-package-download command line option, 486
gitlab-generic-package-upload command line option, 487
gitlab-group-transfer-project command line option, 494
gitlab-project-access-request-approve command line option, 593
gitlab-project-access-request-create command line option, 593
gitlab-project-access-request-delete command line option, 593
gitlab-project-access-request-list command line option, 594
gitlab-project-access-token-create command line option, 594
gitlab-project-access-token-delete command line option, 595
gitlab-project-access-token-get command line option, 595
gitlab-project-access-token-list command line option, 595
gitlab-project-access-token-rotate command line option, 596
gitlab-project-additional-statistics-get command line option, 596
gitlab-project-approval-get command line option, 596
gitlab-project-approval-rule-create command line option, 597
gitlab-project-approval-rule-delete command line option, 598
gitlab-project-approval-rule-get command line option, 598
gitlab-project-approval-rule-list command line option, 598
gitlab-project-approval-rule-update command line option, 599
gitlab-project-approval-update command

line option, 597
gitlab-project-artifact-delete command line option, 599
gitlab-project-artifact-download command line option, 600
gitlab-project-artifact-raw command line option, 600
gitlab-project-audit-event-get command line option, 600
gitlab-project-audit-event-list command line option, 601
gitlab-project-badge-create command line option, 601
gitlab-project-badge-delete command line option, 602
gitlab-project-badge-get command line option, 602
gitlab-project-badge-list command line option, 602
gitlab-project-badge-render command line option, 603
gitlab-project-badge-update command line option, 603
gitlab-project-board-create command line option, 603
gitlab-project-board-delete command line option, 604
gitlab-project-board-get command line option, 604
gitlab-project-board-list command line option, 604
gitlab-project-board-list-create command line option, 605
gitlab-project-board-list-delete command line option, 605
gitlab-project-board-list-get command line option, 606
gitlab-project-board-list-list command line option, 606
gitlab-project-board-list-update command line option, 607
gitlab-project-board-update command line option, 605
gitlab-project-branch-create command line option, 607
gitlab-project-branch-delete command line option, 607
gitlab-project-branch-get command line option, 608
gitlab-project-branch-list command line option, 608
gitlab-project-ci-lint-create command line option, 608
gitlab-project-ci-lint-get command line option, 609
gitlab-project-ci-lint-validate command line option, 609
gitlab-project-cluster-agent-create command line option, 612
gitlab-project-cluster-agent-delete command line option, 612
gitlab-project-cluster-agent-get command line option, 612
gitlab-project-cluster-agent-list command line option, 612
gitlab-project-cluster-create command line option, 610
gitlab-project-cluster-delete command line option, 610
gitlab-project-cluster-get command line option, 610
gitlab-project-cluster-list command line option, 611
gitlab-project-cluster-update command line option, 611
gitlab-project-commit-cherry-pick command line option, 613
gitlab-project-commit-comment-create command line option, 617
gitlab-project-commit-comment-list command line option, 617
gitlab-project-commit-create command line option, 613
gitlab-project-commit-diff command line option, 614
gitlab-project-commit-discussion-create command line option, 618
gitlab-project-commit-discussion-get command line option, 618
gitlab-project-commit-discussion-list command line option, 618
gitlab-project-commit-discussion-note-create command line option, 619
gitlab-project-commit-discussion-note-delete command line option, 620
gitlab-project-commit-discussion-note-get command line option, 620
gitlab-project-commit-discussion-note-update command line option, 620
gitlab-project-commit-get command line option, 614
gitlab-project-commit-list command line option, 614
gitlab-project-commit-merge-requests command line option, 615
gitlab-project-commit-refs command line option, 615
gitlab-project-commit-revert command

line option, 616
gitlab-project-commit-sequence command line option, 616
gitlab-project-commit-signature command line option, 616
gitlab-project-commit-status-create command line option, 621
gitlab-project-commit-status-list command line option, 621
gitlab-project-custom-attribute-delete command line option, 622
gitlab-project-custom-attribute-get command line option, 622
gitlab-project-custom-attribute-list command line option, 623
gitlab-project-deploy-token-create command line option, 623
gitlab-project-deploy-token-delete command line option, 623
gitlab-project-deploy-token-get command line option, 624
gitlab-project-deploy-token-list command line option, 624
gitlab-project-deployment-approval command line option, 625
gitlab-project-deployment-create command line option, 625
gitlab-project-deployment-get command line option, 625
gitlab-project-deployment-list command line option, 626
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-deployment-update command line option, 626
gitlab-project-dockerfile-template-get command line option, 629
gitlab-project-dockerfile-template-list command line option, 629
gitlab-project-environment-create command line option, 630
gitlab-project-environment-delete command line option, 630
gitlab-project-environment-get command line option, 630
gitlab-project-environment-list command line option, 631
gitlab-project-environment-stop command line option, 631
gitlab-project-environment-update command line option, 631
gitlab-project-event-list command line option, 632
gitlab-project-export-create command line option, 633
gitlab-project-export-download command line option, 633
gitlab-project-export-get command line option, 633
gitlab-project-external-status-check-create command line option, 634
gitlab-project-external-status-check-delete command line option, 634
gitlab-project-external-status-check-list command line option, 634
gitlab-project-external-status-check-update command line option, 635
gitlab-project-feature-flag-create command line option, 635
gitlab-project-feature-flag-delete command line option, 636
gitlab-project-feature-flag-get command line option, 636
gitlab-project-feature-flag-list command line option, 636
gitlab-project-feature-flag-update command line option, 637
gitlab-project-feature-flag-user-list-create command line option, 637
gitlab-project-feature-flag-user-list-delete command line option, 638
gitlab-project-feature-flag-user-list-get command line option, 638
gitlab-project-feature-flag-user-list-list command line option, 638
gitlab-project-feature-flag-user-list-update command line option, 639
gitlab-project-file-blame command line option, 639
gitlab-project-file-create command line option, 640
gitlab-project-file-delete command line option, 640
gitlab-project-file-get command line option, 641
gitlab-project-file-raw command line option, 641
gitlab-project-file-update command line option, 641
gitlab-project-fork-create command line option, 642
gitlab-project-fork-list command line option, 643
gitlab-project-gitignore-template-get command line option, 643
gitlab-project-gitignore-template-list command line option, 644
gitlab-project-gitlabciyml-template-get

command line option, 644
gitlab-project-gitlabciyml-template-list command line option, 645
gitlab-project-hook-create command line option, 645
gitlab-project-hook-delete command line option, 646
gitlab-project-hook-get command line option, 646
gitlab-project-hook-list command line option, 647
gitlab-project-hook-update command line option, 647
gitlab-project-import-get command line option, 648
gitlab-project-integration-available command line option, 648
gitlab-project-integration-delete command line option, 649
gitlab-project-integration-get command line option, 649
gitlab-project-integration-list command line option, 649
gitlab-project-integration-update command line option, 649
gitlab-project-invitation-create command line option, 650
gitlab-project-invitation-delete command line option, 650
gitlab-project-invitation-get command line option, 651
gitlab-project-invitation-list command line option, 651
gitlab-project-invitation-update command line option, 651
gitlab-project-issue-add-spent-time command line option, 652
gitlab-project-issue-award-emoji-create command line option, 659
gitlab-project-issue-award-emoji-delete command line option, 659
gitlab-project-issue-award-emoji-get command line option, 660
gitlab-project-issue-award-emoji-list command line option, 660
gitlab-project-issue-closed-by command line option, 652
gitlab-project-issue-create command line option, 653
gitlab-project-issue-delete command line option, 653
gitlab-project-issue-discussion-create command line option, 661
gitlab-project-issue-discussion-get command line option, 661
gitlab-project-issue-discussion-list command line option, 661
gitlab-project-issue-discussion-note-create command line option, 662
gitlab-project-issue-discussion-note-delete command line option, 662
gitlab-project-issue-discussion-note-get command line option, 663
gitlab-project-issue-discussion-note-list command line option, 663
gitlab-project-issue-discussion-note-update command line option, 664
gitlab-project-issue-get command line option, 653
gitlab-project-issue-link-create command line option, 664
gitlab-project-issue-link-delete command line option, 665
gitlab-project-issue-link-list command line option, 665
gitlab-project-issue-list command line option, 654
gitlab-project-issue-move command line option, 655
gitlab-project-issue-note-award-emoji-create command line option, 667
gitlab-project-issue-note-award-emoji-delete command line option, 668
gitlab-project-issue-note-award-emoji-get command line option, 668
gitlab-project-issue-note-award-emoji-list command line option, 669
gitlab-project-issue-note-create command line option, 665
gitlab-project-issue-note-delete command line option, 666
gitlab-project-issue-note-get command line option, 666
gitlab-project-issue-note-list command line option, 666
gitlab-project-issue-note-update command line option, 667
gitlab-project-issue-participants command line option, 655
gitlab-project-issue-related-merge-requests command line option, 655
gitlab-project-issue-reorder command line option, 656
gitlab-project-issue-reset-spent-time command line option, 656
gitlab-project-issue-reset-time-estimate command line option, 656
gitlab-project-issue-resource-iteration-event-get

command line option, 669
gitlab-project-issue-resource-iteration-event-list command line option, 670
gitlab-project-issue-resource-label-event-get command line option, 670
gitlab-project-issue-resource-label-event-list command line option, 670
gitlab-project-issue-resource-milestone-event-get command line option, 671
gitlab-project-issue-resource-milestone-event-list command line option, 671
gitlab-project-issue-resource-state-event-get command line option, 672
gitlab-project-issue-resource-state-event-list command line option, 672
gitlab-project-issue-resource-weight-event-get command line option, 673
gitlab-project-issue-resource-weight-event-list command line option, 673
gitlab-project-issue-subscribe command line option, 656
gitlab-project-issue-template-get command line option, 674
gitlab-project-issue-template-list command line option, 674
gitlab-project-issue-time-estimate command line option, 657
gitlab-project-issue-time-stats command line option, 657
gitlab-project-issue-todo command line option, 657
gitlab-project-issue-unsubscribe command line option, 658
gitlab-project-issue-update command line option, 658
gitlab-project-issue-user-agent-detail command line option, 659
gitlab-project-issues-statistics-get command line option, 675
gitlab-project-job-artifact command line option, 675
gitlab-project-job-artifacts command line option, 675
gitlab-project-job-cancel command line option, 676
gitlab-project-job-delete-artifacts command line option, 676
gitlab-project-job-erase command line option, 676
gitlab-project-job-get command line option, 676
gitlab-project-job-keep-artifacts command line option, 677
gitlab-project-job-list command line option, 677
gitlab-project-job-play command line option, 677
gitlab-project-job-retry command line option, 678
gitlab-project-job-token-scope-get command line option, 678
gitlab-project-job-token-scope-update command line option, 679
gitlab-project-job-trace command line option, 678
gitlab-project-key-create command line option, 679
gitlab-project-key-delete command line option, 679
gitlab-project-key-enable command line option, 679
gitlab-project-key-get command line option, 680
gitlab-project-key-list command line option, 680
gitlab-project-key-update command line option, 680
gitlab-project-label-create command line option, 681
gitlab-project-label-delete command line option, 681
gitlab-project-label-get command line option, 682
gitlab-project-label-list command line option, 682
gitlab-project-label-subscribe command line option, 682
gitlab-project-label-unsubscribe command line option, 682
gitlab-project-label-update command line option, 683
gitlab-project-license-template-get command line option, 683
gitlab-project-license-template-list command line option, 684
gitlab-project-member-all-get command line option, 686
gitlab-project-member-all-list command line option, 686
gitlab-project-member-create command line option, 684
gitlab-project-member-delete command line option, 685
gitlab-project-member-get command line option, 685
gitlab-project-member-list command line option, 685
gitlab-project-member-update command

line option, 685
gitlab-project-merge-request-add-spent-time command line option, 687
gitlab-project-merge-request-approval-get command line option, 697
gitlab-project-merge-request-approval-rule-create command line option, 698
gitlab-project-merge-request-approval-rule-delete command line option, 698
gitlab-project-merge-request-approval-rule-get command line option, 698
gitlab-project-merge-request-approval-rule-list command line option, 699
gitlab-project-merge-request-approval-rule-update command line option, 699
gitlab-project-merge-request-approval-state-get command line option, 700
gitlab-project-merge-request-approval-update command line option, 697
gitlab-project-merge-request-approve command line option, 687
gitlab-project-merge-request-award-emoji-create command line option, 700
gitlab-project-merge-request-award-emoji-delete command line option, 701
gitlab-project-merge-request-award-emoji-get command line option, 701
gitlab-project-merge-request-award-emoji-list command line option, 701
gitlab-project-merge-request-cancel-merge-when-pipeline-succeeds command line option, 687
gitlab-project-merge-request-changes command line option, 688
gitlab-project-merge-request-closes-issues command line option, 688
gitlab-project-merge-request-commits command line option, 688
gitlab-project-merge-request-create command line option, 689
gitlab-project-merge-request-delete command line option, 690
gitlab-project-merge-request-diff-get command line option, 702
gitlab-project-merge-request-diff-list command line option, 702
gitlab-project-merge-request-discussion-create command line option, 703
gitlab-project-merge-request-discussion-get command line option, 703
gitlab-project-merge-request-discussion-list command line option, 704
gitlab-project-merge-request-discussion-note-create command line option, 705
gitlab-project-merge-request-discussion-note-delete command line option, 705
gitlab-project-merge-request-discussion-note-get command line option, 705
gitlab-project-merge-request-discussion-note-list command line option, 706
gitlab-project-merge-request-discussion-note-update command line option, 706
gitlab-project-merge-request-discussion-update command line option, 704
gitlab-project-merge-request-draft-note-create command line option, 707
gitlab-project-merge-request-draft-note-delete command line option, 707
gitlab-project-merge-request-draft-note-get command line option, 708
gitlab-project-merge-request-draft-note-list command line option, 708
gitlab-project-merge-request-draft-note-update command line option, 708
gitlab-project-merge-request-get command line option, 690
gitlab-project-merge-request-list command line option, 691
gitlab-project-merge-request-merge command line option, 692
gitlab-project-merge-request-merge-ref command line option, 692
gitlab-project-merge-request-note-award-emoji-create command line option, 711
gitlab-project-merge-request-note-award-emoji-delete command line option, 711
gitlab-project-merge-request-note-award-emoji-get command line option, 712
gitlab-project-merge-request-note-award-emoji-list command line option, 712
gitlab-project-merge-request-note-create command line option, 709
gitlab-project-merge-request-note-delete command line option, 709
gitlab-project-merge-request-note-get command line option, 710
gitlab-project-merge-request-note-list command line option, 710
gitlab-project-merge-request-note-update command line option, 710
gitlab-project-merge-request-participants command line option, 692
gitlab-project-merge-request-pipeline-create command line option, 713
gitlab-project-merge-request-pipeline-list command line option, 713
gitlab-project-merge-request-rebase command line option, 693
gitlab-project-merge-request-related-issues

command line option, 693
gitlab-project-merge-request-reset-approvals command line option, 693
gitlab-project-merge-request-reset-spent-time command line option, 694
gitlab-project-merge-request-reset-time-estimate command line option, 694
gitlab-project-merge-request-resource-label-event-get command line option, 713
gitlab-project-merge-request-resource-label-event-list command line option, 714
gitlab-project-merge-request-resource-milestone-event-get command line option, 714
gitlab-project-merge-request-resource-milestone-event-list command line option, 715
gitlab-project-merge-request-resource-state-event-get command line option, 715
gitlab-project-merge-request-resource-state-event-list command line option, 716
gitlab-project-merge-request-reviewer-detail-list command line option, 716
gitlab-project-merge-request-status-check-list command line option, 717
gitlab-project-merge-request-subscribe command line option, 694
gitlab-project-merge-request-template-get command line option, 717
gitlab-project-merge-request-template-list command line option, 718
gitlab-project-merge-request-time-estimate command line option, 694
gitlab-project-merge-request-time-stats command line option, 695
gitlab-project-merge-request-todo command line option, 695
gitlab-project-merge-request-unapprove command line option, 695
gitlab-project-merge-request-unsubscribe command line option, 695
gitlab-project-merge-request-update command line option, 696
gitlab-project-merge-train-list command line option, 718
gitlab-project-milestone-create command line option, 719
gitlab-project-milestone-delete command line option, 719
gitlab-project-milestone-get command line option, 720
gitlab-project-milestone-issues command line option, 720
gitlab-project-milestone-list command line option, 720
gitlab-project-milestone-merge-requests command line option, 721
gitlab-project-milestone-update command line option, 721
gitlab-project-note-get command line option, 721
gitlab-project-note-list command line option, 722
gitlab-project-notification-settings-get command line option, 722
gitlab-project-notification-settings-update command line option, 723
gitlab-project-package-delete command line option, 724
gitlab-project-package-file-delete command line option, 725
gitlab-project-package-file-list command line option, 725
gitlab-project-package-get command line option, 724
gitlab-project-package-list command line option, 724
gitlab-project-package-pipeline-list command line option, 726
gitlab-project-package-protection-rule-create command line option, 727
gitlab-project-package-protection-rule-delete command line option, 727
gitlab-project-package-protection-rule-list command line option, 727
gitlab-project-package-protection-rule-update command line option, 728
gitlab-project-pages-delete command line option, 728
gitlab-project-pages-domain-create command line option, 729
gitlab-project-pages-domain-delete command line option, 729
gitlab-project-pages-domain-get command line option, 730
gitlab-project-pages-domain-list command line option, 730
gitlab-project-pages-domain-update command line option, 730
gitlab-project-pages-get command line option, 728
gitlab-project-pages-update command line option, 729
gitlab-project-pipeline-bridge-list command line option, 733
gitlab-project-pipeline-cancel command line option, 731
gitlab-project-pipeline-create command line option, 731
gitlab-project-pipeline-delete command

line option, 731
gitlab-project-pipeline-get command line option, 732
gitlab-project-pipeline-job-list command line option, 734
gitlab-project-pipeline-list command line option, 732
gitlab-project-pipeline-retry command line option, 733
gitlab-project-pipeline-schedule-create command line option, 734
gitlab-project-pipeline-schedule-delete command line option, 735
gitlab-project-pipeline-schedule-get command line option, 735
gitlab-project-pipeline-schedule-list command line option, 735
gitlab-project-pipeline-schedule-pipeline-list command line option, 737
gitlab-project-pipeline-schedule-play command line option, 736
gitlab-project-pipeline-schedule-take-ownership command line option, 736
gitlab-project-pipeline-schedule-update command line option, 736
gitlab-project-pipeline-schedule-variable-create command line option, 738
gitlab-project-pipeline-schedule-variable-delete command line option, 738
gitlab-project-pipeline-schedule-variable-update command line option, 738
gitlab-project-pipeline-test-report-get command line option, 739
gitlab-project-pipeline-test-report-summary-get command line option, 739
gitlab-project-pipeline-variable-list command line option, 740
gitlab-project-protected-branch-create command line option, 741
gitlab-project-protected-branch-delete command line option, 741
gitlab-project-protected-branch-get command line option, 741
gitlab-project-protected-branch-list command line option, 742
gitlab-project-protected-branch-update command line option, 742
gitlab-project-protected-environment-create command line option, 742
gitlab-project-protected-environment-delete command line option, 743
gitlab-project-protected-environment-get command line option, 743
gitlab-project-protected-environment-list command line option, 743
gitlab-project-protected-tag-create command line option, 744
gitlab-project-protected-tag-delete command line option, 744
gitlab-project-protected-tag-get command line option, 744
gitlab-project-protected-tag-list command line option, 745
gitlab-project-pull-mirror-create command line option, 745
gitlab-project-pull-mirror-get command line option, 746
gitlab-project-pull-mirror-start command line option, 746
gitlab-project-pull-mirror-update command line option, 746
gitlab-project-push-rules-create command line option, 747
gitlab-project-push-rules-delete command line option, 747
gitlab-project-push-rules-get command line option, 747
gitlab-project-push-rules-update command line option, 748
gitlab-project-registry-protection-rule-create command line option, 749
gitlab-project-registry-protection-rule-list command line option, 749
gitlab-project-registry-protection-rule-update command line option, 750
gitlab-project-registry-repository-delete command line option, 750
gitlab-project-registry-repository-list command line option, 750
gitlab-project-registry-repository-protection-rule-cre command line option, 751
gitlab-project-registry-repository-protection-rule-del command line option, 751
gitlab-project-registry-repository-protection-rule-lis command line option, 752
gitlab-project-registry-repository-protection-rule-upd command line option, 752
gitlab-project-registry-tag-delete command line option, 753
gitlab-project-registry-tag-delete-in-bulk command line option, 753
gitlab-project-registry-tag-get command line option, 754
gitlab-project-registry-tag-list command line option, 754
gitlab-project-release-create command line option, 754
gitlab-project-release-delete command

line option, 755
gitlab-project-release-get command line option, 755
gitlab-project-release-link-create command line option, 757
gitlab-project-release-link-delete command line option, 757
gitlab-project-release-link-get command line option, 757
gitlab-project-release-link-list command line option, 758
gitlab-project-release-link-update command line option, 758
gitlab-project-release-list command line option, 755
gitlab-project-release-update command line option, 756
gitlab-project-remote-mirror-create command line option, 759
gitlab-project-remote-mirror-delete command line option, 759
gitlab-project-remote-mirror-list command line option, 759
gitlab-project-remote-mirror-sync command line option, 760
gitlab-project-remote-mirror-update command line option, 760
gitlab-project-resource-group-get command line option, 761
gitlab-project-resource-group-list command line option, 761
gitlab-project-resource-group-upcoming-job-list command line option, 762
gitlab-project-resource-group-update command line option, 762
gitlab-project-runner-create command line option, 763
gitlab-project-runner-delete command line option, 763
gitlab-project-runner-list command line option, 763
gitlab-project-secure-file-create command line option, 764
gitlab-project-secure-file-delete command line option, 764
gitlab-project-secure-file-download command line option, 764
gitlab-project-secure-file-get command line option, 765
gitlab-project-secure-file-list command line option, 765
gitlab-project-service-account-access-token-create command line option, 769
gitlab-project-service-account-access-token-delete command line option, 769
gitlab-project-service-account-access-token-list command line option, 770
gitlab-project-service-account-access-token-rotate command line option, 771
gitlab-project-service-account-create command line option, 767
gitlab-project-service-account-delete command line option, 767
gitlab-project-service-account-list command line option, 768
gitlab-project-service-account-update command line option, 768
gitlab-project-service-available command line option, 765
gitlab-project-service-delete command line option, 766
gitlab-project-service-get command line option, 766
gitlab-project-service-list command line option, 766
gitlab-project-service-update command line option, 767
gitlab-project-snippet-award-emoji-create command line option, 774
gitlab-project-snippet-award-emoji-delete command line option, 774
gitlab-project-snippet-award-emoji-get command line option, 774
gitlab-project-snippet-award-emoji-list command line option, 775
gitlab-project-snippet-content command line option, 771
gitlab-project-snippet-create command line option, 771
gitlab-project-snippet-delete command line option, 772
gitlab-project-snippet-discussion-create command line option, 775
gitlab-project-snippet-discussion-get command line option, 776
gitlab-project-snippet-discussion-list command line option, 776
gitlab-project-snippet-discussion-note-create command line option, 777
gitlab-project-snippet-discussion-note-delete command line option, 777
gitlab-project-snippet-discussion-note-get command line option, 778
gitlab-project-snippet-discussion-note-update command line option, 778
gitlab-project-snippet-get command line option, 772
gitlab-project-snippet-list command

line option, 772
gitlab-project-snippet-note-award-emoji-create command line option, 780
gitlab-project-snippet-note-award-emoji-delete command line option, 781
gitlab-project-snippet-note-award-emoji-get command line option, 781
gitlab-project-snippet-note-award-emoji-list command line option, 781
gitlab-project-snippet-note-create command line option, 778
gitlab-project-snippet-note-delete command line option, 779
gitlab-project-snippet-note-get command line option, 779
gitlab-project-snippet-note-list command line option, 779
gitlab-project-snippet-note-update command line option, 780
gitlab-project-snippet-update command line option, 773
gitlab-project-snippet-user-agent-detail command line option, 773
gitlab-project-storage-get command line option, 782
gitlab-project-tag-create command line option, 782
gitlab-project-tag-delete command line option, 783
gitlab-project-tag-get command line option, 783
gitlab-project-tag-list command line option, 783
gitlab-project-trigger-create command line option, 784
gitlab-project-trigger-delete command line option, 784
gitlab-project-trigger-get command line option, 784
gitlab-project-trigger-list command line option, 785
gitlab-project-trigger-update command line option, 785
gitlab-project-user-list command line option, 786
gitlab-project-variable-create command line option, 786
gitlab-project-variable-delete command line option, 787
gitlab-project-variable-get command line option, 787
gitlab-project-variable-list command line option, 787
gitlab-project-variable-update command line option, 788
gitlab-project-wiki-create command line option, 789
gitlab-project-wiki-delete command line option, 789
gitlab-project-wiki-get command line option, 789
gitlab-project-wiki-list command line option, 789
gitlab-project-wiki-update command line option, 790
gitlab-project-wiki-upload command line option, 790
gitlab-todo-list command line option, 801
--projects-limit
gitlab-user-create command line option, 805
gitlab-user-update command line option, 810
--protected
gitlab-group-variable-create command line option, 560
gitlab-group-variable-update command line option, 561
gitlab-project-variable-create command line option, 787
gitlab-project-variable-update command line option, 788
gitlab-variable-create command line option, 823
gitlab-variable-update command line option, 824
--protected-branch-ids
gitlab-project-approval-rule-create command line option, 598
gitlab-project-external-status-check-create command line option, 634
gitlab-project-external-status-check-update command line option, 635
--provider
gitlab-group-ldap-group-link-create command line option, 537
gitlab-ldap-group-list command line option, 567
gitlab-user-create command line option, 805
gitlab-user-list command line option, 808
gitlab-user-update command line option, 810
--public
gitlab-user-project-create command line option, 820
--public-builds
gitlab-project-create command line

option, 580
gitlab-project-update command line option, 591
gitlab-user-project-create command line option, 820
--public-email
gitlab-user-create command line option, 806
gitlab-user-list command line option, 807
gitlab-user-update command line option, 810
--push-access-level
gitlab-project-protected-branch-create command line option, 741
--push-events
gitlab-group-hook-create command line option, 527
gitlab-group-hook-update command line option, 529
gitlab-project-hook-create command line option, 645
gitlab-project-hook-update command line option, 647
--query
gitlab-group-invitation-list command line option, 531
gitlab-project-invitation-list command line option, 651
--raw
gitlab-group-variable-create command line option, 560
gitlab-group-variable-update command line option, 561
gitlab-project-variable-create command line option, 787
gitlab-project-variable-update command line option, 788
gitlab-variable-create command line option, 823
gitlab-variable-update command line option, 824
--read-code
gitlab-group-member-role-create command line option, 542
gitlab-member-role-create command line option, 569
--read-dependency
gitlab-group-member-role-create command line option, 542
gitlab-member-role-create command line option, 569
--read-runners
gitlab-group-member-role-create command line option, 542
gitlab-member-role-create command line option, 569
--read-vulnerability
gitlab-group-member-role-create command line option, 542
gitlab-member-role-create command line option, 569
--reassign-issue
gitlab-group-notification-settings-update command line option, 548
gitlab-notification-settings-update command line option, 574
gitlab-project-notification-settings-update command line option, 723
--reassign-merge-request
gitlab-group-notification-settings-update command line option, 548
gitlab-notification-settings-update command line option, 574
gitlab-project-notification-settings-update command line option, 723
--recursive
gitlab-project-repository-tree command line option, 585
--redirect-uri
gitlab-application-create command line option, 466
--ref
gitlab-project-branch-create command line option, 607
gitlab-project-ci-lint-create command line option, 609
gitlab-project-ci-lint-get command line option, 609
gitlab-project-ci-lint-validate command line option, 609
gitlab-project-commit-status-create command line option, 621
gitlab-project-deployment-create command line option, 625
gitlab-project-file-blame command line option, 639
gitlab-project-file-get command line option, 641
gitlab-project-file-raw command line option, 641
gitlab-project-pipeline-create command line option, 731
gitlab-project-pipeline-list command line option, 732
gitlab-project-pipeline-schedule-create command line option, 735
gitlab-project-pipeline-schedule-update command line option, 736

gitlab-project-release-create command line option, 755
gitlab-project-repository-tree command line option, 585
gitlab-project-tag-create command line option, 783
gitlab-project-trigger-pipeline command line option, 587
--ref-name
gitlab-project-artifact-delete command line option, 599
gitlab-project-artifact-download command line option, 600
gitlab-project-artifact-raw command line option, 600
gitlab-project-commit-list command line option, 614
--refs
gitlab-project-repository-merge-base command line option, 585
--reject-unsigned-commits
gitlab-group-push-rules-create command line option, 551
gitlab-group-push-rules-update command line option, 552
gitlab-project-push-rules-create command line option, 747
gitlab-project-push-rules-update command line option, 748
--released-at
gitlab-project-release-update command line option, 756
--releases-access-level
gitlab-project-create command line option, 580
gitlab-project-update command line option, 591
--releases-events
gitlab-group-hook-create command line option, 527
gitlab-group-hook-update command line option, 529
--remove-group
gitlab-group-member-role-create command line option, 542
gitlab-member-role-create command line option, 569
--remove-project
gitlab-group-member-role-create command line option, 542
gitlab-member-role-create command line option, 569
--remove-source-branch
gitlab-project-merge-request-create command line option, 689
gitlab-project-merge-request-update command line option, 696
--remove-source-branch-after-merge
gitlab-project-create command line option, 580
gitlab-project-update command line option, 591
--render-html
gitlab-project-merge-request-get command line option, 690
--reopen-issue
gitlab-group-notification-settings-update command line option, 547
gitlab-notification-settings-update command line option, 574
gitlab-project-notification-settings-update command line option, 723
--reopen-merge-request
gitlab-group-notification-settings-update command line option, 548
gitlab-notification-settings-update command line option, 574
gitlab-project-notification-settings-update command line option, 723
--repos-max-capacity
gitlab-geo-node-update command line option, 489
--repository-access-level
gitlab-project-create command line option, 580
gitlab-project-update command line option, 591
--repository-checksum-failed
gitlab-project-list command line option, 583
gitlab-user-project-list command line option, 822
--repository-id
gitlab-project-registry-tag-delete command line option, 753
gitlab-project-registry-tag-delete-in-bulk command line option, 753
gitlab-project-registry-tag-get command line option, 754
gitlab-project-registry-tag-list command line option, 754
--repository-path-pattern
gitlab-project-registry-protection-rule-create command line option, 749
gitlab-project-registry-protection-rule-update command line option, 750
gitlab-project-registry-repository-protection-rule-cre
command line option, 751

gitlab-project-registry-repository-protection-rule-update command line option, 752
--repository-storage
gitlab-project-create command line option, 580
gitlab-project-list command line option, 583
gitlab-project-update command line option, 591
--repository-storages
gitlab-application-settings-update command line option, 470
--represented-as
gitlab-project-deployment-approval command line option, 625
--request-access-enabled
gitlab-group-create command line option, 492
gitlab-group-update command line option, 496
gitlab-project-create command line option, 580
gitlab-project-update command line option, 592
--require-two-factor-authentication
gitlab-group-create command line option, 491
gitlab-group-update command line option, 495
--required-approval-count
gitlab-project-protected-environment-create command line option, 743
--requirements-access-level
gitlab-project-create command line option, 579
gitlab-project-update command line option, 591
--reset-approvals-on-push
gitlab-project-approval-update command line option, 597
--reset-password
gitlab-user-create command line option, 805
--resolve-discussion
gitlab-project-merge-request-draft-note-create command line option, 707
--resolve-outdated-diff-discussions
gitlab-project-create command line option, 580
gitlab-project-update command line option, 592
--resolved
gitlab-project-merge-request-discussion-update command line option, 704
--resource-group-key
gitlab-project-resource-group-upcoming-job-list command line option, 762
--restrict-user-defined-variables
gitlab-project-update command line option, 591
--restricted-visibility-levels
gitlab-application-settings-update command line option, 470
--reviewer-id
gitlab-merge-request-list command line option, 571
gitlab-project-deployment-merge-request-list command line option, 628
--reviewer-ids
gitlab-project-merge-request-create command line option, 689
gitlab-project-merge-request-update command line option, 696
--reviewer-username
gitlab-merge-request-list command line option, 571
gitlab-project-deployment-merge-request-list command line option, 628
--revoked
gitlab-group-service-account-access-token-list command line option, 558
gitlab-project-service-account-access-token-list command line option, 770
--rsa-key-restriction
gitlab-application-settings-update command line option, 471
--rule-type
gitlab-group-approval-rule-create command line option, 499
--run-untagged
gitlab-current-user-runner-create command line option, 482
gitlab-runner-create command line option, 791
gitlab-runner-update command line option, 793
--runner-id
gitlab-project-runner-create command line option, 763
gitlab-runner-job-list command line option, 794
--runner-type
gitlab-current-user-runner-create command line option, 482
--saml-group-name
gitlab-group-saml-group-link-create command line option, 553
--scope

gitlab-event-list command line option, 485
gitlab-group-merge-request-list command line option, 543
gitlab-group-runner-list command line option, 552
gitlab-group-search command line option, 493
gitlab-issue-list command line option, 565
gitlab-merge-request-list command line option, 571
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-event-list command line option, 632
gitlab-project-feature-flag-list command line option, 636
gitlab-project-issue-list command line option, 654
gitlab-project-job-list command line option, 677
gitlab-project-merge-request-list command line option, 691
gitlab-project-merge-train-list command line option, 718
gitlab-project-pipeline-bridge-list command line option, 733
gitlab-project-pipeline-job-list command line option, 734
gitlab-project-pipeline-list command line option, 732
gitlab-project-runner-list command line option, 763
gitlab-project-search command line option, 585
gitlab-runner-all command line option, 791
gitlab-runner-all-list command line option, 794
gitlab-runner-list command line option, 792
gitlab-user-event-list command line option, 814
--scopes
gitlab-application-create command line option, 466
gitlab-group-access-token-create command line option, 498
gitlab-group-deploy-token-create command line option, 512
gitlab-group-deploy-token-list command line option, 513
gitlab-group-service-account-access-token-create command line option, 556
gitlab-project-access-token-create command line option, 594
gitlab-project-deploy-token-create command line option, 623
gitlab-project-deploy-token-list command line option, 624
gitlab-project-service-account-access-token-create command line option, 769
gitlab-user-impersonation-token-create command line option, 816
gitlab-user-personal-access-token-create command line option, 819
--search
gitlab-group-billable-member-list command line option, 504
gitlab-group-descendant-group-list command line option, 513
gitlab-group-epic-list command line option, 515
gitlab-group-issue-list command line option, 532
gitlab-group-iteration-list command line option, 534
gitlab-group-list command line option, 493
gitlab-group-merge-request-list command line option, 544
gitlab-group-milestone-list command line option, 546
gitlab-group-project-list command line option, 549
gitlab-group-search command line option, 493
gitlab-group-service-account-access-token-list command line option, 558
gitlab-group-subgroup-list command line option, 559
gitlab-issue-list command line option, 566
gitlab-ldap-group-list command line option, 567
gitlab-merge-request-list command line option, 572
gitlab-namespace-list command line option, 573
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-environment-list command line option, 631
gitlab-project-feature-flag-user-list-list command line option, 638
gitlab-project-fork-list command line option, 643

gitlab-project-issue-list command line option, 654
gitlab-project-list command line option, 583
gitlab-project-merge-request-list command line option, 691
gitlab-project-milestone-list command line option, 720
gitlab-project-search command line option, 586
gitlab-project-service-account-access-token-list command line option, 770
gitlab-project-tag-list command line option, 783
gitlab-project-user-list command line option, 786
gitlab-shared-project-list command line option, 796
gitlab-starred-project-list command line option, 800
gitlab-user-list command line option, 807
gitlab-user-project-list command line option, 821
--search-namespaces
gitlab-project-list command line option, 583
--security-and-compliance-access-level
gitlab-project-create command line option, 580
gitlab-project-update command line option, 592
--server-url
gitlab command line option, 464
--service-desk-enabled
gitlab-project-update command line option, 592
--session-expire-delay
gitlab-application-settings-update command line option, 470
--sha
gitlab-project-deployment-create command line option, 625
gitlab-project-merge-request-approve command line option, 687
gitlab-project-pipeline-list command line option, 732
gitlab-project-repository-archive command line option, 584
gitlab-project-repository-blob command line option, 584
gitlab-project-repository-raw-blob command line option, 585
--share-with-group-lock
gitlab-group-create command line option, 491
gitlab-group-update command line option, 495
--shared-runners-enabled
gitlab-project-create command line option, 580
gitlab-project-update command line option, 592
--shared-runners-minutes-limit
gitlab-group-create command line option, 492
gitlab-group-update command line option, 496
--shared-runners-setting
gitlab-group-update command line option, 496
--shared-secret
gitlab-project-external-status-check-create command line option, 634
gitlab-project-external-status-check-update command line option, 635
--should-remove-source-branch
gitlab-project-merge-request-merge command line option, 692
--show-default-award-emojis
gitlab-project-create command line option, 580
gitlab-project-update command line option, 592
--sign-in-text
gitlab-application-settings-update command line option, 470
--signup-enabled
gitlab-application-settings-update command line option, 469
--silent-mode-enabled
gitlab-application-settings-update command line option, 469
--simple
gitlab-group-project-list command line option, 549
gitlab-project-fork-list command line option, 643
gitlab-project-list command line option, 583
gitlab-shared-project-list command line option, 797
gitlab-starred-project-list command line option, 800
gitlab-user-project-list command line option, 821
--since
gitlab-project-commit-list command line option, 614

--skip-confirmation
    gitlab-user-create command line option, 806
    gitlab-user-email-create command line option, 813
--skip-groups
    gitlab-group-descendant-group-list command line option, 513
    gitlab-group-list command line option, 493
    gitlab-group-subgroup-list command line option, 559
--skip-ldap
    gitlab-user-list command line option, 808
--skip-login
    gitlab command line option, 465
--skip-reconfirmation
    gitlab-user-update command line option, 810
--skip-users
    gitlab-project-user-list command line option, 786
--skype
    gitlab-user-create command line option, 805
    gitlab-user-update command line option, 809
--slug
    gitlab-group-wiki-delete command line option, 562
    gitlab-group-wiki-get command line option, 562
    gitlab-group-wiki-update command line option, 563
    gitlab-project-integration-available command line option, 648
    gitlab-project-integration-delete command line option, 649
    gitlab-project-integration-get command line option, 649
    gitlab-project-integration-update command line option, 650
    gitlab-project-service-available command line option, 765
    gitlab-project-service-delete command line option, 766
    gitlab-project-service-get command line option, 766
    gitlab-project-service-update command line option, 767
    gitlab-project-wiki-delete command line option, 789
    gitlab-project-wiki-get command line option, 789
    gitlab-project-wiki-update command line option, 790
    gitlab-project-wiki-upload command line option, 790
--snippet-id
    gitlab-project-snippet-award-emoji-create command line option, 774
    gitlab-project-snippet-award-emoji-delete command line option, 774
    gitlab-project-snippet-award-emoji-get command line option, 774
    gitlab-project-snippet-award-emoji-list command line option, 775
    gitlab-project-snippet-discussion-create command line option, 775
    gitlab-project-snippet-discussion-get command line option, 776
    gitlab-project-snippet-discussion-list command line option, 776
    gitlab-project-snippet-discussion-note-create command line option, 777
    gitlab-project-snippet-discussion-note-delete command line option, 777
    gitlab-project-snippet-discussion-note-get command line option, 778
    gitlab-project-snippet-discussion-note-update command line option, 778
    gitlab-project-snippet-note-award-emoji-create command line option, 780
    gitlab-project-snippet-note-award-emoji-delete command line option, 781
    gitlab-project-snippet-note-award-emoji-get command line option, 781
    gitlab-project-snippet-note-award-emoji-list command line option, 782
    gitlab-project-snippet-note-create command line option, 778
    gitlab-project-snippet-note-delete command line option, 779
    gitlab-project-snippet-note-get command line option, 779
    gitlab-project-snippet-note-list command line option, 779
    gitlab-project-snippet-note-update command line option, 780
--snippets-access-level
    gitlab-project-create command line option, 580
    gitlab-project-update command line option, 592
--snippets-enabled
    gitlab-project-create command line option, 580
    gitlab-project-update command line

option, 592
gitlab-user-project-create command line option, 820
--sort
gitlab-bulk-import-all-entity-list command line option, 476
gitlab-bulk-import-entity-list command line option, 477
gitlab-bulk-import-list command line option, 475
gitlab-event-list command line option, 485
gitlab-group-billable-member-list command line option, 504
gitlab-group-descendant-group-list command line option, 513
gitlab-group-epic-list command line option, 515
gitlab-group-issue-list command line option, 532
gitlab-group-list command line option, 493
gitlab-group-merge-request-list command line option, 543
gitlab-group-package-list command line option, 548
gitlab-group-project-list command line option, 549
gitlab-group-service-account-access-token-list command line option, 558
gitlab-group-service-account-list command line option, 555
gitlab-group-subgroup-list command line option, 559
gitlab-issue-list command line option, 565
gitlab-merge-request-list command line option, 571
gitlab-project-deployment-list command line option, 626
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-event-list command line option, 632
gitlab-project-fork-list command line option, 643
gitlab-project-issue-list command line option, 654
gitlab-project-job-list command line option, 677
gitlab-project-list command line option, 583
gitlab-project-merge-request-list command line option, 691
gitlab-project-package-list command line option, 724
gitlab-project-pipeline-list command line option, 732
gitlab-project-release-list command line option, 755
gitlab-project-resource-group-list command line option, 761
gitlab-project-service-account-access-token-list command line option, 770
gitlab-project-service-account-list command line option, 768
gitlab-project-tag-list command line option, 783
gitlab-runner-job-list command line option, 794
gitlab-service-account-list command line option, 795
gitlab-shared-project-list command line option, 796
gitlab-starred-project-list command line option, 800
gitlab-user-event-list command line option, 814
gitlab-user-project-list command line option, 821
--source
gitlab-project-pipeline-list command line option, 732
--source-branch
gitlab-group-merge-request-list command line option, 544
gitlab-merge-request-list command line option, 572
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-merge-request-create command line option, 689
gitlab-project-merge-request-list command line option, 691
--source-id
gitlab-user-membership-get command line option, 819
--source-topic-id
gitlab-topic-merge command line option, 803
--squash
gitlab-project-merge-request-create command line option, 689
gitlab-project-merge-request-update command line option, 696
--squash-commit-template
gitlab-project-update command line option, 592

--squash-option
    gitlab-project-create command line option, 580
    gitlab-project-update command line option, 592
    gitlab-user-project-create command line option, 820
--ssl-verify
    gitlab command line option, 464
--starred
    gitlab-group-project-list command line option, 550
    gitlab-project-fork-list command line option, 643
    gitlab-project-list command line option, 583
    gitlab-shared-project-list command line option, 797
    gitlab-starred-project-list command line option, 800
    gitlab-user-project-list command line option, 821
--start-branch
    gitlab-project-file-create command line option, 640
    gitlab-project-file-update command line option, 642
--start-date
    gitlab-group-epic-create command line option, 514
    gitlab-group-epic-update command line option, 516
    gitlab-group-milestone-create command line option, 544
    gitlab-group-milestone-update command line option, 546
    gitlab-project-milestone-create command line option, 719
    gitlab-project-milestone-update command line option, 721
--starts-at
    gitlab-broadcast-message-create command line option, 473
    gitlab-broadcast-message-update command line option, 474
--state
    gitlab-group-issue-list command line option, 532
    gitlab-group-iteration-list command line option, 534
    gitlab-group-merge-request-list command line option, 543
    gitlab-group-milestone-list command line option, 546
    gitlab-group-service-account-access-token-list command line option, 558
    gitlab-issue-list command line option, 565
    gitlab-merge-request-list command line option, 571
    gitlab-project-commit-status-create command line option, 621
    gitlab-project-deployment-merge-request-list command line option, 628
    gitlab-project-issue-list command line option, 654
    gitlab-project-merge-request-list command line option, 691
    gitlab-project-milestone-list command line option, 720
    gitlab-project-service-account-access-token-list command line option, 770
    gitlab-todo-list command line option, 801
    gitlab-user-impersonation-token-list command line option, 817
--state-event
    gitlab-group-milestone-update command line option, 546
    gitlab-project-issue-update command line option, 658
    gitlab-project-merge-request-update command line option, 696
    gitlab-project-milestone-create command line option, 719
    gitlab-project-milestone-update command line option, 721
--states
    gitlab-project-environment-list command line option, 631
--statistics
    gitlab-group-descendant-group-list command line option, 514
    gitlab-group-list command line option, 493
    gitlab-group-subgroup-list command line option, 559
    gitlab-project-fork-list command line option, 643
    gitlab-project-list command line option, 583
    gitlab-starred-project-list command line option, 800
    gitlab-user-project-list command line option, 821
--status
    gitlab-bulk-import-all-entity-list command line option, 476
    gitlab-bulk-import-entity-list command

line option, 477
gitlab-bulk-import-list command line option, 475
gitlab-group-package-list command line option, 548
gitlab-project-deployment-approval command line option, 625
gitlab-project-deployment-create command line option, 625
gitlab-project-deployment-list command line option, 626
gitlab-project-package-list command line option, 724
gitlab-project-pipeline-list command line option, 732
gitlab-runner-all-list command line option, 794
gitlab-runner-job-list command line option, 794
gitlab-runner-list command line option, 792
gitlab-user-list command line option, 808
--strategies
gitlab-project-feature-flag-create command line option, 636
gitlab-project-feature-flag-update command line option, 637
--subgroup-creation-level
gitlab-group-create command line option, 491
gitlab-group-update command line option, 495
--subgroup-events
gitlab-group-hook-create command line option, 527
gitlab-group-hook-update command line option, 529
--submodule
gitlab-project-update-submodule command line option, 592
--sudo
gitlab-application-appearance-get command line option, 466
gitlab-application-appearance-update command line option, 467
gitlab-application-create command line option, 465
gitlab-application-delete command line option, 466
gitlab-application-list command line option, 466
gitlab-application-settings-get command line option, 468
gitlab-application-settings-update command line option, 469
gitlab-application-statistics-get command line option, 472
gitlab-audit-event-get command line option, 472
gitlab-audit-event-list command line option, 472
gitlab-broadcast-message-create command line option, 473
gitlab-broadcast-message-delete command line option, 473
gitlab-broadcast-message-get command line option, 473
gitlab-broadcast-message-list command line option, 474
gitlab-broadcast-message-update command line option, 474
gitlab-bulk-import-all-entity-list command line option, 476
gitlab-bulk-import-create command line option, 475
gitlab-bulk-import-entity-get command line option, 476
gitlab-bulk-import-entity-list command line option, 477
gitlab-bulk-import-get command line option, 475
gitlab-bulk-import-list command line option, 475
gitlab-ci-lint-create command line option, 477
gitlab-current-user-email-create command line option, 478
gitlab-current-user-email-delete command line option, 478
gitlab-current-user-email-get command line option, 479
gitlab-current-user-email-list command line option, 479
gitlab-current-user-get command line option, 478
gitlab-current-user-gpg-key-create command line option, 479
gitlab-current-user-gpg-key-delete command line option, 479
gitlab-current-user-gpg-key-get command line option, 480
gitlab-current-user-gpg-key-list command line option, 480
gitlab-current-user-key-create command line option, 480
gitlab-current-user-key-delete command line option, 481
gitlab-current-user-key-get command

line option, 481
gitlab-current-user-key-list command line option, 481
gitlab-current-user-runner-create command line option, 482
gitlab-current-user-status-get command line option, 482
gitlab-current-user-status-update command line option, 483
gitlab-deploy-key-create command line option, 483
gitlab-deploy-key-list command line option, 483
gitlab-deploy-token-list command line option, 484
gitlab-dockerfile-get command line option, 484
gitlab-dockerfile-list command line option, 484
gitlab-event-list command line option, 485
gitlab-feature-delete command line option, 486
gitlab-feature-list command line option, 486
gitlab-generic-package-download command line option, 487
gitlab-generic-package-upload command line option, 487
gitlab-geo-node-delete command line option, 487
gitlab-geo-node-get command line option, 488
gitlab-geo-node-list command line option, 488
gitlab-geo-node-update command line option, 489
gitlab-gitignore-get command line option, 489
gitlab-gitignore-list command line option, 489
gitlab-gitlabciyml-get command line option, 490
gitlab-gitlabciyml-list command line option, 490
gitlab-group-access-request-approve command line option, 496
gitlab-group-access-request-create command line option, 496
gitlab-group-access-request-delete command line option, 497
gitlab-group-access-request-list command line option, 497
gitlab-group-access-token-create command line option, 497
gitlab-group-access-token-delete command line option, 498
gitlab-group-access-token-get command line option, 498
gitlab-group-access-token-list command line option, 498
gitlab-group-access-token-rotate command line option, 499
gitlab-group-approval-rule-create command line option, 499
gitlab-group-approval-rule-get command line option, 500
gitlab-group-approval-rule-list command line option, 500
gitlab-group-approval-rule-update command line option, 500
gitlab-group-audit-event-get command line option, 501
gitlab-group-audit-event-list command line option, 501
gitlab-group-badge-create command line option, 502
gitlab-group-badge-delete command line option, 502
gitlab-group-badge-get command line option, 502
gitlab-group-badge-list command line option, 502
gitlab-group-badge-render command line option, 503
gitlab-group-badge-update command line option, 503
gitlab-group-billable-member-delete command line option, 503
gitlab-group-billable-member-list command line option, 504
gitlab-group-billable-member-membership-list command line option, 504
gitlab-group-board-create command line option, 505
gitlab-group-board-delete command line option, 505
gitlab-group-board-get command line option, 505
gitlab-group-board-list command line option, 506
gitlab-group-board-list-create command line option, 507
gitlab-group-board-list-delete command line option, 507
gitlab-group-board-list-get command line option, 507
gitlab-group-board-list-list command

line option, 508
gitlab-group-board-list-update command line option, 508
gitlab-group-board-update command line option, 506
gitlab-group-cluster-create command line option, 509
gitlab-group-cluster-delete command line option, 509
gitlab-group-cluster-get command line option, 509
gitlab-group-cluster-list command line option, 510
gitlab-group-cluster-update command line option, 510
gitlab-group-create command line option, 491
gitlab-group-custom-attribute-delete command line option, 511
gitlab-group-custom-attribute-get command line option, 511
gitlab-group-custom-attribute-list command line option, 511
gitlab-group-delete command line option, 492
gitlab-group-deploy-token-create command line option, 512
gitlab-group-deploy-token-delete command line option, 512
gitlab-group-deploy-token-get command line option, 512
gitlab-group-deploy-token-list command line option, 513
gitlab-group-descendant-group-list command line option, 513
gitlab-group-epic-award-emoji-create command line option, 516
gitlab-group-epic-award-emoji-delete command line option, 517
gitlab-group-epic-award-emoji-get command line option, 517
gitlab-group-epic-award-emoji-list command line option, 517
gitlab-group-epic-create command line option, 514
gitlab-group-epic-delete command line option, 515
gitlab-group-epic-discussion-note-create command line option, 518
gitlab-group-epic-discussion-note-delete command line option, 518
gitlab-group-epic-discussion-note-get command line option, 519
gitlab-group-epic-discussion-note-update command line option, 519
gitlab-group-epic-get command line option, 515
gitlab-group-epic-issue-create command line option, 520
gitlab-group-epic-issue-delete command line option, 520
gitlab-group-epic-issue-list command line option, 520
gitlab-group-epic-issue-update command line option, 521
gitlab-group-epic-list command line option, 515
gitlab-group-epic-note-award-emoji-create command line option, 523
gitlab-group-epic-note-award-emoji-delete command line option, 523
gitlab-group-epic-note-award-emoji-get command line option, 524
gitlab-group-epic-note-award-emoji-list command line option, 524
gitlab-group-epic-note-create command line option, 521
gitlab-group-epic-note-delete command line option, 521
gitlab-group-epic-note-get command line option, 522
gitlab-group-epic-note-list command line option, 522
gitlab-group-epic-note-update command line option, 523
gitlab-group-epic-resource-label-event-get command line option, 525
gitlab-group-epic-resource-label-event-list command line option, 525
gitlab-group-epic-update command line option, 516
gitlab-group-export-create command line option, 526
gitlab-group-export-download command line option, 526
gitlab-group-export-get command line option, 526
gitlab-group-get command line option, 492
gitlab-group-hook-create command line option, 527
gitlab-group-hook-delete command line option, 527
gitlab-group-hook-get command line option, 528
gitlab-group-hook-list command line option, 528
gitlab-group-hook-update command line option, 529

gitlab-group-import-get command line option, 529
gitlab-group-invitation-create command line option, 530
gitlab-group-invitation-delete command line option, 530
gitlab-group-invitation-get command line option, 531
gitlab-group-invitation-list command line option, 531
gitlab-group-invitation-update command line option, 531
gitlab-group-issue-list command line option, 532
gitlab-group-issues-statistics-get command line option, 533
gitlab-group-iteration-list command line option, 533
gitlab-group-label-create command line option, 534
gitlab-group-label-delete command line option, 535
gitlab-group-label-get command line option, 535
gitlab-group-label-list command line option, 535
gitlab-group-label-subscribe command line option, 536
gitlab-group-label-unsubscribe command line option, 536
gitlab-group-label-update command line option, 536
gitlab-group-ldap-group-link-create command line option, 537
gitlab-group-ldap-group-link-delete command line option, 537
gitlab-group-ldap-group-link-list command line option, 537
gitlab-group-list command line option, 493
gitlab-group-member-all-get command line option, 540
gitlab-group-member-all-list command line option, 540
gitlab-group-member-create command line option, 538
gitlab-group-member-delete command line option, 538
gitlab-group-member-get command line option, 538
gitlab-group-member-list command line option, 539
gitlab-group-member-role-create command line option, 541
gitlab-group-member-role-delete command line option, 542
gitlab-group-member-role-list command line option, 542
gitlab-group-member-update command line option, 539
gitlab-group-merge-request-list command line option, 543
gitlab-group-milestone-create command line option, 544
gitlab-group-milestone-delete command line option, 545
gitlab-group-milestone-get command line option, 545
gitlab-group-milestone-issues command line option, 545
gitlab-group-milestone-list command line option, 545
gitlab-group-milestone-merge-requests command line option, 546
gitlab-group-milestone-update command line option, 546
gitlab-group-notification-settings-get command line option, 547
gitlab-group-notification-settings-update command line option, 547
gitlab-group-package-list command line option, 548
gitlab-group-project-list command line option, 549
gitlab-group-push-rules-create command line option, 550
gitlab-group-push-rules-delete command line option, 551
gitlab-group-push-rules-get command line option, 551
gitlab-group-push-rules-update command line option, 552
gitlab-group-runner-list command line option, 552
gitlab-group-saml-group-link-create command line option, 553
gitlab-group-saml-group-link-delete command line option, 553
gitlab-group-saml-group-link-get command line option, 554
gitlab-group-saml-group-link-list command line option, 554
gitlab-group-service-account-access-token-create command line option, 556
gitlab-group-service-account-access-token-delete command line option, 557
gitlab-group-service-account-access-token-list command line option, 557

gitlab-group-service-account-access-token-rotate command line option, 558
gitlab-group-service-account-create command line option, 554
gitlab-group-service-account-delete command line option, 555
gitlab-group-service-account-list command line option, 555
gitlab-group-service-account-update command line option, 556
gitlab-group-subgroup-list command line option, 559
gitlab-group-update command line option, 495
gitlab-group-variable-create command line option, 560
gitlab-group-variable-delete command line option, 560
gitlab-group-variable-get command line option, 560
gitlab-group-variable-list command line option, 561
gitlab-group-variable-update command line option, 561
gitlab-group-wiki-create command line option, 562
gitlab-group-wiki-delete command line option, 562
gitlab-group-wiki-get command line option, 562
gitlab-group-wiki-list command line option, 563
gitlab-group-wiki-update command line option, 563
gitlab-hook-create command line option, 564
gitlab-hook-delete command line option, 564
gitlab-hook-get command line option, 564
gitlab-hook-list command line option, 564
gitlab-issue-get command line option, 565
gitlab-issue-list command line option, 565
gitlab-issues-statistics-get command line option, 566
gitlab-key-get command line option, 566
gitlab-ldap-group-list command line option, 567
gitlab-license-get command line option, 567
gitlab-license-list command line option, 568
gitlab-member-role-create command line option, 569
gitlab-member-role-delete command line option, 570
gitlab-member-role-list command line option, 570
gitlab-merge-request-list command line option, 571
gitlab-namespace-get command line option, 573
gitlab-namespace-list command line option, 573
gitlab-notification-settings-get command line option, 573
gitlab-notification-settings-update command line option, 574
gitlab-pages-domain-list command line option, 575
gitlab-personal-access-token-delete command line option, 575
gitlab-personal-access-token-get command line option, 575
gitlab-personal-access-token-list command line option, 575
gitlab-project-access-request-approve command line option, 593
gitlab-project-access-request-create command line option, 593
gitlab-project-access-request-delete command line option, 593
gitlab-project-access-request-list command line option, 594
gitlab-project-access-token-create command line option, 594
gitlab-project-access-token-delete command line option, 595
gitlab-project-access-token-get command line option, 595
gitlab-project-access-token-list command line option, 595
gitlab-project-access-token-rotate command line option, 596
gitlab-project-additional-statistics-get command line option, 596
gitlab-project-approval-get command line option, 596
gitlab-project-approval-rule-create command line option, 597
gitlab-project-approval-rule-delete command line option, 598
gitlab-project-approval-rule-get command line option, 598
gitlab-project-approval-rule-list command line option, 598
gitlab-project-approval-rule-update command line option, 599

gitlab-project-approval-update command line option, 597
gitlab-project-artifact-delete command line option, 599
gitlab-project-artifact-download command line option, 600
gitlab-project-artifact-raw command line option, 600
gitlab-project-audit-event-get command line option, 600
gitlab-project-audit-event-list command line option, 601
gitlab-project-badge-create command line option, 601
gitlab-project-badge-delete command line option, 602
gitlab-project-badge-get command line option, 602
gitlab-project-badge-list command line option, 602
gitlab-project-badge-render command line option, 603
gitlab-project-badge-update command line option, 603
gitlab-project-board-create command line option, 603
gitlab-project-board-delete command line option, 604
gitlab-project-board-get command line option, 604
gitlab-project-board-list command line option, 604
gitlab-project-board-list-create command line option, 605
gitlab-project-board-list-delete command line option, 605
gitlab-project-board-list-get command line option, 606
gitlab-project-board-list-list command line option, 606
gitlab-project-board-list-update command line option, 607
gitlab-project-board-update command line option, 605
gitlab-project-branch-create command line option, 607
gitlab-project-branch-delete command line option, 607
gitlab-project-branch-get command line option, 608
gitlab-project-branch-list command line option, 608
gitlab-project-ci-lint-create command line option, 608
gitlab-project-ci-lint-get command line option, 609
gitlab-project-ci-lint-validate command line option, 609
gitlab-project-cluster-agent-create command line option, 612
gitlab-project-cluster-agent-delete command line option, 612
gitlab-project-cluster-agent-get command line option, 612
gitlab-project-cluster-agent-list command line option, 612
gitlab-project-cluster-create command line option, 610
gitlab-project-cluster-delete command line option, 610
gitlab-project-cluster-get command line option, 610
gitlab-project-cluster-list command line option, 611
gitlab-project-cluster-update command line option, 611
gitlab-project-commit-cherry-pick command line option, 613
gitlab-project-commit-comment-create command line option, 617
gitlab-project-commit-comment-list command line option, 617
gitlab-project-commit-create command line option, 613
gitlab-project-commit-diff command line option, 614
gitlab-project-commit-discussion-create command line option, 618
gitlab-project-commit-discussion-get command line option, 618
gitlab-project-commit-discussion-list command line option, 618
gitlab-project-commit-discussion-note-create command line option, 619
gitlab-project-commit-discussion-note-delete command line option, 619
gitlab-project-commit-discussion-note-get command line option, 620
gitlab-project-commit-discussion-note-update command line option, 620
gitlab-project-commit-get command line option, 614
gitlab-project-commit-list command line option, 614
gitlab-project-commit-merge-requests command line option, 615
gitlab-project-commit-refs command line option, 615

gitlab-project-commit-revert command line option, 616
gitlab-project-commit-sequence command line option, 616
gitlab-project-commit-signature command line option, 616
gitlab-project-commit-status-create command line option, 621
gitlab-project-commit-status-list command line option, 621
gitlab-project-create command line option, 578
gitlab-project-custom-attribute-delete command line option, 622
gitlab-project-custom-attribute-get command line option, 622
gitlab-project-custom-attribute-list command line option, 623
gitlab-project-delete command line option, 581
gitlab-project-deploy-token-create command line option, 623
gitlab-project-deploy-token-delete command line option, 623
gitlab-project-deploy-token-get command line option, 624
gitlab-project-deploy-token-list command line option, 624
gitlab-project-deployment-approval command line option, 625
gitlab-project-deployment-create command line option, 625
gitlab-project-deployment-get command line option, 625
gitlab-project-deployment-list command line option, 626
gitlab-project-deployment-merge-request-list command line option, 627
gitlab-project-deployment-update command line option, 626
gitlab-project-dockerfile-template-get command line option, 629
gitlab-project-dockerfile-template-list command line option, 629
gitlab-project-environment-create command line option, 630
gitlab-project-environment-delete command line option, 630
gitlab-project-environment-get command line option, 630
gitlab-project-environment-list command line option, 631
gitlab-project-environment-stop command line option, 631
gitlab-project-environment-update command line option, 631
gitlab-project-event-list command line option, 632
gitlab-project-export-create command line option, 633
gitlab-project-export-download command line option, 633
gitlab-project-export-get command line option, 633
gitlab-project-external-status-check-create command line option, 634
gitlab-project-external-status-check-delete command line option, 634
gitlab-project-external-status-check-list command line option, 634
gitlab-project-external-status-check-update command line option, 635
gitlab-project-feature-flag-create command line option, 635
gitlab-project-feature-flag-delete command line option, 636
gitlab-project-feature-flag-get command line option, 636
gitlab-project-feature-flag-list command line option, 636
gitlab-project-feature-flag-update command line option, 637
gitlab-project-feature-flag-user-list-create command line option, 637
gitlab-project-feature-flag-user-list-delete command line option, 638
gitlab-project-feature-flag-user-list-get command line option, 638
gitlab-project-feature-flag-user-list-list command line option, 638
gitlab-project-feature-flag-user-list-update command line option, 639
gitlab-project-file-blame command line option, 639
gitlab-project-file-create command line option, 640
gitlab-project-file-delete command line option, 640
gitlab-project-file-get command line option, 641
gitlab-project-file-raw command line option, 641
gitlab-project-file-update command line option, 641
gitlab-project-fork-create command line option, 642
gitlab-project-fork-list command line option, 642

gitlab-project-get command line option, 581
gitlab-project-gitignore-template-get command line option, 643
gitlab-project-gitignore-template-list command line option, 644
gitlab-project-gitlabciyml-template-get command line option, 644
gitlab-project-gitlabciyml-template-list command line option, 645
gitlab-project-hook-create command line option, 645
gitlab-project-hook-delete command line option, 646
gitlab-project-hook-get command line option, 646
gitlab-project-hook-list command line option, 647
gitlab-project-hook-update command line option, 647
gitlab-project-import-get command line option, 648
gitlab-project-integration-available command line option, 648
gitlab-project-integration-delete command line option, 649
gitlab-project-integration-get command line option, 649
gitlab-project-integration-list command line option, 649
gitlab-project-integration-update command line option, 649
gitlab-project-invitation-create command line option, 650
gitlab-project-invitation-delete command line option, 650
gitlab-project-invitation-get command line option, 651
gitlab-project-invitation-list command line option, 651
gitlab-project-invitation-update command line option, 651
gitlab-project-issue-add-spent-time command line option, 652
gitlab-project-issue-award-emoji-create command line option, 659
gitlab-project-issue-award-emoji-delete command line option, 659
gitlab-project-issue-award-emoji-get command line option, 660
gitlab-project-issue-award-emoji-list command line option, 660
gitlab-project-issue-closed-by command line option, 652
gitlab-project-issue-create command line option, 653
gitlab-project-issue-delete command line option, 653
gitlab-project-issue-discussion-create command line option, 661
gitlab-project-issue-discussion-get command line option, 661
gitlab-project-issue-discussion-list command line option, 661
gitlab-project-issue-discussion-note-create command line option, 662
gitlab-project-issue-discussion-note-delete command line option, 662
gitlab-project-issue-discussion-note-get command line option, 663
gitlab-project-issue-discussion-note-list command line option, 663
gitlab-project-issue-discussion-note-update command line option, 664
gitlab-project-issue-get command line option, 653
gitlab-project-issue-link-create command line option, 664
gitlab-project-issue-link-delete command line option, 665
gitlab-project-issue-link-list command line option, 665
gitlab-project-issue-list command line option, 654
gitlab-project-issue-move command line option, 655
gitlab-project-issue-note-award-emoji-create command line option, 667
gitlab-project-issue-note-award-emoji-delete command line option, 668
gitlab-project-issue-note-award-emoji-get command line option, 668
gitlab-project-issue-note-award-emoji-list command line option, 668
gitlab-project-issue-note-create command line option, 665
gitlab-project-issue-note-delete command line option, 666
gitlab-project-issue-note-get command line option, 666
gitlab-project-issue-note-list command line option, 666
gitlab-project-issue-note-update command line option, 667
gitlab-project-issue-participants command line option, 655
gitlab-project-issue-related-merge-requests command line option, 655

gitlab-project-issue-reorder command line option, 656
gitlab-project-issue-reset-spent-time command line option, 656
gitlab-project-issue-reset-time-estimate command line option, 656
gitlab-project-issue-resource-iteration-event-get command line option, 669
gitlab-project-issue-resource-iteration-event-list command line option, 669
gitlab-project-issue-resource-label-event-get command line option, 670
gitlab-project-issue-resource-label-event-list command line option, 670
gitlab-project-issue-resource-milestone-event-get command line option, 671
gitlab-project-issue-resource-milestone-event-list command line option, 671
gitlab-project-issue-resource-state-event-get command line option, 672
gitlab-project-issue-resource-state-event-list command line option, 672
gitlab-project-issue-resource-weight-event-get command line option, 673
gitlab-project-issue-resource-weight-event-list command line option, 673
gitlab-project-issue-subscribe command line option, 656
gitlab-project-issue-template-get command line option, 674
gitlab-project-issue-template-list command line option, 674
gitlab-project-issue-time-estimate command line option, 657
gitlab-project-issue-time-stats command line option, 657
gitlab-project-issue-todo command line option, 657
gitlab-project-issue-unsubscribe command line option, 658
gitlab-project-issue-update command line option, 658
gitlab-project-issue-user-agent-detail command line option, 659
gitlab-project-issues-statistics-get command line option, 675
gitlab-project-job-artifact command line option, 675
gitlab-project-job-artifacts command line option, 675
gitlab-project-job-cancel command line option, 676
gitlab-project-job-delete-artifacts command line option, 676
gitlab-project-job-erase command line option, 676
gitlab-project-job-get command line option, 676
gitlab-project-job-keep-artifacts command line option, 677
gitlab-project-job-list command line option, 677
gitlab-project-job-play command line option, 677
gitlab-project-job-retry command line option, 678
gitlab-project-job-token-scope-get command line option, 678
gitlab-project-job-token-scope-update command line option, 678
gitlab-project-job-trace command line option, 678
gitlab-project-key-create command line option, 679
gitlab-project-key-delete command line option, 679
gitlab-project-key-enable command line option, 679
gitlab-project-key-get command line option, 680
gitlab-project-key-list command line option, 680
gitlab-project-key-update command line option, 680
gitlab-project-label-create command line option, 681
gitlab-project-label-delete command line option, 681
gitlab-project-label-get command line option, 682
gitlab-project-label-list command line option, 682
gitlab-project-label-subscribe command line option, 682
gitlab-project-label-unsubscribe command line option, 682
gitlab-project-label-update command line option, 683
gitlab-project-license-template-get command line option, 683
gitlab-project-license-template-list command line option, 684
gitlab-project-list command line option, 582
gitlab-project-member-all-get command line option, 686
gitlab-project-member-all-list command line option, 686

gitlab-project-member-create command line option, 684
gitlab-project-member-delete command line option, 684
gitlab-project-member-get command line option, 685
gitlab-project-member-list command line option, 685
gitlab-project-member-update command line option, 685
gitlab-project-merge-request-add-spent-time command line option, 687
gitlab-project-merge-request-approval-get command line option, 697
gitlab-project-merge-request-approval-rule-create command line option, 698
gitlab-project-merge-request-approval-rule-delete command line option, 698
gitlab-project-merge-request-approval-rule-get command line option, 698
gitlab-project-merge-request-approval-rule-list command line option, 699
gitlab-project-merge-request-approval-rule-update command line option, 699
gitlab-project-merge-request-approval-state-get command line option, 700
gitlab-project-merge-request-approval-update command line option, 697
gitlab-project-merge-request-approve command line option, 687
gitlab-project-merge-request-award-emoji-create command line option, 700
gitlab-project-merge-request-award-emoji-delete command line option, 701
gitlab-project-merge-request-award-emoji-get command line option, 701
gitlab-project-merge-request-award-emoji-list command line option, 701
gitlab-project-merge-request-cancel-merge-when-pipeline-succeeds command line option, 687
gitlab-project-merge-request-changes command line option, 688
gitlab-project-merge-request-closes-issues command line option, 688
gitlab-project-merge-request-commits command line option, 688
gitlab-project-merge-request-create command line option, 689
gitlab-project-merge-request-delete command line option, 690
gitlab-project-merge-request-diff-get command line option, 702
gitlab-project-merge-request-diff-list command line option, 702
gitlab-project-merge-request-discussion-create command line option, 703
gitlab-project-merge-request-discussion-get command line option, 703
gitlab-project-merge-request-discussion-list command line option, 704
gitlab-project-merge-request-discussion-note-create command line option, 705
gitlab-project-merge-request-discussion-note-delete command line option, 705
gitlab-project-merge-request-discussion-note-get command line option, 705
gitlab-project-merge-request-discussion-note-list command line option, 706
gitlab-project-merge-request-discussion-note-update command line option, 706
gitlab-project-merge-request-discussion-update command line option, 704
gitlab-project-merge-request-draft-note-create command line option, 707
gitlab-project-merge-request-draft-note-delete command line option, 707
gitlab-project-merge-request-draft-note-get command line option, 708
gitlab-project-merge-request-draft-note-list command line option, 708
gitlab-project-merge-request-draft-note-update command line option, 708
gitlab-project-merge-request-get command line option, 690
gitlab-project-merge-request-list command line option, 691
gitlab-project-merge-request-merge command line option, 692
gitlab-project-merge-request-merge-ref command line option, 692
gitlab-project-merge-request-note-award-emoji-create command line option, 711
gitlab-project-merge-request-note-award-emoji-delete command line option, 711
gitlab-project-merge-request-note-award-emoji-get command line option, 711
gitlab-project-merge-request-note-award-emoji-list command line option, 712
gitlab-project-merge-request-note-create command line option, 709
gitlab-project-merge-request-note-delete command line option, 709
gitlab-project-merge-request-note-get command line option, 710
gitlab-project-merge-request-note-list command line option, 710
gitlab-project-merge-request-note-update command line option, 710

gitlab-project-merge-request-participants command line option, 692
gitlab-project-merge-request-pipeline-create command line option, 713
gitlab-project-merge-request-pipeline-list command line option, 713
gitlab-project-merge-request-rebase command line option, 693
gitlab-project-merge-request-related-issues command line option, 693
gitlab-project-merge-request-reset-approvals command line option, 693
gitlab-project-merge-request-reset-spent-time command line option, 694
gitlab-project-merge-request-reset-time-estimate command line option, 694
gitlab-project-merge-request-resource-label-event-get command line option, 713
gitlab-project-merge-request-resource-label-event-list command line option, 714
gitlab-project-merge-request-resource-milestone-event-get command line option, 714
gitlab-project-merge-request-resource-milestone-event-list command line option, 715
gitlab-project-merge-request-resource-state-event-get command line option, 715
gitlab-project-merge-request-resource-state-event-list command line option, 715
gitlab-project-merge-request-reviewer-detail-list command line option, 716
gitlab-project-merge-request-status-check-list command line option, 717
gitlab-project-merge-request-subscribe command line option, 694
gitlab-project-merge-request-template-get command line option, 717
gitlab-project-merge-request-template-list command line option, 718
gitlab-project-merge-request-time-estimate command line option, 694
gitlab-project-merge-request-time-stats command line option, 695
gitlab-project-merge-request-todo command line option, 695
gitlab-project-merge-request-unapprove command line option, 695
gitlab-project-merge-request-unsubscribe command line option, 695
gitlab-project-merge-request-update command line option, 696
gitlab-project-merge-train-list command line option, 718
gitlab-project-milestone-create command line option, 719
gitlab-project-milestone-delete command line option, 719
gitlab-project-milestone-get command line option, 719
gitlab-project-milestone-issues command line option, 720
gitlab-project-milestone-list command line option, 720
gitlab-project-milestone-merge-requests command line option, 721
gitlab-project-milestone-update command line option, 721
gitlab-project-note-get command line option, 721
gitlab-project-note-list command line option, 722
gitlab-project-notification-settings-get command line option, 722
gitlab-project-notification-settings-update command line option, 723
gitlab-project-package-delete command line option, 723
gitlab-project-package-file-delete command line option, 725
gitlab-project-package-file-list command line option, 725
gitlab-project-package-get command line option, 724
gitlab-project-package-list command line option, 724
gitlab-project-package-pipeline-list command line option, 726
gitlab-project-package-protection-rule-create command line option, 726
gitlab-project-package-protection-rule-delete command line option, 727
gitlab-project-package-protection-rule-list command line option, 727
gitlab-project-package-protection-rule-update command line option, 728
gitlab-project-pages-delete command line option, 728
gitlab-project-pages-domain-create command line option, 729
gitlab-project-pages-domain-delete command line option, 729
gitlab-project-pages-domain-get command line option, 730
gitlab-project-pages-domain-list command line option, 730
gitlab-project-pages-domain-update command line option, 730
gitlab-project-pages-get command line option, 728

gitlab-project-pages-update command line option, 729
gitlab-project-pipeline-bridge-list command line option, 733
gitlab-project-pipeline-cancel command line option, 731
gitlab-project-pipeline-create command line option, 731
gitlab-project-pipeline-delete command line option, 731
gitlab-project-pipeline-get command line option, 732
gitlab-project-pipeline-job-list command line option, 734
gitlab-project-pipeline-list command line option, 732
gitlab-project-pipeline-retry command line option, 733
gitlab-project-pipeline-schedule-create command line option, 734
gitlab-project-pipeline-schedule-delete command line option, 735
gitlab-project-pipeline-schedule-get command line option, 735
gitlab-project-pipeline-schedule-list command line option, 735
gitlab-project-pipeline-schedule-pipeline-list command line option, 737
gitlab-project-pipeline-schedule-play command line option, 736
gitlab-project-pipeline-schedule-take-ownership command line option, 736
gitlab-project-pipeline-schedule-update command line option, 736
gitlab-project-pipeline-schedule-variable-create command line option, 738
gitlab-project-pipeline-schedule-variable-delete command line option, 738
gitlab-project-pipeline-schedule-variable-update command line option, 738
gitlab-project-pipeline-test-report-get command line option, 739
gitlab-project-pipeline-test-report-summary-get command line option, 739
gitlab-project-pipeline-variable-list command line option, 740
gitlab-project-protected-branch-create command line option, 740
gitlab-project-protected-branch-delete command line option, 741
gitlab-project-protected-branch-get command line option, 741
gitlab-project-protected-branch-list command line option, 742
gitlab-project-protected-branch-update command line option, 742
gitlab-project-protected-environment-create command line option, 742
gitlab-project-protected-environment-delete command line option, 743
gitlab-project-protected-environment-get command line option, 743
gitlab-project-protected-environment-list command line option, 743
gitlab-project-protected-tag-create command line option, 744
gitlab-project-protected-tag-delete command line option, 744
gitlab-project-protected-tag-get command line option, 744
gitlab-project-protected-tag-list command line option, 745
gitlab-project-pull-mirror-create command line option, 745
gitlab-project-pull-mirror-get command line option, 745
gitlab-project-pull-mirror-start command line option, 746
gitlab-project-pull-mirror-update command line option, 746
gitlab-project-push-rules-create command line option, 747
gitlab-project-push-rules-delete command line option, 747
gitlab-project-push-rules-get command line option, 747
gitlab-project-push-rules-update command line option, 748
gitlab-project-registry-protection-rule-create command line option, 749
gitlab-project-registry-protection-rule-list command line option, 749
gitlab-project-registry-protection-rule-update command line option, 750
gitlab-project-registry-repository-delete command line option, 750
gitlab-project-registry-repository-list command line option, 750
gitlab-project-registry-repository-protection-rule-cre command line option, 751
gitlab-project-registry-repository-protection-rule-del command line option, 751
gitlab-project-registry-repository-protection-rule-lis command line option, 752
gitlab-project-registry-repository-protection-rule-upd command line option, 752
gitlab-project-registry-tag-delete command line option, 753

gitlab-project-registry-tag-delete-in-bulk command line option, 753
gitlab-project-registry-tag-get command line option, 753
gitlab-project-registry-tag-list command line option, 754
gitlab-project-release-create command line option, 754
gitlab-project-release-delete command line option, 755
gitlab-project-release-get command line option, 755
gitlab-project-release-link-create command line option, 757
gitlab-project-release-link-delete command line option, 757
gitlab-project-release-link-get command line option, 757
gitlab-project-release-link-list command line option, 758
gitlab-project-release-link-update command line option, 758
gitlab-project-release-list command line option, 755
gitlab-project-release-update command line option, 756
gitlab-project-remote-mirror-create command line option, 759
gitlab-project-remote-mirror-delete command line option, 759
gitlab-project-remote-mirror-list command line option, 759
gitlab-project-remote-mirror-sync command line option, 760
gitlab-project-remote-mirror-update command line option, 760
gitlab-project-resource-group-get command line option, 761
gitlab-project-resource-group-list command line option, 761
gitlab-project-resource-group-upcoming-job-list command line option, 762
gitlab-project-resource-group-update command line option, 761
gitlab-project-runner-create command line option, 763
gitlab-project-runner-delete command line option, 763
gitlab-project-runner-list command line option, 763
gitlab-project-secure-file-create command line option, 764
gitlab-project-secure-file-delete command line option, 764
gitlab-project-secure-file-download command line option, 764
gitlab-project-secure-file-get command line option, 765
gitlab-project-secure-file-list command line option, 765
gitlab-project-service-account-access-token-create command line option, 769
gitlab-project-service-account-access-token-delete command line option, 769
gitlab-project-service-account-access-token-list command line option, 770
gitlab-project-service-account-access-token-rotate command line option, 771
gitlab-project-service-account-create command line option, 767
gitlab-project-service-account-delete command line option, 767
gitlab-project-service-account-list command line option, 768
gitlab-project-service-account-update command line option, 768
gitlab-project-service-available command line option, 765
gitlab-project-service-delete command line option, 766
gitlab-project-service-get command line option, 766
gitlab-project-service-list command line option, 766
gitlab-project-service-update command line option, 767
gitlab-project-snippet-award-emoji-create command line option, 774
gitlab-project-snippet-award-emoji-delete command line option, 774
gitlab-project-snippet-award-emoji-get command line option, 774
gitlab-project-snippet-award-emoji-list command line option, 775
gitlab-project-snippet-content command line option, 771
gitlab-project-snippet-create command line option, 771
gitlab-project-snippet-delete command line option, 772
gitlab-project-snippet-discussion-create command line option, 775
gitlab-project-snippet-discussion-get command line option, 776
gitlab-project-snippet-discussion-list command line option, 776
gitlab-project-snippet-discussion-note-create command line option, 777

gitlab-project-snippet-discussion-note-delete command line option, 777
gitlab-project-snippet-discussion-note-get command line option, 777
gitlab-project-snippet-discussion-note-update command line option, 778
gitlab-project-snippet-get command line option, 772
gitlab-project-snippet-list command line option, 772
gitlab-project-snippet-note-award-emoji-create command line option, 780
gitlab-project-snippet-note-award-emoji-delete command line option, 781
gitlab-project-snippet-note-award-emoji-get command line option, 781
gitlab-project-snippet-note-award-emoji-list command line option, 781
gitlab-project-snippet-note-create command line option, 778
gitlab-project-snippet-note-delete command line option, 779
gitlab-project-snippet-note-get command line option, 779
gitlab-project-snippet-note-list command line option, 779
gitlab-project-snippet-note-update command line option, 780
gitlab-project-snippet-update command line option, 773
gitlab-project-snippet-user-agent-detail command line option, 773
gitlab-project-storage-get command line option, 782
gitlab-project-tag-create command line option, 782
gitlab-project-tag-delete command line option, 783
gitlab-project-tag-get command line option, 783
gitlab-project-tag-list command line option, 783
gitlab-project-trigger-create command line option, 784
gitlab-project-trigger-delete command line option, 784
gitlab-project-trigger-get command line option, 784
gitlab-project-trigger-list command line option, 785
gitlab-project-trigger-update command line option, 785
gitlab-project-update command line option, 589
gitlab-project-user-list command line option, 786
gitlab-project-variable-create command line option, 786
gitlab-project-variable-delete command line option, 787
gitlab-project-variable-get command line option, 787
gitlab-project-variable-list command line option, 787
gitlab-project-variable-update command line option, 788
gitlab-project-wiki-create command line option, 789
gitlab-project-wiki-delete command line option, 789
gitlab-project-wiki-get command line option, 789
gitlab-project-wiki-list command line option, 789
gitlab-project-wiki-update command line option, 790
gitlab-project-wiki-upload command line option, 790
gitlab-registry-repository-get command line option, 791
gitlab-runner-all-list command line option, 794
gitlab-runner-create command line option, 791
gitlab-runner-delete command line option, 792
gitlab-runner-get command line option, 792
gitlab-runner-job-list command line option, 794
gitlab-runner-list command line option, 792
gitlab-runner-update command line option, 793
gitlab-service-account-create command line option, 795
gitlab-service-account-list command line option, 795
gitlab-service-account-update command line option, 796
gitlab-shared-project-list command line option, 796
gitlab-snippet-create command line option, 797
gitlab-snippet-delete command line option, 798
gitlab-snippet-get command line option, 798

gitlab-snippet-list command line option, 798
gitlab-snippet-update command line option, 799
gitlab-starred-project-list command line option, 800
gitlab-todo-delete command line option, 801
gitlab-todo-list command line option, 801
gitlab-topic-create command line option, 802
gitlab-topic-delete command line option, 803
gitlab-topic-get command line option, 803
gitlab-topic-list command line option, 803
gitlab-topic-update command line option, 804
gitlab-user-activities-list command line option, 810
gitlab-user-contributed-project-list command line option, 811
gitlab-user-create command line option, 805
gitlab-user-custom-attribute-delete command line option, 811
gitlab-user-custom-attribute-get command line option, 812
gitlab-user-custom-attribute-list command line option, 812
gitlab-user-delete command line option, 806
gitlab-user-email-create command line option, 812
gitlab-user-email-delete command line option, 813
gitlab-user-email-get command line option, 813
gitlab-user-email-list command line option, 813
gitlab-user-event-list command line option, 814
gitlab-user-get command line option, 807
gitlab-user-gpg-key-create command line option, 815
gitlab-user-gpg-key-delete command line option, 815
gitlab-user-gpg-key-get command line option, 815
gitlab-user-gpg-key-list command line option, 815
gitlab-user-impersonation-token-create command line option, 816
gitlab-user-impersonation-token-delete command line option, 816
gitlab-user-impersonation-token-get command line option, 816
gitlab-user-impersonation-token-list command line option, 817
gitlab-user-key-create command line option, 817
gitlab-user-key-delete command line option, 817
gitlab-user-key-get command line option, 818
gitlab-user-key-list command line option, 818
gitlab-user-list command line option, 807
gitlab-user-membership-get command line option, 818
gitlab-user-membership-list command line option, 819
gitlab-user-personal-access-token-create command line option, 819
gitlab-user-project-create command line option, 820
gitlab-user-project-list command line option, 821
gitlab-user-status-get command line option, 822
gitlab-user-update command line option, 809
gitlab-variable-create command line option, 823
gitlab-variable-delete command line option, 823
gitlab-variable-get command line option, 823
gitlab-variable-list command line option, 823
gitlab-variable-update command line option, 824
--suggestion-commit-message
gitlab-project-update command line option, 592
--tag
gitlab-project-deployment-create command line option, 625
--tag-list
gitlab-current-user-runner-create command line option, 482
gitlab-group-runner-list command line option, 553
gitlab-project-create command line option, 580
gitlab-project-runner-list command line option, 763
gitlab-project-update command line

option, 592
gitlab-runner-all-list command line option, 794
gitlab-runner-create command line option, 791
gitlab-runner-list command line option, 792
gitlab-runner-update command line option, 793
--tag-name
gitlab-project-release-create command line option, 754
gitlab-project-release-delete command line option, 755
gitlab-project-release-get command line option, 755
gitlab-project-release-link-create command line option, 757
gitlab-project-release-link-delete command line option, 757
gitlab-project-release-link-get command line option, 757
gitlab-project-release-link-list command line option, 758
gitlab-project-release-link-update command line option, 758
gitlab-project-release-update command line option, 756
gitlab-project-tag-create command line option, 783
--tag-push-events
gitlab-group-hook-create command line option, 527
gitlab-group-hook-update command line option, 529
gitlab-project-hook-create command line option, 646
gitlab-project-hook-update command line option, 647
--target-access-levels
gitlab-broadcast-message-create command line option, 473
gitlab-broadcast-message-update command line option, 474
--target-branch
gitlab-group-merge-request-list command line option, 544
gitlab-merge-request-list command line option, 572
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-merge-request-create command line option, 689
gitlab-project-merge-request-list command line option, 691
gitlab-project-merge-request-update command line option, 696
--target-issue-iid
gitlab-project-issue-link-create command line option, 664
--target-project-id
gitlab-project-issue-link-create command line option, 664
gitlab-project-merge-request-create command line option, 689
--target-topic-id
gitlab-topic-merge command line option, 803
--target-type
gitlab-event-list command line option, 485
gitlab-project-event-list command line option, 632
gitlab-user-event-list command line option, 814
--target-url
gitlab-project-commit-status-create command line option, 621
--tasks-project-id
gitlab-group-invitation-create command line option, 530
gitlab-project-invitation-create command line option, 650
--tasks-to-be-done
gitlab-group-invitation-create command line option, 530
gitlab-group-member-create command line option, 538
gitlab-project-invitation-create command line option, 650
gitlab-project-member-create command line option, 684
--template-name
gitlab-project-create command line option, 580
--template-project-id
gitlab-project-create command line option, 580
--terminal-max-session-time
gitlab-application-settings-update command line option, 471
--terms
gitlab-application-settings-update command line option, 471
--theme-id
gitlab-user-create command line option, 806
gitlab-user-update command line option,

810
--timeout
gitlab command line option, 465
--title
gitlab-application-appearance-update command line option, 467
gitlab-current-user-key-create command line option, 480
gitlab-deploy-key-create command line option, 483
gitlab-group-epic-create command line option, 514
gitlab-group-epic-update command line option, 516
gitlab-group-milestone-create command line option, 544
gitlab-group-milestone-update command line option, 546
gitlab-group-wiki-create command line option, 562
gitlab-group-wiki-update command line option, 563
gitlab-project-issue-create command line option, 653
gitlab-project-issue-update command line option, 658
gitlab-project-key-create command line option, 679
gitlab-project-key-update command line option, 680
gitlab-project-merge-request-create command line option, 689
gitlab-project-merge-request-update command line option, 696
gitlab-project-milestone-create command line option, 719
gitlab-project-milestone-update command line option, 721
gitlab-project-snippet-create command line option, 771
gitlab-project-snippet-update command line option, 773
gitlab-project-wiki-create command line option, 789
gitlab-project-wiki-update command line option, 790
gitlab-snippet-create command line option, 797
gitlab-snippet-update command line option, 799
gitlab-topic-create command line option, 802
gitlab-user-key-create command line option, 817
--to
gitlab-project-repository-compare command line option, 584
--to-namespace
gitlab-project-transfer command line option, 586
--to-project-id
gitlab-project-issue-move command line option, 655
--token
gitlab-group-hook-create command line option, 527
gitlab-group-hook-update command line option, 529
gitlab-project-hook-create command line option, 646
gitlab-project-hook-update command line option, 648
gitlab-project-trigger-pipeline command line option, 587
gitlab-runner-create command line option, 791
gitlab-runner-verify command line option, 793
--top-level-only
gitlab-group-list command line option, 493
--topic
gitlab-project-list command line option, 583
--topics
gitlab-project-create command line option, 580
gitlab-project-update command line option, 592
--trailers
gitlab-project-commit-list command line option, 615
--twitter
gitlab-user-create command line option, 805
gitlab-user-update command line option, 809
--two-factor
gitlab-user-list command line option, 808
--two-factor-grace-period
gitlab-group-create command line option, 491
gitlab-group-update command line option, 495
--type
gitlab-project-commit-refs command line option, 615
gitlab-runner-all-list command line

option, 794
gitlab-runner-list command line option, 792
gitlab-todo-list command line option, 801
gitlab-user-membership-list command line option, 819
--unprotect-access-level
gitlab-project-protected-branch-create command line option, 741
--until
gitlab-project-commit-list command line option, 614
--updated-after
gitlab-group-issue-list command line option, 532
gitlab-group-iteration-list command line option, 534
gitlab-group-merge-request-list command line option, 543
gitlab-issue-list command line option, 566
gitlab-merge-request-list command line option, 571
gitlab-project-deployment-list command line option, 626
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-issue-list command line option, 654
gitlab-project-merge-request-list command line option, 691
--updated-at
gitlab-application-settings-update command line option, 470
gitlab-project-issue-update command line option, 658
--updated-before
gitlab-group-issue-list command line option, 533
gitlab-group-iteration-list command line option, 534
gitlab-group-merge-request-list command line option, 543
gitlab-issue-list command line option, 566
gitlab-merge-request-list command line option, 571
gitlab-project-deployment-list command line option, 626
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-issue-list command line option, 654
gitlab-project-merge-request-list command line option, 691
--url
gitlab-geo-node-update command line option, 489
gitlab-group-hook-create command line option, 527
gitlab-group-hook-update command line option, 529
gitlab-hook-create command line option, 564
gitlab-project-hook-create command line option, 645
gitlab-project-hook-update command line option, 647
gitlab-project-pull-mirror-update command line option, 746
gitlab-project-release-link-create command line option, 757
gitlab-project-release-link-update command line option, 758
gitlab-project-remote-mirror-create command line option, 759
--use-custom-template
gitlab-project-create command line option, 580
--user-agent
gitlab command line option, 465
--user-id
gitlab-group-billable-member-membership-list command line option, 505
gitlab-group-invitation-create command line option, 530
gitlab-group-member-create command line option, 538
gitlab-group-service-account-access-token-create command line option, 556
gitlab-group-service-account-access-token-delete command line option, 557
gitlab-group-service-account-access-token-list command line option, 557
gitlab-group-service-account-access-token-rotate command line option, 558
gitlab-personal-access-token-list command line option, 575
gitlab-project-invitation-create command line option, 650
gitlab-project-member-create command line option, 684
gitlab-project-service-account-access-token-create command line option, 769
gitlab-project-service-account-access-token-delete command line option, 769
gitlab-project-service-account-access-token-list command line option, 770

gitlab-project-service-account-access-token-rotate command line option, 771
gitlab-starred-project-list command line option, 800
gitlab-user-contributed-project-list command line option, 811
gitlab-user-custom-attribute-delete command line option, 811
gitlab-user-custom-attribute-get command line option, 812
gitlab-user-custom-attribute-list command line option, 812
gitlab-user-email-create command line option, 812
gitlab-user-email-delete command line option, 813
gitlab-user-email-get command line option, 813
gitlab-user-email-list command line option, 813
gitlab-user-event-list command line option, 814
gitlab-user-gpg-key-create command line option, 815
gitlab-user-gpg-key-delete command line option, 815
gitlab-user-gpg-key-get command line option, 815
gitlab-user-gpg-key-list command line option, 815
gitlab-user-impersonation-token-create command line option, 816
gitlab-user-impersonation-token-delete command line option, 816
gitlab-user-impersonation-token-get command line option, 816
gitlab-user-impersonation-token-list command line option, 817
gitlab-user-key-create command line option, 817
gitlab-user-key-delete command line option, 818
gitlab-user-key-get command line option, 818
gitlab-user-key-list command line option, 818
gitlab-user-membership-get command line option, 819
gitlab-user-membership-list command line option, 819
gitlab-user-personal-access-token-create command line option, 819
gitlab-user-project-create command line option, 820
gitlab-user-project-list command line option, 821
gitlab-user-status-get command line option, 822
--user-ids
gitlab-group-approval-rule-create command line option, 499
gitlab-project-approval-rule-create command line option, 598
gitlab-project-merge-request-approval-rule-create command line option, 698
gitlab-project-merge-request-approval-rule-update command line option, 700
--user-oauth-applications
gitlab-application-settings-update command line option, 470
--user-show-add-ssh-key-message
gitlab-application-settings-update command line option, 471
--user-xids
gitlab-project-feature-flag-user-list-create command line option, 638
gitlab-project-feature-flag-user-list-update command line option, 639
--username
gitlab-group-deploy-token-create command line option, 512
gitlab-group-member-create command line option, 538
gitlab-group-service-account-create command line option, 555
gitlab-group-service-account-update command line option, 556
gitlab-project-deploy-token-create command line option, 623
gitlab-project-member-create command line option, 684
gitlab-project-pipeline-list command line option, 732
gitlab-project-service-account-create command line option, 767
gitlab-project-service-account-update command line option, 768
gitlab-service-account-create command line option, 795
gitlab-service-account-update command line option, 796
gitlab-user-create command line option, 805
gitlab-user-list command line option, 807
gitlab-user-update command line option, 809
--usernames
gitlab-project-approval-rule-create

command line option, 598
gitlab-project-merge-request-approval-rule-create command line option, 698
gitlab-project-merge-request-approval-rule-update command line option, 700
--value
gitlab-group-variable-create command line option, 560
gitlab-group-variable-update command line option, 561
gitlab-project-pipeline-schedule-variable-create command line option, 738
gitlab-project-pipeline-schedule-variable-update command line option, 739
gitlab-project-variable-create command line option, 786
gitlab-project-variable-update command line option, 788
gitlab-variable-create command line option, 823
gitlab-variable-update command line option, 824
--variable-type
gitlab-group-variable-create command line option, 560
gitlab-group-variable-update command line option, 561
gitlab-project-variable-create command line option, 787
gitlab-project-variable-update command line option, 788
gitlab-variable-create command line option, 823
gitlab-variable-update command line option, 824
--verbose
gitlab command line option, 464
--version
gitlab command line option, 464
gitlab-project-feature-flag-create command line option, 636
--view
gitlab-group-merge-request-list command line option, 543
gitlab-merge-request-list command line option, 571
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-merge-request-list command line option, 691
--visibility
gitlab-group-create command line option, 491
gitlab-group-project-list command line option, 549
gitlab-group-update command line option, 495
gitlab-project-create command line option, 580
gitlab-project-fork-list command line option, 643
gitlab-project-list command line option, 583
gitlab-project-snippet-create command line option, 772
gitlab-project-snippet-update command line option, 773
gitlab-project-update command line option, 592
gitlab-shared-project-list command line option, 796
gitlab-snippet-create command line option, 798
gitlab-snippet-update command line option, 799
gitlab-starred-project-list command line option, 800
gitlab-user-project-create command line option, 820
gitlab-user-project-list command line option, 821
--wall-enabled
gitlab-user-project-create command line option, 820
--website-url
gitlab-user-create command line option, 806
gitlab-user-update command line option, 810
--wiki
gitlab-project-snapshot command line option, 586
--wiki-access-level
gitlab-project-create command line option, 580
gitlab-project-update command line option, 592
--wiki-checksum-failed
gitlab-project-list command line option, 583
gitlab-user-project-list command line option, 822
--wiki-enabled
gitlab-project-create command line option, 580
gitlab-project-update command line option, 592
gitlab-user-project-create command line

option, 820
--wiki-events
gitlab-project-hook-update command line option, 648
--wiki-page-events
gitlab-group-hook-create command line option, 527
gitlab-group-hook-update command line option, 529
gitlab-project-hook-create command line option, 646
--wip
gitlab-group-merge-request-list command line option, 544
gitlab-merge-request-list command line option, 572
gitlab-project-deployment-merge-request-list command line option, 628
gitlab-project-merge-request-list command line option, 691
--with-content
gitlab-group-wiki-list command line option, 563
gitlab-project-wiki-list command line option, 789
--with-custom-attributes
gitlab-group-descendant-group-list command line option, 514
gitlab-group-list command line option, 493
gitlab-group-project-list command line option, 550
gitlab-group-subgroup-list command line option, 559
gitlab-project-fork-list command line option, 643
gitlab-project-list command line option, 583
gitlab-shared-project-list command line option, 797
gitlab-starred-project-list command line option, 800
gitlab-user-project-list command line option, 821
--with-issues-enabled
gitlab-group-project-list command line option, 550
gitlab-project-fork-list command line option, 643
gitlab-project-list command line option, 583
gitlab-shared-project-list command line option, 797
gitlab-starred-project-list command line option, 800
gitlab-user-project-list command line option, 821
--with-labels-details
gitlab-merge-request-list command line option, 571
gitlab-project-deployment-merge-request-list command line option, 628
--with-merge-requests-enabled
gitlab-group-project-list command line option, 550
gitlab-project-fork-list command line option, 643
gitlab-project-list command line option, 583
gitlab-shared-project-list command line option, 797
gitlab-starred-project-list command line option, 800
gitlab-user-project-list command line option, 821
--with-merge-status-recheck
gitlab-merge-request-list command line option, 571
gitlab-project-deployment-merge-request-list command line option, 628
--with-programming-language
gitlab-project-list command line option, 583
gitlab-user-project-list command line option, 822
--with-security-reports
gitlab-group-project-list command line option, 550
--with-shared
gitlab-group-project-list command line option, 550
--with-stats
gitlab-project-commit-list command line option, 615
--without-project-bots
gitlab-user-list command line option, 808
--without-projects
gitlab-user-list command line option, 808
--yaml-errors
gitlab-project-pipeline-list command line option, 732
-c
gitlab command line option, 464
-d
gitlab command line option, 464
-f
gitlab command line option, 464
-g

gitlab command line option, 464
-h
gitlab command line option, 464
gitlab-application command line option, 465
gitlab-application-appearance command line option, 466
gitlab-application-appearance-get command line option, 466
gitlab-application-appearance-update command line option, 467
gitlab-application-create command line option, 465
gitlab-application-delete command line option, 466
gitlab-application-list command line option, 466
gitlab-application-settings command line option, 467
gitlab-application-settings-get command line option, 467
gitlab-application-settings-update command line option, 469
gitlab-application-statistics command line option, 471
gitlab-application-statistics-get command line option, 472
gitlab-audit-event command line option, 472
gitlab-audit-event-get command line option, 472
gitlab-audit-event-list command line option, 472
gitlab-broadcast-message command line option, 473
gitlab-broadcast-message-create command line option, 473
gitlab-broadcast-message-delete command line option, 473
gitlab-broadcast-message-get command line option, 473
gitlab-broadcast-message-list command line option, 474
gitlab-broadcast-message-update command line option, 474
gitlab-bulk-import command line option, 474
gitlab-bulk-import-all-entity command line option, 475
gitlab-bulk-import-all-entity-list command line option, 476
gitlab-bulk-import-create command line option, 475
gitlab-bulk-import-entity command line option, 476
gitlab-bulk-import-entity-get command line option, 476
gitlab-bulk-import-entity-list command line option, 476
gitlab-bulk-import-get command line option, 475
gitlab-bulk-import-list command line option, 475
gitlab-ci-lint command line option, 477
gitlab-ci-lint-create command line option, 477
gitlab-ci-lint-validate command line option, 477
gitlab-current-user command line option, 478
gitlab-current-user-email command line option, 478
gitlab-current-user-email-create command line option, 478
gitlab-current-user-email-delete command line option, 478
gitlab-current-user-email-get command line option, 478
gitlab-current-user-email-list command line option, 479
gitlab-current-user-get command line option, 478
gitlab-current-user-gpg-key command line option, 479
gitlab-current-user-gpg-key-create command line option, 479
gitlab-current-user-gpg-key-delete command line option, 479
gitlab-current-user-gpg-key-get command line option, 480
gitlab-current-user-gpg-key-list command line option, 480
gitlab-current-user-key command line option, 480
gitlab-current-user-key-create command line option, 480
gitlab-current-user-key-delete command line option, 481
gitlab-current-user-key-get command line option, 481
gitlab-current-user-key-list command line option, 481
gitlab-current-user-runner command line option, 481
gitlab-current-user-runner-create command line option, 482
gitlab-current-user-status command line option, 482

gitlab-current-user-status-get command line option, 482
gitlab-current-user-status-update command line option, 483
gitlab-deploy-key command line option, 483
gitlab-deploy-key-create command line option, 483
gitlab-deploy-key-list command line option, 483
gitlab-deploy-token command line option, 484
gitlab-deploy-token-list command line option, 484
gitlab-dockerfile command line option, 484
gitlab-dockerfile-get command line option, 484
gitlab-dockerfile-list command line option, 484
gitlab-event command line option, 485
gitlab-event-list command line option, 485
gitlab-feature command line option, 486
gitlab-feature-delete command line option, 486
gitlab-feature-list command line option, 486
gitlab-generic-package command line option, 486
gitlab-generic-package-download command line option, 486
gitlab-generic-package-upload command line option, 487
gitlab-geo-node command line option, 487
gitlab-geo-node-current-failures command line option, 487
gitlab-geo-node-delete command line option, 487
gitlab-geo-node-get command line option, 488
gitlab-geo-node-list command line option, 488
gitlab-geo-node-repair command line option, 488
gitlab-geo-node-status command line option, 488
gitlab-geo-node-update command line option, 489
gitlab-gitignore command line option, 489
gitlab-gitignore-get command line option, 489
gitlab-gitignore-list command line option, 489
gitlab-gitlabciyml command line option, 490
gitlab-gitlabciyml-get command line option, 490
gitlab-gitlabciyml-list command line option, 490
gitlab-group command line option, 490
gitlab-group-access-request command line option, 496
gitlab-group-access-request-approve command line option, 496
gitlab-group-access-request-create command line option, 496
gitlab-group-access-request-delete command line option, 497
gitlab-group-access-request-list command line option, 497
gitlab-group-access-token command line option, 497
gitlab-group-access-token-create command line option, 497
gitlab-group-access-token-delete command line option, 498
gitlab-group-access-token-get command line option, 498
gitlab-group-access-token-list command line option, 498
gitlab-group-access-token-rotate command line option, 499
gitlab-group-approval-rule command line option, 499
gitlab-group-approval-rule-create command line option, 499
gitlab-group-approval-rule-get command line option, 500
gitlab-group-approval-rule-list command line option, 500
gitlab-group-approval-rule-update command line option, 500
gitlab-group-audit-event command line option, 500
gitlab-group-audit-event-get command line option, 501
gitlab-group-audit-event-list command line option, 501
gitlab-group-badge command line option, 501
gitlab-group-badge-create command line option, 501
gitlab-group-badge-delete command line option, 502
gitlab-group-badge-get command line option, 502
gitlab-group-badge-list command line

option, 502
gitlab-group-badge-render command line option, 503
gitlab-group-badge-update command line option, 503
gitlab-group-billable-member command line option, 503
gitlab-group-billable-member-delete command line option, 503
gitlab-group-billable-member-list command line option, 504
gitlab-group-billable-member-membership command line option, 504
gitlab-group-billable-member-membership-list command line option, 504
gitlab-group-board command line option, 505
gitlab-group-board-create command line option, 505
gitlab-group-board-delete command line option, 505
gitlab-group-board-get command line option, 505
gitlab-group-board-list command line option, 506
gitlab-group-board-list-create command line option, 507
gitlab-group-board-list-delete command line option, 507
gitlab-group-board-list-get command line option, 507
gitlab-group-board-list-list command line option, 508
gitlab-group-board-list-update command line option, 508
gitlab-group-board-update command line option, 506
gitlab-group-cluster command line option, 508
gitlab-group-cluster-create command line option, 509
gitlab-group-cluster-delete command line option, 509
gitlab-group-cluster-get command line option, 509
gitlab-group-cluster-list command line option, 510
gitlab-group-cluster-update command line option, 510
gitlab-group-create command line option, 491
gitlab-group-custom-attribute command line option, 510
gitlab-group-custom-attribute-delete command line option, 511
gitlab-group-custom-attribute-get command line option, 511
gitlab-group-custom-attribute-list command line option, 511
gitlab-group-delete command line option, 492
gitlab-group-deploy-token command line option, 511
gitlab-group-deploy-token-create command line option, 512
gitlab-group-deploy-token-delete command line option, 512
gitlab-group-deploy-token-get command line option, 512
gitlab-group-deploy-token-list command line option, 512
gitlab-group-descendant-group command line option, 513
gitlab-group-descendant-group-list command line option, 513
gitlab-group-epic command line option, 514
gitlab-group-epic-award-emoji command line option, 516
gitlab-group-epic-award-emoji-create command line option, 516
gitlab-group-epic-award-emoji-delete command line option, 517
gitlab-group-epic-award-emoji-get command line option, 517
gitlab-group-epic-award-emoji-list command line option, 517
gitlab-group-epic-create command line option, 514
gitlab-group-epic-delete command line option, 515
gitlab-group-epic-discussion-note command line option, 518
gitlab-group-epic-discussion-note-create command line option, 518
gitlab-group-epic-discussion-note-delete command line option, 518
gitlab-group-epic-discussion-note-get command line option, 519
gitlab-group-epic-discussion-note-update command line option, 519
gitlab-group-epic-get command line option, 515
gitlab-group-epic-issue command line option, 519
gitlab-group-epic-issue-create command line option, 519
gitlab-group-epic-issue-delete command

line option, 520
gitlab-group-epic-issue-list command line option, 520
gitlab-group-epic-issue-update command line option, 521
gitlab-group-epic-list command line option, 515
gitlab-group-epic-note command line option, 521
gitlab-group-epic-note-award-emoji command line option, 523
gitlab-group-epic-note-award-emoji-create command line option, 523
gitlab-group-epic-note-award-emoji-delete command line option, 523
gitlab-group-epic-note-award-emoji-get command line option, 524
gitlab-group-epic-note-award-emoji-list command line option, 524
gitlab-group-epic-note-create command line option, 521
gitlab-group-epic-note-delete command line option, 521
gitlab-group-epic-note-get command line option, 522
gitlab-group-epic-note-list command line option, 522
gitlab-group-epic-note-update command line option, 522
gitlab-group-epic-resource-label-event command line option, 525
gitlab-group-epic-resource-label-event-get command line option, 525
gitlab-group-epic-resource-label-event-list command line option, 525
gitlab-group-epic-update command line option, 516
gitlab-group-export command line option, 525
gitlab-group-export-create command line option, 526
gitlab-group-export-download command line option, 526
gitlab-group-export-get command line option, 526
gitlab-group-get command line option, 492
gitlab-group-hook command line option, 526
gitlab-group-hook-create command line option, 527
gitlab-group-hook-delete command line option, 527
gitlab-group-hook-get command line option, 528
gitlab-group-hook-list command line option, 528
gitlab-group-hook-update command line option, 528
gitlab-group-import command line option, 529
gitlab-group-import-get command line option, 529
gitlab-group-invitation command line option, 530
gitlab-group-invitation-create command line option, 530
gitlab-group-invitation-delete command line option, 530
gitlab-group-invitation-get command line option, 531
gitlab-group-invitation-list command line option, 531
gitlab-group-invitation-update command line option, 531
gitlab-group-issue command line option, 532
gitlab-group-issue-list command line option, 532
gitlab-group-issues-statistics command line option, 533
gitlab-group-issues-statistics-get command line option, 533
gitlab-group-iteration command line option, 533
gitlab-group-iteration-list command line option, 533
gitlab-group-label command line option, 534
gitlab-group-label-create command line option, 534
gitlab-group-label-delete command line option, 535
gitlab-group-label-get command line option, 535
gitlab-group-label-list command line option, 535
gitlab-group-label-subscribe command line option, 535
gitlab-group-label-unsubscribe command line option, 536
gitlab-group-label-update command line option, 536
gitlab-group-ldap-group-link command line option, 536
gitlab-group-ldap-group-link-create command line option, 537
gitlab-group-ldap-group-link-delete command line option, 537

gitlab-group-ldap-group-link-list command line option, 537
gitlab-group-ldap-sync command line option, 492
gitlab-group-list command line option, 492
gitlab-group-member command line option, 538
gitlab-group-member-all command line option, 539
gitlab-group-member-all-get command line option, 540
gitlab-group-member-all-list command line option, 540
gitlab-group-member-create command line option, 538
gitlab-group-member-delete command line option, 538
gitlab-group-member-get command line option, 538
gitlab-group-member-list command line option, 539
gitlab-group-member-role command line option, 540
gitlab-group-member-role-create command line option, 541
gitlab-group-member-role-delete command line option, 542
gitlab-group-member-role-list command line option, 542
gitlab-group-member-update command line option, 539
gitlab-group-merge-request command line option, 543
gitlab-group-merge-request-list command line option, 543
gitlab-group-milestone command line option, 544
gitlab-group-milestone-create command line option, 544
gitlab-group-milestone-delete command line option, 545
gitlab-group-milestone-get command line option, 545
gitlab-group-milestone-issues command line option, 545
gitlab-group-milestone-list command line option, 545
gitlab-group-milestone-merge-requests command line option, 546
gitlab-group-milestone-update command line option, 546
gitlab-group-notification-settings command line option, 547
gitlab-group-notification-settings-get command line option, 547
gitlab-group-notification-settings-update command line option, 547
gitlab-group-package command line option, 548
gitlab-group-package-list command line option, 548
gitlab-group-project command line option, 549
gitlab-group-project-list command line option, 549
gitlab-group-push-rules command line option, 550
gitlab-group-push-rules-create command line option, 550
gitlab-group-push-rules-delete command line option, 551
gitlab-group-push-rules-get command line option, 551
gitlab-group-push-rules-update command line option, 552
gitlab-group-restore command line option, 493
gitlab-group-runner command line option, 552
gitlab-group-runner-list command line option, 552
gitlab-group-saml-group-link command line option, 553
gitlab-group-saml-group-link-create command line option, 553
gitlab-group-saml-group-link-delete command line option, 553
gitlab-group-saml-group-link-get command line option, 554
gitlab-group-saml-group-link-list command line option, 554
gitlab-group-search command line option, 493
gitlab-group-service-account command line option, 554
gitlab-group-service-account-access-token command line option, 556
gitlab-group-service-account-access-token-create command line option, 556
gitlab-group-service-account-access-token-delete command line option, 557
gitlab-group-service-account-access-token-list command line option, 557
gitlab-group-service-account-access-token-rotate command line option, 558
gitlab-group-service-account-create command line option, 554

gitlab-group-service-account-delete command line option, 555
gitlab-group-service-account-list command line option, 555
gitlab-group-service-account-update command line option, 556
gitlab-group-share command line option, 494
gitlab-group-subgroup command line option, 558
gitlab-group-subgroup-list command line option, 559
gitlab-group-transfer command line option, 494
gitlab-group-transfer-project command line option, 494
gitlab-group-unshare command line option, 494
gitlab-group-update command line option, 495
gitlab-group-variable command line option, 559
gitlab-group-variable-create command line option, 560
gitlab-group-variable-delete command line option, 560
gitlab-group-variable-get command line option, 560
gitlab-group-variable-list command line option, 561
gitlab-group-variable-update command line option, 561
gitlab-group-wiki command line option, 562
gitlab-group-wiki-create command line option, 562
gitlab-group-wiki-delete command line option, 562
gitlab-group-wiki-get command line option, 562
gitlab-group-wiki-list command line option, 563
gitlab-group-wiki-update command line option, 563
gitlab-hook command line option, 563
gitlab-hook-create command line option, 564
gitlab-hook-delete command line option, 564
gitlab-hook-get command line option, 564
gitlab-hook-list command line option, 564
gitlab-issue command line option, 565
gitlab-issue-get command line option, 565
gitlab-issue-list command line option, 565
gitlab-issues-statistics command line option, 566
gitlab-issues-statistics-get command line option, 566
gitlab-key command line option, 566
gitlab-key-get command line option, 566
gitlab-ldap-group command line option, 567
gitlab-ldap-group-list command line option, 567
gitlab-license command line option, 567
gitlab-license-get command line option, 567
gitlab-license-list command line option, 568
gitlab-member-role command line option, 568
gitlab-member-role-create command line option, 569
gitlab-member-role-delete command line option, 570
gitlab-member-role-list command line option, 570
gitlab-merge-request command line option, 570
gitlab-merge-request-list command line option, 571
gitlab-namespace command line option, 572
gitlab-namespace-exists command line option, 572
gitlab-namespace-get command line option, 573
gitlab-namespace-list command line option, 573
gitlab-notification-settings command line option, 573
gitlab-notification-settings-get command line option, 573
gitlab-notification-settings-update command line option, 574
gitlab-pages-domain command line option, 574
gitlab-pages-domain-list command line option, 574
gitlab-personal-access-token command line option, 575
gitlab-personal-access-token-delete command line option, 575
gitlab-personal-access-token-get command line option, 575
gitlab-personal-access-token-list command line option, 575
gitlab-personal-access-token-rotate

command line option, 576
gitlab-project command line option, 576
gitlab-project-access-request command line option, 593
gitlab-project-access-request-approve command line option, 593
gitlab-project-access-request-create command line option, 593
gitlab-project-access-request-delete command line option, 593
gitlab-project-access-request-list command line option, 594
gitlab-project-access-token command line option, 594
gitlab-project-access-token-create command line option, 594
gitlab-project-access-token-delete command line option, 595
gitlab-project-access-token-get command line option, 595
gitlab-project-access-token-list command line option, 595
gitlab-project-access-token-rotate command line option, 596
gitlab-project-additional-statistics command line option, 596
gitlab-project-additional-statistics-get command line option, 596
gitlab-project-approval command line option, 596
gitlab-project-approval-get command line option, 596
gitlab-project-approval-rule command line option, 597
gitlab-project-approval-rule-create command line option, 597
gitlab-project-approval-rule-delete command line option, 598
gitlab-project-approval-rule-get command line option, 598
gitlab-project-approval-rule-list command line option, 598
gitlab-project-approval-rule-update command line option, 599
gitlab-project-approval-update command line option, 597
gitlab-project-archive command line option, 576
gitlab-project-artifact command line option, 599
gitlab-project-artifact-delete command line option, 599
gitlab-project-artifact-download command line option, 599
gitlab-project-artifact-raw command line option, 600
gitlab-project-audit-event command line option, 600
gitlab-project-audit-event-get command line option, 600
gitlab-project-audit-event-list command line option, 601
gitlab-project-badge command line option, 601
gitlab-project-badge-create command line option, 601
gitlab-project-badge-delete command line option, 602
gitlab-project-badge-get command line option, 602
gitlab-project-badge-list command line option, 602
gitlab-project-badge-render command line option, 602
gitlab-project-badge-update command line option, 603
gitlab-project-board command line option, 603
gitlab-project-board-create command line option, 603
gitlab-project-board-delete command line option, 604
gitlab-project-board-get command line option, 604
gitlab-project-board-list command line option, 604, 605
gitlab-project-board-list-create command line option, 605
gitlab-project-board-list-delete command line option, 605
gitlab-project-board-list-get command line option, 606
gitlab-project-board-list-list command line option, 606
gitlab-project-board-list-update command line option, 606
gitlab-project-board-update command line option, 604
gitlab-project-branch command line option, 607
gitlab-project-branch-create command line option, 607
gitlab-project-branch-delete command line option, 607
gitlab-project-branch-get command line option, 607
gitlab-project-branch-list command line option, 608

gitlab-project-ci-lint command line option, 608
gitlab-project-ci-lint-create command line option, 608
gitlab-project-ci-lint-get command line option, 609
gitlab-project-ci-lint-validate command line option, 609
gitlab-project-cluster command line option, 609
gitlab-project-cluster-agent command line option, 611
gitlab-project-cluster-agent-create command line option, 612
gitlab-project-cluster-agent-delete command line option, 612
gitlab-project-cluster-agent-get command line option, 612
gitlab-project-cluster-agent-list command line option, 612
gitlab-project-cluster-create command line option, 610
gitlab-project-cluster-delete command line option, 610
gitlab-project-cluster-get command line option, 610
gitlab-project-cluster-list command line option, 611
gitlab-project-cluster-update command line option, 611
gitlab-project-commit command line option, 613
gitlab-project-commit-cherry-pick command line option, 613
gitlab-project-commit-comment command line option, 616
gitlab-project-commit-comment-create command line option, 617
gitlab-project-commit-comment-list command line option, 617
gitlab-project-commit-create command line option, 613
gitlab-project-commit-diff command line option, 614
gitlab-project-commit-discussion command line option, 617
gitlab-project-commit-discussion-create command line option, 618
gitlab-project-commit-discussion-get command line option, 618
gitlab-project-commit-discussion-list command line option, 618
gitlab-project-commit-discussion-note command line option, 619
gitlab-project-commit-discussion-note-create command line option, 619
gitlab-project-commit-discussion-note-delete command line option, 619
gitlab-project-commit-discussion-note-get command line option, 620
gitlab-project-commit-discussion-note-update command line option, 620
gitlab-project-commit-get command line option, 614
gitlab-project-commit-list command line option, 614
gitlab-project-commit-merge-requests command line option, 615
gitlab-project-commit-refs command line option, 615
gitlab-project-commit-revert command line option, 616
gitlab-project-commit-sequence command line option, 616
gitlab-project-commit-signature command line option, 616
gitlab-project-commit-status command line option, 621
gitlab-project-commit-status-create command line option, 621
gitlab-project-commit-status-list command line option, 621
gitlab-project-create command line option, 578
gitlab-project-create-fork-relation command line option, 581
gitlab-project-custom-attribute command line option, 622
gitlab-project-custom-attribute-delete command line option, 622
gitlab-project-custom-attribute-get command line option, 622
gitlab-project-custom-attribute-list command line option, 622
gitlab-project-delete command line option, 581
gitlab-project-delete-fork-relation command line option, 581
gitlab-project-delete-merged-branches command line option, 581
gitlab-project-deploy-token command line option, 623
gitlab-project-deploy-token-create command line option, 623
gitlab-project-deploy-token-delete command line option, 623
gitlab-project-deploy-token-get command line option, 624

gitlab-project-deploy-token-list command line option, 624
gitlab-project-deployment command line option, 624
gitlab-project-deployment-approval command line option, 625
gitlab-project-deployment-create command line option, 625
gitlab-project-deployment-get command line option, 625
gitlab-project-deployment-list command line option, 626
gitlab-project-deployment-merge-request command line option, 627
gitlab-project-deployment-merge-request-list command line option, 627
gitlab-project-deployment-update command line option, 626
gitlab-project-dockerfile-template command line option, 629
gitlab-project-dockerfile-template-get command line option, 629
gitlab-project-dockerfile-template-list command line option, 629
gitlab-project-environment command line option, 630
gitlab-project-environment-create command line option, 630
gitlab-project-environment-delete command line option, 630
gitlab-project-environment-get command line option, 630
gitlab-project-environment-list command line option, 631
gitlab-project-environment-stop command line option, 631
gitlab-project-environment-update command line option, 631
gitlab-project-event command line option, 632
gitlab-project-event-list command line option, 632
gitlab-project-export command line option, 633
gitlab-project-export-create command line option, 633
gitlab-project-export-download command line option, 633
gitlab-project-export-get command line option, 633
gitlab-project-external-status-check command line option, 633
gitlab-project-external-status-check-create command line option, 634
gitlab-project-external-status-check-delete command line option, 634
gitlab-project-external-status-check-list command line option, 634
gitlab-project-external-status-check-update command line option, 635
gitlab-project-feature-flag command line option, 635
gitlab-project-feature-flag-create command line option, 635
gitlab-project-feature-flag-delete command line option, 636
gitlab-project-feature-flag-get command line option, 636
gitlab-project-feature-flag-list command line option, 636
gitlab-project-feature-flag-update command line option, 637
gitlab-project-feature-flag-user-list command line option, 637
gitlab-project-feature-flag-user-list-create command line option, 637
gitlab-project-feature-flag-user-list-delete command line option, 638
gitlab-project-feature-flag-user-list-get command line option, 638
gitlab-project-feature-flag-user-list-list command line option, 638
gitlab-project-feature-flag-user-list-update command line option, 639
gitlab-project-file command line option, 639
gitlab-project-file-blame command line option, 639
gitlab-project-file-create command line option, 640
gitlab-project-file-delete command line option, 640
gitlab-project-file-get command line option, 641
gitlab-project-file-raw command line option, 641
gitlab-project-file-update command line option, 641
gitlab-project-fork command line option, 642
gitlab-project-fork-create command line option, 642
gitlab-project-fork-list command line option, 642
gitlab-project-get command line option, 581
gitlab-project-gitignore-template command line option, 643

gitlab-project-gitignore-template-get command line option, 643
gitlab-project-gitignore-template-list command line option, 644
gitlab-project-gitlabciyml-template command line option, 644
gitlab-project-gitlabciyml-template-get command line option, 644
gitlab-project-gitlabciyml-template-list command line option, 645
gitlab-project-hook command line option, 645
gitlab-project-hook-create command line option, 645
gitlab-project-hook-delete command line option, 646
gitlab-project-hook-get command line option, 646
gitlab-project-hook-list command line option, 646
gitlab-project-hook-update command line option, 647
gitlab-project-housekeeping command line option, 582
gitlab-project-import command line option, 648
gitlab-project-import-get command line option, 648
gitlab-project-integration command line option, 648
gitlab-project-integration-available command line option, 648
gitlab-project-integration-delete command line option, 648
gitlab-project-integration-get command line option, 649
gitlab-project-integration-list command line option, 649
gitlab-project-integration-update command line option, 649
gitlab-project-invitation command line option, 650
gitlab-project-invitation-create command line option, 650
gitlab-project-invitation-delete command line option, 650
gitlab-project-invitation-get command line option, 651
gitlab-project-invitation-list command line option, 651
gitlab-project-invitation-update command line option, 651
gitlab-project-issue command line option, 652
gitlab-project-issue-add-spent-time command line option, 652
gitlab-project-issue-award-emoji command line option, 659
gitlab-project-issue-award-emoji-create command line option, 659
gitlab-project-issue-award-emoji-delete command line option, 659
gitlab-project-issue-award-emoji-get command line option, 660
gitlab-project-issue-award-emoji-list command line option, 660
gitlab-project-issue-closed-by command line option, 652
gitlab-project-issue-create command line option, 652
gitlab-project-issue-delete command line option, 653
gitlab-project-issue-discussion command line option, 660
gitlab-project-issue-discussion-create command line option, 661
gitlab-project-issue-discussion-get command line option, 661
gitlab-project-issue-discussion-list command line option, 661
gitlab-project-issue-discussion-note command line option, 662
gitlab-project-issue-discussion-note-create command line option, 662
gitlab-project-issue-discussion-note-delete command line option, 662
gitlab-project-issue-discussion-note-get command line option, 663
gitlab-project-issue-discussion-note-list command line option, 663
gitlab-project-issue-discussion-note-update command line option, 664
gitlab-project-issue-get command line option, 653
gitlab-project-issue-link command line option, 664
gitlab-project-issue-link-create command line option, 664
gitlab-project-issue-link-delete command line option, 664
gitlab-project-issue-link-list command line option, 665
gitlab-project-issue-list command line option, 654
gitlab-project-issue-move command line option, 655
gitlab-project-issue-note command line option, 665

gitlab-project-issue-note-award-emoji command line option, 667
gitlab-project-issue-note-award-emoji-create command line option, 667
gitlab-project-issue-note-award-emoji-delete command line option, 668
gitlab-project-issue-note-award-emoji-get command line option, 668
gitlab-project-issue-note-award-emoji-list command line option, 668
gitlab-project-issue-note-create command line option, 665
gitlab-project-issue-note-delete command line option, 666
gitlab-project-issue-note-get command line option, 666
gitlab-project-issue-note-list command line option, 666
gitlab-project-issue-note-update command line option, 667
gitlab-project-issue-participants command line option, 655
gitlab-project-issue-related-merge-requests command line option, 655
gitlab-project-issue-reorder command line option, 656
gitlab-project-issue-reset-spent-time command line option, 656
gitlab-project-issue-reset-time-estimate command line option, 656
gitlab-project-issue-resource-iteration-event command line option, 669
gitlab-project-issue-resource-iteration-event-get command line option, 669
gitlab-project-issue-resource-iteration-event-list command line option, 669
gitlab-project-issue-resource-label-event command line option, 670
gitlab-project-issue-resource-label-event-get command line option, 670
gitlab-project-issue-resource-label-event-list command line option, 670
gitlab-project-issue-resource-milestone-event command line option, 671
gitlab-project-issue-resource-milestone-event-get command line option, 671
gitlab-project-issue-resource-milestone-event-list command line option, 671
gitlab-project-issue-resource-state-event command line option, 672
gitlab-project-issue-resource-state-event-get command line option, 672
gitlab-project-issue-resource-state-event-list command line option, 672
gitlab-project-issue-resource-weight-event command line option, 673
gitlab-project-issue-resource-weight-event-get command line option, 673
gitlab-project-issue-resource-weight-event-list command line option, 673
gitlab-project-issue-subscribe command line option, 656
gitlab-project-issue-template command line option, 674
gitlab-project-issue-template-get command line option, 674
gitlab-project-issue-template-list command line option, 674
gitlab-project-issue-time-estimate command line option, 657
gitlab-project-issue-time-stats command line option, 657
gitlab-project-issue-todo command line option, 657
gitlab-project-issue-unsubscribe command line option, 657
gitlab-project-issue-update command line option, 658
gitlab-project-issue-user-agent-detail command line option, 659
gitlab-project-issues-statistics command line option, 675
gitlab-project-issues-statistics-get command line option, 675
gitlab-project-job command line option, 675
gitlab-project-job-artifact command line option, 675
gitlab-project-job-artifacts command line option, 675
gitlab-project-job-cancel command line option, 676
gitlab-project-job-delete-artifacts command line option, 676
gitlab-project-job-erase command line option, 676
gitlab-project-job-get command line option, 676
gitlab-project-job-keep-artifacts command line option, 677
gitlab-project-job-list command line option, 677
gitlab-project-job-play command line option, 677
gitlab-project-job-retry command line option, 678
gitlab-project-job-token-scope command line option, 678

gitlab-project-job-token-scope-get command line option, 678
gitlab-project-job-token-scope-update command line option, 678
gitlab-project-job-trace command line option, 678
gitlab-project-key command line option, 679
gitlab-project-key-create command line option, 679
gitlab-project-key-delete command line option, 679
gitlab-project-key-enable command line option, 679
gitlab-project-key-get command line option, 680
gitlab-project-key-list command line option, 680
gitlab-project-key-update command line option, 680
gitlab-project-label command line option, 681
gitlab-project-label-create command line option, 681
gitlab-project-label-delete command line option, 681
gitlab-project-label-get command line option, 681
gitlab-project-label-list command line option, 682
gitlab-project-label-subscribe command line option, 682
gitlab-project-label-unsubscribe command line option, 682
gitlab-project-label-update command line option, 683
gitlab-project-languages command line option, 582
gitlab-project-license-template command line option, 683
gitlab-project-license-template-get command line option, 683
gitlab-project-license-template-list command line option, 683
gitlab-project-list command line option, 582
gitlab-project-member command line option, 684
gitlab-project-member-all command line option, 686
gitlab-project-member-all-get command line option, 686
gitlab-project-member-all-list command line option, 686
gitlab-project-member-create command line option, 684
gitlab-project-member-delete command line option, 684
gitlab-project-member-get command line option, 685
gitlab-project-member-list command line option, 685
gitlab-project-member-update command line option, 685
gitlab-project-merge-request command line option, 687
gitlab-project-merge-request-add-spent-time command line option, 687
gitlab-project-merge-request-approval command line option, 696
gitlab-project-merge-request-approval-get command line option, 697
gitlab-project-merge-request-approval-rule command line option, 697
gitlab-project-merge-request-approval-rule-create command line option, 697
gitlab-project-merge-request-approval-rule-delete command line option, 698
gitlab-project-merge-request-approval-rule-get command line option, 698
gitlab-project-merge-request-approval-rule-list command line option, 699
gitlab-project-merge-request-approval-rule-update command line option, 699
gitlab-project-merge-request-approval-state command line option, 700
gitlab-project-merge-request-approval-state-get command line option, 700
gitlab-project-merge-request-approval-update command line option, 697
gitlab-project-merge-request-approve command line option, 687
gitlab-project-merge-request-award-emoji command line option, 700
gitlab-project-merge-request-award-emoji-create command line option, 700
gitlab-project-merge-request-award-emoji-delete command line option, 701
gitlab-project-merge-request-award-emoji-get command line option, 701
gitlab-project-merge-request-award-emoji-list command line option, 701
gitlab-project-merge-request-cancel-merge-when-pipelin command line option, 687
gitlab-project-merge-request-changes command line option, 688
gitlab-project-merge-request-closes-issues command line option, 688

gitlab-project-merge-request-commits
    command line option, 688
gitlab-project-merge-request-create
    command line option, 689
gitlab-project-merge-request-delete
    command line option, 690
gitlab-project-merge-request-diff
    command line option, 702
gitlab-project-merge-request-diff-get
    command line option, 702
gitlab-project-merge-request-diff-list
    command line option, 702
gitlab-project-merge-request-discussion
    command line option, 703
gitlab-project-merge-request-discussion-create
    command line option, 703
gitlab-project-merge-request-discussion-get
    command line option, 703
gitlab-project-merge-request-discussion-list
    command line option, 703
gitlab-project-merge-request-discussion-note
    command line option, 704
gitlab-project-merge-request-discussion-note-create
    command line option, 704
gitlab-project-merge-request-discussion-note-delete
    command line option, 705
gitlab-project-merge-request-discussion-note-get
    command line option, 705
gitlab-project-merge-request-discussion-note-list
    command line option, 706
gitlab-project-merge-request-discussion-note-update
    command line option, 706
gitlab-project-merge-request-discussion-update
    command line option, 704
gitlab-project-merge-request-draft-note
    command line option, 707
gitlab-project-merge-request-draft-note-create
    command line option, 707
gitlab-project-merge-request-draft-note-delete
    command line option, 707
gitlab-project-merge-request-draft-note-get
    command line option, 708
gitlab-project-merge-request-draft-note-list
    command line option, 708
gitlab-project-merge-request-draft-note-update
    command line option, 708
gitlab-project-merge-request-get
    command line option, 690
gitlab-project-merge-request-list
    command line option, 691
gitlab-project-merge-request-merge
    command line option, 692
gitlab-project-merge-request-merge-ref
    command line option, 692
gitlab-project-merge-request-note
    command line option, 709
gitlab-project-merge-request-note-award-emoji
    command line option, 711
gitlab-project-merge-request-note-award-emoji-create
    command line option, 711
gitlab-project-merge-request-note-award-emoji-delete
    command line option, 711
gitlab-project-merge-request-note-award-emoji-get
    command line option, 711
gitlab-project-merge-request-note-award-emoji-list
    command line option, 712
gitlab-project-merge-request-note-create
    command line option, 709
gitlab-project-merge-request-note-delete
    command line option, 709
gitlab-project-merge-request-note-get
    command line option, 709
gitlab-project-merge-request-note-list
    command line option, 710
gitlab-project-merge-request-note-update
    command line option, 710
gitlab-project-merge-request-participants
    command line option, 692
gitlab-project-merge-request-pipeline
    command line option, 712
gitlab-project-merge-request-pipeline-create
    command line option, 712
gitlab-project-merge-request-pipeline-list
    command line option, 713
gitlab-project-merge-request-rebase
    command line option, 693
gitlab-project-merge-request-related-issues
    command line option, 693
gitlab-project-merge-request-reset-approvals
    command line option, 693
gitlab-project-merge-request-reset-spent-time
    command line option, 693
gitlab-project-merge-request-reset-time-estimate
    command line option, 694
gitlab-project-merge-request-resource-label-event
    command line option, 713
gitlab-project-merge-request-resource-label-event-get
    command line option, 713
gitlab-project-merge-request-resource-label-event-list
    command line option, 714
gitlab-project-merge-request-resource-milestone-event
    command line option, 714
gitlab-project-merge-request-resource-milestone-event-
    command line option, 714
gitlab-project-merge-request-resource-milestone-event-
    command line option, 715
gitlab-project-merge-request-resource-state-event
    command line option, 715

gitlab-project-merge-request-resource-state-event-get command line option, 715
gitlab-project-merge-request-resource-state-event-list command line option, 715
gitlab-project-merge-request-reviewer-detail command line option, 716
gitlab-project-merge-request-reviewer-detail-list command line option, 716
gitlab-project-merge-request-status-check command line option, 716
gitlab-project-merge-request-status-check-list command line option, 717
gitlab-project-merge-request-subscribe command line option, 694
gitlab-project-merge-request-template command line option, 717
gitlab-project-merge-request-template-get command line option, 717
gitlab-project-merge-request-template-list command line option, 718
gitlab-project-merge-request-time-estimate command line option, 694
gitlab-project-merge-request-time-stats command line option, 695
gitlab-project-merge-request-todo command line option, 695
gitlab-project-merge-request-unapprove command line option, 695
gitlab-project-merge-request-unsubscribe command line option, 695
gitlab-project-merge-request-update command line option, 696
gitlab-project-merge-train command line option, 718
gitlab-project-merge-train-list command line option, 718
gitlab-project-milestone command line option, 719
gitlab-project-milestone-create command line option, 719
gitlab-project-milestone-delete command line option, 719
gitlab-project-milestone-get command line option, 719
gitlab-project-milestone-issues command line option, 720
gitlab-project-milestone-list command line option, 720
gitlab-project-milestone-merge-requests command line option, 720
gitlab-project-milestone-update command line option, 721
gitlab-project-mirror-pull command line option, 583
gitlab-project-mirror-pull-details command line option, 584
gitlab-project-note command line option, 721
gitlab-project-note-get command line option, 721
gitlab-project-note-list command line option, 722
gitlab-project-notification-settings command line option, 722
gitlab-project-notification-settings-get command line option, 722
gitlab-project-notification-settings-update command line option, 723
gitlab-project-package command line option, 723
gitlab-project-package-delete command line option, 723
gitlab-project-package-file command line option, 725
gitlab-project-package-file-delete command line option, 725
gitlab-project-package-file-list command line option, 725
gitlab-project-package-get command line option, 724
gitlab-project-package-list command line option, 724
gitlab-project-package-pipeline command line option, 726
gitlab-project-package-pipeline-list command line option, 726
gitlab-project-package-protection-rule command line option, 726
gitlab-project-package-protection-rule-create command line option, 726
gitlab-project-package-protection-rule-delete command line option, 727
gitlab-project-package-protection-rule-list command line option, 727
gitlab-project-package-protection-rule-update command line option, 728
gitlab-project-pages command line option, 728
gitlab-project-pages-delete command line option, 728
gitlab-project-pages-domain command line option, 729
gitlab-project-pages-domain-create command line option, 729
gitlab-project-pages-domain-delete command line option, 729
gitlab-project-pages-domain-get command line option, 730

gitlab-project-pages-domain-list command line option, 730
gitlab-project-pages-domain-update command line option, 730
gitlab-project-pages-get command line option, 728
gitlab-project-pages-update command line option, 728
gitlab-project-pipeline command line option, 731
gitlab-project-pipeline-bridge command line option, 733
gitlab-project-pipeline-bridge-list command line option, 733
gitlab-project-pipeline-cancel command line option, 731
gitlab-project-pipeline-create command line option, 731
gitlab-project-pipeline-delete command line option, 731
gitlab-project-pipeline-get command line option, 731
gitlab-project-pipeline-job command line option, 733
gitlab-project-pipeline-job-list command line option, 734
gitlab-project-pipeline-list command line option, 732
gitlab-project-pipeline-retry command line option, 733
gitlab-project-pipeline-schedule command line option, 734
gitlab-project-pipeline-schedule-create command line option, 734
gitlab-project-pipeline-schedule-delete command line option, 735
gitlab-project-pipeline-schedule-get command line option, 735
gitlab-project-pipeline-schedule-list command line option, 735
gitlab-project-pipeline-schedule-pipeline command line option, 737
gitlab-project-pipeline-schedule-pipeline-list command line option, 737
gitlab-project-pipeline-schedule-play command line option, 736
gitlab-project-pipeline-schedule-take-ownership command line option, 736
gitlab-project-pipeline-schedule-update command line option, 736
gitlab-project-pipeline-schedule-variable command line option, 737
gitlab-project-pipeline-schedule-variable-create command line option, 738
gitlab-project-pipeline-schedule-variable-delete command line option, 738
gitlab-project-pipeline-schedule-variable-update command line option, 738
gitlab-project-pipeline-test-report command line option, 739
gitlab-project-pipeline-test-report-get command line option, 739
gitlab-project-pipeline-test-report-summary command line option, 739
gitlab-project-pipeline-test-report-summary-get command line option, 739
gitlab-project-pipeline-variable command line option, 739
gitlab-project-pipeline-variable-list command line option, 740
gitlab-project-protected-branch command line option, 740
gitlab-project-protected-branch-create command line option, 740
gitlab-project-protected-branch-delete command line option, 741
gitlab-project-protected-branch-get command line option, 741
gitlab-project-protected-branch-list command line option, 741
gitlab-project-protected-branch-update command line option, 742
gitlab-project-protected-environment command line option, 742
gitlab-project-protected-environment-create command line option, 742
gitlab-project-protected-environment-delete command line option, 743
gitlab-project-protected-environment-get command line option, 743
gitlab-project-protected-environment-list command line option, 743
gitlab-project-protected-tag command line option, 744
gitlab-project-protected-tag-create command line option, 744
gitlab-project-protected-tag-delete command line option, 744
gitlab-project-protected-tag-get command line option, 744
gitlab-project-protected-tag-list command line option, 745
gitlab-project-pull-mirror command line option, 745
gitlab-project-pull-mirror-create command line option, 745
gitlab-project-pull-mirror-get command line option, 745

gitlab-project-pull-mirror-start command line option, 746
gitlab-project-pull-mirror-update command line option, 746
gitlab-project-push-rules command line option, 746
gitlab-project-push-rules-create command line option, 746
gitlab-project-push-rules-delete command line option, 747
gitlab-project-push-rules-get command line option, 747
gitlab-project-push-rules-update command line option, 748
gitlab-project-registry-protection-rule command line option, 748
gitlab-project-registry-protection-rule-create command line option, 749
gitlab-project-registry-protection-rule-list command line option, 749
gitlab-project-registry-protection-rule-update command line option, 750
gitlab-project-registry-repository command line option, 750
gitlab-project-registry-repository-delete command line option, 750
gitlab-project-registry-repository-list command line option, 750
gitlab-project-registry-repository-protection-rule command line option, 751
gitlab-project-registry-repository-protection-rule-create command line option, 751
gitlab-project-registry-repository-protection-rule-delete command line option, 751
gitlab-project-registry-repository-protection-rule-list command line option, 752
gitlab-project-registry-repository-protection-rule-update command line option, 752
gitlab-project-registry-tag command line option, 752
gitlab-project-registry-tag-delete command line option, 753
gitlab-project-registry-tag-delete-in-bulk command line option, 753
gitlab-project-registry-tag-get command line option, 753
gitlab-project-registry-tag-list command line option, 754
gitlab-project-release command line option, 754
gitlab-project-release-create command line option, 754
gitlab-project-release-delete command line option, 755
gitlab-project-release-get command line option, 755
gitlab-project-release-link command line option, 756
gitlab-project-release-link-create command line option, 756
gitlab-project-release-link-delete command line option, 757
gitlab-project-release-link-get command line option, 757
gitlab-project-release-link-list command line option, 758
gitlab-project-release-link-update command line option, 758
gitlab-project-release-list command line option, 755
gitlab-project-release-update command line option, 756
gitlab-project-remote-mirror command line option, 759
gitlab-project-remote-mirror-create command line option, 759
gitlab-project-remote-mirror-delete command line option, 759
gitlab-project-remote-mirror-list command line option, 759
gitlab-project-remote-mirror-sync command line option, 760
gitlab-project-remote-mirror-update command line option, 760
gitlab-project-repository-archive command line option, 584
gitlab-project-repository-blob command line option, 584
gitlab-project-repository-compare command line option, 584
gitlab-project-repository-contributors command line option, 584
gitlab-project-repository-merge-base command line option, 585
gitlab-project-repository-raw-blob command line option, 585
gitlab-project-repository-tree command line option, 585
gitlab-project-resource-group command line option, 760
gitlab-project-resource-group-get command line option, 761
gitlab-project-resource-group-list command line option, 761
gitlab-project-resource-group-upcoming-job command line option, 762
gitlab-project-resource-group-upcoming-job-list command line option, 762

gitlab-project-resource-group-update command line option, 761
gitlab-project-restore command line option, 585
gitlab-project-runner command line option, 762
gitlab-project-runner-create command line option, 763
gitlab-project-runner-delete command line option, 763
gitlab-project-runner-list command line option, 763
gitlab-project-search command line option, 585
gitlab-project-secure-file command line option, 764
gitlab-project-secure-file-create command line option, 764
gitlab-project-secure-file-delete command line option, 764
gitlab-project-secure-file-download command line option, 764
gitlab-project-secure-file-get command line option, 765
gitlab-project-secure-file-list command line option, 765
gitlab-project-service command line option, 765
gitlab-project-service-account command line option, 767
gitlab-project-service-account-access-token command line option, 768
gitlab-project-service-account-access-token-create command line option, 769
gitlab-project-service-account-access-token-delete command line option, 769
gitlab-project-service-account-access-token-list command line option, 770
gitlab-project-service-account-access-token-rotate command line option, 771
gitlab-project-service-account-create command line option, 767
gitlab-project-service-account-delete command line option, 767
gitlab-project-service-account-list command line option, 768
gitlab-project-service-account-update command line option, 768
gitlab-project-service-available command line option, 765
gitlab-project-service-delete command line option, 766
gitlab-project-service-get command line option, 766
gitlab-project-service-list command line option, 766
gitlab-project-service-update command line option, 766
gitlab-project-share command line option, 586
gitlab-project-snapshot command line option, 586
gitlab-project-snippet command line option, 771
gitlab-project-snippet-award-emoji command line option, 773
gitlab-project-snippet-award-emoji-create command line option, 774
gitlab-project-snippet-award-emoji-delete command line option, 774
gitlab-project-snippet-award-emoji-get command line option, 774
gitlab-project-snippet-award-emoji-list command line option, 775
gitlab-project-snippet-content command line option, 771
gitlab-project-snippet-create command line option, 771
gitlab-project-snippet-delete command line option, 772
gitlab-project-snippet-discussion command line option, 775
gitlab-project-snippet-discussion-create command line option, 775
gitlab-project-snippet-discussion-get command line option, 776
gitlab-project-snippet-discussion-list command line option, 776
gitlab-project-snippet-discussion-note command line option, 776
gitlab-project-snippet-discussion-note-create command line option, 777
gitlab-project-snippet-discussion-note-delete command line option, 777
gitlab-project-snippet-discussion-note-get command line option, 777
gitlab-project-snippet-discussion-note-update command line option, 778
gitlab-project-snippet-get command line option, 772
gitlab-project-snippet-list command line option, 772
gitlab-project-snippet-note command line option, 778
gitlab-project-snippet-note-award-emoji command line option, 780
gitlab-project-snippet-note-award-emoji-create command line option, 780

gitlab-project-snippet-note-award-emoji-delete command line option, 781
gitlab-project-snippet-note-award-emoji-get command line option, 781
gitlab-project-snippet-note-award-emoji-list command line option, 781
gitlab-project-snippet-note-create command line option, 778
gitlab-project-snippet-note-delete command line option, 779
gitlab-project-snippet-note-get command line option, 779
gitlab-project-snippet-note-list command line option, 779
gitlab-project-snippet-note-update command line option, 780
gitlab-project-snippet-update command line option, 773
gitlab-project-snippet-user-agent-detail command line option, 773
gitlab-project-star command line option, 586
gitlab-project-storage command line option, 782
gitlab-project-storage-get command line option, 782
gitlab-project-tag command line option, 782
gitlab-project-tag-create command line option, 782
gitlab-project-tag-delete command line option, 783
gitlab-project-tag-get command line option, 783
gitlab-project-tag-list command line option, 783
gitlab-project-transfer command line option, 586
gitlab-project-trigger command line option, 784
gitlab-project-trigger-create command line option, 784
gitlab-project-trigger-delete command line option, 784
gitlab-project-trigger-get command line option, 784
gitlab-project-trigger-list command line option, 785
gitlab-project-trigger-pipeline command line option, 587
gitlab-project-trigger-update command line option, 785
gitlab-project-unarchive command line option, 587
gitlab-project-unshare command line option, 587
gitlab-project-unstar command line option, 587
gitlab-project-update command line option, 589
gitlab-project-update-submodule command line option, 592
gitlab-project-upload command line option, 592
gitlab-project-user command line option, 785
gitlab-project-user-list command line option, 786
gitlab-project-variable command line option, 786
gitlab-project-variable-create command line option, 786
gitlab-project-variable-delete command line option, 787
gitlab-project-variable-get command line option, 787
gitlab-project-variable-list command line option, 787
gitlab-project-variable-update command line option, 788
gitlab-project-wiki command line option, 788
gitlab-project-wiki-create command line option, 788
gitlab-project-wiki-delete command line option, 789
gitlab-project-wiki-get command line option, 789
gitlab-project-wiki-list command line option, 789
gitlab-project-wiki-update command line option, 790
gitlab-project-wiki-upload command line option, 790
gitlab-registry-repository command line option, 790
gitlab-registry-repository-get command line option, 791
gitlab-runner command line option, 791
gitlab-runner-all command line option, 791, 793
gitlab-runner-all-list command line option, 793
gitlab-runner-create command line option, 791
gitlab-runner-delete command line option, 792
gitlab-runner-get command line option,

792
gitlab-runner-job command line option, 794
gitlab-runner-job-list command line option, 794
gitlab-runner-list command line option, 792
gitlab-runner-update command line option, 793
gitlab-runner-verify command line option, 793
gitlab-service-account command line option, 795
gitlab-service-account-create command line option, 795
gitlab-service-account-list command line option, 795
gitlab-service-account-update command line option, 796
gitlab-shared-project command line option, 796
gitlab-shared-project-list command line option, 796
gitlab-snippet command line option, 797
gitlab-snippet-content command line option, 797
gitlab-snippet-create command line option, 797
gitlab-snippet-delete command line option, 798
gitlab-snippet-get command line option, 798
gitlab-snippet-list command line option, 798
gitlab-snippet-list-all command line option, 799
gitlab-snippet-list-public command line option, 799
gitlab-snippet-update command line option, 799
gitlab-snippet-user-agent-detail command line option, 799
gitlab-starred-project command line option, 800
gitlab-starred-project-list command line option, 800
gitlab-todo command line option, 801
gitlab-todo-delete command line option, 801
gitlab-todo-list command line option, 801
gitlab-todo-mark-all-as-done command line option, 802
gitlab-todo-mark-as-done command line option, 802
gitlab-topic command line option, 802
gitlab-topic-create command line option, 802
gitlab-topic-delete command line option, 802
gitlab-topic-get command line option, 803
gitlab-topic-list command line option, 803
gitlab-topic-merge command line option, 803
gitlab-topic-update command line option, 804
gitlab-user command line option, 804
gitlab-user-activate command line option, 804
gitlab-user-activities command line option, 810
gitlab-user-activities-list command line option, 810
gitlab-user-approve command line option, 804
gitlab-user-ban command line option, 804
gitlab-user-block command line option, 805
gitlab-user-contributed-project command line option, 811
gitlab-user-contributed-project-list command line option, 811
gitlab-user-create command line option, 805
gitlab-user-custom-attribute command line option, 811
gitlab-user-custom-attribute-delete command line option, 811
gitlab-user-custom-attribute-get command line option, 812
gitlab-user-custom-attribute-list command line option, 812
gitlab-user-deactivate command line option, 806
gitlab-user-delete command line option, 806
gitlab-user-email command line option, 812
gitlab-user-email-create command line option, 812
gitlab-user-email-delete command line option, 813
gitlab-user-email-get command line option, 813
gitlab-user-email-list command line option, 813
gitlab-user-event command line option, 814

gitlab-user-event-list command line option, 814
gitlab-user-follow command line option, 806
gitlab-user-get command line option, 807
gitlab-user-gpg-key command line option, 814
gitlab-user-gpg-key-create command line option, 814
gitlab-user-gpg-key-delete command line option, 815
gitlab-user-gpg-key-get command line option, 815
gitlab-user-gpg-key-list command line option, 815
gitlab-user-impersonation-token command line option, 816
gitlab-user-impersonation-token-create command line option, 816
gitlab-user-impersonation-token-delete command line option, 816
gitlab-user-impersonation-token-get command line option, 816
gitlab-user-impersonation-token-list command line option, 817
gitlab-user-key command line option, 817
gitlab-user-key-create command line option, 817
gitlab-user-key-delete command line option, 817
gitlab-user-key-get command line option, 818
gitlab-user-key-list command line option, 818
gitlab-user-list command line option, 807
gitlab-user-membership command line option, 818
gitlab-user-membership-get command line option, 818
gitlab-user-membership-list command line option, 819
gitlab-user-personal-access-token command line option, 819
gitlab-user-personal-access-token-create command line option, 819
gitlab-user-project command line option, 820
gitlab-user-project-create command line option, 820
gitlab-user-project-list command line option, 821
gitlab-user-reject command line option, 808
gitlab-user-status command line option, 822
gitlab-user-status-get command line option, 822
gitlab-user-unban command line option, 808
gitlab-user-unblock command line option, 809
gitlab-user-unfollow command line option, 809
gitlab-user-update command line option, 809
gitlab-variable command line option, 822
gitlab-variable-create command line option, 822
gitlab-variable-delete command line option, 823
gitlab-variable-get command line option, 823
gitlab-variable-list command line option, 823
gitlab-variable-update command line option, 824
-o
gitlab command line option, 464
-v
gitlab command line option, 464

A

access_tokens (*gitlab.v4.objects.Group attribute*), 227
access_tokens (*gitlab.v4.objects.GroupServiceAccount attribute*), 262
access_tokens (*gitlab.v4.objects.Project attribute*), 275
access_tokens (*gitlab.v4.objects.ProjectServiceAccount attribute*), 387
AccessLevel (*class in gitlab.const*), 432
AccessRequestMixin (*class in gitlab.mixins*), 450
accessrequests (*gitlab.v4.objects.Group attribute*), 227
accessrequests (*gitlab.v4.objects.Project attribute*), 275
activate() (*gitlab.v4.objects.User method*), 413
add_spent_time() (*gitlab.mixins.TimeTrackingMixin method*), 457
additionalstatistics (*gitlab.v4.objects.Project attribute*), 275
ADMIN (*gitlab.const.AccessLevel attribute*), 432
all() (*gitlab.v4.objects.RunnerManager method*), 404
allowlist (*gitlab.v4.objects.ProjectJobTokenScope attribute*), 332
Any (*class in gitlab.v4.objects*), 208
api_url (*gitlab.Gitlab property*), 198
api_version (*gitlab.Gitlab property*), 198
appearance (*gitlab.Gitlab attribute*), 198
Application (*class in gitlab.v4.objects*), 208

ApplicationAppearance (*class in gitlab.v4.objects*), 209
ApplicationAppearanceManager (*class in gitlab.v4.objects*), 209
ApplicationManager (*class in gitlab.v4.objects*), 210
applications (*gitlab.Gitlab attribute*), 198
ApplicationSettings (*class in gitlab.v4.objects*), 210
ApplicationSettingsManager (*class in gitlab.v4.objects*), 210
ApplicationStatistics (*class in gitlab.v4.objects*), 212
ApplicationStatisticsManager (*class in gitlab.v4.objects*), 213
approval() (*gitlab.v4.objects.ProjectDeployment method*), 300
approval_rules (*gitlab.v4.objects.Group attribute*), 227
approval_rules (*gitlab.v4.objects.ProjectMergeRequest attribute*), 346
approval_state (*gitlab.v4.objects.ProjectMergeRequest attribute*), 346
approvalrules (*gitlab.v4.objects.Project attribute*), 275
approvals (*gitlab.v4.objects.Project attribute*), 275
approvals (*gitlab.v4.objects.ProjectMergeRequest attribute*), 346
approve() (*gitlab.mixins.AccessRequestMixin method*), 450
approve() (*gitlab.v4.objects.ProjectMergeRequest method*), 346
approve() (*gitlab.v4.objects.User method*), 413
archive() (*gitlab.v4.objects.Project method*), 276
artifact() (*gitlab.v4.objects.ProjectJob method*), 329
artifacts (*gitlab.v4.objects.Project attribute*), 276
artifacts() (*gitlab.v4.objects.ProjectJob method*), 329
asdict() (*gitlab.base.RESTObject method*), 428
attributes (*gitlab.base.RESTObject property*), 429
audit_events (*gitlab.Gitlab attribute*), 198
audit_events (*gitlab.v4.objects.Group attribute*), 227
audit_events (*gitlab.v4.objects.Project attribute*), 276
AuditEvent (*class in gitlab.v4.objects*), 213
AuditEventManager (*class in gitlab.v4.objects*), 213
auth() (*gitlab.Gitlab method*), 198
available() (*gitlab.v4.objects.ProjectIntegrationManager method*), 317
awardemojis (*gitlab.v4.objects.GroupEpicNote attribute*), 242
awardemojis (*gitlab.v4.objects.ProjectIssue attribute*), 318
awardemojis (*gitlab.v4.objects.ProjectIssueNote attribute*), 324
awardemojis (*gitlab.v4.objects.ProjectMergeRequest attribute*), 346
awardemojis (*gitlab.v4.objects.ProjectMergeRequestNote attribute*), 357
awardemojis (*gitlab.v4.objects.ProjectSnippet attribute*), 389
awardemojis (*gitlab.v4.objects.ProjectSnippetNote attribute*), 392

B

BadgeRenderMixin (*class in gitlab.mixins*), 451
badges (*gitlab.v4.objects.Group attribute*), 227
badges (*gitlab.v4.objects.Project attribute*), 276
ban() (*gitlab.v4.objects.User method*), 414
billable_members (*gitlab.v4.objects.Group attribute*), 227
blame() (*gitlab.v4.objects.ProjectFileManager method*), 310
BLOBS (*gitlab.const.SearchScope attribute*), 432
block() (*gitlab.v4.objects.User method*), 414
boards (*gitlab.v4.objects.Group attribute*), 227
boards (*gitlab.v4.objects.Project attribute*), 276
branch (*gitlab.v4.objects.ProjectFile attribute*), 308
branches (*gitlab.v4.objects.Project attribute*), 276
bridges (*gitlab.v4.objects.ProjectPipeline attribute*), 368
BroadcastMessage (*class in gitlab.v4.objects*), 213
BroadcastMessageManager (*class in gitlab.v4.objects*), 214
broadcastmessages (*gitlab.Gitlab attribute*), 198
bulk_import_entities (*gitlab.Gitlab attribute*), 199
bulk_imports (*gitlab.Gitlab attribute*), 199
bulk_publish() (*gitlab.v4.objects.ProjectMergeRequestDraftNoteManag method*), 355
BulkImport (*class in gitlab.v4.objects*), 214
BulkImportAllEntity (*class in gitlab.v4.objects*), 215
BulkImportAllEntityManager (*class in gitlab.v4.objects*), 215
BulkImportEntity (*class in gitlab.v4.objects*), 215
BulkImportEntityManager (*class in gitlab.v4.objects*), 215
BulkImportManager (*class in gitlab.v4.objects*), 216

C

cancel() (*gitlab.v4.objects.ProjectJob method*), 330
cancel() (*gitlab.v4.objects.ProjectPipeline method*), 368
cancel_merge_when_pipeline_succeeds() (*gitlab.v4.objects.ProjectMergeRequest method*), 346
changes() (*gitlab.v4.objects.ProjectMergeRequest method*), 346
cherry_pick() (*gitlab.v4.objects.ProjectCommit method*), 293
ci_lint (*gitlab.Gitlab attribute*), 199
ci_lint (*gitlab.v4.objects.Project attribute*), 276
CiLint (*class in gitlab.v4.objects*), 216
CiLintManager (*class in gitlab.v4.objects*), 216

closed_by() (*gitlab.v4.objects.ProjectIssue method*), 318
closes_issues() (*gitlab.v4.objects.ProjectMergeRequest method*), 347
cls_to_gitlab_resource() (*in module gitlab.cli*), 430
cluster_agents (*gitlab.v4.objects.Project attribute*), 276
clusters (*gitlab.v4.objects.Group attribute*), 227
clusters (*gitlab.v4.objects.Project attribute*), 276
comments (*gitlab.v4.objects.ProjectCommit attribute*), 294
commit_message (*gitlab.v4.objects.ProjectFile attribute*), 308
COMMITS (*gitlab.const.SearchScope attribute*), 432
commits (*gitlab.v4.objects.Project attribute*), 276
commits() (*gitlab.v4.objects.ProjectMergeRequest method*), 347
compound_metrics() (*gitlab.v4.objects.SidekiqManager method*), 406
ConfigError, 431
content (*gitlab.v4.objects.ProjectFile attribute*), 308
content() (*gitlab.v4.objects.ProjectSnippet method*), 389
content() (*gitlab.v4.objects.Snippet method*), 407
contributed_projects (*gitlab.v4.objects.User attribute*), 414
copy_dict() (*in module gitlab.utils*), 461
create() (*gitlab.mixins.CreateMixin method*), 451
create() (*gitlab.v4.objects.GroupClusterManager method*), 236
create() (*gitlab.v4.objects.GroupEpicIssueManager method*), 241
create() (*gitlab.v4.objects.ProjectClusterManager method*), 293
create() (*gitlab.v4.objects.ProjectCommitStatusManager method*), 299
create() (*gitlab.v4.objects.ProjectFileManager method*), 310
create() (*gitlab.v4.objects.ProjectForkManager method*), 314
create() (*gitlab.v4.objects.ProjectIssueLinkManager method*), 322
create() (*gitlab.v4.objects.ProjectPipelineManager method*), 370
create() (*gitlab.v4.objects.ProjectPullMirrorManager method*), 377
create_fork_relation() (*gitlab.v4.objects.Project method*), 276
CreateMixin (*class in gitlab.mixins*), 451
CRUDMixin (*class in gitlab.mixins*), 451
current_failures() (*gitlab.v4.objects.GeoNodeManager method*), 226
current_page (*gitlab.base.RESTObjectList property*), 429
current_page (*gitlab.GitlabList property*), 208
CurrentUser (*class in gitlab.v4.objects*), 217
CurrentUserEmail (*class in gitlab.v4.objects*), 217
CurrentUserEmailManager (*class in gitlab.v4.objects*), 217
CurrentUserGPGKey (*class in gitlab.v4.objects*), 217
CurrentUserGPGKeyManager (*class in gitlab.v4.objects*), 218
CurrentUserKey (*class in gitlab.v4.objects*), 218
CurrentUserKeyManager (*class in gitlab.v4.objects*), 218
CurrentUserManager (*class in gitlab.v4.objects*), 218
CurrentUserRunner (*class in gitlab.v4.objects*), 219
CurrentUserRunnerManager (*class in gitlab.v4.objects*), 219
CurrentUserStatus (*class in gitlab.v4.objects*), 219
CurrentUserStatusManager (*class in gitlab.v4.objects*), 220
CUSTOM (*gitlab.const.NotificationLevel attribute*), 432
CustomAction (*class in gitlab.cli*), 430
customattributes (*gitlab.v4.objects.Group attribute*), 227
customattributes (*gitlab.v4.objects.Project attribute*), 276
customattributes (*gitlab.v4.objects.User attribute*), 414

D

deactivate() (*gitlab.v4.objects.User method*), 414
decode() (*gitlab.v4.objects.ProjectFile method*), 308
delete() (*gitlab.mixins.DeleteMixin method*), 452
delete() (*gitlab.mixins.ObjectDeleteMixin method*), 455
delete() (*gitlab.v4.objects.GroupLDAPGroupLink method*), 249
delete() (*gitlab.v4.objects.ProjectArtifactManager method*), 286
delete() (*gitlab.v4.objects.ProjectFile method*), 309
delete() (*gitlab.v4.objects.ProjectFileManager method*), 311
delete_artifacts() (*gitlab.v4.objects.ProjectJob method*), 330
delete_fork_relation() (*gitlab.v4.objects.Project method*), 276
delete_in_bulk() (*gitlab.v4.objects.ProjectRegistryTagManager method*), 381
delete_merged_branches() (*gitlab.v4.objects.RepositoryMixin method*), 398

DeleteMixin (*class in gitlab.mixins*), 452
DeployKey (*class in gitlab.v4.objects*), 220
DeployKeyManager (*class in gitlab.v4.objects*), 220
deploykeys (*gitlab.Gitlab attribute*), 199
deployments (*gitlab.v4.objects.Project attribute*), 277
DeployToken (*class in gitlab.v4.objects*), 221
DeployTokenManager (*class in gitlab.v4.objects*), 221
deploytokens (*gitlab.Gitlab attribute*), 199
deploytokens (*gitlab.v4.objects.Group attribute*), 227
deploytokens (*gitlab.v4.objects.Project attribute*), 277
descendant_groups (*gitlab.v4.objects.Group attribute*), 227
DEVELOPER (*gitlab.const.AccessLevel attribute*), 432
die() (*in module gitlab.cli*), 430
diff() (*gitlab.v4.objects.ProjectCommit method*), 294
diffs (*gitlab.v4.objects.ProjectMergeRequest attribute*), 347
DISABLED (*gitlab.const.NotificationLevel attribute*), 432
discussions (*gitlab.v4.objects.ProjectCommit attribute*), 294
discussions (*gitlab.v4.objects.ProjectIssue attribute*), 319
discussions (*gitlab.v4.objects.ProjectMergeRequest attribute*), 347
discussions (*gitlab.v4.objects.ProjectSnippet attribute*), 390
Dockerfile (*class in gitlab.v4.objects*), 221
dockerfile_templates (*gitlab.v4.objects.Project attribute*), 277
DockerfileManager (*class in gitlab.v4.objects*), 221
dockerfiles (*gitlab.Gitlab attribute*), 199
docs() (*in module gitlab.cli*), 431
download() (*gitlab.mixins.DownloadMixin method*), 452
download() (*gitlab.v4.objects.GenericPackageManager method*), 223
download() (*gitlab.v4.objects.ProjectArtifactManager method*), 286
download() (*gitlab.v4.objects.ProjectSecureFile method*), 386
DownloadMixin (*class in gitlab.mixins*), 452
draft_notes (*gitlab.v4.objects.ProjectMergeRequest attribute*), 347

E

emails (*gitlab.v4.objects.CurrentUser attribute*), 217
emails (*gitlab.v4.objects.User attribute*), 414
enable() (*gitlab.v4.objects.ProjectKeyManager method*), 333
enable_debug() (*gitlab.Gitlab method*), 199
encoded_id (*gitlab.base.RESTObject property*), 429
EncodedId (*class in gitlab.utils*), 460
entities (*gitlab.v4.objects.BulkImport attribute*), 215
environments (*gitlab.v4.objects.Project attribute*), 277
epics (*gitlab.v4.objects.Group attribute*), 227
erase() (*gitlab.v4.objects.ProjectJob method*), 330
Event (*class in gitlab.v4.objects*), 221
EventManager (*class in gitlab.v4.objects*), 222
events (*gitlab.Gitlab attribute*), 199
events (*gitlab.v4.objects.Project attribute*), 277
events (*gitlab.v4.objects.User attribute*), 414
exists() (*gitlab.v4.objects.NamespaceManager method*), 273
exports (*gitlab.v4.objects.Group attribute*), 227
exports (*gitlab.v4.objects.Project attribute*), 277
external_status_checks (*gitlab.v4.objects.Project attribute*), 277

F

Feature (*class in gitlab.v4.objects*), 222
feature_flags (*gitlab.v4.objects.Project attribute*), 277
feature_flags_user_lists (*gitlab.v4.objects.Project attribute*), 277
FeatureManager (*class in gitlab.v4.objects*), 222
features (*gitlab.Gitlab attribute*), 199
file_path (*gitlab.v4.objects.ProjectFile attribute*), 309
files (*gitlab.v4.objects.Project attribute*), 277
follow() (*gitlab.v4.objects.User method*), 414
followers_users (*gitlab.v4.objects.User attribute*), 415
following_users (*gitlab.v4.objects.User attribute*), 415
forks (*gitlab.v4.objects.Project attribute*), 277
format() (*gitlab.utils.MaskingFormatter method*), 460
from_config() (*gitlab.Gitlab class method*), 199

G

generic_packages (*gitlab.v4.objects.Project attribute*), 277
GenericPackage (*class in gitlab.v4.objects*), 223
GenericPackageManager (*class in gitlab.v4.objects*), 223
GeoNode (*class in gitlab.v4.objects*), 224
GeoNodeManager (*class in gitlab.v4.objects*), 225
geonodes (*gitlab.Gitlab attribute*), 200
get() (*gitlab.mixins.GetMixin method*), 453
get() (*gitlab.mixins.GetWithoutIdMixin method*), 453
get() (*gitlab.v4.objects.KeyManager method*), 268
get() (*gitlab.v4.objects.ProjectFileManager method*), 311
get_base_url() (*in module gitlab.utils*), 461
get_content_type() (*in module gitlab.utils*), 462
get_id() (*gitlab.base.RESTObject method*), 429
get_license() (*gitlab.Gitlab method*), 200
GetMixin (*class in gitlab.mixins*), 453
GetWithoutIdMixin (*class in gitlab.mixins*), 453
Gitignore (*class in gitlab.v4.objects*), 226

gitignore_templates (*gitlab.v4.objects.Project attribute*), 277
GitignoreManager (*class in gitlab.v4.objects*), 226
gitignores (*gitlab.Gitlab attribute*), 200
Gitlab (*class in gitlab*), 197
gitlab (*gitlab.base.RESTManager attribute*), 428
gitlab command line option
    --api-version, 465
    --config-file, 464
    --debug, 464
    --fancy, 464
    --fields, 464
    --gitlab, 464
    --help, 464
    --job-token, 465
    --no-mask-credentials, 465
    --no-ssl-verify, 465
    --oauth-token, 465
    --order-by, 465
    --output, 464
    --pagination, 465
    --per-page, 465
    --private-token, 465
    --server-url, 464
    --skip-login, 465
    --ssl-verify, 464
    --timeout, 465
    --user-agent, 465
    --verbose, 464
    --version, 464
    -c, 464
    -d, 464
    -f, 464
    -g, 464
    -h, 464
    -o, 464
    -v, 464
gitlab.base
    module, 428
gitlab.cli
    module, 430
gitlab.config
    module, 431
gitlab.const
    module, 432
gitlab.exceptions
    module, 433
gitlab.mixins
    module, 450
gitlab.utils
    module, 460
gitlab.v4
    module, 428
gitlab.v4.objects
    module, 208
gitlab_resource_to_cls() (*in module gitlab.cli*), 431
gitlab-application command line option
    --help, 465
    -h, 465
gitlab-application-appearance command line option
    --help, 466
    -h, 466
gitlab-application-appearance-get command line option
    --help, 466
    --sudo, 466
    -h, 466
gitlab-application-appearance-update command line option
    --description, 467
    --email-header-and-footer-enabled, 467
    --favicon, 467
    --footer-message, 467
    --header-logo, 467
    --header-message, 467
    --help, 467
    --logo, 467
    --message-background-color, 467
    --message-font-color, 467
    --new-project-guidelines, 467
    --sudo, 467
    --title, 467
    -h, 467
gitlab-application-create command line option
    --confidential, 466
    --help, 465
    --name, 466
    --redirect-uri, 466
    --scopes, 466
    --sudo, 465
    -h, 465
gitlab-application-delete command line option
    --help, 466
    --id, 466
    --sudo, 466
    -h, 466
gitlab-application-list command line option
    --get-all, 466
    --help, 466
    --no-get-all, 466
    --page, 466
    --per-page, 466
    --sudo, 466
    -h, 466

gitlab-application-settings command line option
--help, 467
-h, 467
gitlab-application-settings-get command line option
--help, 467
--sudo, 468
-h, 467
gitlab-application-settings-update command line option
--after-sign-out-path, 470
--allow-local-requests-from-hooks-and-services, 471
--allow-local-requests-from-system-hooks, 471
--allow-local-requests-from-web-hooks-and-services, 471
--asset-proxy-allowlist, 471
--asset-proxy-enabled, 471
--asset-proxy-url, 471
--asset-proxy-whitelist, 471
--container-registry-token-expire-delay, 470
--created-at, 470
--default-branch-protection, 470
--default-group-visibility, 470
--default-project-visibility, 470
--default-projects-limit, 469
--default-snippet-visibility, 470
--disabled-oauth-sign-in-sources, 470
--domain-allowlist, 470
--domain-blacklist, 470
--domain-blacklist-enabled, 470
--domain-denylist, 470
--domain-denylist-enabled, 470
--domain-whitelist, 470
--dsa-key-restriction, 471
--ecdsa-key-restriction, 471
--ed25519-key-restriction, 471
--enforce-terms, 471
--external-authorization-service-default-label, 470
--external-authorization-service-enabled, 470
--external-authorization-service-timeout, 470
--external-authorization-service-url, 470
--file-template-project-id, 471
--first-day-of-week, 471
--geo-node-allowed-ips, 471
--gravatar-enabled, 470
--help, 469
--home-page-url, 470
--id, 469
--import-sources, 470
--instance-statistics-visibility-private, 471
--local-markdown-version, 471
--max-attachment-size, 470
--outbound-local-requests-whitelist, 470
--password-authentication-enabled-for-web, 469
--performance-bar-allowed-group-id, 471
--plantuml-enabled, 470
--plantuml-url, 471
--polling-interval-multiplier, 471
--repository-storages, 470
--restricted-visibility-levels, 470
--rsa-key-restriction, 471
--session-expire-delay, 470
--sign-in-text, 470
--signup-enabled, 469
--silent-mode-enabled, 469
--sudo, 469
--terminal-max-session-time, 471
--terms, 471
--updated-at, 470
--user-oauth-applications, 470
--user-show-add-ssh-key-message, 471
-h, 469
gitlab-application-statistics command line option
--help, 471
-h, 471
gitlab-application-statistics-get command line option
--help, 472
--sudo, 472
-h, 472
gitlab-audit-event command line option
--help, 472
-h, 472
gitlab-audit-event-get command line option
--help, 472
--id, 472
--sudo, 472
-h, 472
gitlab-audit-event-list command line option
--created-after, 472
--created-before, 472
--entity-id, 472
--entity-type, 472
--get-all, 472
--help, 472
--no-get-all, 472
--page, 472
--per-page, 472

--sudo, 472
-h, 472
gitlab-broadcast-message command line option
--help, 473
-h, 473
gitlab-broadcast-message-create command line option
--color, 473
--ends-at, 473
--font, 473
--help, 473
--message, 473
--starts-at, 473
--sudo, 473
--target-access-levels, 473
-h, 473
gitlab-broadcast-message-delete command line option
--help, 473
--id, 473
--sudo, 473
-h, 473
gitlab-broadcast-message-get command line option
--help, 473
--id, 473
--sudo, 473
-h, 473
gitlab-broadcast-message-list command line option
--get-all, 474
--help, 474
--no-get-all, 474
--page, 474
--per-page, 474
--sudo, 474
-h, 474
gitlab-broadcast-message-update command line option
--color, 474
--ends-at, 474
--font, 474
--help, 474
--id, 474
--message, 474
--starts-at, 474
--sudo, 474
--target-access-levels, 474
-h, 474
gitlab-bulk-import command line option
--help, 474
-h, 474
gitlab-bulk-import-all-entity command line option
--help, 475
-h, 475
gitlab-bulk-import-all-entity-list command line option
--get-all, 476
--help, 476
--no-get-all, 476
--page, 476
--per-page, 476
--sort, 476
--status, 476
--sudo, 476
-h, 476
gitlab-bulk-import-create command line option
--configuration, 475
--entities, 475
--help, 475
--sudo, 475
-h, 475
gitlab-bulk-import-entity command line option
--help, 476
-h, 476
gitlab-bulk-import-entity-get command line option
--bulk-import-id, 476
--help, 476
--id, 476
--sudo, 476
-h, 476
gitlab-bulk-import-entity-list command line option
--bulk-import-id, 477
--get-all, 477
--help, 476
--no-get-all, 477
--page, 477
--per-page, 477
--sort, 477
--status, 477
--sudo, 477
-h, 476
gitlab-bulk-import-get command line option
--help, 475
--id, 475
--sudo, 475
-h, 475
gitlab-bulk-import-list command line option
--get-all, 475
--help, 475
--no-get-all, 475

--page, 475
--per-page, 475
--sort, 475
--status, 475
--sudo, 475
-h, 475
gitlab-ci-lint command line option
--help, 477
-h, 477
gitlab-ci-lint-create command line option
--content, 477
--help, 477
--include-jobs, 477
--include-merged-yaml, 477
--sudo, 477
-h, 477
gitlab-ci-lint-validate command line option
--content, 477
--help, 477
--include-jobs, 478
--include-merged-yaml, 477
-h, 477
gitlab-current-user command line option
--help, 478
-h, 478
gitlab-current-user-email command line option
--help, 478
-h, 478
gitlab-current-user-email-create command line option
--email, 478
--help, 478
--sudo, 478
-h, 478
gitlab-current-user-email-delete command line option
--help, 478
--id, 478
--sudo, 478
-h, 478
gitlab-current-user-email-get command line option
--help, 478
--id, 479
--sudo, 479
-h, 478
gitlab-current-user-email-list command line option
--get-all, 479
--help, 479
--no-get-all, 479
--page, 479
--per-page, 479
--sudo, 479
-h, 479
gitlab-current-user-get command line option
--help, 478
--sudo, 478
-h, 478
gitlab-current-user-gpg-key command line option
--help, 479
-h, 479
gitlab-current-user-gpg-key-create command line option
--help, 479
--key, 479
--sudo, 479
-h, 479
gitlab-current-user-gpg-key-delete command line option
--help, 479
--id, 479
--sudo, 479
-h, 479
gitlab-current-user-gpg-key-get command line option
--help, 480
--id, 480
--sudo, 480
-h, 480
gitlab-current-user-gpg-key-list command line option
--get-all, 480
--help, 480
--no-get-all, 480
--page, 480
--per-page, 480
--sudo, 480
-h, 480
gitlab-current-user-key command line option
--help, 480
-h, 480
gitlab-current-user-key-create command line option
--help, 480
--key, 480
--sudo, 480
--title, 480
-h, 480
gitlab-current-user-key-delete command line option
--help, 481
--id, 481
--sudo, 481
-h, 481

gitlab-current-user-key-get command line option
--help, 481
--id, 481
--sudo, 481
-h, 481
gitlab-current-user-key-list command line option
--get-all, 481
--help, 481
--no-get-all, 481
--page, 481
--per-page, 481
--sudo, 481
-h, 481
gitlab-current-user-runner command line option
--help, 481
-h, 481
gitlab-current-user-runner-create command line option
--access-level, 482
--description, 482
--group-id, 482
--help, 482
--locked, 482
--maintenance-note, 482
--maximum-timeout, 482
--paused, 482
--project-id, 482
--run-untagged, 482
--runner-type, 482
--sudo, 482
--tag-list, 482
-h, 482
gitlab-current-user-status command line option
--help, 482
-h, 482
gitlab-current-user-status-get command line option
--help, 482
--sudo, 482
-h, 482
gitlab-current-user-status-update command line option
--emoji, 483
--help, 483
--message, 483
--sudo, 483
-h, 483
gitlab-deploy-key command line option
--help, 483
-h, 483
gitlab-deploy-key-create command line option
--expires-at, 483
--help, 483
--key, 483
--sudo, 483
--title, 483
-h, 483
gitlab-deploy-key-list command line option
--get-all, 483
--help, 483
--no-get-all, 483
--page, 483
--per-page, 483
--sudo, 483
-h, 483
gitlab-deploy-token command line option
--help, 484
-h, 484
gitlab-deploy-token-list command line option
--get-all, 484
--help, 484
--no-get-all, 484
--page, 484
--per-page, 484
--sudo, 484
-h, 484
gitlab-dockerfile command line option
--help, 484
-h, 484
gitlab-dockerfile-get command line option
--help, 484
--name, 484
--sudo, 484
-h, 484
gitlab-dockerfile-list command line option
--get-all, 485
--help, 484
--no-get-all, 485
--page, 485
--per-page, 485
--sudo, 484
-h, 484
gitlab-event command line option
--help, 485
-h, 485
gitlab-event-list command line option
--action, 485
--after, 485
--before, 485
--get-all, 485
--help, 485
--no-get-all, 485

--page, 485
--per-page, 485
--scope, 485
--sort, 485
--sudo, 485
--target-type, 485
-h, 485
gitlab-feature command line option
--help, 486
-h, 486
gitlab-feature-delete command line option
--help, 486
--name, 486
--sudo, 486
-h, 486
gitlab-feature-list command line option
--get-all, 486
--help, 486
--no-get-all, 486
--page, 486
--per-page, 486
--sudo, 486
-h, 486
gitlab-generic-package command line option
--help, 486
-h, 486
gitlab-generic-package-download command line option
--file-name, 487
--help, 486
--package-name, 487
--package-version, 487
--project-id, 486
--sudo, 487
-h, 486
gitlab-generic-package-upload command line option
--file-name, 487
--help, 487
--package-name, 487
--package-version, 487
--path, 487
--project-id, 487
--sudo, 487
-h, 487
gitlab-geo-node command line option
--help, 487
-h, 487
gitlab-geo-node-current-failures command line option
--help, 487
--id, 487
-h, 487
gitlab-geo-node-delete command line option
--help, 487
--id, 488
--sudo, 487
-h, 487
gitlab-geo-node-get command line option
--help, 488
--id, 488
--sudo, 488
-h, 488
gitlab-geo-node-list command line option
--get-all, 488
--help, 488
--no-get-all, 488
--page, 488
--per-page, 488
--sudo, 488
-h, 488
gitlab-geo-node-repair command line option
--help, 488
--id, 488
-h, 488
gitlab-geo-node-status command line option
--help, 488
--id, 488
-h, 488
gitlab-geo-node-update command line option
--enabled, 489
--files-max-capacity, 489
--help, 489
--id, 489
--repos-max-capacity, 489
--sudo, 489
--url, 489
-h, 489
gitlab-gitignore command line option
--help, 489
-h, 489
gitlab-gitignore-get command line option
--help, 489
--name, 489
--sudo, 489
-h, 489
gitlab-gitignore-list command line option
--get-all, 489
--help, 489
--no-get-all, 489
--page, 489
--per-page, 489
--sudo, 489
-h, 489
gitlab-gitlabciyml command line option
--help, 490
-h, 490
gitlab-gitlabciyml-get command line option

--help, 490
--name, 490
--sudo, 490
-h, 490
gitlab-gitlabciyml-list command line option
--get-all, 490
--help, 490
--no-get-all, 490
--page, 490
--per-page, 490
--sudo, 490
-h, 490
gitlab-group command line option
--help, 490
-h, 490
gitlab-group-access-request command line option
--help, 496
-h, 496
gitlab-group-access-request-approve command line option
--access-level, 496
--group-id, 496
--help, 496
--id, 496
--sudo, 496
-h, 496
gitlab-group-access-request-create command line option
--group-id, 496
--help, 496
--sudo, 496
-h, 496
gitlab-group-access-request-delete command line option
--group-id, 497
--help, 497
--id, 497
--sudo, 497
-h, 497
gitlab-group-access-request-list command line option
--get-all, 497
--group-id, 497
--help, 497
--no-get-all, 497
--page, 497
--per-page, 497
--sudo, 497
-h, 497
gitlab-group-access-token command line option
--help, 497
-h, 497
gitlab-group-access-token-create command line option
--access-level, 498
--expires-at, 498
--group-id, 497
--help, 497
--name, 498
--scopes, 498
--sudo, 497
-h, 497
gitlab-group-access-token-delete command line option
--group-id, 498
--help, 498
--id, 498
--sudo, 498
-h, 498
gitlab-group-access-token-get command line option
--group-id, 498
--help, 498
--id, 498
--sudo, 498
-h, 498
gitlab-group-access-token-list command line option
--get-all, 498
--group-id, 498
--help, 498
--no-get-all, 498
--page, 498
--per-page, 498
--sudo, 498
-h, 498
gitlab-group-access-token-rotate command line option
--expires-at, 499
--group-id, 499
--help, 499
--id, 499
--sudo, 499
-h, 499
gitlab-group-approval-rule command line option
--help, 499
-h, 499
gitlab-group-approval-rule-create command line option
--approvals-required, 499
--group-id, 499
--group-ids, 499
--help, 499
--name, 499
--rule-type, 499

--sudo, 499
--user-ids, 499
-h, 499
gitlab-group-approval-rule-get command line option
--group-id, 500
--help, 500
--id, 500
--sudo, 500
-h, 500
gitlab-group-approval-rule-list command line option
--get-all, 500
--group-id, 500
--help, 500
--no-get-all, 500
--page, 500
--per-page, 500
--sudo, 500
-h, 500
gitlab-group-approval-rule-update command line option
--group-id, 500
--help, 500
--id, 500
--sudo, 500
-h, 500
gitlab-group-audit-event command line option
--help, 500
-h, 500
gitlab-group-audit-event-get command line option
--group-id, 501
--help, 501
--id, 501
--sudo, 501
-h, 501
gitlab-group-audit-event-list command line option
--created-after, 501
--created-before, 501
--get-all, 501
--group-id, 501
--help, 501
--no-get-all, 501
--page, 501
--per-page, 501
--sudo, 501
-h, 501
gitlab-group-badge command line option
--help, 501
-h, 501
gitlab-group-badge-create command line option
--group-id, 502
--help, 501
--image-url, 502
--link-url, 502
--sudo, 502
-h, 501
gitlab-group-badge-delete command line option
--group-id, 502
--help, 502
--id, 502
--sudo, 502
-h, 502
gitlab-group-badge-get command line option
--group-id, 502
--help, 502
--id, 502
--sudo, 502
-h, 502
gitlab-group-badge-list command line option
--get-all, 502
--group-id, 502
--help, 502
--no-get-all, 502
--page, 502
--per-page, 502
--sudo, 502
-h, 502
gitlab-group-badge-render command line option
--group-id, 503
--help, 503
--id, 503
--image-url, 503
--link-url, 503
--sudo, 503
-h, 503
gitlab-group-badge-update command line option
--group-id, 503
--help, 503
--id, 503
--image-url, 503
--link-url, 503
--sudo, 503
-h, 503
gitlab-group-billable-member command line option
--help, 503
-h, 503
gitlab-group-billable-member-delete command line option

--group-id, 504
--help, 503
--id, 504
--sudo, 503
-h, 503
gitlab-group-billable-member-list command line option
--get-all, 504
--group-id, 504
--help, 504
--no-get-all, 504
--page, 504
--per-page, 504
--search, 504
--sort, 504
--sudo, 504
-h, 504
gitlab-group-billable-member-membership command line option
--help, 504
-h, 504
gitlab-group-billable-member-membership-list command line option
--get-all, 505
--group-id, 504
--help, 504
--no-get-all, 505
--page, 505
--per-page, 505
--sudo, 504
--user-id, 505
-h, 504
gitlab-group-board command line option
--help, 505
-h, 505
gitlab-group-board-create command line option
--group-id, 505
--help, 505
--name, 505
--sudo, 505
-h, 505
gitlab-group-board-delete command line option
--group-id, 505
--help, 505
--id, 505
--sudo, 505
-h, 505
gitlab-group-board-get command line option
--group-id, 506
--help, 505
--id, 506
--sudo, 505
-h, 505
gitlab-group-board-list command line option
--get-all, 506
--group-id, 506
--help, 506
--no-get-all, 506
--page, 506
--per-page, 506
--sudo, 506
-h, 506
gitlab-group-board-list-create command line option
--assignee-id, 507
--board-id, 507
--group-id, 507
--help, 507
--iteration-id, 507
--label-id, 507
--milestone-id, 507
--sudo, 507
-h, 507
gitlab-group-board-list-delete command line option
--board-id, 507
--group-id, 507
--help, 507
--id, 507
--sudo, 507
-h, 507
gitlab-group-board-list-get command line option
--board-id, 507
--group-id, 507
--help, 507
--id, 507
--sudo, 507
-h, 507
gitlab-group-board-list-list command line option
--board-id, 508
--get-all, 508
--group-id, 508
--help, 508
--no-get-all, 508
--page, 508
--per-page, 508
--sudo, 508
-h, 508
gitlab-group-board-list-update command line option
--board-id, 508
--group-id, 508
--help, 508
--id, 508

--position, 508
--sudo, 508
-h, 508
gitlab-group-board-update command line option
--group-id, 506
--help, 506
--id, 506
--sudo, 506
-h, 506
gitlab-group-cluster command line option
--help, 508
-h, 508
gitlab-group-cluster-create command line option
--domain, 509
--enabled, 509
--environment-scope, 509
--group-id, 509
--help, 509
--managed, 509
--name, 509
--platform-kubernetes-attributes, 509
--sudo, 509
-h, 509
gitlab-group-cluster-delete command line option
--group-id, 509
--help, 509
--id, 509
--sudo, 509
-h, 509
gitlab-group-cluster-get command line option
--group-id, 509
--help, 509
--id, 509
--sudo, 509
-h, 509
gitlab-group-cluster-list command line option
--get-all, 510
--group-id, 510
--help, 510
--no-get-all, 510
--page, 510
--per-page, 510
--sudo, 510
-h, 510
gitlab-group-cluster-update command line option
--domain, 510
--environment-scope, 510
--group-id, 510
--help, 510
--id, 510
--management-project-id, 510
--name, 510
--platform-kubernetes-attributes, 510
--sudo, 510
-h, 510
gitlab-group-create command line option
--auto-devops-enabled, 491
--avatar, 491
--default-branch-protection, 492
--description, 491
--emails-disabled, 491
--extra-shared-runners-minutes-limit, 492
--help, 491
--lfs-enabled, 491
--membership-lock, 491
--mentions-disabled, 491
--name, 491
--parent-id, 492
--path, 491
--project-creation-level, 491
--request-access-enabled, 492
--require-two-factor-authentication, 491
--share-with-group-lock, 491
--shared-runners-minutes-limit, 492
--subgroup-creation-level, 491
--sudo, 491
--two-factor-grace-period, 491
--visibility, 491
-h, 491
gitlab-group-custom-attribute command line option
--help, 510
-h, 510
gitlab-group-custom-attribute-delete command line option
--group-id, 511
--help, 511
--key, 511
--sudo, 511
-h, 511
gitlab-group-custom-attribute-get command line option
--group-id, 511
--help, 511
--key, 511
--sudo, 511
-h, 511
gitlab-group-custom-attribute-list command line option
--get-all, 511
--group-id, 511
--help, 511

--no-get-all, 511
--page, 511
--per-page, 511
--sudo, 511
-h, 511
gitlab-group-delete command line option
--help, 492
--id, 492
--sudo, 492
-h, 492
gitlab-group-deploy-token command line option
--help, 511
-h, 511
gitlab-group-deploy-token-create command line option
--expires-at, 512
--group-id, 512
--help, 512
--name, 512
--scopes, 512
--sudo, 512
--username, 512
-h, 512
gitlab-group-deploy-token-delete command line option
--group-id, 512
--help, 512
--id, 512
--sudo, 512
-h, 512
gitlab-group-deploy-token-get command line option
--group-id, 512
--help, 512
--id, 512
--sudo, 512
-h, 512
gitlab-group-deploy-token-list command line option
--get-all, 513
--group-id, 513
--help, 512
--no-get-all, 513
--page, 513
--per-page, 513
--scopes, 513
--sudo, 513
-h, 512
gitlab-group-descendant-group command line option
--help, 513
-h, 513
gitlab-group-descendant-group-list command line option
--all-available, 513
--get-all, 514
--group-id, 513
--help, 513
--min-access-level, 514
--no-get-all, 514
--order-by, 513
--owned, 514
--page, 514
--per-page, 514
--search, 513
--skip-groups, 513
--sort, 513
--statistics, 514
--sudo, 513
--with-custom-attributes, 514
-h, 513
gitlab-group-epic command line option
--help, 514
-h, 514
gitlab-group-epic-award-emoji command line option
--help, 516
-h, 516
gitlab-group-epic-award-emoji-create command line option
--epic-iid, 516
--group-id, 516
--help, 516
--name, 516
--sudo, 516
-h, 516
gitlab-group-epic-award-emoji-delete command line option
--epic-iid, 517
--group-id, 517
--help, 517
--id, 517
--sudo, 517
-h, 517
gitlab-group-epic-award-emoji-get command line option
--epic-iid, 517
--group-id, 517
--help, 517
--id, 517
--sudo, 517
-h, 517
gitlab-group-epic-award-emoji-list command line option
--epic-iid, 517
--get-all, 517

--group-id, 517
--help, 517
--no-get-all, 517
--page, 517
--per-page, 517
--sudo, 517
-h, 517
gitlab-group-epic-create command line option
--description, 514
--end-date, 514
--group-id, 514
--help, 514
--labels, 514
--start-date, 514
--sudo, 514
--title, 514
-h, 514
gitlab-group-epic-delete command line option
--group-id, 515
--help, 515
--iid, 515
--sudo, 515
-h, 515
gitlab-group-epic-discussion-note command line option
--help, 518
-h, 518
gitlab-group-epic-discussion-note-create command line option
--body, 518
--created-at, 518
--discussion-id, 518
--epic-id, 518
--group-id, 518
--help, 518
--sudo, 518
-h, 518
gitlab-group-epic-discussion-note-delete command line option
--discussion-id, 518
--epic-id, 518
--group-id, 518
--help, 518
--id, 518
--sudo, 518
-h, 518
gitlab-group-epic-discussion-note-get command line option
--discussion-id, 519
--epic-id, 519
--group-id, 519
--help, 519
--id, 519
--sudo, 519
-h, 519
gitlab-group-epic-discussion-note-update command line option
--body, 519
--discussion-id, 519
--epic-id, 519
--group-id, 519
--help, 519
--id, 519
--sudo, 519
-h, 519
gitlab-group-epic-get command line option
--group-id, 515
--help, 515
--iid, 515
--sudo, 515
-h, 515
gitlab-group-epic-issue command line option
--help, 519
-h, 519
gitlab-group-epic-issue-create command line option
--epic-iid, 520
--group-id, 520
--help, 519
--issue-id, 520
--sudo, 520
-h, 519
gitlab-group-epic-issue-delete command line option
--epic-iid, 520
--epic-issue-id, 520
--group-id, 520
--help, 520
--sudo, 520
-h, 520
gitlab-group-epic-issue-list command line option
--epic-iid, 520
--get-all, 520
--group-id, 520
--help, 520
--no-get-all, 520
--page, 520
--per-page, 520
--sudo, 520
-h, 520
gitlab-group-epic-issue-update command line option
--epic-iid, 521
--epic-issue-id, 521
--group-id, 521

--help, 521
--move-after-id, 521
--move-before-id, 521
--sudo, 521
-h, 521
gitlab-group-epic-list command line option
--author-id, 515
--get-all, 515
--group-id, 515
--help, 515
--labels, 515
--no-get-all, 516
--order-by, 515
--page, 515
--per-page, 515
--search, 515
--sort, 515
--sudo, 515
-h, 515
gitlab-group-epic-note command line option
--help, 521
-h, 521
gitlab-group-epic-note-award-emoji command line option
--help, 523
-h, 523
gitlab-group-epic-note-award-emoji-create command line option
--epic-iid, 523
--group-id, 523
--help, 523
--name, 523
--note-id, 523
--sudo, 523
-h, 523
gitlab-group-epic-note-award-emoji-delete command line option
--epic-iid, 523
--group-id, 523
--help, 523
--id, 524
--note-id, 523
--sudo, 523
-h, 523
gitlab-group-epic-note-award-emoji-get command line option
--epic-iid, 524
--group-id, 524
--help, 524
--id, 524
--note-id, 524
--sudo, 524
-h, 524
gitlab-group-epic-note-award-emoji-list command line option
--epic-iid, 524
--get-all, 524
--group-id, 524
--help, 524
--no-get-all, 524
--note-id, 524
--page, 524
--per-page, 524
--sudo, 524
-h, 524
gitlab-group-epic-note-create command line option
--body, 521
--created-at, 521
--epic-id, 521
--group-id, 521
--help, 521
--sudo, 521
-h, 521
gitlab-group-epic-note-delete command line option
--epic-id, 522
--group-id, 522
--help, 521
--id, 522
--sudo, 521
-h, 521
gitlab-group-epic-note-get command line option
--epic-id, 522
--group-id, 522
--help, 522
--id, 522
--sudo, 522
-h, 522
gitlab-group-epic-note-list command line option
--epic-id, 522
--get-all, 522
--group-id, 522
--help, 522
--no-get-all, 522
--page, 522
--per-page, 522
--sudo, 522
-h, 522
gitlab-group-epic-note-update command line option
--body, 523
--epic-id, 523
--group-id, 523
--help, 522

--id, 523
--sudo, 523
-h, 522
gitlab-group-epic-resource-label-event command line option
--help, 525
-h, 525
gitlab-group-epic-resource-label-event-get command line option
--epic-id, 525
--group-id, 525
--help, 525
--id, 525
--sudo, 525
-h, 525
gitlab-group-epic-resource-label-event-list command line option
--epic-id, 525
--get-all, 525
--group-id, 525
--help, 525
--no-get-all, 525
--page, 525
--per-page, 525
--sudo, 525
-h, 525
gitlab-group-epic-update command line option
--description, 516
--end-date, 516
--group-id, 516
--help, 516
--iid, 516
--labels, 516
--start-date, 516
--sudo, 516
--title, 516
-h, 516
gitlab-group-export command line option
--help, 525
-h, 525
gitlab-group-export-create command line option
--group-id, 526
--help, 526
--sudo, 526
-h, 526
gitlab-group-export-download command line option
--group-id, 526
--help, 526
--sudo, 526
-h, 526
gitlab-group-export-get command line option
--group-id, 526
--help, 526
--sudo, 526
-h, 526
gitlab-group-get command line option
--help, 492
--id, 492
--sudo, 492
-h, 492
gitlab-group-hook command line option
--help, 526
-h, 526
gitlab-group-hook-create command line option
--confidential-issues-events, 527
--confidential-note-events, 527
--deployment-events, 527
--enable-ssl-verification, 527
--group-id, 527
--help, 527
--issues-events, 527
--job-events, 527
--merge-requests-events, 527
--note-events, 527
--pipeline-events, 527
--push-events, 527
--releases-events, 527
--subgroup-events, 527
--sudo, 527
--tag-push-events, 527
--token, 527
--url, 527
--wiki-page-events, 527
-h, 527
gitlab-group-hook-delete command line option
--group-id, 527
--help, 527
--id, 527
--sudo, 527
-h, 527
gitlab-group-hook-get command line option
--group-id, 528
--help, 528
--id, 528
--sudo, 528
-h, 528
gitlab-group-hook-list command line option
--get-all, 528
--group-id, 528
--help, 528
--no-get-all, 528
--page, 528
--per-page, 528

--sudo, 528
-h, 528
gitlab-group-hook-update command line option
--confidential-issues-events, 529
--confidential-note-events, 529
--deployment-events, 529
--enable-ssl-verification, 529
--group-id, 529
--help, 528
--id, 529
--issues-events, 529
--job-events, 529
--merge-requests-events, 529
--note-events, 529
--pipeline-events, 529
--push-events, 529
--releases-events, 529
--subgroup-events, 529
--sudo, 529
--tag-push-events, 529
--token, 529
--url, 529
--wiki-page-events, 529
-h, 528
gitlab-group-import command line option
--help, 529
-h, 529
gitlab-group-import-get command line option
--group-id, 529
--help, 529
--sudo, 529
-h, 529
gitlab-group-invitation command line option
--help, 530
-h, 530
gitlab-group-invitation-create command line option
--access-level, 530
--email, 530
--expires-at, 530
--group-id, 530
--help, 530
--invite-source, 530
--sudo, 530
--tasks-project-id, 530
--tasks-to-be-done, 530
--user-id, 530
-h, 530
gitlab-group-invitation-delete command line option
--email, 530
--group-id, 530
--help, 530
--sudo, 530
-h, 530
gitlab-group-invitation-get command line option
--email, 531
--group-id, 531
--help, 531
--sudo, 531
-h, 531
gitlab-group-invitation-list command line option
--get-all, 531
--group-id, 531
--help, 531
--no-get-all, 531
--page, 531
--per-page, 531
--query, 531
--sudo, 531
-h, 531
gitlab-group-invitation-update command line option
--access-level, 531
--email, 531
--expires-at, 531
--group-id, 531
--help, 531
--sudo, 531
-h, 531
gitlab-group-issue command line option
--help, 532
-h, 532
gitlab-group-issue-list command line option
--assignee-id, 532
--author-id, 532
--created-after, 532
--created-before, 532
--get-all, 533
--group-id, 532
--help, 532
--iids, 532
--iteration-id, 532
--labels, 532
--milestone, 532
--my-reaction-emoji, 532
--no-get-all, 533
--order-by, 532
--page, 533
--per-page, 533
--search, 532
--sort, 532
--state, 532
--sudo, 532
--updated-after, 532

--updated-before, 533
-h, 532
gitlab-group-issues-statistics command line option
--help, 533
-h, 533
gitlab-group-issues-statistics-get command line option
--group-id, 533
--help, 533
--sudo, 533
-h, 533
gitlab-group-iteration command line option
--help, 533
-h, 533
gitlab-group-iteration-list command line option
--get-all, 534
--group-id, 534
--help, 533
--in, 534
--include-ancestors, 534
--include-descendants, 534
--no-get-all, 534
--page, 534
--per-page, 534
--search, 534
--state, 534
--sudo, 533
--updated-after, 534
--updated-before, 534
-h, 533
gitlab-group-label command line option
--help, 534
-h, 534
gitlab-group-label-create command line option
--color, 534
--description, 534
--group-id, 534
--help, 534
--name, 534
--priority, 534
--sudo, 534
-h, 534
gitlab-group-label-delete command line option
--group-id, 535
--help, 535
--name, 535
--sudo, 535
-h, 535
gitlab-group-label-get command line option
--group-id, 535
--help, 535
--name, 535
--sudo, 535
-h, 535
gitlab-group-label-list command line option
--get-all, 535
--group-id, 535
--help, 535
--no-get-all, 535
--page, 535
--per-page, 535
--sudo, 535
-h, 535
gitlab-group-label-subscribe command line option
--group-id, 536
--help, 535
--name, 536
--sudo, 536
-h, 535
gitlab-group-label-unsubscribe command line option
--group-id, 536
--help, 536
--name, 536
--sudo, 536
-h, 536
gitlab-group-label-update command line option
--color, 536
--description, 536
--group-id, 536
--help, 536
--name, 536
--new-name, 536
--priority, 536
--sudo, 536
-h, 536
gitlab-group-ldap-group-link command line option
--help, 536
-h, 536
gitlab-group-ldap-group-link-create command line option
--cn, 537
--filter, 537
--group-access, 537
--group-id, 537
--help, 537
--provider, 537
--sudo, 537
-h, 537
gitlab-group-ldap-group-link-delete command line option

--group-id, 537
--help, 537
--id, 537
--sudo, 537
-h, 537
gitlab-group-ldap-group-link-list command line option
--get-all, 537
--group-id, 537
--help, 537
--no-get-all, 537
--page, 537
--per-page, 537
--sudo, 537
-h, 537
gitlab-group-ldap-sync command line option
--help, 492
--id, 492
-h, 492
gitlab-group-list command line option
--all-available, 493
--get-all, 493
--help, 492
--min-access-level, 493
--no-get-all, 493
--order-by, 493
--owned, 493
--page, 493
--per-page, 493
--search, 493
--skip-groups, 493
--sort, 493
--statistics, 493
--sudo, 493
--top-level-only, 493
--with-custom-attributes, 493
-h, 492
gitlab-group-member command line option
--help, 538
-h, 538
gitlab-group-member-all command line option
--help, 539
-h, 539
gitlab-group-member-all-get command line option
--group-id, 540
--help, 540
--id, 540
--sudo, 540
-h, 540
gitlab-group-member-all-list command line option
--get-all, 540
--group-id, 540
--help, 540
--no-get-all, 540
--page, 540
--per-page, 540
--sudo, 540
-h, 540
gitlab-group-member-create command line option
--access-level, 538
--expires-at, 538
--group-id, 538
--help, 538
--sudo, 538
--tasks-to-be-done, 538
--user-id, 538
--username, 538
-h, 538
gitlab-group-member-delete command line option
--group-id, 538
--help, 538
--id, 538
--sudo, 538
-h, 538
gitlab-group-member-get command line option
--group-id, 539
--help, 538
--id, 539
--sudo, 538
-h, 538
gitlab-group-member-list command line option
--get-all, 539
--group-id, 539
--help, 539
--no-get-all, 539
--page, 539
--per-page, 539
--sudo, 539
-h, 539
gitlab-group-member-role command line option
--help, 540
-h, 540
gitlab-group-member-role-create command line option
--admin-cicd-variables, 541
--admin-compliance-framework, 541
--admin-group-member, 541
--admin-merge-request, 541
--admin-push-rules, 541
--admin-terraform-state, 541
--admin-vulnerability, 541
--admin-web-hook, 541

--archive-project, 541
--base-access-level, 541
--description, 541
--group-id, 541
--help, 541
--manage-deploy-tokens, 541
--manage-group-access-tokens, 541
--manage-merge-request-settings, 541
--manage-project-access-tokens, 542
--manage-security-policy-link, 542
--name, 541
--read-code, 542
--read-dependency, 542
--read-runners, 542
--read-vulnerability, 542
--remove-group, 542
--remove-project, 542
--sudo, 541
-h, 541
gitlab-group-member-role-delete command line option
--group-id, 542
--help, 542
--id, 542
--sudo, 542
-h, 542
gitlab-group-member-role-list command line option
--get-all, 542
--group-id, 542
--help, 542
--no-get-all, 542
--page, 542
--per-page, 542
--sudo, 542
-h, 542
gitlab-group-member-update command line option
--access-level, 539
--expires-at, 539
--group-id, 539
--help, 539
--id, 539
--sudo, 539
-h, 539
gitlab-group-merge-request command line option
--help, 543
-h, 543
gitlab-group-merge-request-list command line option
--approved-by-ids, 544
--approver-ids, 544
--assignee-id, 544
--author-id, 543
--created-after, 543
--created-before, 543
--get-all, 544
--group-id, 543
--help, 543
--labels, 543
--milestone, 543
--my-reaction-emoji, 544
--no-get-all, 544
--order-by, 543
--page, 544
--per-page, 544
--scope, 543
--search, 544
--sort, 543
--source-branch, 544
--state, 543
--sudo, 543
--target-branch, 544
--updated-after, 543
--updated-before, 543
--view, 543
--wip, 544
-h, 543
gitlab-group-milestone command line option
--help, 544
-h, 544
gitlab-group-milestone-create command line option
--description, 544
--due-date, 544
--group-id, 544
--help, 544
--start-date, 544
--sudo, 544
--title, 544
-h, 544
gitlab-group-milestone-delete command line option
--group-id, 545
--help, 545
--id, 545
--sudo, 545
-h, 545
gitlab-group-milestone-get command line option
--group-id, 545
--help, 545
--id, 545
--sudo, 545
-h, 545
gitlab-group-milestone-issues command line option

--group-id, 545
--help, 545
--id, 545
--sudo, 545
-h, 545
gitlab-group-milestone-list command line option
--get-all, 546
--group-id, 545
--help, 545
--iids, 546
--no-get-all, 546
--page, 546
--per-page, 546
--search, 546
--state, 546
--sudo, 545
-h, 545
gitlab-group-milestone-merge-requests command line option
--group-id, 546
--help, 546
--id, 546
--sudo, 546
-h, 546
gitlab-group-milestone-update command line option
--description, 546
--due-date, 546
--group-id, 546
--help, 546
--id, 546
--start-date, 546
--state-event, 546
--sudo, 546
--title, 546
-h, 546
gitlab-group-notification-settings command line option
--help, 547
-h, 547
gitlab-group-notification-settings-get command line option
--group-id, 547
--help, 547
--sudo, 547
-h, 547
gitlab-group-notification-settings-update command line option
--close-issue, 548
--close-merge-request, 548
--group-id, 547
--help, 547
--level, 547
--merge-merge-request, 548
--new-issue, 547
--new-merge-request, 548
--new-note, 547
--notification-email, 547
--reassign-issue, 548
--reassign-merge-request, 548
--reopen-issue, 547
--reopen-merge-request, 548
--sudo, 547
-h, 547
gitlab-group-package command line option
--help, 548
-h, 548
gitlab-group-package-list command line option
--exclude-subgroups, 548
--get-all, 549
--group-id, 548
--help, 548
--include-versionless, 548
--no-get-all, 549
--order-by, 548
--package-name, 548
--package-type, 548
--package-version, 548
--page, 549
--per-page, 549
--sort, 548
--status, 548
--sudo, 548
-h, 548
gitlab-group-project command line option
--help, 549
-h, 549
gitlab-group-project-list command line option
--archived, 549
--get-all, 550
--group-id, 549
--help, 549
--include-subgroups, 550
--min-access-level, 550
--no-get-all, 550
--order-by, 549
--owned, 549
--page, 550
--per-page, 550
--search, 549
--simple, 549
--sort, 549
--starred, 550
--sudo, 549
--visibility, 549

--with-custom-attributes, 550
--with-issues-enabled, 550
--with-merge-requests-enabled, 550
--with-security-reports, 550
--with-shared, 550
-h, 549
gitlab-group-push-rules command line option
--help, 550
-h, 550
gitlab-group-push-rules-create command line option
--author-email-regex, 551
--branch-name-regex, 551
--commit-committer-check, 551
--commit-message-negative-regex, 551
--commit-message-regex, 551
--deny-delete-tag, 551
--file-name-regex, 551
--group-id, 550
--help, 550
--max-file-size, 551
--member-check, 551
--prevent-secrets, 551
--reject-unsigned-commits, 551
--sudo, 550
-h, 550
gitlab-group-push-rules-delete command line option
--group-id, 551
--help, 551
--sudo, 551
-h, 551
gitlab-group-push-rules-get command line option
--group-id, 551
--help, 551
--sudo, 551
-h, 551
gitlab-group-push-rules-update command line option
--author-email-regex, 552
--branch-name-regex, 552
--commit-committer-check, 552
--commit-message-negative-regex, 552
--commit-message-regex, 552
--deny-delete-tag, 552
--file-name-regex, 552
--group-id, 552
--help, 552
--max-file-size, 552
--member-check, 552
--prevent-secrets, 552
--reject-unsigned-commits, 552
--sudo, 552
-h, 552
gitlab-group-restore command line option
--help, 493
--id, 493
-h, 493
gitlab-group-runner command line option
--help, 552
-h, 552
gitlab-group-runner-list command line option
--get-all, 553
--group-id, 552
--help, 552
--no-get-all, 553
--page, 553
--per-page, 553
--scope, 552
--sudo, 552
--tag-list, 553
-h, 552
gitlab-group-saml-group-link command line option
--help, 553
-h, 553
gitlab-group-saml-group-link-create command line option
--access-level, 553
--group-id, 553
--help, 553
--saml-group-name, 553
--sudo, 553
-h, 553
gitlab-group-saml-group-link-delete command line option
--group-id, 553
--help, 553
--name, 553
--sudo, 553
-h, 553
gitlab-group-saml-group-link-get command line option
--group-id, 554
--help, 554
--name, 554
--sudo, 554
-h, 554
gitlab-group-saml-group-link-list command line option
--get-all, 554
--group-id, 554
--help, 554
--no-get-all, 554
--page, 554
--per-page, 554

--sudo, 554
-h, 554
gitlab-group-search command line option
--help, 493
--id, 493
--scope, 493
--search, 493
-h, 493
gitlab-group-service-account command line option
--help, 554
-h, 554
gitlab-group-service-account-access-token command line option
--help, 556
-h, 556
gitlab-group-service-account-access-token-create command line option
--description, 556
--expires-at, 556
--group-id, 556
--help, 556
--name, 556
--scopes, 556
--sudo, 556
--user-id, 556
-h, 556
gitlab-group-service-account-access-token-delete command line option
--group-id, 557
--help, 557
--id, 557
--sudo, 557
--user-id, 557
-h, 557
gitlab-group-service-account-access-token-list command line option
--created-after, 557
--created-before, 557
--expires-after, 557
--expires-before, 557
--get-all, 558
--group-id, 557
--help, 557
--last-used-after, 557
--last-used-before, 558
--no-get-all, 558
--page, 558
--per-page, 558
--revoked, 558
--search, 558
--sort, 558
--state, 558
--sudo, 557
--user-id, 557
-h, 557
gitlab-group-service-account-access-token-rotate command line option
--expires-at, 558
--group-id, 558
--help, 558
--id, 558
--sudo, 558
--user-id, 558
-h, 558
gitlab-group-service-account-create command line option
--email, 555
--group-id, 555
--help, 554
--name, 555
--sudo, 554
--username, 555
-h, 554
gitlab-group-service-account-delete command line option
--group-id, 555
--help, 555
--id, 555
--sudo, 555
-h, 555
gitlab-group-service-account-list command line option
--get-all, 555
--group-id, 555
--help, 555
--no-get-all, 555
--order-by, 555
--page, 555
--per-page, 555
--sort, 555
--sudo, 555
-h, 555
gitlab-group-service-account-update command line option
--email, 556
--group-id, 556
--help, 556
--id, 556
--name, 556
--sudo, 556
--username, 556
-h, 556
gitlab-group-share command line option
--expires-at, 494
--group-access, 494
--group-id, 494
--help, 494

--id, 494
-h, 494
gitlab-group-subgroup command line option
--help, 558
-h, 558
gitlab-group-subgroup-list command line option
--all-available, 559
--get-all, 559
--group-id, 559
--help, 559
--min-access-level, 559
--no-get-all, 559
--order-by, 559
--owned, 559
--page, 559
--per-page, 559
--search, 559
--skip-groups, 559
--sort, 559
--statistics, 559
--sudo, 559
--with-custom-attributes, 559
-h, 559
gitlab-group-transfer command line option
--group-id, 494
--help, 494
--id, 494
-h, 494
gitlab-group-transfer-project command line option
--help, 494
--id, 494
--project-id, 494
-h, 494
gitlab-group-unshare command line option
--group-id, 494
--help, 494
--id, 494
-h, 494
gitlab-group-update command line option
--auto-devops-enabled, 495
--avatar, 495
--default-branch-protection, 496
--description, 495
--emails-disabled, 495
--extra-shared-runners-minutes-limit, 496
--file-template-project-id, 496
--help, 495
--id, 495
--lfs-enabled, 496
--membership-lock, 495
--mentions-disabled, 496
--name, 495
--path, 495
--prevent-forking-outside-group, 496
--project-creation-level, 495
--request-access-enabled, 496
--require-two-factor-authentication, 495
--share-with-group-lock, 495
--shared-runners-minutes-limit, 496
--shared-runners-setting, 496
--subgroup-creation-level, 495
--sudo, 495
--two-factor-grace-period, 495
--visibility, 495
-h, 495
gitlab-group-variable command line option
--help, 559
-h, 559
gitlab-group-variable-create command line option
--description, 560
--environment-scope, 560
--group-id, 560
--help, 560
--key, 560
--masked, 560
--masked-and-hidden, 560
--protected, 560
--raw, 560
--sudo, 560
--value, 560
--variable-type, 560
-h, 560
gitlab-group-variable-delete command line option
--group-id, 560
--help, 560
--key, 560
--sudo, 560
-h, 560
gitlab-group-variable-get command line option
--group-id, 560
--help, 560
--key, 560
--sudo, 560
-h, 560
gitlab-group-variable-list command line option
--get-all, 561
--group-id, 561
--help, 561
--no-get-all, 561
--page, 561
--per-page, 561
--sudo, 561

-h, 561
gitlab-group-variable-update command line option
--description, 561
--environment-scope, 561
--group-id, 561
--help, 561
--key, 561
--masked, 561
--protected, 561
--raw, 561
--sudo, 561
--value, 561
--variable-type, 561
-h, 561
gitlab-group-wiki command line option
--help, 562
-h, 562
gitlab-group-wiki-create command line option
--content, 562
--format, 562
--group-id, 562
--help, 562
--sudo, 562
--title, 562
-h, 562
gitlab-group-wiki-delete command line option
--group-id, 562
--help, 562
--slug, 562
--sudo, 562
-h, 562
gitlab-group-wiki-get command line option
--group-id, 562
--help, 562
--slug, 562
--sudo, 562
-h, 562
gitlab-group-wiki-list command line option
--get-all, 563
--group-id, 563
--help, 563
--no-get-all, 563
--page, 563
--per-page, 563
--sudo, 563
--with-content, 563
-h, 563
gitlab-group-wiki-update command line option
--content, 563
--format, 563
--group-id, 563
--help, 563
--slug, 563
--sudo, 563
--title, 563
-h, 563
gitlab-hook command line option
--help, 563
-h, 563
gitlab-hook-create command line option
--help, 564
--sudo, 564
--url, 564
-h, 564
gitlab-hook-delete command line option
--help, 564
--id, 564
--sudo, 564
-h, 564
gitlab-hook-get command line option
--help, 564
--id, 564
--sudo, 564
-h, 564
gitlab-hook-list command line option
--get-all, 564
--help, 564
--no-get-all, 564
--page, 564
--per-page, 564
--sudo, 564
-h, 564
gitlab-issue command line option
--help, 565
-h, 565
gitlab-issue-get command line option
--help, 565
--id, 565
--sudo, 565
-h, 565
gitlab-issue-list command line option
--assignee-id, 565
--author-id, 565
--created-after, 566
--created-before, 566
--get-all, 566
--help, 565
--iids, 565
--iteration-id, 565
--labels, 565
--milestone, 565
--my-reaction-emoji, 565
--no-get-all, 566
--order-by, 565

--page, 566
--per-page, 566
--scope, 565
--search, 566
--sort, 565
--state, 565
--sudo, 565
--updated-after, 566
--updated-before, 566
-h, 565
gitlab-issues-statistics command line option
--help, 566
-h, 566
gitlab-issues-statistics-get command line option
--help, 566
--sudo, 566
-h, 566
gitlab-key command line option
--help, 566
-h, 566
gitlab-key-get command line option
--help, 566
--id, 566
--sudo, 566
-h, 566
gitlab-ldap-group command line option
--help, 567
-h, 567
gitlab-ldap-group-list command line option
--get-all, 567
--help, 567
--no-get-all, 567
--page, 567
--per-page, 567
--provider, 567
--search, 567
--sudo, 567
-h, 567
gitlab-license command line option
--help, 567
-h, 567
gitlab-license-get command line option
--fullname, 567
--help, 567
--key, 567
--project, 567
--sudo, 567
-h, 567
gitlab-license-list command line option
--get-all, 568
--help, 568
--no-get-all, 568
--page, 568
--per-page, 568
--popular, 568
--sudo, 568
-h, 568
gitlab-member-role command line option
--help, 568
-h, 568
gitlab-member-role-create command line option
--admin-cicd-variables, 569
--admin-compliance-framework, 569
--admin-group-member, 569
--admin-merge-request, 569
--admin-push-rules, 569
--admin-terraform-state, 569
--admin-vulnerability, 569
--admin-web-hook, 569
--archive-project, 569
--base-access-level, 569
--description, 569
--help, 569
--manage-deploy-tokens, 569
--manage-group-access-tokens, 569
--manage-merge-request-settings, 569
--manage-project-access-tokens, 569
--manage-security-policy-link, 569
--name, 569
--read-code, 569
--read-dependency, 569
--read-runners, 569
--read-vulnerability, 569
--remove-group, 569
--remove-project, 569
--sudo, 569
-h, 569
gitlab-member-role-delete command line option
--help, 570
--id, 570
--sudo, 570
-h, 570
gitlab-member-role-list command line option
--get-all, 570
--help, 570
--no-get-all, 570
--page, 570
--per-page, 570
--sudo, 570
-h, 570
gitlab-merge-request command line option
--help, 570
-h, 570

gitlab-merge-request-list command line option
--approved-by-ids, 571
--approver-ids, 571
--assignee-id, 571
--author-id, 571
--author-username, 571
--created-after, 571
--created-before, 571
--deployed-after, 572
--deployed-before, 572
--environment, 572
--get-all, 572
--help, 571
--in, 572
--labels, 571
--milestone, 571
--my-reaction-emoji, 572
--no-get-all, 572
--not, 572
--order-by, 571
--page, 572
--per-page, 572
--reviewer-id, 571
--reviewer-username, 571
--scope, 571
--search, 572
--sort, 571
--source-branch, 572
--state, 571
--sudo, 571
--target-branch, 572
--updated-after, 571
--updated-before, 571
--view, 571
--wip, 572
--with-labels-details, 571
--with-merge-status-recheck, 571
-h, 571
gitlab-namespace command line option
--help, 572
-h, 572
gitlab-namespace-exists command line option
--help, 572
--id, 572
--namespace, 572
--parent-id, 572
-h, 572
gitlab-namespace-get command line option
--help, 573
--id, 573
--sudo, 573
-h, 573
gitlab-namespace-list command line option
--get-all, 573
--help, 573
--no-get-all, 573
--page, 573
--per-page, 573
--search, 573
--sudo, 573
-h, 573
gitlab-notification-settings command line option
--help, 573
-h, 573
gitlab-notification-settings-get command line option
--help, 573
--sudo, 573
-h, 573
gitlab-notification-settings-update command line option
--close-issue, 574
--close-merge-request, 574
--help, 574
--level, 574
--merge-merge-request, 574
--new-issue, 574
--new-merge-request, 574
--new-note, 574
--notification-email, 574
--reassign-issue, 574
--reassign-merge-request, 574
--reopen-issue, 574
--reopen-merge-request, 574
--sudo, 574
-h, 574
gitlab-pages-domain command line option
--help, 574
-h, 574
gitlab-pages-domain-list command line option
--get-all, 575
--help, 574
--no-get-all, 575
--page, 575
--per-page, 575
--sudo, 575
-h, 574
gitlab-personal-access-token command line option
--help, 575
-h, 575
gitlab-personal-access-token-delete command line option
--help, 575
--id, 575

--sudo, 575
-h, 575
gitlab-personal-access-token-get command line option
--help, 575
--id, 575
--sudo, 575
-h, 575
gitlab-personal-access-token-list command line option
--get-all, 576
--help, 575
--no-get-all, 576
--page, 576
--per-page, 576
--sudo, 575
--user-id, 575
-h, 575
gitlab-personal-access-token-rotate command line option
--expires-at, 576
--help, 576
--id, 576
-h, 576
gitlab-project command line option
--help, 576
-h, 576
gitlab-project-access-request command line option
--help, 593
-h, 593
gitlab-project-access-request-approve command line option
--access-level, 593
--help, 593
--id, 593
--project-id, 593
--sudo, 593
-h, 593
gitlab-project-access-request-create command line option
--help, 593
--project-id, 593
--sudo, 593
-h, 593
gitlab-project-access-request-delete command line option
--help, 593
--id, 594
--project-id, 593
--sudo, 593
-h, 593
gitlab-project-access-request-list command line option
--get-all, 594
--help, 594
--no-get-all, 594
--page, 594
--per-page, 594
--project-id, 594
--sudo, 594
-h, 594
gitlab-project-access-token command line option
--help, 594
-h, 594
gitlab-project-access-token-create command line option
--access-level, 594
--expires-at, 594
--help, 594
--name, 594
--project-id, 594
--scopes, 594
--sudo, 594
-h, 594
gitlab-project-access-token-delete command line option
--help, 595
--id, 595
--project-id, 595
--sudo, 595
-h, 595
gitlab-project-access-token-get command line option
--help, 595
--id, 595
--project-id, 595
--sudo, 595
-h, 595
gitlab-project-access-token-list command line option
--get-all, 595
--help, 595
--no-get-all, 595
--page, 595
--per-page, 595
--project-id, 595
--sudo, 595
-h, 595
gitlab-project-access-token-rotate command line option
--expires-at, 596
--help, 596
--id, 596
--project-id, 596
--sudo, 596
-h, 596

gitlab-project-additional-statistics command line option
--help, 596
-h, 596
gitlab-project-additional-statistics-get command line option
--help, 596
--project-id, 596
--sudo, 596
-h, 596
gitlab-project-approval command line option
--help, 596
-h, 596
gitlab-project-approval-get command line option
--help, 596
--project-id, 596
--sudo, 596
-h, 596
gitlab-project-approval-rule command line option
--help, 597
-h, 597
gitlab-project-approval-rule-create command line option
--applies-to-all-protected-branches, 598
--approvals-required, 598
--group-ids, 598
--help, 597
--name, 598
--project-id, 597
--protected-branch-ids, 598
--sudo, 597
--user-ids, 598
--usernames, 598
-h, 597
gitlab-project-approval-rule-delete command line option
--help, 598
--id, 598
--project-id, 598
--sudo, 598
-h, 598
gitlab-project-approval-rule-get command line option
--help, 598
--id, 598
--project-id, 598
--sudo, 598
-h, 598
gitlab-project-approval-rule-list command line option
--get-all, 599
--help, 598
--no-get-all, 599
--page, 599
--per-page, 599
--project-id, 598
--sudo, 598
-h, 598
gitlab-project-approval-rule-update command line option
--help, 599
--id, 599
--project-id, 599
--sudo, 599
-h, 599
gitlab-project-approval-update command line option
--approvals-before-merge, 597
--disable-overriding-approvers-per-merge-request, 597
--help, 597
--merge-requests-author-approval, 597
--merge-requests-disable-committers-approval, 597
--project-id, 597
--reset-approvals-on-push, 597
--sudo, 597
-h, 597
gitlab-project-archive command line option
--help, 576
--id, 576
-h, 576
gitlab-project-artifact command line option
--help, 599
-h, 599
gitlab-project-artifact-delete command line option
--help, 599
--project-id, 599
--ref-name, 599
--sudo, 599
-h, 599
gitlab-project-artifact-download command line option
--help, 599
--job, 600
--job-token, 600
--project-id, 600
--ref-name, 600
--sudo, 600
-h, 599
gitlab-project-artifact-raw command line option
--artifact-path, 600
--help, 600
--job, 600

--project-id, 600
--ref-name, 600
--sudo, 600
-h, 600
gitlab-project-audit-event command line option
--help, 600
-h, 600
gitlab-project-audit-event-get command line option
--help, 600
--id, 600
--project-id, 600
--sudo, 600
-h, 600
gitlab-project-audit-event-list command line option
--created-after, 601
--created-before, 601
--get-all, 601
--help, 601
--no-get-all, 601
--page, 601
--per-page, 601
--project-id, 601
--sudo, 601
-h, 601
gitlab-project-badge command line option
--help, 601
-h, 601
gitlab-project-badge-create command line option
--help, 601
--image-url, 601
--link-url, 601
--project-id, 601
--sudo, 601
-h, 601
gitlab-project-badge-delete command line option
--help, 602
--id, 602
--project-id, 602
--sudo, 602
-h, 602
gitlab-project-badge-get command line option
--help, 602
--id, 602
--project-id, 602
--sudo, 602
-h, 602
gitlab-project-badge-list command line option
--get-all, 602
--help, 602
--no-get-all, 602
--page, 602
--per-page, 602
--project-id, 602
--sudo, 602
-h, 602
gitlab-project-badge-render command line option
--help, 602
--id, 603
--image-url, 603
--link-url, 603
--project-id, 603
--sudo, 603
-h, 602
gitlab-project-badge-update command line option
--help, 603
--id, 603
--image-url, 603
--link-url, 603
--project-id, 603
--sudo, 603
-h, 603
gitlab-project-board command line option
--help, 603
-h, 603
gitlab-project-board-create command line option
--help, 603
--name, 603
--project-id, 603
--sudo, 603
-h, 603
gitlab-project-board-delete command line option
--help, 604
--id, 604
--project-id, 604
--sudo, 604
-h, 604
gitlab-project-board-get command line option
--help, 604
--id, 604
--project-id, 604
--sudo, 604
-h, 604
gitlab-project-board-list command line option
--get-all, 604
--help, 604, 605

--no-get-all, 604
--page, 604
--per-page, 604
--project-id, 604
--sudo, 604
-h, 604, 605
gitlab-project-board-list-create command line option
--assignee-id, 605
--board-id, 605
--help, 605
--iteration-id, 605
--label-id, 605
--milestone-id, 605
--project-id, 605
--sudo, 605
-h, 605
gitlab-project-board-list-delete command line option
--board-id, 605
--help, 605
--id, 606
--project-id, 605
--sudo, 605
-h, 605
gitlab-project-board-list-get command line option
--board-id, 606
--help, 606
--id, 606
--project-id, 606
--sudo, 606
-h, 606
gitlab-project-board-list-list command line option
--board-id, 606
--get-all, 606
--help, 606
--no-get-all, 606
--page, 606
--per-page, 606
--project-id, 606
--sudo, 606
-h, 606
gitlab-project-board-list-update command line option
--board-id, 607
--help, 606
--id, 607
--position, 607
--project-id, 607
--sudo, 607
-h, 606
gitlab-project-board-update command line option
--help, 604
--id, 605
--project-id, 605
--sudo, 605
-h, 604
gitlab-project-branch command line option
--help, 607
-h, 607
gitlab-project-branch-create command line option
--branch, 607
--help, 607
--project-id, 607
--ref, 607
--sudo, 607
-h, 607
gitlab-project-branch-delete command line option
--help, 607
--name, 607
--project-id, 607
--sudo, 607
-h, 607
gitlab-project-branch-get command line option
--help, 607
--name, 608
--project-id, 608
--sudo, 608
-h, 607
gitlab-project-branch-list command line option
--get-all, 608
--help, 608
--no-get-all, 608
--page, 608
--per-page, 608
--project-id, 608
--sudo, 608
-h, 608
gitlab-project-ci-lint command line option
--help, 608
-h, 608
gitlab-project-ci-lint-create command line option
--content, 608
--dry-run, 608
--help, 608
--include-jobs, 608
--project-id, 608
--ref, 609
--sudo, 608

-h, 608
gitlab-project-ci-lint-get command line option
--content-ref, 609
--dry-run, 609
--dry-run-ref, 609
--help, 609
--include-jobs, 609
--project-id, 609
--ref, 609
--sudo, 609
-h, 609
gitlab-project-ci-lint-validate command line option
--content, 609
--dry-run, 609
--help, 609
--include-jobs, 609
--project-id, 609
--ref, 609
--sudo, 609
-h, 609
gitlab-project-cluster command line option
--help, 609
-h, 609
gitlab-project-cluster-agent command line option
--help, 611
-h, 611
gitlab-project-cluster-agent-create command line option
--help, 612
--name, 612
--project-id, 612
--sudo, 612
-h, 612
gitlab-project-cluster-agent-delete command line option
--help, 612
--id, 612
--project-id, 612
--sudo, 612
-h, 612
gitlab-project-cluster-agent-get command line option
--help, 612
--id, 612
--project-id, 612
--sudo, 612
-h, 612
gitlab-project-cluster-agent-list command line option
--get-all, 613
--help, 612
--no-get-all, 613
--page, 612
--per-page, 613
--project-id, 612
--sudo, 612
-h, 612
gitlab-project-cluster-create command line option
--domain, 610
--enabled, 610
--environment-scope, 610
--help, 610
--managed, 610
--name, 610
--platform-kubernetes-attributes, 610
--project-id, 610
--sudo, 610
-h, 610
gitlab-project-cluster-delete command line option
--help, 610
--id, 610
--project-id, 610
--sudo, 610
-h, 610
gitlab-project-cluster-get command line option
--help, 610
--id, 610
--project-id, 610
--sudo, 610
-h, 610
gitlab-project-cluster-list command line option
--get-all, 611
--help, 611
--no-get-all, 611
--page, 611
--per-page, 611
--project-id, 611
--sudo, 611
-h, 611
gitlab-project-cluster-update command line option
--domain, 611
--environment-scope, 611
--help, 611
--id, 611
--management-project-id, 611
--name, 611
--platform-kubernetes-attributes, 611
--project-id, 611
--sudo, 611
-h, 611

gitlab-project-commit command line option
--help, 613
-h, 613
gitlab-project-commit-cherry-pick command line option
--branch, 613
--help, 613
--id, 613
--project-id, 613
--sudo, 613
-h, 613
gitlab-project-commit-comment command line option
--help, 616
-h, 616
gitlab-project-commit-comment-create command line option
--commit-id, 617
--help, 617
--line, 617
--line-type, 617
--note, 617
--path, 617
--project-id, 617
--sudo, 617
-h, 617
gitlab-project-commit-comment-list command line option
--commit-id, 617
--get-all, 617
--help, 617
--no-get-all, 617
--page, 617
--per-page, 617
--project-id, 617
--sudo, 617
-h, 617
gitlab-project-commit-create command line option
--actions, 613
--author-email, 613
--author-name, 614
--branch, 613
--commit-message, 613
--help, 613
--project-id, 613
--sudo, 613
-h, 613
gitlab-project-commit-diff command line option
--help, 614
--id, 614
--project-id, 614
--sudo, 614
-h, 614
gitlab-project-commit-discussion command line option
--help, 617
-h, 617
gitlab-project-commit-discussion-create command line option
--body, 618
--commit-id, 618
--created-at, 618
--help, 618
--project-id, 618
--sudo, 618
-h, 618
gitlab-project-commit-discussion-get command line option
--commit-id, 618
--help, 618
--id, 618
--project-id, 618
--sudo, 618
-h, 618
gitlab-project-commit-discussion-list command line option
--commit-id, 618
--get-all, 619
--help, 618
--no-get-all, 619
--page, 618
--per-page, 619
--project-id, 618
--sudo, 618
-h, 618
gitlab-project-commit-discussion-note command line option
--help, 619
-h, 619
gitlab-project-commit-discussion-note-create command line option
--body, 619
--commit-id, 619
--created-at, 619
--discussion-id, 619
--help, 619
--position, 619
--project-id, 619
--sudo, 619
-h, 619
gitlab-project-commit-discussion-note-delete command line option
--commit-id, 620
--discussion-id, 620
--help, 619
--id, 620

--project-id, 620
--sudo, 619
-h, 619
gitlab-project-commit-discussion-note-get command line option
--commit-id, 620
--discussion-id, 620
--help, 620
--id, 620
--project-id, 620
--sudo, 620
-h, 620
gitlab-project-commit-discussion-note-update command line option
--body, 620
--commit-id, 620
--discussion-id, 620
--help, 620
--id, 620
--project-id, 620
--sudo, 620
-h, 620
gitlab-project-commit-get command line option
--help, 614
--id, 614
--project-id, 614
--sudo, 614
-h, 614
gitlab-project-commit-list command line option
--all, 614
--first-parent, 615
--get-all, 615
--help, 614
--no-get-all, 615
--order, 615
--page, 615
--path, 615
--per-page, 615
--project-id, 614
--ref-name, 614
--since, 614
--sudo, 614
--trailers, 615
--until, 614
--with-stats, 615
-h, 614
gitlab-project-commit-merge-requests command line option
--help, 615
--id, 615
--project-id, 615
--sudo, 615
-h, 615
gitlab-project-commit-refs command line option
--help, 615
--id, 615
--project-id, 615
--sudo, 615
--type, 615
-h, 615
gitlab-project-commit-revert command line option
--branch, 616
--help, 616
--id, 616
--project-id, 616
--sudo, 616
-h, 616
gitlab-project-commit-sequence command line option
--help, 616
--id, 616
--project-id, 616
--sudo, 616
-h, 616
gitlab-project-commit-signature command line option
--help, 616
--id, 616
--project-id, 616
--sudo, 616
-h, 616
gitlab-project-commit-status command line option
--help, 621
-h, 621
gitlab-project-commit-status-create command line option
--commit-id, 621
--context, 621
--coverage, 621
--description, 621
--help, 621
--name, 621
--project-id, 621
--ref, 621
--state, 621
--sudo, 621
--target-url, 621
-h, 621
gitlab-project-commit-status-list command line option
--commit-id, 621
--get-all, 622
--help, 621

--no-get-all, 622
--page, 622
--per-page, 622
--project-id, 621
--sudo, 621
-h, 621
gitlab-project-create command line option
--allow-merge-on-skipped-pipeline, 578
--analytics-access-level, 578
--approvals-before-merge, 578
--auto-cancel-pending-pipelines, 578
--auto-devops-deploy-strategy, 578
--auto-devops-enabled, 578
--autoclose-referenced-issues, 578
--avatar, 578
--build-coverage-regex, 578
--build-git-strategy, 578
--build-timeout, 578
--builds-access-level, 578
--ci-config-path, 579
--container-expiration-policy-attributes, 579
--container-registry-access-level, 579
--container-registry-enabled, 579
--default-branch, 579
--description, 579
--emails-disabled, 579
--environments-access-level, 580
--external-authorization-classification-label, 579
--feature-flags-access-level, 580
--forking-access-level, 579
--group-with-project-templates-id, 579
--help, 578
--import-url, 579
--infrastructure-access-level, 580
--initialize-with-readme, 579
--issues-access-level, 579
--issues-enabled, 579
--jobs-enabled, 579
--lfs-enabled, 579
--merge-method, 579
--merge-pipelines-enabled, 579
--merge-requests-access-level, 579
--merge-requests-enabled, 579
--mirror, 579
--mirror-trigger-builds, 579
--monitor-access-level, 580
--name, 578
--namespace-id, 579
--only-allow-merge-if-all-discussions-are-resolved, 579
--only-allow-merge-if-all-status-checks-passed, 578
--only-allow-merge-if-pipeline-succeeds, 579
--operations-access-level, 579
--packages-enabled, 579
--pages-access-level, 579
--path, 578
--printing-merge-request-link-enabled, 580
--public-builds, 580
--releases-access-level, 580
--remove-source-branch-after-merge, 580
--repository-access-level, 580
--repository-storage, 580
--request-access-enabled, 580
--requirements-access-level, 579
--resolve-outdated-diff-discussions, 580
--security-and-compliance-access-level, 580
--shared-runners-enabled, 580
--show-default-award-emojis, 580
--snippets-access-level, 580
--snippets-enabled, 580
--squash-option, 580
--sudo, 578
--tag-list, 580
--template-name, 580
--template-project-id, 580
--topics, 580
--use-custom-template, 580
--visibility, 580
--wiki-access-level, 580
--wiki-enabled, 580
-h, 578
gitlab-project-create-fork-relation command line option
--forked-from-id, 581
--help, 581
--id, 581
-h, 581
gitlab-project-custom-attribute command line option
--help, 622
-h, 622
gitlab-project-custom-attribute-delete command line option
--help, 622
--key, 622
--project-id, 622
--sudo, 622
-h, 622
gitlab-project-custom-attribute-get command line option
--help, 622
--key, 622

--project-id, 622
--sudo, 622
-h, 622
gitlab-project-custom-attribute-list command line option
--get-all, 623
--help, 622
--no-get-all, 623
--page, 623
--per-page, 623
--project-id, 623
--sudo, 623
-h, 622
gitlab-project-delete command line option
--help, 581
--id, 581
--sudo, 581
-h, 581
gitlab-project-delete-fork-relation command line option
--help, 581
--id, 581
-h, 581
gitlab-project-delete-merged-branches command line option
--help, 581
--id, 581
-h, 581
gitlab-project-deploy-token command line option
--help, 623
-h, 623
gitlab-project-deploy-token-create command line option
--expires-at, 623
--help, 623
--name, 623
--project-id, 623
--scopes, 623
--sudo, 623
--username, 623
-h, 623
gitlab-project-deploy-token-delete command line option
--help, 623
--id, 624
--project-id, 623
--sudo, 623
-h, 623
gitlab-project-deploy-token-get command line option
--help, 624
--id, 624
--project-id, 624
--sudo, 624
-h, 624
gitlab-project-deploy-token-list command line option
--get-all, 624
--help, 624
--no-get-all, 624
--page, 624
--per-page, 624
--project-id, 624
--scopes, 624
--sudo, 624
-h, 624
gitlab-project-deployment command line option
--help, 624
-h, 624
gitlab-project-deployment-approval command line option
--comment, 625
--help, 625
--id, 625
--project-id, 625
--represented-as, 625
--status, 625
--sudo, 625
-h, 625
gitlab-project-deployment-create command line option
--environment, 625
--help, 625
--project-id, 625
--ref, 625
--sha, 625
--status, 625
--sudo, 625
--tag, 625
-h, 625
gitlab-project-deployment-get command line option
--help, 625
--id, 625
--project-id, 625
--sudo, 625
-h, 625
gitlab-project-deployment-list command line option
--environment, 626
--get-all, 626
--help, 626
--no-get-all, 626
--order-by, 626
--page, 626
--per-page, 626

--project-id, 626
--sort, 626
--status, 626
--sudo, 626
--updated-after, 626
--updated-before, 626
-h, 626
gitlab-project-deployment-merge-request command line option
--help, 627
-h, 627
gitlab-project-deployment-merge-request-list command line option
--approved-by-ids, 628
--approver-ids, 628
--assignee-id, 628
--author-id, 628
--author-username, 628
--created-after, 628
--created-before, 628
--deployed-after, 628
--deployed-before, 628
--deployment-id, 627
--environment, 628
--get-all, 629
--help, 627
--in, 628
--labels, 628
--milestone, 628
--my-reaction-emoji, 628
--no-get-all, 629
--not, 628
--order-by, 628
--page, 629
--per-page, 629
--project-id, 628
--reviewer-id, 628
--reviewer-username, 628
--scope, 628
--search, 628
--sort, 628
--source-branch, 628
--state, 628
--sudo, 627
--target-branch, 628
--updated-after, 628
--updated-before, 628
--view, 628
--wip, 628
--with-labels-details, 628
--with-merge-status-recheck, 628
-h, 627
gitlab-project-deployment-update command line option
--help, 626
--id, 626
--project-id, 626
--sudo, 626
-h, 626
gitlab-project-dockerfile-template command line option
--help, 629
-h, 629
gitlab-project-dockerfile-template-get command line option
--help, 629
--name, 629
--project-id, 629
--sudo, 629
-h, 629
gitlab-project-dockerfile-template-list command line option
--get-all, 629
--help, 629
--no-get-all, 629
--page, 629
--per-page, 629
--project-id, 629
--sudo, 629
-h, 629
gitlab-project-environment command line option
--help, 630
-h, 630
gitlab-project-environment-create command line option
--external-url, 630
--help, 630
--name, 630
--project-id, 630
--sudo, 630
-h, 630
gitlab-project-environment-delete command line option
--help, 630
--id, 630
--project-id, 630
--sudo, 630
-h, 630
gitlab-project-environment-get command line option
--help, 630
--id, 630
--project-id, 630
--sudo, 630
-h, 630
gitlab-project-environment-list command line option

--get-all, 631
--help, 631
--name, 631
--no-get-all, 631
--page, 631
--per-page, 631
--project-id, 631
--search, 631
--states, 631
--sudo, 631
-h, 631
gitlab-project-environment-stop command line option
--help, 631
--id, 631
--project-id, 631
--sudo, 631
-h, 631
gitlab-project-environment-update command line option
--external-url, 632
--help, 631
--id, 632
--name, 632
--project-id, 631
--sudo, 631
-h, 631
gitlab-project-event command line option
--help, 632
-h, 632
gitlab-project-event-list command line option
--action, 632
--after, 632
--before, 632
--get-all, 632
--help, 632
--no-get-all, 632
--page, 632
--per-page, 632
--project-id, 632
--scope, 632
--sort, 632
--sudo, 632
--target-type, 632
-h, 632
gitlab-project-export command line option
--help, 633
-h, 633
gitlab-project-export-create command line option
--description, 633
--help, 633
--project-id, 633
--sudo, 633
-h, 633
gitlab-project-export-download command line option
--help, 633
--project-id, 633
--sudo, 633
-h, 633
gitlab-project-export-get command line option
--help, 633
--project-id, 633
--sudo, 633
-h, 633
gitlab-project-external-status-check command line option
--help, 633
-h, 633
gitlab-project-external-status-check-create command line option
--external-url, 634
--help, 634
--name, 634
--project-id, 634
--protected-branch-ids, 634
--shared-secret, 634
--sudo, 634
-h, 634
gitlab-project-external-status-check-delete command line option
--help, 634
--id, 634
--project-id, 634
--sudo, 634
-h, 634
gitlab-project-external-status-check-list command line option
--get-all, 635
--help, 634
--no-get-all, 635
--page, 634
--per-page, 634
--project-id, 634
--sudo, 634
-h, 634
gitlab-project-external-status-check-update command line option
--external-url, 635
--help, 635
--id, 635
--name, 635
--project-id, 635
--protected-branch-ids, 635
--shared-secret, 635

--sudo, 635
-h, 635
gitlab-project-feature-flag command line option
--help, 635
-h, 635
gitlab-project-feature-flag-create command line option
--active, 636
--description, 636
--help, 635
--name, 636
--project-id, 635
--strategies, 636
--sudo, 635
--version, 636
-h, 635
gitlab-project-feature-flag-delete command line option
--help, 636
--name, 636
--project-id, 636
--sudo, 636
-h, 636
gitlab-project-feature-flag-get command line option
--help, 636
--name, 636
--project-id, 636
--sudo, 636
-h, 636
gitlab-project-feature-flag-list command line option
--get-all, 637
--help, 636
--no-get-all, 637
--page, 636
--per-page, 637
--project-id, 636
--scope, 636
--sudo, 636
-h, 636
gitlab-project-feature-flag-update command line option
--active, 637
--description, 637
--help, 637
--name, 637
--new-name, 637
--project-id, 637
--strategies, 637
--sudo, 637
-h, 637
gitlab-project-feature-flag-user-list command line option
--help, 637
-h, 637
gitlab-project-feature-flag-user-list-create command line option
--help, 637
--name, 638
--project-id, 637
--sudo, 637
--user-xids, 638
-h, 637
gitlab-project-feature-flag-user-list-delete command line option
--help, 638
--iid, 638
--project-id, 638
--sudo, 638
-h, 638
gitlab-project-feature-flag-user-list-get command line option
--help, 638
--iid, 638
--project-id, 638
--sudo, 638
-h, 638
gitlab-project-feature-flag-user-list-list command line option
--get-all, 639
--help, 638
--no-get-all, 639
--page, 638
--per-page, 638
--project-id, 638
--search, 638
--sudo, 638
-h, 638
gitlab-project-feature-flag-user-list-update command line option
--help, 639
--iid, 639
--name, 639
--project-id, 639
--sudo, 639
--user-xids, 639
-h, 639
gitlab-project-file command line option
--help, 639
-h, 639
gitlab-project-file-blame command line option
--file-path, 639
--help, 639
--project-id, 639

--ref, 639
--sudo, 639
-h, 639
gitlab-project-file-create command line option
--author-email, 640
--author-name, 640
--branch, 640
--commit-message, 640
--content, 640
--encoding, 640
--execute-filemode, 640
--file-path, 640
--help, 640
--project-id, 640
--start-branch, 640
--sudo, 640
-h, 640
gitlab-project-file-delete command line option
--branch, 640
--commit-message, 640
--file-path, 640
--help, 640
--project-id, 640
--sudo, 640
-h, 640
gitlab-project-file-get command line option
--file-path, 641
--help, 641
--project-id, 641
--ref, 641
--sudo, 641
-h, 641
gitlab-project-file-raw command line option
--file-path, 641
--help, 641
--project-id, 641
--ref, 641
--sudo, 641
-h, 641
gitlab-project-file-update command line option
--author-email, 642
--author-name, 642
--branch, 641
--commit-message, 642
--content, 641
--encoding, 642
--execute-filemode, 642
--file-path, 641
--help, 641
--last-commit-id, 642
--project-id, 641
--start-branch, 642
--sudo, 641
-h, 641
gitlab-project-fork command line option
--help, 642
-h, 642
gitlab-project-fork-create command line option
--help, 642
--namespace, 642
--project-id, 642
--sudo, 642
-h, 642
gitlab-project-fork-list command line option
--archived, 643
--get-all, 643
--help, 642
--membership, 643
--no-get-all, 643
--order-by, 643
--owned, 643
--page, 643
--per-page, 643
--project-id, 643
--search, 643
--simple, 643
--sort, 643
--starred, 643
--statistics, 643
--sudo, 642
--visibility, 643
--with-custom-attributes, 643
--with-issues-enabled, 643
--with-merge-requests-enabled, 643
-h, 642
gitlab-project-get command line option
--help, 581
--id, 581
--sudo, 581
-h, 581
gitlab-project-gitignore-template command line option
--help, 643
-h, 643
gitlab-project-gitignore-template-get command line option
--help, 643
--name, 644
--project-id, 643
--sudo, 643
-h, 643
gitlab-project-gitignore-template-list command line option

--get-all, 644
--help, 644
--no-get-all, 644
--page, 644
--per-page, 644
--project-id, 644
--sudo, 644
-h, 644
gitlab-project-gitlabciyml-template command line option
--help, 644
-h, 644
gitlab-project-gitlabciyml-template-get command line option
--help, 644
--name, 644
--project-id, 644
--sudo, 644
-h, 644
gitlab-project-gitlabciyml-template-list command line option
--get-all, 645
--help, 645
--no-get-all, 645
--page, 645
--per-page, 645
--project-id, 645
--sudo, 645
-h, 645
gitlab-project-hook command line option
--help, 645
-h, 645
gitlab-project-hook-create command line option
--confidential-issues-events, 646
--enable-ssl-verification, 646
--help, 645
--issues-events, 646
--job-events, 646
--merge-requests-events, 646
--note-events, 646
--pipeline-events, 646
--project-id, 645
--push-events, 645
--sudo, 645
--tag-push-events, 646
--token, 646
--url, 645
--wiki-page-events, 646
-h, 645
gitlab-project-hook-delete command line option
--help, 646
--id, 646
--project-id, 646
--sudo, 646
-h, 646
gitlab-project-hook-get command line option
--help, 646
--id, 646
--project-id, 646
--sudo, 646
-h, 646
gitlab-project-hook-list command line option
--get-all, 647
--help, 646
--no-get-all, 647
--page, 647
--per-page, 647
--project-id, 647
--sudo, 647
-h, 646
gitlab-project-hook-update command line option
--confidential-issues-events, 647
--enable-ssl-verification, 648
--help, 647
--id, 647
--issues-events, 647
--job-events, 647
--merge-requests-events, 647
--note-events, 647
--pipeline-events, 648
--project-id, 647
--push-events, 647
--sudo, 647
--tag-push-events, 647
--token, 648
--url, 647
--wiki-events, 648
-h, 647
gitlab-project-housekeeping command line option
--help, 582
--id, 582
-h, 582
gitlab-project-import command line option
--help, 648
-h, 648
gitlab-project-import-get command line option
--help, 648
--project-id, 648
--sudo, 648
-h, 648
gitlab-project-integration command line option

--help, 648
-h, 648
gitlab-project-integration-available command line option
--help, 648
--project-id, 648
--slug, 648
--sudo, 648
-h, 648
gitlab-project-integration-delete command line option
--help, 648
--project-id, 649
--slug, 649
--sudo, 649
-h, 648
gitlab-project-integration-get command line option
--help, 649
--project-id, 649
--slug, 649
--sudo, 649
-h, 649
gitlab-project-integration-list command line option
--get-all, 649
--help, 649
--no-get-all, 649
--page, 649
--per-page, 649
--project-id, 649
--sudo, 649
-h, 649
gitlab-project-integration-update command line option
--help, 649
--project-id, 649
--slug, 650
--sudo, 649
-h, 649
gitlab-project-invitation command line option
--help, 650
-h, 650
gitlab-project-invitation-create command line option
--access-level, 650
--email, 650
--expires-at, 650
--help, 650
--invite-source, 650
--project-id, 650
--sudo, 650
--tasks-project-id, 650
--tasks-to-be-done, 650
--user-id, 650
-h, 650
gitlab-project-invitation-delete command line option
--email, 650
--help, 650
--project-id, 650
--sudo, 650
-h, 650
gitlab-project-invitation-get command line option
--email, 651
--help, 651
--project-id, 651
--sudo, 651
-h, 651
gitlab-project-invitation-list command line option
--get-all, 651
--help, 651
--no-get-all, 651
--page, 651
--per-page, 651
--project-id, 651
--query, 651
--sudo, 651
-h, 651
gitlab-project-invitation-update command line option
--access-level, 651
--email, 651
--expires-at, 651
--help, 651
--project-id, 651
--sudo, 651
-h, 651
gitlab-project-issue command line option
--help, 652
-h, 652
gitlab-project-issue-add-spent-time command line option
--duration, 652
--help, 652
--iid, 652
--project-id, 652
--sudo, 652
-h, 652
gitlab-project-issue-award-emoji command line option
--help, 659
-h, 659
gitlab-project-issue-award-emoji-create command line option

--help, 659
--issue-iid, 659
--name, 659
--project-id, 659
--sudo, 659
-h, 659
gitlab-project-issue-award-emoji-delete command line option
--help, 659
--id, 659
--issue-iid, 659
--project-id, 659
--sudo, 659
-h, 659
gitlab-project-issue-award-emoji-get command line option
--help, 660
--id, 660
--issue-iid, 660
--project-id, 660
--sudo, 660
-h, 660
gitlab-project-issue-award-emoji-list command line option
--get-all, 660
--help, 660
--issue-iid, 660
--no-get-all, 660
--page, 660
--per-page, 660
--project-id, 660
--sudo, 660
-h, 660
gitlab-project-issue-closed-by command line option
--help, 652
--iid, 652
--project-id, 652
--sudo, 652
-h, 652
gitlab-project-issue-create command line option
--assignee-id, 653
--assignee-ids, 653
--confidential, 653
--created-at, 653
--description, 653
--discussion-to-resolve, 653
--due-date, 653
--help, 652
--labels, 653
--merge-request-to-resolve-discussions-of, 653
--milestone-id, 653
--project-id, 653
--sudo, 653
--title, 653
-h, 652
gitlab-project-issue-delete command line option
--help, 653
--iid, 653
--project-id, 653
--sudo, 653
-h, 653
gitlab-project-issue-discussion command line option
--help, 660
-h, 660
gitlab-project-issue-discussion-create command line option
--body, 661
--created-at, 661
--help, 661
--issue-iid, 661
--project-id, 661
--sudo, 661
-h, 661
gitlab-project-issue-discussion-get command line option
--help, 661
--id, 661
--issue-iid, 661
--project-id, 661
--sudo, 661
-h, 661
gitlab-project-issue-discussion-list command line option
--get-all, 662
--help, 661
--issue-iid, 661
--no-get-all, 662
--page, 661
--per-page, 661
--project-id, 661
--sudo, 661
-h, 661
gitlab-project-issue-discussion-note command line option
--help, 662
-h, 662
gitlab-project-issue-discussion-note-create command line option
--body, 662
--created-at, 662
--discussion-id, 662
--help, 662
--issue-iid, 662

--project-id, 662
--sudo, 662
-h, 662
gitlab-project-issue-discussion-note-delete command line option
--discussion-id, 663
--help, 662
--id, 663
--issue-iid, 662
--project-id, 662
--sudo, 662
-h, 662
gitlab-project-issue-discussion-note-get command line option
--discussion-id, 663
--help, 663
--id, 663
--issue-iid, 663
--project-id, 663
--sudo, 663
-h, 663
gitlab-project-issue-discussion-note-list command line option
--discussion-id, 663
--get-all, 663
--help, 663
--issue-iid, 663
--no-get-all, 663
--page, 663
--per-page, 663
--project-id, 663
--sudo, 663
-h, 663
gitlab-project-issue-discussion-note-update command line option
--body, 664
--discussion-id, 664
--help, 664
--id, 664
--issue-iid, 664
--project-id, 664
--sudo, 664
-h, 664
gitlab-project-issue-get command line option
--help, 653
--iid, 653
--project-id, 653
--sudo, 653
-h, 653
gitlab-project-issue-link command line option
--help, 664
-h, 664
gitlab-project-issue-link-create command line option
--help, 664
--issue-iid, 664
--project-id, 664
--sudo, 664
--target-issue-iid, 664
--target-project-id, 664
-h, 664
gitlab-project-issue-link-delete command line option
--help, 664
--issue-iid, 665
--issue-link-id, 665
--project-id, 665
--sudo, 665
-h, 664
gitlab-project-issue-link-list command line option
--get-all, 665
--help, 665
--issue-iid, 665
--no-get-all, 665
--page, 665
--per-page, 665
--project-id, 665
--sudo, 665
-h, 665
gitlab-project-issue-list command line option
--assignee-id, 654
--author-id, 654
--created-after, 654
--created-before, 654
--get-all, 655
--help, 654
--iids, 654
--iteration-id, 654
--labels, 654
--milestone, 654
--my-reaction-emoji, 654
--no-get-all, 655
--order-by, 654
--page, 654
--per-page, 654
--project-id, 654
--scope, 654
--search, 654
--sort, 654
--state, 654
--sudo, 654
--updated-after, 654
--updated-before, 654
-h, 654

gitlab-project-issue-move command line option
--help, 655
--iid, 655
--project-id, 655
--sudo, 655
--to-project-id, 655
-h, 655
gitlab-project-issue-note command line option
--help, 665
-h, 665
gitlab-project-issue-note-award-emoji command line option
--help, 667
-h, 667
gitlab-project-issue-note-award-emoji-create command line option
--help, 667
--issue-iid, 667
--name, 668
--note-id, 667
--project-id, 667
--sudo, 667
-h, 667
gitlab-project-issue-note-award-emoji-delete command line option
--help, 668
--id, 668
--issue-iid, 668
--note-id, 668
--project-id, 668
--sudo, 668
-h, 668
gitlab-project-issue-note-award-emoji-get command line option
--help, 668
--id, 668
--issue-iid, 668
--note-id, 668
--project-id, 668
--sudo, 668
-h, 668
gitlab-project-issue-note-award-emoji-list command line option
--get-all, 669
--help, 668
--issue-iid, 669
--no-get-all, 669
--note-id, 669
--page, 669
--per-page, 669
--project-id, 669
--sudo, 668
-h, 668
gitlab-project-issue-note-create command line option
--body, 666
--created-at, 666
--help, 665
--issue-iid, 666
--project-id, 665
--sudo, 665
-h, 665
gitlab-project-issue-note-delete command line option
--help, 666
--id, 666
--issue-iid, 666
--project-id, 666
--sudo, 666
-h, 666
gitlab-project-issue-note-get command line option
--help, 666
--id, 666
--issue-iid, 666
--project-id, 666
--sudo, 666
-h, 666
gitlab-project-issue-note-list command line option
--get-all, 667
--help, 666
--issue-iid, 666
--no-get-all, 667
--page, 667
--per-page, 667
--project-id, 666
--sudo, 666
-h, 666
gitlab-project-issue-note-update command line option
--body, 667
--help, 667
--id, 667
--issue-iid, 667
--project-id, 667
--sudo, 667
-h, 667
gitlab-project-issue-participants command line option
--help, 655
--iid, 655
--project-id, 655
--sudo, 655
-h, 655

gitlab-project-issue-related-merge-requests command line option
--help, 655
--iid, 655
--project-id, 655
--sudo, 655
-h, 655
gitlab-project-issue-reorder command line option
--help, 656
--iid, 656
--move-after-id, 656
--move-before-id, 656
--project-id, 656
--sudo, 656
-h, 656
gitlab-project-issue-reset-spent-time command line option
--help, 656
--iid, 656
--project-id, 656
--sudo, 656
-h, 656
gitlab-project-issue-reset-time-estimate command line option
--help, 656
--iid, 656
--project-id, 656
--sudo, 656
-h, 656
gitlab-project-issue-resource-iteration-event command line option
--help, 669
-h, 669
gitlab-project-issue-resource-iteration-event-get command line option
--help, 669
--id, 669
--issue-iid, 669
--project-id, 669
--sudo, 669
-h, 669
gitlab-project-issue-resource-iteration-event-list command line option
--get-all, 670
--help, 669
--issue-iid, 670
--no-get-all, 670
--page, 670
--per-page, 670
--project-id, 670
--sudo, 669
-h, 669
gitlab-project-issue-resource-label-event command line option
--help, 670
-h, 670
gitlab-project-issue-resource-label-event-get command line option
--help, 670
--id, 670
--issue-iid, 670
--project-id, 670
--sudo, 670
-h, 670
gitlab-project-issue-resource-label-event-list command line option
--get-all, 671
--help, 670
--issue-iid, 671
--no-get-all, 671
--page, 671
--per-page, 671
--project-id, 670
--sudo, 670
-h, 670
gitlab-project-issue-resource-milestone-event command line option
--help, 671
-h, 671
gitlab-project-issue-resource-milestone-event-get command line option
--help, 671
--id, 671
--issue-iid, 671
--project-id, 671
--sudo, 671
-h, 671
gitlab-project-issue-resource-milestone-event-list command line option
--get-all, 672
--help, 671
--issue-iid, 671
--no-get-all, 672
--page, 672
--per-page, 672
--project-id, 671
--sudo, 671
-h, 671
gitlab-project-issue-resource-state-event command line option
--help, 672
-h, 672
gitlab-project-issue-resource-state-event-get command line option
--help, 672
--id, 672

--issue-iid, 672
--project-id, 672
--sudo, 672
-h, 672
gitlab-project-issue-resource-state-event-list command line option
--get-all, 673
--help, 672
--issue-iid, 672
--no-get-all, 673
--page, 672
--per-page, 673
--project-id, 672
--sudo, 672
-h, 672
gitlab-project-issue-resource-weight-event command line option
--help, 673
-h, 673
gitlab-project-issue-resource-weight-event-get command line option
--help, 673
--id, 673
--issue-iid, 673
--project-id, 673
--sudo, 673
-h, 673
gitlab-project-issue-resource-weight-event-list command line option
--get-all, 674
--help, 673
--issue-iid, 673
--no-get-all, 674
--page, 673
--per-page, 673
--project-id, 673
--sudo, 673
-h, 673
gitlab-project-issue-subscribe command line option
--help, 656
--iid, 657
--project-id, 656
--sudo, 656
-h, 656
gitlab-project-issue-template command line option
--help, 674
-h, 674
gitlab-project-issue-template-get command line option
--help, 674
--name, 674
--project-id, 674
--sudo, 674
-h, 674
gitlab-project-issue-template-list command line option
--get-all, 674
--help, 674
--no-get-all, 674
--page, 674
--per-page, 674
--project-id, 674
--sudo, 674
-h, 674
gitlab-project-issue-time-estimate command line option
--duration, 657
--help, 657
--iid, 657
--project-id, 657
--sudo, 657
-h, 657
gitlab-project-issue-time-stats command line option
--help, 657
--iid, 657
--project-id, 657
--sudo, 657
-h, 657
gitlab-project-issue-todo command line option
--help, 657
--iid, 657
--project-id, 657
--sudo, 657
-h, 657
gitlab-project-issue-unsubscribe command line option
--help, 657
--iid, 658
--project-id, 658
--sudo, 658
-h, 657
gitlab-project-issue-update command line option
--assignee-id, 658
--assignee-ids, 658
--confidential, 658
--description, 658
--discussion-locked, 658
--due-date, 658
--help, 658
--iid, 658
--labels, 658
--milestone-id, 658
--project-id, 658

--state-event, 658
--sudo, 658
--title, 658
--updated-at, 658
-h, 658
gitlab-project-issue-user-agent-detail command line option
--help, 659
--iid, 659
--project-id, 659
--sudo, 659
-h, 659
gitlab-project-issues-statistics command line option
--help, 675
-h, 675
gitlab-project-issues-statistics-get command line option
--help, 675
--project-id, 675
--sudo, 675
-h, 675
gitlab-project-job command line option
--help, 675
-h, 675
gitlab-project-job-artifact command line option
--help, 675
--id, 675
--project-id, 675
--sudo, 675
-h, 675
gitlab-project-job-artifacts command line option
--help, 675
--id, 675
--project-id, 675
--sudo, 675
-h, 675
gitlab-project-job-cancel command line option
--help, 676
--id, 676
--project-id, 676
--sudo, 676
-h, 676
gitlab-project-job-delete-artifacts command line option
--help, 676
--id, 676
--project-id, 676
--sudo, 676
-h, 676
gitlab-project-job-erase command line option
--help, 676
--id, 676
--project-id, 676
--sudo, 676
-h, 676
gitlab-project-job-get command line option
--help, 676
--id, 676
--project-id, 676
--sudo, 676
-h, 676
gitlab-project-job-keep-artifacts command line option
--help, 677
--id, 677
--project-id, 677
--sudo, 677
-h, 677
gitlab-project-job-list command line option
--get-all, 677
--help, 677
--no-get-all, 677
--order-by, 677
--page, 677
--per-page, 677
--project-id, 677
--scope, 677
--sort, 677
--sudo, 677
-h, 677
gitlab-project-job-play command line option
--help, 677
--id, 677
--project-id, 677
--sudo, 677
-h, 677
gitlab-project-job-retry command line option
--help, 678
--id, 678
--project-id, 678
--sudo, 678
-h, 678
gitlab-project-job-token-scope command line option
--help, 678
-h, 678
gitlab-project-job-token-scope-get command line option
--help, 678
--project-id, 678
--sudo, 678

-h, 678
gitlab-project-job-token-scope-update command line option
--help, 678
--project-id, 679
--sudo, 678
-h, 678
gitlab-project-job-trace command line option
--help, 678
--id, 678
--project-id, 678
--sudo, 678
-h, 678
gitlab-project-key command line option
--help, 679
-h, 679
gitlab-project-key-create command line option
--can-push, 679
--expires-at, 679
--help, 679
--key, 679
--project-id, 679
--sudo, 679
--title, 679
-h, 679
gitlab-project-key-delete command line option
--help, 679
--id, 679
--project-id, 679
--sudo, 679
-h, 679
gitlab-project-key-enable command line option
--help, 679
--key-id, 680
--project-id, 679
--sudo, 679
-h, 679
gitlab-project-key-get command line option
--help, 680
--id, 680
--project-id, 680
--sudo, 680
-h, 680
gitlab-project-key-list command line option
--get-all, 680
--help, 680
--no-get-all, 680
--page, 680
--per-page, 680
--project-id, 680
--sudo, 680
-h, 680
gitlab-project-key-update command line option
--can-push, 681
--expires-at, 681
--help, 680
--id, 680
--project-id, 680
--sudo, 680
--title, 680
-h, 680
gitlab-project-label command line option
--help, 681
-h, 681
gitlab-project-label-create command line option
--color, 681
--description, 681
--help, 681
--name, 681
--priority, 681
--project-id, 681
--sudo, 681
-h, 681
gitlab-project-label-delete command line option
--help, 681
--name, 681
--project-id, 681
--sudo, 681
-h, 681
gitlab-project-label-get command line option
--help, 681
--name, 682
--project-id, 682
--sudo, 682
-h, 681
gitlab-project-label-list command line option
--get-all, 682
--help, 682
--no-get-all, 682
--page, 682
--per-page, 682
--project-id, 682
--sudo, 682
-h, 682
gitlab-project-label-subscribe command line option
--help, 682
--name, 682
--project-id, 682

--sudo, 682
-h, 682
gitlab-project-label-unsubscribe command line option
--help, 682
--name, 683
--project-id, 682
--sudo, 682
-h, 682
gitlab-project-label-update command line option
--color, 683
--description, 683
--help, 683
--name, 683
--new-name, 683
--priority, 683
--project-id, 683
--sudo, 683
-h, 683
gitlab-project-languages command line option
--help, 582
--id, 582
-h, 582
gitlab-project-license-template command line option
--help, 683
-h, 683
gitlab-project-license-template-get command line option
--help, 683
--key, 683
--project-id, 683
--sudo, 683
-h, 683
gitlab-project-license-template-list command line option
--get-all, 684
--help, 683
--no-get-all, 684
--page, 684
--per-page, 684
--project-id, 684
--sudo, 684
-h, 683
gitlab-project-list command line option
--archived, 582
--get-all, 583
--help, 582
--id-after, 582
--id-before, 582
--last-activity-after, 582
--last-activity-before, 582
--membership, 582
--min-access-level, 583
--no-get-all, 583
--order-by, 583
--owned, 583
--page, 583
--per-page, 583
--repository-checksum-failed, 583
--repository-storage, 583
--search, 583
--search-namespaces, 583
--simple, 583
--sort, 583
--starred, 583
--statistics, 583
--sudo, 582
--topic, 583
--visibility, 583
--wiki-checksum-failed, 583
--with-custom-attributes, 583
--with-issues-enabled, 583
--with-merge-requests-enabled, 583
--with-programming-language, 583
-h, 582
gitlab-project-member command line option
--help, 684
-h, 684
gitlab-project-member-all command line option
--help, 686
-h, 686
gitlab-project-member-all-get command line option
--help, 686
--id, 686
--project-id, 686
--sudo, 686
-h, 686
gitlab-project-member-all-list command line option
--get-all, 686
--help, 686
--no-get-all, 686
--page, 686
--per-page, 686
--project-id, 686
--sudo, 686
-h, 686
gitlab-project-member-create command line option
--access-level, 684
--expires-at, 684
--help, 684
--project-id, 684

--sudo, 684
--tasks-to-be-done, 684
--user-id, 684
--username, 684
-h, 684
gitlab-project-member-delete command line option
--help, 684
--id, 685
--project-id, 685
--sudo, 684
-h, 684
gitlab-project-member-get command line option
--help, 685
--id, 685
--project-id, 685
--sudo, 685
-h, 685
gitlab-project-member-list command line option
--get-all, 685
--help, 685
--no-get-all, 685
--page, 685
--per-page, 685
--project-id, 685
--sudo, 685
-h, 685
gitlab-project-member-update command line option
--access-level, 686
--expires-at, 686
--help, 685
--id, 685
--project-id, 685
--sudo, 685
-h, 685
gitlab-project-merge-request command line option
--help, 687
-h, 687
gitlab-project-merge-request-add-spent-time command line option
--duration, 687
--help, 687
--iid, 687
--project-id, 687
--sudo, 687
-h, 687
gitlab-project-merge-request-approval command line option
--help, 696
-h, 696
gitlab-project-merge-request-approval-get command line option
--help, 697
--mr-iid, 697
--project-id, 697
--sudo, 697
-h, 697
gitlab-project-merge-request-approval-rule command line option
--help, 697
-h, 697
gitlab-project-merge-request-approval-rule-create command line option
--approval-project-rule-id, 698
--approvals-required, 698
--group-ids, 698
--help, 697
--merge-request-iid, 698
--name, 698
--project-id, 698
--sudo, 698
--user-ids, 698
--usernames, 698
-h, 697
gitlab-project-merge-request-approval-rule-delete command line option
--help, 698
--id, 698
--merge-request-iid, 698
--project-id, 698
--sudo, 698
-h, 698
gitlab-project-merge-request-approval-rule-get command line option
--help, 698
--id, 698
--merge-request-iid, 698
--project-id, 698
--sudo, 698
-h, 698
gitlab-project-merge-request-approval-rule-list command line option
--get-all, 699
--help, 699
--merge-request-iid, 699
--no-get-all, 699
--page, 699
--per-page, 699
--project-id, 699
--sudo, 699
-h, 699
gitlab-project-merge-request-approval-rule-update command line option
--approvals-required, 699

--group-ids, 700
--help, 699
--id, 699
--merge-request-iid, 699
--name, 699
--project-id, 699
--sudo, 699
--user-ids, 700
--usernames, 700
-h, 699
gitlab-project-merge-request-approval-state command line option
--help, 700
-h, 700
gitlab-project-merge-request-approval-state-get command line option
--help, 700
--id, 700
--mr-iid, 700
--project-id, 700
--sudo, 700
-h, 700
gitlab-project-merge-request-approval-update command line option
--approvals-required, 697
--help, 697
--mr-iid, 697
--project-id, 697
--sudo, 697
-h, 697
gitlab-project-merge-request-approve command line option
--help, 687
--iid, 687
--project-id, 687
--sha, 687
--sudo, 687
-h, 687
gitlab-project-merge-request-award-emoji command line option
--help, 700
-h, 700
gitlab-project-merge-request-award-emoji-create command line option
--help, 700
--mr-iid, 700
--name, 700
--project-id, 700
--sudo, 700
-h, 700
gitlab-project-merge-request-award-emoji-delete command line option
--help, 701
--id, 701
--mr-iid, 701
--project-id, 701
--sudo, 701
-h, 701
gitlab-project-merge-request-award-emoji-get command line option
--help, 701
--id, 701
--mr-iid, 701
--project-id, 701
--sudo, 701
-h, 701
gitlab-project-merge-request-award-emoji-list command line option
--get-all, 701
--help, 701
--mr-iid, 701
--no-get-all, 702
--page, 701
--per-page, 701
--project-id, 701
--sudo, 701
-h, 701
gitlab-project-merge-request-cancel-merge-when-pipeline-su command line option
--help, 687
--iid, 688
--project-id, 687
--sudo, 687
-h, 687
gitlab-project-merge-request-changes command line option
--access-raw-diffs, 688
--help, 688
--iid, 688
--project-id, 688
--sudo, 688
-h, 688
gitlab-project-merge-request-closes-issues command line option
--help, 688
--iid, 688
--project-id, 688
--sudo, 688
-h, 688
gitlab-project-merge-request-commits command line option
--help, 688
--iid, 688
--project-id, 688
--sudo, 688
-h, 688
gitlab-project-merge-request-create command line option

--allow-collaboration, 689
--allow-maintainer-to-push, 689
--approvals-before-merge, 689
--assignee-id, 689
--assignee-ids, 689
--description, 689
--help, 689
--labels, 689
--milestone-id, 689
--project-id, 689
--remove-source-branch, 689
--reviewer-ids, 689
--source-branch, 689
--squash, 689
--sudo, 689
--target-branch, 689
--target-project-id, 689
--title, 689
-h, 689
gitlab-project-merge-request-delete command line option
--help, 690
--iid, 690
--project-id, 690
--sudo, 690
-h, 690
gitlab-project-merge-request-diff command line option
--help, 702
-h, 702
gitlab-project-merge-request-diff-get command line option
--help, 702
--id, 702
--mr-iid, 702
--project-id, 702
--sudo, 702
-h, 702
gitlab-project-merge-request-diff-list command line option
--get-all, 702
--help, 702
--mr-iid, 702
--no-get-all, 702
--page, 702
--per-page, 702
--project-id, 702
--sudo, 702
-h, 702
gitlab-project-merge-request-discussion command line option
--help, 703
-h, 703
gitlab-project-merge-request-discussion-create command line option
--body, 703
--created-at, 703
--help, 703
--mr-iid, 703
--position, 703
--project-id, 703
--sudo, 703
-h, 703
gitlab-project-merge-request-discussion-get command line option
--help, 703
--id, 703
--mr-iid, 703
--project-id, 703
--sudo, 703
-h, 703
gitlab-project-merge-request-discussion-list command line option
--get-all, 704
--help, 703
--mr-iid, 704
--no-get-all, 704
--page, 704
--per-page, 704
--project-id, 704
--sudo, 704
-h, 703
gitlab-project-merge-request-discussion-note command line option
--help, 704
-h, 704
gitlab-project-merge-request-discussion-note-create command line option
--body, 705
--created-at, 705
--discussion-id, 705
--help, 704
--mr-iid, 705
--project-id, 705
--sudo, 705
-h, 704
gitlab-project-merge-request-discussion-note-delete command line option
--discussion-id, 705
--help, 705
--id, 705
--mr-iid, 705
--project-id, 705
--sudo, 705
-h, 705
gitlab-project-merge-request-discussion-note-get command line option

--discussion-id, 705
--help, 705
--id, 705
--mr-iid, 705
--project-id, 705
--sudo, 705
-h, 705
gitlab-project-merge-request-discussion-note-list command line option
--discussion-id, 706
--get-all, 706
--help, 706
--mr-iid, 706
--no-get-all, 706
--page, 706
--per-page, 706
--project-id, 706
--sudo, 706
-h, 706
gitlab-project-merge-request-discussion-note-update command line option
--body, 706
--discussion-id, 706
--help, 706
--id, 706
--mr-iid, 706
--project-id, 706
--sudo, 706
-h, 706
gitlab-project-merge-request-discussion-update command line option
--help, 704
--id, 704
--mr-iid, 704
--project-id, 704
--resolved, 704
--sudo, 704
-h, 704
gitlab-project-merge-request-draft-note command line option
--help, 707
-h, 707
gitlab-project-merge-request-draft-note-create command line option
--commit-id, 707
--help, 707
--in-reply-to-discussion-id, 707
--mr-iid, 707
--note, 707
--position, 707
--project-id, 707
--resolve-discussion, 707
--sudo, 707
-h, 707
gitlab-project-merge-request-draft-note-delete command line option
--help, 707
--id, 707
--mr-iid, 707
--project-id, 707
--sudo, 707
-h, 707
gitlab-project-merge-request-draft-note-get command line option
--help, 708
--id, 708
--mr-iid, 708
--project-id, 708
--sudo, 708
-h, 708
gitlab-project-merge-request-draft-note-list command line option
--get-all, 708
--help, 708
--mr-iid, 708
--no-get-all, 708
--page, 708
--per-page, 708
--project-id, 708
--sudo, 708
-h, 708
gitlab-project-merge-request-draft-note-update command line option
--help, 708
--id, 709
--mr-iid, 709
--position, 709
--project-id, 708
--sudo, 708
-h, 708
gitlab-project-merge-request-get command line option
--help, 690
--iid, 690
--include-diverged-commits-count, 690
--include-rebase-in-progress, 690
--project-id, 690
--render-html, 690
--sudo, 690
-h, 690
gitlab-project-merge-request-list command line option
--approved-by-ids, 691
--approver-ids, 691
--assignee-id, 691
--author-id, 691
--created-after, 691
--created-before, 691

--get-all, 692
--help, 691
--iids, 691
--labels, 691
--milestone, 691
--my-reaction-emoji, 691
--no-get-all, 692
--order-by, 691
--page, 691
--per-page, 691
--project-id, 691
--scope, 691
--search, 691
--sort, 691
--source-branch, 691
--state, 691
--sudo, 691
--target-branch, 691
--updated-after, 691
--updated-before, 691
--view, 691
--wip, 691
-h, 691
gitlab-project-merge-request-merge command line option
--help, 692
--iid, 692
--merge-commit-message, 692
--merge-when-pipeline-succeeds, 692
--project-id, 692
--should-remove-source-branch, 692
--sudo, 692
-h, 692
gitlab-project-merge-request-merge-ref command line option
--help, 692
--iid, 692
--project-id, 692
--sudo, 692
-h, 692
gitlab-project-merge-request-note command line option
--help, 709
-h, 709
gitlab-project-merge-request-note-award-emoji command line option
--help, 711
-h, 711
gitlab-project-merge-request-note-award-emoji-create command line option
--help, 711
--mr-iid, 711
--name, 711
--note-id, 711
--project-id, 711
--sudo, 711
-h, 711
gitlab-project-merge-request-note-award-emoji-delete command line option
--help, 711
--id, 711
--mr-iid, 711
--note-id, 711
--project-id, 711
--sudo, 711
-h, 711
gitlab-project-merge-request-note-award-emoji-get command line option
--help, 711
--id, 712
--mr-iid, 712
--note-id, 712
--project-id, 712
--sudo, 711
-h, 711
gitlab-project-merge-request-note-award-emoji-list command line option
--get-all, 712
--help, 712
--mr-iid, 712
--no-get-all, 712
--note-id, 712
--page, 712
--per-page, 712
--project-id, 712
--sudo, 712
-h, 712
gitlab-project-merge-request-note-create command line option
--body, 709
--help, 709
--mr-iid, 709
--project-id, 709
--sudo, 709
-h, 709
gitlab-project-merge-request-note-delete command line option
--help, 709
--id, 709
--mr-iid, 709
--project-id, 709
--sudo, 709
-h, 709
gitlab-project-merge-request-note-get command line option
--help, 709
--id, 710
--mr-iid, 710

--project-id, 710
--sudo, 710
-h, 709
gitlab-project-merge-request-note-list command line option
--get-all, 710
--help, 710
--mr-iid, 710
--no-get-all, 710
--page, 710
--per-page, 710
--project-id, 710
--sudo, 710
-h, 710
gitlab-project-merge-request-note-update command line option
--body, 710
--help, 710
--id, 710
--mr-iid, 710
--project-id, 710
--sudo, 710
-h, 710
gitlab-project-merge-request-participants command line option
--help, 692
--iid, 693
--project-id, 692
--sudo, 692
-h, 692
gitlab-project-merge-request-pipeline command line option
--help, 712
-h, 712
gitlab-project-merge-request-pipeline-create command line option
--help, 712
--mr-iid, 713
--project-id, 713
--sudo, 713
-h, 712
gitlab-project-merge-request-pipeline-list command line option
--get-all, 713
--help, 713
--mr-iid, 713
--no-get-all, 713
--page, 713
--per-page, 713
--project-id, 713
--sudo, 713
-h, 713
gitlab-project-merge-request-rebase command line option
--help, 693
--iid, 693
--project-id, 693
--sudo, 693
-h, 693
gitlab-project-merge-request-related-issues command line option
--help, 693
--iid, 693
--project-id, 693
--sudo, 693
-h, 693
gitlab-project-merge-request-reset-approvals command line option
--help, 693
--iid, 693
--project-id, 693
--sudo, 693
-h, 693
gitlab-project-merge-request-reset-spent-time command line option
--help, 693
--iid, 694
--project-id, 694
--sudo, 694
-h, 693
gitlab-project-merge-request-reset-time-estimate command line option
--help, 694
--iid, 694
--project-id, 694
--sudo, 694
-h, 694
gitlab-project-merge-request-resource-label-event command line option
--help, 713
-h, 713
gitlab-project-merge-request-resource-label-event-get command line option
--help, 713
--id, 714
--mr-iid, 713
--project-id, 713
--sudo, 713
-h, 713
gitlab-project-merge-request-resource-label-event-list command line option
--get-all, 714
--help, 714
--mr-iid, 714
--no-get-all, 714
--page, 714
--per-page, 714
--project-id, 714

--sudo, 714
-h, 714
gitlab-project-merge-request-resource-milestone-event-list command line option
--help, 714
-h, 714
gitlab-project-merge-request-resource-milestone-event-get command line option
--help, 714
--id, 714
--mr-iid, 714
--project-id, 714
--sudo, 714
-h, 714
gitlab-project-merge-request-resource-milestone-event-list command line option
--get-all, 715
--help, 715
--mr-iid, 715
--no-get-all, 715
--page, 715
--per-page, 715
--project-id, 715
--sudo, 715
-h, 715
gitlab-project-merge-request-resource-state-event command line option
--help, 715
-h, 715
gitlab-project-merge-request-resource-state-event-get command line option
--help, 715
--id, 715
--mr-iid, 715
--project-id, 715
--sudo, 715
-h, 715
gitlab-project-merge-request-resource-state-event-list command line option
--get-all, 716
--help, 715
--mr-iid, 716
--no-get-all, 716
--page, 716
--per-page, 716
--project-id, 716
--sudo, 715
-h, 715
gitlab-project-merge-request-reviewer-detail command line option
--help, 716
-h, 716
gitlab-project-merge-request-reviewer-detail-list command line option
--get-all, 716
--help, 716
--mr-iid, 716
--no-get-all, 716
--page, 716
--per-page, 716
--project-id, 716
--sudo, 716
-h, 716
gitlab-project-merge-request-status-check command line option
--help, 716
-h, 716
gitlab-project-merge-request-status-check-list command line option
--get-all, 717
--help, 717
--merge-request-iid, 717
--no-get-all, 717
--page, 717
--per-page, 717
--project-id, 717
--sudo, 717
-h, 717
gitlab-project-merge-request-subscribe command line option
--help, 694
--iid, 694
--project-id, 694
--sudo, 694
-h, 694
gitlab-project-merge-request-template command line option
--help, 717
-h, 717
gitlab-project-merge-request-template-get command line option
--help, 717
--name, 717
--project-id, 717
--sudo, 717
-h, 717
gitlab-project-merge-request-template-list command line option
--get-all, 718
--help, 718
--no-get-all, 718
--page, 718
--per-page, 718
--project-id, 718
--sudo, 718
-h, 718
gitlab-project-merge-request-time-estimate command line option

--duration, 694
--help, 694
--iid, 694
--project-id, 694
--sudo, 694
-h, 694
gitlab-project-merge-request-time-stats command line option
--help, 695
--iid, 695
--project-id, 695
--sudo, 695
-h, 695
gitlab-project-merge-request-todo command line option
--help, 695
--iid, 695
--project-id, 695
--sudo, 695
-h, 695
gitlab-project-merge-request-unapprove command line option
--help, 695
--iid, 695
--project-id, 695
--sudo, 695
-h, 695
gitlab-project-merge-request-unsubscribe command line option
--help, 695
--iid, 695
--project-id, 695
--sudo, 695
-h, 695
gitlab-project-merge-request-update command line option
--allow-maintainer-to-push, 696
--assignee-id, 696
--description, 696
--discussion-locked, 696
--help, 696
--iid, 696
--labels, 696
--milestone-id, 696
--project-id, 696
--remove-source-branch, 696
--reviewer-ids, 696
--squash, 696
--state-event, 696
--sudo, 696
--target-branch, 696
--title, 696
-h, 696
gitlab-project-merge-train command line option
--help, 718
-h, 718
gitlab-project-merge-train-list command line option
--get-all, 718
--help, 718
--no-get-all, 718
--page, 718
--per-page, 718
--project-id, 718
--scope, 718
--sudo, 718
-h, 718
gitlab-project-milestone command line option
--help, 719
-h, 719
gitlab-project-milestone-create command line option
--description, 719
--due-date, 719
--help, 719
--project-id, 719
--start-date, 719
--state-event, 719
--sudo, 719
--title, 719
-h, 719
gitlab-project-milestone-delete command line option
--help, 719
--id, 719
--project-id, 719
--sudo, 719
-h, 719
gitlab-project-milestone-get command line option
--help, 719
--id, 720
--project-id, 720
--sudo, 719
-h, 719
gitlab-project-milestone-issues command line option
--help, 720
--id, 720
--project-id, 720
--sudo, 720
-h, 720
gitlab-project-milestone-list command line option
--get-all, 720

--help, 720
--iids, 720
--no-get-all, 720
--page, 720
--per-page, 720
--project-id, 720
--search, 720
--state, 720
--sudo, 720
-h, 720
gitlab-project-milestone-merge-requests command line option
--help, 720
--id, 721
--project-id, 721
--sudo, 721
-h, 720
gitlab-project-milestone-update command line option
--description, 721
--due-date, 721
--help, 721
--id, 721
--project-id, 721
--start-date, 721
--state-event, 721
--sudo, 721
--title, 721
-h, 721
gitlab-project-mirror-pull command line option
--help, 583
--id, 583
-h, 583
gitlab-project-mirror-pull-details command line option
--help, 584
--id, 584
-h, 584
gitlab-project-note command line option
--help, 721
-h, 721
gitlab-project-note-get command line option
--help, 721
--id, 721
--project-id, 721
--sudo, 721
-h, 721
gitlab-project-note-list command line option
--get-all, 722
--help, 722
--no-get-all, 722
--page, 722
--per-page, 722
--project-id, 722
--sudo, 722
-h, 722
gitlab-project-notification-settings command line option
--help, 722
-h, 722
gitlab-project-notification-settings-get command line option
--help, 722
--project-id, 722
--sudo, 722
-h, 722
gitlab-project-notification-settings-update command line option
--close-issue, 723
--close-merge-request, 723
--help, 723
--level, 723
--merge-merge-request, 723
--new-issue, 723
--new-merge-request, 723
--new-note, 723
--notification-email, 723
--project-id, 723
--reassign-issue, 723
--reassign-merge-request, 723
--reopen-issue, 723
--reopen-merge-request, 723
--sudo, 723
-h, 723
gitlab-project-package command line option
--help, 723
-h, 723
gitlab-project-package-delete command line option
--help, 723
--id, 724
--project-id, 724
--sudo, 723
-h, 723
gitlab-project-package-file command line option
--help, 725
-h, 725
gitlab-project-package-file-delete command line option
--help, 725
--id, 725
--package-id, 725
--project-id, 725
--sudo, 725
-h, 725

gitlab-project-package-file-list command line option
--get-all, 725
--help, 725
--no-get-all, 725
--package-id, 725
--page, 725
--per-page, 725
--project-id, 725
--sudo, 725
-h, 725
gitlab-project-package-get command line option
--help, 724
--id, 724
--project-id, 724
--sudo, 724
-h, 724
gitlab-project-package-list command line option
--get-all, 724
--help, 724
--include-versionless, 724
--no-get-all, 725
--order-by, 724
--package-name, 724
--package-type, 724
--package-version, 724
--page, 724
--per-page, 724
--project-id, 724
--sort, 724
--status, 724
--sudo, 724
-h, 724
gitlab-project-package-pipeline command line option
--help, 726
-h, 726
gitlab-project-package-pipeline-list command line option
--get-all, 726
--help, 726
--no-get-all, 726
--package-id, 726
--page, 726
--per-page, 726
--project-id, 726
--sudo, 726
-h, 726
gitlab-project-package-protection-rule command line option
--help, 726
-h, 726
gitlab-project-package-protection-rule-create command line option
--help, 726
--minimum-access-level-for-push, 727
--package-name-pattern, 727
--package-type, 727
--project-id, 727
--sudo, 726
-h, 726
gitlab-project-package-protection-rule-delete command line option
--help, 727
--id, 727
--project-id, 727
--sudo, 727
-h, 727
gitlab-project-package-protection-rule-list command line option
--get-all, 727
--help, 727
--no-get-all, 727
--page, 727
--per-page, 727
--project-id, 727
--sudo, 727
-h, 727
gitlab-project-package-protection-rule-update command line option
--help, 728
--id, 728
--minimum-access-level-for-push, 728
--package-name-pattern, 728
--package-type, 728
--project-id, 728
--sudo, 728
-h, 728
gitlab-project-pages command line option
--help, 728
-h, 728
gitlab-project-pages-delete command line option
--help, 728
--project-id, 728
--sudo, 728
-h, 728
gitlab-project-pages-domain command line option
--help, 729
-h, 729
gitlab-project-pages-domain-create command line option
--certificate, 729
--domain, 729
--help, 729

--key, 729
--project-id, 729
--sudo, 729
-h, 729
gitlab-project-pages-domain-delete command line option
--domain, 729
--help, 729
--project-id, 729
--sudo, 729
-h, 729
gitlab-project-pages-domain-get command line option
--domain, 730
--help, 730
--project-id, 730
--sudo, 730
-h, 730
gitlab-project-pages-domain-list command line option
--get-all, 730
--help, 730
--no-get-all, 730
--page, 730
--per-page, 730
--project-id, 730
--sudo, 730
-h, 730
gitlab-project-pages-domain-update command line option
--certificate, 730
--domain, 730
--help, 730
--key, 730
--project-id, 730
--sudo, 730
-h, 730
gitlab-project-pages-get command line option
--help, 728
--project-id, 728
--sudo, 728
-h, 728
gitlab-project-pages-update command line option
--help, 728
--pages-https-only, 729
--pages-unique-domain-enabled, 729
--project-id, 729
--sudo, 729
-h, 728
gitlab-project-pipeline command line option
--help, 731
-h, 731
gitlab-project-pipeline-bridge command line option
--help, 733
-h, 733
gitlab-project-pipeline-bridge-list command line option
--get-all, 733
--help, 733
--no-get-all, 733
--page, 733
--per-page, 733
--pipeline-id, 733
--project-id, 733
--scope, 733
--sudo, 733
-h, 733
gitlab-project-pipeline-cancel command line option
--help, 731
--id, 731
--project-id, 731
--sudo, 731
-h, 731
gitlab-project-pipeline-create command line option
--help, 731
--project-id, 731
--ref, 731
--sudo, 731
-h, 731
gitlab-project-pipeline-delete command line option
--help, 731
--id, 731
--project-id, 731
--sudo, 731
-h, 731
gitlab-project-pipeline-get command line option
--help, 731
--id, 732
--project-id, 732
--sudo, 732
-h, 731
gitlab-project-pipeline-job command line option
--help, 733
-h, 733
gitlab-project-pipeline-job-list command line option
--get-all, 734
--help, 734
--include-retried, 734
--no-get-all, 734

--page, 734
--per-page, 734
--pipeline-id, 734
--project-id, 734
--scope, 734
--sudo, 734
-h, 734
gitlab-project-pipeline-list command line option
--get-all, 732
--help, 732
--name, 732
--no-get-all, 732
--order-by, 732
--page, 732
--per-page, 732
--project-id, 732
--ref, 732
--scope, 732
--sha, 732
--sort, 732
--source, 732
--status, 732
--sudo, 732
--username, 732
--yaml-errors, 732
-h, 732
gitlab-project-pipeline-retry command line option
--help, 733
--id, 733
--project-id, 733
--sudo, 733
-h, 733
gitlab-project-pipeline-schedule command line option
--help, 734
-h, 734
gitlab-project-pipeline-schedule-create command line option
--active, 735
--cron, 735
--cron-timezone, 735
--description, 735
--help, 734
--project-id, 734
--ref, 735
--sudo, 734
-h, 734
gitlab-project-pipeline-schedule-delete command line option
--help, 735
--id, 735
--project-id, 735
--sudo, 735
-h, 735
gitlab-project-pipeline-schedule-get command line option
--help, 735
--id, 735
--project-id, 735
--sudo, 735
-h, 735
gitlab-project-pipeline-schedule-list command line option
--get-all, 736
--help, 735
--no-get-all, 736
--page, 735
--per-page, 735
--project-id, 735
--sudo, 735
-h, 735
gitlab-project-pipeline-schedule-pipeline command line option
--help, 737
-h, 737
gitlab-project-pipeline-schedule-pipeline-list command line option
--get-all, 737
--help, 737
--no-get-all, 737
--page, 737
--per-page, 737
--pipeline-schedule-id, 737
--project-id, 737
--sudo, 737
-h, 737
gitlab-project-pipeline-schedule-play command line option
--help, 736
--id, 736
--project-id, 736
--sudo, 736
-h, 736
gitlab-project-pipeline-schedule-take-ownership command line option
--help, 736
--id, 736
--project-id, 736
--sudo, 736
-h, 736
gitlab-project-pipeline-schedule-update command line option
--active, 737
--cron, 737
--cron-timezone, 737
--description, 736

--help, 736
--id, 736
--project-id, 736
--ref, 736
--sudo, 736
-h, 736
gitlab-project-pipeline-schedule-variable command line option
--help, 737
-h, 737
gitlab-project-pipeline-schedule-variable-create command line option
--help, 738
--key, 738
--pipeline-schedule-id, 738
--project-id, 738
--sudo, 738
--value, 738
-h, 738
gitlab-project-pipeline-schedule-variable-delete command line option
--help, 738
--key, 738
--pipeline-schedule-id, 738
--project-id, 738
--sudo, 738
-h, 738
gitlab-project-pipeline-schedule-variable-update command line option
--help, 738
--key, 739
--pipeline-schedule-id, 738
--project-id, 738
--sudo, 738
--value, 739
-h, 738
gitlab-project-pipeline-test-report command line option
--help, 739
-h, 739
gitlab-project-pipeline-test-report-get command line option
--help, 739
--pipeline-id, 739
--project-id, 739
--sudo, 739
-h, 739
gitlab-project-pipeline-test-report-summary command line option
--help, 739
-h, 739
gitlab-project-pipeline-test-report-summary-get command line option
--help, 739
--pipeline-id, 739
--project-id, 739
--sudo, 739
-h, 739
gitlab-project-pipeline-variable command line option
--help, 739
-h, 739
gitlab-project-pipeline-variable-list command line option
--get-all, 740
--help, 740
--no-get-all, 740
--page, 740
--per-page, 740
--pipeline-id, 740
--project-id, 740
--sudo, 740
-h, 740
gitlab-project-protected-branch command line option
--help, 740
-h, 740
gitlab-project-protected-branch-create command line option
--allow-force-push, 741
--allowed-to-merge, 741
--allowed-to-push, 741
--allowed-to-unprotect, 741
--code-owner-approval-required, 741
--help, 740
--merge-access-level, 741
--name, 741
--project-id, 741
--push-access-level, 741
--sudo, 740
--unprotect-access-level, 741
-h, 740
gitlab-project-protected-branch-delete command line option
--help, 741
--name, 741
--project-id, 741
--sudo, 741
-h, 741
gitlab-project-protected-branch-get command line option
--help, 741
--name, 741
--project-id, 741
--sudo, 741
-h, 741
gitlab-project-protected-branch-list command line option

--get-all, 742
--help, 741
--no-get-all, 742
--page, 742
--per-page, 742
--project-id, 742
--sudo, 742
-h, 741
gitlab-project-protected-branch-update command line option
--help, 742
--name, 742
--project-id, 742
--sudo, 742
-h, 742
gitlab-project-protected-environment command line option
--help, 742
-h, 742
gitlab-project-protected-environment-create command line option
--approval-rules, 743
--deploy-access-levels, 743
--help, 742
--name, 743
--project-id, 742
--required-approval-count, 743
--sudo, 742
-h, 742
gitlab-project-protected-environment-delete command line option
--help, 743
--name, 743
--project-id, 743
--sudo, 743
-h, 743
gitlab-project-protected-environment-get command line option
--help, 743
--name, 743
--project-id, 743
--sudo, 743
-h, 743
gitlab-project-protected-environment-list command line option
--get-all, 744
--help, 743
--no-get-all, 744
--page, 743
--per-page, 743
--project-id, 743
--sudo, 743
-h, 743
gitlab-project-protected-tag command line option
--help, 744
-h, 744
gitlab-project-protected-tag-create command line option
--create-access-level, 744
--help, 744
--name, 744
--project-id, 744
--sudo, 744
-h, 744
gitlab-project-protected-tag-delete command line option
--help, 744
--name, 744
--project-id, 744
--sudo, 744
-h, 744
gitlab-project-protected-tag-get command line option
--help, 744
--name, 745
--project-id, 744
--sudo, 744
-h, 744
gitlab-project-protected-tag-list command line option
--get-all, 745
--help, 745
--no-get-all, 745
--page, 745
--per-page, 745
--project-id, 745
--sudo, 745
-h, 745
gitlab-project-pull-mirror command line option
--help, 745
-h, 745
gitlab-project-pull-mirror-create command line option
--help, 745
--project-id, 745
--sudo, 745
-h, 745
gitlab-project-pull-mirror-get command line option
--help, 745
--project-id, 746
--sudo, 745
-h, 745
gitlab-project-pull-mirror-start command line option

--help, 746
--project-id, 746
--sudo, 746
-h, 746
gitlab-project-pull-mirror-update command line option
--help, 746
--project-id, 746
--sudo, 746
--url, 746
-h, 746
gitlab-project-push-rules command line option
--help, 746
-h, 746
gitlab-project-push-rules-create command line option
--author-email-regex, 747
--branch-name-regex, 747
--commit-committer-check, 747
--commit-message-negative-regex, 747
--commit-message-regex, 747
--deny-delete-tag, 747
--file-name-regex, 747
--help, 746
--max-file-size, 747
--member-check, 747
--prevent-secrets, 747
--project-id, 747
--reject-unsigned-commits, 747
--sudo, 747
-h, 746
gitlab-project-push-rules-delete command line option
--help, 747
--project-id, 747
--sudo, 747
-h, 747
gitlab-project-push-rules-get command line option
--help, 747
--project-id, 747
--sudo, 747
-h, 747
gitlab-project-push-rules-update command line option
--author-email-regex, 748
--branch-name-regex, 748
--commit-committer-check, 748
--commit-message-negative-regex, 748
--commit-message-regex, 748
--deny-delete-tag, 748
--file-name-regex, 748
--help, 748
--max-file-size, 748
--member-check, 748
--prevent-secrets, 748
--project-id, 748
--reject-unsigned-commits, 748
--sudo, 748
-h, 748
gitlab-project-registry-protection-rule command line option
--help, 748
-h, 748
gitlab-project-registry-protection-rule-create command line option
--help, 749
--minimum-access-level-for-delete, 749
--minimum-access-level-for-push, 749
--project-id, 749
--repository-path-pattern, 749
--sudo, 749
-h, 749
gitlab-project-registry-protection-rule-list command line option
--get-all, 749
--help, 749
--no-get-all, 749
--page, 749
--per-page, 749
--project-id, 749
--sudo, 749
-h, 749
gitlab-project-registry-protection-rule-update command line option
--help, 750
--id, 750
--minimum-access-level-for-delete, 750
--minimum-access-level-for-push, 750
--project-id, 750
--repository-path-pattern, 750
--sudo, 750
-h, 750
gitlab-project-registry-repository command line option
--help, 750
-h, 750
gitlab-project-registry-repository-delete command line option
--help, 750
--id, 750
--project-id, 750
--sudo, 750
-h, 750
gitlab-project-registry-repository-list command line option
--get-all, 751

--help, 750
--no-get-all, 751
--page, 751
--per-page, 751
--project-id, 750
--sudo, 750
-h, 750
gitlab-project-registry-repository-protection-rule command line option
--help, 751
-h, 751
gitlab-project-registry-repository-protection-rule-create command line option
--help, 751
--minimum-access-level-for-delete, 751
--minimum-access-level-for-push, 751
--project-id, 751
--repository-path-pattern, 751
--sudo, 751
-h, 751
gitlab-project-registry-repository-protection-rule-delete command line option
--help, 751
--id, 751
--project-id, 751
--sudo, 751
-h, 751
gitlab-project-registry-repository-protection-rule-list command line option
--get-all, 752
--help, 752
--no-get-all, 752
--page, 752
--per-page, 752
--project-id, 752
--sudo, 752
-h, 752
gitlab-project-registry-repository-protection-rule-update command line option
--help, 752
--id, 752
--minimum-access-level-for-delete, 752
--minimum-access-level-for-push, 752
--project-id, 752
--repository-path-pattern, 752
--sudo, 752
-h, 752
gitlab-project-registry-tag command line option
--help, 752
-h, 752
gitlab-project-registry-tag-delete command line option
--help, 753
--name, 753
--project-id, 753
--repository-id, 753
--sudo, 753
-h, 753
gitlab-project-registry-tag-delete-in-bulk command line option
--help, 753
--keep-n, 753
--name, 753
--name-regex-delete, 753
--name-regex-keep, 753
--older-than, 753
--project-id, 753
--repository-id, 753
--sudo, 753
-h, 753
gitlab-project-registry-tag-get command line option
--help, 753
--name, 754
--project-id, 754
--repository-id, 754
--sudo, 753
-h, 753
gitlab-project-registry-tag-list command line option
--get-all, 754
--help, 754
--no-get-all, 754
--page, 754
--per-page, 754
--project-id, 754
--repository-id, 754
--sudo, 754
-h, 754
gitlab-project-release command line option
--help, 754
-h, 754
gitlab-project-release-create command line option
--assets, 755
--description, 755
--help, 754
--name, 755
--project-id, 754
--ref, 755
--sudo, 754
--tag-name, 754
-h, 754
gitlab-project-release-delete command line option
--help, 755
--project-id, 755

--sudo, 755
--tag-name, 755
-h, 755
gitlab-project-release-get command line option
--help, 755
--project-id, 755
--sudo, 755
--tag-name, 755
-h, 755
gitlab-project-release-link command line option
--help, 756
-h, 756
gitlab-project-release-link-create command line option
--direct-asset-path, 757
--filepath, 757
--help, 756
--link-type, 757
--name, 757
--project-id, 757
--sudo, 757
--tag-name, 757
--url, 757
-h, 756
gitlab-project-release-link-delete command line option
--help, 757
--id, 757
--project-id, 757
--sudo, 757
--tag-name, 757
-h, 757
gitlab-project-release-link-get command line option
--help, 757
--id, 757
--project-id, 757
--sudo, 757
--tag-name, 757
-h, 757
gitlab-project-release-link-list command line option
--get-all, 758
--help, 758
--no-get-all, 758
--page, 758
--per-page, 758
--project-id, 758
--sudo, 758
--tag-name, 758
-h, 758
gitlab-project-release-link-update command line option
--direct-asset-path, 758
--filepath, 758
--help, 758
--id, 758
--link-type, 758
--name, 758
--project-id, 758
--sudo, 758
--tag-name, 758
--url, 758
-h, 758
gitlab-project-release-list command line option
--get-all, 756
--help, 755
--include-html-description, 756
--no-get-all, 756
--order-by, 755
--page, 756
--per-page, 756
--project-id, 755
--sort, 755
--sudo, 755
-h, 755
gitlab-project-release-update command line option
--description, 756
--help, 756
--milestones, 756
--name, 756
--project-id, 756
--released-at, 756
--sudo, 756
--tag-name, 756
-h, 756
gitlab-project-remote-mirror command line option
--help, 759
-h, 759
gitlab-project-remote-mirror-create command line option
--enabled, 759
--help, 759
--only-protected-branches, 759
--project-id, 759
--sudo, 759
--url, 759
-h, 759
gitlab-project-remote-mirror-delete command line option
--help, 759
--id, 759

--project-id, 759
--sudo, 759
-h, 759
gitlab-project-remote-mirror-list command line option
--get-all, 760
--help, 759
--no-get-all, 760
--page, 759
--per-page, 760
--project-id, 759
--sudo, 759
-h, 759
gitlab-project-remote-mirror-sync command line option
--help, 760
--id, 760
--project-id, 760
--sudo, 760
-h, 760
gitlab-project-remote-mirror-update command line option
--enabled, 760
--help, 760
--id, 760
--only-protected-branches, 760
--project-id, 760
--sudo, 760
-h, 760
gitlab-project-repository-archive command line option
--format, 584
--help, 584
--id, 584
--sha, 584
-h, 584
gitlab-project-repository-blob command line option
--help, 584
--id, 584
--sha, 584
-h, 584
gitlab-project-repository-compare command line option
--from-, 584
--help, 584
--id, 584
--to, 584
-h, 584
gitlab-project-repository-contributors command line option
--help, 584
--id, 584
-h, 584
gitlab-project-repository-merge-base command line option
--help, 585
--id, 585
--refs, 585
-h, 585
gitlab-project-repository-raw-blob command line option
--help, 585
--id, 585
--sha, 585
-h, 585
gitlab-project-repository-tree command line option
--help, 585
--id, 585
--path, 585
--recursive, 585
--ref, 585
-h, 585
gitlab-project-resource-group command line option
--help, 760
-h, 760
gitlab-project-resource-group-get command line option
--help, 761
--key, 761
--project-id, 761
--sudo, 761
-h, 761
gitlab-project-resource-group-list command line option
--get-all, 761
--help, 761
--include-html-description, 761
--no-get-all, 761
--order-by, 761
--page, 761
--per-page, 761
--project-id, 761
--sort, 761
--sudo, 761
-h, 761
gitlab-project-resource-group-upcoming-job command line option
--help, 762
-h, 762
gitlab-project-resource-group-upcoming-job-list command line option
--get-all, 762
--help, 762
--no-get-all, 762
--page, 762

--per-page, 762
--project-id, 762
--resource-group-key, 762
--sudo, 762
-h, 762
gitlab-project-resource-group-update command line option
--help, 761
--key, 762
--process-mode, 762
--project-id, 762
--sudo, 761
-h, 761
gitlab-project-restore command line option
--help, 585
--id, 585
-h, 585
gitlab-project-runner command line option
--help, 762
-h, 762
gitlab-project-runner-create command line option
--help, 763
--project-id, 763
--runner-id, 763
--sudo, 763
-h, 763
gitlab-project-runner-delete command line option
--help, 763
--id, 763
--project-id, 763
--sudo, 763
-h, 763
gitlab-project-runner-list command line option
--get-all, 763
--help, 763
--no-get-all, 763
--page, 763
--per-page, 763
--project-id, 763
--scope, 763
--sudo, 763
--tag-list, 763
-h, 763
gitlab-project-search command line option
--help, 585
--id, 585
--scope, 585
--search, 586
-h, 585
gitlab-project-secure-file command line option
--help, 764
-h, 764
gitlab-project-secure-file-create command line option
--file, 764
--help, 764
--name, 764
--project-id, 764
--sudo, 764
-h, 764
gitlab-project-secure-file-delete command line option
--help, 764
--id, 764
--project-id, 764
--sudo, 764
-h, 764
gitlab-project-secure-file-download command line option
--help, 764
--id, 764
--project-id, 764
--sudo, 764
-h, 764
gitlab-project-secure-file-get command line option
--help, 765
--id, 765
--project-id, 765
--sudo, 765
-h, 765
gitlab-project-secure-file-list command line option
--get-all, 765
--help, 765
--no-get-all, 765
--page, 765
--per-page, 765
--project-id, 765
--sudo, 765
-h, 765
gitlab-project-service command line option
--help, 765
-h, 765
gitlab-project-service-account command line option
--help, 767
-h, 767
gitlab-project-service-account-access-token command line option
--help, 768
-h, 768
gitlab-project-service-account-access-token-create command line option

--description, 769
--expires-at, 769
--help, 769
--name, 769
--project-id, 769
--scopes, 769
--sudo, 769
--user-id, 769
-h, 769
gitlab-project-service-account-access-token-delete command line option
--help, 769
--id, 769
--project-id, 769
--sudo, 769
--user-id, 769
-h, 769
gitlab-project-service-account-access-token-list command line option
--created-after, 770
--created-before, 770
--expires-after, 770
--expires-before, 770
--get-all, 770
--help, 770
--last-used-after, 770
--last-used-before, 770
--no-get-all, 770
--page, 770
--per-page, 770
--project-id, 770
--revoked, 770
--search, 770
--sort, 770
--state, 770
--sudo, 770
--user-id, 770
-h, 770
gitlab-project-service-account-access-token-rotate command line option
--expires-at, 771
--help, 771
--id, 771
--project-id, 771
--sudo, 771
--user-id, 771
-h, 771
gitlab-project-service-account-create command line option
--email, 767
--help, 767
--name, 767
--project-id, 767
--sudo, 767
--username, 767
-h, 767
gitlab-project-service-account-delete command line option
--help, 767
--id, 767
--project-id, 767
--sudo, 767
-h, 767
gitlab-project-service-account-list command line option
--get-all, 768
--help, 768
--no-get-all, 768
--order-by, 768
--page, 768
--per-page, 768
--project-id, 768
--sort, 768
--sudo, 768
-h, 768
gitlab-project-service-account-update command line option
--email, 768
--help, 768
--id, 768
--name, 768
--project-id, 768
--sudo, 768
--username, 768
-h, 768
gitlab-project-service-available command line option
--help, 765
--project-id, 765
--slug, 765
--sudo, 765
-h, 765
gitlab-project-service-delete command line option
--help, 766
--project-id, 766
--slug, 766
--sudo, 766
-h, 766
gitlab-project-service-get command line option
--help, 766
--project-id, 766
--slug, 766
--sudo, 766
-h, 766
gitlab-project-service-list command line option

--get-all, 766
--help, 766
--no-get-all, 766
--page, 766
--per-page, 766
--project-id, 766
--sudo, 766
-h, 766
gitlab-project-service-update command line option
--help, 766
--project-id, 767
--slug, 767
--sudo, 767
-h, 766
gitlab-project-share command line option
--expires-at, 586
--group-access, 586
--group-id, 586
--help, 586
--id, 586
-h, 586
gitlab-project-snapshot command line option
--help, 586
--id, 586
--wiki, 586
-h, 586
gitlab-project-snippet command line option
--help, 771
-h, 771
gitlab-project-snippet-award-emoji command line option
--help, 773
-h, 773
gitlab-project-snippet-award-emoji-create command line option
--help, 774
--name, 774
--project-id, 774
--snippet-id, 774
--sudo, 774
-h, 774
gitlab-project-snippet-award-emoji-delete command line option
--help, 774
--id, 774
--project-id, 774
--snippet-id, 774
--sudo, 774
-h, 774
gitlab-project-snippet-award-emoji-get command line option
--help, 774
--id, 774
--project-id, 774
--snippet-id, 774
--sudo, 774
-h, 774
gitlab-project-snippet-award-emoji-list command line option
--get-all, 775
--help, 775
--no-get-all, 775
--page, 775
--per-page, 775
--project-id, 775
--snippet-id, 775
--sudo, 775
-h, 775
gitlab-project-snippet-content command line option
--help, 771
--id, 771
--project-id, 771
--sudo, 771
-h, 771
gitlab-project-snippet-create command line option
--content, 772
--description, 772
--file-name, 772
--files, 772
--help, 771
--project-id, 771
--sudo, 771
--title, 771
--visibility, 772
-h, 771
gitlab-project-snippet-delete command line option
--help, 772
--id, 772
--project-id, 772
--sudo, 772
-h, 772
gitlab-project-snippet-discussion command line option
--help, 775
-h, 775
gitlab-project-snippet-discussion-create command line option
--body, 775
--created-at, 775
--help, 775
--project-id, 775
--snippet-id, 775
--sudo, 775
-h, 775

gitlab-project-snippet-discussion-get command line option
--help, 776
--id, 776
--project-id, 776
--snippet-id, 776
--sudo, 776
-h, 776
gitlab-project-snippet-discussion-list command line option
--get-all, 776
--help, 776
--no-get-all, 776
--page, 776
--per-page, 776
--project-id, 776
--snippet-id, 776
--sudo, 776
-h, 776
gitlab-project-snippet-discussion-note command line option
--help, 776
-h, 776
gitlab-project-snippet-discussion-note-create command line option
--body, 777
--created-at, 777
--discussion-id, 777
--help, 777
--project-id, 777
--snippet-id, 777
--sudo, 777
-h, 777
gitlab-project-snippet-discussion-note-delete command line option
--discussion-id, 777
--help, 777
--id, 777
--project-id, 777
--snippet-id, 777
--sudo, 777
-h, 777
gitlab-project-snippet-discussion-note-get command line option
--discussion-id, 778
--help, 777
--id, 778
--project-id, 778
--snippet-id, 778
--sudo, 777
-h, 777
gitlab-project-snippet-discussion-note-update command line option
--body, 778
--discussion-id, 778
--help, 778
--id, 778
--project-id, 778
--snippet-id, 778
--sudo, 778
-h, 778
gitlab-project-snippet-get command line option
--help, 772
--id, 772
--project-id, 772
--sudo, 772
-h, 772
gitlab-project-snippet-list command line option
--get-all, 772
--help, 772
--no-get-all, 773
--page, 772
--per-page, 772
--project-id, 772
--sudo, 772
-h, 772
gitlab-project-snippet-note command line option
--help, 778
-h, 778
gitlab-project-snippet-note-award-emoji command line option
--help, 780
-h, 780
gitlab-project-snippet-note-award-emoji-create command line option
--help, 780
--name, 780
--note-id, 780
--project-id, 780
--snippet-id, 780
--sudo, 780
-h, 780
gitlab-project-snippet-note-award-emoji-delete command line option
--help, 781
--id, 781
--note-id, 781
--project-id, 781
--snippet-id, 781
--sudo, 781
-h, 781
gitlab-project-snippet-note-award-emoji-get command line option
--help, 781
--id, 781

--note-id, 781
--project-id, 781
--snippet-id, 781
--sudo, 781
-h, 781
gitlab-project-snippet-note-award-emoji-list command line option
--get-all, 782
--help, 781
--no-get-all, 782
--note-id, 782
--page, 782
--per-page, 782
--project-id, 781
--snippet-id, 782
--sudo, 781
-h, 781
gitlab-project-snippet-note-create command line option
--body, 778
--help, 778
--project-id, 778
--snippet-id, 778
--sudo, 778
-h, 778
gitlab-project-snippet-note-delete command line option
--help, 779
--id, 779
--project-id, 779
--snippet-id, 779
--sudo, 779
-h, 779
gitlab-project-snippet-note-get command line option
--help, 779
--id, 779
--project-id, 779
--snippet-id, 779
--sudo, 779
-h, 779
gitlab-project-snippet-note-list command line option
--get-all, 779
--help, 779
--no-get-all, 779
--page, 779
--per-page, 779
--project-id, 779
--snippet-id, 779
--sudo, 779
-h, 779
gitlab-project-snippet-note-update command line option
--body, 780
--help, 780
--id, 780
--project-id, 780
--snippet-id, 780
--sudo, 780
-h, 780
gitlab-project-snippet-update command line option
--content, 773
--description, 773
--file-name, 773
--files, 773
--help, 773
--id, 773
--project-id, 773
--sudo, 773
--title, 773
--visibility, 773
-h, 773
gitlab-project-snippet-user-agent-detail command line option
--help, 773
--id, 773
--project-id, 773
--sudo, 773
-h, 773
gitlab-project-star command line option
--help, 586
--id, 586
-h, 586
gitlab-project-storage command line option
--help, 782
-h, 782
gitlab-project-storage-get command line option
--help, 782
--id, 782
--project-id, 782
--sudo, 782
-h, 782
gitlab-project-tag command line option
--help, 782
-h, 782
gitlab-project-tag-create command line option
--help, 782
--message, 783
--project-id, 782
--ref, 783
--sudo, 782
--tag-name, 783
-h, 782

gitlab-project-tag-delete command line option
--help, 783
--name, 783
--project-id, 783
--sudo, 783
-h, 783
gitlab-project-tag-get command line option
--help, 783
--name, 783
--project-id, 783
--sudo, 783
-h, 783
gitlab-project-tag-list command line option
--get-all, 784
--help, 783
--no-get-all, 784
--order-by, 783
--page, 783
--per-page, 784
--project-id, 783
--search, 783
--sort, 783
--sudo, 783
-h, 783
gitlab-project-transfer command line option
--help, 586
--id, 586
--to-namespace, 586
-h, 586
gitlab-project-trigger command line option
--help, 784
-h, 784
gitlab-project-trigger-create command line option
--description, 784
--help, 784
--project-id, 784
--sudo, 784
-h, 784
gitlab-project-trigger-delete command line option
--help, 784
--id, 784
--project-id, 784
--sudo, 784
-h, 784
gitlab-project-trigger-get command line option
--help, 784
--id, 785
--project-id, 784
--sudo, 784
-h, 784
gitlab-project-trigger-list command line option
--get-all, 785
--help, 785
--no-get-all, 785
--page, 785
--per-page, 785
--project-id, 785
--sudo, 785
-h, 785
gitlab-project-trigger-pipeline command line option
--help, 587
--id, 587
--ref, 587
--token, 587
-h, 587
gitlab-project-trigger-update command line option
--description, 785
--help, 785
--id, 785
--project-id, 785
--sudo, 785
-h, 785
gitlab-project-unarchive command line option
--help, 587
--id, 587
-h, 587
gitlab-project-unshare command line option
--group-id, 587
--help, 587
--id, 587
-h, 587
gitlab-project-unstar command line option
--help, 587
--id, 587
-h, 587
gitlab-project-update command line option
--allow-merge-on-skipped-pipeline, 589
--analytics-access-level, 589
--approvals-before-merge, 589
--auto-cancel-pending-pipelines, 589
--auto-devops-deploy-strategy, 589
--auto-devops-enabled, 590
--autoclose-referenced-issues, 590
--avatar, 590
--build-coverage-regex, 590
--build-git-strategy, 590
--build-timeout, 590
--builds-access-level, 590
--ci-allow-fork-pipelines-to-run-in-parent-project, 590

--ci-config-path, 590
--ci-default-git-depth, 590
--ci-forward-deployment-enabled, 590
--ci-separated-caches, 590
--container-expiration-policy-attributes, 590
--container-registry-access-level, 590
--container-registry-enabled, 590
--default-branch, 590
--description, 590
--emails-disabled, 590
--enforce-auth-checks-on-uploads, 590
--environments-access-level, 591
--external-authorization-classification-label, 590
--feature-flags-access-level, 591
--forking-access-level, 590
--help, 589
--id, 589
--import-url, 590
--infrastructure-access-level, 591
--issue-branch-template, 592
--issues-access-level, 590
--issues-enabled, 590
--issues-template, 590
--jobs-enabled, 590
--keep-latest-artifact, 590
--lfs-enabled, 590
--merge-commit-template, 590
--merge-method, 591
--merge-pipelines-enabled, 591
--merge-requests-access-level, 591
--merge-requests-enabled, 591
--merge-requests-template, 591
--merge-trains-enabled, 591
--mirror, 591
--mirror-overwrites-diverged-branches, 591
--mirror-trigger-builds, 591
--mirror-user-id, 591
--monitor-access-level, 591
--mr-default-target-self, 591
--name, 591
--only-allow-merge-if-all-discussions-are-resolved, 591
--only-allow-merge-if-all-status-checks-passed, 589
--only-allow-merge-if-pipeline-succeeds, 591
--only-mirror-protected-branches, 591
--operations-access-level, 591
--packages-enabled, 591
--pages-access-level, 591
--path, 591
--public-builds, 591
--releases-access-level, 591
--remove-source-branch-after-merge, 591
--repository-access-level, 591
--repository-storage, 591
--request-access-enabled, 592
--requirements-access-level, 591
--resolve-outdated-diff-discussions, 592
--restrict-user-defined-variables, 591
--security-and-compliance-access-level, 592
--service-desk-enabled, 592
--shared-runners-enabled, 592
--show-default-award-emojis, 592
--snippets-access-level, 592
--snippets-enabled, 592
--squash-commit-template, 592
--squash-option, 592
--sudo, 589
--suggestion-commit-message, 592
--tag-list, 592
--topics, 592
--visibility, 592
--wiki-access-level, 592
--wiki-enabled, 592
-h, 589
gitlab-project-update-submodule command line option
--branch, 592
--commit-sha, 592
--help, 592
--id, 592
--submodule, 592
-h, 592
gitlab-project-upload command line option
--filename, 593
--filepath, 593
--help, 592
--id, 593
-h, 592
gitlab-project-user command line option
--help, 785
-h, 785
gitlab-project-user-list command line option
--get-all, 786
--help, 786
--no-get-all, 786
--page, 786
--per-page, 786
--project-id, 786
--search, 786
--skip-users, 786
--sudo, 786

-h, 786
gitlab-project-variable command line option
--help, 786
-h, 786
gitlab-project-variable-create command line option
--description, 787
--environment-scope, 787
--help, 786
--key, 786
--masked, 787
--masked-and-hidden, 787
--project-id, 786
--protected, 787
--raw, 787
--sudo, 786
--value, 786
--variable-type, 787
-h, 786
gitlab-project-variable-delete command line option
--help, 787
--key, 787
--project-id, 787
--sudo, 787
-h, 787
gitlab-project-variable-get command line option
--help, 787
--key, 787
--project-id, 787
--sudo, 787
-h, 787
gitlab-project-variable-list command line option
--get-all, 788
--help, 787
--no-get-all, 788
--page, 787
--per-page, 788
--project-id, 787
--sudo, 787
-h, 787
gitlab-project-variable-update command line option
--description, 788
--environment-scope, 788
--help, 788
--key, 788
--masked, 788
--project-id, 788
--protected, 788
--raw, 788
--sudo, 788
--value, 788
--variable-type, 788
-h, 788
gitlab-project-wiki command line option
--help, 788
-h, 788
gitlab-project-wiki-create command line option
--content, 789
--format, 789
--help, 788
--project-id, 789
--sudo, 789
--title, 789
-h, 788
gitlab-project-wiki-delete command line option
--help, 789
--project-id, 789
--slug, 789
--sudo, 789
-h, 789
gitlab-project-wiki-get command line option
--help, 789
--project-id, 789
--slug, 789
--sudo, 789
-h, 789
gitlab-project-wiki-list command line option
--get-all, 790
--help, 789
--no-get-all, 790
--page, 789
--per-page, 790
--project-id, 789
--sudo, 789
--with-content, 789
-h, 789
gitlab-project-wiki-update command line option
--content, 790
--format, 790
--help, 790
--project-id, 790
--slug, 790
--sudo, 790
--title, 790
-h, 790
gitlab-project-wiki-upload command line option
--filename, 790
--filepath, 790
--help, 790

--project-id, 790
--slug, 790
--sudo, 790
-h, 790
gitlab-registry-repository command line option
--help, 790
-h, 790
gitlab-registry-repository-get command line option
--help, 791
--id, 791
--sudo, 791
-h, 791
gitlab-runner command line option
--help, 791
-h, 791
gitlab-runner-all command line option
--help, 791, 793
--id, 791
--scope, 791
-h, 791, 793
gitlab-runner-all-list command line option
--get-all, 794
--help, 793
--no-get-all, 794
--page, 794
--paused, 794
--per-page, 794
--scope, 794
--status, 794
--sudo, 794
--tag-list, 794
--type, 794
-h, 793
gitlab-runner-create command line option
--access-level, 792
--active, 791
--description, 791
--help, 791
--info, 791
--locked, 791
--maximum-timeout, 792
--run-untagged, 791
--sudo, 791
--tag-list, 791
--token, 791
-h, 791
gitlab-runner-delete command line option
--help, 792
--id, 792
--sudo, 792
-h, 792
gitlab-runner-get command line option
--help, 792
--id, 792
--sudo, 792
-h, 792
gitlab-runner-job command line option
--help, 794
-h, 794
gitlab-runner-job-list command line option
--get-all, 794
--help, 794
--no-get-all, 794
--order-by, 794
--page, 794
--per-page, 794
--runner-id, 794
--sort, 794
--status, 794
--sudo, 794
-h, 794
gitlab-runner-list command line option
--get-all, 792
--help, 792
--no-get-all, 792
--page, 792
--paused, 792
--per-page, 792
--scope, 792
--status, 792
--sudo, 792
--tag-list, 792
--type, 792
-h, 792
gitlab-runner-update command line option
--access-level, 793
--active, 793
--description, 793
--help, 793
--id, 793
--locked, 793
--maximum-timeout, 793
--run-untagged, 793
--sudo, 793
--tag-list, 793
-h, 793
gitlab-runner-verify command line option
--help, 793
--id, 793
--token, 793
-h, 793
gitlab-service-account command line option
--help, 795
-h, 795
gitlab-service-account-create command line option

--email, 795
--help, 795
--name, 795
--sudo, 795
--username, 795
-h, 795
gitlab-service-account-list command line option
--get-all, 795
--help, 795
--no-get-all, 795
--order-by, 795
--page, 795
--per-page, 795
--sort, 795
--sudo, 795
-h, 795
gitlab-service-account-update command line option
--email, 796
--help, 796
--id, 796
--name, 796
--sudo, 796
--username, 796
-h, 796
gitlab-shared-project command line option
--help, 796
-h, 796
gitlab-shared-project-list command line option
--archived, 796
--get-all, 797
--group-id, 796
--help, 796
--min-access-level, 797
--no-get-all, 797
--order-by, 796
--page, 797
--per-page, 797
--search, 796
--simple, 797
--sort, 796
--starred, 797
--sudo, 796
--visibility, 796
--with-custom-attributes, 797
--with-issues-enabled, 797
--with-merge-requests-enabled, 797
-h, 796
gitlab-snippet command line option
--help, 797
-h, 797
gitlab-snippet-content command line option
--help, 797
--id, 797
-h, 797
gitlab-snippet-create command line option
--content, 798
--description, 797
--file-name, 798
--files, 798
--help, 797
--sudo, 797
--title, 797
--visibility, 798
-h, 797
gitlab-snippet-delete command line option
--help, 798
--id, 798
--sudo, 798
-h, 798
gitlab-snippet-get command line option
--help, 798
--id, 798
--sudo, 798
-h, 798
gitlab-snippet-list command line option
--get-all, 798
--help, 798
--no-get-all, 798
--page, 798
--per-page, 798
--sudo, 798
-h, 798
gitlab-snippet-list-all command line option
--help, 799
--id, 799
-h, 799
gitlab-snippet-list-public command line option
--help, 799
--id, 799
-h, 799
gitlab-snippet-update command line option
--content, 799
--description, 799
--file-name, 799
--files, 799
--help, 799
--id, 799
--sudo, 799
--title, 799
--visibility, 799
-h, 799
gitlab-snippet-user-agent-detail command line option
--help, 799

--id, 799
-h, 799
gitlab-starred-project command line option
--help, 800
-h, 800
gitlab-starred-project-list command line option
--archived, 800
--get-all, 801
--help, 800
--membership, 800
--min-access-level, 800
--no-get-all, 801
--order-by, 800
--owned, 800
--page, 801
--per-page, 801
--search, 800
--simple, 800
--sort, 800
--starred, 800
--statistics, 800
--sudo, 800
--user-id, 800
--visibility, 800
--with-custom-attributes, 800
--with-issues-enabled, 800
--with-merge-requests-enabled, 800
-h, 800
gitlab-todo command line option
--help, 801
-h, 801
gitlab-todo-delete command line option
--help, 801
--id, 801
--sudo, 801
-h, 801
gitlab-todo-list command line option
--action, 801
--author-id, 801
--get-all, 801
--help, 801
--no-get-all, 802
--page, 801
--per-page, 801
--project-id, 801
--state, 801
--sudo, 801
--type, 801
-h, 801
gitlab-todo-mark-all-as-done command line option
--help, 802
--id, 802
-h, 802
gitlab-todo-mark-as-done command line option
--help, 802
--id, 802
-h, 802
gitlab-topic command line option
--help, 802
-h, 802
gitlab-topic-create command line option
--avatar, 802
--description, 802
--help, 802
--name, 802
--sudo, 802
--title, 802
-h, 802
gitlab-topic-delete command line option
--help, 802
--id, 803
--sudo, 803
-h, 802
gitlab-topic-get command line option
--help, 803
--id, 803
--sudo, 803
-h, 803
gitlab-topic-list command line option
--get-all, 803
--help, 803
--no-get-all, 803
--page, 803
--per-page, 803
--sudo, 803
-h, 803
gitlab-topic-merge command line option
--help, 803
--id, 803
--source-topic-id, 803
--target-topic-id, 803
-h, 803
gitlab-topic-update command line option
--avatar, 804
--description, 804
--help, 804
--id, 804
--name, 804
--sudo, 804
-h, 804
gitlab-user command line option
--help, 804
-h, 804
gitlab-user-activate command line option
--help, 804

--id, 804
-h, 804
gitlab-user-activities command line option
--help, 810
-h, 810
gitlab-user-activities-list command line option
--get-all, 810
--help, 810
--no-get-all, 810
--page, 810
--per-page, 810
--sudo, 810
-h, 810
gitlab-user-approve command line option
--help, 804
--id, 804
-h, 804
gitlab-user-ban command line option
--help, 804
--id, 804
-h, 804
gitlab-user-block command line option
--help, 805
--id, 805
-h, 805
gitlab-user-contributed-project command line option
--help, 811
-h, 811
gitlab-user-contributed-project-list command line option
--get-all, 811
--help, 811
--no-get-all, 811
--page, 811
--per-page, 811
--sudo, 811
--user-id, 811
-h, 811
gitlab-user-create command line option
--admin, 805
--avatar, 806
--bio, 805
--can-create-group, 806
--color-scheme-id, 806
--email, 805
--extern-uid, 805
--external, 806
--help, 805
--linkedin, 805
--location, 806
--name, 805
--organization, 806
--password, 805
--private-profile, 806
--projects-limit, 805
--provider, 805
--public-email, 806
--reset-password, 805
--skip-confirmation, 806
--skype, 805
--sudo, 805
--theme-id, 806
--twitter, 805
--username, 805
--website-url, 806
-h, 805
gitlab-user-custom-attribute command line option
--help, 811
-h, 811
gitlab-user-custom-attribute-delete command line option
--help, 811
--key, 811
--sudo, 811
--user-id, 811
-h, 811
gitlab-user-custom-attribute-get command line option
--help, 812
--key, 812
--sudo, 812
--user-id, 812
-h, 812
gitlab-user-custom-attribute-list command line option
--get-all, 812
--help, 812
--no-get-all, 812
--page, 812
--per-page, 812
--sudo, 812
--user-id, 812
-h, 812
gitlab-user-deactivate command line option
--help, 806
--id, 806
-h, 806
gitlab-user-delete command line option
--help, 806
--id, 806
--sudo, 806
-h, 806
gitlab-user-email command line option
--help, 812
-h, 812

gitlab-user-email-create command line option
--email, 813
--help, 812
--skip-confirmation, 813
--sudo, 812
--user-id, 812
-h, 812
gitlab-user-email-delete command line option
--help, 813
--id, 813
--sudo, 813
--user-id, 813
-h, 813
gitlab-user-email-get command line option
--help, 813
--id, 813
--sudo, 813
--user-id, 813
-h, 813
gitlab-user-email-list command line option
--get-all, 813
--help, 813
--no-get-all, 813
--page, 813
--per-page, 813
--sudo, 813
--user-id, 813
-h, 813
gitlab-user-event command line option
--help, 814
-h, 814
gitlab-user-event-list command line option
--action, 814
--after, 814
--before, 814
--get-all, 814
--help, 814
--no-get-all, 814
--page, 814
--per-page, 814
--scope, 814
--sort, 814
--sudo, 814
--target-type, 814
--user-id, 814
-h, 814
gitlab-user-follow command line option
--help, 806
--id, 806
-h, 806
gitlab-user-get command line option
--help, 807
--id, 807
--sudo, 807
-h, 807
gitlab-user-gpg-key command line option
--help, 814
-h, 814
gitlab-user-gpg-key-create command line option
--help, 814
--key, 815
--sudo, 815
--user-id, 815
-h, 814
gitlab-user-gpg-key-delete command line option
--help, 815
--id, 815
--sudo, 815
--user-id, 815
-h, 815
gitlab-user-gpg-key-get command line option
--help, 815
--id, 815
--sudo, 815
--user-id, 815
-h, 815
gitlab-user-gpg-key-list command line option
--get-all, 815
--help, 815
--no-get-all, 815
--page, 815
--per-page, 815
--sudo, 815
--user-id, 815
-h, 815
gitlab-user-impersonation-token command line option
--help, 816
-h, 816
gitlab-user-impersonation-token-create command line option
--expires-at, 816
--help, 816
--name, 816
--scopes, 816
--sudo, 816
--user-id, 816
-h, 816
gitlab-user-impersonation-token-delete command line option
--help, 816
--id, 816
--sudo, 816

--user-id, 816
-h, 816
gitlab-user-impersonation-token-get command line option
--help, 816
--id, 816
--sudo, 816
--user-id, 816
-h, 816
gitlab-user-impersonation-token-list command line option
--get-all, 817
--help, 817
--no-get-all, 817
--page, 817
--per-page, 817
--state, 817
--sudo, 817
--user-id, 817
-h, 817
gitlab-user-key command line option
--help, 817
-h, 817
gitlab-user-key-create command line option
--help, 817
--key, 817
--sudo, 817
--title, 817
--user-id, 817
-h, 817
gitlab-user-key-delete command line option
--help, 817
--id, 818
--sudo, 817
--user-id, 818
-h, 817
gitlab-user-key-get command line option
--help, 818
--id, 818
--sudo, 818
--user-id, 818
-h, 818
gitlab-user-key-list command line option
--get-all, 818
--help, 818
--no-get-all, 818
--page, 818
--per-page, 818
--sudo, 818
--user-id, 818
-h, 818
gitlab-user-list command line option
--active, 807
--admins, 808
--auditors, 808
--blocked, 807
--created-after, 807
--created-before, 807
--custom-attributes, 808
--exclude-active, 807
--exclude-external, 808
--exclude-humans, 808
--exclude-internal, 808
--extern-uid, 808
--external, 807
--get-all, 808
--help, 807
--humans, 807
--no-get-all, 808
--page, 808
--per-page, 808
--provider, 808
--public-email, 807
--search, 807
--skip-ldap, 808
--status, 808
--sudo, 807
--two-factor, 808
--username, 807
--without-project-bots, 808
--without-projects, 808
-h, 807
gitlab-user-membership command line option
--help, 818
-h, 818
gitlab-user-membership-get command line option
--help, 818
--source-id, 819
--sudo, 818
--user-id, 819
-h, 818
gitlab-user-membership-list command line option
--get-all, 819
--help, 819
--no-get-all, 819
--page, 819
--per-page, 819
--sudo, 819
--type, 819
--user-id, 819
-h, 819
gitlab-user-personal-access-token command line option
--help, 819
-h, 819

gitlab-user-personal-access-token-create command line option
--expires-at, 819
--help, 819
--name, 819
--scopes, 819
--sudo, 819
--user-id, 819
-h, 819
gitlab-user-project command line option
--help, 820
-h, 820
gitlab-user-project-create command line option
--builds-enabled, 820
--default-branch, 820
--description, 820
--help, 820
--import-url, 821
--issues-enabled, 820
--merge-requests-enabled, 820
--name, 820
--only-allow-merge-if-build-succeeds, 821
--public, 820
--public-builds, 820
--snippets-enabled, 820
--squash-option, 820
--sudo, 820
--user-id, 820
--visibility, 820
--wall-enabled, 820
--wiki-enabled, 820
-h, 820
gitlab-user-project-list command line option
--archived, 821
--get-all, 822
--help, 821
--id-after, 822
--id-before, 822
--membership, 821
--min-access-level, 822
--no-get-all, 822
--order-by, 821
--owned, 821
--page, 822
--per-page, 822
--repository-checksum-failed, 822
--search, 821
--simple, 821
--sort, 821
--starred, 821
--statistics, 821
--sudo, 821
--user-id, 821
--visibility, 821
--wiki-checksum-failed, 822
--with-custom-attributes, 821
--with-issues-enabled, 821
--with-merge-requests-enabled, 821
--with-programming-language, 822
-h, 821
gitlab-user-reject command line option
--help, 808
--id, 808
-h, 808
gitlab-user-status command line option
--help, 822
-h, 822
gitlab-user-status-get command line option
--help, 822
--sudo, 822
--user-id, 822
-h, 822
gitlab-user-unban command line option
--help, 808
--id, 808
-h, 808
gitlab-user-unblock command line option
--help, 809
--id, 809
-h, 809
gitlab-user-unfollow command line option
--help, 809
--id, 809
-h, 809
gitlab-user-update command line option
--admin, 810
--avatar, 810
--bio, 810
--can-create-group, 810
--color-scheme-id, 810
--email, 809
--extern-uid, 810
--external, 810
--help, 809
--id, 809
--linkedin, 809
--location, 810
--name, 809
--organization, 810
--password, 809
--private-profile, 810
--projects-limit, 810
--provider, 810
--public-email, 810
--skip-reconfirmation, 810
--skype, 809

--sudo, 809
--theme-id, 810
--twitter, 809
--username, 809
--website-url, 810
-h, 809
gitlab-variable command line option
--help, 822
-h, 822
gitlab-variable-create command line option
--description, 823
--help, 822
--key, 823
--masked, 823
--protected, 823
--raw, 823
--sudo, 823
--value, 823
--variable-type, 823
-h, 822
gitlab-variable-delete command line option
--help, 823
--key, 823
--sudo, 823
-h, 823
gitlab-variable-get command line option
--help, 823
--key, 823
--sudo, 823
-h, 823
gitlab-variable-list command line option
--get-all, 823
--help, 823
--no-get-all, 823
--page, 823
--per-page, 823
--sudo, 823
-h, 823
gitlab-variable-update command line option
--description, 824
--help, 824
--key, 824
--masked, 824
--protected, 824
--raw, 824
--sudo, 824
--value, 824
--variable-type, 824
-h, 824
GitlabActivateError, 433
GitlabAttachFileError, 433
GitlabAuthenticationError, 433
GitlabBanError, 434
GitlabBlockError, 434
GitlabBuildCancelError, 434
GitlabBuildEraseError, 434
GitlabBuildPlayError, 435
GitlabBuildRetryError, 435
GitlabCancelError, 435
GitlabCherryPickError, 435
GitlabCiLintError, 435
Gitlabciyml (*class in gitlab.v4.objects*), 226
gitlabciyml_templates (*gitlab.v4.objects.Project attribute*), 277
GitlabciymlManager (*class in gitlab.v4.objects*), 227
gitlabciymls (*gitlab.Gitlab attribute*), 200
GitlabConfigHelperError, 431
GitlabConfigMissingError, 431
GitlabConfigParser (*class in gitlab.config*), 431
GitlabConnectionError, 436
GitlabCreateError, 436
GitlabDataError, 432
GitlabDeactivateError, 436
GitlabDeleteError, 436
GitlabDeploymentApprovalError, 437
GitlabError, 437
GitlabFollowError, 437
GitlabGetError, 437
GitlabGroupTransferError, 438
GitlabHeadError, 438
GitlabHookTestError, 438
GitlabHousekeepingError, 438
GitlabHttpError, 439
GitlabIDError, 432
GitlabImportError, 439
GitlabInvitationError, 439
GitlabJobCancelError, 439
GitlabJobEraseError, 440
GitlabJobPlayError, 440
GitlabJobRetryError, 440
GitlabLicenseError, 440
GitlabList (*class in gitlab*), 207
GitlabListError, 440
GitlabMarkdownError, 442
GitlabMRApprovalError, 441
GitlabMRClosedError, 441
GitlabMRForbiddenError, 441
GitlabMROnBuildSuccessError, 441
GitlabMRRebaseError, 442
GitlabMRResetApprovalError, 442
GitlabOperationError, 442
GitlabOwnershipError, 443
GitlabParsingError, 443
GitlabPipelineCancelError, 443
GitlabPipelinePlayError, 443
GitlabPipelineRetryError, 444
GitlabProjectDeployKeyError, 444
GitlabPromoteError, 444

GitlabProtectError, 444
GitlabRenderError, 445
GitlabRepairError, 445
GitlabRestoreError, 445
GitlabRetryError, 445
GitlabRevertError, 446
GitlabRotateError, 446
GitlabSearchError, 446
GitlabSetError, 446
GitlabStopError, 446
GitlabSubscribeError, 447
GitlabTimeTrackingError, 447
GitlabTodoError, 447
GitlabTopicMergeError, 447
GitlabTransferProjectError, 448
GitlabUnbanError, 448
GitlabUnblockError, 448
GitlabUnfollowError, 448
GitlabUnsubscribeError, 449
GitlabUpdateError, 449
GitlabUploadError, 449
GitlabUserApproveError, 449
GitlabUserRejectError, 450
GitlabVerifyError, 450
GLOBAL (*gitlab.const.NotificationLevel attribute*), 432
GLOBAL_SNIPPET_TITLES (*gitlab.const.SearchScope attribute*), 432
gpgkeys (*gitlab.v4.objects.CurrentUser attribute*), 217
gpgkeys (*gitlab.v4.objects.User attribute*), 415
Group (*class in gitlab.v4.objects*), 227
GroupAccessRequest (*class in gitlab.v4.objects*), 230
GroupAccessRequestManager (*class in gitlab.v4.objects*), 231
GroupAccessToken (*class in gitlab.v4.objects*), 231
GroupAccessTokenManager (*class in gitlab.v4.objects*), 231
GroupApprovalRule (*class in gitlab.v4.objects*), 231
GroupApprovalRuleManager (*class in gitlab.v4.objects*), 232
GroupAuditEvent (*class in gitlab.v4.objects*), 232
GroupAuditEventManager (*class in gitlab.v4.objects*), 232
GroupBadge (*class in gitlab.v4.objects*), 233
GroupBadgeManager (*class in gitlab.v4.objects*), 233
GroupBillableMember (*class in gitlab.v4.objects*), 233
GroupBillableMemberManager (*class in gitlab.v4.objects*), 233
GroupBillableMemberMembership (*class in gitlab.v4.objects*), 234
GroupBillableMemberMembershipManager (*class in gitlab.v4.objects*), 234
GroupBoard (*class in gitlab.v4.objects*), 234
GroupBoardList (*class in gitlab.v4.objects*), 234
GroupBoardListManager (*class in gitlab.v4.objects*), 235
GroupBoardManager (*class in gitlab.v4.objects*), 235
GroupCluster (*class in gitlab.v4.objects*), 235
GroupClusterManager (*class in gitlab.v4.objects*), 235
GroupCustomAttribute (*class in gitlab.v4.objects*), 236
GroupCustomAttributeManager (*class in gitlab.v4.objects*), 237
GroupDeployToken (*class in gitlab.v4.objects*), 237
GroupDeployTokenManager (*class in gitlab.v4.objects*), 237
GroupDescendantGroup (*class in gitlab.v4.objects*), 238
GroupDescendantGroupManager (*class in gitlab.v4.objects*), 238
GroupEpic (*class in gitlab.v4.objects*), 238
GroupEpicAwardEmoji (*class in gitlab.v4.objects*), 239
GroupEpicAwardEmojiManager (*class in gitlab.v4.objects*), 239
GroupEpicDiscussionNote (*class in gitlab.v4.objects*), 239
GroupEpicDiscussionNoteManager (*class in gitlab.v4.objects*), 239
GroupEpicIssue (*class in gitlab.v4.objects*), 240
GroupEpicIssueManager (*class in gitlab.v4.objects*), 240
GroupEpicManager (*class in gitlab.v4.objects*), 241
GroupEpicNote (*class in gitlab.v4.objects*), 242
GroupEpicNoteAwardEmoji (*class in gitlab.v4.objects*), 242
GroupEpicNoteAwardEmojiManager (*class in gitlab.v4.objects*), 242
GroupEpicNoteManager (*class in gitlab.v4.objects*), 242
GroupEpicResourceLabelEvent (*class in gitlab.v4.objects*), 243
GroupEpicResourceLabelEventManager (*class in gitlab.v4.objects*), 243
GroupExport (*class in gitlab.v4.objects*), 243
GroupExportManager (*class in gitlab.v4.objects*), 243
GroupHook (*class in gitlab.v4.objects*), 244
GroupHookManager (*class in gitlab.v4.objects*), 244
GroupImport (*class in gitlab.v4.objects*), 245
GroupImportManager (*class in gitlab.v4.objects*), 245
GroupInvitation (*class in gitlab.v4.objects*), 246
GroupInvitationManager (*class in gitlab.v4.objects*), 246
GroupIssue (*class in gitlab.v4.objects*), 246
GroupIssueManager (*class in gitlab.v4.objects*), 247
GroupIssuesStatistics (*class in gitlab.v4.objects*), 247
GroupIssuesStatisticsManager (*class in gitlab.v4.objects*), 247

GroupIteration (*class in gitlab.v4.objects*), 248
GroupIterationManager (*class in gitlab.v4.objects*), 248
GroupLabel (*class in gitlab.v4.objects*), 249
GroupLabelManager (*class in gitlab.v4.objects*), 250
GroupLDAPGroupLink (*class in gitlab.v4.objects*), 248
GroupLDAPGroupLinkManager (*class in gitlab.v4.objects*), 249
GroupManager (*class in gitlab.v4.objects*), 250
GroupMember (*class in gitlab.v4.objects*), 253
GroupMemberAll (*class in gitlab.v4.objects*), 253
GroupMemberAllManager (*class in gitlab.v4.objects*), 253
GroupMemberManager (*class in gitlab.v4.objects*), 253
GroupMemberRole (*class in gitlab.v4.objects*), 254
GroupMemberRoleManager (*class in gitlab.v4.objects*), 254
GroupMergeRequest (*class in gitlab.v4.objects*), 255
GroupMergeRequestManager (*class in gitlab.v4.objects*), 255
GroupMilestone (*class in gitlab.v4.objects*), 256
GroupMilestoneManager (*class in gitlab.v4.objects*), 257
GroupNotificationSettings (*class in gitlab.v4.objects*), 258
GroupNotificationSettingsManager (*class in gitlab.v4.objects*), 258
GroupPackage (*class in gitlab.v4.objects*), 258
GroupPackageManager (*class in gitlab.v4.objects*), 259
GroupProject (*class in gitlab.v4.objects*), 259
GroupProjectManager (*class in gitlab.v4.objects*), 259
GroupPushRules (*class in gitlab.v4.objects*), 260
GroupPushRulesManager (*class in gitlab.v4.objects*), 260
GroupRegistryRepositoryManager (*class in gitlab.v4.objects*), 261
GroupRunner (*class in gitlab.v4.objects*), 261
GroupRunnerManager (*class in gitlab.v4.objects*), 261
groups (*gitlab.Gitlab attribute*), 200
groups (*gitlab.v4.objects.Project attribute*), 277
groups_allowlist (*gitlab.v4.objects.ProjectJobTokenScope attribute*), 332
GroupSAMLGroupLink (*class in gitlab.v4.objects*), 262
GroupSAMLGroupLinkManager (*class in gitlab.v4.objects*), 262
GroupServiceAccount (*class in gitlab.v4.objects*), 262
GroupServiceAccountAccessToken (*class in gitlab.v4.objects*), 262
GroupServiceAccountAccessTokenManager (*class in gitlab.v4.objects*), 263
GroupServiceAccountManager (*class in gitlab.v4.objects*), 263
GroupSubgroup (*class in gitlab.v4.objects*), 264
GroupSubgroupManager (*class in gitlab.v4.objects*), 264
GroupVariable (*class in gitlab.v4.objects*), 265
GroupVariableManager (*class in gitlab.v4.objects*), 265
GroupWiki (*class in gitlab.v4.objects*), 266
GroupWikiManager (*class in gitlab.v4.objects*), 266
GUEST (*gitlab.const.AccessLevel attribute*), 432

H

handle_retry() (*gitlab.utils.Retry method*), 461
handle_retry_on_status() (*gitlab.utils.Retry method*), 461
head() (*gitlab.v4.objects.ProjectFileManager method*), 312
headers (*gitlab.Gitlab attribute*), 200
help (*gitlab.cli.CustomAction attribute*), 430
Hook (*class in gitlab.v4.objects*), 266
HookManager (*class in gitlab.v4.objects*), 266
hooks (*gitlab.Gitlab attribute*), 200
hooks (*gitlab.v4.objects.Group attribute*), 228
hooks (*gitlab.v4.objects.Project attribute*), 277
housekeeping() (*gitlab.v4.objects.Project method*), 277
http_delete() (*gitlab.Gitlab method*), 200
http_get() (*gitlab.Gitlab method*), 201
http_head() (*gitlab.Gitlab method*), 201
http_list() (*gitlab.Gitlab method*), 201
http_patch() (*gitlab.Gitlab method*), 202
http_post() (*gitlab.Gitlab method*), 202
http_put() (*gitlab.Gitlab method*), 203
http_request() (*gitlab.Gitlab method*), 203

I

identityproviders (*gitlab.v4.objects.User attribute*), 415
impersonationtokens (*gitlab.v4.objects.User attribute*), 415
import_bitbucket_server() (*gitlab.v4.objects.ProjectManager method*), 340
import_github() (*gitlab.v4.objects.ProjectManager method*), 341
import_group() (*gitlab.v4.objects.GroupManager method*), 252
import_project() (*gitlab.v4.objects.ProjectManager method*), 342
imports (*gitlab.v4.objects.Group attribute*), 228
imports (*gitlab.v4.objects.Project attribute*), 277
in_object (*gitlab.cli.CustomAction attribute*), 430
integrations (*gitlab.v4.objects.Project attribute*), 277
INTERNAL (*gitlab.const.Visibility attribute*), 433
invitations (*gitlab.v4.objects.Group attribute*), 228
invitations (*gitlab.v4.objects.Project attribute*), 277

Issue (*class in gitlab.v4.objects*), 267
issue_templates (*gitlab.v4.objects.Project attribute*), 277
IssueManager (*class in gitlab.v4.objects*), 267
ISSUES (*gitlab.const.SearchScope attribute*), 432
issues (*gitlab.Gitlab attribute*), 204
issues (*gitlab.v4.objects.Group attribute*), 228
issues (*gitlab.v4.objects.GroupEpic attribute*), 238
issues (*gitlab.v4.objects.Project attribute*), 277
issues() (*gitlab.v4.objects.GroupMilestone method*), 256
issues() (*gitlab.v4.objects.ProjectMilestone method*), 362
issues_statistics (*gitlab.Gitlab attribute*), 204
issues_statistics (*gitlab.v4.objects.Group attribute*), 228
issues_statistics (*gitlab.v4.objects.Project attribute*), 277
IssuesStatistics (*class in gitlab.v4.objects*), 268
IssuesStatisticsManager (*class in gitlab.v4.objects*), 268
iterations (*gitlab.v4.objects.Group attribute*), 228
iterations (*gitlab.v4.objects.Project attribute*), 278

J

job_stats() (*gitlab.v4.objects.SidekiqManager method*), 406
job_token_scope (*gitlab.v4.objects.Project attribute*), 278
jobs (*gitlab.v4.objects.Project attribute*), 278
jobs (*gitlab.v4.objects.ProjectPipeline attribute*), 368
jobs (*gitlab.v4.objects.Runner attribute*), 401

K

keep_artifacts() (*gitlab.v4.objects.ProjectJob method*), 330
Key (*class in gitlab.v4.objects*), 268
KeyManager (*class in gitlab.v4.objects*), 268
keys (*gitlab.Gitlab attribute*), 204
keys (*gitlab.v4.objects.CurrentUser attribute*), 217
keys (*gitlab.v4.objects.Project attribute*), 278
keys (*gitlab.v4.objects.User attribute*), 415

L

labels (*gitlab.v4.objects.Group attribute*), 228
labels (*gitlab.v4.objects.Project attribute*), 278
languages() (*gitlab.v4.objects.Project method*), 278
latest() (*gitlab.v4.objects.ProjectPipelineManager method*), 371
ldap_group_links (*gitlab.v4.objects.Group attribute*), 228
ldap_sync() (*gitlab.v4.objects.Group method*), 228
LDAPGroup (*class in gitlab.v4.objects*), 269
LDAPGroupManager (*class in gitlab.v4.objects*), 269
ldapgroups (*gitlab.Gitlab attribute*), 204
License (*class in gitlab.v4.objects*), 270
license_templates (*gitlab.v4.objects.Project attribute*), 278
LicenseManager (*class in gitlab.v4.objects*), 270
licenses (*gitlab.Gitlab attribute*), 204
links (*gitlab.v4.objects.ProjectIssue attribute*), 319
links (*gitlab.v4.objects.ProjectRelease attribute*), 382
list() (*gitlab.mixins.ListMixin method*), 454
list() (*gitlab.v4.objects.LDAPGroupManager method*), 269
list() (*gitlab.v4.objects.UserProjectManager method*), 425
list_all() (*gitlab.v4.objects.SnippetManager method*), 408
list_public() (*gitlab.v4.objects.SnippetManager method*), 409
ListMixin (*class in gitlab.mixins*), 454
lists (*gitlab.v4.objects.GroupBoard attribute*), 234
lists (*gitlab.v4.objects.ProjectBoard attribute*), 289

M

main() (*in module gitlab.cli*), 431
MAINTAINER (*gitlab.const.AccessLevel attribute*), 432
manager (*gitlab.base.RESTObject attribute*), 429
manager (*gitlab.mixins.AccessRequestMixin attribute*), 451
manager (*gitlab.mixins.DownloadMixin attribute*), 453
manager (*gitlab.mixins.ObjectDeleteMixin attribute*), 455
manager (*gitlab.mixins.ParticipantsMixin attribute*), 455
manager (*gitlab.mixins.RefreshMixin attribute*), 455
manager (*gitlab.mixins.SaveMixin attribute*), 456
manager (*gitlab.mixins.SubscribableMixin attribute*), 457
manager (*gitlab.mixins.TimeTrackingMixin attribute*), 458
manager (*gitlab.mixins.TodoMixin attribute*), 459
manager (*gitlab.mixins.UserAgentDetailMixin attribute*), 459
manager (*gitlab.v4.objects.GroupEpicIssue attribute*), 240
manager (*gitlab.v4.objects.GroupLabel attribute*), 249
manager (*gitlab.v4.objects.ProjectFeatureFlag attribute*), 306
manager (*gitlab.v4.objects.ProjectFile attribute*), 309
manager (*gitlab.v4.objects.ProjectLabel attribute*), 334
mark_all_as_done() (*gitlab.v4.objects.TodoManager method*), 411
mark_as_done() (*gitlab.v4.objects.Todo method*), 411
markdown() (*gitlab.Gitlab method*), 204
MaskingFormatter (*class in gitlab.utils*), 460
member_roles (*gitlab.Gitlab attribute*), 205

member_roles (*gitlab.v4.objects.Group attribute*), 228
MemberRole (*class in gitlab.v4.objects*), 270
MemberRoleManager (*class in gitlab.v4.objects*), 270
members (*gitlab.v4.objects.Group attribute*), 228
members (*gitlab.v4.objects.Project attribute*), 278
members_all (*gitlab.v4.objects.Group attribute*), 228
members_all (*gitlab.v4.objects.Project attribute*), 278
memberships (*gitlab.v4.objects.GroupBillableMember attribute*), 233
memberships (*gitlab.v4.objects.User attribute*), 415
MENTION (*gitlab.const.NotificationLevel attribute*), 432
merge() (*gitlab.v4.objects.ProjectMergeRequest method*), 347
merge() (*gitlab.v4.objects.TopicManager method*), 412
merge_config() (*gitlab.Gitlab class method*), 205
merge_ref() (*gitlab.v4.objects.ProjectMergeRequest method*), 348
merge_request_templates (*gitlab.v4.objects.Project attribute*), 278
MERGE_REQUESTS (*gitlab.const.SearchScope attribute*), 432
merge_requests() (*gitlab.v4.objects.GroupMilestone method*), 257
merge_requests() (*gitlab.v4.objects.ProjectCommit method*), 294
merge_requests() (*gitlab.v4.objects.ProjectMilestone method*), 362
merge_trains (*gitlab.v4.objects.Project attribute*), 278
MergeRequest (*class in gitlab.v4.objects*), 271
MergeRequestManager (*class in gitlab.v4.objects*), 272
mergerequests (*gitlab.Gitlab attribute*), 205
mergerequests (*gitlab.v4.objects.Group attribute*), 228
mergerequests (*gitlab.v4.objects.Project attribute*), 278
mergerequests (*gitlab.v4.objects.ProjectDeployment attribute*), 301
message (*gitlab.utils.WarnMessageData attribute*), 461
MILESTONES (*gitlab.const.SearchScope attribute*), 433
milestones (*gitlab.v4.objects.Group attribute*), 228
milestones (*gitlab.v4.objects.Project attribute*), 278
MINIMAL_ACCESS (*gitlab.const.AccessLevel attribute*), 432
mirror_pull() (*gitlab.v4.objects.Project method*), 278
mirror_pull_details() (*gitlab.v4.objects.Project method*), 278
module
    gitlab.base, 428
    gitlab.cli, 430
    gitlab.config, 431
    gitlab.const, 432
    gitlab.exceptions, 433
    gitlab.mixins, 450
    gitlab.utils, 460
    gitlab.v4, 428
    gitlab.v4.objects, 208
move() (*gitlab.v4.objects.ProjectIssue method*), 319

N

Namespace (*class in gitlab.v4.objects*), 273
NamespaceManager (*class in gitlab.v4.objects*), 273
namespaces (*gitlab.Gitlab attribute*), 205
next() (*gitlab.base.RESTObjectList method*), 429
next() (*gitlab.GitlabList method*), 208
next_page (*gitlab.base.RESTObjectList property*), 429
next_page (*gitlab.GitlabList property*), 208
NO_ACCESS (*gitlab.const.AccessLevel attribute*), 432
notes (*gitlab.v4.objects.GroupEpic attribute*), 239
notes (*gitlab.v4.objects.Project attribute*), 279
notes (*gitlab.v4.objects.ProjectCommitDiscussion attribute*), 296
notes (*gitlab.v4.objects.ProjectIssue attribute*), 319
notes (*gitlab.v4.objects.ProjectIssueDiscussion attribute*), 320
notes (*gitlab.v4.objects.ProjectMergeRequest attribute*), 348
notes (*gitlab.v4.objects.ProjectMergeRequestDiscussion attribute*), 353
notes (*gitlab.v4.objects.ProjectSnippet attribute*), 390
notes (*gitlab.v4.objects.ProjectSnippetDiscussion attribute*), 391
NotificationLevel (*class in gitlab.const*), 432
NotificationSettings (*class in gitlab.v4.objects*), 273
notificationsettings (*gitlab.Gitlab attribute*), 205
notificationsettings (*gitlab.v4.objects.Group attribute*), 228
notificationsettings (*gitlab.v4.objects.Project attribute*), 279
NotificationSettingsManager (*class in gitlab.v4.objects*), 274
NoUpdateMixin (*class in gitlab.mixins*), 454

O

ObjectDeleteMixin (*class in gitlab.mixins*), 454
optional (*gitlab.cli.CustomAction attribute*), 430
overload() (*in module gitlab.v4.objects*), 427
OWNER (*gitlab.const.AccessLevel attribute*), 432

P

package_files (*gitlab.v4.objects.ProjectPackage attribute*), 365
package_protection_rules (*gitlab.v4.objects.Project attribute*), 279
packages (*gitlab.v4.objects.Group attribute*), 228
packages (*gitlab.v4.objects.Project attribute*), 279
pages (*gitlab.v4.objects.Project attribute*), 279
PagesDomain (*class in gitlab.v4.objects*), 274
PagesDomainManager (*class in gitlab.v4.objects*), 274

pagesdomains (*gitlab.Gitlab attribute*), 205
pagesdomains (*gitlab.v4.objects.Project attribute*), 279
parent_attrs (*gitlab.base.RESTManager property*), 428
participants() (*gitlab.mixins.ParticipantsMixin method*), 455
ParticipantsMixin (*class in gitlab.mixins*), 455
PARTICIPATING (*gitlab.const.NotificationLevel attribute*), 432
path (*gitlab.base.RESTManager property*), 428
path_with_namespace (*gitlab.v4.objects.Project attribute*), 279
per_page (*gitlab.base.RESTObjectList property*), 430
per_page (*gitlab.GitlabList property*), 208
personal_access_tokens (*gitlab.Gitlab attribute*), 206
personal_access_tokens (*gitlab.v4.objects.User attribute*), 415
PersonalAccessToken (*class in gitlab.v4.objects*), 275
PersonalAccessTokenManager (*class in gitlab.v4.objects*), 275
pformat() (*gitlab.base.RESTObject method*), 429
pipelines (*gitlab.v4.objects.Project attribute*), 279
pipelines (*gitlab.v4.objects.ProjectMergeRequest attribute*), 348
pipelines (*gitlab.v4.objects.ProjectPackage attribute*), 365
pipelines (*gitlab.v4.objects.ProjectPipelineSchedule attribute*), 371
pipelineschedules (*gitlab.v4.objects.Project attribute*), 279
PLANNER (*gitlab.const.AccessLevel attribute*), 432
play() (*gitlab.v4.objects.ProjectJob method*), 331
play() (*gitlab.v4.objects.ProjectPipelineSchedule method*), 371
pprint() (*gitlab.base.RESTObject method*), 429
prev_page (*gitlab.base.RESTObjectList property*), 430
prev_page (*gitlab.GitlabList property*), 208
PRIVATE (*gitlab.const.Visibility attribute*), 433
process_metrics() (*gitlab.v4.objects.SidekiqManager method*), 407
Project (*class in gitlab.v4.objects*), 275
PROJECT_NOTES (*gitlab.const.SearchScope attribute*), 433
ProjectAccessRequest (*class in gitlab.v4.objects*), 283
ProjectAccessRequestManager (*class in gitlab.v4.objects*), 283
ProjectAccessToken (*class in gitlab.v4.objects*), 283
ProjectAccessTokenManager (*class in gitlab.v4.objects*), 283
ProjectAdditionalStatistics (*class in gitlab.v4.objects*), 284
ProjectAdditionalStatisticsManager (*class in gitlab.v4.objects*), 284
ProjectApproval (*class in gitlab.v4.objects*), 284
ProjectApprovalManager (*class in gitlab.v4.objects*), 285
ProjectApprovalRule (*class in gitlab.v4.objects*), 285
ProjectApprovalRuleManager (*class in gitlab.v4.objects*), 285
ProjectArtifact (*class in gitlab.v4.objects*), 286
ProjectArtifactManager (*class in gitlab.v4.objects*), 286
ProjectAudit (*class in gitlab.v4.objects*), 287
ProjectAuditEvent (*class in gitlab.v4.objects*), 288
ProjectAuditEventManager (*class in gitlab.v4.objects*), 288
ProjectAuditManager (*class in gitlab.v4.objects*), 288
ProjectBadge (*class in gitlab.v4.objects*), 288
ProjectBadgeManager (*class in gitlab.v4.objects*), 289
ProjectBoard (*class in gitlab.v4.objects*), 289
ProjectBoardList (*class in gitlab.v4.objects*), 289
ProjectBoardListManager (*class in gitlab.v4.objects*), 290
ProjectBoardManager (*class in gitlab.v4.objects*), 290
ProjectBranch (*class in gitlab.v4.objects*), 290
ProjectBranchManager (*class in gitlab.v4.objects*), 290
ProjectCiLint (*class in gitlab.v4.objects*), 291
ProjectCiLintManager (*class in gitlab.v4.objects*), 291
ProjectCluster (*class in gitlab.v4.objects*), 291
ProjectClusterAgent (*class in gitlab.v4.objects*), 292
ProjectClusterAgentManager (*class in gitlab.v4.objects*), 292
ProjectClusterManager (*class in gitlab.v4.objects*), 292
ProjectCommit (*class in gitlab.v4.objects*), 293
ProjectCommitComment (*class in gitlab.v4.objects*), 295
ProjectCommitCommentManager (*class in gitlab.v4.objects*), 296
ProjectCommitDiscussion (*class in gitlab.v4.objects*), 296
ProjectCommitDiscussionManager (*class in gitlab.v4.objects*), 296
ProjectCommitDiscussionNote (*class in gitlab.v4.objects*), 297
ProjectCommitDiscussionNoteManager (*class in gitlab.v4.objects*), 297
ProjectCommitManager (*class in gitlab.v4.objects*), 297
ProjectCommitStatus (*class in gitlab.v4.objects*), 298
ProjectCommitStatusManager (*class in gitlab.v4.objects*), 298
ProjectCustomAttribute (*class in gitlab.v4.objects*), 299

ProjectCustomAttributeManager (*class in gitlab.v4.objects*), 299
ProjectDeployment (*class in gitlab.v4.objects*), 300
ProjectDeploymentManager (*class in gitlab.v4.objects*), 301
ProjectDeploymentMergeRequest (*class in gitlab.v4.objects*), 301
ProjectDeploymentMergeRequestManager (*class in gitlab.v4.objects*), 302
ProjectDeployToken (*class in gitlab.v4.objects*), 299
ProjectDeployTokenManager (*class in gitlab.v4.objects*), 300
ProjectDockerfileTemplate (*class in gitlab.v4.objects*), 303
ProjectDockerfileTemplateManager (*class in gitlab.v4.objects*), 303
ProjectEnvironment (*class in gitlab.v4.objects*), 303
ProjectEnvironmentManager (*class in gitlab.v4.objects*), 303
ProjectEvent (*class in gitlab.v4.objects*), 304
ProjectEventManager (*class in gitlab.v4.objects*), 304
ProjectExport (*class in gitlab.v4.objects*), 305
ProjectExportManager (*class in gitlab.v4.objects*), 305
ProjectExternalStatusCheck (*class in gitlab.v4.objects*), 305
ProjectExternalStatusCheckManager (*class in gitlab.v4.objects*), 305
ProjectFeatureFlag (*class in gitlab.v4.objects*), 306
ProjectFeatureFlagManager (*class in gitlab.v4.objects*), 306
ProjectFeatureFlagUserList (*class in gitlab.v4.objects*), 307
ProjectFeatureFlagUserListManager (*class in gitlab.v4.objects*), 308
ProjectFile (*class in gitlab.v4.objects*), 308
ProjectFileManager (*class in gitlab.v4.objects*), 309
ProjectFork (*class in gitlab.v4.objects*), 313
ProjectForkManager (*class in gitlab.v4.objects*), 313
ProjectGitignoreTemplate (*class in gitlab.v4.objects*), 314
ProjectGitignoreTemplateManager (*class in gitlab.v4.objects*), 314
ProjectGitlabciymlTemplate (*class in gitlab.v4.objects*), 314
ProjectGitlabciymlTemplateManager (*class in gitlab.v4.objects*), 315
ProjectHook (*class in gitlab.v4.objects*), 315
ProjectHookManager (*class in gitlab.v4.objects*), 315
ProjectImport (*class in gitlab.v4.objects*), 316
ProjectImportManager (*class in gitlab.v4.objects*), 317
ProjectIntegration (*class in gitlab.v4.objects*), 317
ProjectIntegrationManager (*class in gitlab.v4.objects*), 317
ProjectInvitation (*class in gitlab.v4.objects*), 317
ProjectInvitationManager (*class in gitlab.v4.objects*), 317
ProjectIssue (*class in gitlab.v4.objects*), 318
ProjectIssueAwardEmoji (*class in gitlab.v4.objects*), 320
ProjectIssueAwardEmojiManager (*class in gitlab.v4.objects*), 320
ProjectIssueDiscussion (*class in gitlab.v4.objects*), 320
ProjectIssueDiscussionManager (*class in gitlab.v4.objects*), 320
ProjectIssueDiscussionNote (*class in gitlab.v4.objects*), 321
ProjectIssueDiscussionNoteManager (*class in gitlab.v4.objects*), 321
ProjectIssueLink (*class in gitlab.v4.objects*), 321
ProjectIssueLinkManager (*class in gitlab.v4.objects*), 322
ProjectIssueManager (*class in gitlab.v4.objects*), 322
ProjectIssueNote (*class in gitlab.v4.objects*), 324
ProjectIssueNoteAwardEmoji (*class in gitlab.v4.objects*), 324
ProjectIssueNoteAwardEmojiManager (*class in gitlab.v4.objects*), 324
ProjectIssueNoteManager (*class in gitlab.v4.objects*), 324
ProjectIssueResourceIterationEvent (*class in gitlab.v4.objects*), 325
ProjectIssueResourceIterationEventManager (*class in gitlab.v4.objects*), 325
ProjectIssueResourceLabelEvent (*class in gitlab.v4.objects*), 325
ProjectIssueResourceLabelEventManager (*class in gitlab.v4.objects*), 326
ProjectIssueResourceMilestoneEvent (*class in gitlab.v4.objects*), 326
ProjectIssueResourceMilestoneEventManager (*class in gitlab.v4.objects*), 326
ProjectIssueResourceStateEvent (*class in gitlab.v4.objects*), 326
ProjectIssueResourceStateEventManager (*class in gitlab.v4.objects*), 326
ProjectIssueResourceWeightEvent (*class in gitlab.v4.objects*), 327
ProjectIssueResourceWeightEventManager (*class in gitlab.v4.objects*), 327
ProjectIssuesStatistics (*class in gitlab.v4.objects*), 327
ProjectIssuesStatisticsManager (*class in gitlab.v4.objects*), 328
ProjectIssueTemplate (*class in gitlab.v4.objects*), 327

ProjectIssueTemplateManager (*class in gitlab.v4.objects*), 327
ProjectIterationManager (*class in gitlab.v4.objects*), 328
ProjectJob (*class in gitlab.v4.objects*), 328
ProjectJobManager (*class in gitlab.v4.objects*), 332
ProjectJobTokenScope (*class in gitlab.v4.objects*), 332
ProjectJobTokenScopeManager (*class in gitlab.v4.objects*), 332
ProjectKey (*class in gitlab.v4.objects*), 332
ProjectKeyManager (*class in gitlab.v4.objects*), 333
ProjectLabel (*class in gitlab.v4.objects*), 333
ProjectLabelManager (*class in gitlab.v4.objects*), 334
ProjectLicenseTemplate (*class in gitlab.v4.objects*), 335
ProjectLicenseTemplateManager (*class in gitlab.v4.objects*), 335
ProjectManager (*class in gitlab.v4.objects*), 335
ProjectMember (*class in gitlab.v4.objects*), 344
ProjectMemberAll (*class in gitlab.v4.objects*), 344
ProjectMemberAllManager (*class in gitlab.v4.objects*), 345
ProjectMemberManager (*class in gitlab.v4.objects*), 345
ProjectMergeRequest (*class in gitlab.v4.objects*), 345
ProjectMergeRequestApproval (*class in gitlab.v4.objects*), 349
ProjectMergeRequestApprovalManager (*class in gitlab.v4.objects*), 350
ProjectMergeRequestApprovalRule (*class in gitlab.v4.objects*), 350
ProjectMergeRequestApprovalRuleManager (*class in gitlab.v4.objects*), 351
ProjectMergeRequestApprovalState (*class in gitlab.v4.objects*), 351
ProjectMergeRequestApprovalStateManager (*class in gitlab.v4.objects*), 351
ProjectMergeRequestAwardEmoji (*class in gitlab.v4.objects*), 352
ProjectMergeRequestAwardEmojiManager (*class in gitlab.v4.objects*), 352
ProjectMergeRequestDiff (*class in gitlab.v4.objects*), 352
ProjectMergeRequestDiffManager (*class in gitlab.v4.objects*), 352
ProjectMergeRequestDiscussion (*class in gitlab.v4.objects*), 353
ProjectMergeRequestDiscussionManager (*class in gitlab.v4.objects*), 353
ProjectMergeRequestDiscussionNote (*class in gitlab.v4.objects*), 353
ProjectMergeRequestDiscussionNoteManager (*class in gitlab.v4.objects*), 354
ProjectMergeRequestDraftNote (*class in gitlab.v4.objects*), 354
ProjectMergeRequestDraftNoteManager (*class in gitlab.v4.objects*), 354
ProjectMergeRequestManager (*class in gitlab.v4.objects*), 355
ProjectMergeRequestNote (*class in gitlab.v4.objects*), 357
ProjectMergeRequestNoteAwardEmoji (*class in gitlab.v4.objects*), 357
ProjectMergeRequestNoteAwardEmojiManager (*class in gitlab.v4.objects*), 357
ProjectMergeRequestNoteManager (*class in gitlab.v4.objects*), 357
ProjectMergeRequestPipeline (*class in gitlab.v4.objects*), 358
ProjectMergeRequestPipelineManager (*class in gitlab.v4.objects*), 358
ProjectMergeRequestResourceLabelEvent (*class in gitlab.v4.objects*), 358
ProjectMergeRequestResourceLabelEventManager (*class in gitlab.v4.objects*), 358
ProjectMergeRequestResourceMilestoneEvent (*class in gitlab.v4.objects*), 359
ProjectMergeRequestResourceMilestoneEventManager (*class in gitlab.v4.objects*), 359
ProjectMergeRequestResourceStateEvent (*class in gitlab.v4.objects*), 359
ProjectMergeRequestResourceStateEventManager (*class in gitlab.v4.objects*), 359
ProjectMergeRequestReviewerDetail (*class in gitlab.v4.objects*), 360
ProjectMergeRequestReviewerDetailManager (*class in gitlab.v4.objects*), 360
ProjectMergeRequestStatusCheck (*class in gitlab.v4.objects*), 360
ProjectMergeRequestStatusCheckManager (*class in gitlab.v4.objects*), 360
ProjectMergeRequestTemplate (*class in gitlab.v4.objects*), 361
ProjectMergeRequestTemplateManager (*class in gitlab.v4.objects*), 361
ProjectMergeTrain (*class in gitlab.v4.objects*), 361
ProjectMergeTrainManager (*class in gitlab.v4.objects*), 361
ProjectMilestone (*class in gitlab.v4.objects*), 362
ProjectMilestoneManager (*class in gitlab.v4.objects*), 363
ProjectNote (*class in gitlab.v4.objects*), 363
ProjectNoteManager (*class in gitlab.v4.objects*), 363
ProjectNotificationSettings (*class in gitlab.v4.objects*), 364
ProjectNotificationSettingsManager (*class in gitlab.v4.objects*), 364

ProjectPackage (*class in gitlab.v4.objects*), 365
ProjectPackageFile (*class in gitlab.v4.objects*), 365
ProjectPackageFileManager (*class in gitlab.v4.objects*), 365
ProjectPackageManager (*class in gitlab.v4.objects*), 365
ProjectPackagePipeline (*class in gitlab.v4.objects*), 366
ProjectPackagePipelineManager (*class in gitlab.v4.objects*), 366
ProjectPackageProtectionRule (*class in gitlab.v4.objects*), 366
ProjectPackageProtectionRuleManager (*class in gitlab.v4.objects*), 366
ProjectPages (*class in gitlab.v4.objects*), 367
ProjectPagesDomain (*class in gitlab.v4.objects*), 367
ProjectPagesDomainManager (*class in gitlab.v4.objects*), 367
ProjectPagesManager (*class in gitlab.v4.objects*), 368
ProjectPipeline (*class in gitlab.v4.objects*), 368
ProjectPipelineBridge (*class in gitlab.v4.objects*), 369
ProjectPipelineBridgeManager (*class in gitlab.v4.objects*), 369
ProjectPipelineJob (*class in gitlab.v4.objects*), 369
ProjectPipelineJobManager (*class in gitlab.v4.objects*), 369
ProjectPipelineManager (*class in gitlab.v4.objects*), 370
ProjectPipelineSchedule (*class in gitlab.v4.objects*), 371
ProjectPipelineScheduleManager (*class in gitlab.v4.objects*), 372
ProjectPipelineSchedulePipeline (*class in gitlab.v4.objects*), 372
ProjectPipelineSchedulePipelineManager (*class in gitlab.v4.objects*), 372
ProjectPipelineScheduleVariable (*class in gitlab.v4.objects*), 373
ProjectPipelineScheduleVariableManager (*class in gitlab.v4.objects*), 373
ProjectPipelineTestReport (*class in gitlab.v4.objects*), 373
ProjectPipelineTestReportManager (*class in gitlab.v4.objects*), 374
ProjectPipelineTestReportSummary (*class in gitlab.v4.objects*), 374
ProjectPipelineTestReportSummaryManager (*class in gitlab.v4.objects*), 374
ProjectPipelineVariable (*class in gitlab.v4.objects*), 374
ProjectPipelineVariableManager (*class in gitlab.v4.objects*), 374
ProjectProtectedBranch (*class in gitlab.v4.objects*), 375
ProjectProtectedBranchManager (*class in gitlab.v4.objects*), 375
ProjectProtectedEnvironment (*class in gitlab.v4.objects*), 375
ProjectProtectedEnvironmentManager (*class in gitlab.v4.objects*), 376
ProjectProtectedTag (*class in gitlab.v4.objects*), 376
ProjectProtectedTagManager (*class in gitlab.v4.objects*), 376
ProjectPullMirror (*class in gitlab.v4.objects*), 377
ProjectPullMirrorManager (*class in gitlab.v4.objects*), 377
ProjectPushRules (*class in gitlab.v4.objects*), 378
ProjectPushRulesManager (*class in gitlab.v4.objects*), 378
ProjectRegistryProtectionRule (*class in gitlab.v4.objects*), 379
ProjectRegistryProtectionRuleManager (*class in gitlab.v4.objects*), 379
ProjectRegistryRepository (*class in gitlab.v4.objects*), 380
ProjectRegistryRepositoryManager (*class in gitlab.v4.objects*), 380
ProjectRegistryRepositoryProtectionRule (*class in gitlab.v4.objects*), 380
ProjectRegistryRepositoryProtectionRuleManager (*class in gitlab.v4.objects*), 380
ProjectRegistryTag (*class in gitlab.v4.objects*), 381
ProjectRegistryTagManager (*class in gitlab.v4.objects*), 381
ProjectRelease (*class in gitlab.v4.objects*), 382
ProjectReleaseLink (*class in gitlab.v4.objects*), 382
ProjectReleaseLinkManager (*class in gitlab.v4.objects*), 382
ProjectReleaseManager (*class in gitlab.v4.objects*), 383
ProjectRemoteMirror (*class in gitlab.v4.objects*), 383
ProjectRemoteMirrorManager (*class in gitlab.v4.objects*), 384
ProjectResourceGroup (*class in gitlab.v4.objects*), 384
ProjectResourceGroupManager (*class in gitlab.v4.objects*), 384
ProjectResourceGroupUpcomingJob (*class in gitlab.v4.objects*), 385
ProjectResourceGroupUpcomingJobManager (*class in gitlab.v4.objects*), 385
ProjectRunner (*class in gitlab.v4.objects*), 385
ProjectRunnerManager (*class in gitlab.v4.objects*), 385
PROJECTS (*gitlab.const.SearchScope attribute*), 433
projects (*gitlab.Gitlab attribute*), 206
projects (*gitlab.v4.objects.Group attribute*), 228

projects (*gitlab.v4.objects.User attribute*), 415
ProjectSecureFile (*class in gitlab.v4.objects*), 386
ProjectSecureFileManager (*class in gitlab.v4.objects*), 387
ProjectService (*class in gitlab.v4.objects*), 387
ProjectServiceAccount (*class in gitlab.v4.objects*), 387
ProjectServiceAccountAccessToken (*class in gitlab.v4.objects*), 387
ProjectServiceAccountAccessTokenManager (*class in gitlab.v4.objects*), 388
ProjectServiceAccountManager (*class in gitlab.v4.objects*), 388
ProjectServiceManager (*class in gitlab.v4.objects*), 389
ProjectSnippet (*class in gitlab.v4.objects*), 389
ProjectSnippetAwardEmoji (*class in gitlab.v4.objects*), 390
ProjectSnippetAwardEmojiManager (*class in gitlab.v4.objects*), 390
ProjectSnippetDiscussion (*class in gitlab.v4.objects*), 390
ProjectSnippetDiscussionManager (*class in gitlab.v4.objects*), 391
ProjectSnippetDiscussionNote (*class in gitlab.v4.objects*), 391
ProjectSnippetDiscussionNoteManager (*class in gitlab.v4.objects*), 391
ProjectSnippetManager (*class in gitlab.v4.objects*), 392
ProjectSnippetNote (*class in gitlab.v4.objects*), 392
ProjectSnippetNoteAwardEmoji (*class in gitlab.v4.objects*), 392
ProjectSnippetNoteAwardEmojiManager (*class in gitlab.v4.objects*), 393
ProjectSnippetNoteManager (*class in gitlab.v4.objects*), 393
ProjectStorage (*class in gitlab.v4.objects*), 393
ProjectStorageManager (*class in gitlab.v4.objects*), 394
ProjectTag (*class in gitlab.v4.objects*), 394
ProjectTagManager (*class in gitlab.v4.objects*), 394
ProjectTrigger (*class in gitlab.v4.objects*), 394
ProjectTriggerManager (*class in gitlab.v4.objects*), 395
ProjectUser (*class in gitlab.v4.objects*), 395
ProjectUserManager (*class in gitlab.v4.objects*), 395
ProjectVariable (*class in gitlab.v4.objects*), 395
ProjectVariableManager (*class in gitlab.v4.objects*), 396
ProjectWiki (*class in gitlab.v4.objects*), 396
ProjectWikiManager (*class in gitlab.v4.objects*), 397
protected_environments (*gitlab.v4.objects.Project attribute*), 279
protectedbranches (*gitlab.v4.objects.Group attribute*), 228
protectedbranches (*gitlab.v4.objects.Project attribute*), 279
protectedtags (*gitlab.v4.objects.Project attribute*), 279
PUBLIC (*gitlab.const.Visibility attribute*), 433
public() (*gitlab.v4.objects.SnippetManager method*), 409
publish() (*gitlab.v4.objects.ProjectMergeRequestDraftNote method*), 354
pull_mirror (*gitlab.v4.objects.Project attribute*), 279
pushrules (*gitlab.v4.objects.Group attribute*), 228
pushrules (*gitlab.v4.objects.Project attribute*), 279

Q

queue_metrics() (*gitlab.v4.objects.SidekiqManager method*), 407

R

raw() (*gitlab.v4.objects.ProjectArtifactManager method*), 287
raw() (*gitlab.v4.objects.ProjectFileManager method*), 312
rebase() (*gitlab.v4.objects.ProjectMergeRequest method*), 348
RedirectError, 450
refresh() (*gitlab.mixins.RefreshMixin method*), 455
RefreshMixin (*class in gitlab.mixins*), 455
refs() (*gitlab.v4.objects.ProjectCommit method*), 294
register_custom_action() (*in module gitlab.cli*), 431
registry_protection_repository_rules (*gitlab.v4.objects.Project attribute*), 279
registry_protection_rules (*gitlab.v4.objects.Project attribute*), 279
registry_repositories (*gitlab.Gitlab attribute*), 206
registry_repositories (*gitlab.v4.objects.Group attribute*), 228
RegistryRepository (*class in gitlab.v4.objects*), 397
RegistryRepositoryManager (*class in gitlab.v4.objects*), 397
reject() (*gitlab.v4.objects.User method*), 415
related_issues() (*gitlab.v4.objects.ProjectMergeRequest method*), 348
related_merge_requests() (*gitlab.v4.objects.ProjectIssue method*), 319
releases (*gitlab.v4.objects.Project attribute*), 279
remote_import() (*gitlab.v4.objects.ProjectManager method*), 343
remote_import_s3() (*gitlab.v4.objects.ProjectManager method*), 344

remote_mirrors (*gitlab.v4.objects.Project attribute*), 279
remove_none_from_dict() (*in module gitlab.utils*), 462
render() (*gitlab.mixins.BadgeRenderMixin method*), 451
reorder() (*gitlab.v4.objects.ProjectIssue method*), 319
repair() (*gitlab.v4.objects.GeoNode method*), 225
REPORTER (*gitlab.const.AccessLevel attribute*), 432
repositories (*gitlab.v4.objects.Project attribute*), 279
repository_archive() (*gitlab.v4.objects.RepositoryMixin method*), 398
repository_blob() (*gitlab.v4.objects.RepositoryMixin method*), 398
repository_compare() (*gitlab.v4.objects.RepositoryMixin method*), 399
repository_contributors() (*gitlab.v4.objects.RepositoryMixin method*), 399
repository_merge_base() (*gitlab.v4.objects.RepositoryMixin method*), 399
repository_raw_blob() (*gitlab.v4.objects.RepositoryMixin method*), 400
repository_tree() (*gitlab.v4.objects.RepositoryMixin method*), 400
RepositoryMixin (*class in gitlab.v4.objects*), 397
required (*gitlab.cli.CustomAction attribute*), 430
requires_id (*gitlab.cli.CustomAction attribute*), 430
reset_approvals() (*gitlab.v4.objects.ProjectMergeRequest method*), 349
reset_spent_time() (*gitlab.mixins.TimeTrackingMixin method*), 458
reset_time_estimate() (*gitlab.mixins.TimeTrackingMixin method*), 458
resource_groups (*gitlab.v4.objects.Project attribute*), 279
resource_iteration_events (*gitlab.v4.objects.ProjectIssue attribute*), 320
resource_weight_events (*gitlab.v4.objects.ProjectIssue attribute*), 320
resourcelabelevents (*gitlab.v4.objects.GroupEpic attribute*), 239
resourcelabelevents (*gitlab.v4.objects.ProjectIssue attribute*), 320
resourcelabelevents (*gitlab.v4.objects.ProjectMergeRequest attribute*), 349
resourcemilestoneevents (*gitlab.v4.objects.ProjectIssue attribute*), 320
resourcemilestoneevents (*gitlab.v4.objects.ProjectMergeRequest attribute*), 349
resourcestateevents (*gitlab.v4.objects.ProjectIssue attribute*), 320
resourcestateevents (*gitlab.v4.objects.ProjectMergeRequest attribute*), 349
response_content() (*in module gitlab.utils*), 462
RESTManager (*class in gitlab.base*), 428
RESTObject (*class in gitlab.base*), 428
RESTObjectList (*class in gitlab.base*), 429
restore() (*gitlab.v4.objects.Group method*), 228
restore() (*gitlab.v4.objects.Project method*), 279
RetrieveMixin (*class in gitlab.mixins*), 456
Retry (*class in gitlab.utils*), 461
retry() (*gitlab.v4.objects.ProjectJob method*), 331
retry() (*gitlab.v4.objects.ProjectPipeline method*), 368
revert() (*gitlab.v4.objects.ProjectCommit method*), 295
reviewer_details (*gitlab.v4.objects.ProjectMergeRequest attribute*), 349
Runner (*class in gitlab.v4.objects*), 401
RunnerAll (*class in gitlab.v4.objects*), 402
RunnerAllManager (*class in gitlab.v4.objects*), 402
RunnerJob (*class in gitlab.v4.objects*), 402
RunnerJobManager (*class in gitlab.v4.objects*), 402
RunnerManager (*class in gitlab.v4.objects*), 403
runners (*gitlab.Gitlab attribute*), 206
runners (*gitlab.v4.objects.CurrentUser attribute*), 217
runners (*gitlab.v4.objects.Group attribute*), 229
runners (*gitlab.v4.objects.Project attribute*), 280
runners_all (*gitlab.Gitlab attribute*), 206

S

saml_group_links (*gitlab.v4.objects.Group attribute*), 229
save() (*gitlab.mixins.SaveMixin method*), 456
save() (*gitlab.v4.objects.GroupEpicIssue method*), 240
save() (*gitlab.v4.objects.GroupLabel method*), 249
save() (*gitlab.v4.objects.ProjectFeatureFlag method*), 306
save() (*gitlab.v4.objects.ProjectFile method*), 309
save() (*gitlab.v4.objects.ProjectLabel method*), 334
SaveMixin (*class in gitlab.mixins*), 456
search() (*gitlab.Gitlab method*), 206
search() (*gitlab.v4.objects.Group method*), 229
search() (*gitlab.v4.objects.Project method*), 280
SearchScope (*class in gitlab.const*), 432
secure_files (*gitlab.v4.objects.Project attribute*), 280

SECURITY_MANAGER (*gitlab.const.AccessLevel attribute*), 432
sequence() (*gitlab.v4.objects.ProjectCommit method*), 295
service_accounts (*gitlab.Gitlab attribute*), 206
service_accounts (*gitlab.v4.objects.Group attribute*), 229
service_accounts (*gitlab.v4.objects.Project attribute*), 280
ServiceAccount (*class in gitlab.v4.objects*), 404
ServiceAccountManager (*class in gitlab.v4.objects*), 404
services (*gitlab.v4.objects.Project attribute*), 280
set() (*gitlab.mixins.SetMixin method*), 456
set() (*gitlab.v4.objects.FeatureManager method*), 222
set_approvers() (*gitlab.v4.objects.ProjectMergeRequestApprovalManager method*), 350
set_license() (*gitlab.Gitlab method*), 206
SetMixin (*class in gitlab.mixins*), 456
settings (*gitlab.Gitlab attribute*), 206
share() (*gitlab.v4.objects.Group method*), 229
share() (*gitlab.v4.objects.Project method*), 280
shared_projects (*gitlab.v4.objects.Group attribute*), 229
SharedProject (*class in gitlab.v4.objects*), 405
SharedProjectManager (*class in gitlab.v4.objects*), 405
show_caller (*gitlab.utils.WarnMessageData attribute*), 461
sidekiq (*gitlab.Gitlab attribute*), 207
SidekiqManager (*class in gitlab.v4.objects*), 406
signature() (*gitlab.v4.objects.ProjectCommit method*), 295
snapshot() (*gitlab.v4.objects.Project method*), 280
Snippet (*class in gitlab.v4.objects*), 407
SnippetManager (*class in gitlab.v4.objects*), 408
snippets (*gitlab.Gitlab attribute*), 207
snippets (*gitlab.v4.objects.Project attribute*), 281
ssl_verify (*gitlab.Gitlab attribute*), 207
star() (*gitlab.v4.objects.Project method*), 281
starred_projects (*gitlab.v4.objects.User attribute*), 415
StarredProject (*class in gitlab.v4.objects*), 410
StarredProjectManager (*class in gitlab.v4.objects*), 410
start() (*gitlab.v4.objects.ProjectPullMirrorManager method*), 377
start_branch (*gitlab.v4.objects.ProjectFile attribute*), 309
statistics (*gitlab.Gitlab attribute*), 207
status (*gitlab.v4.objects.CurrentUser attribute*), 217
status (*gitlab.v4.objects.User attribute*), 415
status() (*gitlab.v4.objects.GeoNode method*), 225
status() (*gitlab.v4.objects.GeoNodeManager method*), 226
status_checks (*gitlab.v4.objects.ProjectMergeRequest attribute*), 349
statuses (*gitlab.v4.objects.ProjectCommit attribute*), 295
stop() (*gitlab.v4.objects.ProjectEnvironment method*), 303
storage (*gitlab.v4.objects.Project attribute*), 281
subgroups (*gitlab.v4.objects.Group attribute*), 229
SubscribableMixin (*class in gitlab.mixins*), 457
subscribe() (*gitlab.mixins.SubscribableMixin method*), 457
sync() (*gitlab.v4.objects.ProjectRemoteMirror method*), 383

T

tags (*gitlab.v4.objects.Project attribute*), 281
tags (*gitlab.v4.objects.ProjectRegistryRepository attribute*), 380
take_ownership() (*gitlab.v4.objects.ProjectPipelineSchedule method*), 371
test() (*gitlab.v4.objects.GroupHook method*), 244
test() (*gitlab.v4.objects.ProjectHook method*), 315
test_report (*gitlab.v4.objects.ProjectPipeline attribute*), 369
test_report_summary (*gitlab.v4.objects.ProjectPipeline attribute*), 369
time_estimate() (*gitlab.mixins.TimeTrackingMixin method*), 458
time_stats() (*gitlab.mixins.TimeTrackingMixin method*), 458
timeout (*gitlab.Gitlab attribute*), 207
TimeTrackingMixin (*class in gitlab.mixins*), 457
to_json() (*gitlab.base.RESTObject method*), 429
Todo (*class in gitlab.v4.objects*), 411
todo() (*gitlab.mixins.TodoMixin method*), 459
TodoManager (*class in gitlab.v4.objects*), 411
TodoMixin (*class in gitlab.mixins*), 459
todos (*gitlab.Gitlab attribute*), 207
Topic (*class in gitlab.v4.objects*), 412
TopicManager (*class in gitlab.v4.objects*), 412
topics (*gitlab.Gitlab attribute*), 207
total (*gitlab.base.RESTObjectList property*), 430
total (*gitlab.GitlabList property*), 208
total_pages (*gitlab.base.RESTObjectList property*), 430
total_pages (*gitlab.GitlabList property*), 208
trace() (*gitlab.v4.objects.ProjectJob method*), 331
transfer() (*gitlab.v4.objects.Group method*), 229
transfer() (*gitlab.v4.objects.Project method*), 281

transfer_project() (*gitlab.v4.objects.Group method*), 230
trigger_pipeline() (*gitlab.v4.objects.Project method*), 282
triggers (*gitlab.v4.objects.Project attribute*), 282

U

unapprove() (*gitlab.v4.objects.ProjectMergeRequest method*), 349
unarchive() (*gitlab.v4.objects.Project method*), 282
unban() (*gitlab.v4.objects.User method*), 415
unblock() (*gitlab.v4.objects.User method*), 416
unfollow() (*gitlab.v4.objects.User method*), 416
unshare() (*gitlab.v4.objects.Group method*), 230
unshare() (*gitlab.v4.objects.Project method*), 282
unstar() (*gitlab.v4.objects.Project method*), 283
unsubscribe() (*gitlab.mixins.SubscribableMixin method*), 457
upcoming_jobs (*gitlab.v4.objects.ProjectResourceGroup attribute*), 384
update() (*gitlab.mixins.UpdateMixin method*), 459
update() (*gitlab.v4.objects.ApplicationAppearanceManager method*), 209
update() (*gitlab.v4.objects.ApplicationSettingsManager method*), 212
update() (*gitlab.v4.objects.GroupLabelManager method*), 250
update() (*gitlab.v4.objects.ProjectFeatureFlagManager method*), 307
update() (*gitlab.v4.objects.ProjectFileManager method*), 312
update() (*gitlab.v4.objects.ProjectLabelManager method*), 334
update_submodule() (*gitlab.v4.objects.RepositoryMixin method*), 401
UpdateMixin (*class in gitlab.mixins*), 459
upload() (*gitlab.v4.objects.GenericPackageManager method*), 224
url (*gitlab.Gitlab property*), 207
User (*class in gitlab.v4.objects*), 413
user (*gitlab.Gitlab attribute*), 207
user_activities (*gitlab.Gitlab attribute*), 207
user_agent_detail() (*gitlab.mixins.UserAgentDetailMixin method*), 459
UserActivities (*class in gitlab.v4.objects*), 416
UserActivitiesManager (*class in gitlab.v4.objects*), 416
UserAgentDetailMixin (*class in gitlab.mixins*), 459
UserContributedProject (*class in gitlab.v4.objects*), 417
UserContributedProjectManager (*class in gitlab.v4.objects*), 417
UserCustomAttribute (*class in gitlab.v4.objects*), 417
UserCustomAttributeManager (*class in gitlab.v4.objects*), 417
UserEmail (*class in gitlab.v4.objects*), 418
UserEmailManager (*class in gitlab.v4.objects*), 418
UserEvent (*class in gitlab.v4.objects*), 418
UserEventManager (*class in gitlab.v4.objects*), 418
UserGPGKey (*class in gitlab.v4.objects*), 419
UserGPGKeyManager (*class in gitlab.v4.objects*), 419
UserIdentityProviderManager (*class in gitlab.v4.objects*), 419
UserImpersonationToken (*class in gitlab.v4.objects*), 419
UserImpersonationTokenManager (*class in gitlab.v4.objects*), 420
UserKey (*class in gitlab.v4.objects*), 420
UserKeyManager (*class in gitlab.v4.objects*), 420
UserManager (*class in gitlab.v4.objects*), 420
UserMembership (*class in gitlab.v4.objects*), 423
UserMembershipManager (*class in gitlab.v4.objects*), 423
UserPersonalAccessToken (*class in gitlab.v4.objects*), 423
UserPersonalAccessTokenManager (*class in gitlab.v4.objects*), 423
UserProject (*class in gitlab.v4.objects*), 424
UserProjectManager (*class in gitlab.v4.objects*), 424
USERS (*gitlab.const.SearchScope attribute*), 433
users (*gitlab.Gitlab attribute*), 207
users (*gitlab.v4.objects.Project attribute*), 283
UserStatus (*class in gitlab.v4.objects*), 426
UserStatusManager (*class in gitlab.v4.objects*), 426

V

validate() (*gitlab.v4.objects.CiLintManager method*), 216
validate() (*gitlab.v4.objects.ProjectCiLintManager method*), 291
Variable (*class in gitlab.v4.objects*), 426
VariableManager (*class in gitlab.v4.objects*), 426
variables (*gitlab.Gitlab attribute*), 207
variables (*gitlab.v4.objects.Group attribute*), 230
variables (*gitlab.v4.objects.Project attribute*), 283
variables (*gitlab.v4.objects.ProjectPipeline attribute*), 369
variables (*gitlab.v4.objects.ProjectPipelineSchedule attribute*), 372
verify() (*gitlab.v4.objects.RunnerManager method*), 404
version() (*gitlab.Gitlab method*), 207
Visibility (*class in gitlab.const*), 433

W

warn() (*in module gitlab.utils*), 462

WarnMessageData (*class in gitlab.utils*), 461
WATCH (*gitlab.const.NotificationLevel attribute*), 432
WIKI_BLOBS (*gitlab.const.SearchScope attribute*), 433
wikis (*gitlab.v4.objects.Group attribute*), 230
wikis (*gitlab.v4.objects.Project attribute*), 283