

gitlab-projects-migrate

Author: Adrian DC

Description: Migrate GitLab projects from a GitLab group to another GitLab's group

Date: 21/08/2026

Version: 6.9.0-1-g9d635f9

► Usage

python v6.9.0 | 3.10 | 3.11 | 3.12 | 3.13 | downloads rate limited by upstream service

license Apache License 2.0

pipeline passed | bugs 0 | code smells 0 | coverage 40.5% | lines of code 3.4k

quality gate passed

pre-commit enabled | **commitizen** friendly | gci enabled | pre-commit-crocodile enabled

radiandevcore guidelines

Migrate GitLab projects from a GitLab group to another GitLab's group

Documentation: <https://radiandevcore.gitlab.io/tools/gitlab-projects-migrate>

Package: <https://pypi.org/project/gitlab-projects-migrate/>

Table of contents

- [Purpose](#)
- [Examples](#)
- [Usage](#)
- [Python GitLab configuration file](#)
- [Userspace available settings](#)
- [Environment available configurations](#)
- [Dependencies](#)
- [References](#)

Purpose

The migration can be performed between entirely different GitLab servers.

The following steps are required before using the tool:

- The groups need to be created manually by the user or by a GitLab administrator
- The GitLab user tokens must be created with an `api` scope (a short expiration date is recommended)

Examples

```
# Show the helper menu
gitlab-projects-migrate

# Migrate projects from one group to another, then migrate subgroups
gitlab-projects-migrate 'https://gitlab.com/old/group' 'https://gitlab.com/new/group'
gitlab-projects-migrate 'https://gitlab.com/old/group/subgroup1' 'https://gitlab.com/new/group/subgroup1'
gitlab-projects-migrate 'https://gitlab.com/old/group/subgroup2' 'https://gitlab.com/new/group/subgroup2'

# Migrate projects from one group to another, then archive or delete sources
gitlab-projects-migrate --archive-sources 'https://gitlab.com/old_group_1' 'https://gitlab.com/new_group_1'
gitlab-projects-migrate --delete-sources 'https://gitlab.com/old_group_2' 'https://gitlab.com/new_group_2'

# Migrate projects from one GitLab to another GitLab
gitlab-projects-migrate 'https://old.gitlab.com/group/subgroup' 'https://new.gitlab.com'

# Copy a project between two groups
gitlab-projects-migrate 'https://gitlab.com/old/group/project' 'https://gitlab.com/new/group'

# Copy and rename a project within a group
gitlab-projects-migrate 'https://gitlab.com/group/project' 'https://gitlab.com/group' 'new_project_name'

# Copy projects as templates and reset imported entities automatically
gitlab-projects-migrate --reset-entities 'Template/Issues' 'https://gitlab.com/group/template_issues'
'https://gitlab.com/group' 'issues'
gitlab-projects-migrate --reset-entities 'Template/Repository' 'https://gitlab.com/group/template_repository'
'https://gitlab.com/group' 'repository'
```

Usage

```
usage: gitlab-projects-migrate [-h] [--version] [--no-color] [--update-check] [--settings] [--set GROUP KEY VAL]
                             [-c FILES] [--archive-exports FOLDER] [--archive-sources | --delete-sources]
                             [--dry-run] [--exclude-group] [--exclude-subgroups] [--exclude-projects]
                             [--flatten-group] [--migrate-packages] [--confirm] [--overwrite]
                             [--rename-project NAME] [--normalize-names] [--available-entities]
                             [--reset-entities ENTITIES] [--set-avatar FILE] [--update-descriptions] [--]
                             [input_url_path] [output_url_namespace] [rename_single_project]
```

gitlab-projects-migrate: Migrate GitLab projects from a GitLab group to another GitLab's group

internal arguments:

```
-h, --help                # Show this help message
--version                 # Show the current version
--no-color                # Disable colors outputs with 'NO_COLOR=1'
                           # (or default settings: [themes] > no_color)
--update-check            # Check for newer package updates
--settings                # Show the current settings path and contents
--set GROUP KEY VAL       # Set settings specific 'VAL' value to [GROUP] > KEY
                           # or unset by using 'UNSET' as 'VAL'
```

credentials arguments:

```
-c FILES, --config FILES # Python GitLab configuration files (default: PYTHON_GITLAB_CFG environment)
```

migration arguments:

```
--archive-exports FOLDER # Store exported projects and groups to a folder
--archive-sources         # Archive sources after successful migration
--delete-sources          # Delete sources after successful migration
--dry-run                 # Enable dry run mode to check without saving
--exclude-group           # Exclude parent group migration
--exclude-subgroups       # Exclude children subgroups migration
--exclude-projects        # Exclude children projects migration
--flatten-group           # Flatten group projects upon migration
--migrate-packages        # Migrate input GitLab Packages to output GitLab projects
--confirm                 # Automatically confirm all removal and contents warnings
--overwrite               # Overwrite existing projects on output GitLab
--rename-project NAME     # Rename GitLab output project (only for single input project)
--normalize-names         # Normalize GitLab output group and project names
--available-entities      # List the available GitLab export/import entities known by the tool
--reset-entities ENTITIES # List of GitLab export/import entities to reset separated by ", " (default: Members)
```

general settings arguments:

```
--set-avatar FILE        # Set avatar of GitLab output projects and groups
--update-descriptions     # Update description of GitLab output projects and groups automatically
```

positional arguments:

```
--                # Positional arguments separator (recommended)
input_url_path     # Input GitLab group or project path URL
output_url_namespace # Output GitLab group or user namespace URL
rename_single_project # Rename GitLab output project (only for single input project)
```

environment variables:

```
GITLAB_INPUT_TOKEN # Input GitLab API token environment variable (fallback: GITLAB_TOKEN)
GITLAB_OUTPUT_TOKEN # Output GitLab API token environment variable (fallback: GITLAB_TOKEN)
CI_JOB_TOKEN        # GitLab CI job token environment variable (CI only)
```


Python GitLab configuration file

`gitlab-projects-migrate` uses the same configuration files as the `python-gitlab` API, holding domains, URL and private tokens credentials for the GitLab instances.

The default user configuration file can be created at `~/.python-gitlab.cfg`.

The `-c` or `--config` parameters can provide specific configuration files, otherwise the `PYTHON_GITLAB_CFG` environment variable can be used.

Example `~/.python-gitlab.cfg` **configuration file:**

`~/.python-gitlab.cfg`

```
[global]
default = gitlab.com
ssl_verify = true
timeout = 5

[gitlab.com]
url = https://gitlab.com
private_token = glpat-...

[gitlab.local.dev]
url = https://gitlab.local.dev
private_token = glpat-...

[gitlab.private.dev:4243]
url = https://gitlab.private.dev:4243
private_token = glpat-...
ssl_verify = /usr/local/share/ca-certificates/gitlab.private.dev.crt
```

`python-gitlab` **configuration files documentation:** [Getting started with the CLI / Configuration files](#)

Userspace available settings

`gitlab-projects-migrate` creates a `settings.ini` configuration file in a userspace folder.

For example, it allows to disable the automated updates daily check (`[updates] > enabled`)

The `settings.ini` file location and contents can be shown with the following command:

```
gitlab-projects-migrate --settings
```

Environment available configurations

`gitlab-projects-migrate` uses `colored` for colors outputs and `questionary` for interactive menus.

If colors of both outputs types do not match the terminal's theme,
an environment variable `NO_COLOR=1` can be defined to disable colors.

Dependencies

- [colored](#): Terminal colors and styles
 - [python-gitlab](#): A python wrapper for the GitLab API
 - [questionary](#): Interactive terminal user interfaces
 - [setuptools](#): Build and manage Python packages
 - [update-checker](#): Check for package updates on PyPI
-

References

- [commitizen](#): Simple commit conventions for internet citizens
- [git-cliff](#): CHANGELOG generator
- [gitlab-release](#): Utility for publishing on GitLab
- [gcil](#): Launch .gitlab-ci.yml jobs locally
- [mkdocs](#): Project documentation with Markdown
- [mkdocs-coverage](#): Coverage plugin for mkdocs documentation
- [mkdocs-exporter](#): Exporter plugin for mkdocs documentation
- [mkdocs-material](#): Material theme for mkdocs documentation
- [mypy](#): Optional static typing for Python
- [pre-commit](#): A framework for managing and maintaining pre-commit hooks
- [pre-commit-crocodile](#): Git hooks intended for developers using pre-commit
- [PyPI](#): The Python Package Index
- [twine](#): Utility for publishing on PyPI

gitlab-projects-migrate