

gitlab-projects-issues

Author: Adrian DC

Description: Generate GitLab project issues and milestones statistics automatically

Date: 02/02/2025

Version: 3.2.0-6-gec054fe

► Usage

pypi v3.2.0 python 3.8 | 3.9 | 3.10 | 3.11 | 3.12 downloads 429/month license Apache License 2.0

pipeline passed bugs 0 code smells 0 coverage 70.5% lines of code 1.3k

quality gate passed

pre-commit enabled commitizen friendly gcil enabled pre-commit-crocodile enabled

Generate GitLab project issues and milestones statistics automatically

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

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

Table of contents

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

Purpose

This tool can automatically generate issues and milestones statistics, by analyzing project's issues, detecting milestones and assignees.

If issues without time estimations are found, `~?` will be shown before time outputs.

Milestone statistics will automatically be injected in the milestone description, with a markdown table of assignees, timings, progress and issues total.

The following step is required before using the tool:

- The GitLab user tokens must be created with an `api` scope (a short expiration date is recommended)

Examples

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

# Inject milestones statistics into milestones' description
gitlab-projects-issues --milestones-statistics 'https://gitlab.com/group/project'







# Inject milestones statistics into milestones' description (with default 20h time per unestimated issues)
gitlab-projects-issues --milestones-statistics --default-estimate '20' 'https://gitlab.com/group/project'
```

Outputs

Milestones statistics will automatically be added to each milestone's description.

The following example shows how milestones statistics may look on a project:

Milestone statistics - MILESTONE NAME

Assignees	Issues	Estimated	Spent	Remaining	Progress
Without assignee	10	18d	18d	/	 100.00%
User ONE	22	42d	5d	37d	 11.90%
User TWO	29	50d 2h	20d 6h	29d 4h	 41.29%
User THREE	7	9d 2h	3d 4h	5d 6h	 37.84%
User FOUR	6	21d	/	21d	 0.00%
Total	74	135d 4h	47d 2h	93d 2h	 34.87%

Last update using gitlab-projects-issues : 2024-06-01 19:38:48 UTC

Usage

```
usage: gitlab-projects-issues [-h] [--version] [--no-color] [--update-check] [--settings] [--set GROUP KEY VAL]
                             [-c FILES] [--dump] [--default-estimate ESTIMATE] [--exclude-closed-issues]
                             [--milestone MILESTONE] [--create-milestone | --get-milestone | --milestones-
statistics]
                             [--exclude-closed-milestones] [--set-milestone-description TEXT]
                             [--set-milestone-state STATE] [--set-milestone-start-date DATE]
                             [--set-milestone-due-date DATE] [--]
                             [url_path]
```

gitlab-projects-issues: Generate GitLab project issues and milestones statistics automatically

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)
```

common arguments:

```
--dump                    # Dump Python objects of projects
```

issues arguments:

```
--default-estimate ESTIMATE # Default issue time estimate if none provided in hours (default: 8)
--exclude-closed-issues     # Exclude issues in closed state
```

milestones arguments:

```
--milestone MILESTONE     # Use a specific milestone by name, by ID, or "None"
--create-milestone         # Create a new milestone
--get-milestone            # Get existing milestone
--milestones-statistics    # Inject milestones statistics into milestones description
--exclude-closed-milestones # Exclude milestones in closed state
--set-milestone-description TEXT # Set milestone description
--set-milestone-state STATE # Set milestone state [activate,close]
--set-milestone-start-date DATE # Set milestone start date
--set-milestone-due-date DATE # Set milestone due date
```

positional arguments:

```
--                # Positional arguments separator (recommended)
url_path           # GitLab project path URL
```

environment variables:

```
GITLAB_TOKEN # GitLab API token environment variable
CI_JOB_TOKEN  # GitLab CI job token environment variable (CI only)
```

Python GitLab configuration file

`gitlab-projects-issues` 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-issues` 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-issues --settings
```

Environment available configurations

`gitlab-projects-issues` uses `colored` for colors outputs.

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
 - [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-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-issues

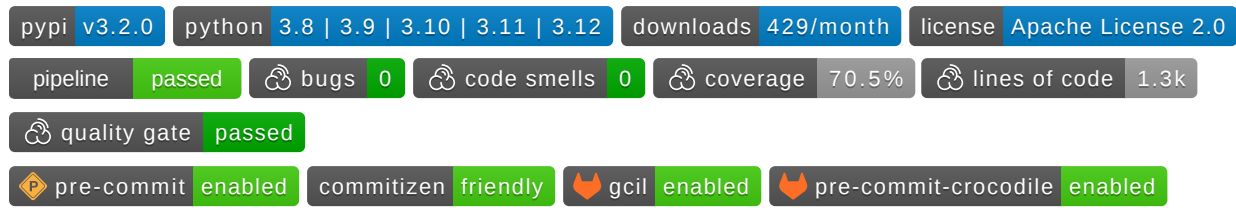
Author: Adrian DC

Description: Generate GitLab project issues and milestones statistics automatically

Date: 02/02/2025

Version: 3.2.0-6-gec054fe

► Webhooks



Generate GitLab project issues and milestones statistics automatically

Table of contents

- [Issues access token](#)
- [CI/CD webhooks project](#)
- [Issues webhook configuration](#)

Issues access token

- **Settings:**
 - **Action:** Add new token
 - **Token name:** <path/to/webhooks/project>-reporter-api
 - **Expiration date:** yyyy-12-31
 - **Select a role:** Reporter
 - **Select scopes:**
 - **api:** Enable
 - **Action:** Create project access token
 - **Your new project access token:**
 - **Action:** Copy project access token
-

CI/CD webhooks project

- **Settings / CI/CD / Variables:**
 - **Action:** Add variable
 - **Visibility:** Masked and hidden
 - **Protect variable:** Enable
 - **Expand variable reference:** Enable
 - **Key:** `GITLAB_TOKEN`
 - **Value:** `PASTE_<path/to/webhooks/project>-reporter-api_TOKEN`
 - **Action:** Add variable
- **Settings / Access tokens:**
 - **Action:** Add new token
 - **Token name:** `webhook-issues`
 - **Expiration date:** `yyyy-12-31`
 - **Select a role:** `Maintainer`
 - **Select scopes:**
 - **api:** Enable
 - **Action:** Create project access token
 - **Your new project access token:**
 - **Action:** Copy project access token
- **Repository sources with a `gitlab-projects-issues` CI/CD pipeline:**

Webhooks / README.md

```
# GROUP - PATH - Webhooks

**Documentation:** <https://radiandevcore.gitlab.io/tools/gitlab-projects-issues>
```

Webhooks / .gitignore

```
# Cache intermediates
.cache
```

Webhooks / .gitlab-ci.yml

```
.image-issues:
  image: registry.gitlab.com/radiandevcore/tools/gitlab-projects-issues/gitlab-projects-issues:latest

variables:
  # Predefined CI variables
  GITLAB_TOKEN:
    value: ''
    description: 'GitLab private token'
  ISSUES_DEFAULT_ESTIMATE:
    value: '20'
    description: 'Issues - Default estimated times for issues'
  ISSUES_PROJECT:
    value: 'group/path/to/issues'
    description: 'Issues - GitLab project path with namespace'

stages:
  - issues

issues:milestones:
  stage: issues
  extends:
    - .image-issues
  interruptible: true
  resource_group: 'issues.milestones'
  script:
    # Debounce pipeline executions
    - |
      set +x
      mkdir -p ./cache/
      touch -d '-5 minutes' ./cache/timestamp_debounce
      if [ -f ./cache/timestamp_last ] && [ ./cache/timestamp_last -nt ./cache/timestamp_debounce ]; then
        echo ' '
        echo ' [INFO] Debouncing API access, cancelling job...'
        echo ' '
        exit 0
      else
        touch ./cache/timestamp_last
      fi
    # Generate milestones statistics
    - |
      gitlab-projects-issues \
        --milestones-statistics \
        --default-estimate "${ISSUES_DEFAULT_ESTIMATE:?}" \
        "${CI_SERVER_URL}/${ISSUES_PROJECT:?}"

cache:
  paths:
    - .cache/timestamp_last
  start_in: 5 minutes
  when: delayed
```

Issues webhook configuration

- **Settings / Webhooks:**
 - **Action:** Add new webhook
 - **URL:** `https://<GITLAB_DOMAIN>/api/v4/projects/<PATH%2FTO%2FWEBHOOK%2FPROJECT>/pipeline?ref=<BRANCH>`
 - **Action:** Add custom headers
 - **Header name:** `PRIVATE-TOKEN`
 - **Header value:** `PASTE_webhook-issues_TOKEN`
 - **Name:** `gitlab-projects-issues`
 - **Trigger:**
 - **Issues events:** Enable
 - **Enable SSL verification:** Enable
 - **Action:** Add webhook
 - **Action:** Test (`gitlab-projects-issues`) / Issue events

gitlab-projects-issues