
badges-gitlab

Felipe P. Silva

Nov 25, 2023

CONTENTS

1	Installation	3
2	General Usage	5
3	Author	7
4	Further Documentation	9
	Python Module Index	25
	Index	27

This project was created to generate badges for Gitlab in CI jobs, mainly for private repositories where other common methods are not available (direct API Calls, shields.io, etc...).

By default, Gitlab supports only two types of badges: pipeline and test coverage.

These badges are better detailed at: [Gitlab Project Badges](#).

INSTALLATION

You can install this package from pypi using pip.

```
$ pip install badges-gitlab
```


GENERAL USAGE

```
usage: badges-gitlab [-h] [-p PATH] [-t TOKEN] [--junit-xml FILE_PATH] [-s LABEL MESSAGE_↵
↵COLOR]
[-lb URLS [URLS ...]] [-V]
```

Generate Gitlab Badges using JSON files and API requests. Program version v0.0.0.

optional arguments:

```
-h, --help                show this help message and exit
-p TEXT, --path TEXT      path where json and badges files will be generated/located_↵
↵(default: ./public/badges/)
-t TEXT, --token TEXT     specify the private-token in command line (default: ${PRIVATE_↵
↵TOKEN})
--junit-xml TEXT          specifies the path of a JUnit XML file for parsing the test_↵
↵results
-s LABEL MESSAGE COLOR, --static-badges LABEL MESSAGE COLOR
                           specify static badges in command line using lists
-lb URLS [URLS ...], --link-badges URLS [URLS ...]
                           specify shields.io urls to download badges
-V, --version             returns the package version
```


Felipe Pinheiro Silva

3.1 Contributors

Benjamin Maréchal (irmo322)

FURTHER DOCUMENTATION

Slowly moving documentation to [ReadTheDocs](#).

4.1 Usage

4.1.1 Common Usage

Install this package from pip and run it in your project folder.

```
$ pip install badges-gitlab
$ badges-gitlab
```

This package was intended to be used in CI jobs, but if you want to test locally, you must point a folder with the json files in the format used by [shields.io endpoint](#), otherwise it won't work because most of the badges uses the Gitlab API Token and CI Environment Variables.

4.1.2 Continuous Integration Job

Below it is an example of a job running at the end of the pipeline in the default branch (main) and using cache for job optimization. Make sure to adequate your pipeline.

To ensure all possible badges are generated, include the [personal access token](#) as an environment variable direct into the `.gitlab-ci.yml` or in the [CI/CD Variables configuration](#).

```
badges:
  image: python:3.9
  stage: badges
  variables:
    PIP_CACHE_DIR: "$CI_PROJECT_DIR/.cache/pip"
    PRIVATE_TOKEN: $ACCESS_TOKEN
  cache:
    key: badges
    paths:
      - .cache/pip
      - venv/
  before_script:
    - python -V
    - pip install virtualenv
    - virtualenv venv
```

(continues on next page)

(continued from previous page)

```
- source venv/bin/activate
script:
- pip install badges-gitlab
- badges-gitlab -V
- badges-gitlab
artifacts:
  when: always
  paths:
  - public/badges/*.svg
  expire_in: 3 months
rules:
- if: $CI_COMMIT_BRANCH == $CI_DEFAULT_BRANCH
  when: always
  allow_failure: true
```

As the badges are generated only during this job, and if you want to make sure there are available for some time. So, adjust the expiration of the artifacts accordingly.

Schedule Pipelines

Some badges have dynamic data and are generated only during this job, and the data can be outdated. If you don't want to use third party APIs all the time to generate the badges (sometimes these APIs fail), you could use the [pipeline schedule](#) function in conjunction with the rules option to run once a day as example.

```
rules:
- if: '$CI_PIPELINE_SOURCE == "schedule"'
```

Non compatible Python jobs

It is possible to use artifacts from previous jobs with information to generate new badges.

The json files must be “shields.io” compliant and located in the folder specified in the badges job (default: public/badges). Below is the accepted format.

```
{
  "schemaVersion": 1,
  "label": "hello",
  "message": "sweet world",
  "color": "orange"
}
```

Dockerfile Example

Alternatively, you can optimize even further the job building a docker image for this job and uploading to [Gitlab Container Registry](#).

```
FROM python:3.9-alpine

MAINTAINER foo@bar.com

RUN pip install badges-gitlab
```

4.1.3 Using the Badges

This package seeks to use the post jobs availability of the artifacts [through links](#), which are described in [Gitlab Documentation](#).

Gitlab Project Badges

You can insert using the project badges [Project Badges](#).

Examples of a link for the project license in project badges section:

```
https://gitlab.com/{project_path}/-/jobs/artifacts/{default_branch}/raw/public/badges/
↪license_name.svg?job=badges
```

Readme

Other option is to use in the Readme file, through links. In Gitlab you can leverage the [relative links feature](#).

Example of a link in a markdown Readme.

```
![License](../jobs/artifacts/main/raw/public/badges/license_name.svg?job=badges)
```

4.2 Configuration

It is now possible to configure this tool using pyproject.toml. Currently the parameters path, junit_xml, static_badges and link_badges are supported. Example of pyproject.toml section:

```
[tool.badges_gitlab]
path = "public/badges"
junit_xml = "tests/report.xml"
# List of Lists Format [{"label", "message", "color"}]
static_badges = [
    ["conventional commits", "1.0.0", "yellow"]
]
# List of Links
link_badges = [
    'https://img.shields.io/pypi/wheel/badges-gitlab'
]
```

Priority is:

- Command line parameters
- Toml file configuration

4.3 Frequently Asked Questions

Is this project for me?

Although it is possible to generate badges with other API's such as shields.io, usually this process is not available in private repositories.

So if you are hosting a public project, this package is not specifically meant for you as you can workaround with other easier implementations.

One good project to be consulted is from [@asdoi](https://gitlab.com/asdoi/gitlab-badges), available on <https://gitlab.com/asdoi/gitlab-badges> and https://gitlab.com/asdoi/git_badges.

But, if you are hosting a private project and don't want to expose your project (Gitlab pages) or don't want to risk exposing your credentials (API Requests), maybe this project is for you.

Another reason would be to avoid overloading servers (e.g. shields.io) with unnecessary requests for (re)creating badges.

How does it work?

Some design choices were made to create this package.

1. The badges' generation were converted into two stages:
 - The first stage uses the Gitlab API (if the private-token turns out to be valid) to generate the json for some badges.
 - The second stage gets all the JSON files from the target folder and creates badges using anybadge.
2. These two stages have a purpose, if any other CI Pipeline job generates json files with their own data, you can also use these files to create badges.
3. The default directory is `/public/badges`:
 - This folder may be used later for Gitlab pages, although this can be modified through parameters.

4.4 Contributing

Merge requests are welcome. For major changes, please open an issue first to discuss what you would like to change.

Please make sure to update tests as appropriate.

Below are instructions to set up the environment and testing.

4.4.1 Installing Environment

This package uses the Pipenv Virtual Environment for managing the dependencies. They are all listed in the Pipfile. Current supported version is Python ≥ 3.8 , and this virtual environment is configured for Python 3.8.

Install pipenv if you don't have it yet.

```
$ pip install -U pipenv
```

Clone the Repository and download the dependencies.

```
$ git clone https://gitlab.com/felipe_public/badges-gitlab.git
$ cd badges-gitlab
$ pipenv install --dev
```

4.4.2 Testing

This project uses some tools for static code analysis and the python embedded unittest for Unit Testing.

To run locally the static tests, a script was developed.

```
$ pipenv run statictest
```

To run unittests locally you can use a scripted short version.

```
$ pipenv run unit
```

Dependencies Requirements

This package depends on the following dependencies:

- Python Gitlab API
- Anybadge
- Iso8601
- xmldict
- toml

4.5 Modules Documentation

4.5.1 CLI

Main Package File, parses CLI arguments and calls functions.

`badges_gitlab.cli.main()` → None

Main Function for calling arg parser and executing functions.

`badges_gitlab.cli.parse_args(args) → Namespace`

Create arguments and parse them returning already parsed arguments.

Args:

args: arguments to parse

Returns:

argparse.Namespace: argparse object with parser arguments

4.5.2 Badges API

This module uses the Gitlab API Functions to create json files for badges.

`badges_gitlab.badges_api.create_api_badges(directory_path: Any, private_token: str) → None`

Authenticates to API and call the json creation functions.

Main function in the module.

Args:

directory_path (Any): destination path to the json files. private_token (str): user token with api access for getting project data.

`badges_gitlab.badges_api.general_data(project_ref: Any, directory_path: Any) → None`

Retrieves General Data from project.

Licences may not be found, therefore the Try function

Args:

project_ref (Any): project referece from gitlab directory_path (Any): path where json file will be written

`badges_gitlab.badges_api.issues(project_ref: Any, directory_path: Any) → None`

Retrieve project issues data from the project dict.

Format : {'statistics': {'counts': {'all': 30, 'closed': 13, 'opened': 7}}}

Args:

project_ref (Any): project referece from gitlab directory_path (Any): path where json file will be written

`badges_gitlab.badges_api.releases_commits(project_ref: Any, directory_path: Any) → None`

Retrieves Releases, Tags and Commits related data.

Args:

project_ref (Any): project referece from gitlab directory_path (Any): path where json file will be written

`badges_gitlab.badges_api.validate_path(directory_path: Any) → None`

Validates destination path, if not found, creates it.

Args:

directory_path (Any): path to validate.

4.5.3 Badges JSON

This module has functions to manipulate and generate standardized json files.

`badges_gitlab.badges_json.json_badge(directory_path, filename: str, json_string: dict) → None`

Write to JSON file to disk to the specified directory.

Args:

directory_path (_type_): destination directory filename (str): destination json filename json_string (dict): dictionary to be converted to json

`badges_gitlab.badges_json.print_json(label: str, message: str, color: str) → dict`

Returns a JSON (Dict) in the format used by shields.io to create badges.

Args:

label (str): label for the badge message (str): message in the badge color (str): color of the badge

Returns:

dict: final dictionary version with all fields.

`badges_gitlab.badges_json.validate_json_path(directory_path: Any) → None`

Validate destination path if there are any json files.

Args:

directory_path (Any): destination path to check

4.5.4 Badges Static

This module handles generation of static badges.

It reads from pyproject.toml or from command line parameters

`badges_gitlab.badges_static.convert_list_json_badge(badges_list: list) → list`

Converts the list of badges list to json format to be printed in the json file.

Args:

badges_list (list): _description_

Raises:

TypeError: if it is no a list.

Returns:

list: list of json dicts

`badges_gitlab.badges_static.download_badges(directory: str, badges_urls: list)`

Get the badges from websites and save it locally.

Now this was written specifically for shields.io but it must be studied to use other websites.

Args:

directory (str): destination directory badges_urls (list): list of badges url to be used

`badges_gitlab.badges_static.extract_svg_title(xml_svg) → str`

Get the raw SVG (XML), convert to dict and retrieve the title.

Args:

xml_svg: svg in xml format

Returns:

str: returns the label as snake case

`badges_gitlab.badges_static.print_static_badges(directory: str, badges_list: list)`

Call functions in order to write a file with json badge information.

Args:

directory (str): destination for the file badges_list (list): list of badges (dict) to be used

`badges_gitlab.badges_static.to_snake_case(value: str) → str`

Convert the label from a badge to snake case.

Args:

value (_type_): string to evaluate.

Returns:

str: string in snake in case

4.5.5 Badges SVG

Creates standardized badges files using anybadge package.

This function creates the badge svg file from json file with the shields.io (<https://shields.io/endpoint>) json format badge format = {"schemaVersion":1,"label":"hello","message":"sweet world","color":"orange"}

`badges_gitlab.badges_svg.print_badges(directory_path: Any) → None`

Print badges from json files.

Iterates within the directory finding json files and then use anybadge pkg to generate them.

Args:

directory_path (Any): _description_

`badges_gitlab.badges_svg.replace_space(string: str) → str`

Replaces any spaces to make it easier to use later as url in badges linking.

Args:

string (str): string to evaluate

Returns:

str: string without space

`badges_gitlab.badges_svg.validate_json_path(directory_path: Any) → bool`

Check if there is any json files or the directory is valid.

Args:

directory_path (Any): destination path to evaluate.

Returns:

bool: validation result

4.5.6 Badges Test

Generate Tests Badges related by parsing JUnit XML files.

`badges_gitlab.badges_test.create_badges_test(json_directory, file_path: str) → str`

Parses a JUnit XML file to extract general information about the unit tests.

Args:

json_directory: file where the json file will be saved file_path (str): junit xml file path

Returns:

str: string with the operation result

`badges_gitlab.badges_test.create_test_json_badges(json_directory, test_results: list) → str`

Create badges from a test results list.

This function returns parses a list with the test summary to json format. The list order must be: total tests, total failures, total errors, total_skipped, total_time

Args:

json_directory: directory where json will be saved. test_results (list): list of the test results

Returns:

str: string with the result of the operation

`badges_gitlab.badges_test.tests_statistics(stats_tests_dict: dict, testsuite) → dict`

Returns the Test Statistics Dictionary with added values from the testsuite.

Args:

stats_tests_dict (dict): dictionary with listed tests testsuite ([junitparser.junitparser.TestSuite]): a testsuite xml node needed for filling the stats tests dictionaty

Returns:

dict: returns the stats_tests_dict with the new values.

4.5.7 Read pyproject.toml

Read pyproject.toml and import the parameters to be used in the function.

`badges_gitlab.read_pyproject.load_pyproject(file_path: str) → dict`

Load the tool.badges_gitlab section from the toml file.

Most of the cases it is pyproject.toml because it is hardcoded into main function

Args:

file_path (str): path for the pyproject.toml

Returns:

dict: dictionary with the configuration

`badges_gitlab.read_pyproject.pyproject_config(file_path: str) → dict`

Load pyproject.toml and return the content as dict.

Args:

file_path (str): file path for the pyproject

Returns:

dict: dictionary with the configuration for the badges_gitlab tool

`badges_gitlab.read_pyproject.pyproject_exists(file_path: str) → bool`

Verify if the file exists, used internally.

Args:

file_path (str): file path for the pyproject file

Returns:

bool: true or false for the operation.

MIT License

Copyright (c) 2021 Felipe Pinheiro Silva

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

4.6 CHANGELOG

4.6.1 v1.0.1 (2023-11-25)

Fix

- fix: update bugtracker link in the pyproject.toml ([c1ec397](#))

4.6.2 v1.0.0 (2023-11-25)

Chore

- chore(release): v1.0.0

Automatically generated by python-semantic-release [skip ci] ([e20eb56](#))

- chore: organize dependencies in pyproject ([3544c14](#))
- chore: updates project to poetry and fixes files for it ([0a5ce67](#))

Ci

- ci: update release step for poetry ([e21522d](#))
- ci: update code coverage setup (#34) ([fc7e095](#))

Documentation

- docs: update docstrings in functions ([9ea8759](#))
- docs: update links in the badges (#36)

Closes #36. Please enter the commit message for your changes. Lines starting ([3effbb9](#))

- docs: update read the docs configuration ([0156323](#))

Test

- test: refactor to be in different files (#22)

Closes #22. ([d93d78a](#))

4.6.3 v0.8.4 (2022-06-06)

Chore

- chore(release): 0.8.4

Automatically generated by python-semantic-release [skip ci] ([4017558](#))

Ci

- ci: adds build test job to run gitlab-badges before going into main ([57fc795](#))
- ci: replaces badges job with non docker image process ([a67709a](#))
- ci: updates deps job by removing the pipen remove environment ([2aba05e](#))

Fix

- fix: adds flags to lists to supress warnings on python-gitlab api lists(#31) ([22e246a](#))
- fix: updates call to the API on issues statistics (#33)

This changes the format of the call to python-gitlab from issuesstatiscs to issues_statistics. ([6289f12](#))

- fix: updates dependencies, fixing python-gitlab (#31) ([804d5ba](#))

4.6.4 v0.8.3 (2021-10-16)

Chore

- chore(release): 0.8.3

Automatically generated by python-semantic-release [skip ci] ([a35216f](#))

Ci

- ci: fixes release problem

Includes install git and change semver execution to pipenv run. ([dc70afb](#))

- ci: include SAST tool

As a free feature for Gitlab, I am introducing the SAST scanning to verify for vulnerabilities and understand how this tool works. ([3bc52a9](#))

- ci: adds dependencies to dev for mypy (#27)

Mypy requires dependencies to run their tests, so it was included in Pipfile as dev. ([0abe582](#))

- ci: removes docker images from tests and releases (#27)

With this modification, the stages on the tests uses the default dependencies job to manage dependencies. This prevents problems with changes in dependencies not being caught in CI. Also, modifies the Deps task to recreate Venv before every pipeline. ([ed9968a](#))

Fix

- fix: change definition of color for test result badges (#29)

This modification will output green badges if there are some skipped tests. Previously, skipped tests returned red badges.

This closes #29. ([0845f4c](#))

- fix: minor typo in badges test complete svg (#30)

Includes a missing white space to correctly generate a badge.

This closes #30 ([82e567b](#))

- fix: includes back the xmlltodict (#28)

This includes the xmlltodict that was wrongly removed in the previous version.

This closes #28. ([57aa4bb](#))

4.6.5 v0.8.2 (2021-10-11)

Chore

- chore(release): 0.8.2

Automatically generated by python-semantic-release [skip ci] ([dbfd5d7](#))

- chore: updates lock file due to dep changes

Change in deps from xmlltodict to junit-parser required to update lock file. ([e8b8abb](#))

CI

- ci: enables pipelines on MR events. ([9b7fc36](#))

Fix

- fix: bug with parsing xml without testsuites (#26)
- Replacing xmlltodict by junitparser, with replacement in Pipfile and setup.py.
- Adding relevant unit tests.
- Adding irmo322 (Benjamin Maréchal) to contributors in README.md ([f4b621c](#))

Test

- test(static): code improvements for static tests ([26124ee](#))

4.6.6 v0.8.1 (2021-09-12)

Chore

- chore(release): 0.8.1

Automatically generated by python-semantic-release [skip ci] ([aedd4df](#))

- chore: enables the first version by removing the flag major on zero ([e9a6ac8](#))
- chore: adds issue templates (#4)

This closes #4. ([a3b887b](#))

- chore: includes python-semantic-release in dev deps ([4bfa52](#))

Ci

- ci: fix expire in indentation ([48f3723](#))
- ci: update badges expiration date ([182c9bc](#))
- ci: creates an image for static test jobs

This commit refactors the static tests stage by including a docker image for these jobs. ([a2d8265](#))

Documentation

- docs: update docs to include asdoi projects (#24).

This closes #24. ([7ffe89a](#))

Fix

- fix: bug with parsing xml with one testsuite only (#25)

Includes a try except statement in case returns an error from iteration. It expects usually an array of dictionaries, but this considers when there is only a single dictionary.

This closes #25. ([6be7060](#))

- fix: adds incrementing func for stats tests dict (#25)

This function incrementes the test statistics dictionary that is going to be used for the create_badges_test function. It is part of the solution for issue #25. ([41417d1](#))

Test

- test(unit): adds test for bug in parsing junit xml file (#25) ([b957044](#))
- test(unit): adds test for function tests_statistics (#25) ([d0afd79](#))
- test(unit): add vcrpy to mock http requests (#23)

Tests that need http requests passes through vcrpy which records the response and replay on the next iterations.

This closes #23. ([a972e29](#))

4.6.7 v0.8.0 (2021-05-26)

Chore

- chore(release): 0.8.0

Automatically generated by python-semantic-release [skip ci] ([ed4310b](#))

Ci

- ci: updates docker badges image to 0.7.0 ([d7a52a6](#))

Feature

- feat: include documentation using sphinx

The documentation is build using sphinx and published in the readthedocs website. For that, the python version was downgraded to 3.8 and all related files were changed, including Pipfile and Pipfile.lock, breaking current development environment. ([6eb6b3f](#))

4.6.8 v0.7.0 (2021-05-23)

Chore

- chore(release): 0.7.0

Automatically generated by python-semantic-release [skip ci] ([9a7883a](#))

- chore: adds shields.io links to pyproject.toml ([7c9c430](#))

Documentation

- docs: update documentation to feature #13 ([52d6879](#))

Test

- test(unit): adds tests for shields.io feature (#13) ([bf5a488](#))

PYTHON MODULE INDEX

b

- `badges_gitlab.badges_api`, [14](#)
- `badges_gitlab.badges_json`, [15](#)
- `badges_gitlab.badges_static`, [15](#)
- `badges_gitlab.badges_svg`, [16](#)
- `badges_gitlab.badges_test`, [16](#)
- `badges_gitlab.cli`, [13](#)
- `badges_gitlab.read_pyproject`, [17](#)

B

badges_gitlab.badges_api
module, 14
badges_gitlab.badges_json
module, 15
badges_gitlab.badges_static
module, 15
badges_gitlab.badges_svg
module, 16
badges_gitlab.badges_test
module, 16
badges_gitlab.cli
module, 13
badges_gitlab.read_pyproject
module, 17

C

convert_list_json_badge() (in
badges_gitlab.badges_static), 15
create_api_badges() (in
badges_gitlab.badges_api), 14
create_badges_test() (in
badges_gitlab.badges_test), 16
create_test_json_badges() (in
badges_gitlab.badges_test), 17

D

download_badges() (in
badges_gitlab.badges_static), 15

E

extract_svg_title() (in
badges_gitlab.badges_static), 15

G

general_data() (in module badges_gitlab.badges_api),
14

I

issues() (in module badges_gitlab.badges_api), 14

J

json_badge() (in module badges_gitlab.badges_json),
15

L

load_pyproject() (in module
badges_gitlab.read_pyproject), 17

M

main() (in module badges_gitlab.cli), 13
module
badges_gitlab.badges_api, 14
badges_gitlab.badges_json, 15
badges_gitlab.badges_static, 15
badges_gitlab.badges_svg, 16
badges_gitlab.badges_test, 16
badges_gitlab.cli, 13
badges_gitlab.read_pyproject, 17

P

parse_args() (in module badges_gitlab.cli), 13
print_badges() (in module badges_gitlab.badges_svg),
16
print_json() (in module badges_gitlab.badges_json),
15
print_static_badges() (in module
badges_gitlab.badges_static), 15
pyproject_config() (in module
badges_gitlab.read_pyproject), 17
pyproject_exists() (in module
badges_gitlab.read_pyproject), 17

R

releases_commits() (in module
badges_gitlab.badges_api), 14
replace_space() (in module
badges_gitlab.badges_svg), 16

T

tests_statistics() (in module
badges_gitlab.badges_test), 17

`to_snake_case()` (in *module*
badges_gitlab.badges_static), 16

V

`validate_json_path()` (in *module*
badges_gitlab.badges_json), 15

`validate_json_path()` (in *module*
badges_gitlab.badges_svg), 16

`validate_path()` (in *module*
badges_gitlab.badges_api), 14