

---

# **exporter**

***Release 0.0.2***

**Martin Šafránek**

**Mar 10, 2021**



**CONTENTS:**

<b>1</b>	<b>Quickstart</b>	<b>3</b>
1.1	How to install . . . . .	3
1.2	Examples . . . . .	3
<b>2</b>	<b>Projects file format</b>	<b>5</b>
2.1	Example . . . . .	5
<b>3</b>	<b>Config file format</b>	<b>7</b>
3.1	Example . . . . .	7
<b>4</b>	<b>Return codes</b>	<b>9</b>
<b>5</b>	<b>Testing</b>	<b>11</b>
<b>6</b>	<b>Manpage</b>	<b>13</b>
<b>7</b>	<b>API documentation</b>	<b>15</b>
7.1	Package . . . . .	15
<b>8</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>
	<b>Index</b>	<b>21</b>



Exporter automates exporting projects from [FIT CTU GitLab](#) to [GitHub](#). It allows choosing which projects to export, their renaming, and visibility on GitHub. Also, there is an option for exporting all of your GitLab projects at once.

Check Quickstart [HERE](#) for examples.



## QUICKSTART

## 1.1 How to install

Install from [PyPi](#)

```
$ pip install fit-ctu-gitlab-exporter
```

## 1.2 Examples

Allowed config file format is described [here](#),  
projects file format is described [here](#).

### 1.2.1 1. Export multiple projects

```
$ exporter -c config -p projects
```

where content of file `projects` is

```
my_gitlab_project  
my_other_gitlab_project
```

### 1.2.2 2. Export, rename and set visibility for multiple projects

```
$ exporter -c config -p projects
```

where content of file `projects` is

```
my_gitlab_project -> avl_tree_cpp  
my_secret_project -> secret_project private
```

### 1.2.3 3. Export all projects

Export all of your GitLab projects. Run at most `batch-size` exports simultaneously.

```
$ exporter -c config --export-all --batch-size=5
```



## PROJECTS FILE FORMAT

Projects file is required and is passed by `--projects`, `-p` option. Each line corresponds to one exported project. Format of each line is

```
gitlab_name [-> github_name] [private/public]
```

### 2.1 Example

Below is an example of a valid project file. Every format is used here.

```
1 my_project
2 my_secret_project private
3 agl_progtest -> avl_tree_cpp
4 fit_bachelor_thesis -> bachelor_thesis public
5 fit_bachelor_thesis -> bachelor_thesis_backup private
```

Explanation of each line

1. Specify only the GitLab project name. GitHub name is the same and visibility is default.

```
my_project
```

2. Same as above but explicitly set visibility of exported GitHub project.

```
my_secret_project private
```

3. Set different GitHub name to export to.

```
agl_progtest -> avl_tree_cpp
```

- 4-5. Explicitly add visibility with a different name on GitHub.

```
fit_bachelor_thesis -> bachelor_thesis public
fit_bachelor_thesis -> bachelor_thesis_backup private
```



## CONFIG FILE FORMAT

Config file is required and is passed by `-c`, `--config` option. It must contain GitHub and GitLab tokens. Application uses them for accessing GitHub and GitLab API.

### 3.1 Example

Below is the only valid format of a config file.

```
[gitlab]
token=XXXXXXXXXXXXXXXXXXXXX

[github]
token=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```



## RETURN CODES

After export is finished, user is presented with the result.

Result for each export may contain multiple codes:

1. EXPORTED  
Project has been successfully exported.
2. SKIPPED  
Export for this project has been skipped because it's name already exists in GitHub (only without `overwrite` flag).
3. SUCCESS\_ROLLBACK  
Everything for the project export is undone. Eg deleting created GitHub repo if it did not exist before.
4. ERROR\_ROLLBACK  
Error during rollback.
5. RUN\_ERROR  
Error during run of the export. Does not interfere with `SUCCESS_ROLLBACK`.
6. NOT\_RUNNED  
Export has not been started for this project.
7. OVERWRITTEN  
GitHub project has been overwritten (only with `overwrite` flag).
8. INTERRUPTED  
Run of the export has been interrupted by the user (by sending `Ctrl+C`). Does not interfere with `SUCCESS_ROLLBACK`.
9. DRY\_RUN  
Application was run in `dry-run` mode. No changes were made.
10. NO\_GITLAB\_PROJECT  
There is not GitLab project for the given name.
11. MULTIPLE\_GITLAB\_PROJECTS  
There are multiple GitLab projects for the given name.



## TESTING

Exporter has set of tests that it is supposed to pass. All tests are written using `pytest`.

Command for running the tests is `tox`. Easy way how to run all tests is to run this command in `cloned Exporter repository`.

```
$ tox # run all tests
```





## MANPAGE

Usage: exporter [OPTIONS]

Tool for exporting projects from FIT CTU GitLab to GitHub

## Options:

<code>--version</code>	Show the version and exit.
<code>-c, --config FILENAME</code>	File containing GitHub and GitLab tokens. [required]
<code>--export-all</code>	Export all GitLab projects associated with given token.
<code>-p, --projects FILENAME</code>	Project names to export. See Documentation for format. Option is mutually exclusive with export-all.
<code>--purge-gh</code>	Prompt for GitHub token with admin access, delete all repos and exit. Dangerous!
<code>--debug</code>	Run application in debug mode. Application is unstable in this mode.
<code>--conflict-policy [skip overwrite]</code>	[skip] skip export for project names which already exists on GitHub.[overwrite] overwrite any GitHub project which already exists.
<code>--tmp-dir PATH</code>	Temporary directory to store data during export. [default: tmp]
<code>--task-timeout FLOAT</code>	Timeout for unresponding export task. [default: 30.0]
<code>--unique</code>	Prevent GitHub name conflicts by appending random string at the end of exported project name.
<code>--visibility [public private]</code>	Visibility of the exported project on GitHub [default: private]
<code>--batch-size INTEGER</code>	Maximum count of simultaneously running tasks. [default: 10]

(continues on next page)

(continued from previous page)

<code>--dry-run</code>	Do not perform any changes on GitLab and Github.
<code>--help</code>	Show this message and exit.

## API DOCUMENTATION

### 7.1 Package

#### 7.1.1 Submodules

#### 7.1.2 `exporter.cli` module

#### 7.1.3 `exporter.config` module

#### 7.1.4 `exporter.helpers` module

`exporter.helpers.ensure_tmp_dir` (*path*)

Acquire temporal directory, asking user for overwrite existing if necessary.

**Parameters** `path` – directory to acquire

**Returns** empty temporal directory

`exporter.helpers.flatten` (*t*)

Flatten list of lists

`exporter.helpers.rndstr` (*length*)

Generate random string of given length which contains digits and lowercase ASCII characters

`exporter.helpers.split_to_batches` (*iterable, n=1*)

Split iterable to batched of given size

#### 7.1.5 `exporter.logic` module

#### 7.1.6 Module contents



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### e

`exporter.helpers`, [15](#)





## INDEX

### E

`ensure_tmp_dir()` (*in module `exporter.helpers`*), 15  
`exporter.helpers`  
    module, 15

### F

`flatten()` (*in module `exporter.helpers`*), 15

### M

module  
    `exporter.helpers`, 15

### R

`rndstr()` (*in module `exporter.helpers`*), 15

### S

`split_to_batches()` (*in module `exporter.helpers`*),  
15