

---

# **OctoCheese**

***Release 0.7.0***

## **Copy PyPI Packages to GitHub Releases**

**Dominic Davis-Foster**

**Jan 24, 2024**



# Contents

<b>1</b>	<b>Installing as a Python package and script</b>	<b>3</b>
1.1	from PyPI . . . . .	3
1.2	from GitHub . . . . .	3
<b>2</b>	<b>GitHub Actions Usage</b>	<b>5</b>
2.1	Configuration . . . . .	5
<b>3</b>	<b>Script Usage</b>	<b>7</b>
<b>4</b>	<b>API Reference</b>	<b>9</b>
4.1	octocheese.__main__ . . . . .	9
4.2	octocheese.colours . . . . .	10
4.3	octocheese.core . . . . .	10
<b>5</b>	<b>Contributing</b>	<b>13</b>
5.1	Coding style . . . . .	13
5.2	Automated tests . . . . .	13
5.3	Type Annotations . . . . .	13
5.4	Build documentation locally . . . . .	14
<b>6</b>	<b>Downloading source code</b>	<b>15</b>
6.1	Building from source . . . . .	16
<b>7</b>	<b>License</b>	<b>17</b>
	<b>Python Module Index</b>	<b>21</b>
	<b>Index</b>	<b>23</b>



This is a GitHub action, Python package and command line script that copies distribution files from PyPI (“The Cheese Shop”) and adds them to the relevant tag in GitHub Releases.



## Installing as a Python package and script

### 1.1 from PyPI

```
$ python3 -m pip install octocheese --user
```

### 1.2 from GitHub

```
$ python3 -m pip install git+https://github.com/domdfcoding/octocheese@master --user
```





## GitHub Actions Usage

In a GitHub Workflow:

```
name: "Copy PyPI to Releases"
on:
  - push

jobs:
  releases:
    runs-on: ubuntu-latest
    steps:
      - uses: domdfcoding/octocheese@v0.7.0
        with:
          pypi_name: "domdf_python_tools"
        env:
          GITHUB_TOKEN: ${secrets.GITHUB_TOKEN}
```

### 2.1 Configuration

#### **pypi\_name**

**Type:** str

**Required:** True

The name of the project on PyPI.

The `GITHUB_TOKEN` must also be supplied otherwise the action will fail.



## Script Usage

Invoke using the `octocheese` command or `python3 -m octocheese`.

Copy PyPI Packages to GitHub Releases.

```
octocheese [OPTIONS] PYPI_NAME
```

### Options

- t, --token <token>**  
**Required** The token to authenticate with the GitHub API. Can also be provided via the 'GITHUB\_TOKEN' environment variable.
- r, --repo <repo>**  
The repository name (in the format <username>/<repository>) or the complete GitHub URL.
- n, --max-tags <max\_tags>**  
The maximum number of tags to process, starting with the most recent.  
  
**Default** -1
- T, --traceback**  
Show the full traceback on error.
- no-self-promotion**  
Don't show information about OctoCheese at the bottom of the release message.
- version**  
Show the version and exit.

### Arguments

**PYPI\_NAME**  
Required argument.

### Environment variables

**GITHUB\_TOKEN**  
Provides a default for `-t` / `--token`



## API Reference

### Table of Contents

- *API Reference*
  - `octocheese.__main__`
  - `octocheese.colours`
  - `octocheese.core`

Changed in version 0.3.0: All methods now take and return `github3.py` objects in place of their `PyGithub` counterparts.

### 4.1 `octocheese.__main__`

Entry points when running as a script.

#### Functions:

---

<code>run(github_token, github_username, ..., ...)</code>	Helper function for when running as script or action.
---	---

---

**run** (*github\_token*, *github\_username*, *repo\_name*, *pypi\_name*, *self\_promotion=True*, *max\_tags=-1*)  
Helper function for when running as script or action.

#### Parameters

- **github\_token** (`Secret`) – The token to authenticate with the GitHub API with. See <https://help.github.com/en/github/authenticating-to-github/creating-a-personal-access-token> for instructions on generating a token.
- **github\_username** (`str`) – The username of the GitHub account that owns the repository.
- **repo\_name** (`str`) – The name of the GitHub repository.
- **pypi\_name** (`str`) – The name of the package on PyPI.
- **self\_promotion** – Show information about OctoCheese at the bottom of the release message. Default `True`.
- **max\_tags** (`int`) – The maximum number of tags to process, starting with the most recent. Set to `-1` to process all tags. Default `-1`.

Changed in version 0.1.0: Added the `self_promotion` option.

Changed in version 0.3.0: Added the `max_tags` option.

## 4.2 octocheese.colours

Functions for printing coloured text.

### Functions:

<code>error(text)</code>	Prints the given text in red to stderr.
<code>success(text)</code>	Prints the given text in green to stdout.
<code>warning(text)</code>	Prints the given text in yellow to stderr.

#### **error** (*text*)

Prints the given text in red to stderr.

**Parameters** **text** (*str*) – The text to print.

#### **success** (*text*)

Prints the given text in green to stdout.

**Parameters** **text** (*str*) – The text to print.

#### **warning** (*text*)

Prints the given text in yellow to stderr.

**Parameters** **text** (*str*) – The text to print.

## 4.3 octocheese.core

The main logic of octocheese.

### Functions:

<code>copy_pypi_2_github(g, repo_name, ..., ...)</code>	The main function for OctoCheese.
<code>make_release_message(name, version, release_date)</code>	Create a release message.
<code>update_github_release(repo, tag_name, pypi_name)</code>	Update the given release on GitHub with the new name, message, and files.

**copy\_pypi\_2\_github** (*g*, *repo\_name*, *github\_username*, \*, *changelog*="", *pypi\_name*=None, *self\_promotion*=True, *max\_tags*=- 1, *traceback*=False)

The main function for OctoCheese.

#### **Parameters**

- **g** (*GitHub*)
- **repo\_name** (*str*) – The name of the GitHub repository.
- **github\_username** (*str*) – The username of the GitHub account that owns the repository.
- **changelog** (*str*) – Default ' '.
- **pypi\_name** (*Optional[str]*) – The name of the project on PyPI. Default The value of *repo\_name*.

- **self\_promotion** – Show information about OctoCheese at the bottom of the release message. Default `True`.
- **max\_tags** (`int`) – The maximum number of tags to process, starting with the most recent. Set to `-1` to process all tags. Default `-1`.
- **traceback** (`bool`) – Show the full traceback on error. Default `False`.

Changed in version 0.1.0: Added the `self_promotion` option.

Changed in version 0.3.0:

- Added the optional `max_tags` option.
- Added the optional `traceback` parameter.

**make\_release\_message** (*name, version, release\_date, changelog="", self\_promotion=True*)  
Create a release message.

#### Parameters

- **name** (`str`) – The name of the software.
- **version** (`Union[str, float]`) – The version number of the new release.
- **release\_date** (`date`) – The date of the release.
- **changelog** (`str`) – Optional block of text detailing changes made since the previous release.
- **self\_promotion** – Show information about OctoCheese at the bottom of the release message. Default `True`.

**Return type** `str`

**Returns** The release message.

Changed in version 0.1.0: Added the `self_promotion` option.

**update\_github\_release** (*repo, tag\_name, pypi\_name, changelog="", self\_promotion=True, file\_urls=(), traceback=False*)

Update the given release on GitHub with the new name, message, and files.

#### Parameters

- **repo** (`Repository`)
- **tag\_name** (`str`)
- **pypi\_name** (`str`) – The name of the project on PyPI.
- **changelog** (`str`) – The changelog entry for the release. Default `' '`.
- **self\_promotion** (`bool`) – Show information about OctoCheese at the bottom of the release message. Default `True`.
- **file\_urls** (`Union[Iterable[str], Iterable[FileURL]]`) – The files to download from PyPI and add to the release. Either the files URLs themselves, or mappings giving the URL and its sha256 checksum. Default `()`.
- **traceback** (`bool`) – Show the full traceback on error. Default `False`.

**Return type** `Release`

**Returns** The release, and a list of URLs for the current assets.

Changed in version 0.3.0: Now takes a very different set of parameters to the previous version. Please read the current documentation carefully.



## Contributing

OctoCheese uses `tox` to automate testing and packaging, and `pre-commit` to maintain code quality.

Install `pre-commit` with `pip` and install the git hook:

```
$ python -m pip install pre-commit
$ pre-commit install
```

### 5.1 Coding style

`formate` is used for code formatting.

It can be run manually via `pre-commit`:

```
$ pre-commit run formate -a
```

Or, to run the complete autoformatting suite:

```
$ pre-commit run -a
```

### 5.2 Automated tests

Tests are run with `tox` and `pytest`. To run tests for a specific Python version, such as Python 3.6:

```
$ tox -e py36
```

To run tests for all Python versions, simply run:

```
$ tox
```

### 5.3 Type Annotations

Type annotations are checked using `mypy`. Run `mypy` using `tox`:

```
$ tox -e mypy
```

## 5.4 Build documentation locally

The documentation is powered by Sphinx. A local copy of the documentation can be built with `tox`:

```
$ tox -e docs
```

## Downloading source code

The OctoCheese source code is available on GitHub, and can be accessed from the following URL: <https://github.com/domdfcoding/octocheese>

If you have git installed, you can clone the repository with the following command:

```
$ git clone https://github.com/domdfcoding/octocheese
```

```
Cloning into 'octocheese'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 173 (delta 16), reused 17 (delta 6), pack-reused 126
Receiving objects: 100% (173/173), 126.56 KiB | 678.00 KiB/s, done.
Resolving deltas: 100% (66/66), done.
```

Alternatively, the code can be downloaded in a 'zip' file by clicking:

*Clone or download → Download Zip*

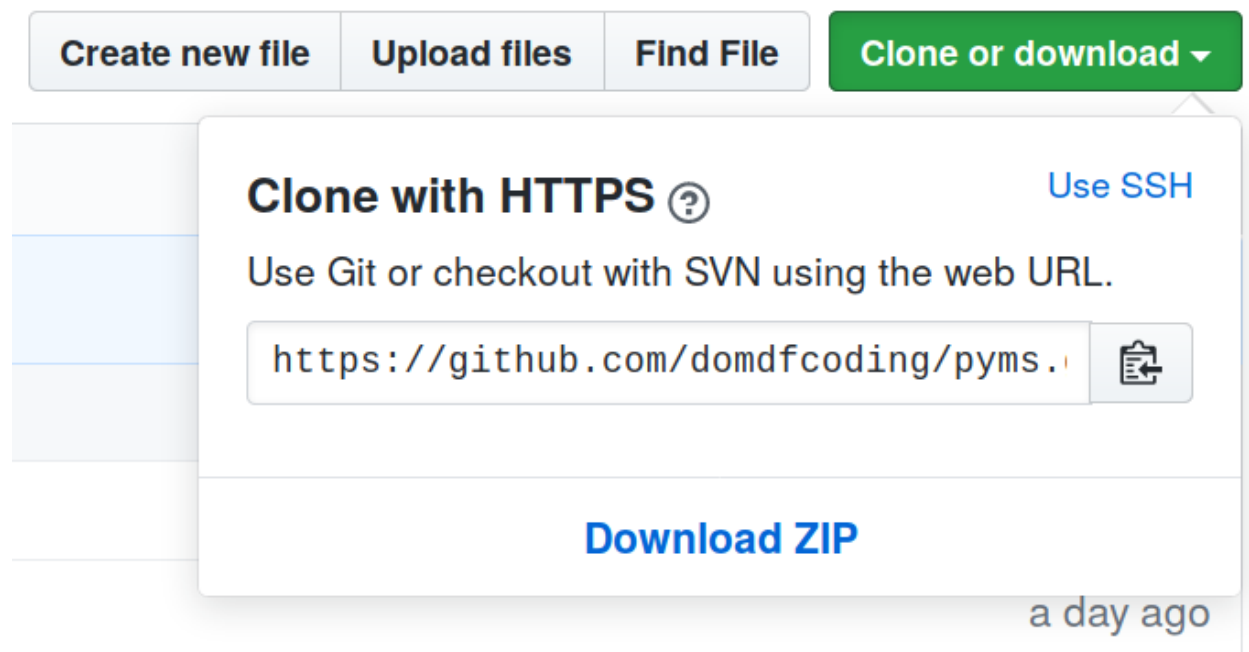


Fig. 1: Downloading a 'zip' file of the source code

## 6.1 Building from source

The recommended way to build OctoCheese is to use `tox`:

```
$ tox -e build
```

The source and wheel distributions will be in the directory `dist`.

If you wish, you may also use `pep517.build` or another **PEP 517**-compatible build tool.

## License

OctoCheese is licensed under the [GNU Lesser General Public License v3.0](#)

---

Permissions of this copyleft license are conditioned on making available complete source code of licensed works and modifications under the same license or the GNU GPLv3. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights. However, a larger work using the licensed work through interfaces provided by the licensed work may be distributed under different terms and without source code for the larger work.

### Permissions

- Commercial use – The licensed material and derivatives may be used for commercial purposes.
- Modification – The licensed material may be modified.
- Distribution – The licensed material may be distributed.
- Patent use – This license provides an express grant of patent rights from contributors.
- Private use – The licensed material may be used and modified in private.

### Conditions

- License and copyright notice – A copy of the license and copyright notice must be included with the licensed material.
- Disclose source – Source code must be made available when the licensed material is distributed.
- State changes – Changes made to the licensed material must be documented.
- Same license (library) – Modifications must be released under the same license when distributing the licensed material. In some cases a similar or related license may be used, or this condition may not apply to works that use the licensed material as a library.

### Limitations

- Liability – This license includes a limitation of liability.
- Warranty – This license explicitly states that it does NOT provide any warranty.

[See more information on choosealicense.com](#) ⇒

---

GNU LESSER GENERAL PUBLIC LICENSE  
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

(continues on next page)

(continued from previous page)

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

#### 0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

#### 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

#### 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

#### 3. Object Code Incorporating Material from Library Header Files.

(continues on next page)

(continued from previous page)

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the object code with a copy of the GNU GPL and this license document.

#### 4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
- c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
- d) Do one of the following:
  - 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
  - 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.
- e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the

(continues on next page)

(continued from previous page)

Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

#### 5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

#### 6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.



## Python Module Index

### O

`octocheese.__main__`, [9](#)  
`octocheese.colours`, [10](#)  
`octocheese.core`, [10](#)



## Symbols

-T  
     octocheese command line option, 7  
 --max-tags <max\_tags>  
     octocheese command line option, 7  
 --no-self-promotion  
     octocheese command line option, 7  
 --repo <repo>  
     octocheese command line option, 7  
 --token <token>  
     octocheese command line option, 7  
 --traceback  
     octocheese command line option, 7  
 --version  
     octocheese command line option, 7  
 -n  
     octocheese command line option, 7  
 -r  
     octocheese command line option, 7  
 -t  
     octocheese command line option, 7

## C

copy\_pypi\_2\_github() (in module  
     octocheese.core), 10

## E

error() (in module octocheese.colours), 10

## G

GNU Lesser General Public License  
     v3.0, 17

## M

make\_release\_message() (in module  
     octocheese.core), 11  
 module  
     octocheese.\_\_main\_\_, 9  
     octocheese.colours, 10  
     octocheese.core, 10

## O

octocheese command line option

-T, 7  
 --max-tags <max\_tags>, 7  
 --no-self-promotion, 7  
 --repo <repo>, 7  
 --token <token>, 7  
 --traceback, 7  
 --version, 7  
 -n, 7  
 -r, 7  
 -t, 7  
 PYPI\_NAME, 7  
 octocheese.\_\_main\_\_  
     module, 9  
 octocheese.colours  
     module, 10  
 octocheese.core  
     module, 10

## P

PYPI\_NAME  
     octocheese command line option, 7  
 pypi\_name (configuration value), 5  
 Python Enhancement Proposals  
     PEP 517, 16

## R

run() (in module octocheese.\_\_main\_\_), 9

## S

success() (in module octocheese.colours), 10

## U

update\_github\_release() (in module  
     octocheese.core), 11

## W

warning() (in module octocheese.colours), 10