
Cookiecutter Python Package

Release 0.0.0

Digimach

Feb 08, 2022

USER’S GUIDE:

1	Indices and tables	3
2	User’s Guide	5
2.1	Quickstart	5
3	Addendums	7
3.1	LICENSE	7
3.2	Contributor Covenant Code of Conduct	7
3.3	CREDITS	9

This project aims to provide a structure for new and existing Python projects along with a guide on how to complete a project to include tests, documentation, console scripts (executables), code coverage and helpful linters.

For new users to Python this project can provide a helpful jump start and a one stop reference to all important basic steps a project needs to be complete. For existing Python users, this project can provide an easy to replicate project structure with features usable right out of the box.

This project is a [Cookiecutter](#) template.

INDICES AND TABLES

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

USER'S GUIDE

2.1 Quickstart

This is a short summary of all the components this project provides along with basic examples.

2.1.1 Usage

1. Install cookiecutter in your Python environment. Ex: `pip install cookiecutter`
2. Use this template project to generate your project cookiecutter <https://github.com/Digimach/cookiecutter-pypackage.git>
3. The *Template Variables* section defines the Cookiecutter variables that will be prompted and used to configure the generated project.
4. If you want to work in a Git repository for the generated project see *Git Repository*.

2.1.2 Template Variables

- `project_title`: The title of the project that is being created.
- `project_slug`: The slug is used to give the Python package an install name. This should be compliant with [PEP-0008: Package and Module Names](#)
- `project_short_description`: Project description that will be added to `setup.py` and published with the package.
- `project_author_email`: The author email address that will be used for license generation and also published with the package in `setup.py`
- `project_author_name`: The author name that will be used for license generation and also published with the package in `setup.py`
- `project_source_url`: HTTP accessible URL that can link to the source of the generated project.
- `project_doc_url`: HTTP accessible URL that can link to the documentation of the generated project.
- `install_requires`: List of Python packages, in CSV form, that are required by the generated project at install.
- `project_extra_dev_requires`: List of Python packages, in CSV form, that are required by the generated project during development.
- `project_extra_test_requires`: List of Python packages, in CSV form, that are required to test the generated project.
- `project_python_max_version`: The maximum version of Python the project will be supported on.

- `project_python_min_version`: The minimum version of Python the project will be supported on. Has to be lesser or equal than `project_python_max_version`.
- `project_valid_python_versions`: List of Python versions, in CSV form, that are valid. This can be used to define the versions in between `project_python_min_version` and `project_python_max_version`, inclusive, that this project will be supported on.
- `initialize_git_repo`: If true, will setup a Git repository in the generated project and do an initial commit. You can amend the initial commit or add on top. See [Git Repository](#) for more information.

2.1.3 Git Repository

If `initialize_git_repo` is yes (the default) a Git repository will be initialized in the target repository. This will also do a commit for you which you can then amend or add on top. This allows you to diff against the templated project with your own work.

You just have to add the remote to push the Git repository:

```
git remote add origin <repo url>
```

If you chose `initialize_git_repo` to be no you can initialize the Git repository in your templated project later by doing:

```
git init
git add --all
git remote add origin <repo url>
git push
```

ADDENDUMS

Design notes, legal information and changelog are here for the interested.

3.1 LICENSE

BSD-3-Clause

Copyright 2020 Digimach and individual contributors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

3.2 Contributor Covenant Code of Conduct

3.2.1 Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

3.2.2 Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

3.2.3 Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

3.2.4 Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

3.2.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at abuse@digimach.com. All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

3.2.6 Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 1.4, available at <https://www.contributor-covenant.org/version/1/4/code-of-conduct.html>

For answers to common questions about this code of conduct, see <https://www.contributor-covenant.org/faq>

3.3 CREDITS

The CREDIT file gives authors and copyright holders the credit for providing inspiration and design ideas. The author wants to acknowledge their contribution but maintains copyright as per the LICENSE file.

3.3.1 Licenses

LICENSES used in the project derivation (“generated project”) except for Proprietary License are templated from the following sources:

1. [Choose A License](#)
2. [Open Source Initiative](#)

Proprietary license was written after consulting:

1. M., S. L. (2004). Chapter 5: Non-Open Source Licenses. In Understanding open source & free software licensing (pp. 114-146). Sebastopol (Calif.): O’Reilly.

3.3.2 Design

Some parts of the design philosophy have been inspired from the following sources:

1. [audreyfeldroy/cookiecutter-pypackage](#)
2. [pallets/flask](#)
3. [django/django](#)