ProjectTemplate Documentation

Release 0.4.0

David R. Smith

Apr 01, 2017

Contents

Contents:

CHAPTER 1

ProjectTemplate

A Python project template with several best-practice features/integrations

- Continuous integration at Travis-CI
  - Coverage for several Python and Numpy versions
  - Coverage for Linux and OS X
- Documentation with `Sphinx`
  - Run `make docs` to generate docs
  - Hosted at Read the Docs
- Testing with `pytest` and `tox`
  - Run `pytest` or `make test` to test in current environment
  - Run `tox` or `make test-all` to test in multiple virtual envs
    - Test matrix covers several Python and Numpy versions
- Quaity checks with `coverage` and `flake8`
  - Run `coverage` or `make coverage` for test coverage report
  - Run `flake8` or `make lint` for code style/quality checks
  - Analysis at Code Climate and QuantifiedCode
- Dependency updates and Python 3 compatability at PyUp.io
- Version management with `bumpversion`
- Makefile recipes
  - Run `make` for recipe summaries

This project is adapted from the Cookiecutter package utility and the PyPackage template.

CHAPTER 2

Installation

Stable release

To install ProjectTemplate, run this command in your terminal:

```
$ pip install project_template
```

This is the preferred method to install ProjectTemplate, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

From sources

The sources for ProjectTemplate can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/drsmith48/project_template
```

Or download the tarball:

```
$ curl  -OL https://github.com/drsmith48/project_template/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER 3

Usage

To use ProjectTemplate in a project:

```
import project_template
```

CHAPTER 4

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

Types of Contributions

Report Bugs

Report bugs at https://github.com/drsmith48/project_template/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

Write Documentation

ProjectTemplate could always use more documentation, whether as part of the official ProjectTemplate docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/drsmith48/project_template/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

Get Started!

Ready to contribute? Here's how to set up *project_template* for local development.

1. Fork the *project_template* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/project_template.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv project_template
$ cd project_template/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 project_template tests
$ python setup.py test or py.test
$ tox
```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/drsmith48/project_template/pull_requests and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

```
$ py.test tests.test_project_template
```

CHAPTER 5

Credits

Development Lead

- David R. Smith <drsmith48@gmail.com>

Contributors

None yet. Why not be the first?

CHAPTER 6

History

0.1.0 (2017-03-27)

- First release on PyPI.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search