pyscaffoldext-cookiecutter Documentation

Release unknown

Anderson Bravalheri

Jul 14, 2019

Contents

Combines the flexibility of **Cookiecutter** templates with the power of **PyScaffold**.

CHAPTER 1

Contents

1.1 pyscaffoldext-cookiecutter

Extension that combines the flexibility of **Cookiecutter** templates with the power of **PyScaffold**.

Cookiecutter is a flexible utility that allows the definition of templates for a diverse range of software projects. On the other hand, PyScaffold is focused in a good out-of-the-box experience for developing distributable Python packages (exclusively). Despite the different objectives, it is possible to combine the power of both tools to create a custom Python project setup.

1.1.1 Quickstart

This extension can be directly installed with `pip`:

```
$ pip install pyscaffoldext-cookiecutter
```

Or, if you prefer `pipx`:

```
$ pipx install pyscaffold  # if you haven't installed pyscaffold yet
$ pipx inject pyscaffold pyscaffoldext-cookiecutter
```

Note that, after the installation, `putup -h` will show a new option `--cookiecutter TEMPLATE`. Use this option to point out which template you want to use (path or url). The file structure created by Cookiecutter will be refined by PyScaffold afterwards. For example:

```
$ putup my-proj1 --cookiecutter ~/my-templates/default
$ putup my-proj2 --cookiecutter gh:something/from-github
```

Please refer to Cookiecutter documentation for more details on possible URLs and abbreviations.

1.1.2 Cookiecutter templates with PyScaffold

The following example shows how to create a new package named `mypkg`, that uses a Cookiecutter template, but is enhanced by PyScaffold's features:

```
$ putup mypkg --cookiecutter gh:pyscaffold/cookiecutter-pypackage
```

This is roughly equivalent to first create a project using the Cookiecutter template and convert it to PyScaffold afterwards:

```
$ cookiecutter --no-input gh:pyscaffold/cookiecutter-pypackage project_name=mypkg
$ putup mypkg --force
```

Note: For complex Cookiecutter templates calling `cookiecutter` and `putup` separately may be a better option, since it is possible to answer specific template questions or at least set values for Cookiecutter variables.

Warning: Although using Cookiecutter templates is a viable solution to customize a project that was set up with PyScaffold, the recommended way is to help improve PyScaffold by contributing an extension.

Suitable templates

Note that PyScaffold will overwrite some files generated by Cookiecutter, like `setup.py`, the `__init__.py` file under the package folder and most of the `docs` folder, in order to provide setuptools_scm and sphinx integration. Therefore not all Cookiecutter templates are suitable for this approach.

Ideally, interoperable templates should focus on the file structure inside the `src` folder instead of packaging or distributing, since PyScaffold already handles it under-the-hood. This also means that your template should adhere to the src-layout if you want to generate files within your Python package.

In addition, PyScaffold runs Cookiecutter with the `--no-input` flag activated and thus the user is not prompted for manual configuration. Instead, PyScaffold injects the following parameters:

```
author
email
project_name
package_name
project_short_description
```

Accordingly, the template file structure should be similar to:

```
cookiecutter-something/
└── {{cookiecutter.project_name}}/
    └── src/
        └── {{cookiecutter.package_name}}/
            └── ...
```

See Cookiecutter for more information about template creation.

Note: PyScaffold uses Cookiecutter only for its ability to create files. Pre/post hooks that perform any other kind of side effect are not guaranteed to work.

1.1.3 Note

This project has been set up using PyScaffold 3.2. For details and usage information on PyScaffold see https://pyscaffold.org/.

1.2 pyscaffoldext

1.2.1 pyscaffoldext package

Subpackages

pyscaffoldext.cookiecutter package

Submodules

pyscaffoldext.cookiecutter.extension module

Module contents

Module contents

1.3 License

1.4 Contributors

- Anderson Bravalheri <andersonbravalheri@gmail.com>

1.5 Changelog

1.5.1 Version 0.1

- Initial release

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

Index

P