flake8-ownership

Release 2.0.2

Joe Joyce

Jul 15, 2019

Contents

1 Usage 1

2 An Apology for the Configuration Syntax 3

3 Changelog 5

4 Contributing 9

5 Internals 11

6 Roadmap 13

Python Module Index 15

Index 17

CHAPTER 1

Usage

flake8-ownership was inspired by flake8-copyright[1] and a personal desire to learn how to write flake8 extensions. It's meant to be a little more powerful than flake8-copyright[2] while not quite as flexible as flake8-regex[3].

flake8-ownership can make sure your codebase has proper `:author:`, `:copyright:`, and `:license:` tags in each file. The required content of those tags is set in the flake8 configuration[4].

For example, the configuration for this project is:

```
[flake8]
author-re = ^Joe Joyce <joe@decafjoe.com>$
copyright-re = ^Copyright \(c\) Joe Joyce and contributors<COMMA> 2016-<YEAR>.$
license-re = ^BSD$
```

This configuration ensures that each file in the project has the following lines:

```
:author: Joe Joyce <joe@decafjoe.com>
:copyright: Copyright (c) Joe Joyce and contributors, 2016-2018.
:license: BSD
```

If any of those lines are missing, it's a violation. If they don't match the regex, it's a violation. *My apologies* (page 3) for the weird `<COMMA>` and `<YEAR>` stuff. Those special strings will be substituted with an actual comma and the current year, respectively. (For all three tags.)

Note that all three settings are optional; if you do not specify any of the `-re` settings, flake8-ownership will not do any checks. If you specify one or two, it will check *only* those one or two.

You may also specify multiple valid author/copyright/license regexes by supplying a comma separated list:

```
[flake8]
author-re =
```

(continues on next page)

[1] https://pypi.python.org/pypi/flake8-copyright
[2] https://pypi.python.org/pypi/flake8-copyright
[3] https://pypi.python.org/pypi/flake8-regex
[4] http://flake8.pycqa.org/en/latest/user/configuration.html

(continued from previous page)

```
  ^Joe Joyce <joe@decafjoe.com>$,
  ^John Everyman <john@example.com>$
```

With that configuration, either of the following lines passes:

```
:author: Joe Joyce <joe@decafjoe.com>
:author: John Everyman <john@example.com>
```

CHAPTER 2

An Apology for the Configuration Syntax

That silly <COMMA> and <YEAR> stuff is ugly, I know. The design requirements were:

1. Must support flexible specification of acceptable author/copyright/license lines. Since this is a programmer product, regexes are a good choice here.
2. Projects may include code from other projects. That code will have author/copyright/license lines that are different than the main project's. So there must be a way to specify multiple acceptable values for each.

The last one tripped things up. Flake8 supports providing a list of values using a comma separated list:

```
[flake8]
some-setting =
  first thing,
  second thing,
  third thing
```

Great, but copyright lines often have commas in them:

```
[flake8]
copyright-re =
  Copyright 2016-2017, Joe Joyce,
  Copyright 2015 by Other Project and its contributers.
```

Whoops! The configuration parser reads this as three different regexes. Maybe we can escape the comma?

```
[flake8]
copyright-re =
 Copyright 2016-2017\, Joe Joyce,
 Copyright 2015 by Other project and its contributors.
```

No dice.

Ok, so we'll have to have a placeholder and do some interpolation. Python format strings?

```
[flake8]
copyright-re =
 Copyright 2016-2017%(comma)s Joe Joyce,
 Copyright 2015 by Other Project and its contributers.
```

Bzzzt. The config parser tries to interpolate this value from the config file itself. And we don't control the config parsing behavior so we can't tell it not to do that.

And that's why I ended up with the crappy, where-the-heck-did-that-come-from syntax that flake8-ownership uses:

```
[flake8]
copyright-re =
 Copyright 2016-2017<COMMA> Joe Joyce,
 Copyright 2015 by Other Project and its contributers.
```

The config parser doesn't do anything clever with this, and flake8-ownership can replace `<COMMA>` with `,` when it processes the config values.

CHAPTER 3

Changelog

3.1 2.0.1

- Fixes errors when options are undefined or `None` (from Sam Bull of Kopernio)

3.2 2.0.0

- Drops support for Python 2.6 and Python 3.3

3.3 1.1.1

- Fixes typo-ed reference to project page in package metadata

3.4 1.1.0

- Moves project to GitHub
- Integrates with Travis CI
- Documentation updates
- Documentation published on Read The Docs

3.5 1.0.2

- Fixes embarrassing bug that was causing this extension to do essentially nothing.

3.6 1.0.1

- Fixes specification of the `flake8.extension` entry point.

3.7 1.0.0

- Bumps version to 1.0.0.

3.8 0.10.3

- Links the issue tracker in the README.

3.9 0.10.2

- Fixed packaging bug related to changes in 0.10.1.

3.10 0.10.1

- Cleaned up packaging (include CHANGELOG and LICENSE in package, do not include redundant README.rst).

3.11 0.10.0

- Formalized and documented Python 3 and PyPy support.

3.12 0.9.4

- Fixed copyright on this project.

3.13 0.9.3

- Added this changelog.

3.14 0.9.2

- Initial public release.

3.15 Pre-0.9.2

Initial releases and testing of the release tooling.

CHAPTER 4

Contributing

There shouldn't be much to contribute. This is a pretty small, simple project that doesn't aim to be any more ambitious than it already is. If you run into bugs, please file them in the project's issue tracker on GitHub. Code and documentation patches are welcome, but please file a feature request before spending much time, so we can make sure the change is in line with the goals of the project.

That said, here's the info about the development environment.

4.1 Prerequisites

Requirements:

- Common development tools like git, make, a C compiler, etc
- Python 3.6 – this is the "main" interpreter used for development
- Virtualenv

Recommendations:

- All supported Python interpreters
  - Python 2.7
  - Python 3.4
  - Python 3.5
  - Python 3.6
  - PyPy
  - PyPy 3

Suggestions:

- LaTeX – for building the documentation as a PDF

4.2 Setup

The repository can be cloned anywhere you like on your local machine. At any time, you can delete the entire project and its environment by `rm -rf` -ing the local directory.

The following instructions clone the repository to `~/flake8-ownership`:

```
cd
git clone https://github.com/decafjoe/flake8-ownership.git
cd flake8-ownership
make env
```

Wait ~10m and you should be good to go!

Note that all dependencies are installed underneath the repository directory (take a peek at `.env/`). To delete the development environment artifacts, you can run `make pristine` (see below). To delete everything, simply `rm -rf` the clone.

4.3 Tooling

flake8-ownership's developer tooling is exposed via `make`. Run `make` with no arguments to get a list of available targets. All targets except `make release` are idempotent, so they can be run at any time.

Environment:

- `make env` installs the development environment; subsequent runs update the environment if required
- `make clean` deletes build artifacts like .pyc files, sdists, etc
- `make pristine` kills the local development environment
- `make check-update` checks for updates to Python packages installed in the development environment

Documentation:

- `make html` generates the HTML documentation to `doc/_build/html/`
- `make pdf` generates the PDF documentation to `doc/_build/latex/clik.pdf`
- `make docs` builds both HTML and PDF documentation to their respective locations

Build:

- `make dist` builds a sdist into `dist/`
- `make release` builds a clean sdist, uploads it to PyPI, tags the commit with the current version number, bumps the version, then commits the new version number and pushes it up to GitHub (this is largely implemented by the `tool/pre-release` and `tool/post-release` scripts)

QA:

- `make lint` runs the Flake8 linter on the Python files in the project
- `make test` runs the test suite against the "main" development interpreter
- `make test-all` runs the linter, runs the test suite against all supported interpreters, and generates a coverage report to `coverage/`

CHAPTER 5

Internals

5.1 Extension

Flake8 extension for checking author, copyright, and license.

author Joe Joyce <joe@decafjoe.com>

copyright Copyright (c) Joe Joyce and contributors, 2016-2019.

license BSD

`flake8_ownership.__version__ = '2.0.2'`
Version of the extension.

Type `str`[5]

`flake8_ownership.author_re`
Regex that matches the `:author:` line.

Type `re`[6]

`flake8_ownership.copyright_re`
Regex that matches the `:copyright:` line.

Type `re`[7]

`flake8_ownership.license_re`
Regex that matches the `:license:` line.

Type `re`[8]

class `flake8_ownership.Checker`(*tree*, *filename*)
Flake8 checker class that checks for author, copyright, and license.

[5] https://docs.python.org/3/library/stdtypes.html#str
[6] https://docs.python.org/3/library/re.html#re.compile
[7] https://docs.python.org/3/library/re.html#re.compile
[8] https://docs.python.org/3/library/re.html#re.compile

classmethod add_options(*parser*)
Add –author-re, –copyright-re, and –license-re options.

author_re = None
List of regexes of valid :author: values.

Type `list`[9] of `re`[10] instances.

codes = 'O10'
Prefix for the error codes emitted from this class.

Type `str`[11]

copyright_re = None
List of regexes of valid :copyright: values.

Type `list`[12] of `re`[13] instances.

license_re = None
List of regexes of valid :license: values.

Type `list`[14] of `re`[15] instances.

name = 'flake8_ownership'
Name of the checker.

Type `str`[16]

classmethod parse_options(*options*)
Process the supplied configuration.

This populates the *author_re* (page 11), *copyright_re* (page 11), and *license_re* (page 11) attributes. For each configuration option, this substitutes `<COMMA>` and `<YEAR>` as appropriate, then compiles each regex.

run()
Run the *Checker* (page 11) on a `filename`.

version = '2.0.2'
Version of the checker.

Type `str`[17]

5.2 Tests

[9] https://docs.python.org/3/library/stdtypes.html#list
[10] https://docs.python.org/3/library/re.html#module-re
[11] https://docs.python.org/3/library/stdtypes.html#str
[12] https://docs.python.org/3/library/stdtypes.html#list
[13] https://docs.python.org/3/library/re.html#module-re
[14] https://docs.python.org/3/library/stdtypes.html#list
[15] https://docs.python.org/3/library/re.html#module-re
[16] https://docs.python.org/3/library/stdtypes.html#str
[17] https://docs.python.org/3/library/stdtypes.html#str

CHAPTER 6

Roadmap

6.1 Support Flake8 < 3.0?

I have no personal reason to do this, but if there are other people that are stuck on older versions, it shouldn't be too hard to use flake8-polyfill to make flake8-ownership work for those folks. The required changes to testing might be a little tricky though.

- genindex
- modindex
- search

Python Module Index

f

flake8_ownership, 11

Index

Symbols

`__version__` (*in module flake8_ownership*), 11

A

`add_options()` (*flake8_ownership.Checker class method*), 11
`author_re` (*flake8_ownership.Checker attribute*), 12
`author_re` (*in module flake8_ownership*), 11

C

`Checker` (*class in flake8_ownership*), 11
`codes` (*flake8_ownership.Checker attribute*), 12
`copyright_re` (*flake8_ownership.Checker attribute*), 12
`copyright_re` (*in module flake8_ownership*), 11

F

`flake8_ownership` (*module*), 11

L

`license_re` (*flake8_ownership.Checker attribute*), 12
`license_re` (*in module flake8_ownership*), 11

N

`name` (*flake8_ownership.Checker attribute*), 12

P

`parse_options()` (*flake8_ownership.Checker class method*), 12

R

`run()` (*flake8_ownership.Checker method*), 12

V

`version` (*flake8_ownership.Checker attribute*), 12