
flake2lint

Release 0.4.3

Tool and pre-commit hook to augment Flake8 noqa comments with PyLint comments.

Dominic Davis-Foster

Apr 26, 2024

Contents

1	Installation	1
1.1	from PyPI	1
1.2	from GitHub	1
2	Usage	3
2.1	flake2lint	3
2.2	pre-commit hook	3
2.3	Supported Flake8 Codes	4
2.4	Example	4
3	Public API	5
3.1	find_noga	5
3.2	find_pylint_disable	5
3.3	process_file	5
4	Contributing	7
4.1	Coding style	7
4.2	Automated tests	7
4.3	Type Annotations	7
4.4	Build documentation locally	8
5	Downloading source code	9
5.1	Building from source	10
6	License	11
	Python Module Index	13
	Index	15

Installation

1.1 from PyPI

```
$ python3 -m pip install flake2lint --user
```

1.2 from GitHub

```
$ python3 -m pip install git+https://github.com/domdfcoding/flake2lint@master --user
```


Usage

2.1 flake2lint

Augment Flake8 noqa comments with PyLint comments.

```
flake2lint [OPTIONS] [FILENAMES]...
```

Options

- v, --verbose**
Show verbose output.
- r, --recursive**
Permits the use of the pattern `***` to match any files, directories and subdirectories.

Arguments

FILENAMES

Optional argument(s). Default None

Changed in version 0.3.0: Added the `-v / --verbose` option.

2.2 pre-commit hook

Sample `.pre-commit-config.yaml`:

```
- repo: https://github.com/domdfcoding/flake2lint
  rev: v0.4.0
  hooks:
    - id: flake2lint
```

See [pre-commit](#) for further information.

2.3 Supported Flake8 Codes

flake2lint currently augments the following flake8 codes:

- A001 → redefined-builtin
New in version 0.4.0.
- A002 → redefined-builtin
- A003 → redefined-builtin

Contributions to add support for more codes are more than welcome. The relevant code is [here](#).

2.4 Example

Before:

```
class FancyDialog(wx.Dialog):  
  
    def __init__(  
        self,  
        parent,  
        id=wx.ID_ANY, # noqa: A002  
        title="My Fancy Dialog",  
        pos=wx.DefaultPosition,  
        size=wx.DefaultSize,  
        style=wx.DEFAULT_DIALOG_STYLE,  
        name=wx.DialogNameStr,  
        data=None  
    ): ...
```

After:

```
class FancyDialog(wx.Dialog):  
  
    def __init__(  
        self,  
        parent,  
        id=wx.ID_ANY, # noqa: A002 # pylint: disable=redefined-builtin  
        title="My Fancy Dialog",  
        pos=wx.DefaultPosition,  
        size=wx.DefaultSize,  
        style=wx.DEFAULT_DIALOG_STYLE,  
        name=wx.DialogNameStr,  
        data=None  
    ): ...
```


Public API

pre-commit hook to augment Flake8 noqa comments with PyLint comments.

Functions:

<code>find_noqa(physical_line)</code>	Search a string for # noqa: ... comments.
<code>find_pylint_disable(physical_line)</code>	Search a string for # pylint: disable=... comments.
<code>process_file(filename)</code>	Augment Flake8 noqa comments with PyLint comments in the given file.

find_noqa (*physical_line*)

Search a string for # noqa: ... comments.

Parameters `physical_line` (`str`)

Return type `Optional[Match[str]]`

find_pylint_disable (*physical_line*)

Search a string for # pylint: disable=... comments.

Parameters `physical_line` (`str`)

Return type `Optional[Match[str]]`

process_file (*filename*)

Augment Flake8 noqa comments with PyLint comments in the given file.

Parameters `filename` (`Union[str, Path, PathLike]`)

Return type `bool`

Returns `True` if the file contents were changed. `False` otherwise.

Contributing

`flake2lint` 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
```

4.1 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
```

4.2 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
```

4.3 Type Annotations

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

```
$ tox -e mypy
```

4.4 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 `flake2lint` source code is available on GitHub, and can be accessed from the following URL: <https://github.com/domdfcoding/flake2lint>

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

```
$ git clone https://github.com/domdfcoding/flake2lint
```

```
Cloning into 'flake2lint'...
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

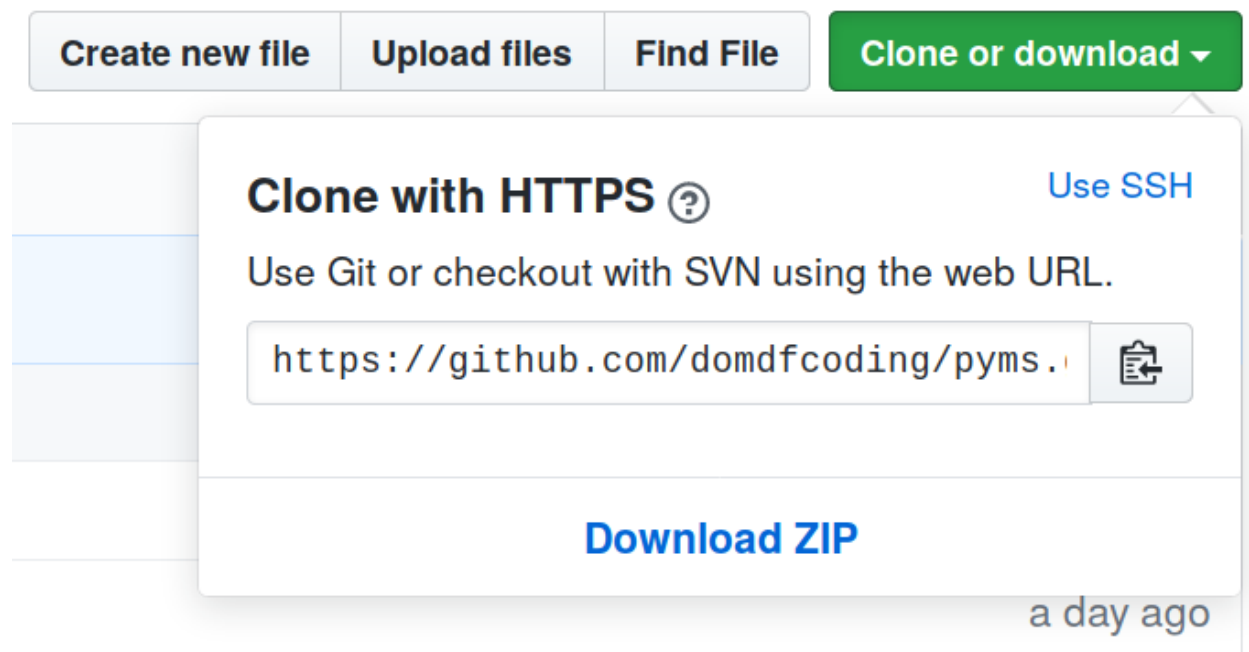


Fig. 1: Downloading a ‘zip’ file of the source code

5.1 Building from source

The recommended way to build `flake2lint` 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

flake2lint is licensed under the [MIT License](#)

A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

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.
- 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.

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](#) ⇒

```
Copyright (c) 2021 Dominic Davis-Foster
```

```
Permission is hereby granted, free of charge, to any person obtaining a copy  
of this software and associated documentation files (the "Software"), to deal  
in the Software without restriction, including without limitation the rights  
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and to permit persons to whom the Software is  
furnished to do so, subject to the following conditions:
```

```
The above copyright notice and this permission notice shall be included in all  
copies or substantial portions of the Software.
```

```
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,  
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF  
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,  
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR  
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE  
OR OTHER DEALINGS IN THE SOFTWARE.
```


Python Module Index

f

`flake2lint`, 5

Symbols

- `--recursive`
 - `flake2lint` command line option, 3
- `--verbose`
 - `flake2lint` command line option, 3
- `-r`
 - `flake2lint` command line option, 3
- `-v`
 - `flake2lint` command line option, 3

F

- `FILENAMES`
 - `flake2lint` command line option, 3
- `find_noqa()` (*in module `flake2lint`*), 5
- `find_pylint_disable()` (*in module `flake2lint`*), 5
- `flake2lint`
 - module, 5
- `flake2lint` command line option
 - `--recursive`, 3
 - `--verbose`, 3
 - `-r`, 3
 - `-v`, 3
 - `FILENAMES`, 3

M

- MIT License, 11
- module
 - `flake2lint`, 5

P

- `process_file()` (*in module `flake2lint`*), 5
- Python Enhancement Proposals
 - PEP 517, 10