
flake8-helper

Release 0.2.2

A helper library for Flake8 plugins.

Dominic Davis-Foster

Nov 22, 2023

Contents

1	Installation	1
1.1	from PyPI	1
1.2	from Anaconda	1
1.3	from GitHub	1
2	flake8_helper	3
2.1	_V	3
2.2	Visitor	3
2.3	_P	4
2.4	Plugin	4
3	Downloading source code	5
3.1	Building from source	6
4	License	7
	Python Module Index	9
	Index	11

Installation

1.1 from PyPI

```
$ python3 -m pip install flake8-helper --user
```

1.2 from Anaconda

First add the required channels

```
$ conda config --add channels https://conda.anaconda.org/conda-forge  
$ conda config --add channels https://conda.anaconda.org/domdfcoding
```

Then install

```
$ conda install flake8-helper
```

1.3 from GitHub

```
$ python3 -m pip install git+https://github.com/python-formate/flake8-helper@master --user
```


flake8_helper

A helper library for Flake8 plugins.

Data:

<code>_V</code>	Invariant <code>TypeVar</code> bound to <code>flake8_helper.Visitor</code> .
<code>_P</code>	Invariant <code>TypeVar</code> bound to <code>flake8_helper.Plugin</code> .

Classes:

<code>Visitor()</code>	AST node visitor.
<code>Plugin(tree)</code>	Abstract base class for Flake8 plugins.

```
_V = TypeVar(_V, bound=Visitor)  
Type: TypeVar  
Invariant TypeVar bound to flake8_helper.Visitor.
```

```
class Visitor  
    Bases: NodeVisitor  
    AST node visitor.
```

Attributes:

<code>errors</code>	The list of Flake8 errors identified by the visitor.
---------------------	--

Methods:

<code>report_error(node, error)</code>	Report an error for the given node.
--	-------------------------------------

```
errors  
Type: List[Tuple[int, int, str]]  
The list of Flake8 errors identified by the visitor.
```

```
report_error (node, error)  
Report an error for the given node.
```

Parameters

- `node` (`AST`)
- `error` (`str`)

```
_P = TypeVar(_P, bound=Plugin)
```

Type: `TypeVar`

Invariant `TypeVar` bound to `flake8_helper.Plugin`.

```
class Plugin(tree)
```

Bases: `ABC`, `Generic`

Abstract base class for Flake8 plugins.

Parameters `tree` (`AST`) – The abstract syntax tree (AST) to check.

Minimum example:

Attributes:

<code>name</code>	The plugin name.
<code>version</code>	The plugin version.
<code>visitor_class</code>	The visitor class to use to traverse the AST.

Methods:

<code>run()</code>	Traverse the Abstract Syntax Tree and identify errors.
--------------------	--

```
abstract property name
```

The plugin name.

Return type `str`

```
abstract property version
```

The plugin version.

Return type `str`

```
abstract property visitor_class
```

The visitor class to use to traverse the AST.

Return type `Type[~_V]`

```
run()
```

Traverse the Abstract Syntax Tree and identify errors.

Yields a tuple of (line number, column offset, error message, type(self))

Return type `Iterator[Tuple[int, int, str, Type[~_P]]]`

Downloading source code

The `flake8-helper` source code is available on GitHub, and can be accessed from the following URL: <https://github.com/python-formate/flake8-helper>

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

```
$ git clone https://github.com/python-formate/flake8-helper
```

```
Cloning into 'flake8-helper'...
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

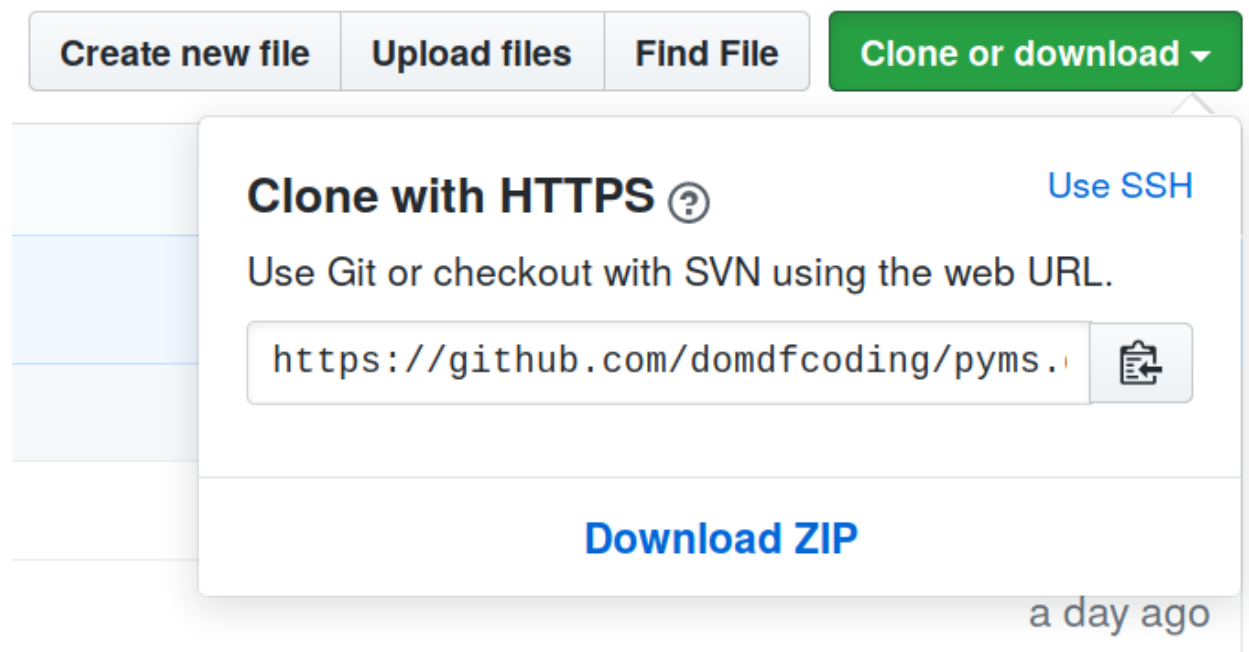


Fig. 1: Downloading a 'zip' file of the source code

3.1 Building from source

The recommended way to build `flake8-helper` 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

flake8-helper 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

flake8_helper, 3

Symbols

`_P` (in module *flake8_helper*), 3

`_V` (in module *flake8_helper*), 3

E

`errors` (*Visitor attribute*), 3

F

`flake8_helper`
module, 3

M

MIT License, 7

module
 flake8_helper, 3

N

`name()` (*Plugin property*), 4

P

`Plugin` (*class in flake8_helper*), 4

Python Enhancement Proposals
 PEP 517, 6

R

`report_error()` (*Visitor method*), 3

`run()` (*Plugin method*), 4

V

`version()` (*Plugin property*), 4

`Visitor` (*class in flake8_helper*), 3

`visitor_class()` (*Plugin property*), 4