flake8-helper

Release 0.2.2

A helper library for Flake8 plugins.

Dominic Davis-Foster

Nov 22, 2023

Contents

Chapter

ONE

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
```

Chapter

TWO

flake8_helper

A helper library for Flake8 plugins.

Data:

_V	Invariant TypeVar bound to *flake8_helper.Visitor*.
_P	Invariant TypeVar bound to *flake8_helper.Plugin*.

Classes:

Visitor()	AST node visitor.
Plugin(tree)	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:

errors	The list of Flake8 errors identified by the visitor.

Methods:

report_error(node, error)	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:

name	The plugin name.
version	The plugin version.
visitor_class	The visitor class to use to traverse the AST.

Methods:

run()	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]]]

Chapter

THREE

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

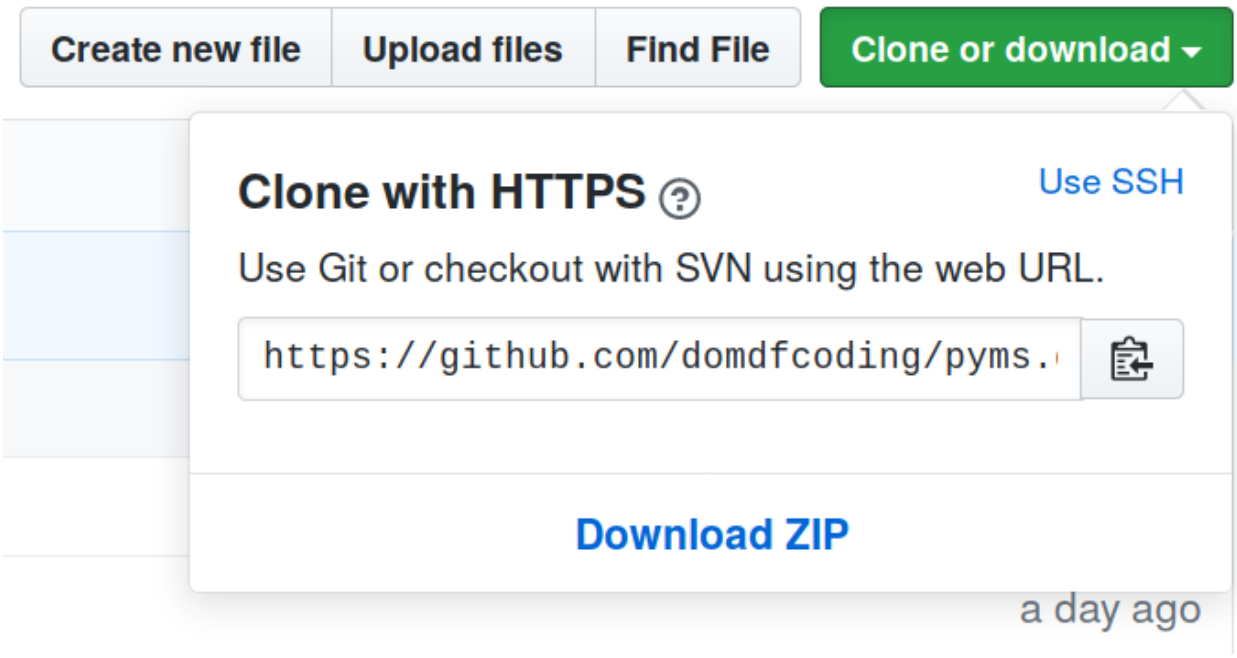

Fig. 1: Downloading a 'zip' file of the source code

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.

Chapter

FOUR

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 ⇒

Python Module Index

f

Index

Symbols

E

F

M

N

P

R

V