flake8-sphinx-links

Release 0.2.1

Dominic Davis-Foster

Mar 18, 2021

CONTENTS

A Flake8 plugin to check docstrings for double backticked strings which should be links to the Python documentation.

CHAPTER

ONE

INSTALLATION

1.1 from PyPI

```
$ python3 -m pip install flake8_sphinx_links --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_sphinx_links
```

1.3 from GitHub

```
$ python3 -m pip install git+https://github.com/domdfcoding/flake8-sphinx-links@master --user
```

CHAPTER

TWO

DOCUMENTATION

2.1 Usage

This library provides the Flake8 plugin `flake8-sphinx-links` to check docstrings for double backticked strings which should be links to the Python documentation.

For example, ``True`` should be py:obj:`True`, which Sphinx will render as a link to the Python documentation. See *Examples* for further examples.

reStructuredText .rst files are not currently checked.

2.1.1 Flake8 codes

Code	Description
SXL001	Double backticked strings should be a link to Python documentation.

2.1.2 Pre-commit hook

`flake8-sphinx-links` can also be used as a `pre-commit` hook See pre-commit for instructions

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

```
- repo: https://gitlab.com/pycqa/flake8
  rev: 3.8.4
  hooks:
  - id: flake8
    additional_dependencies:
    - flake8-sphinx-links==0.2.1
```

2.2 Examples

``True`` => :py:obj:`True`

`True` => True

``False`` => :py:obj:`False`

`False` => `False`

``None`` => :py:obj:`None`

`None` => `None`

``NotImplemented`` => :py:obj:`NotImplemented`

`NotImplemented` => `NotImplemented`

``Ellipsis`` => :py:obj:`Ellipsis`

`Ellipsis` => `Ellipsis`

``__debug__`` => :py:obj:`__debug__`

`__debug__` => `__debug__`

``quit`` => :py:obj:`quit`

`quit` => `quit`

``exit`` => :py:obj:`exit`

`exit` => `exit`

``copyright`` => :py:obj:`python:copyright`

`copyright` => copyright

``credits`` => :py:obj:`credits`

`credits` => credits

``license`` => :py:obj:`license`

`license` => license

``ValueError`` => :exc:`ValueError`

`ValueError` => ValueError

``BaseException`` => :exc:`BaseException`

`BaseException` => BaseException

``ValueError`` => :exc:`ValueError`

`ValueError` => ValueError

``int`` => :class:`int`

`int` => int

``str`` => :class:`str`

`str` => `str`

2.3 API Reference

A Flake8 plugin to check docstrings for double backticked strings which should be links to the Python documentation.

Classes:

Plugin(tree)	Flake8 plugin to check docstrings for double backticked strings which should be links to the Python documentation.
Visitor()	AST visitor to check docstrings for double backticked strings which should be links to the Python documentation.

Data:

exc	List of keywords which should become :py:exc:`<keyword>`
py_obj	List of keywords which should become :py:obj:`<keyword>`
py_obj_python	List of keywords that should become :py:obj:`python:<keyword>` to prevent conflict with Sphinx objects.
regex	Regex to match keywords that should be Sphinx links.

class Plugin(*tree*)

Bases: Plugin[*Visitor*]

Flake8 plugin to check docstrings for double backticked strings which should be links to the Python documentation.

Attributes:

name
version

Classes:

visitor_class	alias of *Visitor*

name: str = 'flake8_sphinx_links'

version: str = '0.2.1'

visitor_class

alias of *Visitor*

class Visitor

Bases: Visitor

AST visitor to check docstrings for double backticked strings which should be links to the Python documenta-

tion.

Methods:

visit_ClassDef(node)	
visit_FunctionDef(node)	
visit_Module(node)	

visit_ClassDef(*node*)

visit_FunctionDef(*node*)

visit_Module(*node*)

exc = ['BaseException', 'Exception', 'ArithmeticError', 'BufferError', 'LookupError', 'Asse

Type: List[str]

List of keywords which should become :py:exc:`<keyword>`

py_obj = ['True', 'False', 'None', 'NotImplemented', 'Ellipsis', '__debug__', 'quit', 'exit

Type: List[str]

List of keywords which should become :py:obj:`<keyword>`

py_obj_python = ['copyright']

Type: List[str]

List of keywords that should become :py:obj:`python:<keyword>` to prevent conflict with Sphinx objects.

regex

Type: Pattern

Regex to match keywords that should be Sphinx links.

Pattern	(``)(True|False|None|NotImplemented|Ellipsis|__debug__|quit|exit|credits|license|

CHAPTER

THREE

CONTRIBUTING

3.1 Overview

`flake8_sphinx_links` 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
```

3.2 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
```

3.3 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
```

3.4 Type Annotations

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

```
$ tox -e mypy
```

3.5 Build documentation locally

The documentation is powered by Sphinx. A local copy of the documentation can be built with `tox`:

```
$ tox -e docs
```

3.6 Downloading source code

The `flake8_sphinx_links` source code is available on GitHub, and can be accessed from the following URL: https://github.com/domdfcoding/flake8-sphinx-links

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

```
$ git clone https://github.com/domdfcoding/flake8-sphinx-links"
> Cloning into 'flake8-sphinx-links'...
> 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

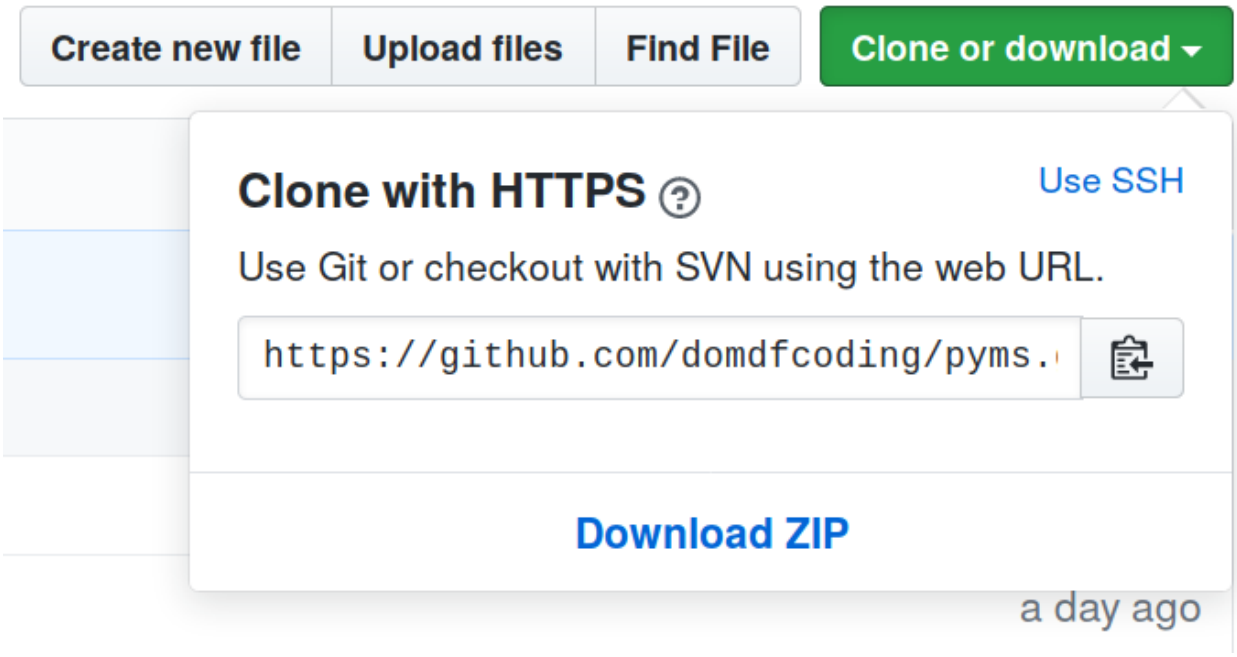

Fig. 1: Downloading a 'zip' file of the source code

3.6.1 Building from source

The recommended way to build `flake8_sphinx_links` 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.

PYTHON MODULE INDEX

f

PYTHON MODULE INDEX

f

INDEX