
git-pylint-commit-hook Documentation

Release 2.0.0

Sebastian Dahlgren

Oct 24, 2018

Contents

1	Installation	3
2	Usage	5
2.1	Basic configuration	5
2.2	Command line options	5
2.3	Support for <code>.pylintrc</code> files	6
3	License	7
4	Overview	9
5	Author	11
6	Source code	13
7	Indices and tables	15

Contents:

CHAPTER 1

Installation

Use `pip` to install `git-pylint-commit-hook`.

```
pip install git-pylint-commit-hook
```

The command `git-pylint-commit-hook` will now be accessible in your system.

2.1 Basic configuration

To use `git-pylint-commit-hook` in a project, create a new file under `/project/root/.git/hooks/pre-commit` and add the following to that file:

```
#!/usr/bin/env bash
git-pylint-commit-hook
```

Save the file and make it executable:

```
chmod +x .git/hooks/pre-commit
```

Your Python files should now be checked upon commit.

2.2 Command line options

The `git-pylint-commit-hook` can be configured using command line options. The command line options are:

<code>--limit LIMIT</code>	Score limit, files with a lower score will stop the commit. Default: 8.0
<code>--pylint PYLINT</code>	Path to pylint executable. Default: <code>pylint</code>
<code>--pylintrc PYLINTRC</code>	Path to pylintrc file. Options in the pylintrc will override the command line parameters. Default: <code>pylintsearch order</code>
<code>--pylint-params PYLINT_PARAMS</code>	Custom pylint parameters to add to the pylint command
<code>--suppress-report</code>	Suppress report output if pylint fails
<code>--always-show-violations</code>	Show violations in case of pass as well
<code>--version</code>	Print current version number
<code>--ignore IGNORED_FILES</code>	

(continues on next page)

(continued from previous page)

<code>--stash</code>	Add regex to blacklist files or directories, allowing to avoid running pylint those files. Stash any unstaged changes while linting (changes are unstashed automatically unless the process is forcibly killed)
----------------------	--

You can simply append those to the command created in the **Basic configuration** above.

2.3 Support for `.pylintrc` files

`git-pylint-commit-hook` will automatically find your pylint configuration files, according to the pylint configuration file default read order. Any configuration found for the project will be used in the `git-pylint-commit-hook`.

You can also define a custom pylint configuration file using the `--pylintrc` command line option.

2.3.1 pylint configuration

Settings are loaded by default from the `.pylintrc` file in the root of your repo.

```
[pre-commit-hook]
command=custom_pylint
params=--rcfile=/path/to/another/pylint.rc
limit=8.0
```

`command` is for the actual command, for instance if pylint is not installed globally, but is in a virtualenv inside the project itself.

`params` lets you pass custom parameters to pylint

`limit` is the lowest value which you want to allow for a pylint score. Any lower than this, and the script will fail and won't commit.

Any of these can be bypassed directly in the pre-commit hook itself. You can also set a different default place to look for the `pylintrc` file.

CHAPTER 3

License

APACHE LICENSE 2.0 Copyright 2013 Sebastian Dahlgren

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

CHAPTER 4

Overview

The `git-pylint-commit-hook` is a tool for checking Python code using `pylint` before the code is committed to git.

All files that end with `.py` or that have `python` in the she bang will be automatically checked before they are committed to the repo.

CHAPTER 5

Author

This project is maintained by [Sebastian Dahlgren](#) ([GitHub](#) | [Twitter](#) | [LinkedIn](#))

CHAPTER 6

Source code

The source code is available on the projects [GitHub repository](#).

CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`