
django-smelly-tokens

Release 0.4.4

Jun 06, 2018

Contents:

1	Django Smelly Tokens	1
1.1	Installation	1
1.2	Usage	1
1.3	Exceptions	1
1.4	Adding your own tokens	2
1.5	References	2
2	Changelog	3
2.1	0.4.4 (2018-06-06)	3
2.2	0.4 (2017-05-06)	3
2.3	0.3 (2017-05-02)	3
3	Indices and tables	5

CHAPTER 1

Django Smelly Tokens

It is sometimes possible to tell that code stinks even without looking at it. This Django app tries to accomplish that.

1.1 Installation

```
pip install django-smelly-tokens
```

1.2 Usage

In your `settings.py` define `SHELLY_TOKENS_APPLICATIONS` list with packages you want to inspect.

In a package with code quality tests (e.g. `test_tokens.py`) import tests you want to check your apps against:

```
from smelly_tokens.test_smelly_tokens import (
    EvalTokenTestCase,
    PdbTokenTestCase,
)
```

Run `django-admin.py test` or `./manage.py test` or `py.test` or `nose` or whatever runner you're using.

1.3 Exceptions

To silence known errors PEP8-style `noqa` comment can be used in the beginning of a file:

```
# smelly_tokens: noqa
```

Or in-line:

```
eval('print 123') # noqa
```

To exclude an entire directory add its path to `SHELLY_TOKENS_EXCLUDE_DIRS` list in settings.

1.4 Adding your own tokens

To create a new type of smelly token test case, inherit `SmellyTokensTestCase` and override `_tokens` list:

```
from django.test import TestCase
from smelly_tokens.test_smelly_tokens import SmellyTokensTestCase

class OOPTokensTestCase(SmellyTokensTestCase, TestCase):
    """ OOP hater. """
    _tokens = ['class', 'object', 'Object']
```

1.5 References

- [Full documentation](#);

2.1 0.4.4 (2018-06-06)

- #movingtogitlab

2.2 0.4 (2017-05-06)

- Introduced zest.releaser.
- Released to PyPI.
- Documentation.

2.3 0.3 (2017-05-02)

- Python 2 / 3 compatibility.
- Multiple tokens per class.

CHAPTER 3

Indices and tables

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