dj-spam Documentation

Release 0.2.0

Daniel Roy Greenfeld

December 18, 2016

Contents

1 Work in Progress: dj-spam **3**
- 1.1 Documentation . . . 3
- 1.2 Features . . . 3
- 1.3 Quickstart . . . 3
- 1.4 admin . . . 4
- 1.5 emailing managers . . . 4
- 1.6 Running tests locally . . . 4

2 Installation **5**

3 Usage **7**

4 Contributing **9**
- 4.1 Types of Contributions . . . 9
- 4.2 Get Started! . . . 10
- 4.3 Pull Request Guidelines . . . 10
- 4.4 Tips . . . 11

5 Credits **13**
- 5.1 Development Lead . . . 13
- 5.2 Contributors . . . 13

6 History **15**
- 6.1 0.3.0 ??? . . . 15
- 6.2 0.2.0 (2-15-07-29) . . . 15
- 6.3 0.1.0 (2-15-07-28) . . . 15

Contents:

CHAPTER 1

Work in Progress: dj-spam

Django + Flagging Spam Made Easy

1.1 Documentation

The full documentation is at https://dj-spam.readthedocs.io.

1.2 Features

- For Django 1.8+
- For Python 2.7/3.3+
- Direct foreign key from the model to the spam report. Avoiding content types and using explicit foreign keys makes for less kludgy databases.
- Powered by conventions used all over Django:
  - Have the appropriate `__str__()` or `__unicode__()` method on your models.
  - Flaggable models should have `get_absolute_url()` methods.

1.3 Quickstart

Install dj-spam:

```
pip install dj-spam
```

Configure it into your project:

```
# settings.py
INSTALLED_APPS += ['spam', ]
```

```
# urls.py
url(r'^spam/', include('spam.urls', namespace='spam')),
```

For any model you want to flag:

```
from spam import Spammable

class MyModel(Spammable, models.Model):
    # Define your model here. Spammable attaches
    #   the spam_flag field to your model as a ManyToManyField.

    @models.permalink
    def get_absolute_url(self):
        # Not required, but it allows dj-spam to link back to the offending
        # content in the report spam view.
        return 'absolute link to model detail view'
```

Run Migrations

```
./manage migrate
```

Then, in the model's related view:

```
from spam import SpammableMixin

class MyModelDetailView(SpammableMixin, DetailView):
    class = MyModel
```

This empowers you with the view method `spam_report_url` which you can use to define the URL to the reporting form:

```
<a href="{{ view.spam_report_url }}">Report Spam</a>
```

1.4 admin

dj-spam comes with a simple admin view.

1.5 emailing managers

dj-spam emails *settings.MANAGERS* every time something is flagged. If you don't set *settings.MANAGERS*, it will email *settings.ADMINS*.

1.6 Running tests locally

```
coverage run ./manage.py test
```

CHAPTER 2

Installation

At the command line:

```
$ easy_install dj-spam
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv dj-spam
$ pip install dj-spam
```

CHAPTER 3

Usage

To use dj-spam in a project:

```
import spam
```

CHAPTER 4

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/pydanny/dj-spam/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "feature" is open to whoever wants to implement it.

4.1.4 Write Documentation

dj-spam could always use more documentation, whether as part of the official dj-spam docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/pydanny/dj-spam/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *dj-spam* for local development.

1. Fork the *dj-spam* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/dj-spam.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv dj-spam
$ cd dj-spam/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 spam tests
$ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, and 3.3, and for PyPy. Check https://travis-ci.org/pydanny/dj-spam/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_spam
```

CHAPTER 5

Credits

5.1 Development Lead

- Daniel Roy Greenfeld <pydanny@gmail.com>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

History

6.1 0.3.0 ???

- Added codecov.io badge
- Officially supporting Python 3.5

6.2 0.2.0 (2-15-07-29)

- Added admin functionality.
- Fixed broken spam report form.
- Email of managers when content is flagged as spam.

6.3 0.1.0 (2-15-07-28)

- First release on PyPI.