Django Username Tools Documentation

Release 0.3.0

keshaB Paudel

Mar 03, 2018

Contents

Contents:

CHAPTER 1

Django Username Tools

Utilities and fields that validate usernames during registration. Useful for Django projects that allow public user registration.

1.1 Documentation

The full documentation is at https://django-username-tools.readthedocs.io.

1.2 Installation

See Installation for instructions.

1.3 Usage

See Usage for usage info.

1.4 Features

- Validates using a blacklist of usernames. Comes with a default set of blacklisted usernames taken from the-big-username-blacklist project
- A ready-to-use *UsernameModelField* for custom user models and *UsernameFormField* for user registration form.
- Readable source code with 100% test coverage.

1.5 TODO

- Add email username field
- A default blacklist of disposable email domains
- API docs for modules, classes and functions

1.6 Why use model to store the blocked usernames?

Using database to store blacklisted username has several advantages. I belive that a blacklist of usernames should be treated like any other data in the project.

- The list can be updated dynamically from code or by using the django admin.
- The list can vary depending on different factors such as locality, and the scope of project etc.
- Database backend allows more sophisticated lookups that we can leverage if required.

1.7 Running Tests

Clone the repository. If you use pipenv, which I highly recommend, run the following commands:

```
pipenv install -d
pipenv run ./runtests.py
```

If you don't use pipenv, run the following commands:

```
source <YOURVIRTUALENV>/bin/activate
(myenv) $ pip install -r requirements_dev.txt
(myenv) $ python runtests.py
```

1.8 Credits

Tools used in rendering this package:

- Cookiecutter
- cookiecutter-djangopackage

CHAPTER 2

Installation

At the command line:

```
$ pip install django-username-tools
```

Now add *username_tools* to your *INSTALLED_APPS*.

```
INSTALLED_APPS = (
    ...
    'username_tools',
    ...
)
```

It is necessary to populate the database with usernames to blacklist. Go to your project directory and run this command:

```
python manage.py populate_blacklist
```

CHAPTER 3

Usage

You can either use the *UsernameFormField* or *UsernameModelField*.

3.1 Username form field

If you are using the default *User* model from *django.contrib.auth* then you should use the *UsernameFormField*.

```
from django import forms
from username_tools.fields import UsernameFormField


class MyUserRegistrationForm(forms.Form):
    username = UsernameFormField()

    # other form fields
```

3.2 Custom user models

If you have defined a custom user model in your project, you can use *UsernameModelField* as your username field.

```
from django.db import models
from username_tools.fields import UsernameModelField


class MyCustomUser(models.Model):
    username = UsernameModelField()

    # other model fields
```

3.3 Populate default blocked usernames

Use *populate_blacklist* management command to populate from the command line interface:

```
python manage.py populate_blacklist
```

Or, you can use the *populate* manager method to populate from your code or the python shell.

```
from username_blacklist.models import UsernameBlacklist
UsernameBlacklist.objects.populate()
```

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/poudel/django-username-tools/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

Django Username Tools could always use more documentation, whether as part of the official Django Username Tools 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/poudel/django-username-tools/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 *django-username-tools* for local development.

1. Fork the *django-username-tools* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/django-username-tools.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 django-username-tools
$ cd django-username-tools/
$ 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 username_tools 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/poudel/django-username-tools/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_username_tools
```

CHAPTER 5

Credits

5.1 Development Lead

- keshaB Paudel <self@keshab.net>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

History

6.1 0.1.0 (2018-02-20)

- First release on PyPI.