thecut-emailfield Documentation

Release 1.0.2

The Cut Creative

August 16, 2016

Contents

1 thecut-emailfield **3**
- 1.1 Documentation . . . 3
- 1.2 Quickstart . . . 3
- 1.3 Credits . . . 3

2 Installation instructions **5**

3 Usage **7**
- 3.1 Validating a model field . . . 7
- 3.2 Validating a form field . . . 7

4 API Reference **9**

5 Testing **11**
- 5.1 Running unit tests . . . 11
- 5.2 Available tests . . . 12

6 History **13**
- 6.1 1.0.2 (2016-08-16) . . . 13
- 6.2 1.0.1 (2016-08-16) . . . 13
- 6.3 1.0 (2016-08-16) . . . 13
- 6.4 0.4 (2016-06-28) . . . 13
- 6.5 0.3 (2016-06-28) . . . 13
- 6.6 0.2.4 (2016-03-11) . . . 14
- 6.7 0.2.3 (2015-11-23) . . . 14
- 6.8 0.2.2 (2015-11-23) . . . 14
- 6.9 0.2.1 (2015-01-13) . . . 14
- 6.10 0.2 (2014-09-18) . . . 14
- 6.11 0.1 (2014-06-17) . . . 14

7 Credits **15**

Contents:

CHAPTER 1

thecut-emailfield

A Django email field which validates by checking for an MX record on the domain.

1.1 Documentation

The full documentation is at https://thecut-emailfield.readthedocs.org.

1.2 Quickstart

Install `thecut-emailfield` using the installation instructions found in the project documentation.

An example `models.py`:

```
# -*- coding: utf-8 -*-
from __future__ import absolute_import, unicode_literals
from django.db import models
from thecut.emailfield.models import EmailField


class EmailModel(models.Model):

    email = EmailField()
```

An example `forms.py`:

```
# -*- coding: utf-8 -*-
from __future__ import absolute_import, unicode_literals
from django import forms
from thecut.emailfield.forms import EmailField


class EmailForm(forms.Form):

    email = EmailField()
```

1.3 Credits

See `AUTHORS.rst`.

CHAPTER 2

Installation instructions

1. Install via pip / pypi:

```
$ pip install thecut-emailfield
```

2. Add to your project's `INSTALLED_APPS` setting:

```
INSTALLED_APPS = [
    # ...
    'thecut.emailfield'
    # ...
]
```

3. Sync your project's migrations:

```
$ python manage.py migrate emailfield
```

CHAPTER 3

Usage

3.1 Validating a model field

You can use `thecut-emailfield` to validate email addresses on a model by using *`EmailField`*.

An example `models.py`:

```
# -*- coding: utf-8 -*-
from __future__ import absolute_import, unicode_literals
from django.db import models
from thecut.emailfield.models import EmailField


class EmailModel(models.Model):

    email = EmailField()
```

3.2 Validating a form field

You can use `thecut-emailfield` to validate email addresses on a form by using *`EmailField`*.

An example `forms.py`:

```
# -*- coding: utf-8 -*-
from __future__ import absolute_import, unicode_literals
from django import forms
from thecut.emailfield.forms import EmailField


class EmailForm(forms.Form):

    email = EmailField()
```

CHAPTER 4

API Reference

class `thecut.emailfield.models.`**`EmailField`**(**args*, ***kwargs*)

An email address model field that performs MX record validation.

class `thecut.emailfield.forms.`**`EmailField`**(*max_length=None*, *min_length=None*, **args*, ***kwargs*)

An email address model field that performs MX record validation.

CHAPTER 5

Testing

5.1 Running unit tests

5.1.1 Using your system's Python / Django

You can perform basic testing against your system's Python / Django.

1. Install the test suite requirements:

```
$ pip install -r requirements-test.txt
```

2. Ensure a version of Django is installed:

```
$ pip install Django
```

3. Run the test runner:

```
$ python runtests.py
```

5.1.2 Using a virtualenv

You can use `virtualenv` to test without polluting your system's Python environment.

1. Install `virtualenv`:

```
$ pip install virtualenv
```

2. Create and activate a `virtualenv`:

```
$ cd thecut-emailfield
$ virtualenv .
$ source bin/activate
(thecut-emailfield) $
```

3. Follow 'Using your system's Python / Django' above.

5.1.3 Using tox

You can use tox to automatically test the application on a number of different Python and Django versions.

1. Install `tox`:

```
$ pip install -r requirements-test.txt
```

2. Run `tox`:

```
(thecut-emailfield) $ tox --recreate
```

Tox assumes that a number of different Python versions are available on your system. If you do not have all required versions of Python installed on your system, running the tests will fail. See `tox.ini` for a list of Python versions that are used during testing.

5.1.4 Test coverage

The included `tox` configuration automatically detects test code coverage with `coverage`:

```
$ coverage report
```

5.2 Available tests

5.2.1 Forms

class `thecut.emailfield.tests.test_forms.`**`TestEmailFormField`**(*methodName='runTest'*)

Tests for the *`thecut.emailfield.models.EmailField`* model field.

`test_accepts_domain_with_mx_record`(**args*, ***keywargs*)

Accept email address for a domain with an MX record.

`test_rejects_domain_without_mx_record`(**args*, ***keywargs*)

Reject email address for a domain without an MX record.

5.2.2 Models

class `thecut.emailfield.tests.test_models.`**`TestEmailModelField`**(*methodName='runTest'*)

Tests for the *`thecut.emailfield.models.EmailField`* model field.

`test_accepts_domain_with_mx_record`(**args*, ***keywargs*)

Accept email address for a domain with an MX record.

`test_rejects_domain_without_mx_record`(**args*, ***keywargs*)

Reject email address for a domain without an MX record.

CHAPTER 6

History

6.1 1.0.2 (2016-08-16)

- Documentation bug fixes.

6.2 1.0.1 (2016-08-16)

- Documentation bug fixes.

6.3 1.0 (2016-08-16)

- Testing improvements
- Documentation improvements.

6.4 0.4 (2016-06-28)

- Redesigned testing environment.
- Added basic documentation.

6.5 0.3 (2016-06-28)

- Stop automated testing for Django 1.5 and Django 1.6.
- Improved code coverage exception list.
- Improved code coverage reporting with Travis.
- Removed pre-Django 1.5 email validation code.

6.6 0.2.4 (2016-03-11)

- Configured code coverage exceptions.
- Unicode packaging fix (`version.py`).

6.7 0.2.3 (2015-11-23)

- Configured continuous integration via Travis.
- Configured code coverage tracking via codecov.

6.8 0.2.2 (2015-11-23)

- Added unit tests.
- Added basic unit test environment (with `tox`).
- Accounted for `dnspython` incompatibility with Python 3.

6.9 0.2.1 (2015-01-13)

- Added `LICENSE`, `README`.etc

6.10 0.2 (2014-09-18)

- Added Python 3 packging support.
- Added extra validation of email addresses in Django 1.4

6.11 0.1 (2014-06-17)

- First release.

CHAPTER 7

Credits

- Josh Crompton <josh.crompton@thecut.net.au>
- Matt Austin <matt.austin@thecut.net.au>
- Kye Russell <kye.russell@thecut.net.au>

E

EmailField (class in thecut.emailfield.forms), 9
EmailField (class in thecut.emailfield.models), 9

T

test_accepts_domain_with_mx_record() (thecut.emailfield.tests.test_forms.TestEmailFormField method), 12
test_accepts_domain_with_mx_record() (thecut.emailfield.tests.test_models.TestEmailModelField method), 12
test_rejects_domain_without_mx_record() (thecut.emailfield.tests.test_forms.TestEmailFormField method), 12
test_rejects_domain_without_mx_record() (thecut.emailfield.tests.test_models.TestEmailModelField method), 12
TestEmailFormField (class in thecut.emailfield.tests.test_forms), 12
TestEmailModelField (class in thecut.emailfield.tests.test_models), 12