

---

# **Django Foundation Formtags Documentation**

***Release 0.1***

**Christopher Clarke**

**Jun 06, 2018**



<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Usage</b>	<b>5</b>
2.1	Required versus optional fields . . . . .	5
<b>3</b>	<b>Contributing</b>	<b>7</b>
3.1	Types of Contributions . . . . .	7
3.2	Get Started! . . . . .	8
<b>4</b>	<b>Authors</b>	<b>11</b>
4.1	Development Leads . . . . .	11
4.2	Core Committers . . . . .	11
4.3	Contributors . . . . .	11
<b>5</b>	<b>History</b>	<b>13</b>
5.1	0.0.9 (2017-06-05) . . . . .	13
5.2	0.0.8 (2017-05-09) . . . . .	13
5.3	0.0.7 (2016-09-25) . . . . .	13
5.4	0.0.6 (2016-08-05) . . . . .	13
5.5	0.0.5 (2016-07-12) . . . . .	13



Django template tags to work with Zurb Foundation forms



# CHAPTER 1

---

## Installation

---

Create a virtualenv:

```
$ virtualenv my-env
```

At the command line:

```
$ pip install django-foundation-formtags
```





## CHAPTER 2

---

### Usage

---

To start use django-foundation-forms in a project you must include in your settings:

```
INSTALLED_APPS = (  
    'foundation_formtags',  
)
```

In the template load foundation tags by:

```
{% load foundation_formtags %}
```

To use the django-form-foundation filter:

```
<form class="form" action="{{ url }}" method="POST">  
    {% csrf_token %}  
    {{ form|as_foundation }}  
</form>
```

To use the django-form-foundation field tags:

```
<form class="form" action="{{ url }}" method="POST">  
    {% csrf_token %}  
    {% render_field form.name %}  
  
    {% render_field form.subject %}  
  
    {% render_field form.message %}  
</form>
```

### 2.1 Required versus optional fields

The normal behaviour is to mark required fields with \*. If you want to mark only optional fields, you can add the following setting to your project settings file.

FOUNDATION\_FORMTAGS\_USE\_OPTIONAL = True

To learn more about optional fields read this [blog](#).

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

### 3.1 Types of Contributions

#### 3.1.1 Report Bugs

Report bugs at <https://github.com/chrisdev/django-foundation-formtags/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.

#### 3.1.2 Fix Bugs

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

#### 3.1.3 Implement Features

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

### 3.1.4 Write Documentation

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

### 3.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/chrisdev/django-foundation-formtags/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 :)

## 3.2 Get Started!

Ready to contribute? Here's how to set up *foundation\_formtags* for local development.

1. Fork the *foundation\_formtags* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/django-foundation-formtags.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 foundation_formtags
$ cd foundation_formtags/
$ 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 foundation_formtags
$ 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.

### 3.2.1 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, 3.3, and 3.4, and for PyPy. Check [https://travis-ci.org/chrisdev/django-foundation-formtags/pull\\_requests](https://travis-ci.org/chrisdev/django-foundation-formtags/pull_requests) and make sure that the tests pass for all supported Python versions.

### 3.2.2 Tips

To run a subset of tests:

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



### 4.1 Development Leads

- Christopher Martin Clarke (@chrisdev)

### 4.2 Core Committers

- Lendl Smith (@ilendl2)
- Parbhat Puri (@parbhat)
- Daniel Andrilik (@andrlik)

### 4.3 Contributors

None yet. Why not be the first?





### 5.1 0.0.9 (2017-06-05)

- Cleaned up travis configuration for tox to improve testing

### 5.2 0.0.8 (2017-05-09)

- Make it Compatible with Django 1.11

### 5.3 0.0.7 (2016-09-25)

- Make Compatible with Django 1.10
- Minimum version is now Django 1.8. For earlier versions of Django use 0.0.6

### 5.4 0.0.6 (2016-08-05)

Fix for checkbox rendering and New error class for Non field errors in Foundation 6

### 5.5 0.0.5 (2016-07-12)

- First release on PyPI.