

---

# **crispy-templates Documentation**

***Release 0.2.0***

**Ryan Castner**

January 17, 2017



<b>1</b>	<b>crispy-templates</b>	<b>3</b>
1.1	Documentation . . . . .	3
1.2	Quickstart . . . . .	3
1.3	Credits . . . . .	3
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>Layouts</b>	<b>9</b>
4.1	Field . . . . .	9
<b>5</b>	<b>Contributing</b>	<b>11</b>
5.1	Types of Contributions . . . . .	11
5.2	Get Started! . . . . .	12
<b>6</b>	<b>Credits</b>	<b>13</b>
6.1	Development Lead . . . . .	13
6.2	Contributors . . . . .	13
<b>7</b>	<b>History</b>	<b>15</b>
7.1	0.2.0 (2017-01-17) . . . . .	15
7.2	0.1.0 (2017-01-14) . . . . .	15



Contents:



---

## crispy-templates

---

A collection of useful templates for customizing crispy forms layouts.

### 1.1 Documentation

The full documentation is at <https://crispy-templates.readthedocs.io>.

### 1.2 Quickstart

`crispy-templates` depends on `[crispy-forms]`(<https://github.com/django-crispy-forms/django-crispy-forms#django-crispy-forms>), make sure that you have properly installed `crispy forms` first.

Install `crispy-templates`:

```
pip install crispy-templates
```

Add it to your `INSTALLED_APPS`:

```
INSTALLED_APPS = (  
    ...  
    'crispy_templates.apps.CrispyTemplatesConfig',  
    ...  
)
```

Define your layout template in the `crispy_forms.helper.FormHelper` object:

Add the javascript by running `python manage.py collectstatic` and then including it in your template:

```
{% static 'crispy_templates/js/crispy_templates.js' %}
```

**Preferably add it to the footer of your page so as to not delay the page load. Note** - Currently only supports the `bootstrap3` template pack that comes with `crispy_forms`.

### 1.3 Credits

Tools used in rendering this package:

- [Cookiecutter](#)
- [cookiecutter-djangopackage](#)



---

# Installation

---

At the command line:

```
$ easy_install crispy-templates
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv crispy-templates  
$ pip install crispy-templates
```



---

**Usage**

---

To use crispy-templates in a project, add it to your *INSTALLED\_APPS*:

```
INSTALLED_APPS = (  
    ...  
    'crispy_templates.apps.CrispyTemplatesConfig',  
    ...  
)
```



---

## Layouts

---

Crispy forms allows us to override the *template* attribute passed to their layout objects, by including the *crispy\_templates* template you will get the new template format.

### 4.1 Field

These are templates related to the *Field* layout object.

#### 4.1.1 Radio Button Group Select

Instead of the regular dropdown select widget or a `RadioSelect` widget we can create a radio button group.

Radio button groups should be used when the number of choices is less than five. See <https://uxdesign.cc/design-better-forms-96fadca0f49c#.tq12xdk3z> for an example of what we are trying to achieve.

To implement this type of button group, in your layout:

```
self.helper.layout = Layout(
    ...
    Field('field_name', template='crispy_templates/radio_btngroup_field.html'),
    ...
)
```

Then add the javascript to the bottom of your page in the template:

```
{% block script %} {% static 'crispy_templates/js/crispy_templates.js' %}
{% endblock %}
```

**The select in the form will now render as a button group. Note** - You need the *buttons.js* javascript included in your html. See <https://getbootstrap.com/javascript/#buttons>



---

## 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:

### 5.1 Types of Contributions

#### 5.1.1 Report Bugs

Report bugs at <https://github.com/audiolion/crispy-templates/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.

#### 5.1.2 Fix Bugs

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

#### 5.1.3 Implement Features

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

#### 5.1.4 Write Documentation

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

#### 5.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/audiolion/crispy-templates/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 :)

## 5.2 Get Started!

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

1. Fork the *crispy-templates* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/crispy-templates.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 crispy-templates
$ cd crispy-templates/
$ 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. 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
```

6. Submit a pull request through the GitHub website.

---

**Credits**

---

## 6.1 Development Lead

- Ryan Castner <[castner.rr@gmail.com](mailto:castner.rr@gmail.com)>

## 6.2 Contributors

None yet. Why not be the first?



---

## History

---

### 7.1 0.2.0 (2017-01-17)

- Fixed method of rendering field so form would pull out the data
- Refactored how the project templates were structured and how to include the new template override
- Updated documentation

### 7.2 0.1.0 (2017-01-14)

- First release on PyPI.