

---

# **Wagtail Form Blocks Documentation**

***Release 0.2.0***

**Tim Leguijt**

**Aug 09, 2017**



---

## Contents

---

<b>1</b>	<b>Wagtail Form Blocks</b>	<b>3</b>
1.1	Features . . . . .	3
1.2	Credits . . . . .	3
<b>2</b>	<b>Installation</b>	<b>5</b>
2.1	Stable release . . . . .	5
2.2	From sources . . . . .	5
2.3	Configure Django . . . . .	5
<b>3</b>	<b>Usage</b>	<b>7</b>
3.1	Define custom form model . . . . .	7
<b>4</b>	<b>Contributing</b>	<b>9</b>
4.1	Types of Contributions . . . . .	9
4.2	Get Started! . . . . .	10
4.3	Pull Request Guidelines . . . . .	11
4.4	Tips . . . . .	11
<b>5</b>	<b>Indices and tables</b>	<b>13</b>



Contents:



# CHAPTER 1

---

## Wagtail Form Blocks

---

A Wagtail module that provides content blocks to display and process user defined forms

- Documentation: <https://wagtailformblocks.readthedocs.io>.

### Features

- TODO

### Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.





### Stable release

To install Wagtail Form Blocks, run this command in your terminal:

```
$ pip install wagtailformblocks
```

This is the preferred method to install Wagtail Form Blocks, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

### From sources

The sources for Wagtail Form Blocks can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/LUKKIEN/wagtailformblocks
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/LUKKIEN/wagtailformblocks/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

### Configure Django

Add `wagtailformblocks`, `wagtail.wagtailforms` and `wagtail.contrib.modeladmin` to your `INSTALLED_APPS` in settings:

```
INSTALLED_APPS += [  
    'wagtailformblocks',  
    'wagtail.wagtailforms',  
    'wagtail.contrib.modeladmin'  
]
```

Add `wagtailformblocks.url` to your url config to enable the form processing views:

```
from wagtailformblocks import urls as wagtailformblock_urls  
  
urlpatterns = [  
    ...  
    url(r'^forms/', include(wagtailformblock_urls)),  
    ...  
]
```

For `wagtailformblocks` to work correctly the form submit actions has to be processed by a javascript. A simple implementation is provided as a static file.

To use this example add the following line to you HTML template:

```
<script src="{% static 'wagtailformblocks/js/formhandler.js' %}"></script>
```

Make sure jQuery is loaded as well.

Once installed, `wagtailformblocks` are ready to be used.

It is shipped with an `EmailForm` model by default, based on wagtails own `AbstractEmailForm` page type.

## Define custom form model

If you want to define your own form model with it's own specific behaviour, simply subclass the `BaseForm` class.

Example:

```
from django.db import models

from wagtailformblocks.models import BaseForm


class MyCustomForm(BaseForm):
    send_quotation = models.BooleanField(verbose_name=_('send me a quotation'),
                                        default=True)

    class Meta:
        verbose_name = _('My custom form')

    panels = BaseForm.panels + [
        FieldPanel('send_quotation',),
    ]

    def process_form_submission(self, form):
        super(MyCustomForm, self).process_form_submission(form)
        if self.send_quotation:
            # Do some extra stuff here
```

Your extra model will show up in the Wagtail CMS admin automatically.



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

### Types of Contributions

#### Report Bugs

Report bugs at <https://github.com/LUKKIEN/wagtailformblocks/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.

#### Fix Bugs

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

#### Implement Features

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

## Write Documentation

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

## Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/LUKKIEN/wagtailformblocks/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 :)

## Get Started!

Ready to contribute? Here's how to set up *wagtailformblocks* for local development.

1. Fork the *wagtailformblocks* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/wagtailformblocks.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 wagtailformblocks
$ cd wagtailformblocks/
$ 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 wagtailformblocks tests
$ python setup.py test or 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.

## 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, 3.4 and 3.5, and for PyPy. Check [https://travis-ci.org/tleguijt/wagtailformblocks/pull\\_requests](https://travis-ci.org/tleguijt/wagtailformblocks/pull_requests) and make sure that the tests pass for all supported Python versions.

## Tips

To run a subset of tests:

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





## CHAPTER 5

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`