

---

# Django File Context Documentation

*Release 0.2.7*

**George Silva**

**Mar 27, 2018**



---

# Contents

---

<b>1</b>	<b>Django File Context</b>	<b>3</b>
1.1	Documentation . . . . .	3
1.2	Quickstart . . . . .	3
1.3	Features . . . . .	4
1.4	Usage . . . . .	4
1.5	Running Tests . . . . .	4
1.6	Credits . . . . .	4
<b>2</b>	<b>Installation</b>	<b>7</b>
<b>3</b>	<b>Usage</b>	<b>9</b>
<b>4</b>	<b>Contributing</b>	<b>11</b>
4.1	Types of Contributions . . . . .	11
4.2	Get Started! . . . . .	12
4.3	Pull Request Guidelines . . . . .	13
4.4	Tips . . . . .	13
<b>5</b>	<b>Credits</b>	<b>15</b>
5.1	Development Lead . . . . .	15
5.2	Contributors . . . . .	15
<b>6</b>	<b>History</b>	<b>17</b>
6.1	0.1.0 (2017-10-25) . . . . .	17



Contents:



File Context provides an easy way to store different documents/attachments

### 1.1 Documentation

The full documentation is at <https://django-file-context.readthedocs.io>.

### 1.2 Quickstart

Install Django File Context:

```
pip install django-file-context
```

Add it to your *INSTALLED\_APPS*:

```
INSTALLED_APPS = (  
    ...  
    'file_context.apps.FileContextConfig',  
    ...  
)
```

You need to add this to your *INSTALLED\_APPS*, because we have our own models.

Add Django File Context's URL patterns:

```
from file_context import urls as file_context_urls
```

```
urlpatterns = [
    ...
    url(r'^$', include(file_context_urls)),
    ...
]
```

This is optional. Actually, you can include your own URLs.

### 1.3 Features

- Generic File model so you can store different kinds of files, images, etc, using a single model; \* Attach/Detach the file model to other models \* Cool descriptor, so you don't have to keep mangling GenericForeignKeys inside your models.

### 1.4 Usage

1. First of all, define your first model

```
:: python from file_context.managers import Files
```

```
class MyModel(models.Model):
    name = models.CharField(max_length=128)
    files = Files()
```

2. That's it.
3. You can use the Files API to attach files to MyModel instances, using:

```
:: python
```

```
uploaded_file = File.objects.get(pk=1) a = MyModel.objects.create(name='foo')
a.files.attach(uploaded_file) a.files.detach(uploaded_file) a.files.clear()
```

### 1.5 Running Tests

Does the code actually work?

```
source <YOURVIRTUALENV>/bin/activate
(myenv) $ pip install tox
(myenv) $ tox
```

### 1.6 Credits

Tools used in rendering this package:

- Huge props to django-taggit that inspired me to do the descriptor idea!
- Cookiecutter



- `cookiecutter-djangopackage`



## CHAPTER 2

---

### Installation

---

At the command line:

```
$ easy_install django-file-context
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv django-file-context  
$ pip install django-file-context
```



To use Django File Context in a project, add it to your *INSTALLED\_APPS*:

```
INSTALLED_APPS = (  
    ...  
    'file_context.apps.FileContextConfig',  
    ...  
)
```

Add Django File Context's URL patterns:

```
from file_context import urls as file_context_urls  
  
urlpatterns = [  
    ...  
    url(r'^$', include(file_context_urls)),  
    ...  
]
```



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/george-silva/django-file-context/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 File Context could always use more documentation, whether as part of the official Django File Context 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/george-silva/django-file-context/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-file-context* for local development.

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

```
$ git clone git@github.com:your_name_here/django-file-context.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-file-context
$ cd django-file-context/
$ 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 file_context 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/george-silva/django-file-context/pull\\_requests](https://travis-ci.org/george-silva/django-file-context/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_file_context
```



### 5.1 Development Lead

- George Silva <[george@sigmageosistemas.com.br](mailto:george@sigmageosistemas.com.br)>

### 5.2 Contributors

None yet. Why not be the first?



### 6.1 0.1.0 (2017-10-25)

- First release on PyPI.