
Django File Chooser Documentation

Release 0.1

Martin van Wingerden

Apr 24, 2017

Contents

1	Installation	3
2	Quickstart	5
2.1	Example template	5
2.2	Example urls.py	5
2.3	Example views	6
2.4	Template tags and filters	6
2.5	Settings	6
3	Templates	7
3.1	filechooser/filedatatable.js	7
3.2	filechooser/filedatatable.html	7
4	History	9
5	Indices and tables	11

Contents:

CHAPTER 1

Installation

The preferred way to install `django-filechooser` is `pip`:

```
$ pip install django-filechooser
```

Alternatively, you can install download or clone this repo and install from its folder with:

```
$ pip install -e .
```

In your project, you should add `django-filechooser` to your `requirements.txt`.

Be sure to use `virtualenv` if you develop python projects.

Add to `INSTALLED_APPS` in your `settings.py`:

```
'filechooser',
```

After installation, the [Quickstart](#) will get you on your way to using `django-filechooser`.

After *Installation*, you can use `django-filechooser` in your templates.:

Load the `filechooser` library and use the `filechooser_*` tags:

Example template

```
{# Load the tag library #}
{% load filechooser %}

{# Load CSS #}
{% filechooser_css %}

{# Display a filetable #}
{% filechooser_filetable id="identifier" %}

{# Load CSS and JavaScript #}
{% filechooser_javascript id="identifier" %}

{# Read the documentation for more information #}
```

Example urls.py

Note we request an object from the views to retrieve an url object to be added to the list urls

```
urlpatterns = patterns('',
    url(r'^$', views.choose, name='choose'),
    # Add the url-pattern as generated by the filechooser
    views.filechooser().url_pattern()
)
```

Example views

```
# Import the FileChooser object
from filechooser import FileChooser

# Define a method which returns a FileChooser, with a certain ID field (this must
# be the same as the one defined in the template) and root directory for the file
# browsing and finally a callback method to process the selected file
def filechooser():
    return FileChooser("identifier", settings.FILEBROWSER_DIRECTORY, process)

# Define a method to handle selected files.
def process(filename):
    # do something with filename received from the filechooser
```

Template tags and filters

Refer to `templatetags` for more information.

Settings

You can set defaults for `django-filechooser` in your settings file. Refer to `settings` for more information.

You can customize the output of `django-filechooser` by writing your own templates. These templates are available:

filechooser/filedatatable.js

This renders the javascript for the filetable.

filechooser/filedatatable.html

This renders the html for the filetable including the breadcrumb-trail.

CHAPTER 4

History

1.0.0 (??) Initial setup of seperate module

CHAPTER 5

Indices and tables

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