Basic File Manager Documentation

Release 0.6

Simonas Kazlauskas

Mar 25, 2017

Contents

BFM is an Apache Licenced server file manager for Django made with ease of use and new web technologies in mind.

BFM looks like Django Admin page therefore BFM shouldn't cause any unfamiliarity for anyone who used Django before.

Also, it has minimal dependencies.

CHAPTER 1

Features

- Uploading multiple files at once,
- Live upload status,
- Browsing files...
- and directories,
- Doesn’t have a lot of dependencies,
- Extends Django admin templates, so has familiar feeling,
- Ability to add upload applet to your application admin.

CHAPTER 2

Next steps

Introduction

Why another? There's adminfiles and filebrowser already.

Why software is created in the first place? Because there is somebody who is not satisfied with already existing solutions. I'm one of those people and I think that I can do better.

Why use BFM?

1. You hate Flash, love HTML5, CSS3, JavaScript and have modern browser.
2. You want simple and fast interface for managing your files online.
3. You couldn't even make another two work. BFM is simple to setup.

Why you would not want to use BFM yet?

1. You need to support IE, which is actually not even a browser.
2. You want a mature project.
3. If you want integrate it with TinyMCE or something of the like, you shouldn't think about BFM.

Support tables, dependencies

Browser

Version, required to have full functionality of BFM. Some parts of BFM may still work with older browsers, but advanced functionality, like file uploading won't.

Browser	Version
Chrome/Chromium	7 +
Firefox	5 +
Opera	12.0 + *
IE	10 + *
Safari	5.0 +

* XHR2, which is used internally to upload files and report upload status, is not supported by stable versions of these browsers.

Dependencies

BFM was only tested on Python 2.6 and Python 2.7, so it is safe to assume, that dependencies are as follows:

- Python: 2.6 or 2.7,
- Django: 1.3 or 1.4

Installation

pip and easy_install

Easiest way to install BFM is to use `pip`:

```
$ pip install django_bfm
```

And second to easiest is with `easy_install`:

```
$ easy_install django_bfm
```

Note: Using easy_install is discouraged. Why? Read here.

GIT repository

Alternatively, you can clone and install from Github repository, where project is developed.

```
$ git clone git://github.com/simukis/django-bfm.git
$ cd django-bfm
$ python2 setup.py install
```

or with pip:

```
pip install -e git+git://github.com/simukis/django-bfm.git#egg=Package
```

Note: Do not forget to compile .coffee files yourself if you are cloning repository from `master`. You can do so by running `manage.py compile_coffee`.

Configuration

Enabling BFM

After downloading and installing BFM, you need to configure your Django project.

1. Add `'django_bfm',` to your `INSTALLED_APPS` in **settings.py**,
2. Add `url(r'^files/', include('django_bfm.urls')),` to your `urlpatterns` in **urls.py**,
3. Make sure you have staticfiles enabled (with a context processor) and run *python manage.py collectstatic*,
4. Make sure, that static files are served correctly by your production server.

Note: You don't need to run collectstatic if you are working in develovepment server. It serves files automatically.

Next steps

Settings

Users can customize their BFM installation by specifying options in their `settings.py` file.

All settings are read from single dictionary `BFM`.

Example

```
BFM = {
    'FILES_PER_PAGE': 25,
    'SIMULTANEOUS_UPLOADS': 4
}
```

Directories and URLs

If these values are not set, then BFM defaults to values defined in `MEDIA_ROOT` and `MEDIA_URL`.

BFM['MEDIA_DIRECTORY']

The absoulute path to the directory, where files accessible by BFM resides. Please note, that BFM will not be able to access directories, that are parents to specified one.

Example: `/home/example.com/media/uploads/`

Defaults to: `MEDIA_ROOT`

BFM['MEDIA_URL']

URL to use when referring to media located in `BFM_MEDIA_DIRECTORY`.

Example: `/media/uploads/` or `http://media.example.com/uploads/`

Note: It **must** end in slash.

Defaults to: `MEDIA_URL`

Other BFM options

BFM['FILES_PER_PAGE']

BFM uses Pagination to ensure, that reasonable ammount of files is displayed at once. You may specify, how much that "reasonable" is for you by giving this variable an integer value.

Example: `50`

Default: `20` **BFM['ADMIN_UPLOAD_DIRECTORY']**

Directory path relative to `BFM['MEDIA_DIRECTORY']`. Tells BFM where to save files uploaded with *Admin Applet*.

Example: `admin_files/` and files will be uploaded to `{{BFM['MEDIA_DIRECTORY']}}/admin_files/`.

Default: empty string.

BFM['SIMULTANEOUS_UPLOADS']

Tells BFM how much files to upload at once. Requires integer value.

Example: `3`

Default: `2`

Note: If you experiencing issues like dropped connections with uploader then set this option to `1`.

Related settings

LOGIN_URL

To make sure, that only authorized users fiddle with files, BFM requires user to be logged in. If user is not logged in, it will be redirected to LOGIN_URL.

Admin Applet

Warning: BFM javascript file downloads all files it requires including newest version of jQuery, so all `$` and `jQuery` variables defined before will be rewritten. `django.jQuery` variable won't change.

Making admin applet to appear in your application admin panel requires more work because application specific files should be edited.

Firstly, open your application `admin.py` file of an application in which you want applet to appear and place these lines just after other imports.

```
from django.core.urlresolvers import reverse
from django.utils.functional import lazy
reverse_lazy = lazy(reverse, str)
```

Note: If you are using Django 1.4, you can just import it: `from django.core.urlresolvers import reverse_lazy`.

Then in your `SomethingAdmin` class add another class named Media, if it doesn't exist yet.

Finally, add `reverse_lazy('bfm_opt')` into `js` tuple of `Media` class.

In the end `Media` class should look something like this:

```
class EntryAdmin(admin.ModelAdmin):
    class Media:
        # ... Your other css and js ...
        js += (reverse_lazy('bfm_opt'),)
    #Your admin continues here...
```

That's all. You're ready to see your applet in http://example.com/admin/application/model.

You may want to change some options, that changes behavior of Admin applet. You can see them in uploader-settings.

CHAPTER 3

Indices and tables

- genindex
- search