django-file-repository Documentation
Release b

Oscar Carballal Prego

November 05, 2016

django-file-repository's documentation

django-file-repository (djangofr from now on) is a simple file repository. It can be used as a catalog of "thingz" like you books, music or films, or you can use it as a repository, storing files and having the option to download them later.

Main features

- File tagging
- Public and private files
- Private user zone
- Internazionalization (English, Spanish and Galician at this moment)
- And more...

Contents

3.1 Quickstart

To start using django-file-repository (*djangofr* from now on) right away you can use the default installation.

3.1.1 Download

You can download *djangofr* from GitHub:

Stable:

```
http://github.com/cidadania/django-file-repository/downloads
```

Development:

```
git clone git://github.com/cidadania/django-file-repository.git
```

3.1.2 Configuration

The only parameters required to work with *djangofr* are:

> **Warning:** Currently *djangofr* does not support SSL or certificated servers. Support will be included in the future.

Note: If you will not allow user registration you can skip the configuration step.

EMAIL_HOST This is your SMTP server.

> **default** localhost

EMAIL_PORT Port for your server.

> **default** 25

EMAIL_HOST_USER Your email user.

> **default** None

EMAIL_HOST_PASSWORD Your email password.

> **default** None

3.1.3 Build database and start working!

After that, you will only have to run this commands:

```
./manage.py syncdb (to make the database tables)

and after it

./manage.py runserver (it will start the development server on port 8000)
```

The instructions for deployment in production servers are pretty well explained in the django documentation

3.2 Download

Download djangofr from GitHub.

Development:

```
git clone git://github.com/cidadania/django-file-repository,git
```

Stable:

```
http://github.com/cidadania/django-file-repository/downloads
```

3.3 Settings

djangofr does noeed some extra settings apart from the defaults in settings.py We tell you here which are they.

ACCOUNT_ACTIVATION_DAYS This value sets the valid tiome of an activation key after it was sent. If the user does not activate the account in that time, the key expires.

> **default** 7

EMAIL_HOST | EMAIL_PORT | EMAIL_HOST_USER | EMAIL_HOST_PASSWORD This values are for sending the validation email to the users.

> **default** None

LOGIN_URL, LOGIN_REDIRECT_URL, LOGOUT_URL This values tell django where to search for the login, logout and redirect URLs Usually you will want this with the default value, since to modify it you have to modify the application URLs.

> **defaults**
>
> > • /user/login
> >
> > • /user/logout
> >
> > • /

FORCE_LOWERCASE_TAGS Forces the tags to be stored and managed in lowercase form. This is useful, unless you want for any reason manage case sensitive urls.

> **default** True

3.4 `repository` — Repository

3.4.1 `repository.admin` — Administration

3.4.2 `repository.models` — Data models

3.4.3 `repository.forms` — Forms

3.4.4 `repository.urls` — URLs

add/ Adds a new file to the repository

> **reverse** add_file

cat/<cat_id> Shows the documents in a specific category.

> **values** cat_id (Category ID)
>
> **reverse** category_view

myfiles/ Shows the private files a user has.

> **reverse** my_files

<file_id>/delete <file_id>/edit

> Delete or edit the file with <file_id>
>
> **values** file_id

<file_id>/ View detailed info about a particular file.

> **values** file_id

3.4.5 `repository.views` — Views

Indices and tables

- genindex
- modindex
- search