django-project-template Documentation

Release 0.1.0

Dalton Hubble

Sep 27, 2017

Contents

A Django 1.6+ project template that provides a great skeleton application structure with some essential features:

- Separate dev, stag, and prod environments.
  - switch environments easily based on APP_ENV env var.
  - separate development, staging, and production settings
  - separate development and production (incl. staging) requirements
- django-storages static asset storage backend integration with S3.
- Database configuration via environment variable
  - production database passwords kept in environment variables and out of source.
  - dj-database-url configures the database from the DATABASE_URL env var.
  - integrates with Amazon RDS, Heroku DBs, and others easily.
- Simple deployment to Heroku
  - Forced HTTP redirection to HTTPS (except in development)
  - Uses Heroku's free piggyback wildcard SSL.
- Helps you construct organized web applications that follow the 12factor methodology.
- Skeleton 404 and 500 error pages.

CHAPTER 1

Foreword

TODO

CHAPTER 2

Getting Started

Prerequisites

Be sure you have the following installed on your development machine.

- Python >= 2.7
- Git >= 1.7
- Virtualenv >= 1.9
- Database (either one):
  - MySQL >= 5.6 (guide)
  - Postgres >= 9.2 (guide)
- (recommended) Virtualenvwrapper >= 4.0

Creating a Project

Create a virtual environment. With virtualenv,

```
virtualenv venv             # create a virtualenv
source venv/bin/activate    # activate the Python virtualenv
```

or with virtualenvwrapper,

```
mkvirtualenv django         # create and activate environment
workon django               # reactivate existing environment
```

Install Django inside the virtual environment:

```
pip install django
```

If you have cloned django-project-template with:

```
git clone https://github.com/dghubble/django-project-template
```

you can generate a new project with,

```
django-admin.py startproject djproj [dest] \
--template /path/to/django-project-template \
-e py,rst,md,html \
-n Procfile
```

Otherwise,

```
django-admin.py startproject djproj [dest] \
--template https://github.com/dghubble/django-project-template/archive/master.zip \
-e py,rst,md,html \
-n Procfile
```

A djproj **project directory** containing a README.rst, a manage.py script, and a djproj Python **package directory** was created. Rename the outer (project) directory to anything you wish, but don't modify the inner (package) directory, its the package name for you Django application.

```
mv djproj newname        # optionally rename project directory
```

If you specified a destination path after djproj in the startproject command, the project directory *contents* will be generated at that path location, not within a generated project directory.

Project Tweaks

`django-admin.py startproject`'s templating options do not currently allow passing additional options to control the generation of a project. That means a few minor tweaks need to be made to the project before its ready for your development team.

We encourage you to use MySQL or PostgreSQL in development, to match whichever database you plan to use in your production environments. Comment or delete mentions of the databases you don't plan to use in the following sections inside your generated Django application:

- README.rst `Prerequisites` section database dependencies
- database specific requirements in requirements/development.txt, requirements/production.txt
- djproj/settings/common.py `DATABASES` setting

In particular, ensure that the `DATABASES` default database url string has the correct scheme, username, password, port, and database name to match a database developers can use on their development machines. It is generally ok to store local database authentication credentials in source, since your local database account is likely a throw-away account. However, if this is a concern, remove the 'default' keyword argument to `dj_database_url.config` entirely and just set the `DATABASE_URL` environment variable which will be picked up by `dj_database_url`.

If you'd still like to use SQLite3 in development (we don't recommend it), no special requirements are needed and a SQLite `DATABASES` url string is available in djproj/settings/common.py.

Start Developing

You have successfully generated a Django application based on the `django-project-template`. You can now commit and distribute the initial version to other developers on your team if you wish. The README generated inside the application has been customized for your project. It documents local setup, workflows, and deployments.

You can see an *example* of the documentation generated for a Django project next.

CHAPTER 3

Contributing

Want a particular feature? Discovered a bug or problem? Open an Issue or send a Pull Request and I'll respond as soon as I can.

CHAPTER 4

Example

Generated Project

Docs for a Generated Project

CHAPTER 5

Indices and tables

- genindex
- modindex
- search