Django

Introduce

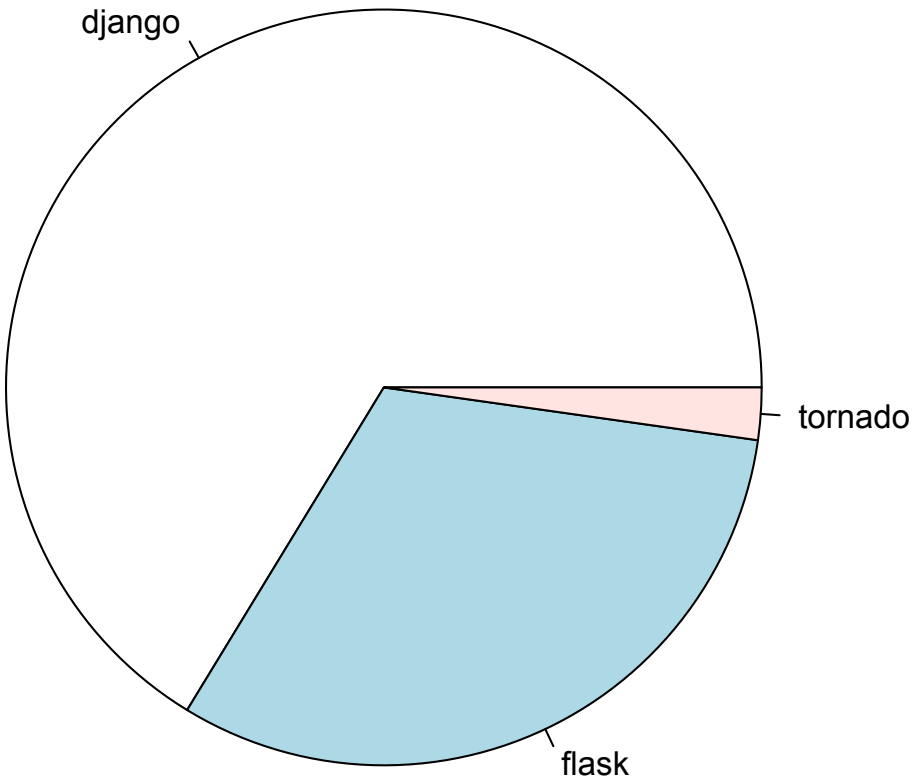
贵州大学

hnzhang1@gzu.edu.cn

December 24, 2018

Overview

github repository of python web framworks (2018.12.24)



django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Django features

- ridiculously fast
- fully loaded
  Django includes dozens of extras you can use to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds, and many more tasks —right out of the box.
- reassuringly secure
  Django includes dozens of extras you can use to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds, and many more tasks —right out of the box.
- exceedingly scalable
- incredibly versatile
  Companies, organizations and governments have used Django to build all sorts of things —from content management systems to social networks to scientific computing platforms.

install

- quick install guide[1]
  `https://docs.djangoproject.com/en/2.1/intro/install/`
- complete installation guide
  `https://docs.djangoproject.com/en/2.1/topics/install/`

[1] Python includes a lightweight database called SQLite so you won't need to set up a database just yet. Django includes a lightweight web server you can use for testing, so you won't need to set up Apache until you're ready to deploy Django in production.

quick install guide

1. install python 3.7
2. install Django
   - 2.1 install pip3
   - 2.2 pip3 install django
3. verifying

```
$ python3
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 26 2018, 23:26:24)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for mo
>>> import django
>>> print(django.get_version())
2.1.4
>>>
```

create a project

From the command line, cd into a directory where you'd like to store your code, then run the following command:

```
$ pwd
/Users/hainingzhang/Documents/GitHub/dm-web/web
$ django-admin startproject mysite
$ tree
.
|____mysite
| |____mysite
| | |______init__.py
| | |____settings.py
| | |____urls.py
| | |____wsgi.py
| |____manage.py
```

runserver

Let's verify our Django project works. Change into the outer mysite directory, and run the following commands:

```
$ python3 manage.py runserver
Performing system checks...
...
December 24, 2018 - 11:37:11
Django version 2.1.4, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Now that the server's running, visit http://127.0.0.1:8000/ with your Web browser. You'll see a "Congratulations!"page, with a rocket taking off. It worked![2]

[2] don't use this server in anything resembling a production environment. At the same time a new file db.sqlite3 is created.

project files

- the outer mysite/ root directory is just a container for your project.
- manage.py: A command-line utility that lets you interact with this Django project in various ways.
- the inner mysite/ directory
  is the actual Python package for your project. Its name is the Python package name you'll need to use to import anything inside it (e.g. mysite.urls).
- ___init___.py: An empty file that tells Python that this directory should be considered a Python package.
- settings.py: Settings/configuration for this Django project.
- urls.py: The URL declarations for this Django project; a "table of contents"of your Django-powered site.
- wsgi.py: An entry-point for WSGI-compatible web servers to serve your project.

crate an app

Our environment –a “project”–is set up, we’re set to start doing work. To create our app, make sure you’re in the same directory as manage.py and type this command:

```
$ python3 manage.py startapp blog
```

app structure

```
|____mysite
| |______init__.py
| |______pycache__
| |____settings.py
| |____urls.py
| |____wsgi.py
|____db.sqlite3
|____blog
| |____migrations
| |____models.py
| |______init__.py
| |____apps.py
| |____admin.py
| |____tests.py
| |____views.py
|____manage.py
```

projects vs. apps

What's the difference between a project and an app? An app is a Web application that does something –e.g., a Weblog system, a database of public records or a simple poll app. A project is a collection of configuration and apps for a particular website. A project can contain multiple apps. An app can be in multiple projects.

creating models

A model is the single, definitive source of truth about your data. It contains the essential fields and behaviors of the data you're storing. (like a table in the database)
Edit the blog/models.py file:

Q&A