Web Dev in Django

Django

Nathan St. Amour

OU ACM

September 27, 2017

Outline

1. Django
   - Overview
   - Structure
   - More Structure
   - Starting a project
   - Apps
   - Templates
   - Views
   - Urls
   - New Page

Overview

- High level framework to help manage developement
- Develop server side operations in python
- Easy to deploy to heroku

Structure

- Django is similar to the Model View Controller scheme.
- Django refers to their schema as Model View Template(MVT)

Model is a python class description of any databases used in the project

View is a python file used to recieve requests and determine what html response to generate

Template is the html code of the project

More Structure

There are three parts of django that you will want to be familiar with:

Templates defined in your app and are the html source pages of you website

Views also defined in your app and are python functions to determine how to render html responses

Urls defined in your main project's directory, they tell your website what to do when a url is typed into the browser

Starting a project

- To start a project type this commands:

```
$ django-admin.py startproject <project>
    <directoryToStartIn>
```

This is the directory structure you will get:

```
[nate@neewamp acm]$ ls
acm  manage.py
[nate@neewamp acm]$ ls acm/
__init__.py __pycache__ settings.py urls.py wsgi.py
[nate@neewamp acm]$
```

- To start your project locally run this command:

```
[nate@neewamp templates]$ python manage.py runserver
[nate@neewamp templates]$
```

Apps

To start making additions to the base that was set up after running the startproject command run a startapp command.

```
[nate@neewamp acm]$ python manage.py startapp mypage
[nate@neewamp acm]$ ls
acm  db.sqlite3 manage.py mypage
```

Apps

Each app will have it's own settings views and html pages. When you create a new app update the main app's settings.py to register the new app.

```
INSTALLED_APPS = [
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'mypage'
]
```

Templates

In the new app you created you can start to create your templates:

- Make a directory called templates under the app with the name you gave to the startapp(mine is called mypage)
- In that directory put your html source code
- It's nice to create subdirectories for each set of related pages to keep things neat

Views

- Inside of a view you should write functions that take a request object and return some response for django to display. More on request objects.
- The function hello returns an html page defined in mypage/templates/hello.html

Example View:

```
from django.http import HttpResponse
from django.shortcuts import render
import datetime
def hello(request):
   today = datetime.datetime.now().date()
   return render(request, "hello.html", {"today" : today})
def index(request):
    return HttpResponse(’<a href="/hello">Hello World</a>’)
def home(request):
   return render(request, "home/index.html", {})
```

Urls

Defined in the app created by the startproject command urls.py can be used to:

- Map urls to code in views.py that can be executed and return a response
- The urls are mapped with python regular expressions

```
from django.conf.urls import url
from django.contrib import admin
import mypage.views

urlpatterns = [
    url(r’ˆadmin/’, admin.site.urls),
    url(r’ˆ$’, mypage.views.index, name = ’index’),
    url(r’ˆhello/’, mypage.views.hello, name = ’hello’),
    url(r’home/’, mypage.views.home, name = ’home’),
]
```

New Page

- Download the base and index pages from my github https://github.com/neewamp/ACMPresent/tree/master/template
- Add the files to the new app you created. Putting base.html in the template directory and index.html in a directory called home under the template directory
- The views and urls file described above have the correct additions to allow your website to display the html

```
nate@neewamp templates]$ ls
base.html hello.html home
[nate@neewamp templates]$ ls home/
index.html
```

What To Do Next

- Go out and do a django tutorial Tutorial
- Host your website on heroku Tutorial