**First-Django-Project Log**

**Create Django Virtual Environment**

conda create --name **django11** Django=1.10

The following NEW packages will be INSTALLED:

certifi pkgs/main/win-64::certifi-2019.3.9-py36\_0

**django** pkgs/free/win-64::**django-1.10**.5-py36\_0

pip pkgs/main/win-64::pip-19.1.1-py36\_0

**python**  pkgs/main/win-64::**python-3.6.8**-h9f7ef89\_7

setuptools pkgs/main/win-64::setuptools-41.0.1-py36\_0

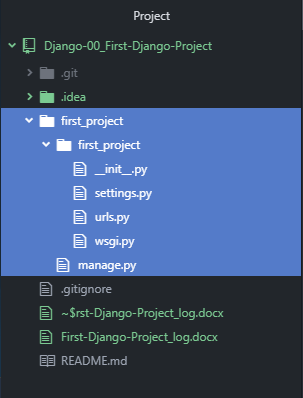
sqlite pkgs/main/win-64::sqlite-3.28.0-he774522\_0

vc pkgs/main/win-64::vc-14.1-h0510ff6\_4

vs2015\_runtime pkgs/main/win-64::vs2015\_runtime-14.15.26706-h3a45250\_4

wheel pkgs/main/win-64::wheel-0.33.4-py36\_0

wincertstore pkgs/main/win-64::wincertstore-0.2-py36h7fe50ca\_0

**Step 1: Create Django Project**

activate **django11**

D:\dev **>** **cd D:\dev\Django-00\_First-Django-Project**

D:\dev\Django-00\_First-Django-Project **>** **django-admin startproject first\_project**

Directory Structure

* first\_project [folder]
  + manage.py
    - This is a Python script that we will use a lot.
    - It will be associated with many commands as we build our web-app.
    - To run server:

> python manage.py runserver

* + - \_\_init\_\_.py
      * This is a blank Python script that (due to its special name) let’s Python know that this directory can be treated as a package.
    - settings.py
      * This is where you store all your project settings
    - urls.py
      * This is a Python script that will store all the URL patterns for your project.
        + Basically, the different pages of your web-application.
    - wsgi.py
      * This is a Python script that acts as the **W**eb **S**erver **G**ateway **I**nterface.
      * It will later-on help us deploy our web-app to production.

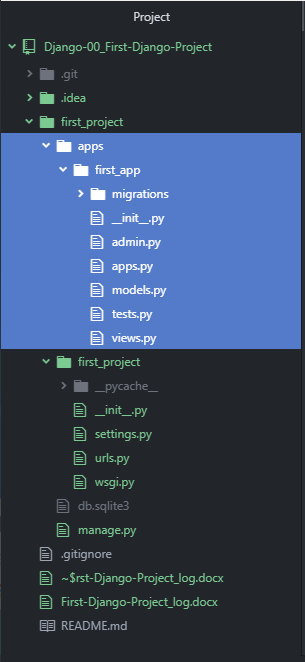
Run local web-server

D:\dev\Django-00\_First-Django-Project **> cd first\_project**

D:\dev\Django-00\_First-Django-Project\first\_project **>** **python manage.py runserver**

* Copy URL into browser
  + <http://127.0.0.1:8000/>

Terminology

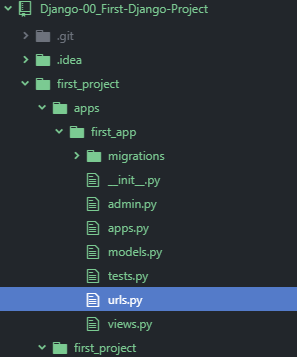
* Migrations
  + A migration allows you to move databases from one desing to another.
    - This is also reversible.
  + So you can “migrate” your database.
* Django Project
  + ****A Django project is a collection of applications and configurations that when combined-together will make up the full web-application (i.e. your complete website running with Django)

**Step 2: Manually create a folder for projects apps**

D:\dev\Django-00\_First-Django-Project\first\_project **>** **mkdir apps**

D:\dev\Django-00\_First-Django-Project\first\_project **>** **cd apps**

D:\dev\Django-00\_First-Django-Project\first\_project\apps **>** **python ..\manage.py startapp first\_app**

****D:\dev\Django-00\_First-Django-Project\first\_project\apps **>** **cd first\_app**

D:\dev\Django-00\_First-Django-Project\first\_project\apps\first\_app **>** **touch urls.py**

* Modify first\_project/settings.py

INSTALLED\_APPS = [  
 'apps.first\_app', # JOSH: added  
 'django.contrib.admin',  
 'django.contrib.auth',  
 'django.contrib.contenttypes',  
 'django.contrib.sessions',  
 'django.contrib.messages',  
 'django.contrib.staticfiles',  
]

* + Add apps/first\_app to list of INSTALLED\_APPS
* Add code snippet to **project**-level urls.py file

from django.conf.urls import url, include  
urlpatterns = [  
 url(r'^', include('apps.first\_app.urls')),  
 # url(r'^admin/', admin.sites.urls)  
]

* + Place in app-name in the include function
* Add code snipped to **app**-level urls.py file

from django.conf.urls import url  
from . import views  
  
urlpatterns = [  
 url(r'^$', views.index),  
]

* Add code snipped to **app**-level views.py file

# Create your views here.  
from django.shortcuts import render, HttpResponse  
def index(request):  
 return HttpResponse("this is the equivalent

of @app.route('/')!")

* Re-run server:

