Django – Basic Project and Application setup

*Assumption is that python is installed and configured. – adhamlin 11/2020*

1. **Create a virtual environment**:   
   python -m venv <virt-env-name>  
   EXAMPLE: python -m venv django\_env
2. **Activate virtual environment**:   
   <virt-env-name>\Scripts\activate (Linux: source <virt-env-name>/bin/activate)  
   EXAMPLE: django\_env\Scripts\activate
3. **Install Django**:   
   pip install Django
4. **Start a Project**:   
   django-admin startproject <project-name> **.** <-- *Note the single space & period. This is so it doesn’t nest the manage.py file inside another folder within the current directory. Try it without to see.*EXAMPLE: django-admin startproject django\_project **.**
5. **Create an initial database**:   
   python manage.py migrate
6. **View Project - Test run the Django server**:   
   python manage.py runserver
7. **Start an Application for the project**:   
   python manage.py startapp <app-name>  
   EXAMPLE: python manage.py startapp django\_app
8. **Define a class model** in ‘<app-name>\models.py’ file. Tell Django how to will work with the data stored in the application.  
   EXAMPLE:

from django.db import models  
  
# Create your models here.  
# This model tells Django how to work with the data that will be stored in the app  
class Topic(models.Model):  
 *"""A topic the admin user enters.  
 This class inherits from the 'Model' parent class that's included in Django.  
 """* text = models.CharField(max\_length=255)  
 date\_added = models.DateTimeField(auto\_now\_add=True)  
  
 def \_\_str\_\_(self):  
 return self.text

1. **Activate the class model** in the ‘<project-name>\settings.py’ file (Edit to add to INSTALLED\_APPS).

# Application definition  
INSTALLED\_APPS = [  
 # my apps - here in case we need to override any behaviors of the default apps  
 'djlog\_app',  
  
 # default Django app upon setup  
 'django.contrib.admin',  
 'django.contrib.auth',  
 'django.contrib.contenttypes',  
 'django.contrib.sessions',  
 'django.contrib.messages',  
 'django.contrib.staticfiles',  
  
]

1. Tell Django to determine how to **modify** the **initial db** to match the class model:   
   python manage.py makemigrations <app-name>  
   EXAMPLE: python manage.py makemigrations django\_app
2. **Apply that modification** to the initial db:  
   python manage.py migrate <app-name>  
   EXAMPLE: python manage.py migrate django\_app
3. **Create a Super User**/Admin:  
   python manage.py createsuperuser
4. **Register the model** with the admin site in ‘<app-name>\admin.py’

from django.contrib import admin  
# import the class models from models.py and register them  
from .models import Topic  
  
# Register your models here.  
admin.site.register(Topic)

1. **View admin site** http://127.0.0.1:8000/admin/   
   login, add, view or edit model contents under the <app-name>
2. **REPEATABLE:** Define another class model in ‘<app-name>\models.py’ file.
3. **REPEATABLE:** Tell Django to determine how to modify the db to match the new class model:   
   python manage.py makemigrations <app-name>
4. **REPEATABLE:** Apply that modification to the db:  
   python manage.py migrate <application-name>
5. **REPEATABLE:** Register the entry model with the admin site in ‘<app-name>\admin.py’
6. **REPEATABLE:** View admin site to add, view or edit entry model contents under the <app-name>
7. **Repeat 15-19 as need for the admin site.**