**CREATE YOUR VIRTUAL ENVIRORMENT:**

conda create -n myNewWebApp python=3.7 anaconda

activate myNewWebApp

python -m pip install Django

pip install django-extensions

pip install djangorestframework

**START YOUR CORE PROJECT:**

django-admin startproject core

**START YOUR APPLICATION:**

python manage.py startapp baseApp

**SETTUP YOUR SCRIPTS FOLDER:**

C:\espacios\API-Test-Project\core>mkdir scripts

C:\espacios\API-Test-Project\core>touch scripts/\_\_init\_\_.py

**SETTUP YOUR STATIC FILES FOLDER:**

C:\espacios\API-Test-Project\core>mkdir staticFiles

C:\espacios\API-Test-Project\core\staticFiles> mkdir cssFiles dataSets docs\_and\_images javaScriptFiles

**INSTALL YOUR APPLICATION, DJANGO-REST-FRAMEWORK, DJANGO-EXTENSIONS:**  
**C:\espacios\API-Test-Project\core\baseApp**

INSTALLED\_APPS = [

'django\_extensions',

'baseApp.apps.BaseappConfig',

'rest\_framework',

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

]

**INSTALL YOUR STATIC and MEDIA**

import os  
STATIC\_URL = '/staticFiles/'

MEDIA\_URL = '/docs\_and\_images/'

#Add this code to tell the application where to look for 'staticFiles'

STATICFILES\_DIRS = [

os.path.join(BASE\_DIR, 'staticFiles')

] #Note BASE\_DIR is somewhere in the SETTINGS file.

**CREATE YOUR FIRST TEST MODEL:**

from django.db import models  
  
class testTable(models.Model):

employee\_id = models.AutoField(primary\_key=True)

person\_name = models.CharField( max\_length = 200 )

gender = models.CharField( max\_length = 200 )

salary = models.IntegerField (default = 0 )

def \_\_str\_\_(self):

return " ID of the person: " + str(self.employee\_id) + " Name of the person : " + str(self.person\_name) + " Gender of the person : " + str(self.gender) + " Salary of the person : " + str(self.salary)

**INSTALL YOUR MODELS:**

python manage.py makemigrations

python manage.py migrate

**MAKE MODELS AVAIABLE IN ADMIN:**  
**C:\espacios\API-Test-Project\core\baseApp\admin.py>**

################################

# REGISTER YOUR MODELS HERE

################################

from django.contrib import admin

from .models import testTable

admin.site.register(testTable)

**LOAD TEST DATA USING LOADER SCRIPT:**

C:\espacios\API-Test-Project\core>python manage.py runscript loader

**CREATE SUPER USER:**

C:\espacios\API-Test-Project\core>python manage.py createsuperuser

**--- At this point you should be able to see data on admin ---**

**SETTUP YOUR TEMPLATES AND A TEST JAVASCRIPT D3 FILE**

C:\espacios\API-Test-Project\core\staticFiles\javascriptFiles\ vim barChart.js  
--------------------------------------------------------  
Write this test code in your file to test the connection.  
--------------------------------------------------------  
console.log(d3);

C:\espacios\API-Test-Project\core\baseApp\ mkdir templates

C:\espacios\API-Test-Project\core\baseApp\templates\ mkdir baseApp

C:\espacios\API-Test-Project\core\baseApp\templates\baseApp\ vim chart1.html

--------------------------------------------------------  
Write this test code in your file to test the connection.  
--------------------------------------------------------  
{% load static %}

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>TEST BAR CHART</title>

<script src="https://d3js.org/d3.v5.min.js"></script>

</head>

<body>

{% block content %}{% endblock %}

</body>

<script src="{% static 'javascriptFiles/barChart.js' %}"></script>

</html>

**CREATE VIEW FOR THE TEST HTML:**  
**C:\espacios\API-Test-Project\core\baseApp\views.py**

#############################################################################

#RENDERS

#############################################################################

from django.shortcuts import render

from django.shortcuts import redirect

#RENDER HOME PAGE

# -------------------------

def renderChart1(request):

return render(request, 'baseApp/chart1.html')

**CREATE URL.PY FOR THE TEST HTML:**  
**C:\espacios\API-Test-Project\core\urls.py**

####################################################

# THIS FILE CALLS EACH INDIVISUAL URL IN EVERY APP

####################################################

from django.urls import include, path

from baseApp import views

urlpatterns = [

path('', include('baseApp.urls')),

]

**C:\espacios\API-Test-Project\core\baseApp\ vim urls.py**

#################################################

# THIS FILE CALLS THE VIEWS AND HANDLES THE URLS

#################################################

from django.urls import path, include

from . import views

app\_name = 'baseApp'

urlpatterns = [

# CONTAINERS

path('', views.renderChart1, name='chart1'),

]

**SETUP DJANGO-REST FRAMEWORK FOR API CALLS:  
  
  
  
SAMPLE SERIALIZER (DJANGO REST FRAMEWORK MODEL)  
C:\espacios\API-Test-Project\core\baseApp\ mkdir serializers.py**

################################################

#-----------------------------------------------

#ACCESSING THE API

#-----------------------------------------------

'''

HTTP Methods | OPERATIONS

POST | CREATE

GET | READ (list or detailed View)

PUT or PATCH | UPDATE

DELETE | DELETE

'''

################################################

from rest\_framework import serializers

from baseApp.models import testTable

class testTableSerializer(serializers.HyperlinkedModelSerializer):

'''serializer for the test\_table object'''

class Meta:

model = testTable

fields = ['employee\_id','gender','salary']

**CONFIGURE THE API VIEW:**

from baseApp.models import testTable

from baseApp.serializers import testTableSerializer

from rest\_framework import viewsets

from rest\_framework import permissions

class CreateTestTableView(viewsets.ModelViewSet):

queryset = testTable.objects.all() #This is the querry, its taking all the items.

serializer\_class = testTableSerializer

permission\_classes = [permissions.IsAuthenticated]

**CONFIGURE THE ROUTER:  
  
C:\espacios\API-Test-Project\core\urls.py**

from django.urls import include, path

**from rest\_framework import routers**

from baseApp import views

router = routers.DefaultRouter()

router.register(r'renderCreateTestTable', views.CreateTestTableView)

urlpatterns = [

path('', include('baseApp.urls')),

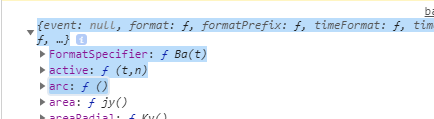
**path('api', include(router.urls)),**

**path('api-auth/', include('rest\_framework.urls', namespace='rest\_framework'))**

]

**At this point you should have access to:**  
  
THE API CALL: <http://127.0.0.1:8000/apirenderCreateTestTable/>

A BLANK HOME PAGE: <http://127.0.0.1:8000/> If you console log here you should be able to see D3



**Django-LiveReload-Server:**

https://github.com/tjwalch/django-livereload-server

**Query with shell:**

**Django Queries.**

[**https://docs.djangoproject.com/en/3.1/ref/models/querysets/#id4**](https://docs.djangoproject.com/en/3.1/ref/models/querysets/#id4)

**Use Django-filter to build forms.**

**https://www.youtube.com/watch?v=nle3u6Ww6Xk&t=75s**

python manage.py shell

from django.db import models

from baseApp.models import unemployment\_Event as tb

All:

q1 = tb.objects.all()

Filter:

Q2 = tb.objects.all(gender= Males)