https://www.youtube.com/watch?v=Pwwz4_AvHDU&t=1080s

Python Django + PostgreSQL | REST API Tutorial

1. First create folder
2. pip install   virtualenvwrapper-win
3. mkvirtualenv apiwork
4. pip install django
5. django-admin startproject   firstproject
6. pip install djangorestframework
7. pip install django-cors-headers
8. Inside project
9. Python manage.py startapp apiapp
10. Now register the module

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

CORS_ORIGIN_ALLOW_ALL = True

MIDDLEWARE = [
    'corsheaders.middleware.CorsMiddleware',
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

11. Create models.py from app

```
from django.db import models

# Create your models here.


class Department(models.Model):
```

```python
    DepartmentId = models.AutoField(primary_key=True)
    DepartmentName = models.CharField(max_length=500)


class Employees(models.Model):
    EmployeeId = models.AutoField(primary_key=True)
    EmployeeName = models.CharField(max_length=100)
    Department = models.CharField(max_length=500)
    DateOfJoining = models.DateField()
    PhotoFileName = models.CharField(max_length=500)
```

12. Data base adapter install

-> pip install psycopg2

13. Create database now.

12. Database settings.py from project

```python
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'apidb',
        'USER': 'postgres',
        'PASSWORD': 'saikat',
        'HOST': 'localhost',
        'PORT': '5432'
    }
}
```

14. python   manage.py   makemigrations

15. python manage.py   migrate

16. create serializers.py file into app

```python
from rest_framework import serializers
from .models import Department, Employees


class DepartmentSerializers(serializers.ModelSerializer):
    class Meta:
        model = Department
        fields = ('DepartmentId', 'DepartmentName')


class EmployeeSerializers(serializers.ModelSerializer):
    class Meta:
        model = Employees
        fields = ('EmployeeId', 'EmployeeName', 'Department',
                  'DateOfJoining', 'PhotoFileName')
```

17. Create API methods in views.py from app

```python
from django.shortcuts import render
from django.views.decorators.csrf import csrf_exempt
from rest_framework.parsers import JSONParser
from django.http.response import JsonResponse

from .models import Department, Employees
from apiapp.serializers import DepartmentSerializers, EmployeeSerializers

from django.core.files.storage import default_storage

# Create your views here.


@csrf_exempt
def departmentApi(request, id=0):
    # ------------Get Department Data-----------
    if request.method == 'GET':
        departments = Department.objects.all()
        department_serializer = DepartmentSerializers(departments, many=True)
        return JsonResponse(department_serializer.data, safe=False)
    # ------------Save Department Data-----------
    elif request.method == 'POST':
        department_data = JSONParser().parse(request)
        departments_serializer = DepartmentSerializers(data=department_data)
        if departments_serializer.is_valid():
            departments_serializer.save()
            return JsonResponse("Added Successfully", safe=False)
        return JsonResponse('Failed to Add', safe=False)
    # ------------Update Department Data-----------
    elif request.method == 'PUT':
        department_data = JSONParser().parse(request)
        department = Department.objects.get(
            DepartmentId=department_data['DepartmentId'])
        departments_serializer = DepartmentSerializers(
            department, data=department_data)
        if departments_serializer.is_valid():
            departments_serializer.save()
            return JsonResponse("Update Successfully", safe=False)
        return JsonResponse('Failed to Update', safe=False)
    # ------------Delete Department Data-----------
    elif request.method == 'DELETE':
        department = Department.objects.get(DepartmentId=id)
        department.delete()
        return JsonResponse('Deleted Successfully', safe=False)


@csrf_exempt
```

```python
def employeeApi(request, id=0):
    # ------------Get Employee Data-----------
    if request.method == 'GET':
        employees = Employees.objects.all()
        employee_serializer = EmployeeSerializers(employees, many=True)
        return JsonResponse(employee_serializer.data, safe=False)
    # ------------Save Employee Data-----------
    elif request.method == 'POST':
        employee_data = JSONParser().parse(request)
        employees_serializer = EmployeeSerializers(data=employee_data)
        if employees_serializer.is_valid():
            employees_serializer.save()
            return JsonResponse("Added Successfully", safe=False)
        return JsonResponse('Failed to Add', safe=False)
    # ------------Update Employee Data-----------
    elif request.method == 'PUT':
        employee_data = JSONParser().parse(request)
        employee = Employees.objects.get(
            EmployeeId=employee_data['EmployeeId'])
        employees_serializer = EmployeeSerializers(
            employee, data=employee_data)
        if employees_serializer.is_valid():
            employees_serializer.save()
            return JsonResponse("Updated Successfully", safe=False)
        return JsonResponse('Failed to Update', safe=False)
    # ------------Delete Employee Data-----------
    elif request.method == 'DELETE':
        employee = Employees.objects.get(EmployeeId=id)
        employee.delete()
        return JsonResponse('Deleted Successfully', safe=False)


@csrf_exempt
def SaveFile(request):
    file = request.FILES['file']
    file_name = default_storage.save(file.name, file)
    return JsonResponse(file_name, safe=False)
```

18. Create urls.py in app

```python
from django.conf.urls import url
from apiapp import views

from django.conf.urls.static import static
from django.conf import settings

urlpatterns = [
    url(r'^department$', views.departmentApi),
    url(r'^department/([0-9]+)$', views.departmentApi),

    url(r'^employee$', views.employeeApi),
    url(r'^employee/([0-9]+)$', views.employeeApi),

    url(r'^employee/savefile', views.SaveFile)
]+static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

19. include urls.py from app into main urls.py in project

```
from django.contrib import admin
from django.urls import path

from django.conf.urls import url, include

urlpatterns = [
    path('admin/', admin.site.urls),
    url(r'^', include('apiapp.urls'))
]
```

20. Now api call from postman.

Save Image file:

1. Create photo folder into project
2. Settings.py from project

```

from pathlib import Path
import os

BASE_DIR = Path(__file__).resolve(strict=True).parent.parent
MEDIA_URL = '/photos/'
MEDIA_ROOT = os.path.join(BASE_DIR, "photos")

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent
```

3. Go to views.py added here file api method from app

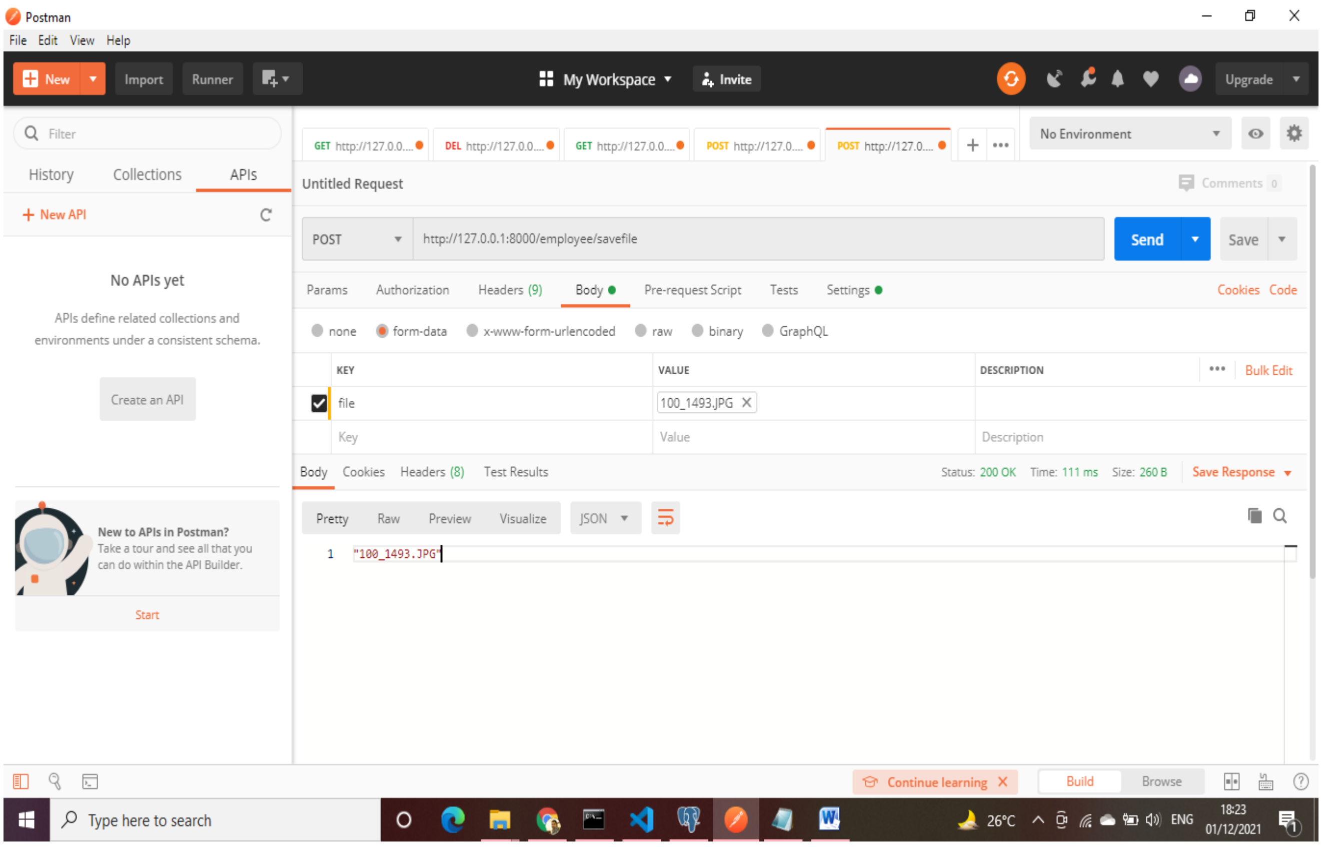
```
from django.shortcuts import render
from django.views.decorators.csrf import csrf_exempt
from rest_framework.parsers import JSONParser
from django.http.response import JsonResponse

from .models import Department, Employees
from apiapp.serializers import DepartmentSerializers, EmployeeSerializers
```

```python
from django.core.files.storage import default_storage
```

```python
@csrf_exempt
def SaveFile(request):
    file = request.FILES['file']
    file_name = default_storage.save(file.name, file)
    return JsonResponse(file_name, safe=False)
```

Must be the call from postman. Use parameter file



POST http://127.0.0.1:8000/employee/savefile

Body: form-data

KEY	VALUE	DESCRIPTION
file	100_1493.JPG	

Status: 200 OK Time: 111 ms Size: 260 B

```
"100_1493.JPG"
```