**Simple Steps to Create a REST API Using Django**

**REST Framework in Win10**

**1st Download & Install : for database software**

[**https://www.postgresql.org/ftp/pgadmin/pgadmin4/v9.1/windows/**](https://www.postgresql.org/ftp/pgadmin/pgadmin4/v9.1/windows/) **- Setup,**

**2nd Create new database, Username & Password**

**3rd Run in Background**

**Download Python Form Office Site**

**https://www.python.org/ftp/python/3.13.2/python-3.13.2-amd64.exe**

**Install File & Restart OS Windows**

**Download VS Code & Install : Open VS Code**

*Run : python –version*

*Run : django-admin –version*

**If Not Present, Then you need to install all packages**

*Run :* pip freeze List Show packages || Not Available then install all packages

*Run :* **pip install -r requirements.txt (if Existing Project then Run important)**

*Run :* pip install django

*Run :* pip install djangorestframework

*Run : pip install django\_rest\_framework*

*Run :* pip install psycopg2

*Run :* python manage.py createsuperuser | **for developer id**

***If Security Error Show Then*** *Run : Set-ExecutionPolicy Unrestricted -Scope Process*

**Create main project Folder (anything name)**

*Run :* python -m venv venv (virtue Env cerate )

**1st Create Main Project :** *Run :* **django-admin startproject app**

**Cd app**

**2nd Create Project Module or app :** *Run :***python manage.py startapp update\_api**

**Under app Folder (api-name-folder or app name)**

*Run : cd* **api-name-folder**

*Run :* venv\Scripts\activate (active)

GoTo main app folder decretory, open **setting.py** and goto section config #

**Application definition- install apps name**

Under add 2 line simple.

  'update\_api',

    'rest\_framework',

save & exit;

After go to database setting: use default

 'ENGINE': 'django.db.backends.postgresql',

        'NAME':'api\_health',

        'USER':'postgres',

        'PASSWORD':'admin',

        'HOST': 'localhost',  # Change if using a remote DB

        'PORT': '5432',  # Default PostgreSQL port

After database credentials add save & exit;

Open **urls.py** And add this, basically api folder root

from django.contrib import admin

from django.urls import path,include

urlpatterns = [

    path('admin/', admin.site.urls),

    path("api/", include("update\_api.urls")),

]

After save & exit; All setting are done in main project.

**Now Open Api Folder or app folder**

**Open apps.py and view code**

from django.apps import AppConfig

class DeleteApiConfig(AppConfig):

    default\_auto\_field = 'django.db.models.BigAutoField'

    name = 'update\_api'

After save & exit;

**open models.py and Create**

from django.db import models

class User(models.Model):

    name = models.CharField(max\_length=10)

    description = models.TextField()

    def \_\_str\_\_(self):

        return self.name

After save & exit;

**Create a black serializers.py page …**

**open serializers.py and edit**

from rest\_framework import serializers

from .models import User

class ItemSerializer(serializers.ModelSerializer):

     class Meta:

          model = User

          fields = ['name','description']

After save & exit;

**open views.py and edit**

from rest\_framework import status

from rest\_framework.response import Response

from rest\_framework.views import APIView

from .models import User

from .serializers import ItemSerializer

from django.db import transaction

class ItemUpdate(APIView):

    def put(self, request):

        # Handling bulk update

        data = request.data

        with transaction.atomic():

                user, created = User.objects.update\_or\_create(

                    id=data.get('id'),

                    defaults={

                        'name': data.get('name'),

                        'description': data.get('description')

                    }

                )

        return Response({'status': 'Updated products successfully'}, status=status.HTTP\_200\_OK)

        #Delete the item from the database

        item.delete()

        #Return a 284 response

        return Response({"msg": "Delete "})

After save & exit;

**open urls.py and edit**

from django.urls import path

from .views import ItemUpdate

urlpatterns = [

    path('products/bulk/', ItemUpdate.as\_view(), name='product-bulk'),

]

… After save & exit;

**Open admin.py and edit (if You Want Show in Django Admin Panel, Otherwise)**

from django.contrib import admin

from .models import User

# Register your models here.

admin.site.register(User)

After save & exit;

cd app

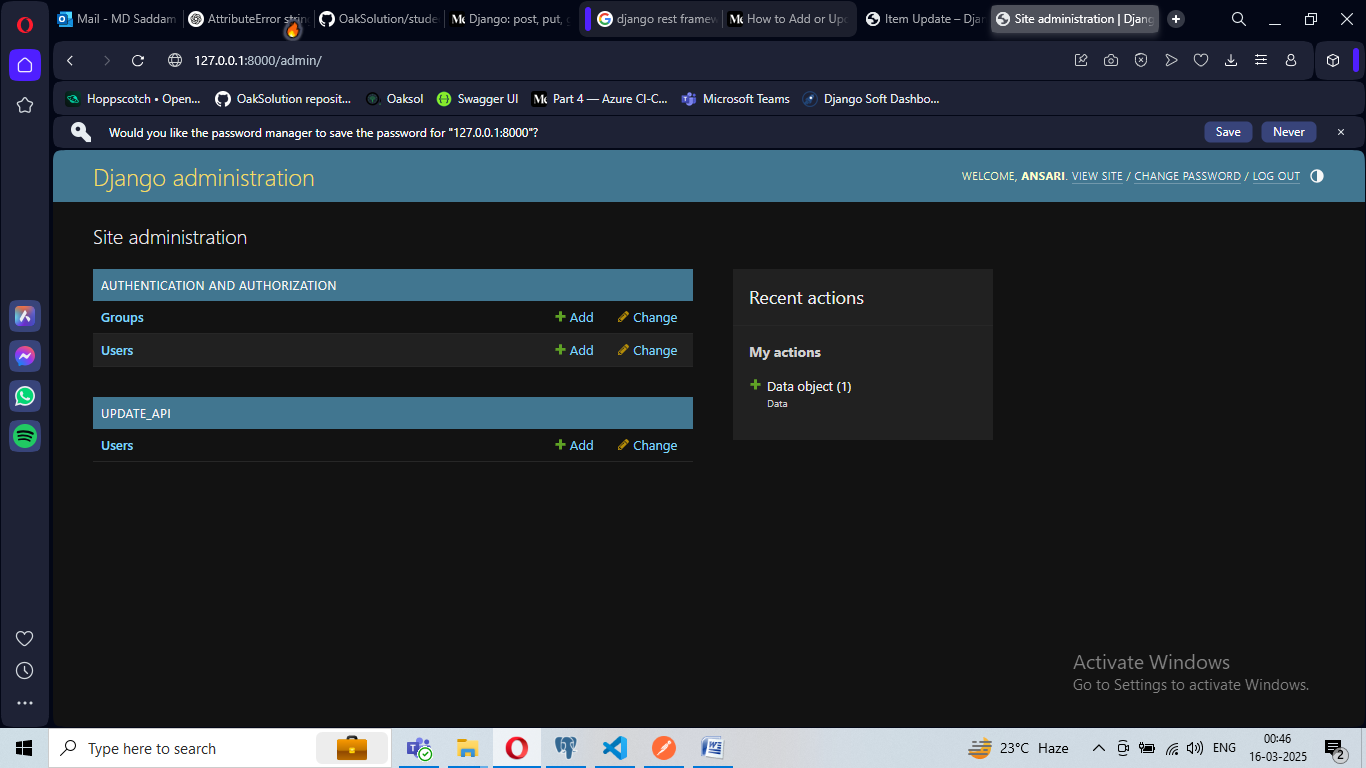
python manage.py makemigrations

python manage.py migrate **every work after**

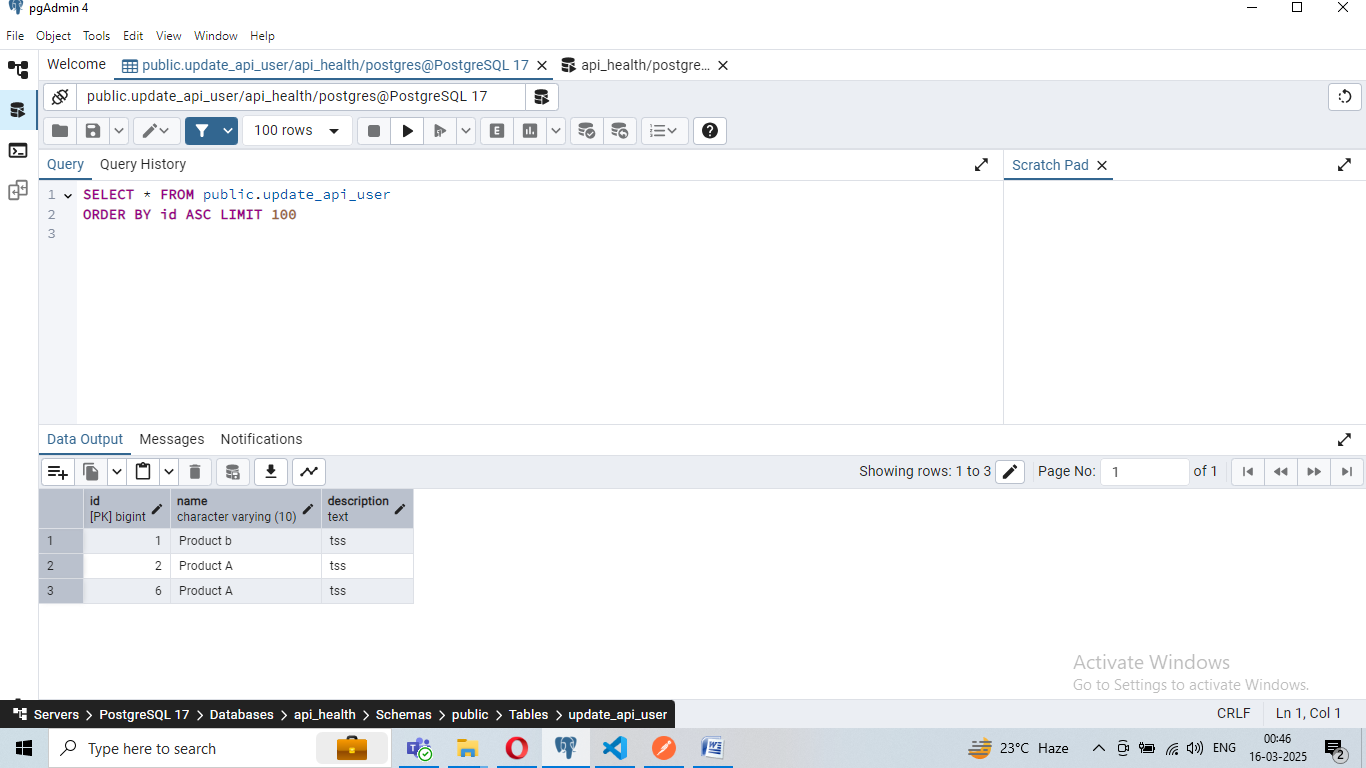
python manage.py runserver 9000 – **Get web address with custom port**

**after run open browser**

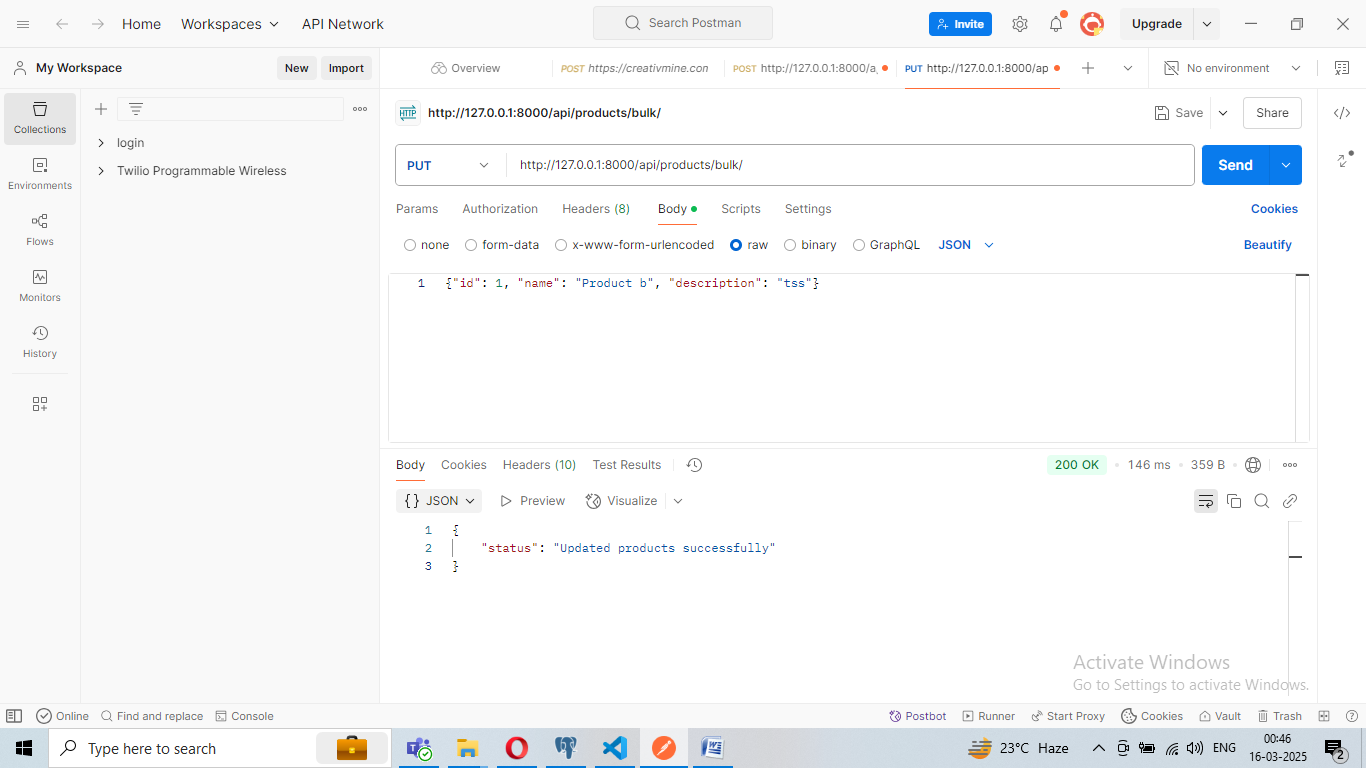
[**http://127.0.0.1:9000/api**](http://127.0.0.1:9000/api)

****

**Admin view mode**

****

**Show data in table under database .**

****

**If not user in database. Then Insert New Data .**

**Ref : https://medium.com/django-unleashed/how-to-add-or-update-multiple-data-entries-in-django-rest-api-with-a-script-42838a08e9fe**