**DJANGO REST API NOTES**

1. First we need to install django rest framework

--------> pip install django djangorestframework

1. Make django project  
   --------> django-admin startproject api\_projectlearning
2. Make app   
   --------> python manage.py startapp api
3. We need to register both app and restframework in settings.py

--------> 'rest\_framework', 'api'

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'rest\_framework',

    'api'

]

1. Make Modelsin models.py

from django.db import models

# Create your models here.

class User(models.Model):

    age = models.IntegerField()

name = models.CharField(max\_length=100)

    def \_\_str\_\_(self):

        return self.name

1. Make a file in the app folder called serializer.py  
   --------> This helps us transform model into Json data that can be accessed in our API

from rest\_framework import serializers

from .models import User

class UserSerializer(serializers.ModelSerializer):

    class Meta:

        model = User

        fields = '\_\_all\_\_'

1. In views.py -> GET resquest

from rest\_framework.decorators import api\_view

from rest\_framework.response import Response

from rest\_framework import status

from .models import User

from .serializer import UserSerializer

# Create your views here.

@api\_view(['GET'])

def get\_user(request):

    return Response(UserSerializer({'name': 'pedro', 'age': 23}).data)

1. In urls.py in app

from django.urls import path

from .views import get\_user

urlpatterns = [

    path('users/', get\_user, name='get\_user')

]

1. In urls.py in project

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

    path('admin/', admin.site.urls),

    path('api/', include('api.urls'))

]

1. POST Request

@api\_view(['POST'])

def create\_user(request):

    serializer = UserSerializer(data=request.data)

    if serializer.is\_valid():

        serializer.save()

        return Response(serializer.data, status=status.HTTP\_201\_CREATED)

    return Response(serializer.errors, status=status.HTTP\_400\_BAD\_REQUEST)

1. CRUD operations that involve index (<int:pk>)

Urls.py

from django.urls import path

from .views import get\_users, create\_user, user\_detail

urlpatterns = [

    path('users/', get\_users, name='get\_user'),

    path('users/create', create\_user, name='create\_user'),

    path('users/<int:pk>', user\_detail, name='user\_detail')

]

Views.py

@api\_view(['GET', 'PUT', 'DELETE'])

def user\_detail(request, pk):

    try:

        user = User.objects.get(pk=pk)

    except User.DoesNotExist:

        return Response(status=status.HTTP\_404\_NOT\_FOUND)

    if request.method == 'GET':

        serializer = UserSerializer(user)

        return Response(serializer.data)

    elif request.method == 'PUT':

        serializer = UserSerializer(user, data=request.data)

        if serializer.is\_valid():

            serializer.save()

            return Response(serializer.data)

        return Response(serializer.errors, status=status.HTTP\_400\_BAD\_REQUEST)

    elif request.methos == 'DELETE':

        user.delete()

        return Response(status=status.HTTP\_204\_NO\_CONTENT)