IMDB Clone API

1. Create folder
2. Create virtual env inside the project
   1. Python -m venv env
3. Go inside the environment
   1. Env\Scripts\activate
4. Pip install Django
5. Pip install djangorestframework
6. Django-admin startproject watchmate
7. Cd watchmate
8. Django-admin startapp watchlist\_app
9. Go to the settings.py and register app and rest\_framework

INSTALLED\_APPS = [

    'watchlist\_app',

    'rest\_framework',

]

1. Create API folder inside the watchlist\_app.
2. Create 3 files.
   1. Urls.py
   2. Views.py
   3. Serializers.py

(File name should be same as here. We create new api folder bcz we want all files togher.)

1. Create a model – models.py

from django.db import models

class WatchList(models.Model):

    title = models.CharField(max\_length=50)

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

    active = models.BooleanField(default=False)

    created\_date = models.DateTimeField(auto\_now\_add=True)

    def \_\_str\_\_(self):

        return self.title

1. Register the model in admin.py

from watchlist\_app.models import WatchList

admin.site.register(WatchList)

1. Make the migrations for WatchList app

Python manage.py makemigrations

Python manage.py migrate

1. Create the super user

Vishal

vishal

**Serializers**

Serializers allow complex data such as querysets and model instances to be converted to native Python datatypes that can then be easily rendered into JSON, XML or other content types. Serializers also provide deserialization, allowing parsed data to be converted back into complex types, after first validating the incoming data.

There are many types of Serializers we are first using Serializer then we use ModelSerializer.

1. Make the serializer for WatchList model

from rest\_framework import serializers

class WatchListSerializer(serializers.Serializer):

    id = serializers.IntegerField(read\_only=True)

    title = serializers.CharField()

    storyline = serializers.CharField()

    active = serializers.BooleanField(default=False)

    created\_date = serializers.DateTimeField()

Here I did not use Meta class bcz I am using Function Based View.

1. create the view for the WatchList model

from watchlist\_app.models import WatchList

from watchlist\_app.api.serializers import WatchListSerializer

from rest\_framework.response import Response

from rest\_framework.decorators import api\_view

@api\_view()

def WatchListView(request):

    wathlist = WatchList.objects.all()

    serializer = WatchListSerializer(wathlist, many=True)

    return Response(serializer.data)

Here why we use many=True

If we have multiple entries or we can say multiple movies so we have to define

Here We are using @api\_view() basically this is a decorator and default it contain get types of method.

Suppose if we want to add new movie then I have to use post method so we can defined inside the @api\_view()

@api\_view([‘GET’,’POST’])

1. create the main url and app url

main.url

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

    path('admin/', admin.site.urls),

    path('watch/', include('watchlist\_app.api.urls')),

]

Watchlist\_app.url

from django.urls import path

from watchlist\_app.api.views import WatchListView

urlpatterns = [

    path('', WatchListView, name='watchlist'),

]

1. Enter the data in watchlist Model through admin panel.
2. Create the Detail View

@api\_view()

def WatchListDetail(request, pk):

    watchlist = WatchList.objects.get(pk=pk)

    serializer = WatchListSerializer(watchlist)

    return Response(serializer.data)

1. Create the url.
2. Import the WatchListDetail View.

    path('<int:pk>/', WatchListDetail, name='watchlistDetail'),

Now I want to add the new movie through Browsable API only (thru Browser Only).

First we have to tell decorator that we want to post the request from Browser

Create the post method in WatchListView app

Create the create method in serializer.

\*\*validated\_data -> bcz it contains all the fields name.

Views.py

@api\_view(['GET','post'])

def WatchListView(request):

    if request.method == 'GET':

        wathlist = WatchList.objects.all()

        serializer = WatchListSerializer(wathlist, many=True)

        return Response(serializer.data)

    if request.method == 'POST':

        serializer = WatchListSerializer(data=request.data)

        if serializer.is\_valid():

            serializer.save()

            return Response(serializer.data)

        else:

            return Response(serializer.errors)

Serializers.py

class WatchListSerializer(serializers.Serializer):

    id = serializers.IntegerField(read\_only=True)

    title = serializers.CharField()

    storyline = serializers.CharField()

    active = serializers.BooleanField(default=False)

    created\_date = serializers.DateTimeField()

    def create(self, validated\_data):

        return WatchList.objects.create(\*\*validated\_data)