

## DJANGO REST FRAME WORK

### REQUIREMENTS FOR DJANGO REST FRAME WORK

- 1.django application
- 2.python version (3.5, 3.6,.....)
- 3.Django version(1.11, .....)

FIRST YOU NEED TO INSTALL DJANGO REST FRAME WORK THROUGH COMMAND PROMPT OF YOUR WINDOW BY WRITING;

Pip install djangorestframework, THEN PRESS ENTER.

NOW CREATE A PROJECT WITHIN A FOLDER WHICH YOU INSTALLED DJANGO REST FRAMEWORK BEFORE AND APP AND THEN OPEN IT WITH CODE EDITOR AS USUAL.

AFTER THAT IF YOU WANT TO USE DJANGO REST FRAMEWORK IN YOUR PROJECT THE FIRST STEP IS TO INSTALL IT IN SETTING.PY UNDER INSTALLED APP AS;

`'rest_framework',`

IN OUR EXAMPLE WE ARE CREATING A MOVIES API, SO THE NEXT STEP IS GOING TO MODELS.PY AND CREATE A MODELS WHICH WILL HOLD MOVIES DATA AS USUAL;

IN YOUR MODEL YOU CAN HAVE;

`Class MovieData(models.Model):`

`Name= models.CharField(max_length=200)`

`Duration = models.FloatField()`

`Rating = models.FloatField()`

NOW MAKE MIGRATIONS TO SERVE THEM.

NOW YOU CAN CREATE SUPER USER TO ADD INFORMATION TO YOUR COLUMNS THROUGH ADMIN PANEL.

AFTER ADDING THOSE INFORMATIONS NOW IS TIME TO CREATE YOUR OWN API TO SHARE THOSE DATA THROUGH AN API.



IN ORDER TO DO THAT THE CONCEPT OF SERIALIZATION COMES

THIS IS A METHOD THAT ALLOWS US TO TAKE DATA FROM THE MODEL AND TO PUT OUT IN AN API.

HOW TO PERFORM SERIALIZATION

IN ORDER TO PERFORM SERIALIZATION GO TO YOUR APP AND CREATE A NEW FILE CALLED `serializers.py`, within a `serializers.py` we are creating our serializer as shown below;

```
from rest_framework import serializers
```

```
from .models import MovieData
```

TO CREATE SERIALIZER WE USE CLASS;

```
class MovieSerializer(serializers.ModelSerializer):
```

CREATE A META TO SHOW A MODEL YOU NEED TO SERIALIZER, AND FIELD YOU WANT TO MENTION IN API RESPONSE.

```
    class Meta:
```

```
        model = MovieData
```

```
        fields = ['id', 'name', 'duration', 'rating']
```

NOW CREATING A VIEW TO DISPLAY THE DATA GENERATED BY `MovieSerializer`.

IN `VIEWS.PY`

```
from rest_framework import ViewSets
```

```
from .serializers import MovieSerializer
```

```
from .models import MovieData
```

BY USING CLASS BASED VIEW

```
class MovieViewSet(ViewSets.ModelViewSet):
```



```
Queryset = MovieData.objects.all()
```

```
Serializer_class = MovieSerializer
```

NOW THE NEXT STEP IS TO CREATE A URL FOR OUR VIEW

IN URLS.PY OF THE PROJECT NOT AN APP

```
From Django.urls import path, include
```

```
From rest_framework import routers
```

```
From movies.views import MovieViewSet
```

NOW IS TIME TO CREATE A ROUTERS BELOW THE ABOVE CODES WRITE;

```
Router = routers.DefaultRouter()
```

NOW REGISTER A ROUTER AND ADD THE PATH NAME EG:movies

```
Router.register('movies', MovieViewSet)
```

NOW INCLUDE THE URLS IN URLSPATTERNS

```
Path(' ', include(router.urls)),
```

NOW RUN THE SERVER TO TEST IF APP IS RUNNING BY WRITING

```
/Movies/
```

UNTILL HERE IS AN EXAMPLE OF SIMPLE API USING DJANGO REST FRAME WORK

END END END

ADDING API END POINT TO OUR REST API

EXAMPLE IF A USER NEEDS TO REQUEST FOR ACTION MOVIES SHOULD GET ONLY ACTION MOVIES AND SOMEONE ELSE NEEDS TO REQUEST FOR COMEDIAN MOVIES SHOULD GET COMEDIAN MOVIES ONLY.

HOW TO IMPLEMENT THIS

SO INORDER TO CATOGORIES MOVIES, YOU MUST ADD A NEW FIELD IN MODELS.PY MAY BE TYP;

```
typ = models.CharField(max_length=200,default = 'action')
```

MAKE MIGRATIONS AGAIN

AFTER THAT YOU NEED TO CREATE A SPECIAL KIND OF VIEW IN VIEWS.PY

EXAMPLE YOU NEED TO CREATE A VIEW FOR ACTION MOVIES, SO

```
Class ActionViewSet(ViewSets.ModeViewSet):
```

```
    Queryset = MovieData.objects.filter(typ='action')
```

```
    Serializer_class = MovieSerializer
```

NOW IS A TIME TO SET URLS PATTERN FOR THIS VIEW

IN URLS.PY OF THE PROJECT

IN THIS CASE THE FIRST THING IS TO CHANGE THE TYPE OF THE ROUTER, SO IN THIS CASE WE ARE GOING TO USE SIMPLE ROUTER, HENCE;

```
Router = router.SimpleRouter()
```

NOW REGISTER A VIEW BELOW THE ABOVE LINE

```
Router.register('action', ActionViewSet)
```

MAKE SURE YOU ARE IMPORTING ACTIONVIEWSET BEFORE

AND THEN RUNSERVER AGAIN BY TYPING;

/Action/ INSTEAD OF movies THIS TIME

BUT THE OUPUT WILL DISPLAY ALL MOVIES BCZ BY DEFAULTY YOU MADE ALL MOVIES TO BE ACTION MOVIES, ALSO INSIDE A FORM A FIELD TYP WON'T BE SEEN SO IN ORDER TO SEE THIS FIELD GO AGAIN TO THE SERIALIZERS.PY UNDER FIELD SECTION ADD TYP.

SO IN ORDER TO CREATE CATEGORIES FOR MOVIES EG: COMEDIAN MOVIES YOU MUST REPEAT THIS PROCEDURES OF CREATING ACTION MOVIES TO CREATE COMEDIAN MOVIES CATEGORY.

```
Class ComedyViewSet(ViewSets.ModalViewSet):
```

```
    Queryset = MovieData.objects.filter(typ='comedy')
```

```
    Serializer_class = MovieSerializer
```

AND THEN CREATE URL AS USUAL

IN URLS.PY OF THE PROJECT

IN THIS CASE THE FIRST THING IS TO CHANGE THE TYPE OF THE ROUTER, SO IN THIS CASE WE ARE GOING TO USE SIMPLE ROUTER, HENCE;

```
Router = router.SimpleRouter()
```

NOW REGISTER A VIEW BELOW THE ABOVE LINE

```
Router.register('comedy, ComedyViewSet)
```

MAKE SURE YOU ARE IMPORTING COMEDYVIEWSET BEFORE

AND THEN RUNSERVER AGAIN BY TYPING;

/comedy/ INSTEAD OF movies THIS TIME

END OF THE API ENDPOINT

HOW TO ADD IMAGES FIELD IN API

THE FIRST STEP IS TO INSTALL A PACKAGE CALLED PILLOW, AND THEN GO TO MODELS .PY ADD THE FIELD CALLED IMAGE;

```
Image = models.ImageField(upload_to='images/', default="images/None/NoImg.jpg")
```

THEN MAKE MIGRATIONS AGAIN

IN ORDER YOU NEED TO USE IMAGE IN API YOU MUST SERIALIZE IT ALSO, HENCE IN ORDER TO SERIALIZE IMAGE GO TO THE SERIALIZERS.PY FILE BTN THE CLASS MovieSerializer AND CLASS Meta WRITE;

```
Image = serializers.ImageField(max_length=None, use_url=True)
```

AND THEN ADD THE FIELD IMAGE UNDER FIELD SECTION.

AND THEN FOLLOW OTHER NORMAL PROCEDURES LIKE CREATING A FOLDER UNDER APP

CALLED MEDIA, ALSO ADDING MEDIA ROOT IN SETTING.PY

BELOWA STATIC\_URL = '/static/' ADD THE FOLLOWING;

```
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

```
MEDIA_URL = '/media/'
```

AND THEN IN URLS.PY OF THE PROJECT

```
From Django.conf.urls.static import static
```

```
From Django.conf import settings
```

```
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

NOW RUN THE SERVER TO SEE THE RESULTS.....

THIS IS THE END OF DJANGO REST FRAME WORK FOR BEGINNERS