**REST API (Django Rest Framework-DRF)**

**Session-20**

**Search Filter:**

* The SearchFilter class supports simple single query parameter based searching, and is based on the Django admin's search functionality.
* When in use, the browsable API will include a SearchFilter control.
* The SearchFilter class will only be applied if the view has a search\_fields attribute set.
* The search\_fields attribute should be a list of names of text type fields on the model, such as CharField or TextField

**Settings.py :**

INSTALLED\_APPS = [  
 'django.contrib.admin',  
 'django.contrib.auth',  
 'django.contrib.contenttypes',  
 'django.contrib.sessions',  
 'django.contrib.messages',  
 'django.contrib.staticfiles',  
 'myapp.apps.MyappConfig',  
 'rest\_framework',

'django\_filters',

]

**models.py :**

from django.db import models  
  
*# Create your models here.*class Student(models.Model):  
 name=models.CharField(max\_length=20)  
 sid=models.IntegerField()  
 saddress=models.CharField(max\_length=20)  
 trainedby=models.CharField(max\_length=20)

**admin.py :**

from django.contrib import admin  
from myapp.models import Student  
*# Register your models here.*@admin.register(Student)  
class StudentAdmin(admin.ModelAdmin):  
 list\_display = ['id','name','sid','saddress','trainedby']

**Now go to terminal and then type the following commands**

* Python manage.py makemigrations
* Python manage.py migrate
* Python manage.py createsuperuser
* Now run server: python manage.py runserver
* Now go to browser: <http://127.0.0.1:8000/admin>
* Insert few records
* Create a new python file with the name serializers.py.

**serializer.py:**

from rest\_framework import serializers  
from myapp.models import Student  
  
class StudentSerializer(serializers.ModelSerializer):  
 class Meta:  
 model=Student  
 fields=['id','name','sid','saddress','trainedby']

**views.py :**

from django.shortcuts import render  
from myapp.serializers import StudentSerializer  
from rest\_framework.generics import ListAPIView  
from myapp.models import Student  
from rest\_framework.filters import SearchFilter  
  
*# Create your views here.*class StudentList(ListAPIView):  
 queryset = Student.objects.all()  
 serializer\_class = StudentSerializer  
 filter\_backends = [SearchFilter]  
 search\_fields=['saddress']  
 *#search\_fields = ['saddress','name']  
 #search\_fields=['^name'] #startswith  
 #search\_fields = ['=name'] #exact match*

**urls.py:**

from django.contrib import admin  
from django.urls import path  
from myapp import views  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('studentapi/',views.StudentList.as\_view()),  
]

* Now run server: python manage.py runserver
* Now go to browser: <http://127.0.0.1:8000/admin/studentapi/?search=hyderabad>
* By default, the search parameter is named 'search', but this may be overridden with the SEARCH\_PARAM setting.
* Go to settings.py and write the following code

REST\_FRAMEWORK={  
 *#'SEARCH\_PARAM':'search' #default* 'SEARCH\_PARAM':'s'  
}

* Now go to browser: <http://127.0.0.1:8000/admin/studentapi/?s=hyderabad>

**Ordering Filter:**

* The OrderingFilter class supports simple query parameter controlled ordering of results.
* By default, the query parameter is named 'ordering', but this may by overridden with the ORDERING\_PARAM setting.

**Views.py:**

from django.shortcuts import render  
from myapp.serializers import StudentSerializer  
from rest\_framework.generics import ListAPIView  
from myapp.models import Student  
from rest\_framework.filters import OrderingFilter  
  
*# Create your views here.*class StudentList(ListAPIView):  
 queryset = Student.objects.all()  
 serializer\_class = StudentSerializer

filter\_backends = [OrderingFilter]  
*#ordering\_fields=['name'] #ascending order  
#ordering\_fields = ['-name'] #descending order*ordering\_fields=['name','saddress'] *#multiple fields ordering*

* Now run server: python manage.py runserver
* Now go to browser: http://127.0.0.1:8000/admin/studentapi/?ordering=name