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

**Session-21**

**Pagination in DRF:**

* Django provides a few classes that help you manage paginated data – that is, data that’s split across several pages, with “Previous/Next” links.
* REST framework includes support for customizable pagination styles. This allows you to modify how large result sets are split into individual pages of data.
* Pagination can be of 3 ways.

1. pageNumberPagination
2. limitOffsetPagination
3. CursorPagination

**Setting pagination style:**

* The pagination style may be set globally, using the DEFAULT\_PAGINATION\_CLASS and PAGE\_SIZE

**Settings.py:**

REST\_FRAMEWORK={  
'DEFAULT\_PAGINATION\_CLASS':'rest\_framework.pagination.PageNumberPagination',  
 'PAGE\_SIZE':3  
}

* You can also set the pagination class on an individual view by using the pagination\_class attribute.

**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',  
 'paginationapp',  
]

**models.py :**

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

**admin.py :**

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

**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 paginationapp.models import Student  
  
class StudentSerializer(serializers.ModelSerializer):  
 class Meta:  
 model=Student  
 fields=['id','name','address','age']

**views.py :**

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

**urls.py:**

from django.contrib import admin  
from django.urls import path  
from paginationapp 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/

**Pagination per view:**

* Create a new python file with the name pagination.py

**Pagination.py:**

from rest\_framework.pagination import PageNumberPagination  
  
class MyPagination(PageNumberPagination):  
 page\_size = 3  
 page\_query\_param = 'p'  
 page\_size\_query\_param = 'records'  
 max\_page\_size = 5

**views.py :**

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

* Now run server: python manage.py runserver
* Now go to browser: <http://127.0.0.1:8000/admin/studentapi/>
* <http://127.0.0.1:8000/studentapi/?page=2>
* http://127.0.0.1:8000/studentapi/?p=2