Name: Areej Alharbi

Phone: 0531158628

Email: areejalharbi57@gmail.com

# Student Task

## Introduction

I write this report to clarify my solution what I did, what I tried to do and how I dealing with the problem that I faced.

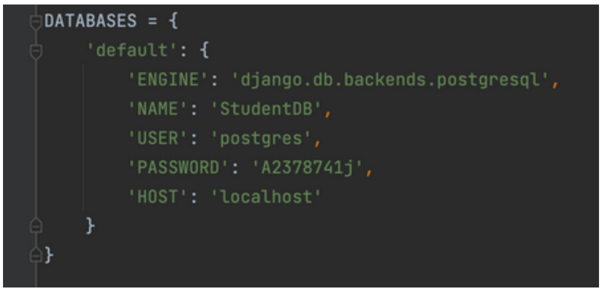
## Tools

1. Postgresql database
2. Postman

## Task Solution

- ### Connection with the database:

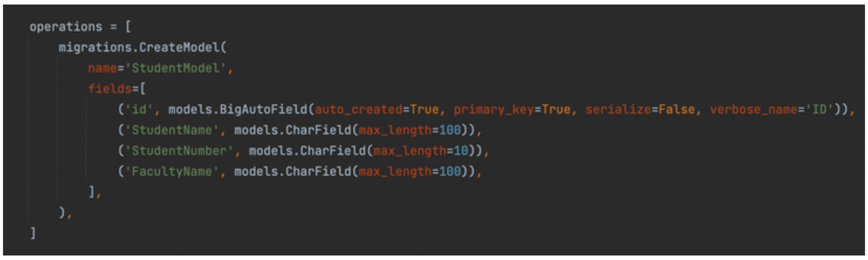
connection



```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'StudentDB',
        'USER': 'postgres',
        'PASSWORD': 'A2378741j',
        'HOST': 'localhost'
    }
}
```

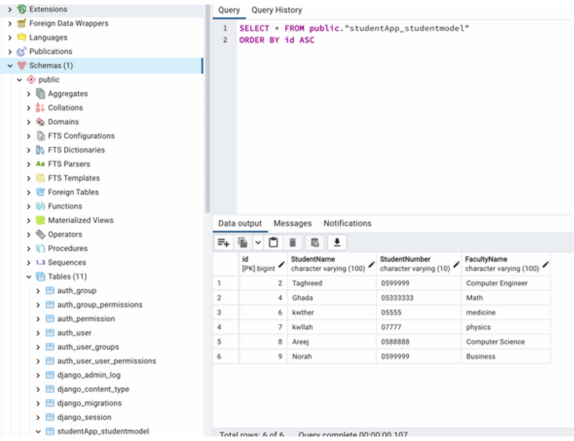
model table creation

I didn't write "id' in my field since Django already create a Default primary key



```
operations = [
    migrations.CreateModel(
        name='StudentModel',
        fields=[
            ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
            ('StudentName', models.CharField(max_length=100)),
            ('StudentNumber', models.CharField(max_length=10)),
            ('FacultyName', models.CharField(max_length=100)),
        ],
    ),
]
```

postgres table after connection



```sql
SELECT * FROM public."studentApp_studentmodel"
ORDER BY id ASC
```

| | id<br>[PK] bigint | StudentName<br>character varying (100) | StudentNumber<br>character varying (10) | FacultyName<br>character varying (100) |
|---|---|---|---|---|
| 1 | 2 | Taghreed | 0599999 | Computer Engineer |
| 2 | 4 | Ghada | 05333333 | Math |
| 3 | 6 | kwther | 05555 | medicine |
| 4 | 7 | kwllah | 07777 | physics |
| 5 | 8 | Areej | 0588888 | Computer Science |
| 6 | 9 | Norah | 0599999 | Business |

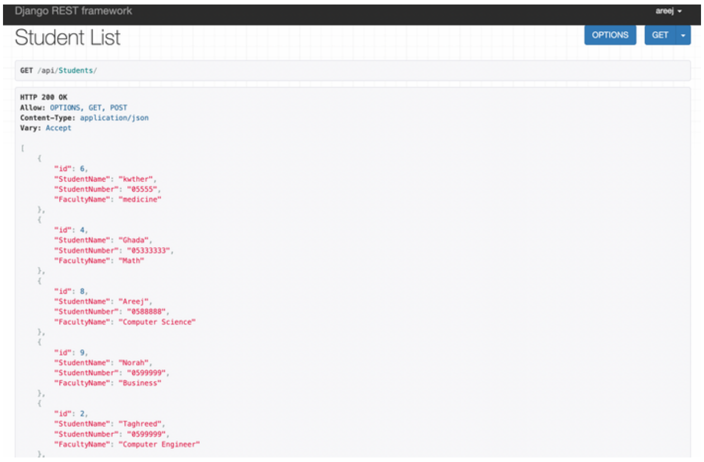
Total rows: 6 of 6    Query complete 00:00:00.107

- CRUD operations:

In  REST Framework I use  function based API view (api_view() Decorator ), to display and test the data I use Django administration, postman and rest framework server.

GET | api/Students

rest framework runserver:

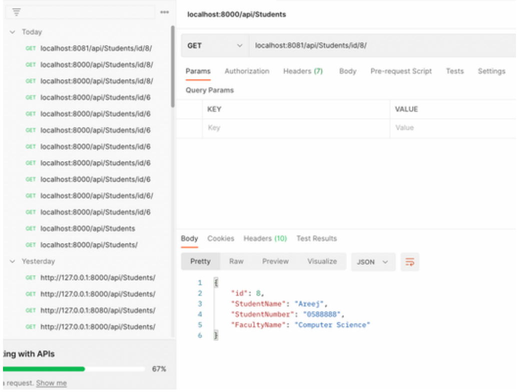


Student List

```
GET /api/Students/

HTTP 200 OK
Allow: OPTIONS, GET, POST
Content-Type: application/json
Vary: Accept

[
    {
        "id": 6,
        "StudentName": "kwther",
        "StudentNumber": "05555",
        "FacultyName": "medicine"
    },
    {
        "id": 4,
        "StudentName": "Ghada",
        "StudentNumber": "05333333",
        "FacultyName": "Math"
    },
    {
        "id": 8,
        "StudentName": "Areej",
        "StudentNumber": "0588888",
        "FacultyName": "Computer Science"
    },
    {
        "id": 9,
        "StudentName": "Norah",
        "StudentNumber": "0599999",
        "FacultyName": "Business"
    },
    {
        "id": 2,
        "StudentName": "Taghreed",
        "StudentNumber": "0599999",
        "FacultyName": "Computer Engineer"
    },
```

GET |  api/Students/id/

postman:


Today
GET localhost:8081/api/Students/id/8/
GET localhost:8000/api/Students/id/8/
GET localhost:8000/api/Students/id/8/
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6/
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students
GET localhost:8000/api/Students/
Yesterday
GET http://127.0.0.1:8000/api/Students/
GET http://127.0.0.1:8000/api/Students/
GET http://127.0.0.1:8080/api/Students/
GET http://127.0.0.1:8000/api/Students/
ing with APIs
67%
a request. Show me
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Query Params
KEY
VALUE
Key
Value
Body
Cookies
Headers (10)
Test Results
Pretty
Raw
Preview
Visualize
JSON

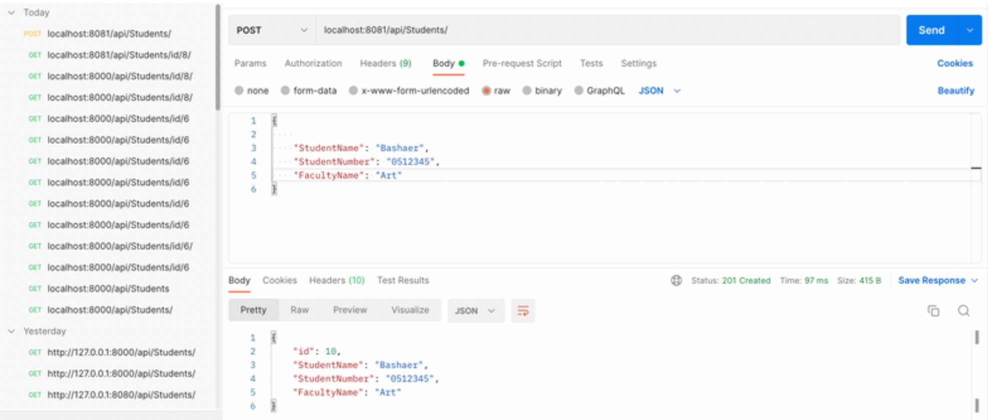
localhost:8000/api/Students

GET localhost:8081/api/Students/id/8/

```json
{
    "id": 8,
    "StudentName": "Areej",
    "StudentNumber": "0588888",
    "FacultyName": "Computer Science"
}
```

POST | api/Students/

postman:



Today
POST localhost:8081/api/Students/
GET localhost:8081/api/Students/id/8/
GET localhost:8000/api/Students/id/8/
GET localhost:8000/api/Students/id/8/
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6/
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students
GET localhost:8000/api/Students/
Yesterday
GET http://127.0.0.1:8000/api/Students/
GET http://127.0.0.1:8000/api/Students/
GET http://127.0.0.1:8080/api/Students/

POST localhost:8081/api/Students/

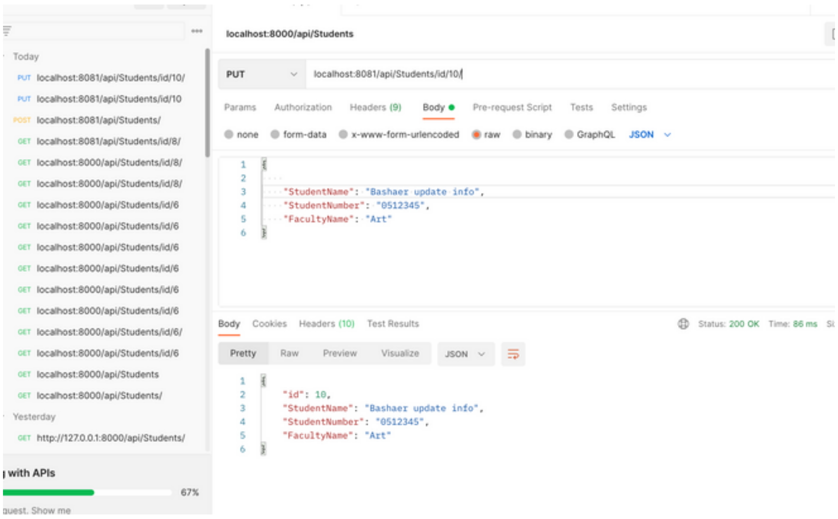
```
{

    "StudentName": "Bashaer",
    "StudentNumber": "0512345",
    "FacultyName": "Art"
}
```

Status: 201 Created Time: 97 ms Size: 415 B

```
{
    "id": 10,
    "StudentName": "Bashaer",
    "StudentNumber": "0512345",
    "FacultyName": "Art"
}
```

PUT | api/Students/id

postman:



Today
PUT localhost:8081/api/Students/id/10/
PUT localhost:8081/api/Students/id/10
POST localhost:8081/api/Students/
GET localhost:8081/api/Students/id/8/
GET localhost:8000/api/Students/id/8/
GET localhost:8000/api/Students/id/8/
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6/
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students
GET localhost:8000/api/Students/
Yesterday
GET http://127.0.0.1:8000/api/Students/

g with APIs
67%
quest. Show me

localhost:8000/api/Students

PUT localhost:8081/api/Students/id/10/

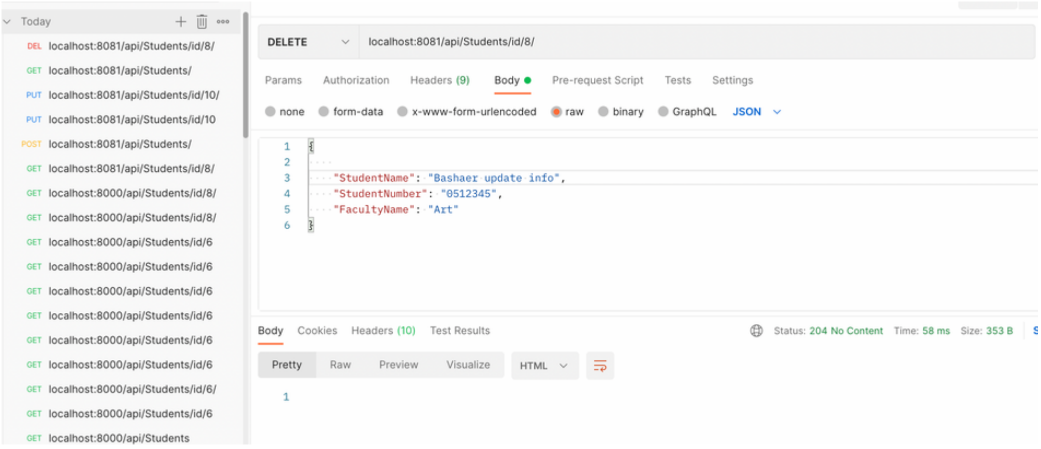
```
{

    "StudentName": "Bashaer update info",
    "StudentNumber": "0512345",
    "FacultyName": "Art"
}
```

Status: 200 OK Time: 86 ms Si

```
{
    "id": 10,
    "StudentName": "Bashaer update info",
    "StudentNumber": "0512345",
    "FacultyName": "Art"
}
```

DELETE | api/Students/id

postman:



Today
DEL localhost:8081/api/Students/id/8/
GET localhost:8081/api/Students/
PUT localhost:8081/api/Students/id/10/
PUT localhost:8081/api/Students/id/10
POST localhost:8081/api/Students/
GET localhost:8081/api/Students/id/8/
GET localhost:8000/api/Students/id/8/
GET localhost:8000/api/Students/id/8/
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students/id/6/
GET localhost:8000/api/Students/id/6
GET localhost:8000/api/Students

DELETE localhost:8081/api/Students/id/8/

```
{

    "StudentName": "Bashaer update info",
    "StudentNumber": "0512345",
    "FacultyName": "Art"
}
```

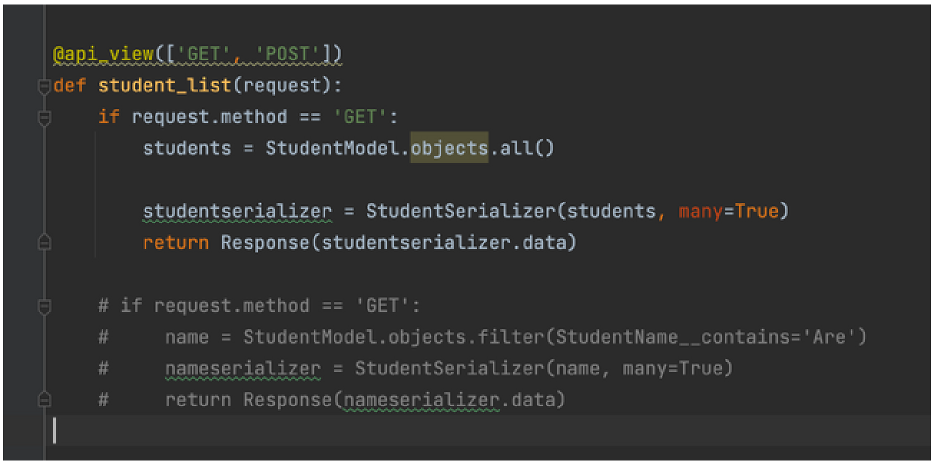
Status: 204 No Content Time: 58 ms Size: 353 B

GET | api/Students?name=kw

in this part I got two problems :

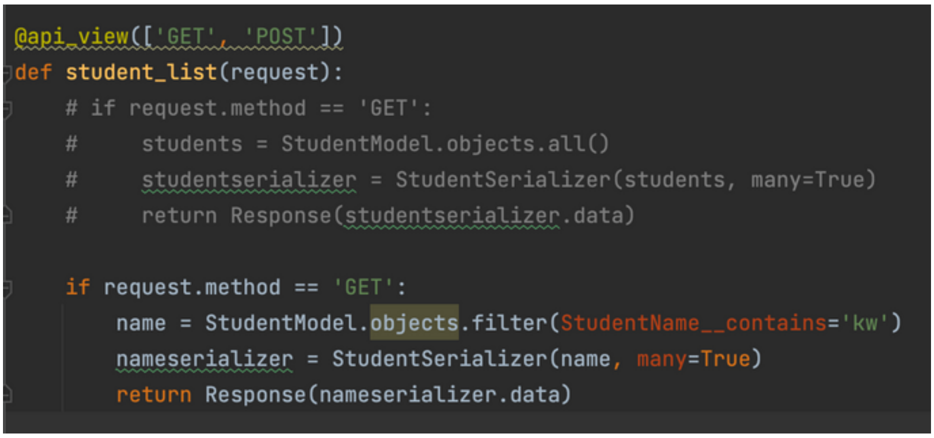
1. I tried to add .filter() function after .all() function in one "GET" request but all data with the filtered data are shown together.

2. I separate .filter()and .all() in different "GET" request but also two functions didn't work correctly until I comment one of them like bellow:



```
@api_view(['GET', 'POST'])
def student_list(request):
    if request.method == 'GET':
        students = StudentModel.objects.all()

        studentserializer = StudentSerializer(students, many=True)
        return Response(studentserializer.data)

    # if request.method == 'GET':
    #     name = StudentModel.objects.filter(StudentName__contains='Are')
    #     nameserializer = StudentSerializer(name, many=True)
    #     return Response(nameserializer.data)
```

this return all objets

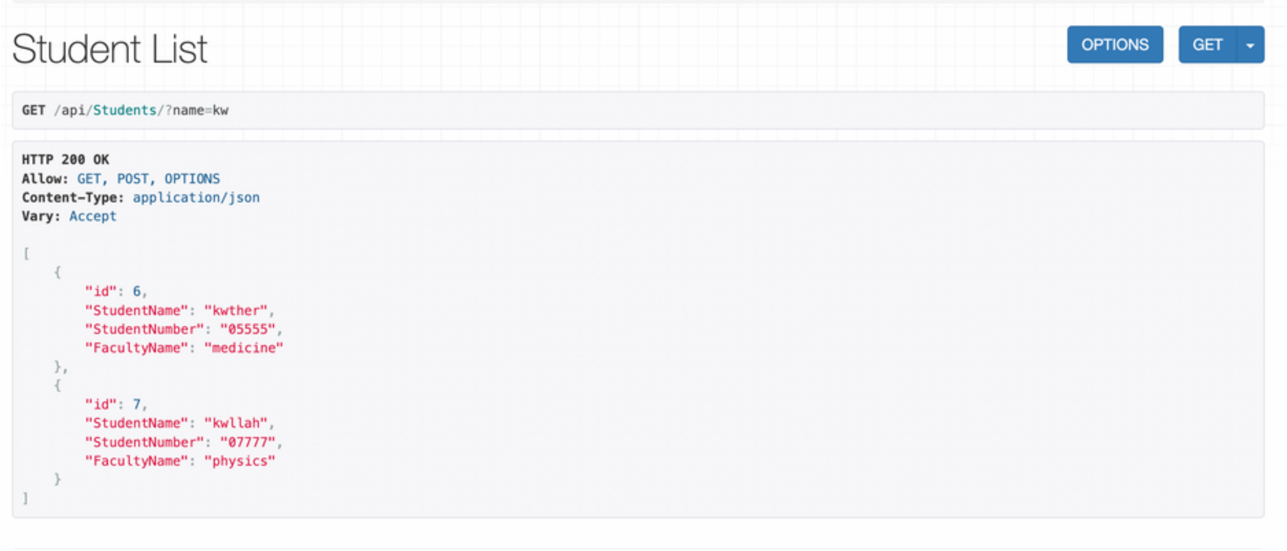


```
@api_view(['GET', 'POST'])
def student_list(request):
    # if request.method == 'GET':
    #     students = StudentModel.objects.all()
    #     studentserializer = StudentSerializer(students, many=True)
    #     return Response(studentserializer.data)

    if request.method == 'GET':
        name = StudentModel.objects.filter(StudentName__contains='kw')
        nameserializer = StudentSerializer(name, many=True)
        return Response(nameserializer.data)
```

this return filtered objets with name 'kw'

GET | api/Students?name=kw

Rest framework runserver:



```
GET /api/Students/?name=kw
```

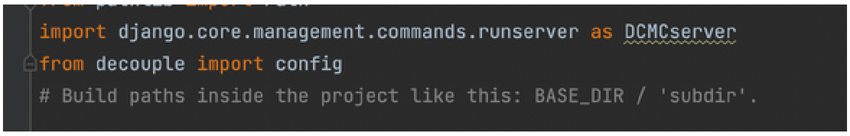
```
HTTP 200 OK
Allow: GET, POST, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 6,
        "StudentName": "kwther",
        "StudentNumber": "05555",
        "FacultyName": "medicine"
    },
    {
        "id": 7,
        "StudentName": "kwllah",
        "StudentNumber": "07777",
        "FacultyName": "physics"
    }
]
```

- ## CORS Middleware:

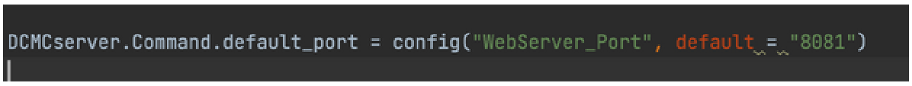
I followed enabling CORS middleware way that Witten in the documentation and GitHub but the original port 8000 didn't change.

```
CORS_ORIGIN_WHITELIST = (
  'http://google.com',
  'http://hostname.example.com',
  'http://127.0.0.1:8081',
)
```

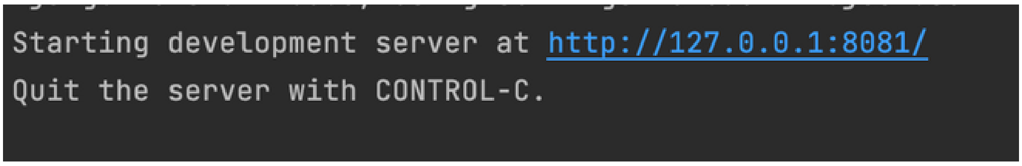
so I found another way to fix this problem and it works correctly with new port 8081.



```
import django.core.management.commands.runserver as DCMCserver
from decouple import config
# Build paths inside the project like this: BASE_DIR / 'subdir'.
```



```
DCMCserver.Command.default_port = config("WebServer_Port", default = "8081")
```



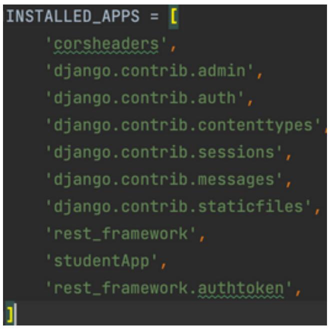
```
Starting development server at http://127.0.0.1:8081/
Quit the server with CONTROL-C.
```

changing the default port to 8081 successfully

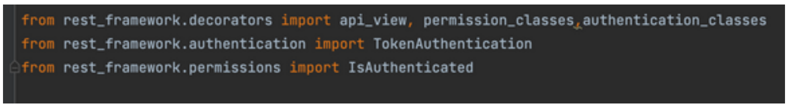
note: I leave the old way trying as a coment just to see how I try to work

- JWT Authentication

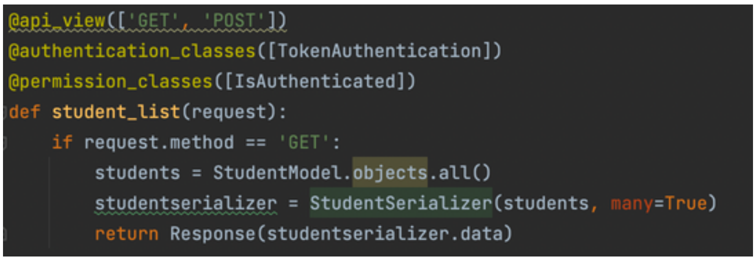
Due to time limit I try to solve the last part in fast, I use Django admin user to  apply Jwt authentication.I will show  the steps that I write it.



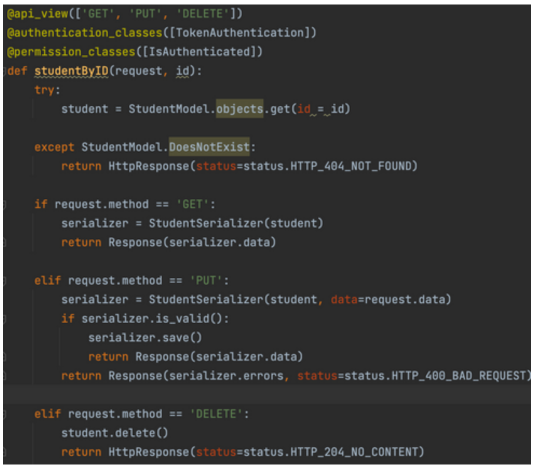
```
INSTALLED_APPS = [
    'corsheaders',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',
    'studentApp',
    'rest_framework.authtoken',
]
```



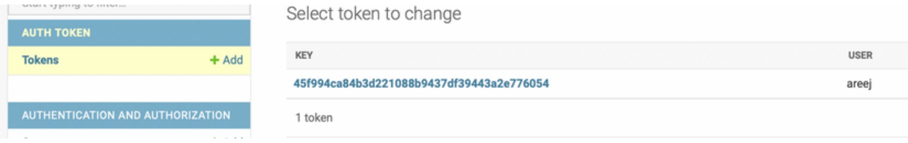
```
from rest_framework.decorators import api_view, permission_classes,authentication_classes
from rest_framework.authentication import TokenAuthentication
from rest_framework.permissions import IsAuthenticated
```



```
@api_view(['GET', 'POST'])
@authentication_classes([TokenAuthentication])
@permission_classes([IsAuthenticated])
def student_list(request):
    if request.method == 'GET':
        students = StudentModel.objects.all()
        studentserializer = StudentSerializer(students, many=True)
        return Response(studentserializer.data)
```



```
@api_view(['GET', 'PUT', 'DELETE'])
@authentication_classes([TokenAuthentication])
@permission_classes([IsAuthenticated])
def studentByID(request, id):
    try:
        student = StudentModel.objects.get(id = id)

    except StudentModel.DoesNotExist:
        return HttpResponse(status=status.HTTP_404_NOT_FOUND)

    if request.method == 'GET':
        serializer = StudentSerializer(student)
        return Response(serializer.data)

    elif request.method == 'PUT':
        serializer = StudentSerializer(student, 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.method == 'DELETE':
        student.delete()
        return HttpResponse(status=status.HTTP_204_NO_CONTENT)
```

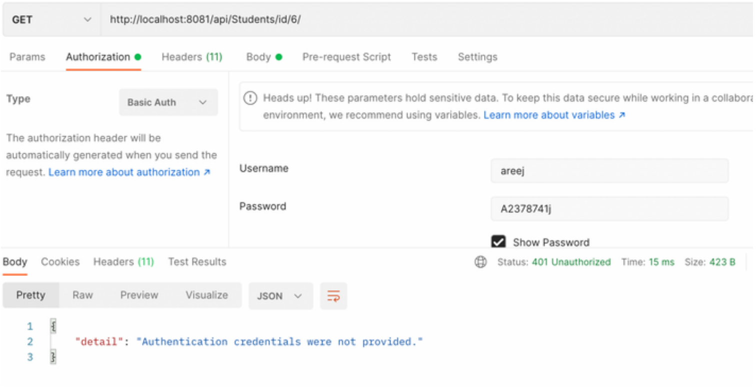


Select token to change

| KEY | USER |
|---|---|
| 45f994ca84b3d221088b9437df39443a2e776054 | areej |

1 token

create token manually



http://localhost:8081/api/Students/id/6/

Type: Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about authorization ↗

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collabora environment, we recommend using variables. Learn more about variables ↗

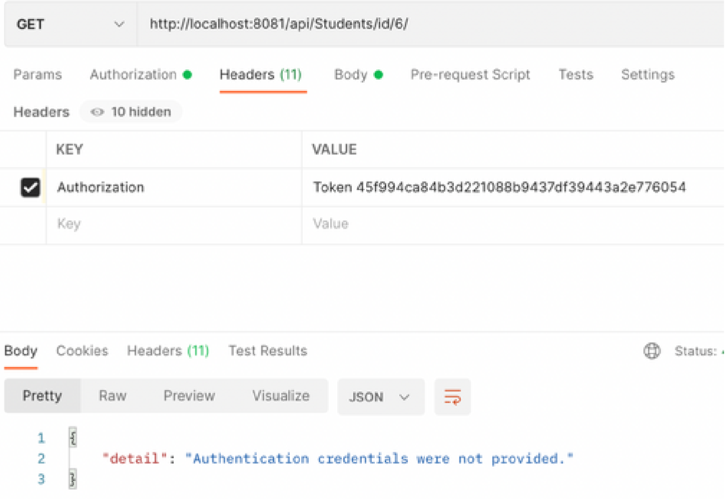
Username: areej

Password: A2378741j

☑ Show Password

Status: 401 Unauthorized Time: 15 ms Size: 423 B

```
{
    "detail": "Authentication credentials were not provided."
}
```



http://localhost:8081/api/Students/id/6/

| | KEY | VALUE |
|---|---|---|
| ☑ | Authorization | Token 45f994ca84b3d221088b9437df39443a2e776054 |
| | Key | Value |

```
{
    "detail": "Authentication credentials were not provided."
}
```

the access denied correctly but after added the token it didn't give an access to the data, I will search to find the problem and I will add email and password to the table insted of Django user.

hint: I leave the steps as a comment so my project can works fine.

thank you