Topic: 5. Serializing / Deserializing JSON Data (GET/POST)

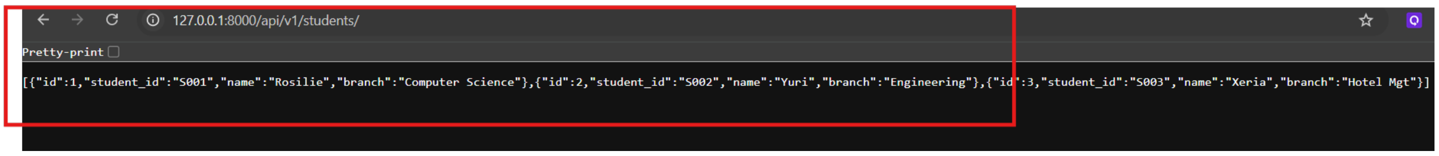
Speaker: Personal | Notebook: API Development using Django Framework

---

To see more details about serializers, view this Youtube clip

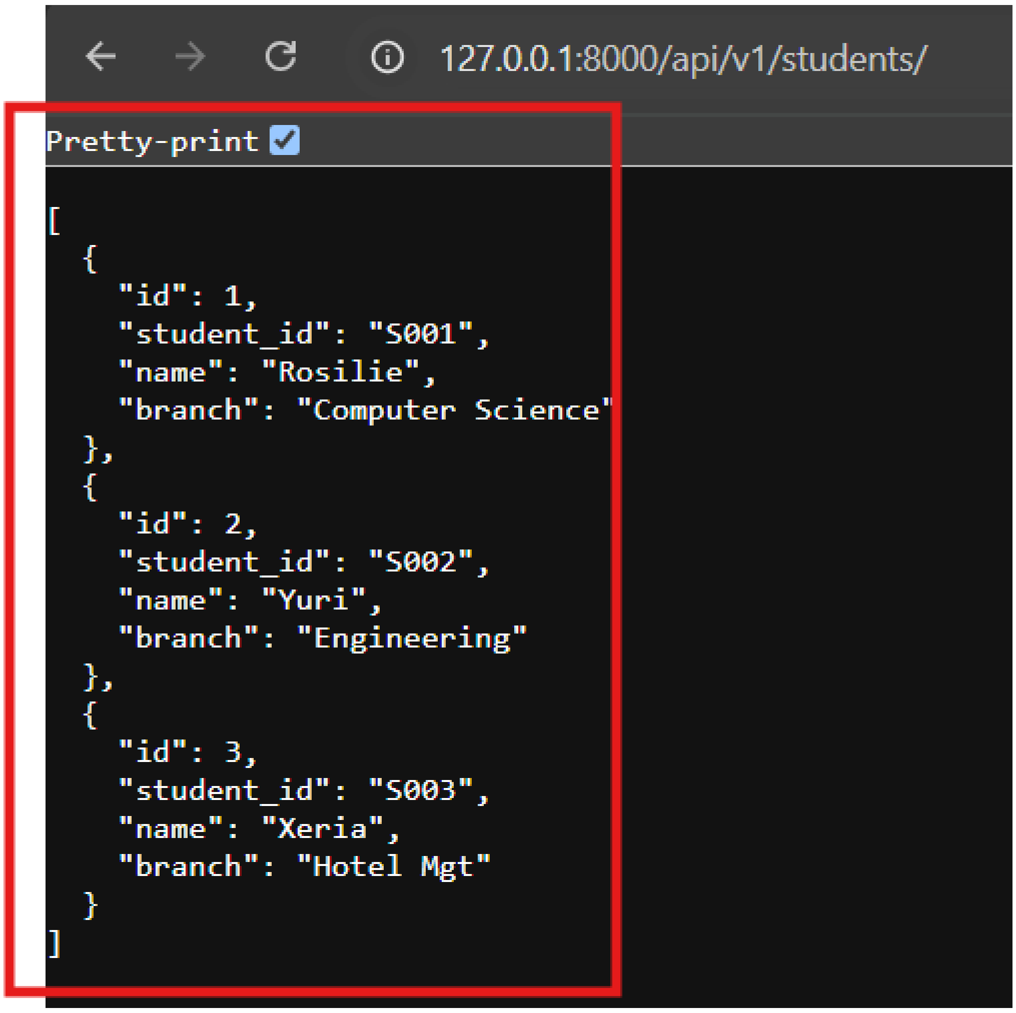
1. To view the JSON file in a formatted style, **we added the Google Chrome extension, JSON FORMATTER:**

BEFORE:



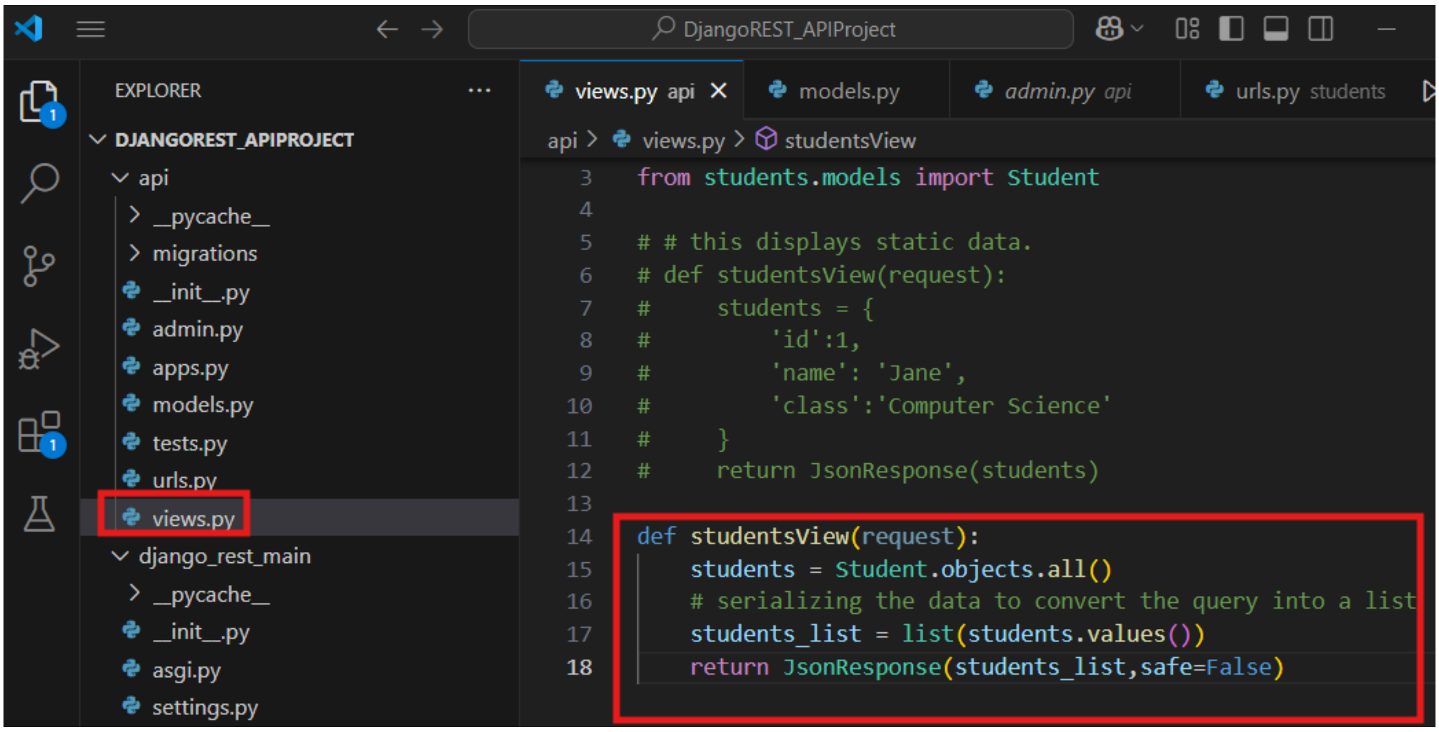
```
127.0.0.1:8000/api/v1/students/
Pretty-print ☐
[{"id":1,"student_id":"S001","name":"Rosilie","branch":"Computer Science"},{"id":2,"student_id":"S002","name":"Yuri","branch":"Engineering"},{"id":3,"student_id":"S003","name":"Xeria","branch":"Hotel Mgt"}]
```

AFTER:



```
127.0.0.1:8000/api/v1/students/
Pretty-print ☑
[
  {
    "id": 1,
    "student_id": "S001",
    "name": "Rosilie",
    "branch": "Computer Science"
  },
  {
    "id": 2,
    "student_id": "S002",
    "name": "Yuri",
    "branch": "Engineering"
  },
  {
    "id": 3,
    "student_id": "S003",
    "name": "Xeria",
    "branch": "Hotel Mgt"
  }
]
```

2. Previously, we manually used serializers to convert our query set into a list. The code is below to show the output in Step 1.

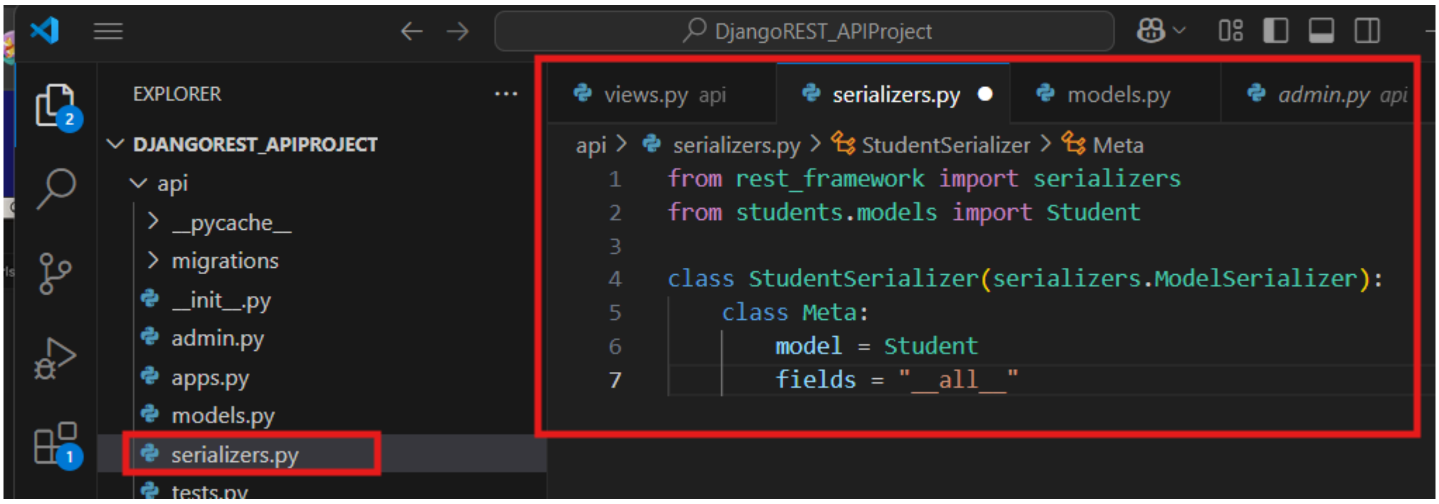


```python
from students.models import Student

# # this displays static data.
# def studentsView(request):
#     students = {
#         'id':1,
#         'name': 'Jane',
#         'class':'Computer Science'
#     }
#     return JsonResponse(students)

def studentsView(request):
    students = Student.objects.all()
    # serializing the data to convert the query into a list
    students_list = list(students.values())
    return JsonResponse(students_list,safe=False)
```

3. In Django, we can use serializer tools. In the API app folder, create a new file SERIALIZERS.PY:

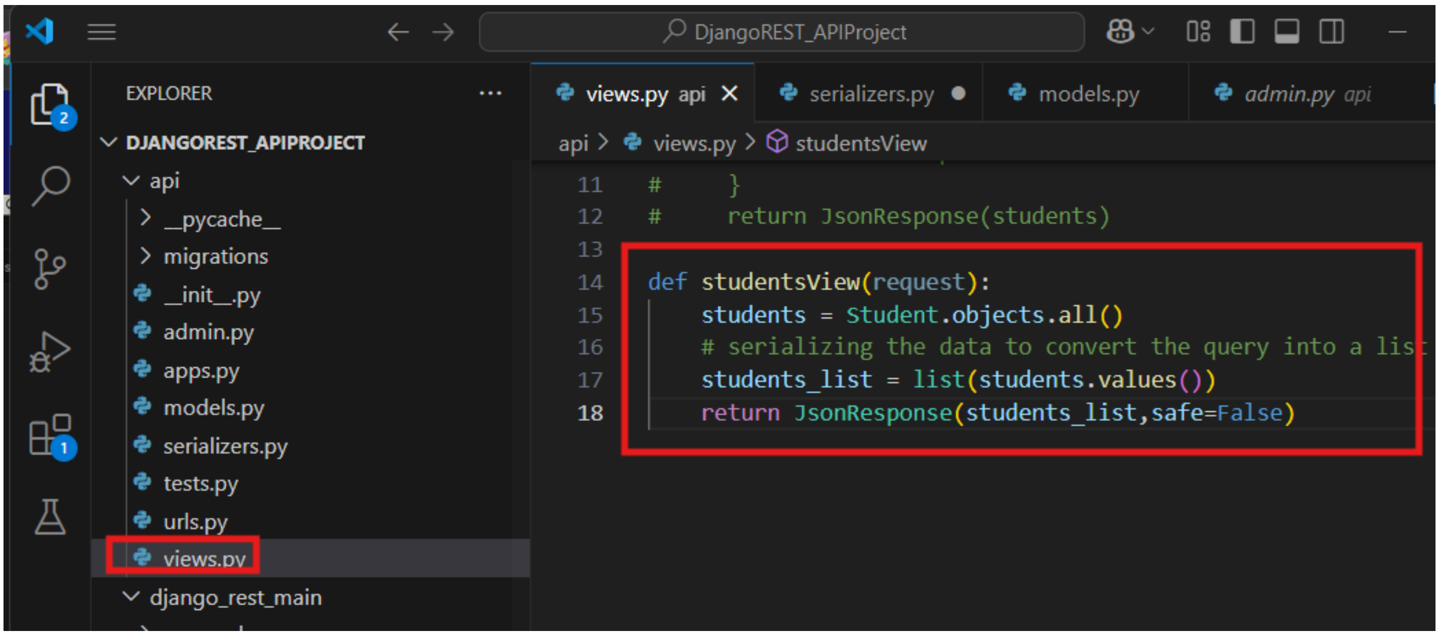


```python
from rest_framework import serializers
from students.models import Student

class StudentSerializer(serializers.ModelSerializer):
    class Meta:
        model = Student
        fields = "__all__"
```

4. Update our API\VIEWS.PY:

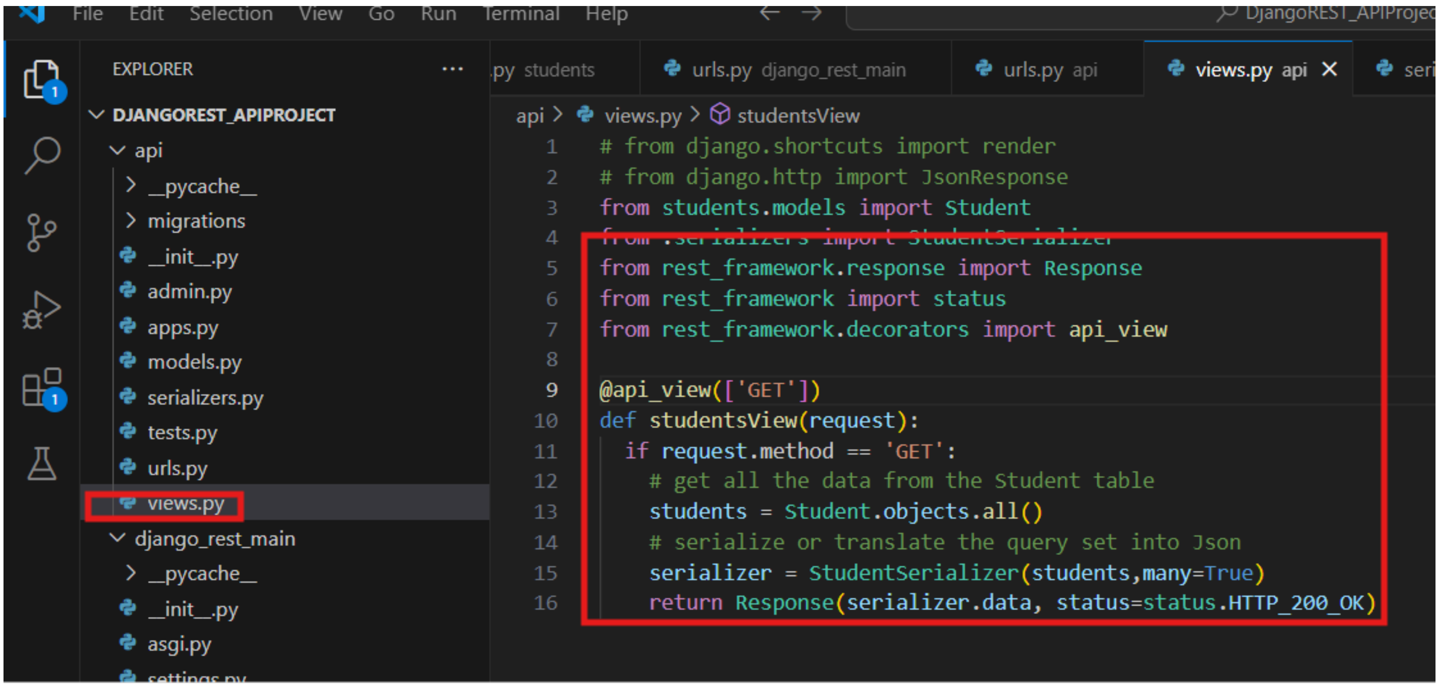
FROM manual serialization:



```python
#     }
#     return JsonResponse(students)

def studentsView(request):
    students = Student.objects.all()
    # serializing the data to convert the query into a lis
    students_list = list(students.values())
    return JsonResponse(students_list,safe=False)
```

TO:

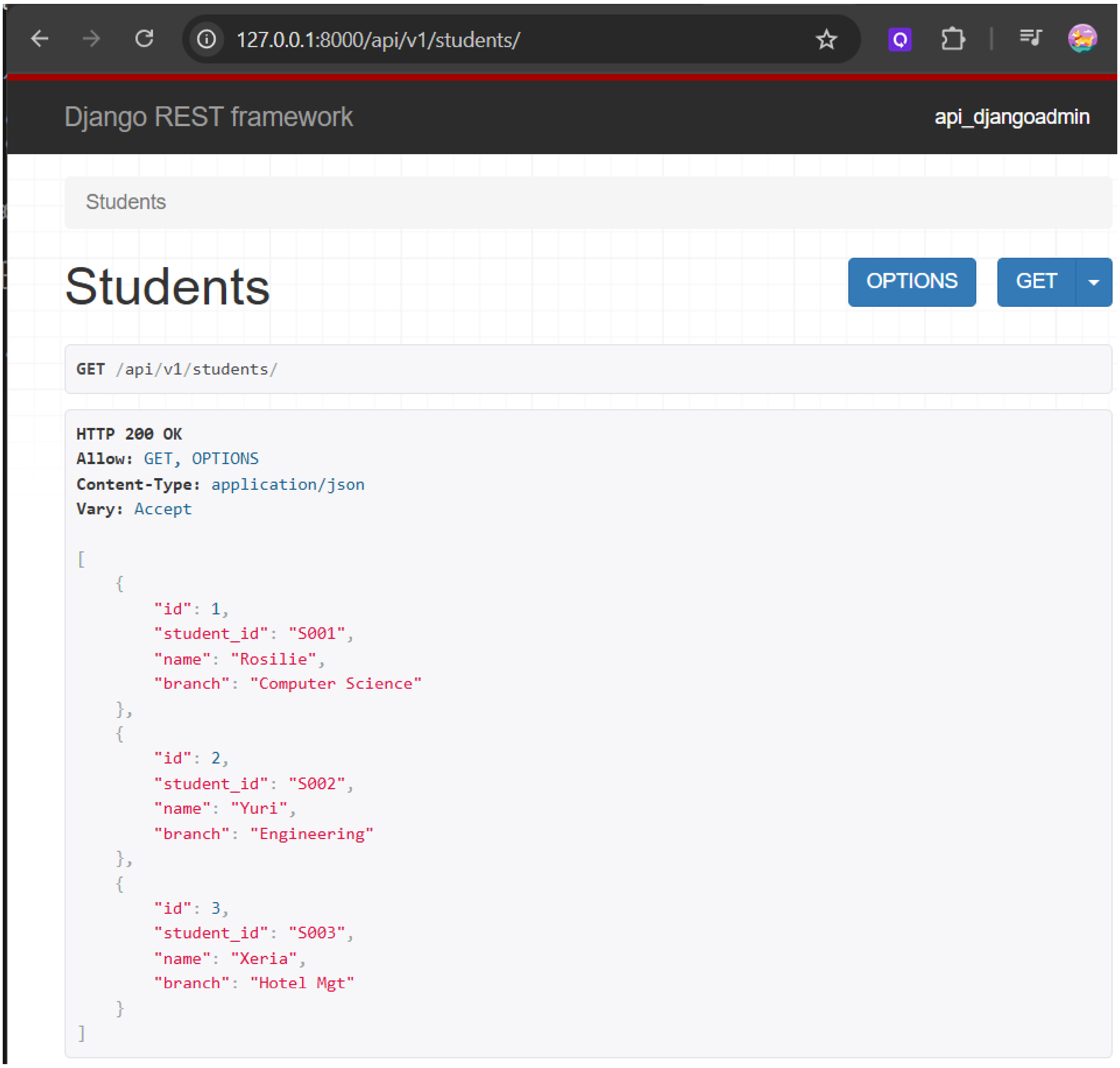


```python
# from django.shortcuts import render
# from django.http import JsonResponse
from students.models import Student
from .serializers import StudentSerializer
from rest_framework.response import Response
from rest_framework import status
from rest_framework.decorators import api_view

@api_view(['GET'])
def studentsView(request):
  if request.method == 'GET':
    # get all the data from the Student table
    students = Student.objects.all()
    # serialize or translate the query set into Json
    serializer = StudentSerializer(students,many=True)
    return Response(serializer.data, status=status.HTTP_200_OK)
```

So, when you run the URL path again:

```
http://127.0.0.1:8000/api/v1/students/
```



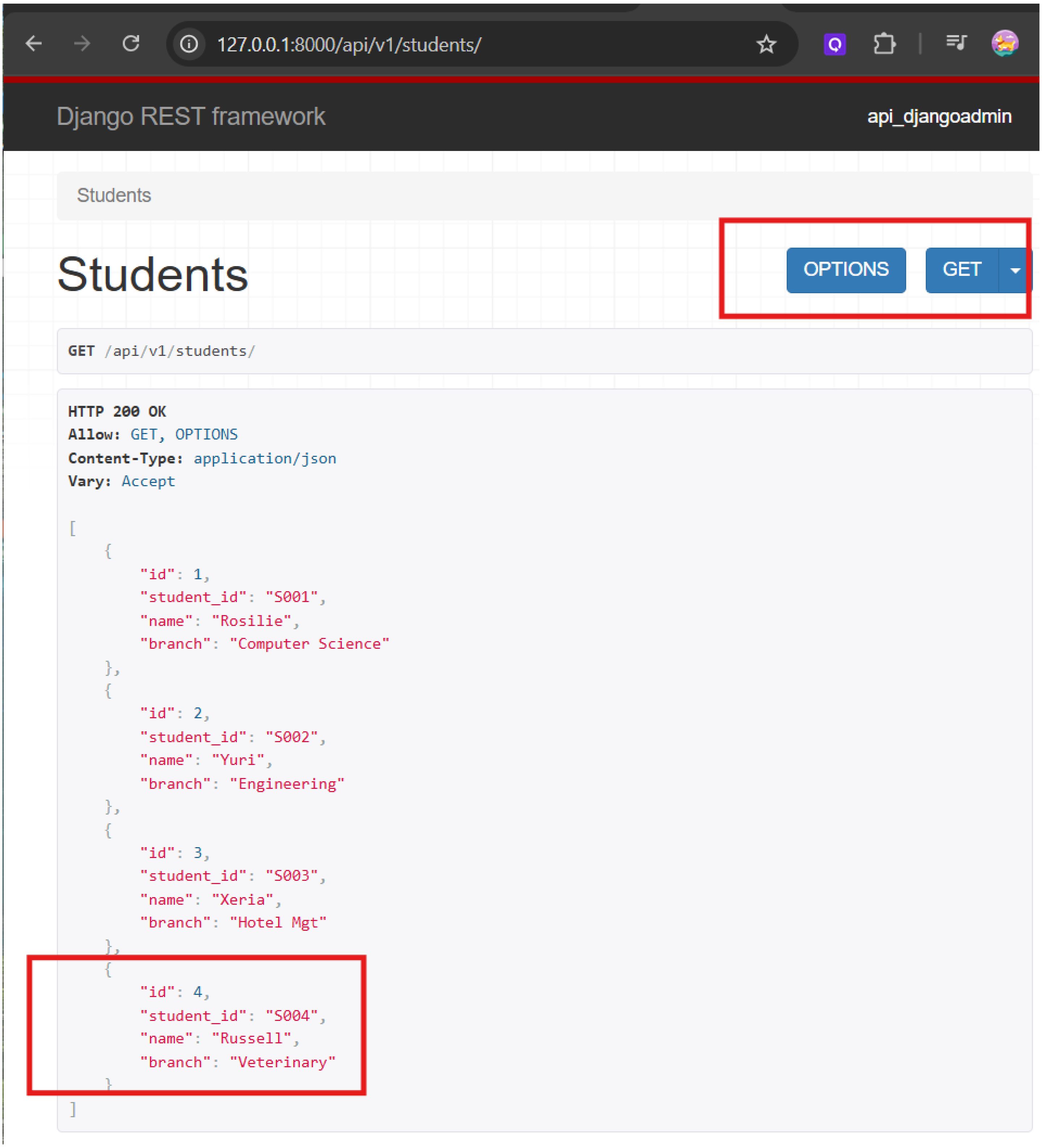
Students

```
GET /api/v1/students/
```

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

[
    {
        "id": 1,
        "student_id": "S001",
        "name": "Rosilie",
        "branch": "Computer Science"
    },
    {
        "id": 2,
        "student_id": "S002",
        "name": "Yuri",
        "branch": "Engineering"
    },
    {
        "id": 3,
        "student_id": "S003",
        "name": "Xeria",
        "branch": "Hotel Mgt"
    }
]
```

5. Now, if you update your database model for a new record and use the GET button from Step 4, you will be able to use GET button to get the latest added records or you can simply refresh your page and that will be considered as a GET method.



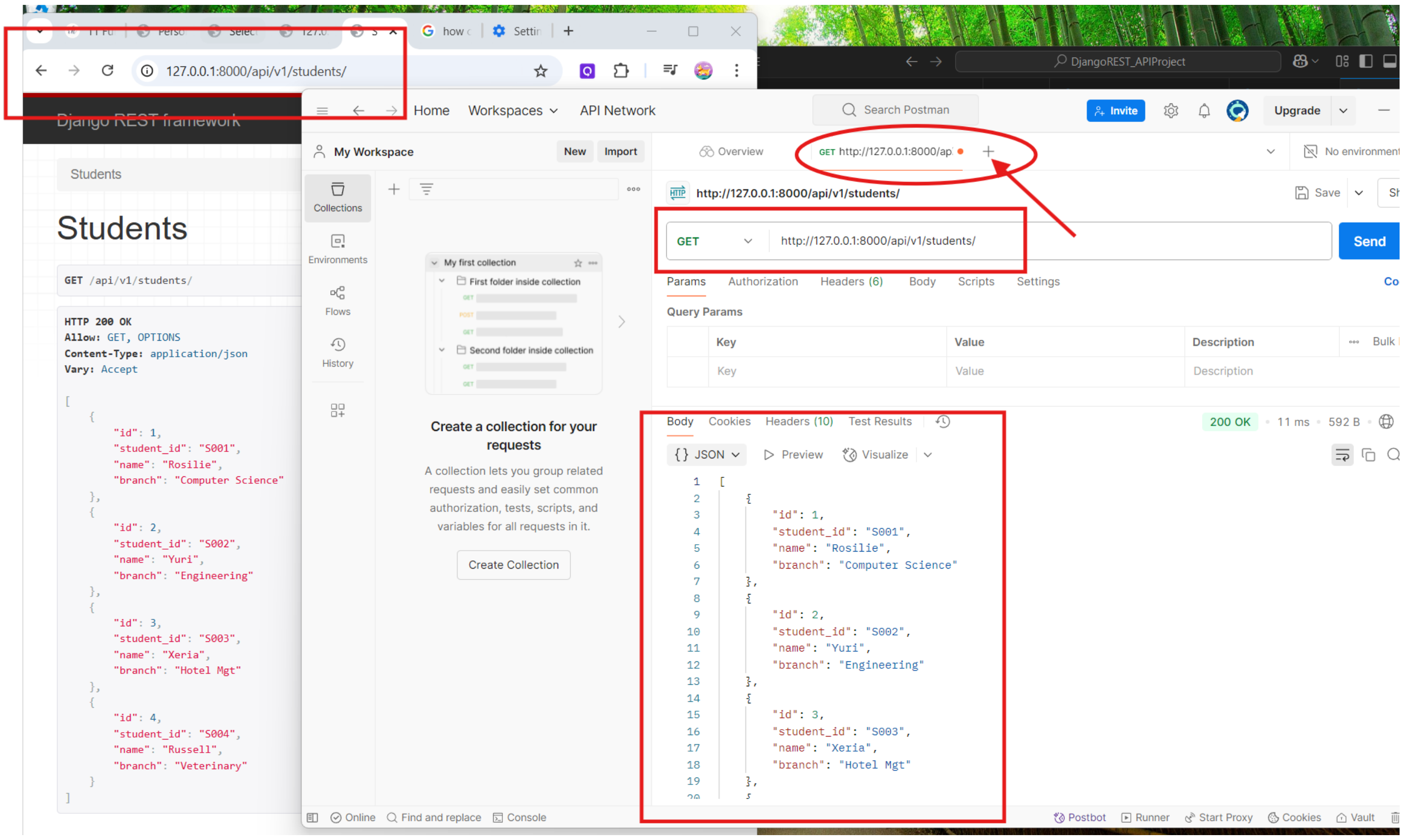
Students

```
GET /api/v1/students/
```

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

[
    {
        "id": 1,
        "student_id": "S001",
        "name": "Rosilie",
        "branch": "Computer Science"
    },
    {
        "id": 2,
        "student_id": "S002",
        "name": "Yuri",
        "branch": "Engineering"
    },
    {
        "id": 3,
        "student_id": "S003",
        "name": "Xeria",
        "branch": "Hotel Mgt"
    },
    {
        "id": 4,
        "student_id": "S004",
        "name": "Russell",
        "branch": "Veterinary"
    }
]
```

6. Now using **POSTMAN**, you can copy the same API link paste it into the POSTMAN search bar and use GET method. It should return all records from the database. Simply click on + and add the same path we used from the browser. To use POSTMAN, this must be installed in your device.

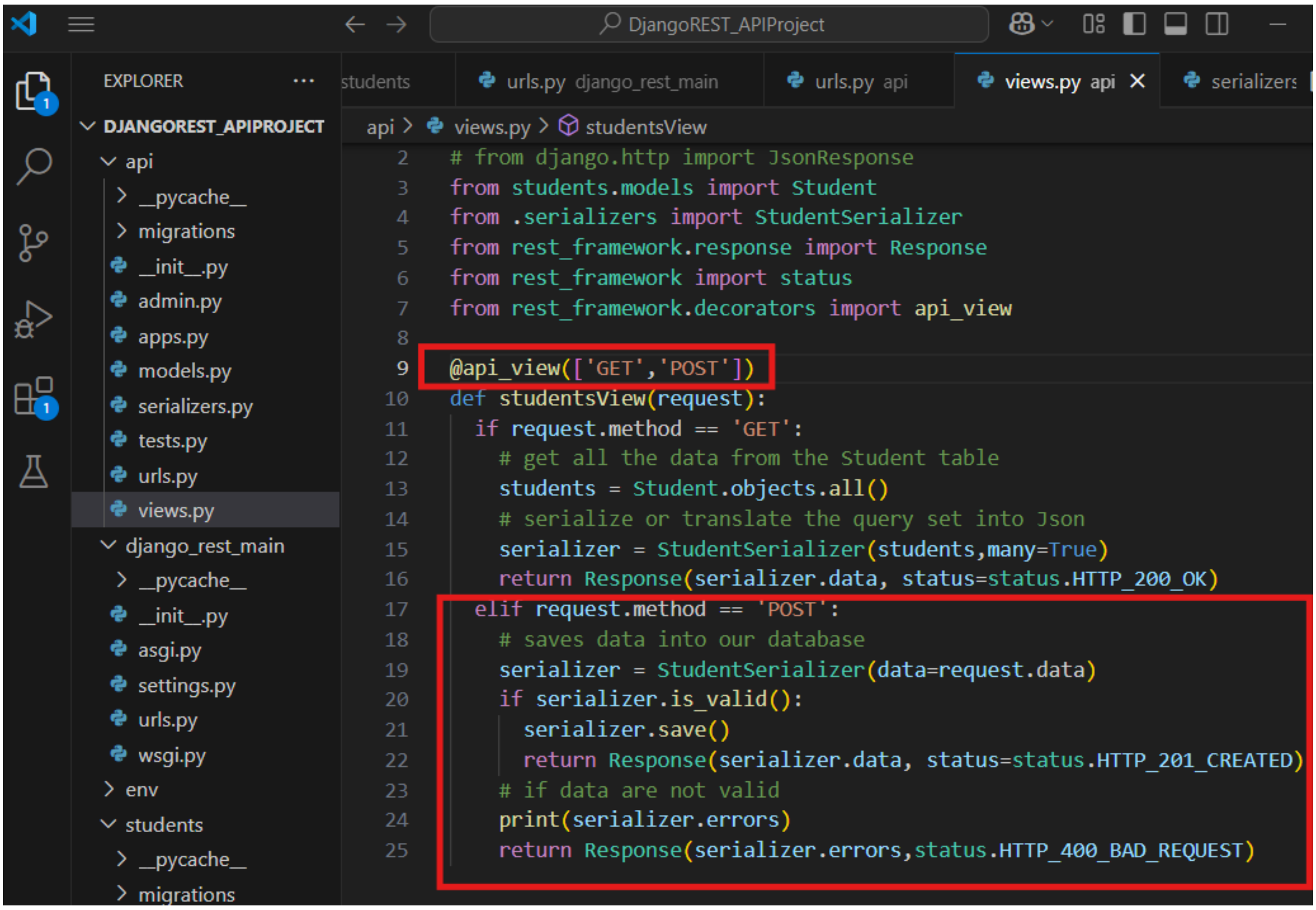


```
GET /api/v1/students/

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

[
    {
        "id": 1,
        "student_id": "S001",
        "name": "Rosilie",
        "branch": "Computer Science"
    },
    {
        "id": 2,
        "student_id": "S002",
        "name": "Yuri",
        "branch": "Engineering"
    },
    {
        "id": 3,
        "student_id": "S003",
        "name": "Xeria",
        "branch": "Hotel Mgt"
    },
    {
        "id": 4,
        "student_id": "S004",
        "name": "Russell",
        "branch": "Veterinary"
    }
]
```

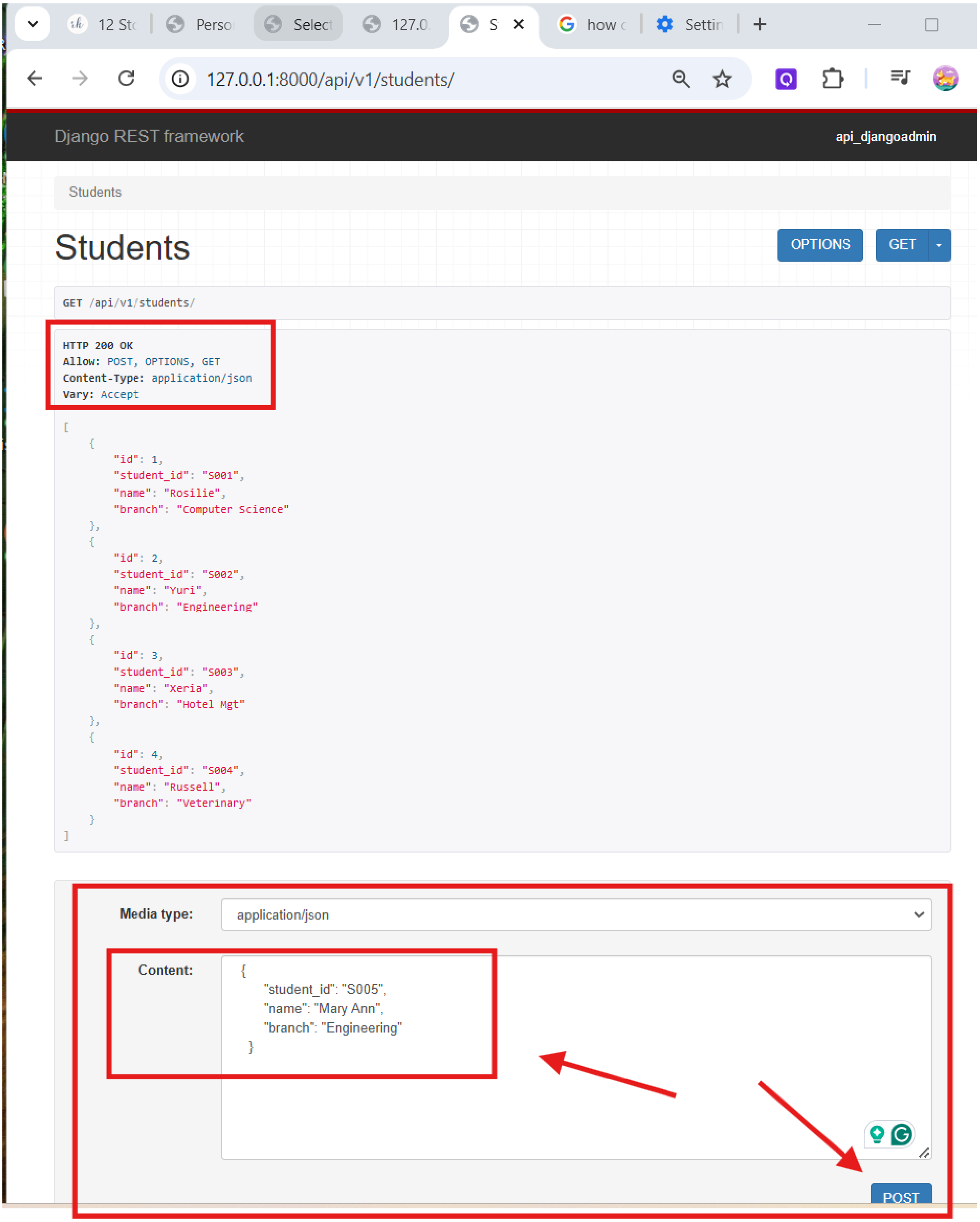
7. To store data using the Django Rest Framework, update the VIEWS.PY to allow for POST method.



```python
# from django.http import JsonResponse
from students.models import Student
from .serializers import StudentSerializer
from rest_framework.response import Response
from rest_framework import status
from rest_framework.decorators import api_view

@api_view(['GET','POST'])
def studentsView(request):
    if request.method == 'GET':
        # get all the data from the Student table
        students = Student.objects.all()
        # serialize or translate the query set into Json
        serializer = StudentSerializer(students,many=True)
        return Response(serializer.data, status=status.HTTP_200_OK)
    elif request.method == 'POST':
        # saves data into our database
        serializer = StudentSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data, status=status.HTTP_201_CREATED)
        # if data are not valid
        print(serializer.errors)
        return Response(serializer.errors,status.HTTP_400_BAD_REQUEST)
```

When you reload your page, then you can insert a new post:

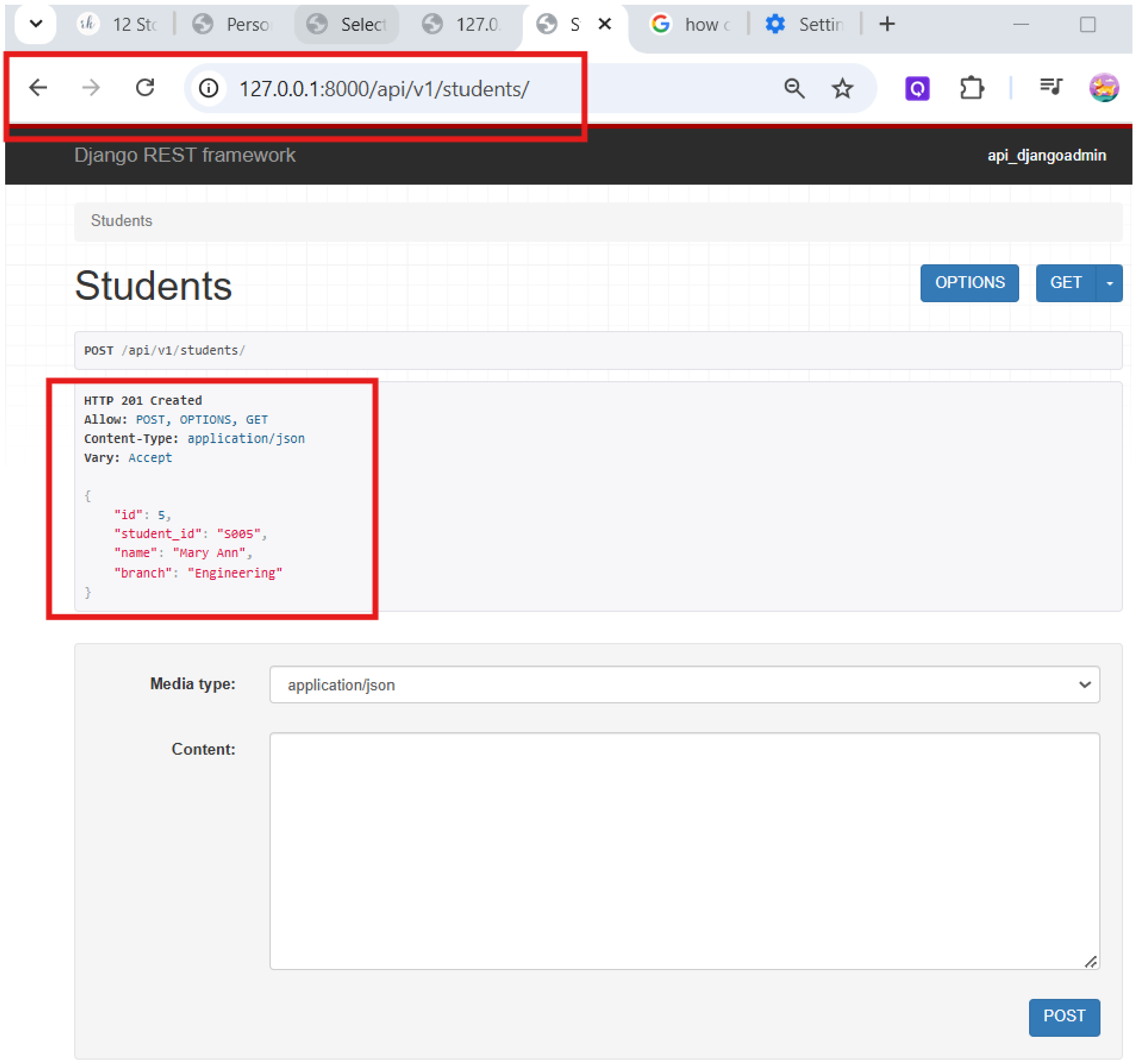


```
GET /api/v1/students/
```

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

[
    {
        "id": 1,
        "student_id": "S001",
        "name": "Rosilie",
        "branch": "Computer Science"
    },
    {
        "id": 2,
        "student_id": "S002",
        "name": "Yuri",
        "branch": "Engineering"
    },
    {
        "id": 3,
        "student_id": "S003",
        "name": "Xeria",
        "branch": "Hotel Mgt"
    },
    {
        "id": 4,
        "student_id": "S004",
        "name": "Russell",
        "branch": "Veterinary"
    }
]
```

```
{
    "student_id": "S005",
    "name": "Mary Ann",
    "branch": "Engineering"
  }
```



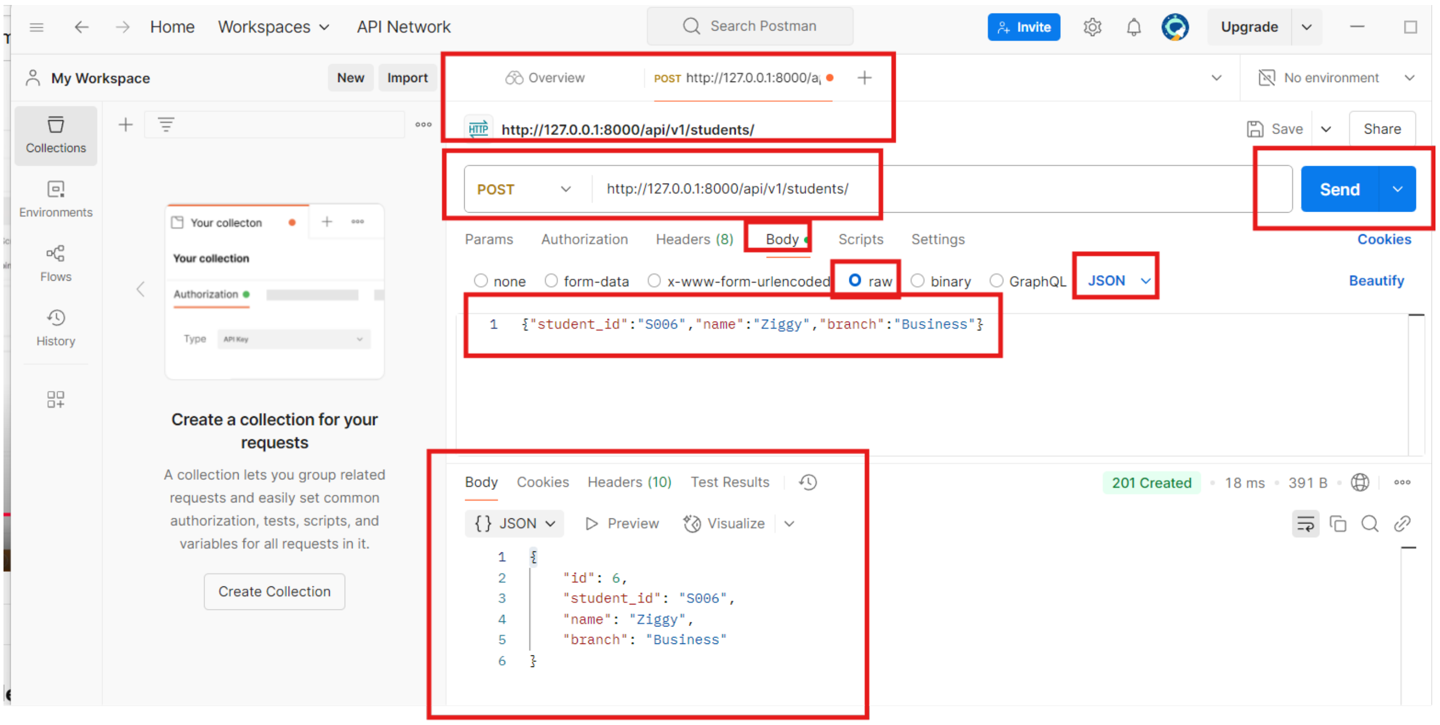
Students

```
POST /api/v1/students/
```

```
HTTP 201 Created
Allow: POST, OPTIONS, GET
Content-Type: application/json
Vary: Accept

{
    "id": 5,
    "student_id": "S005",
    "name": "Mary Ann",
    "branch": "Engineering"
}
```

8. To use POSTMAN, add the path again. Select BODY, then RAW, then JSON. Add your records then select select the SEND method.



POST http://127.0.0.1:8000/api/v1/students/

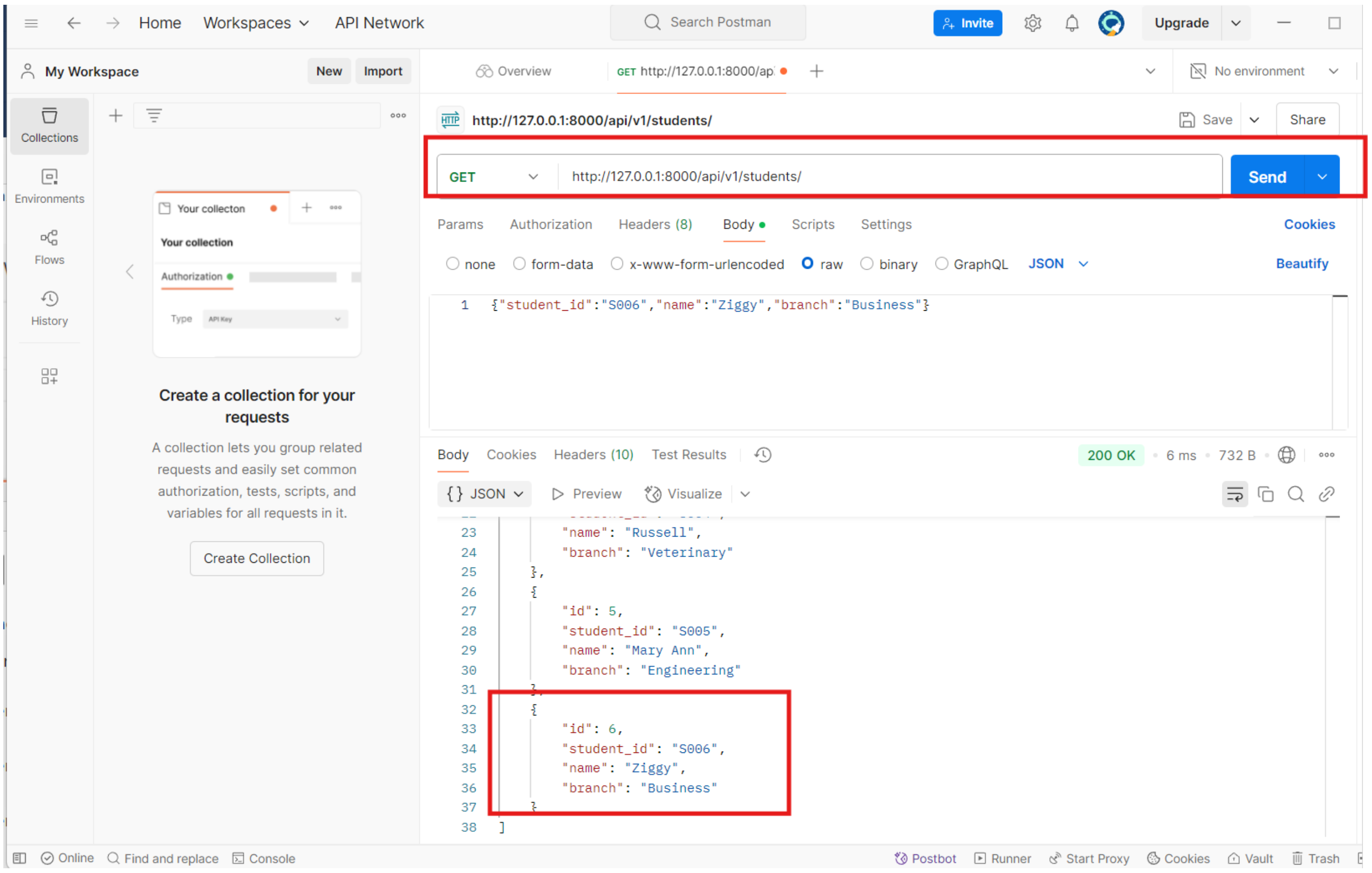
Body: [✓ raw] JSON

```
{"student_id":"S006","name":"Ziggy","branch":"Business"}
```

Body · Cookies · Headers (10) · Test Results — 201 Created · 18 ms · 391 B

```
{
    "id": 6,
    "student_id": "S006",
    "name": "Ziggy",
    "branch": "Business"
}
```

9. Now, to see the newly inserted record, use the GET method. You will then see the newly added record.



GET http://127.0.0.1:8000/api/v1/students/

Body: [✓ raw] JSON

```
{"student_id":"S006","name":"Ziggy","branch":"Business"}
```

Body · Cookies · Headers (10) · Test Results — 200 OK · 6 ms · 732 B

```
        "name": "Russell",
        "branch": "Veterinary"
    },
    {
        "id": 5,
        "student_id": "S005",
        "name": "Mary Ann",
        "branch": "Engineering"
    },
    {
        "id": 6,
        "student_id": "S006",
        "name": "Ziggy",
        "branch": "Business"
    }
]
```