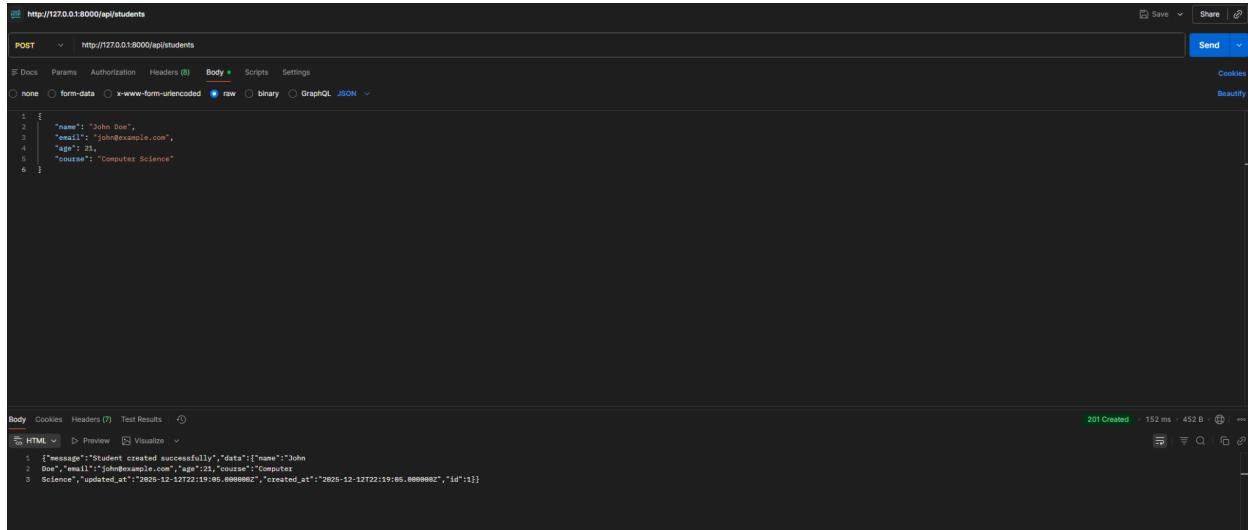


# Lab Task 10

POST

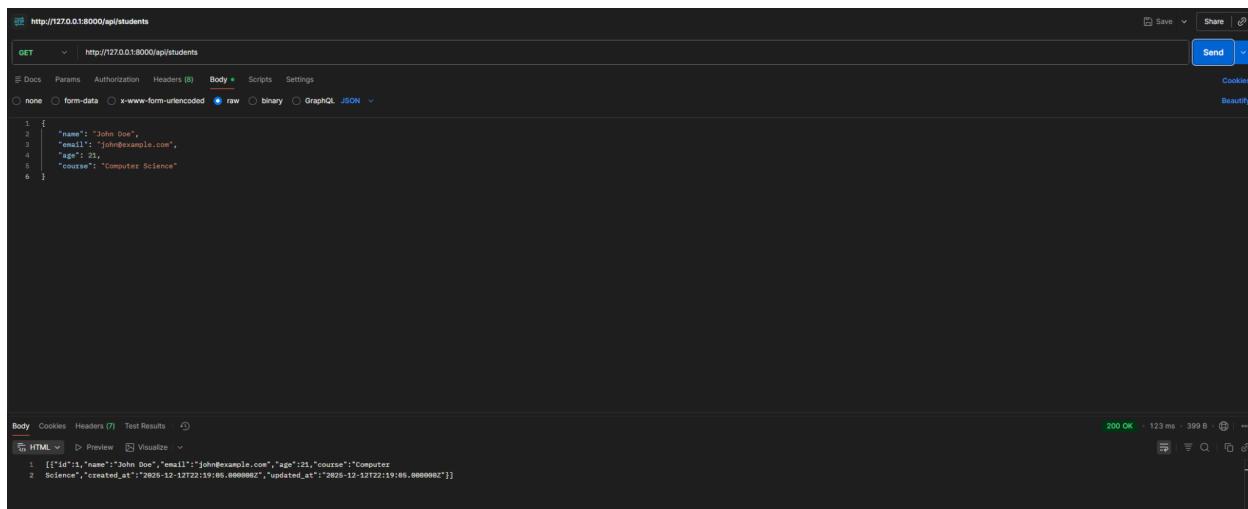


The screenshot shows a POST request to `http://127.0.0.1:8000/api/students`. The request body is a JSON object:

```
[{"name": "John Doe", "email": "john@example.com", "age": 21, "course": "Computer Science"}]
```

The response status is `201 Created` with a response time of 152 ms and a size of 452 B. The response body contains the message "Student created successfully" and the student's details.

GET

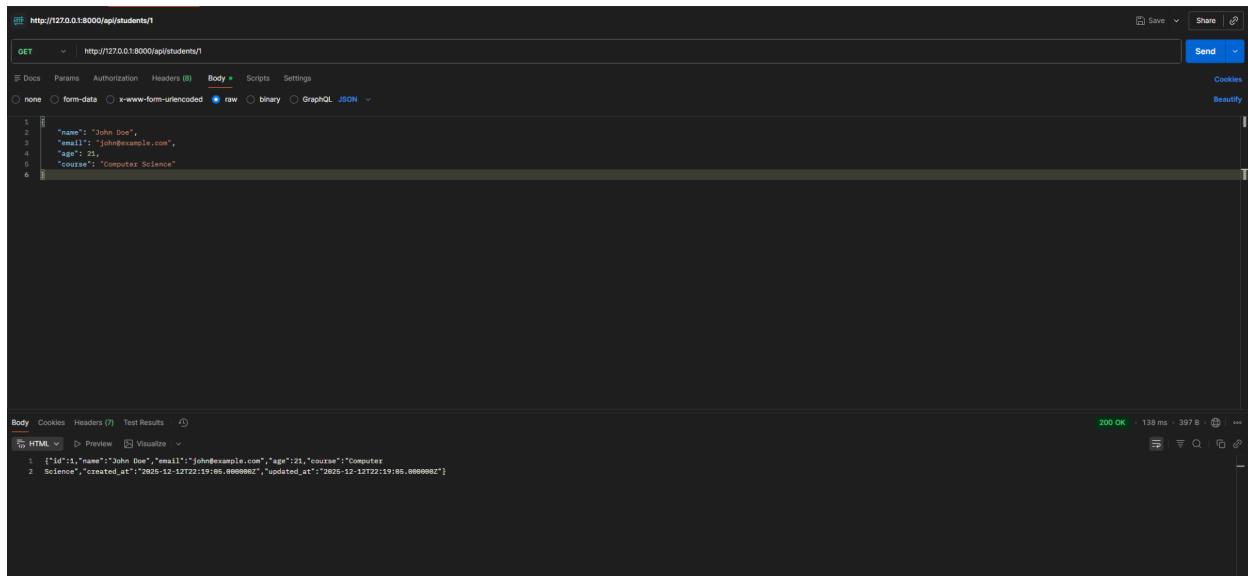


The screenshot shows a GET request to `http://127.0.0.1:8000/api/students`. The request body is a JSON object:

```
[{"name": "John Doe", "email": "john@example.com", "age": 21, "course": "Computer Science"}]
```

The response status is `200 OK` with a response time of 123 ms and a size of 399 B. The response body contains a list of student objects.

## GET at id 1



http://127.0.0.1:8000/api/students/1

GET | http://127.0.0.1:8000/api/students/1

Body | Headers (8) | Body | Scripts | Settings

none | form-data | x-www-form-urlencoded | raw | binary | GraphQL | JSON

```
1 { "name": "John Doe",  
2   "email": "john@example.com",  
3   "age": 21,  
4   "course": "Computer Science"  
5 }
```

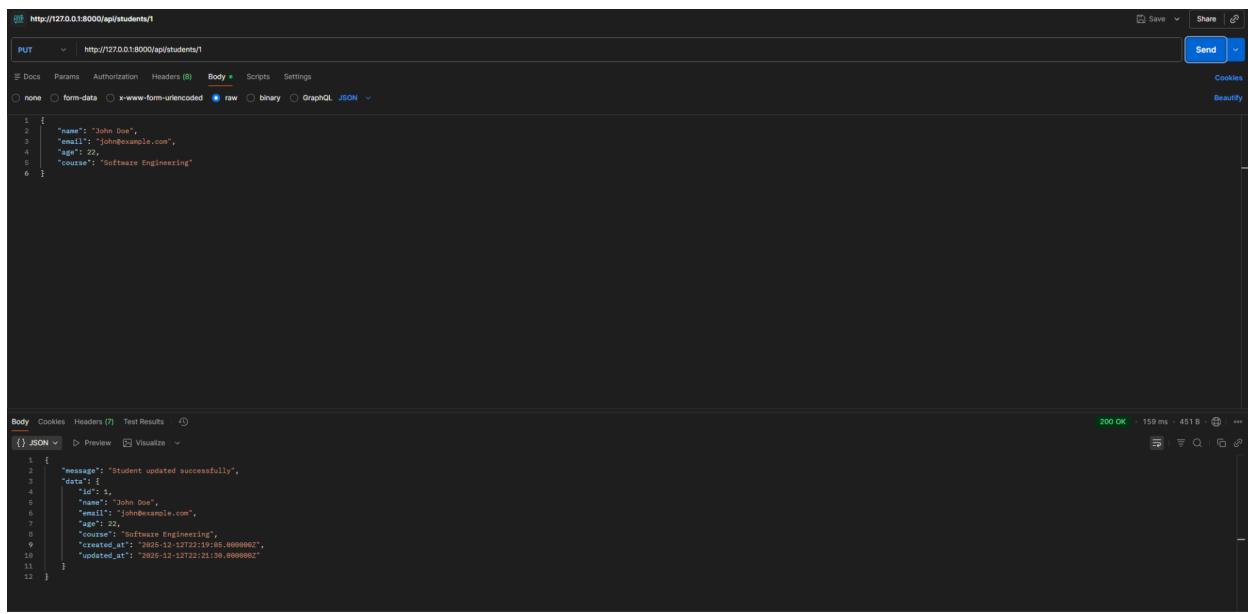
Body Cookies Headers (7) Test Results

HTML | Preview | Visualize |

```
1 {"id":1,"name":"John Doe","email":"john@example.com","age":21,"course":"Computer  
2 Science","created_at":"2023-12-12T22:19:86.000000Z","updated_at":"2023-12-12T22:19:86.000000Z"}
```

200 OK | 138 ms | 397 B |

## PUT



http://127.0.0.1:8000/api/students/1

PUT | http://127.0.0.1:8000/api/students/1

Body Cookies Headers (8) | Body | Scripts | Settings

none | form-data | x-www-form-urlencoded | raw | binary | GraphQL | JSON

```
1 { "name": "John Doe",  
2   "email": "john@example.com",  
3   "age": 22,  
4   "course": "Software Engineering"  
5 }
```

Body Cookies Headers (7) Test Results

JSON | Preview | Visualize |

```
1 { "message": "Student updated successfully",  
2   "data": [  
3     {  
4       "id": 1,  
5       "name": "John Doe",  
6       "email": "john@example.com",  
7       "age": 22,  
8       "course": "Software Engineering",  
9       "created_at": "2023-12-12T22:19:86.000000Z",  
10      "updated_at": "2023-12-12T22:21:59.000000Z"  
11    }  
12  ]}
```

200 OK | 159 ms | 451 B |

## DELETE

The screenshot shows a REST client interface with the following details:

- URL:** `http://127.0.0.1:8000/api/students/1`
- Method:** `DELETE`
- Body:** `x-www-form-urlencoded` (selected)
- Body Content:**

```
1 {
2     "name": "John Doe",
3     "email": "johndoe@example.com",
4     "age": 20,
5     "course": "Software Engineering"
6 }
```
- Response Headers:**
  - 200 OK
  - 125 ms
  - 263 B
- Response Body:**

```
1 {
2     "message": "Student deleted successfully"
3 }
```