

Task 2, Swagger Documentation with the API

Documenting an API helps a developer get a clear idea on how the API operates. This helps someone unfamiliar with the API get a clear picture of how the API works.

Now we create the documentation for our API using Swagger 3.

Getting Started

To run this project, you will need

Java 11+

Maven 3.6+

Postman or any other REST client

Installation

git clone [https://github.com/NikhilKumar2444/Kaiburr-](https://github.com/NikhilKumar2444/Kaiburr-Task-2.git)

Task-2.git

mvn clean install

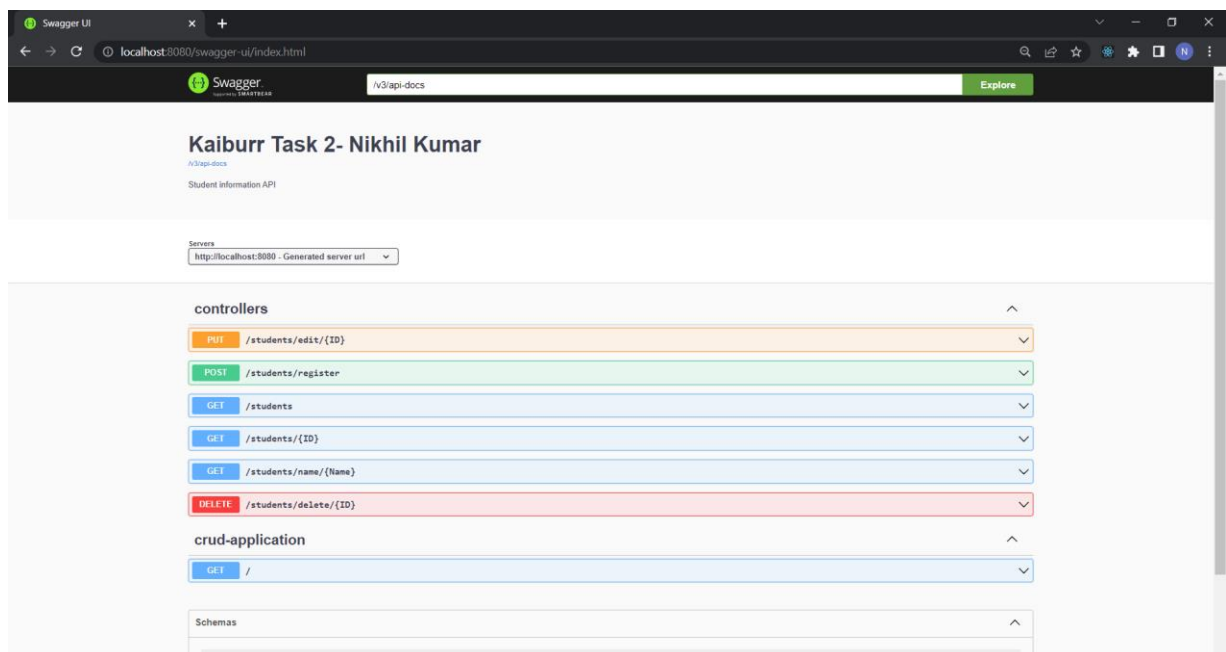
mvn package

java -jar target/crud-0.0.1-SNAPSHOT.jar

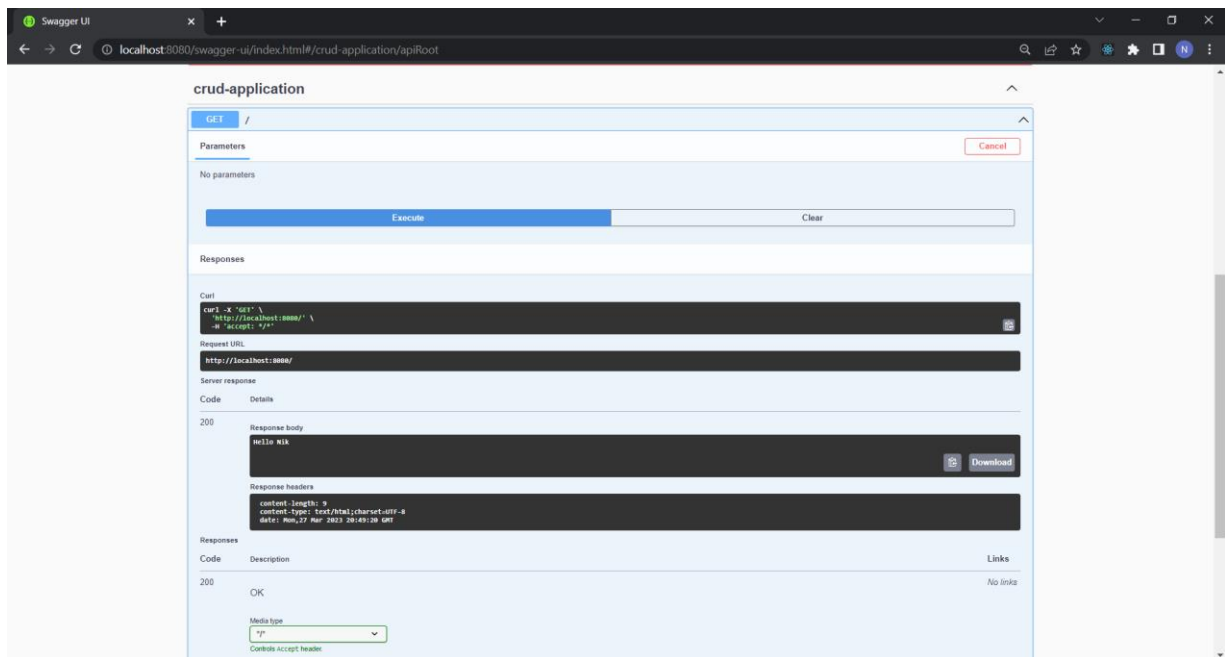
Endpoint for docs - <http://localhost:8080/v3/api-docs>

Endpoint for Swagger UI - <http://localhost:8080/swagger-ui/index.html>

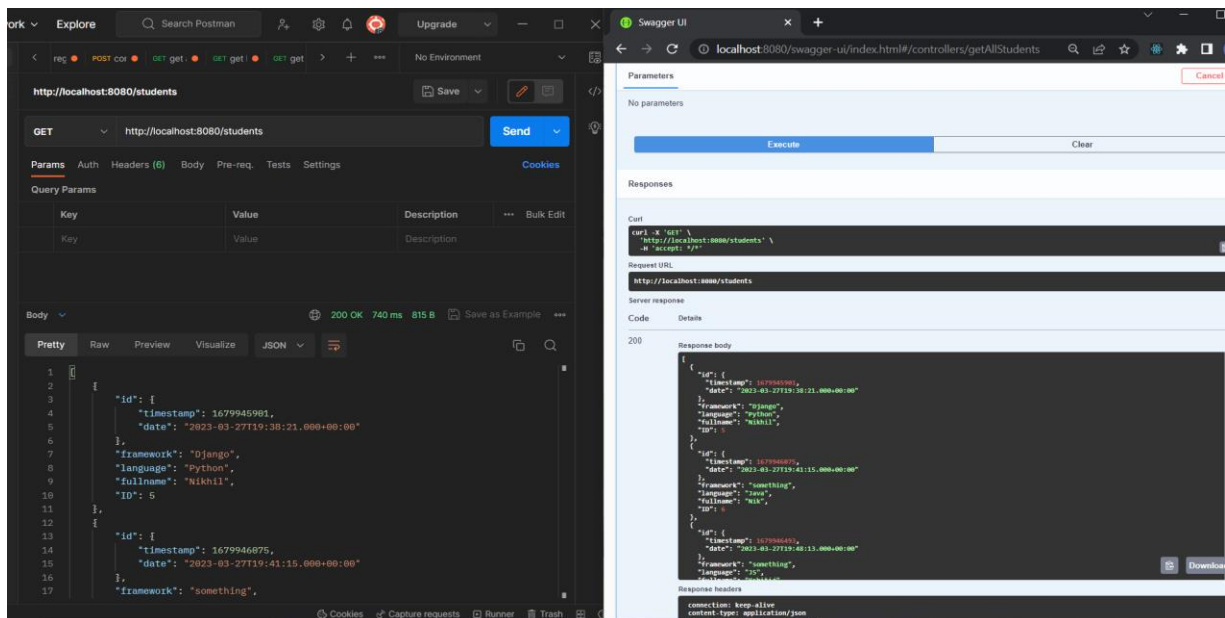
Output description and screenshots



Get("/")



Get ("/students")



Get ("/students/name/{name}")

The image shows two side-by-side screenshots. The left screenshot is from Postman, displaying a GET request to `http://localhost:8080/students/name/Nikhil`. The response body is a JSON array of two student objects. The right screenshot is from Swagger UI, showing the definition for the `GET /students/name/{name}` endpoint. The parameter `name` is required and of type `string`. The response is a `200` status with a JSON body containing the same array of student objects as seen in Postman.

```
GET http://localhost:8080/students/name/Nikhil
```

```
{
  "timestamp": 1679945961,
  "date": "2023-03-27T19:38:21.000+00:00",
  "framework": "Django",
  "language": "Python",
  "fullname": "Nikhil",
  "ID": 5
}, {
  "timestamp": 1679958666,
  "date": "2023-03-27T20:57:46.000+00:00",
  "framework": "Springboot",
  "language": "Java",
  "fullname": "Nikhil",
  "ID": 23
}]
```

Swagger UI Parameters:

Name	Description
name	required string (path)

Responses:

```
200 {
  "id": {
    "timestamp": 1679945961,
    "date": "2023-03-27T19:38:21.000+00:00",
    "framework": "Django",
    "language": "Python",
    "fullname": "Nikhil",
    "ID": 5
  }, {
    "timestamp": 1679958666,
    "date": "2023-03-27T20:57:46.000+00:00",
    "framework": "Springboot",
    "language": "Java",
    "fullname": "Nikhil",
    "ID": 23
  }
}
```

Get ("/students/{ID}")

The image shows two side-by-side screenshots. The left screenshot is from Postman, displaying a GET request to `http://localhost:8080/students/5`. The response body is a single student object with ID 5. The right screenshot is from Swagger UI, showing the definition for the `GET /students/{ID}` endpoint. The parameter `ID` is required and of type `integer`. The response is a `200` status with a JSON body containing the same student object as seen in Postman.

```
GET http://localhost:8080/students/5
```

```
{
  "id": {
    "timestamp": 1679945961,
    "date": "2023-03-27T19:38:21.000+00:00",
    "framework": "Django",
    "language": "Python",
    "fullname": "Nikhil",
    "ID": 5
  }
}
```

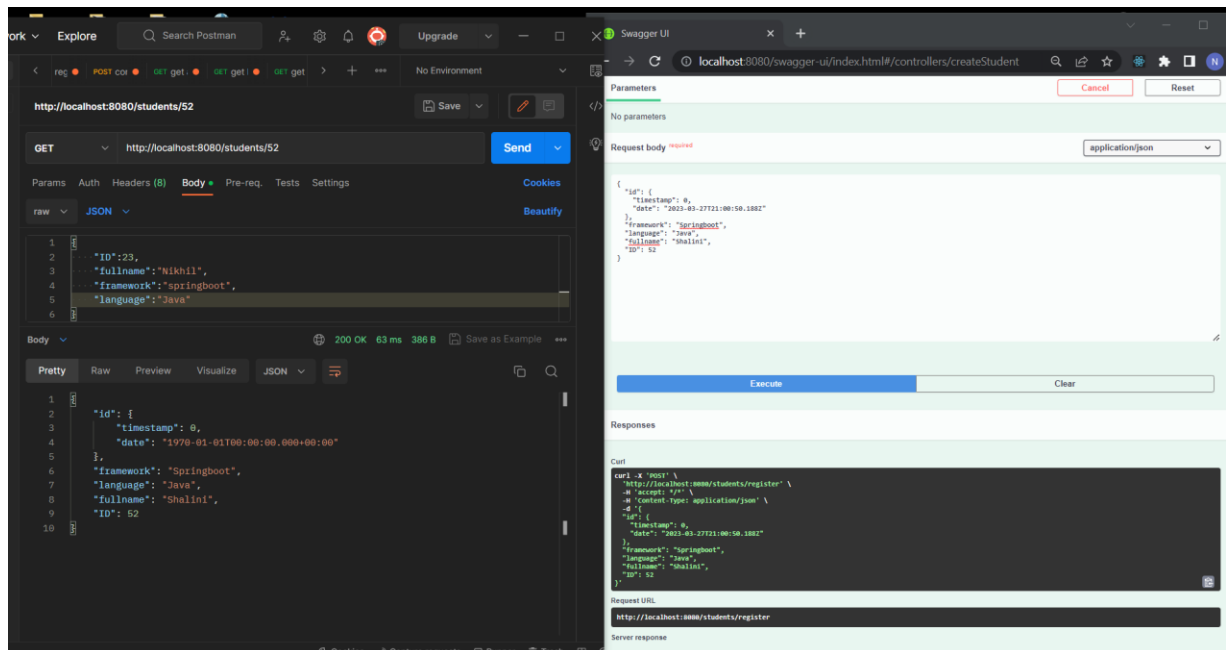
Swagger UI Parameters:

Name	Description
ID	required integer (syntax) (path)

Responses:

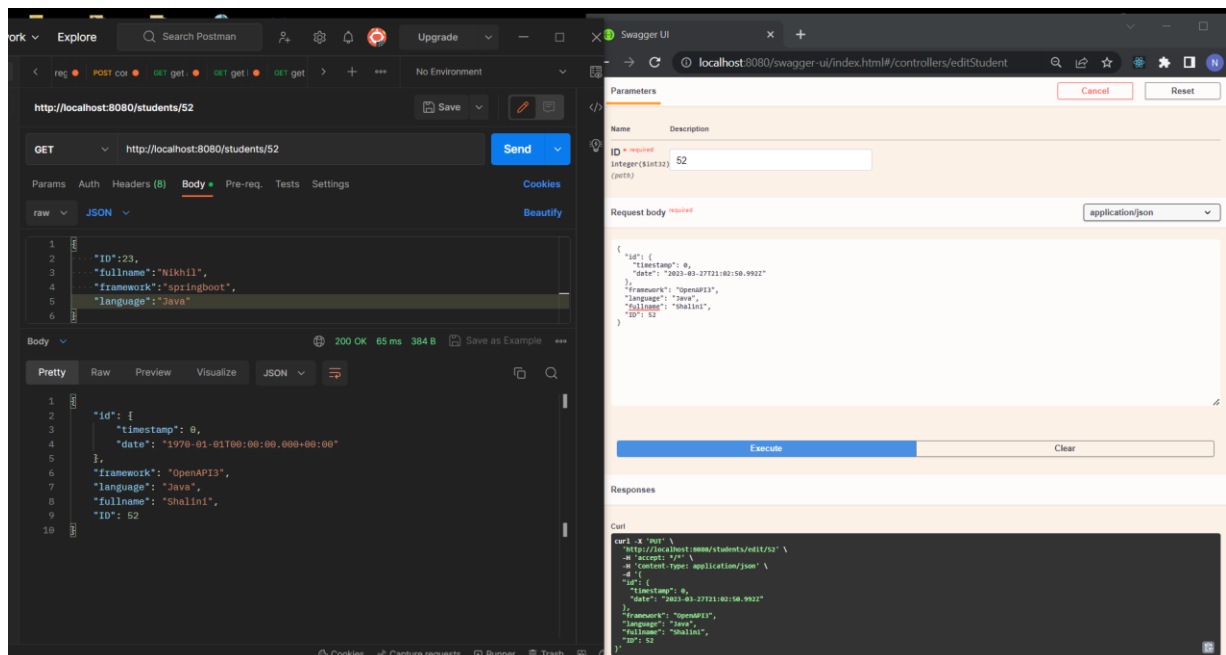
```
200 {
  "id": {
    "timestamp": 1679945961,
    "date": "2023-03-27T19:38:21.000+00:00",
    "framework": "Django",
    "language": "Python",
    "fullname": "Nikhil",
    "ID": 5
  }
}
```

POST (“students/register”) and performing get for ID=52 to verify



Performing get for ID=1 to verify

PUT (“students/edit/{ID}”) Changing “fullname” to Swagger3OpenAPI3 and then performing Get request for ID=1 to verify



Delete ("/students/delete/{ID}") Deleting ID=52:

The image shows two browser windows. The left window is Postman, displaying a GET request to `http://localhost:8080/students/52`. The response body is a JSON object: `{ "id": 52, "fullname": "Nikhil", "framework": "springboot", "language": "java" }`. The right window is Swagger UI, showing the definition for the `DELETE /students/delete/{ID}` endpoint. The parameter `ID` is defined as an integer with a format of `int32`. The response body is a JSON object: `{ "id": 52, "fullname": "Nikhil", "framework": "springboot", "language": "java" }`. The response headers are: `connection: keep-alive, content-type: application/json, date: Mon, 27 Mar 2023 21:04:11 GMT, keep-alive: timeout=60, transfer-encoding: chunked`.

Docs

The image shows the Swagger UI documentation for the `DELETE /students/delete/{ID}` endpoint. The documentation includes the following details:

- Operation:** `deleteStudent`
- Parameters:** `ID` (integer, format: int32, required: true)
- Responses:** `200` (application/json)
- Schema:** `Student`

The response body is a JSON object: `{ "id": 52, "fullname": "Nikhil", "framework": "springboot", "language": "java" }`.