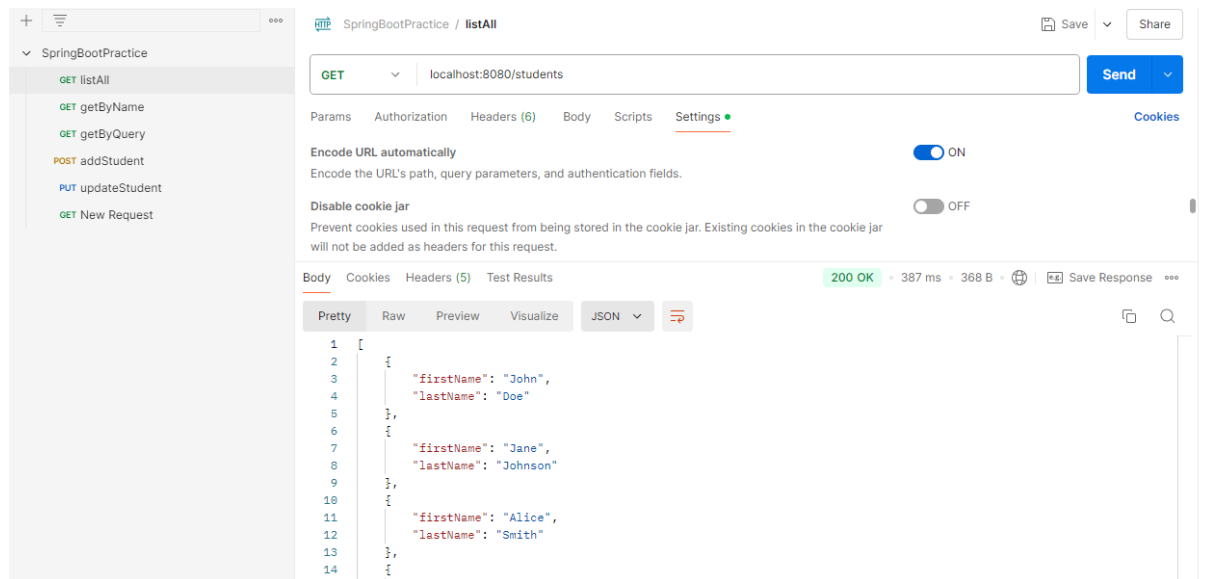


Spring Boot - Api rest, postman practice

1. List all



SpringBootPractice / listAll

GET localhost:8080/students

Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Encode URL automatically ☒ ON
Encode the URL's path, query parameters, and authentication fields.

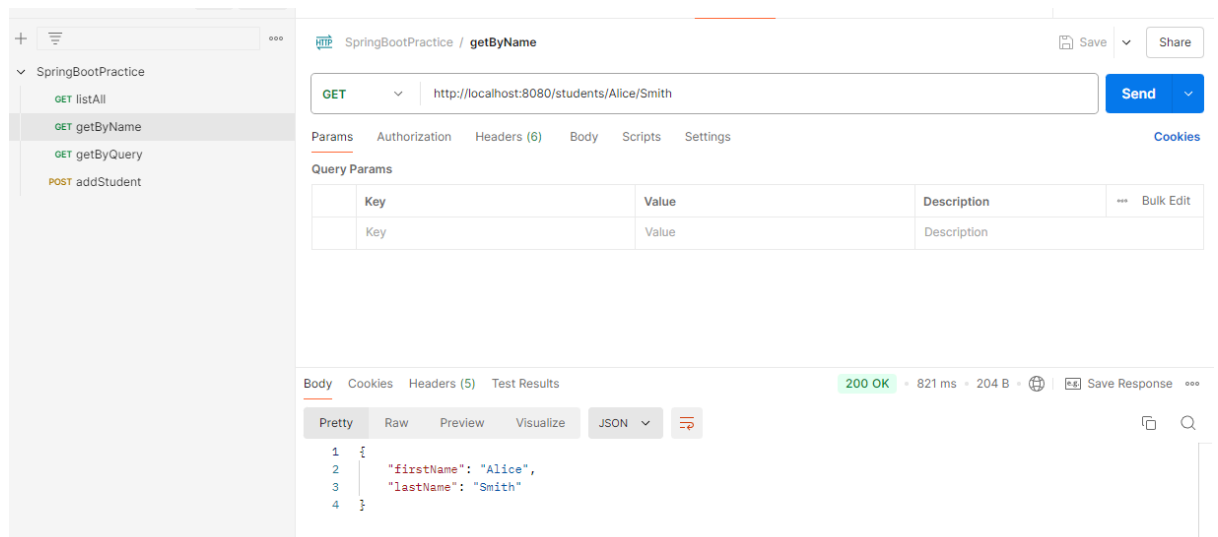
Disable cookie jar ☐ OFF
Prevent cookies used in this request from being stored in the cookie jar. Existing cookies in the cookie jar will not be added as headers for this request.

Body Cookies Headers (5) Test Results 200 OK • 387 ms • 368 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "firstName": "John",
4     "lastName": "Doe"
5   },
6   {
7     "firstName": "Jane",
8     "lastName": "Johnson"
9   },
10  {
11    "firstName": "Alice",
12    "lastName": "Smith"
13  },
14  {
```

2. Get by name



SpringBootPractice / getByName

GET http://localhost:8080/students/Alice/Smith

Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (5) Test Results 200 OK • 821 ms • 204 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "firstName": "Alice",
3   "lastName": "Smith"
4 }
```

David Malte
A00368867

3. Get by query

SpringBootPractice / **getByQuery**

GET Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/> Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/> firstName	Alice		
<input checked="" type="checkbox"/> lastName	Smith		
Key	Value	Description	

Body Cookies Headers (5) Test Results 200 OK • 28 ms • 204 B Save Response

Pretty Raw Preview Visualize JSON Beautify

```
1 {
2   "firstName": "Alice",
3   "lastName": "Smith"
4 }
```

4. Add student

SpringBootPractice / **addStudent**

POST Send

Params Authorization Headers (8) Body Scripts Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "firstName": "David",
3   "lastName": "Malte"
4 }
```

Body Cookies Headers (5) Test Results 200 OK • 1244 ms • 204 B Save Response

Pretty Raw Preview Visualize JSON Beautify

```
1 {
2   "firstName": "David",
3   "lastName": "Malte"
4 }
```

workspace New Import Overview GET Untitled R SpringBoo GET listAll GET getByNamr GET getByQue POST addStudi + No environment

SpringBootPractice / **listAll**

GET Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (5) Test Results 200 OK • 22 ms • 409 B Save Response

Pretty Raw Preview Visualize JSON Beautify

```
13 ],
14 {
15   "firstName": "Bob",
16   "lastName": "Brown"
17 },
18 {
19   "firstName": "Charlie",
20   "lastName": "Davis"
21 },
22 {
23   "firstName": "David",
24   "lastName": "Malte"
25 }
```

David Malte
A00368867

5. Update student

The screenshot displays two screenshots of a REST client interface, likely Postman, used for testing a Spring Boot API.

Top Screenshot: PUT request to update a student

- URL:** `http://localhost:8080/students/John/Doe`
- Method:** PUT
- Body:** JSON format, containing:

```
{  "firstName": "Jonathan",  "lastName": "Doe"}
```
- Response:** 200 OK, 66 ms, 205 B. The response body is shown in JSON format:

```
{  "firstName": "Jonathan",  "lastName": "Doe"}
```

Bottom Screenshot: GET request to list all students

- URL:** `localhost:8080/students`
- Method:** GET
- Settings:** "Encode URL automatically" is ON, "Disable cookie jar" is OFF.
- Response:** 200 OK, 9 ms, 330 B. The response body is shown in JSON format:

```
[  {    "firstName": "Jonathan",    "lastName": "Doe"  },  {    "firstName": "Alice",    "lastName": "Smith"  }]
```

6. Delete student

The screenshot shows a REST client interface with a sidebar on the left listing endpoints for 'SpringBootPractice'. The 'deleteStudent' endpoint is selected. The main panel shows a DELETE request to 'http://localhost:8080/students/Jane/Johnson'. The request body is a JSON object:

```
{  "firstName": "Jane",  "lastName": "Johnson"}
```

. The response is a 200 OK status with a response time of 366 ms and a body size of 205 B. The response body is also shown as a JSON object:

```
{  "firstName": "Jane",  "lastName": "Johnson"}
```

.

SpringBootPractice / deleteStudent

DELETE http://localhost:8080/students/Jane/Johnson

Params Authorization Headers (8) Body Scripts Settings

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

1 {
2 "firstName": "Jane",
3 "lastName": "Johnson"
4 }

Body Cookies Headers (5) Test Results 200 OK · 366 ms · 205 B Save Response

Pretty Raw Preview Visualize JSON

1 {
2 "firstName": "Jane",
3 "lastName": "Johnson"
4 }

7. Register user:

The screenshot shows a REST client interface with a sidebar on the left listing endpoints for 'SpringBootPractice'. The 'registerUser' endpoint is selected. The main panel shows a POST request to 'localhost:8080/register'. The request body is a JSON object:

```
{  "username": "newuser",  "password": "password123"}
```

. The response is a 200 OK status with a response time of 2.45 s and a body size of 362 B. The response body is a plain text message: 'User registered successfully'.

SpringBootPractice / registerUser

POST localhost:8080/register

Params Authorization Headers (8) Body Scripts Settings

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

1 {
2 "username": "newuser",
3 "password": "password123"
4 }

Body Cookies Headers (11) Test Results 200 OK · 2.45 s · 362 B Save Response

Pretty Raw Preview Visualize Text

1 User registered successfully

David Malte
A00368867

8. Authentica user:

The screenshot displays a REST client interface for a project named "SpringBootPractice". The left sidebar lists several endpoints, with "POST authenticateUser" selected. The main panel shows the details of this endpoint, including the method (POST), the URL (localhost:8080/authenticate), and the request body. The request body is a JSON object with the following structure:

```
2  {
3    "username": "newuser",
4    "password": "password123"
5  }
```

The response is shown in the bottom panel, indicating a successful status of 200 OK. The response body is a long alphanumeric string, likely a JWT token, displayed in the "Text" view:

```
1  eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJuZXdlc2VyIiwiaXNjaXoxNzI4MzQzMzEyLCJpYXQ1b3JlE3Mjg2MjkzMTJ9.
    WozGSR_4oyqabtYo0fSt0YVnFrHPQ1gtDC1cmPH-YUdy18-0DRmJIZq6C3noygvCtW-y1wXyHfsS_ttu1Jo_Pw
```