

Spring REST Webservices

Done by Ravi Theja Kolluru

Email ID: ravi.kolluru@accolitedigital.com

Assignment

Create a rest service with following endpoints

Get, Put, Post, Delete

Use any DB to persist your data

Use a password encoder to hash passwords

ANSWER

User Operations

POST

http://localhost:8080/api/user/signup

POST http://localhost:8080/api/user/signup

Params Authorization Headers (8) Body Pre-request Script Tests Settings

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

```
1 {
2   "emailId": "user@gmail.com",
3   "password": "pass"
4 }
```

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 2.90 s Size: 222 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "emailId": "user@gmail.com",
4   "privateMessages": null
5 }
```

	id	email_id	password
	1	user@gmail.com	\$2a\$10\$D.SzEUcFM5i.gWhdIUZsuu5CiIi.3U3Cwn...
	NULL	NULL	NULL

GET

Overview GET http://localhost:80... + ...

http://localhost:8080/api/user/get/1

GET http://localhost:8080/api/user/get/1

Params Authorization Headers (8) Body Pre-request Script Tests Settings

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

```
1 {
2   "emailId": "user@gmail.com",
3   "password": "pass"
4 }
```

Body Cookies Headers (5) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "emailId": "user@gmail.com",
4   "privateMessages": []
5 }
```

PUT

http://localhost:8080/api/user/put/1

PUT http://localhost:8080/api/user/put/1

Params Authorization Headers (8) Body Pre-request Script Tests Settings

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

```
1 {
2   "emailId": "user2@gmail.com",
3   "password": "pass2"
4 }
```

Body Cookies Headers (5) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "emailId": "user2@gmail.com",
4   "privateMessages": []
5 }
```

	id	email_id	password
▶	1	user2@gmail.com	\$2a\$10\$hT/iNBSNduHJ79h0mi/5u0NN/63tupbQbA...
✱	NULL	NULL	NULL

DELETE

Overview **DEL** http://localhost:80... + ...

http://localhost:8080/api/user/delete/1

DELETE http://localhost:8080/api/user/delete/1

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

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

```
1 {
2   "emailId": "user2@gmail.com",
3   "password": "pass2"
4 }
```

Body Cookies Headers (5) Test Results **Status: 200 OK**

Pretty Raw Preview Visualize **Text**

1 Deleted

Result Grid Filter Rows: Edit:

	id	email_id	password
✱	NULL	NULL	NULL

Message Operations

POST

The screenshot shows a REST client interface with a POST request to `http://localhost:8080/api/message/add/2`. The request body is a JSON object: `{"message": "This is message1"}`. The response is also in JSON format, showing the created message details.

Request:

```
POST http://localhost:8080/api/message/add/2
```

Body:

```
{
  "message": "This is message1"
}
```

Response:

```
{
  "id": 1,
  "message": "This is message1",
  "createTime": "2021-01-24T15:38:24.103+00:00"
}
```

Result Grid:

	id	created_time	message	user_id
1	1	2021-01-24 21:08:24.103000	This is message1	2
*	NULL	NULL	NULL	NULL

GET

GET ⌵ http://localhost:8080/api/message/get/2

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings

Headers ⌵ 8 hidden

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	password	pass2	
	Key	Value	Description

Body Cookies Headers (5) Test Results ⊕ Status: 200 OK Ti

Pretty Raw Preview Visualize JSON ⌵ ≡

```

1  {
2    ...
3    "id": 1,
4    "message": "This is message1",
5    "createdTime": "2021-01-24T15:38:24.103+00:00"
6  }
7

```

PUT

PUT ⌵ http://localhost:8080/api/message/put/2

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings

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

```

1  {
2    "message": "This is message3"
3  }

```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ⌵ ≡

```

1  {
2    "id": 3,
3    "message": "This is message3",
4    "createdTime": null
5  }

```

Result Grid				
	id	created_time	message	user_id
▶	3	NULL	This is message3	2
*	NULL	NULL	NULL	NULL

DELETE

Postman interface showing a DELETE request to `http://localhost:8080/api/message/delete/1`. The request body is a JSON object: `{\"message\": \"This is message2\"}`. The response status is 200 OK, and the body is 'Deleted'.

Result Grid				
	id	created_time	message	user_id
*	NULL	NULL	NULL	NULL

Result Grid				
	id	created_time	message	user_id
▶	2	2021-01-24 21:43:30.064000	This is message2	2
*	NULL	NULL	NULL	NULL
