

API type	link	body	description
Get	http://localhost:8090/api/likeRecipe/findAll	None	get all likes in the database
Post	http://localhost:8090/api/likeRecipe/addLikeRecipe	{ "userId": "name", "recipeId": "Apple" }	Add a new record for user to like the recipe
Post	http://localhost:8090/api/likeRecipe/findByUserId	{ "userId": "name" }	Find recipes that the user liked
Post	http://localhost:8090/api/likeRecipe/delete	{ "userId": "name", "recipeId": "Apple" }	Delete the record for user to cancel the like for the recipe

Project Note For Local Running JDK 11

```

1  server.port=8090
2
3  spring.datasource.url=jdbc:mysql://localhost:3306/recipe?serverTimezone=America/New_York
4  spring.datasource.username=root
5  spring.datasource.password=082377
6  spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
7  spring.jpa.properties.hibernate.hbm2ddl.auto=update
8  spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLInnoDBDialect
9  spring.jpa.show-sql=true
10 spring.application.name=Bootstrap Spring Boot

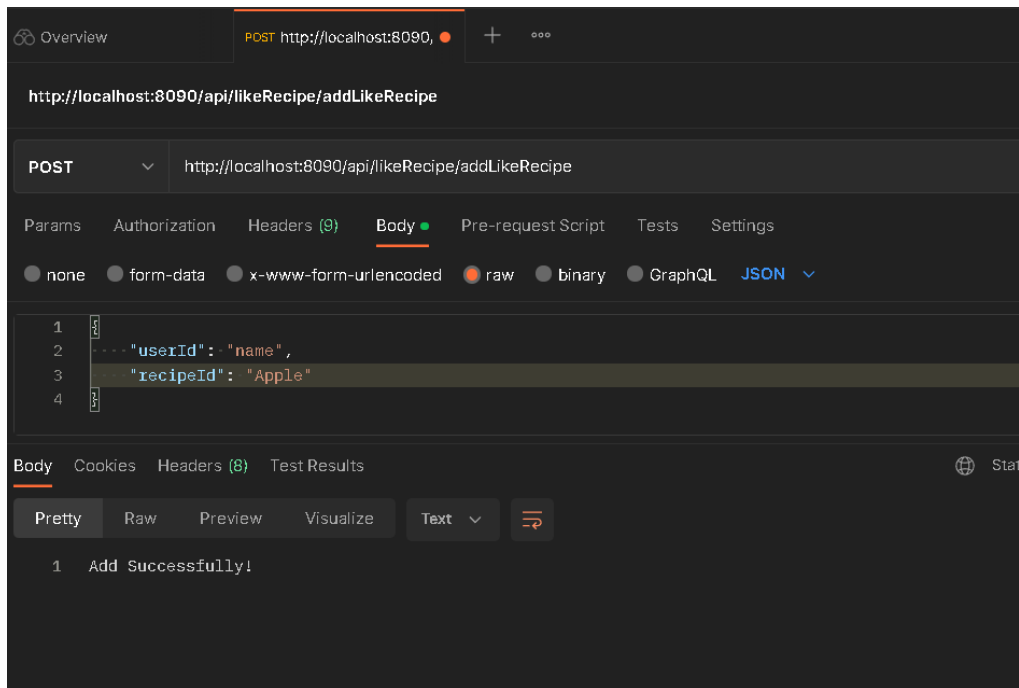
```

1. change the username and password for local MySQL database
2. create the schema named 'recipe'

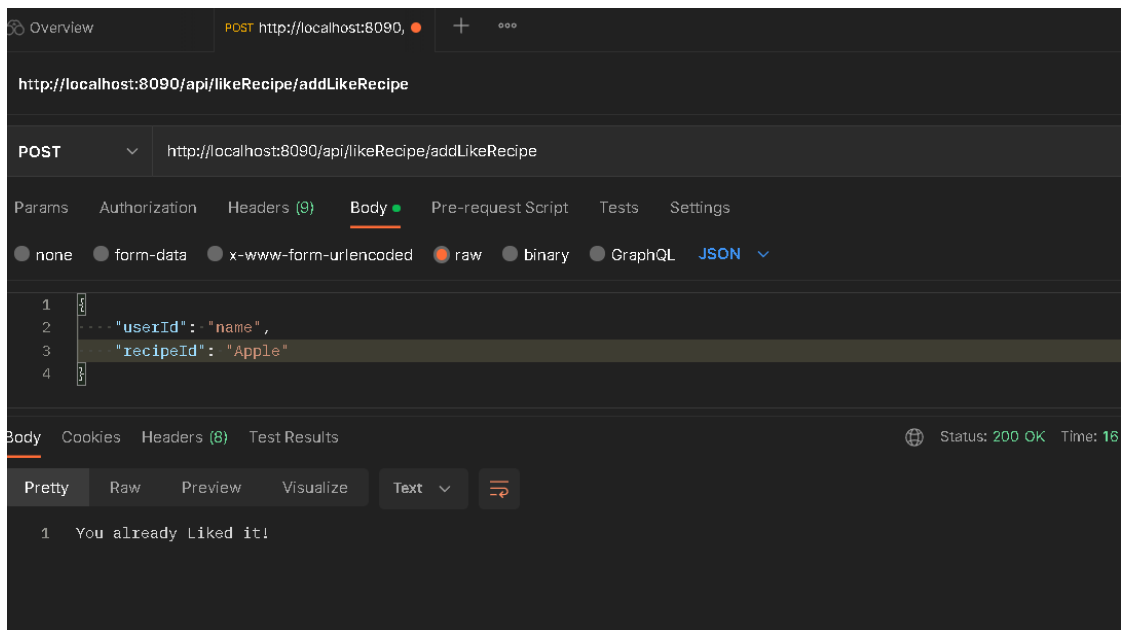
Postman Api Tests

1. AddLikeRecipe

Not existed



Already existed



2. findAll

http://localhost:8090/api/likeRecipe/findAll

GET ▼ http://localhost:8090/api/likeRecipe/findAll

Params Authorization Headers (7) Body Pre-request Script Tests Settings

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

This request does not have a body

Body Cookies Headers (8) Test Results ⌐ Status

Pretty Raw Preview Visualize JSON ▼ ≡

```
1  [
2    {
3      "id": 1,
4      "userId": "name",
5      "recipeId": "Lemon"
6    },
7    {
8      "id": 2,
9      "userId": "name",
10     "recipeId": "Apple"
11   }
12 ]
```

3. findByUserId

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

- URL:** `http://localhost:8090/api/likeRecipe/findByUserId`
- Method:** `POST`
- Body Type:** `JSON`
- Request Body:**

```
1 {  
2   "userId": "name"  
3 }
```
- Response Body:**

```
1 [  
2   {  
3     "id": 1,  
4     "userId": "name",  
5     "recipeId": "Lemon"  
6   },  
7   {  
8     "id": 2,  
9     "userId": "name",  
10    "recipeId": "Apple"  
11  }  
12 ]
```

4. delete existed

The screenshot shows a REST client interface with the URL `http://localhost:8090/api/likeRecipe/delete`. The method is **POST**. The **Body** tab is selected, showing a JSON payload:

```
{  "userId": "name",  "recipeId": "Apple"}
```

. The **Test Results** tab is also visible, showing a single line of text: `Delete Successfully!`. The status bar at the bottom right indicates a status of 200.

Not existed

The screenshot shows the same REST client interface with the URL `http://localhost:8090/api/likeRecipe/delete` and the same JSON payload:

```
{  "userId": "name",  "recipeId": "Apple"}
```

. The **Test Results** tab shows a single line of text: `There is no such recipe`. The status bar at the bottom right indicates a status of 200.