Application to manage Favorite Recipe

API’s to show all available recipes and the actions to create, update and delete a recipe.

POSTMAN tool is used to test the APIs created.

1. Add Recipe API: <http://localhost:8080/recipe/add>

Both user and admin role has access for this API.

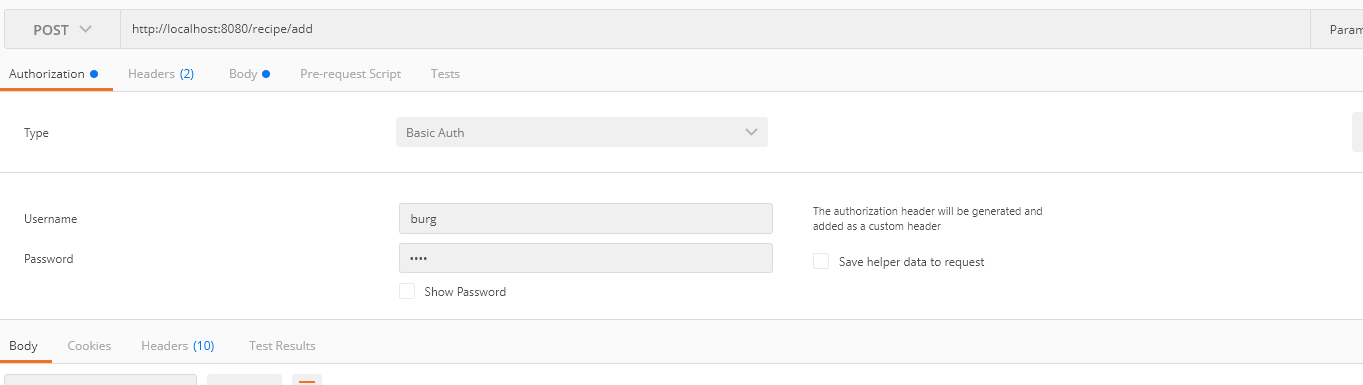
Credentials: username: burg, password: burg

Open Postman:

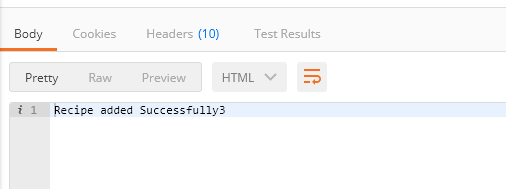
Provide add request API to add a recipe of request type POST with providing basic authentication and add a JSON body to add recipe information.

Sample Body:

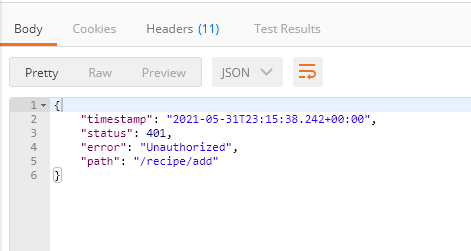
|  |
| --- |
| {  "recipeName":"Masala Dosas",  "createdDate":"01-06-2021 12:34",  "isVeg":true,  "suitableFor":"for everyone",  "ingredients":[  {  "description": "half cup idli rice or parboiled rice or 100 grams idli rice"  },  {  "description": "half cup regular rice or 100 grams regular rice"  },  {  "description": "2 tablespoon poha"  }  ],  "cooking instructions": "In a bowl take the idli rice or parboiled rice along with the regular white rice. Instead of adding regular rice, you can also make the dosa with a total of 1 cup idli rice as I have shown in the video. The video has the recipe ingredients doubled in proportion. To the same bowl, add urad dal and fenugreek seeds. Rinse the rice, lentils, and methi seeds together a couple of times and keep them aside. In a separate bowl, take the poha. Rinse it once or twice in water and then add rinsed poha to the bowl containing the rinsed rice+lentils+methi seeds. Pour 1.5 cups of water. Mix. Cover with a lid and soak everything for 5 to 6 hours."  } |







On not providing the authentication will get the below-unauthorized error.



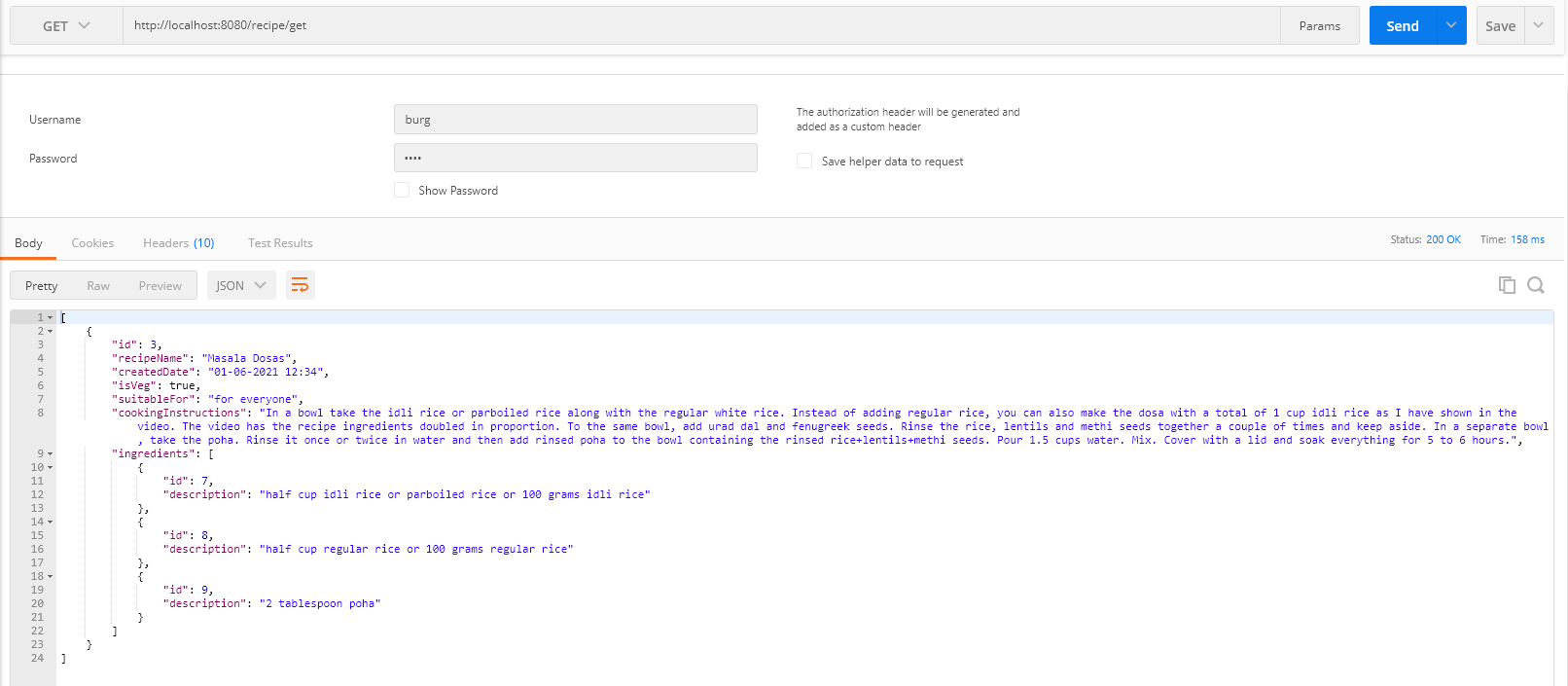
1. List all the Recipes: <http://localhost:8080/recipe/get>

Both user and admin role has access for this api.

Credentials: username: burg, password: burg

Open Postman:

Provide get request api to list all recipes added with providing basic authentication information.



Sample Response:

|  |
| --- |
| [  {  "id": 1,  "recipeName": "Masala Dosa",  "createdDate": "01-06-2021 12:34",  "isVeg": true,  "suitableFor": "for everyone",  "cookingInstructions": "In a bowl take the idli rice or parboiled rice along with the regular white rice. Instead of adding regular rice, you can also make the dosa with a total of 1 cup idli rice as I have shown in the video. The video has the recipe ingredients doubled in proportion. To the same bowl, add urad dal and fenugreek seeds. Rinse the rice, lentils and methi seeds together a couple of times and keep aside. In a separate bowl, take the poha. Rinse it once or twice in water and then add rinsed poha to the bowl containing the rinsed rice+lentils+methi seeds. Pour 1.5 cups water. Mix. Cover with a lid and soak everything for 5 to 6 hours.",  "ingredients": [  {  "id": 1,  "description": "half cup idli rice or parboiled rice or 100 grams idli rice"  },  {  "id": 2,  "description": "half cup regular rice or 100 grams regular rice"  },  {  "id": 3,  "description": "2 tablespoon poha"  }  ]  }  ] |

1. Edit/Update Recipe API: <http://localhost:8080/recipe/update>

Only **admin** role has access for this api.

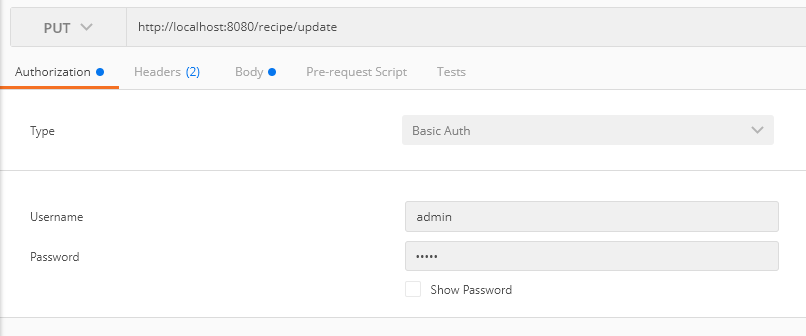
Credentials: username: admin, password: admin

Open Postman:

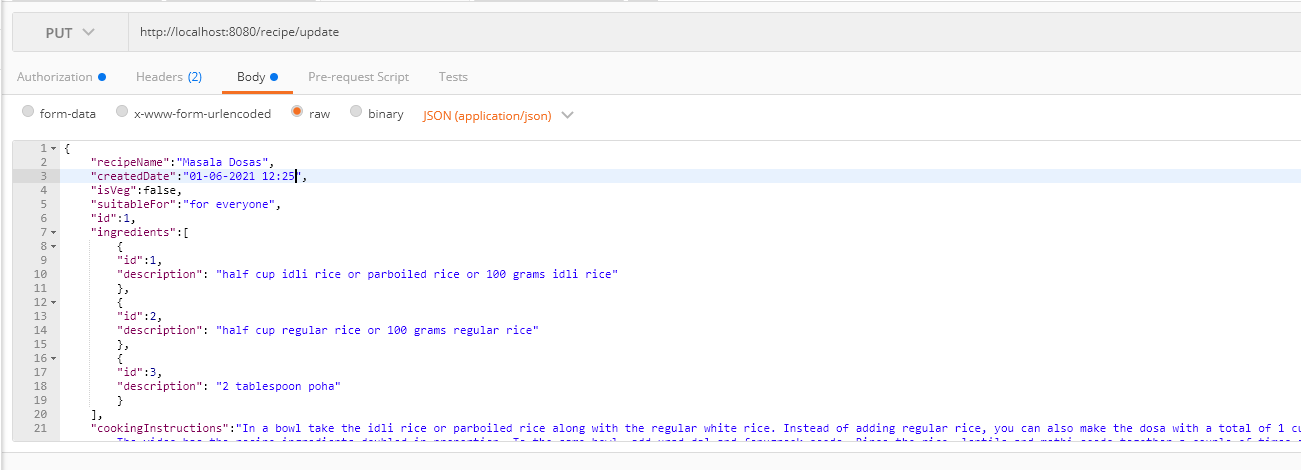
Provide put request api to update recipe added with providing basic authentication information.

Sample Request Body:

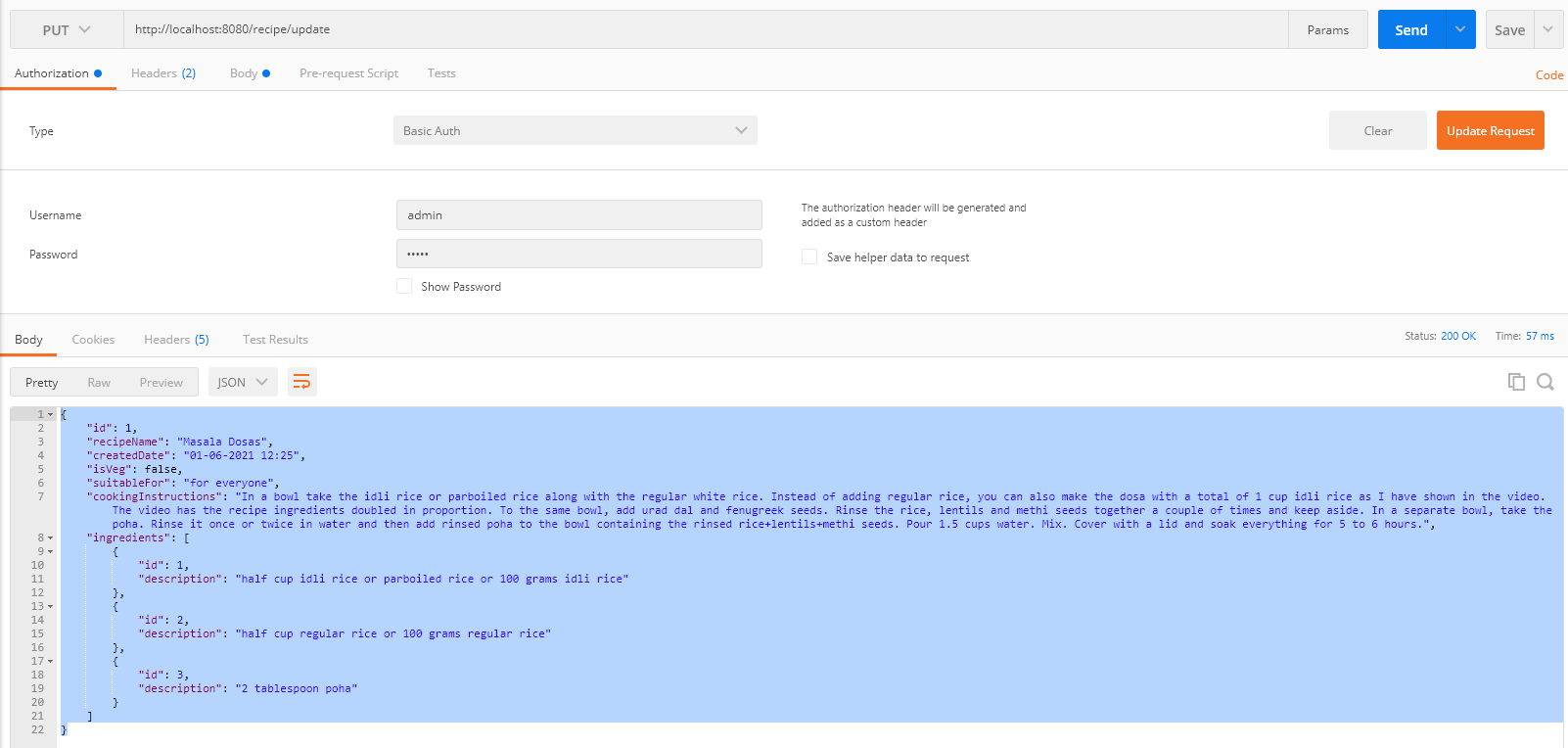
|  |
| --- |
| {  "id": 1,  "recipeName": "Masala Dosas",  "createdDate": "01-06-2021 12:25",  "isVeg": false,  "suitableFor": "for everyone",  "cookingInstructions": "In a bowl take the idli rice or parboiled rice along with the regular white rice. Instead of adding regular rice, you can also make the dosa with a total of 1 cup idli rice as I have shown in the video. The video has the recipe ingredients doubled in proportion. To the same bowl, add urad dal and fenugreek seeds. Rinse the rice, lentils and methi seeds together a couple of times and keep aside. In a separate bowl, take the poha. Rinse it once or twice in water and then add rinsed poha to the bowl containing the rinsed rice+lentils+methi seeds. Pour 1.5 cups water. Mix. Cover with a lid and soak everything for 5 to 6 hours.",  "ingredients": [  {  "id": 1,  "description": "half cup idli rice or parboiled rice or 100 grams idli rice"  },  {  "id": 2,  "description": "half cup regular rice or 100 grams regular rice"  },  {  "id": 3,  "description": "2 tablespoon poha"  }  ]  } |



Request JSON:



Response updated JSON data:



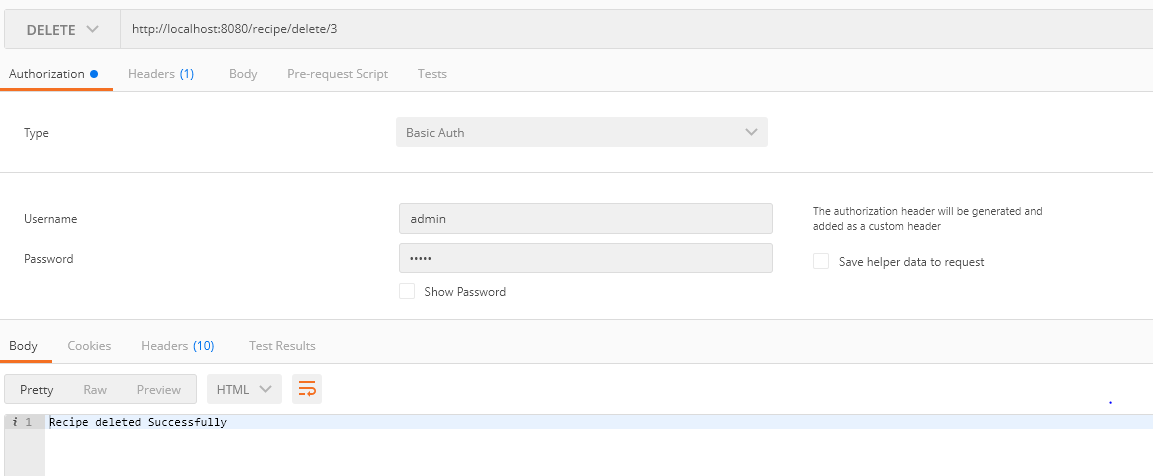
1. Delete Recipe API : <http://localhost:8080/recipe/delete/1>

Only **admin** role has access to this API.

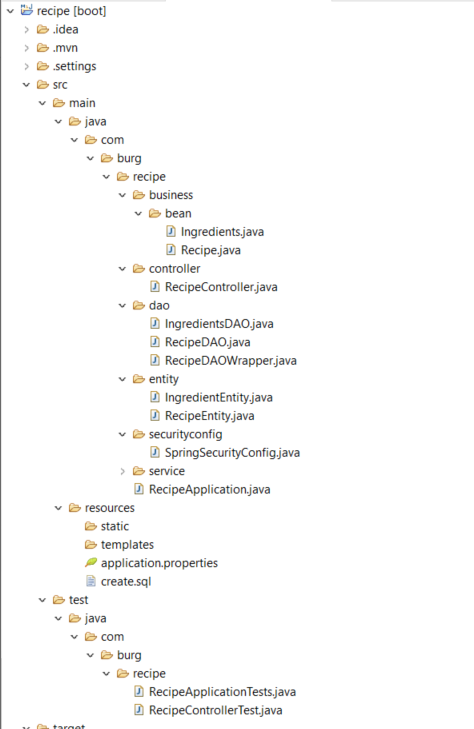
Credentials: username: admin, password: admin

Open Postman:

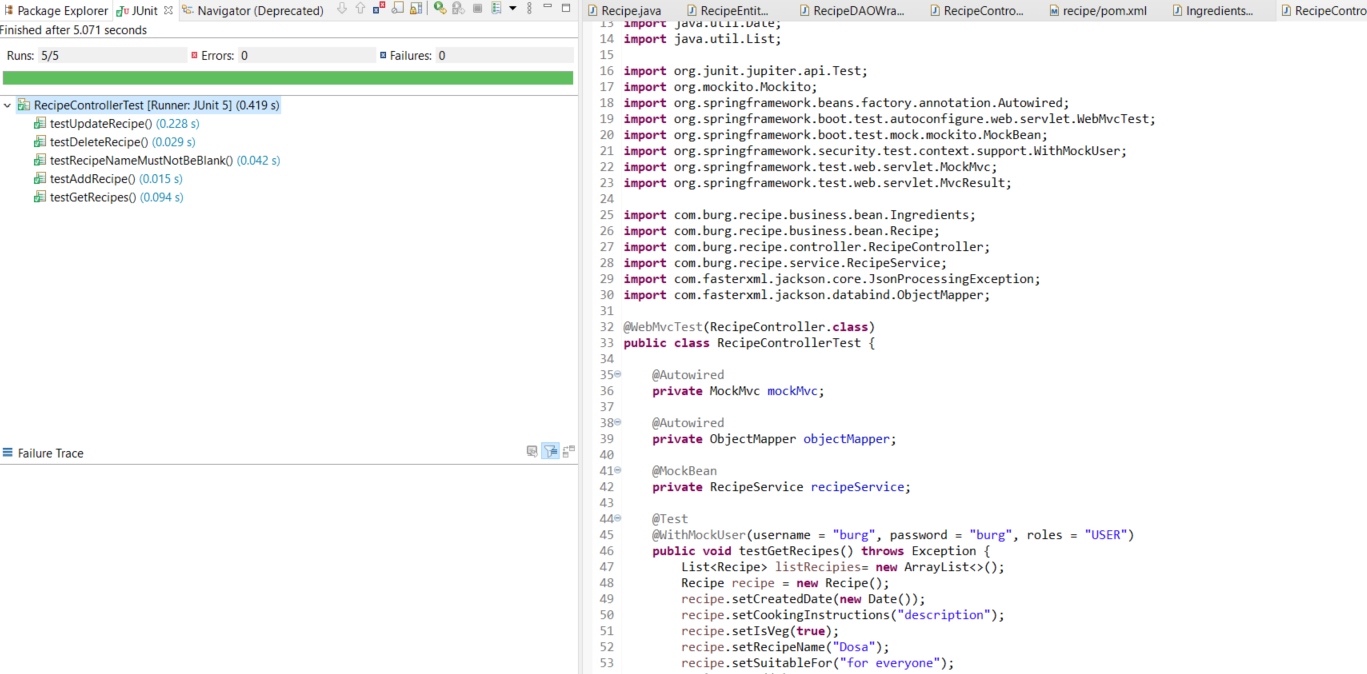
Provide delete request API to delete recipe added with providing basic authentication information.



Project Structure:



Junit Test cases Executed:



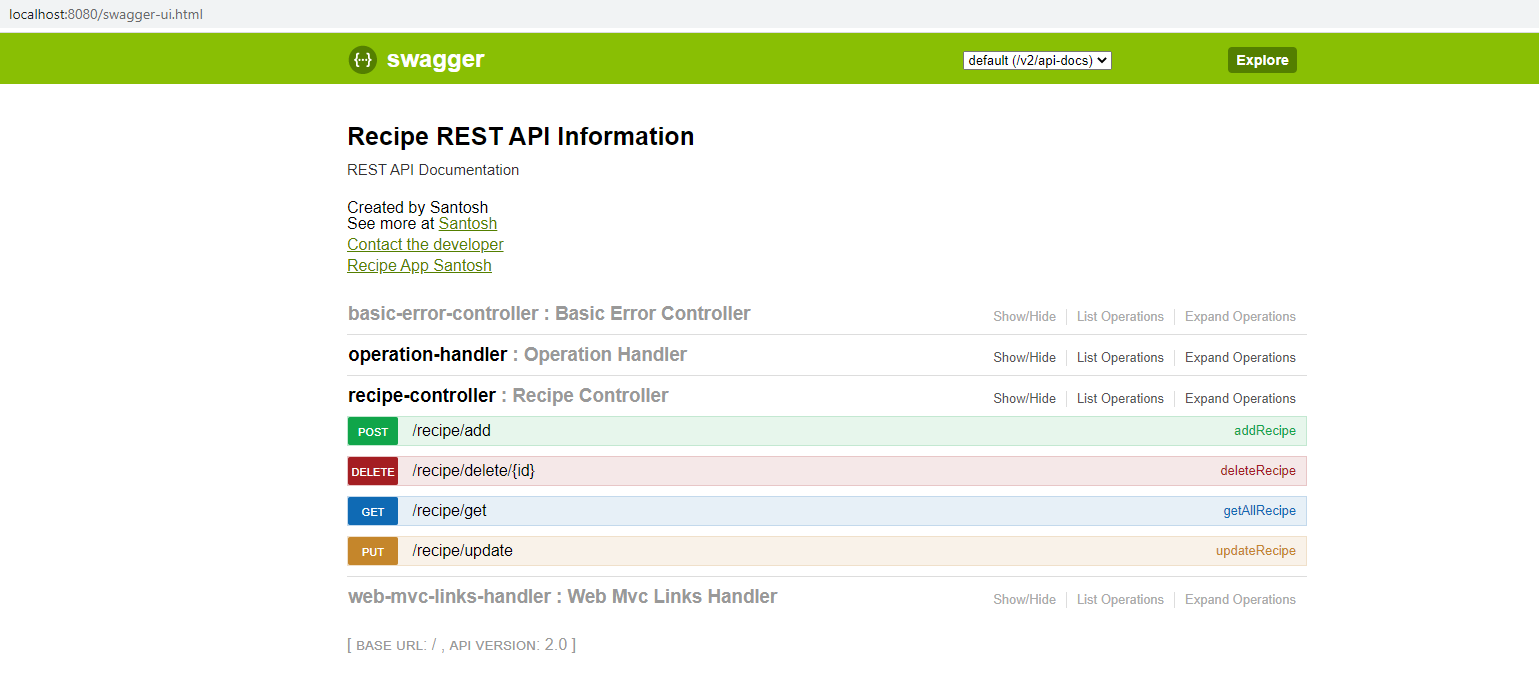
Spring Boot dependencies added

|  |
| --- |
| <dependencies>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-actuator</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-data-jpa</artifactId>  </dependency>  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <scope>runtime</scope>  <version>8.0.25</version>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-security</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>  <dependency>  <groupId>com.fasterxml.jackson.core</groupId>  <artifactId>jackson-core</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-test</artifactId>  <scope>test</scope>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-validation</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.security</groupId>  <artifactId>spring-security-test</artifactId>  <scope>test</scope>  </dependency>  <dependency>  <groupId>io.springfox</groupId>  <artifactId>springfox-swagger2</artifactId>  <version>2.7.0</version>  </dependency>  <dependency>  <groupId>io.springfox</groupId>  <artifactId>springfox-swagger-ui</artifactId>  <version>2.7.0</version>  </dependency>  </dependencies> |

Added Swagger dependency to illustrate the use of API in a single page i.e REST API Documentation

URL: <http://localhost:8080/swaggerui-html>

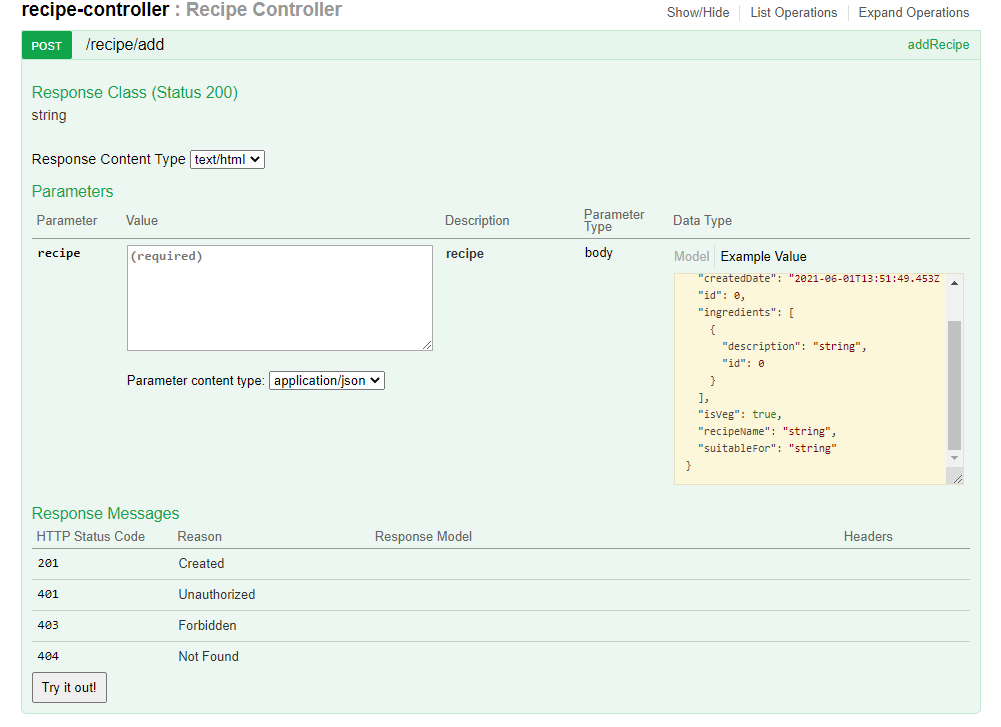
Click on the Recipe Controller link to see list of REST API operations.



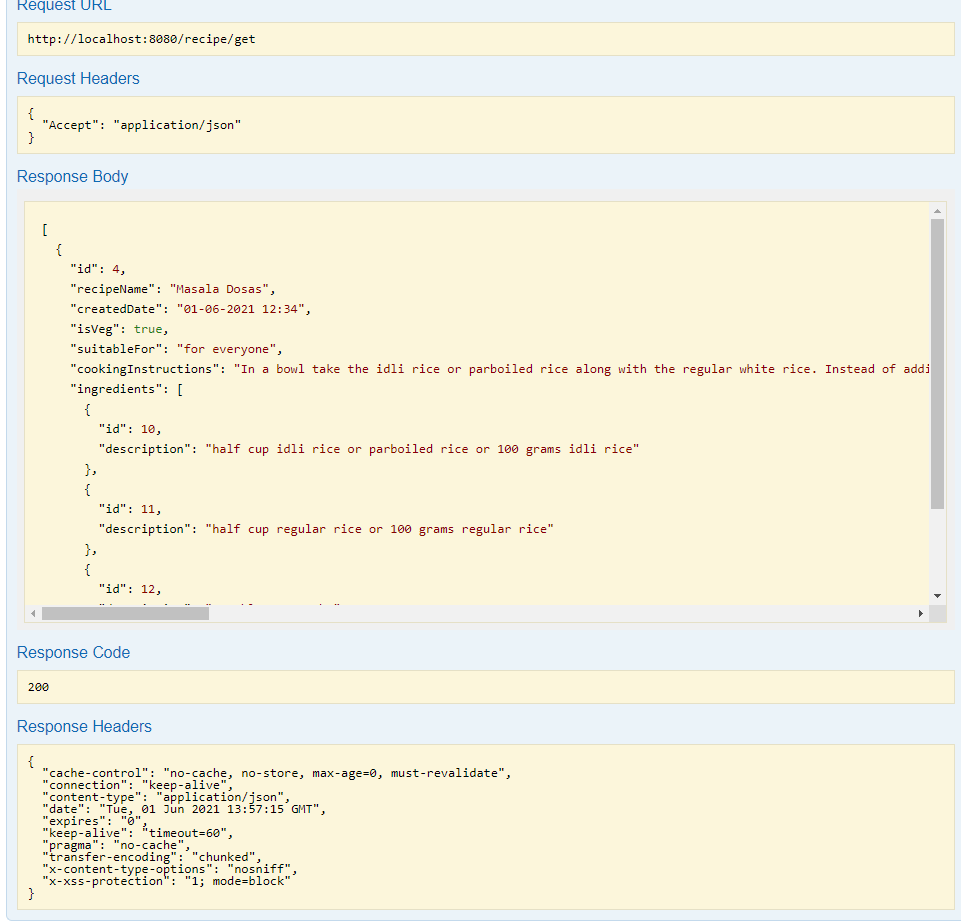
Click on the operations to test the API example datatype to be passed will be provided on the right side.

Provide the parameters and click on the try it out button to see the results. Provide the authentication details.

Username: burg Password: burg



Snapshot showing the list of all recipes on click of Try it out!.



Create Database with name food\_point and then run the application to start testing the APIs:

Note spring security is added to provide authentication and authorization while accessing the application:

Roles created: User and Admin

User role can access only list Recipe and add recipe functionalities.

Admin role can access all the 4 CRUD functionalities i.e list, add, update and delete recipe.

Credentials

Role: User Username: burg Password: burg

Role: Admin Username: admin Password: admin

|  |
| --- |
| Sample DB configuration:  spring.datasource.url=jdbc:mysql://localhost:3306/food\_point  spring.datasource.username=root  spring.datasource.password=  spring.datasource.driver-class-name==com.mysql.jdbc.Driver  spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect  spring.jpa.show-sql=true  spring.jpa.hibernate.ddl-auto=update |

**Tools/Technologies used:**

* Spring Tool Suite 4
* Java 1.8
* Maven
* Spring Boot 2.4.5
* Rest API
* Microservices
* MySQL database
* POSTMAN