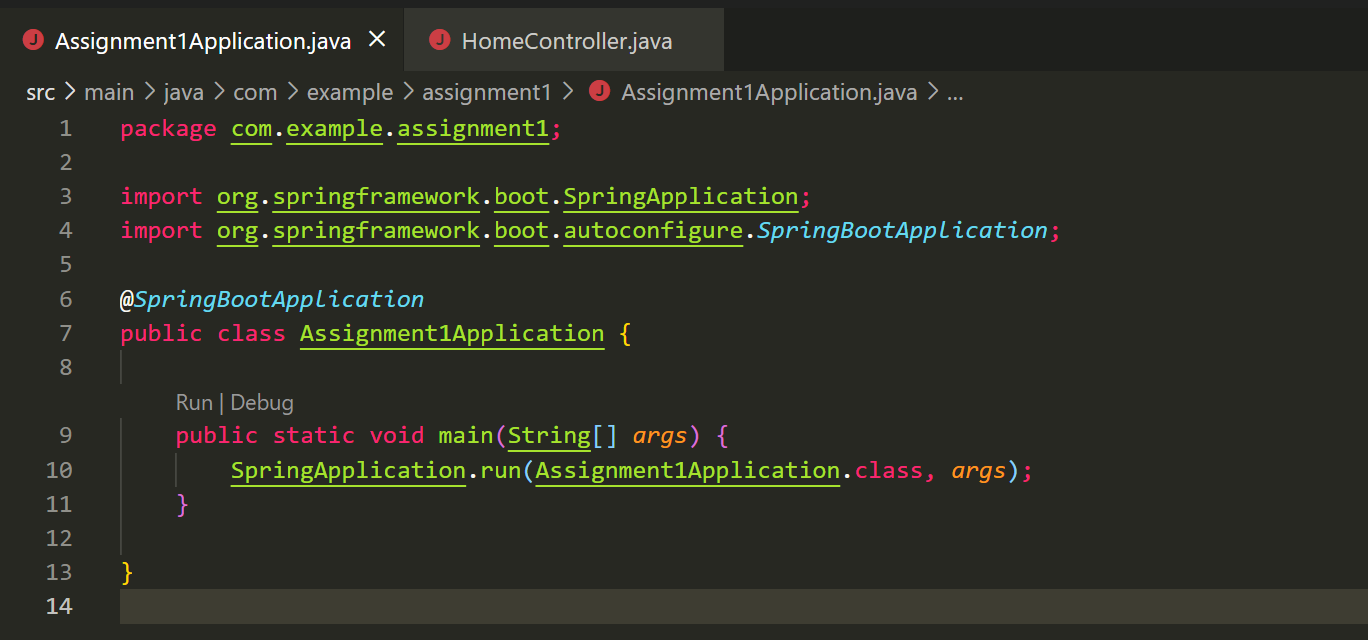
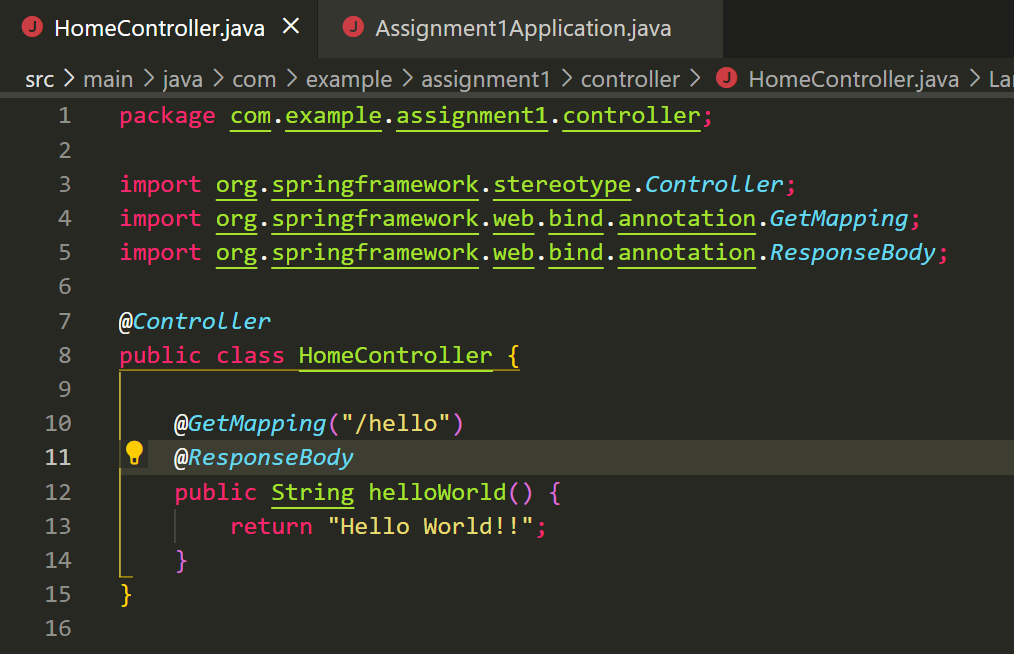
**Spring REST Assignments**

1) Create a RESTful web service that returns "Hello World" message.

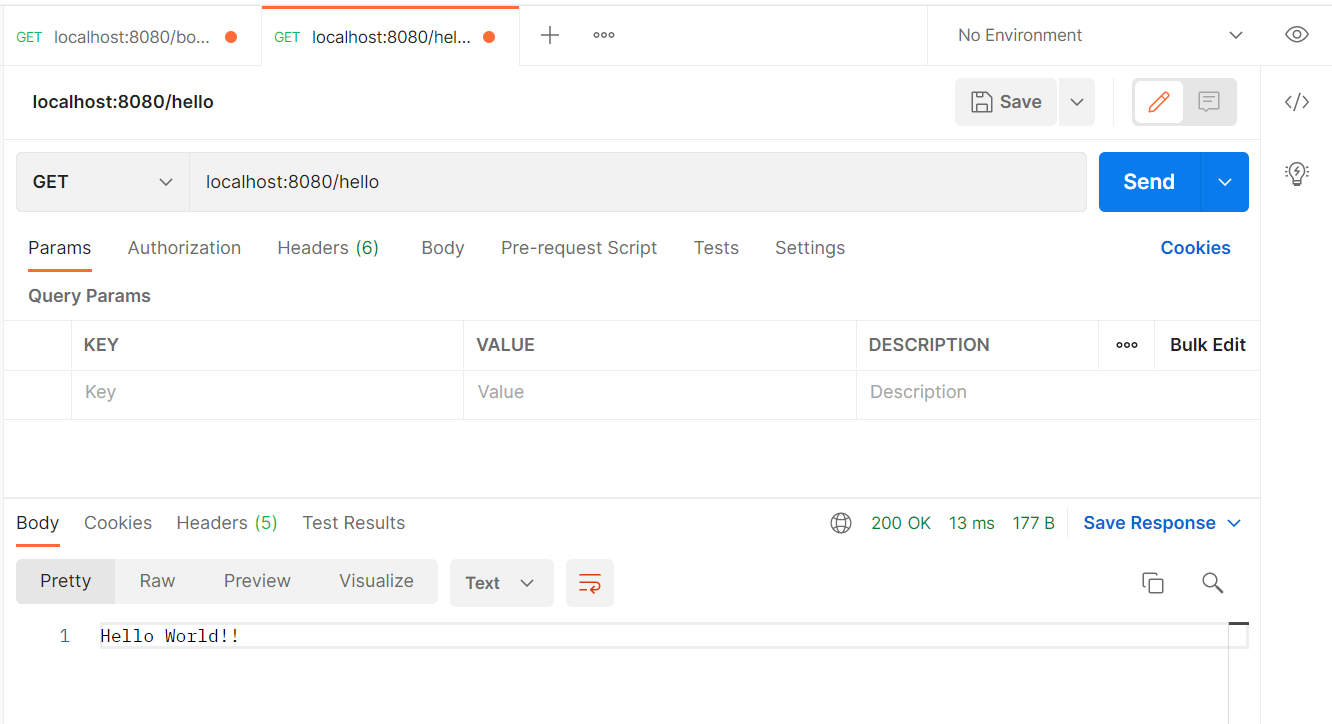
**Application.java**



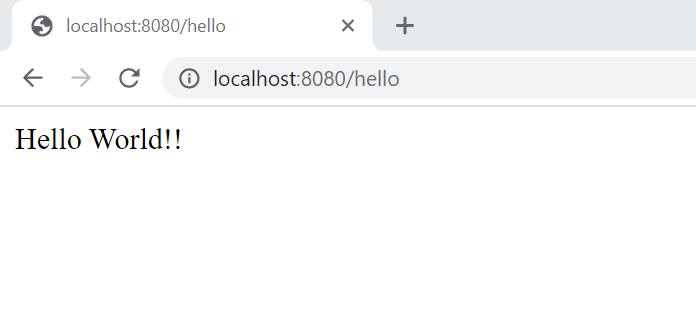
**HomeController.java**



**POSTMAN:**

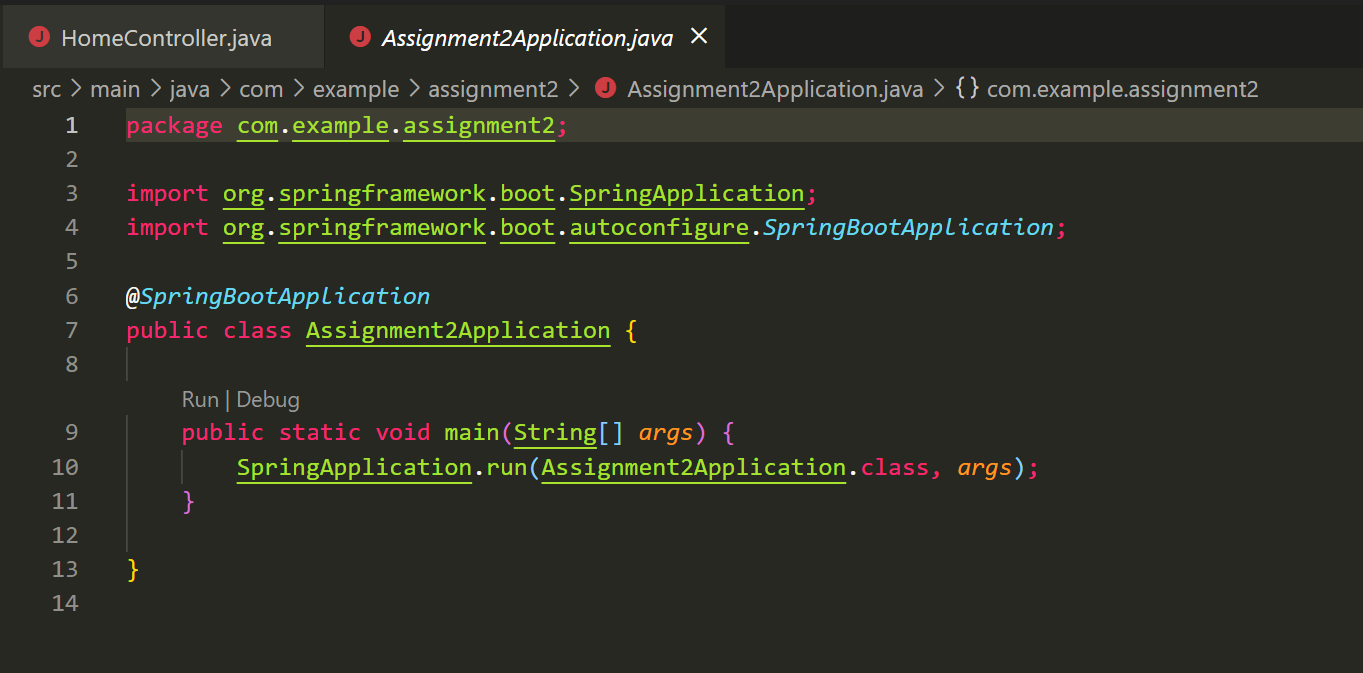
****

**WEB:**

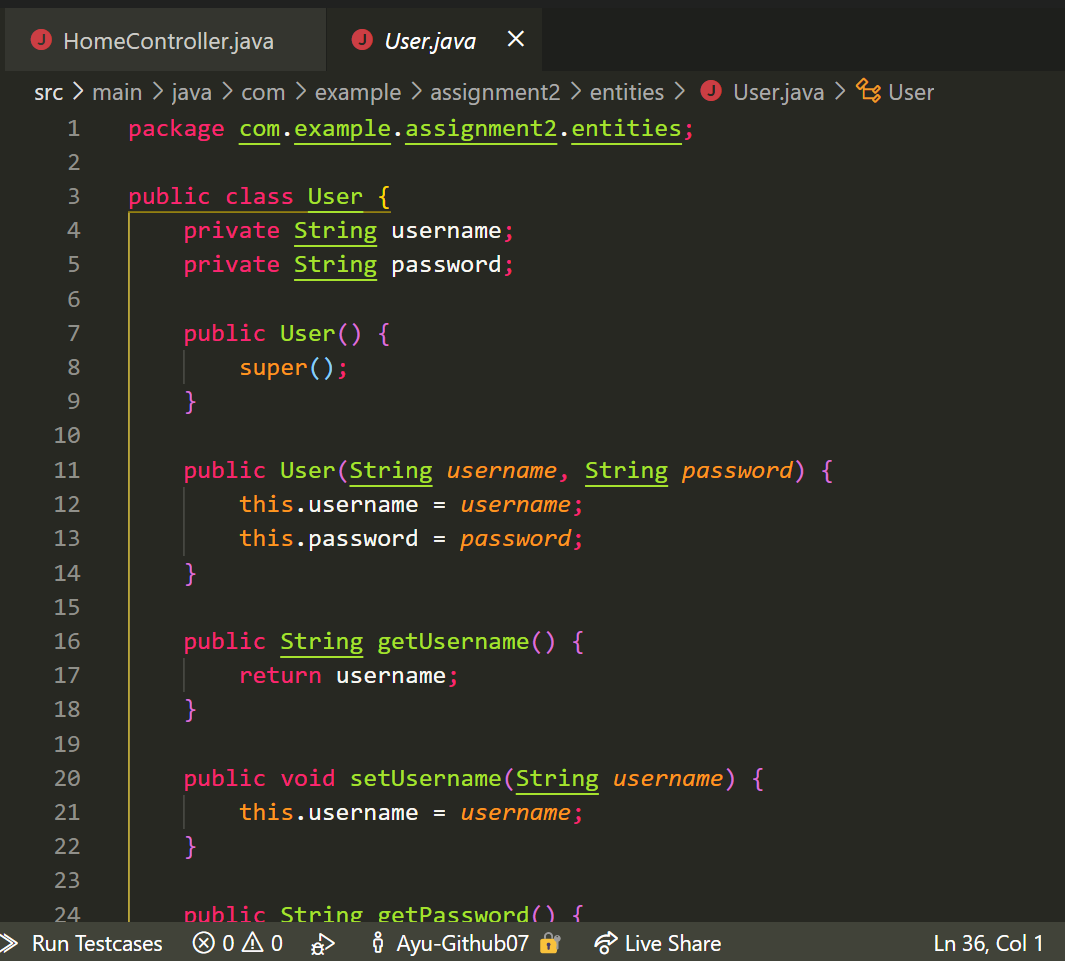
****

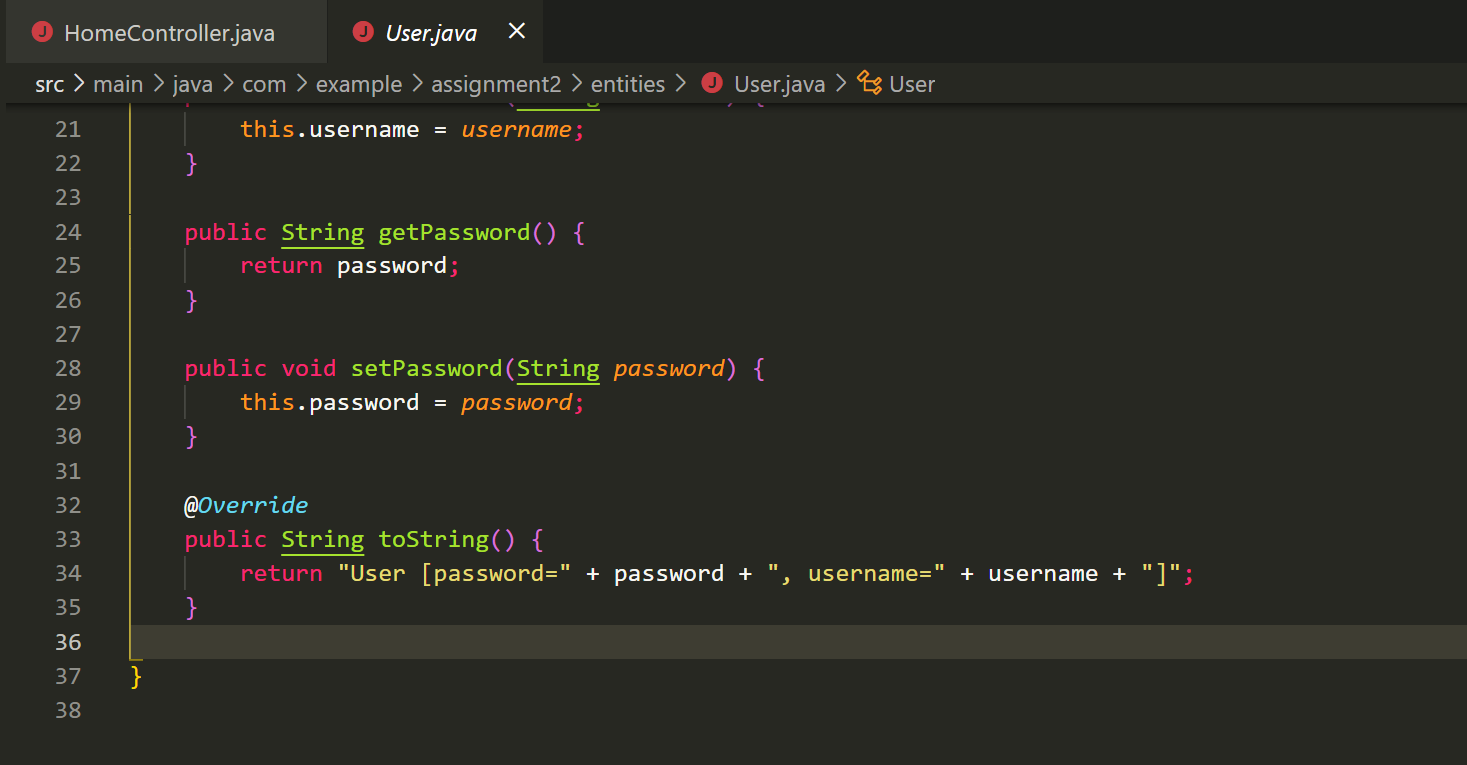
2) Create a RESTful web service that authenticates a user. User will specify his/her credentials i.e. username and password. If username and password are correct, It should return "valid user" message, else "Invalid user" message.

**Application.java**

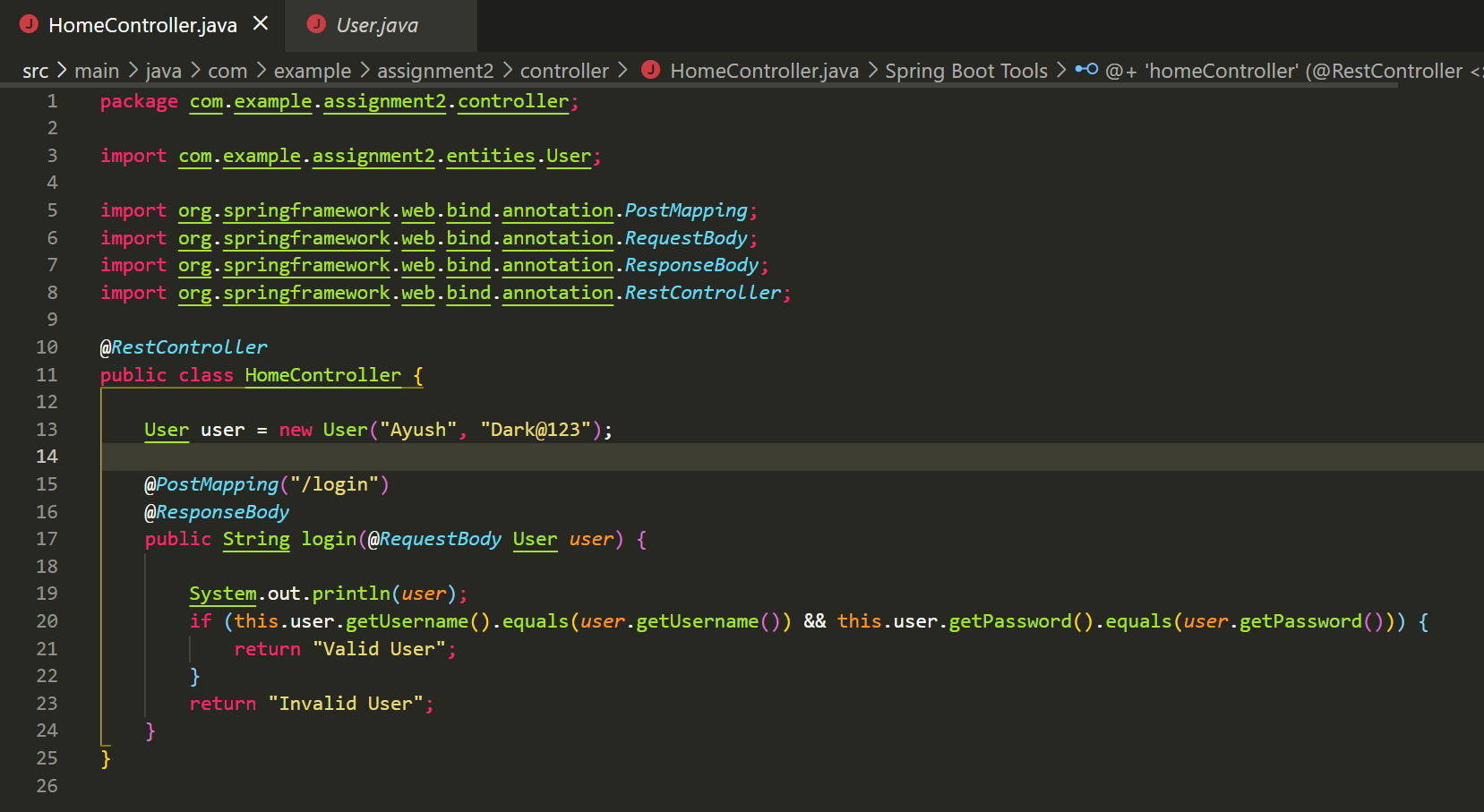


**User.java**

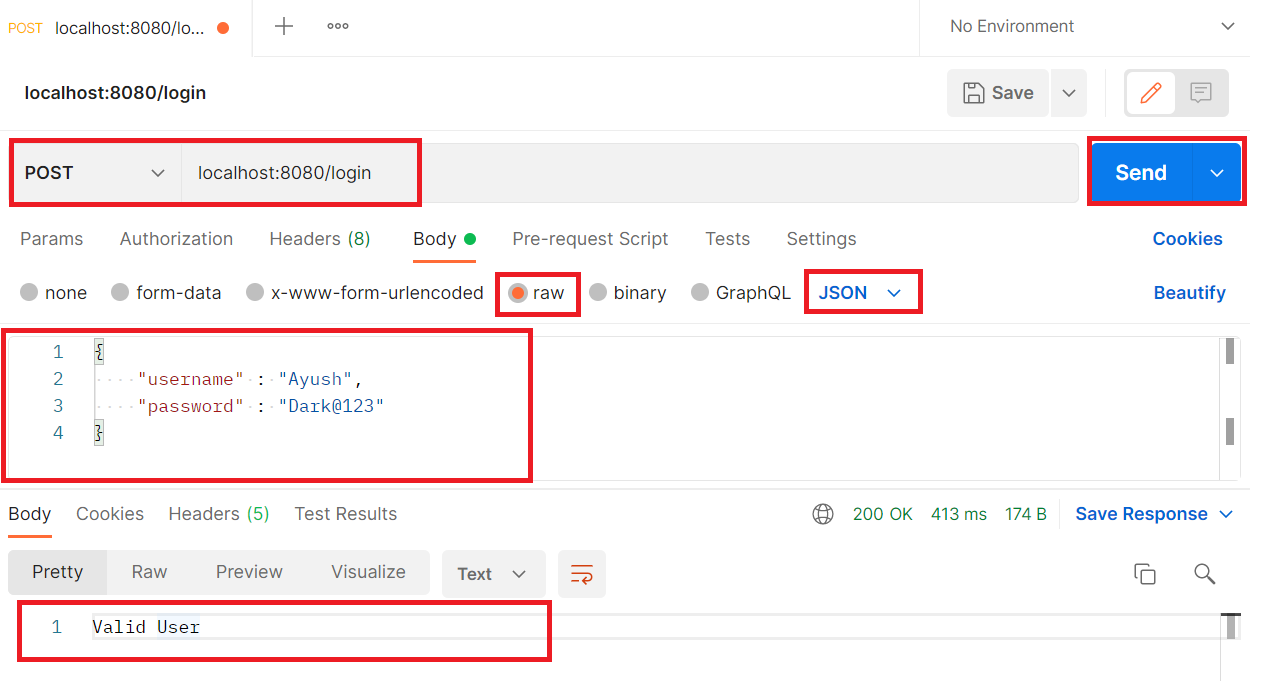
****

****

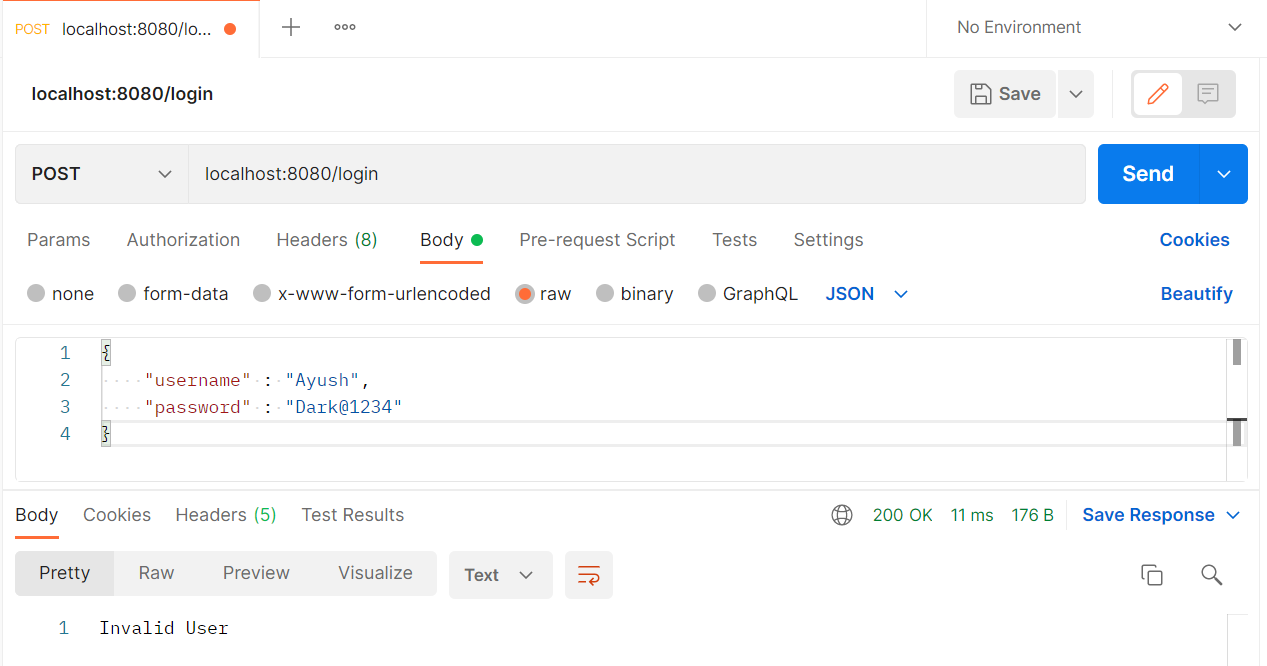
**Controller.java**

****

**Postman for Valid User**



**Postman for Invalid User**

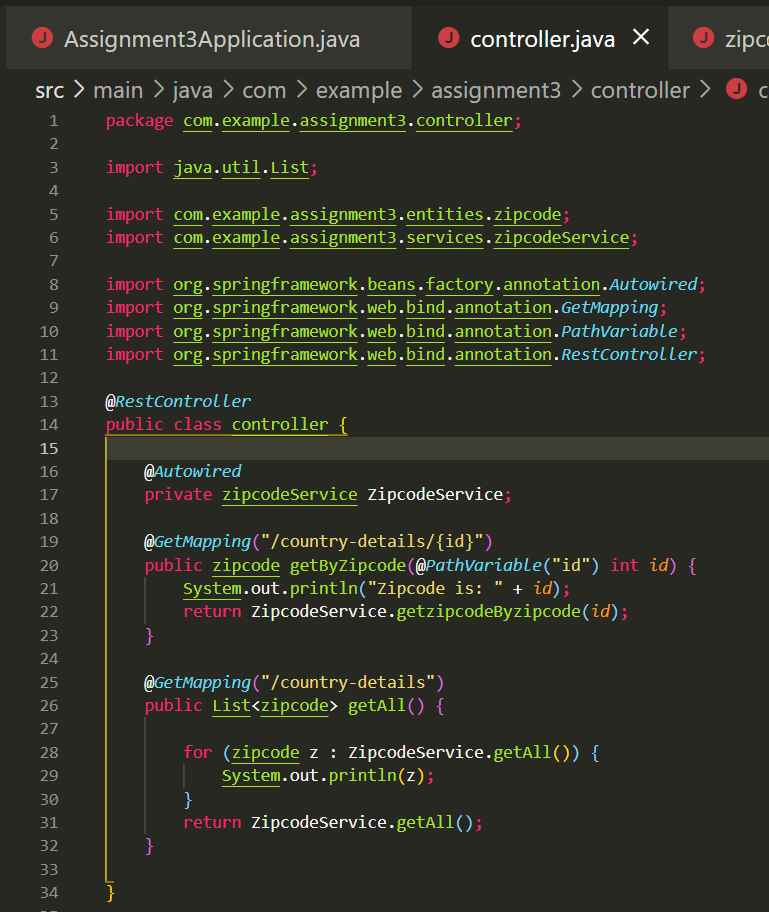
****

3) Create a RESTful web service that returns state, city and country information when user passes zipcode. You can send state, city and country information in JSON format.

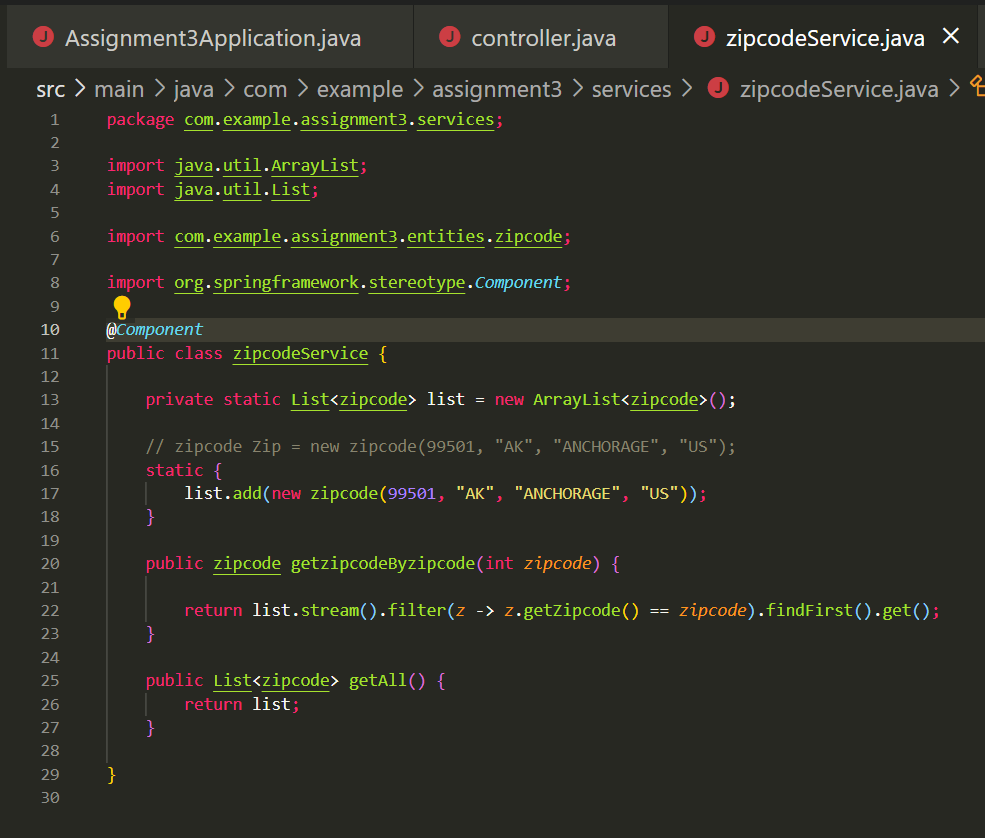
Sample Input: 99501

Sample output: {"state": "AK", City: "ANCHORAGE", "country: "US"}

**Controller.java:**

****

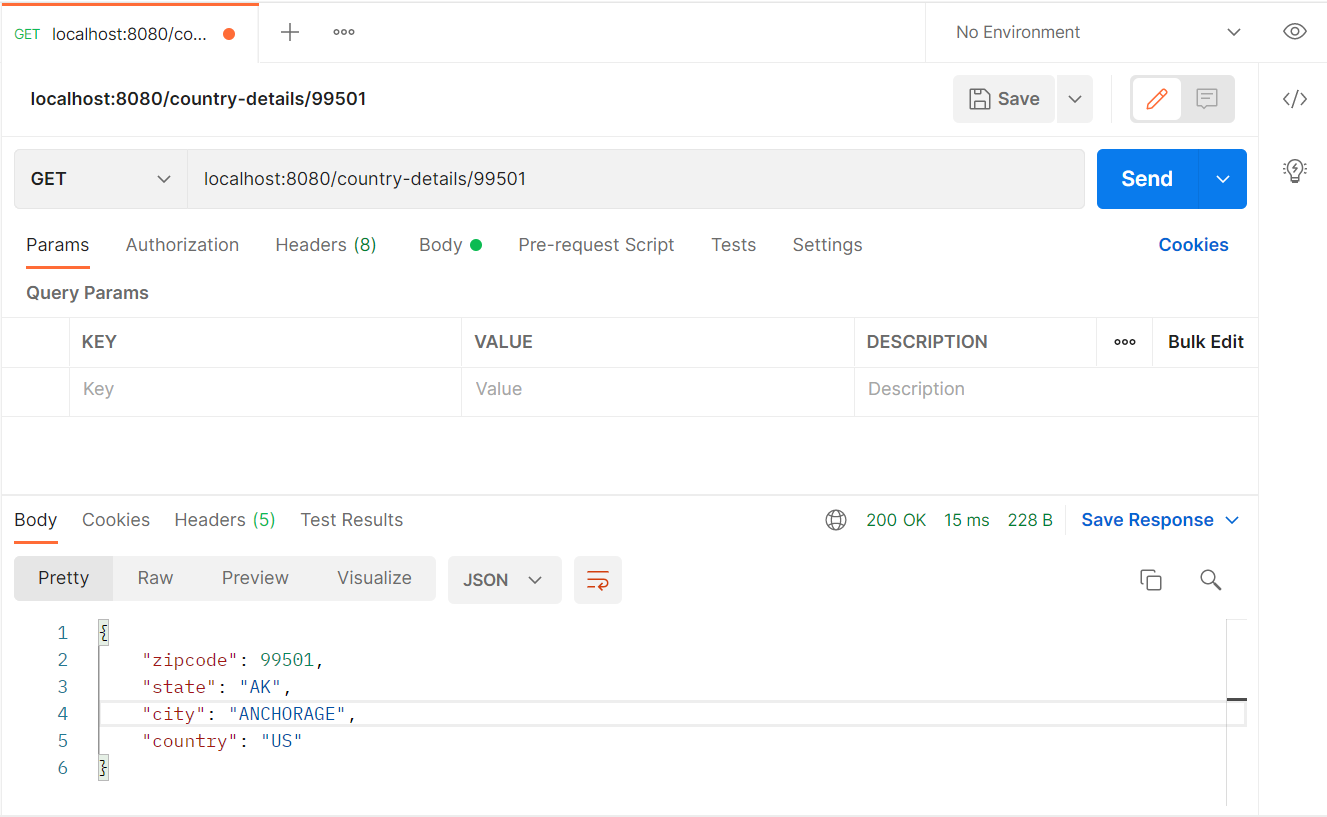
**Service.java:**

****

**Zipcode.java:**

****

**Postman:**

****

4) Create a RESTful web service that validates the credit card. It means we need to check the type of credit card like American Express, Discover, Visa etc and it is valid or not.

Valid length: 16 digits.

First 6 digits must be in one of the following ranges:

601100 through 601109

601120 through 601149

601174

601177 through 601179

601186 through 601199

644000 through 659999

enRoute

Valid length: 15 digits. First four digits must be 2014 or 2149.

JCB

Valid length: 16 to 19 digits.

First 4 digits must be in the range 3528 through 3589.

MasterCard

Valid length: 16 digits.

First digit must be 5 and second digit must be in the range 1 through 5 inclusive. The range is 510000 through 559999.

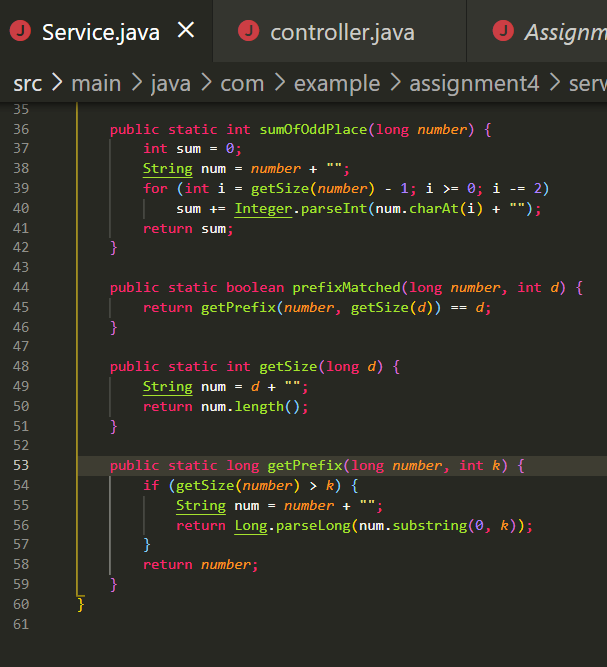
First digit must be 2 and second digit must be in the range 2 through 7 inclusive. The range is 222100 through 272099.

Visa

Valid length: Up to 19 digits. First digit must be a 4.

**Service.java**

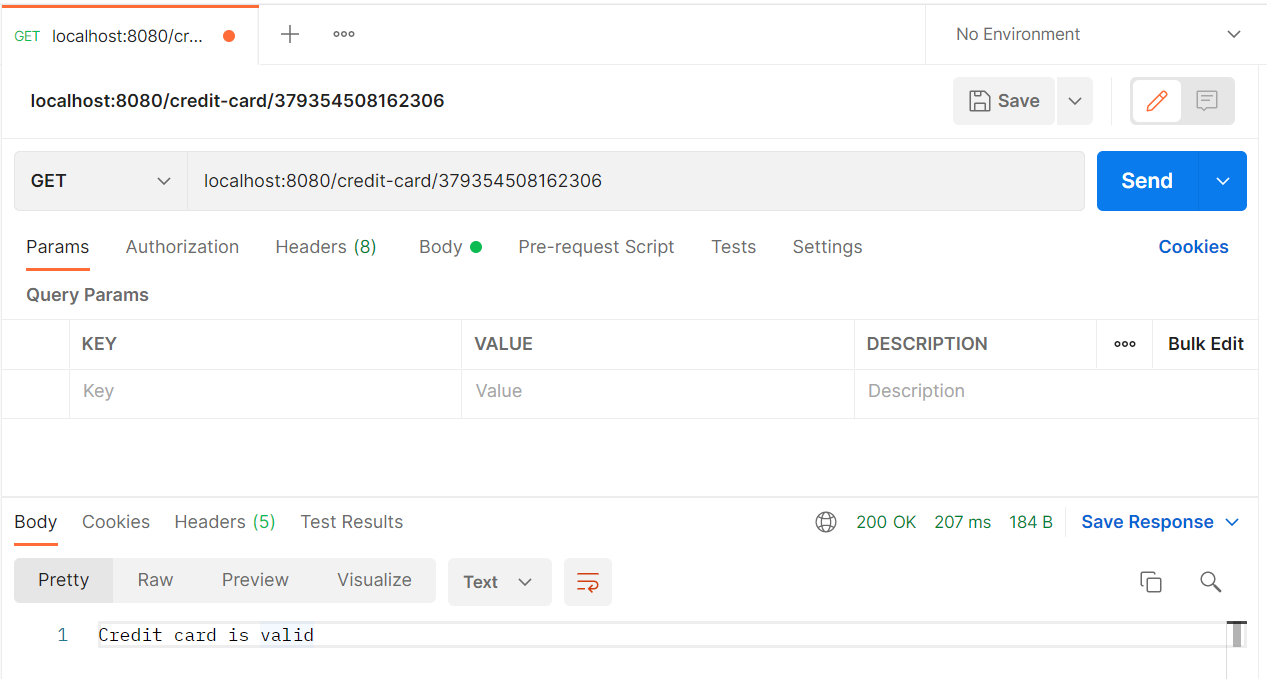


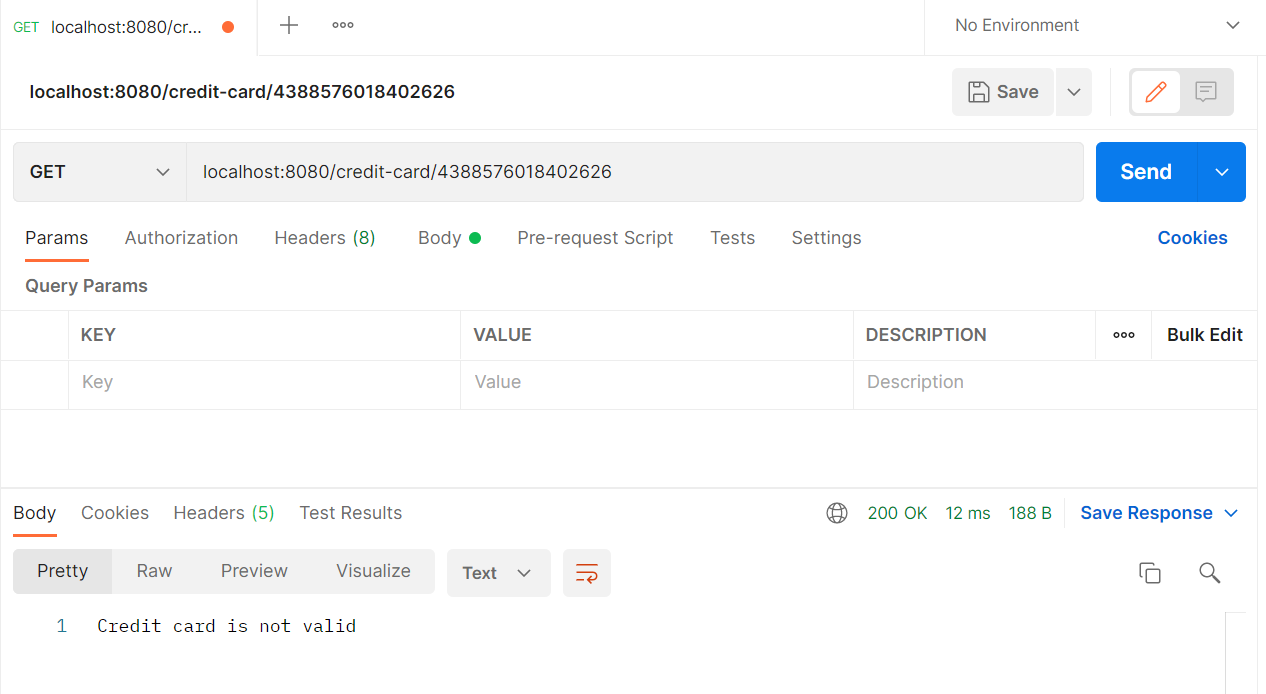


**Controller.java**

****

**POSTMAN:**





5) Develop RESTful web services for "Employee Management System" that manages the information about employees

1. Add a new employee

2. Searching for specific employee

3. Deleting an existing employee

4. Finding all employees

5. Editing/updating employee information.

Create a Employee domain model class having following properties: employeeId, employeeName, employeeDepartment, employeeDesignation, employeeSalary. Employee Id should be generated automatically at database level.

Develop controller, service and repository layers classes.

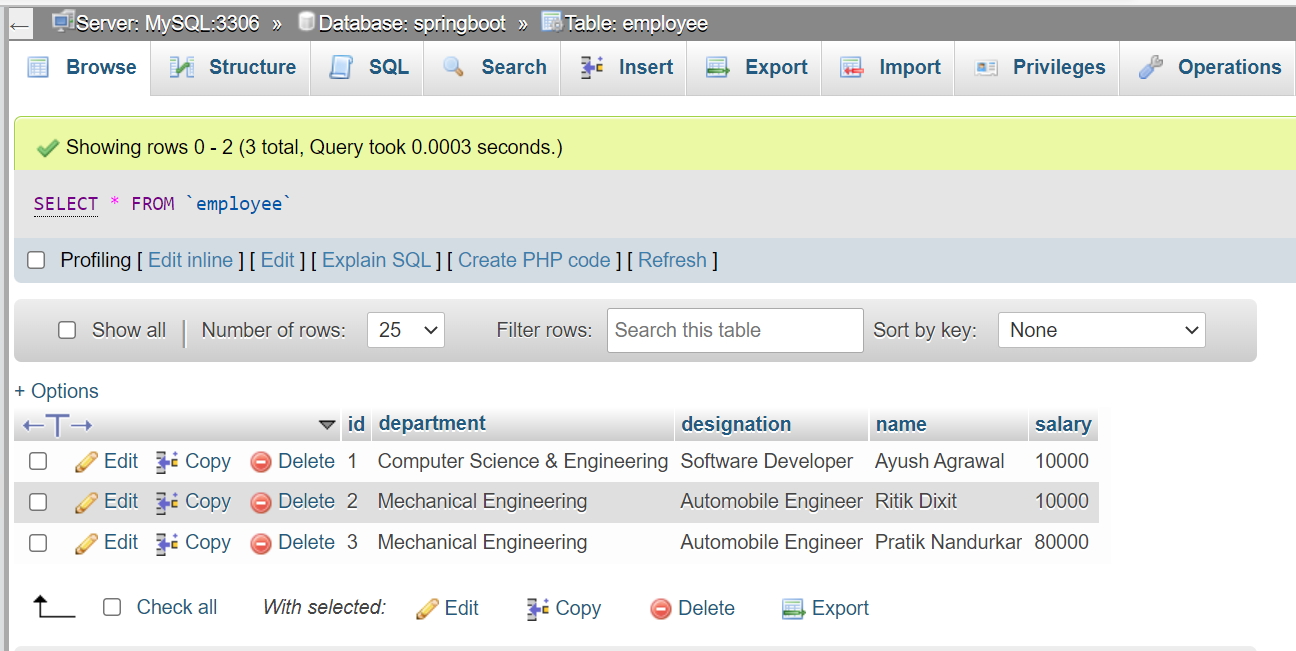
Use CrudRepository from Spring Data.

**Service.java**

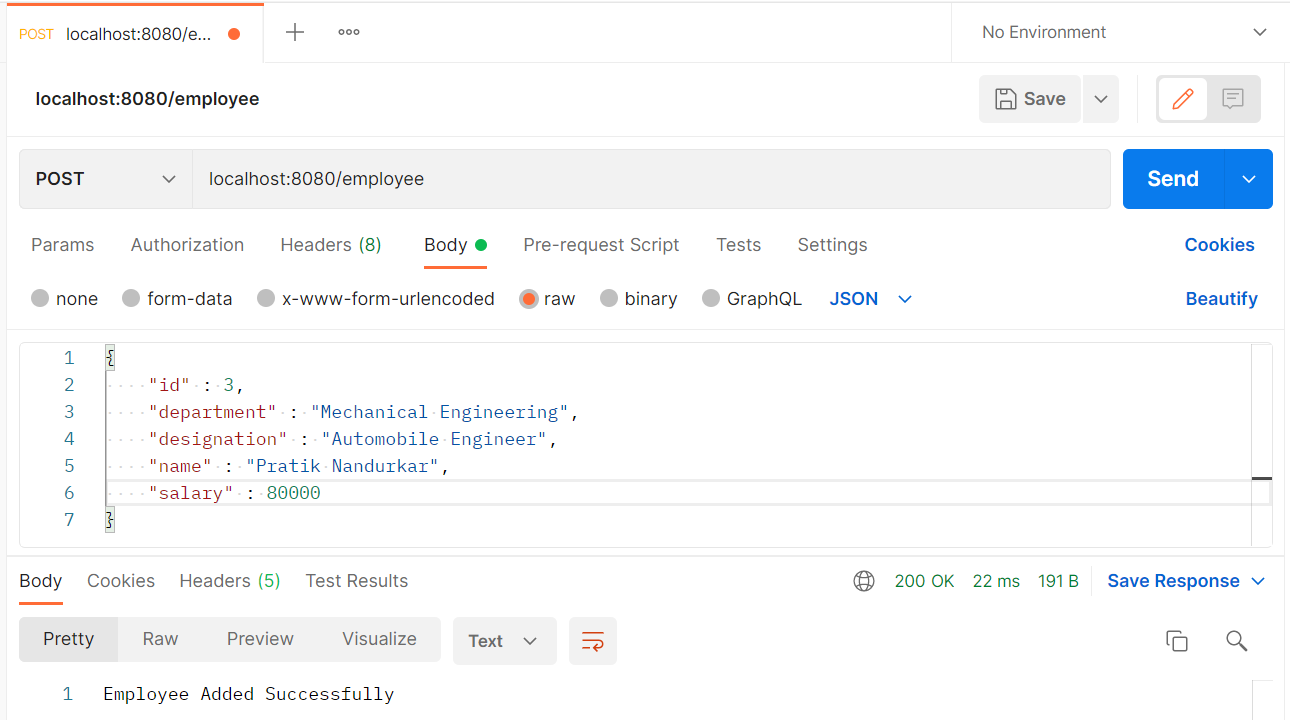
****

****

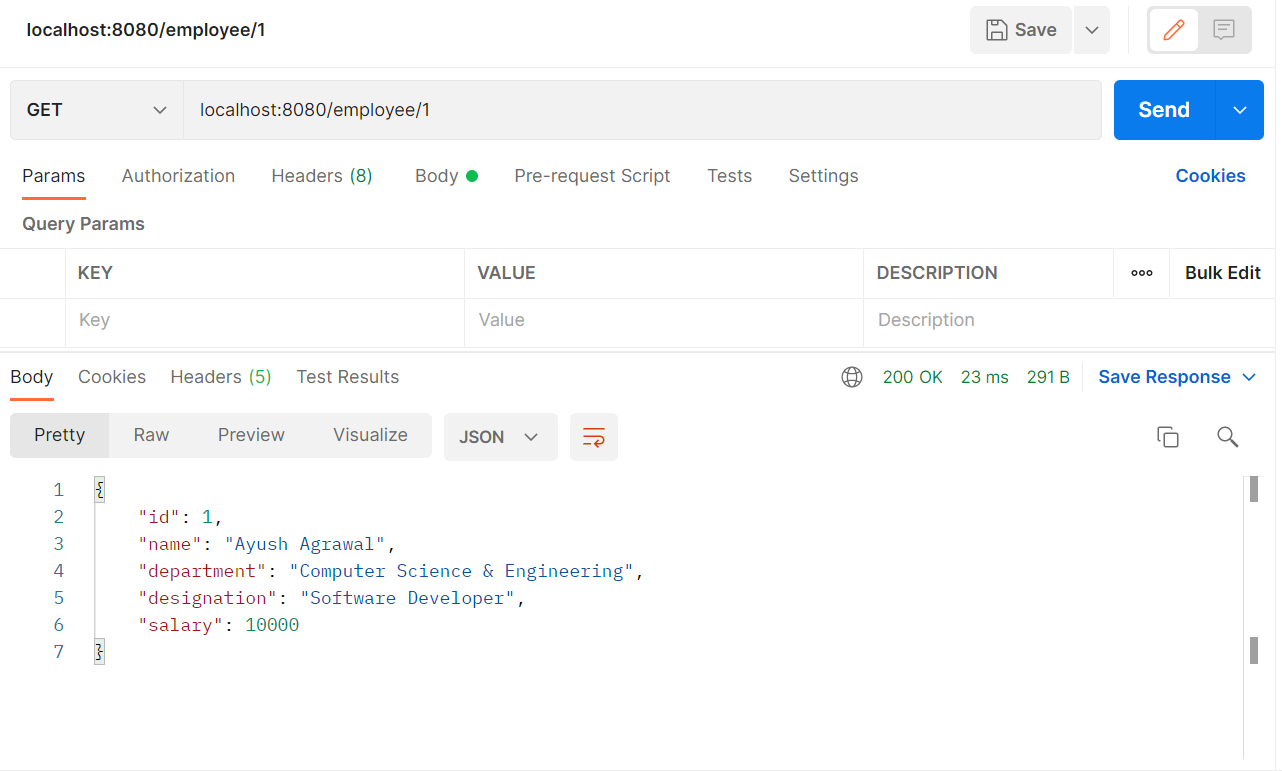
**Database:**



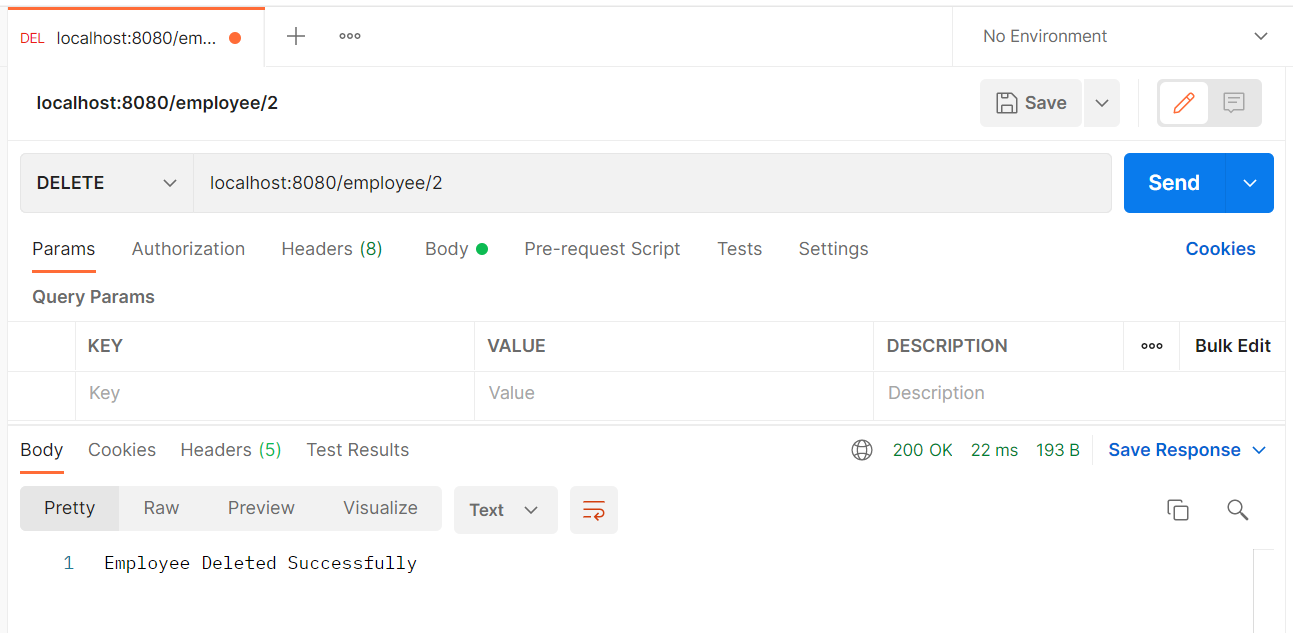
**1) Add a new Employee:**

****

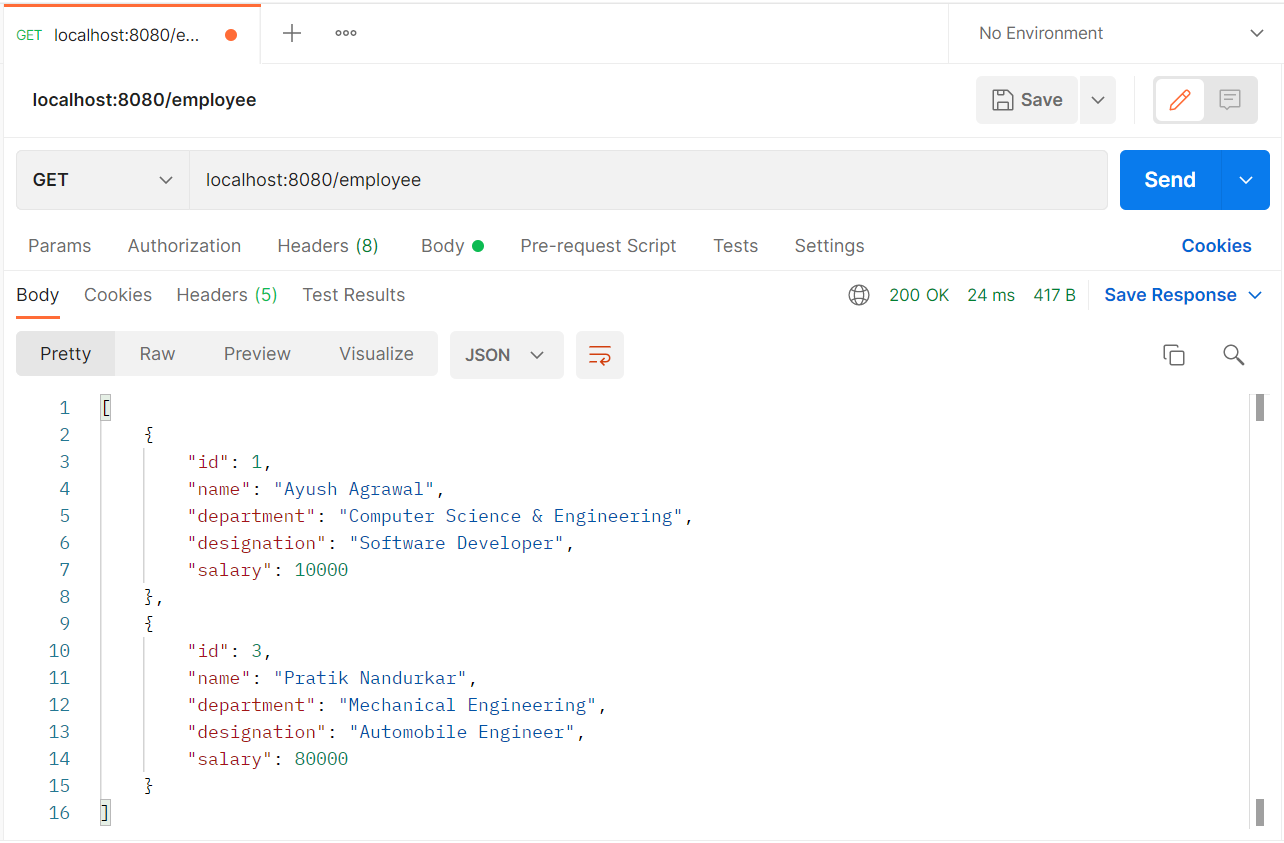
**2) Search for a specific Employee:**

****

**3. Deleting an existing employee**

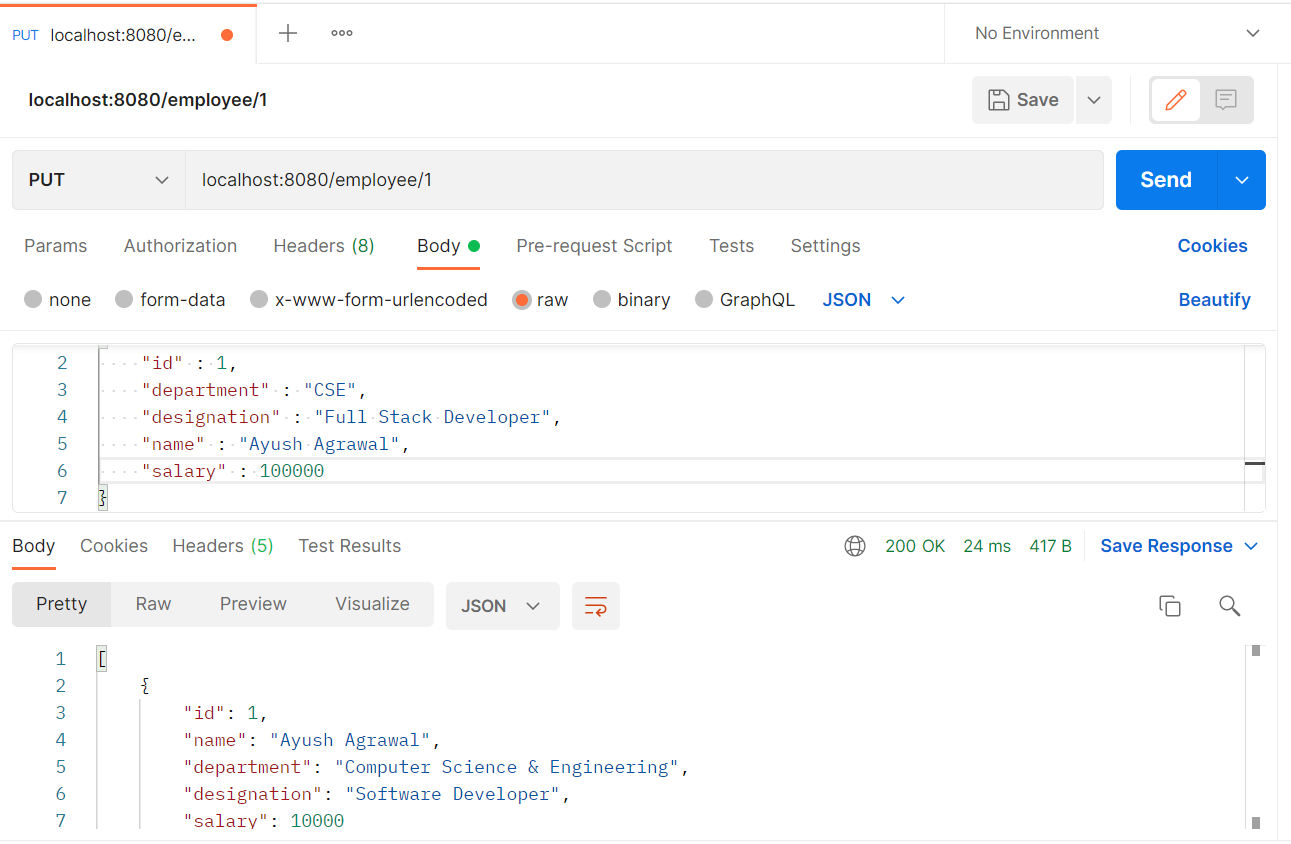
****

**4. Finding all employees**

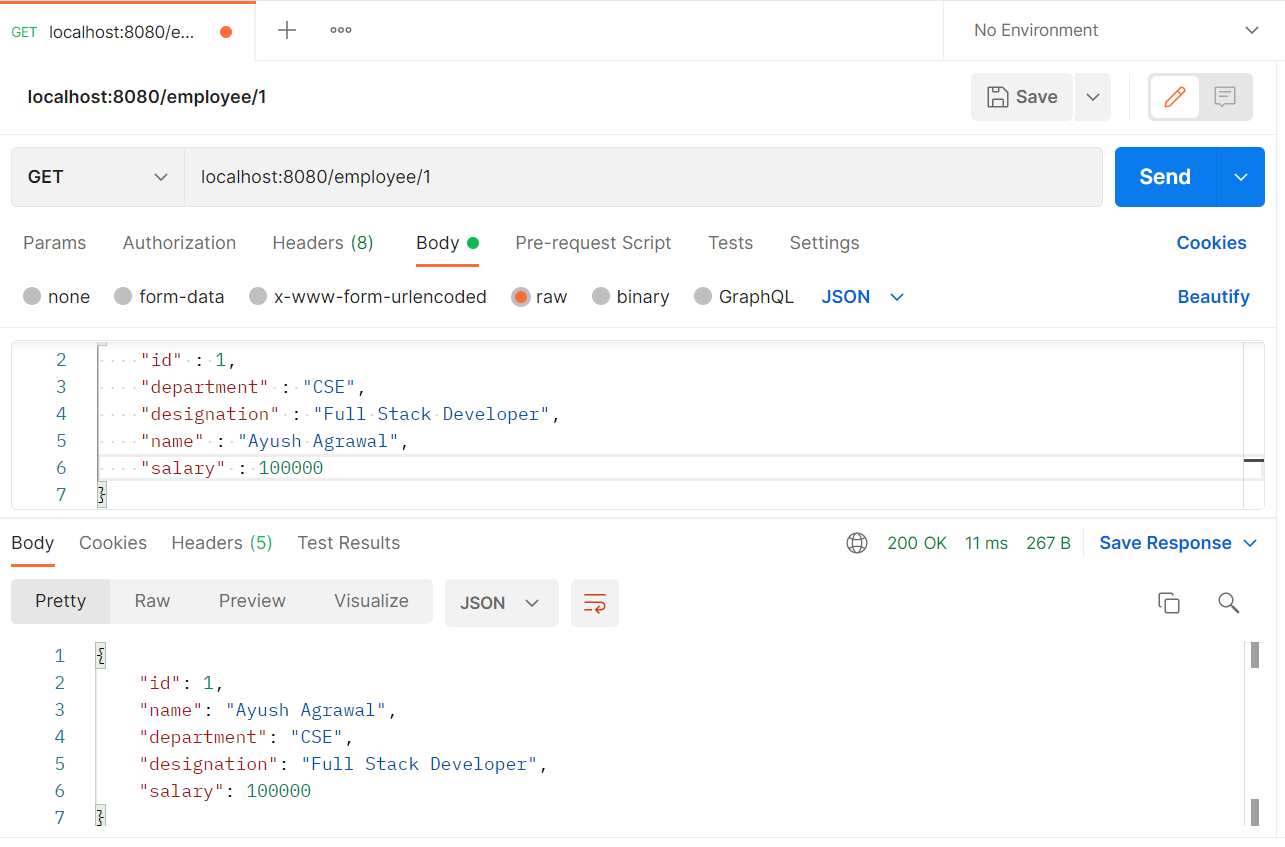
****

**5. Editing/updating employee information.**

**Before Updating**

****

**After Updating**

****

6) Create a Calculator RESTful service that provides following functionality.

1. Addition of the 2 numbers

2. Subtraction of the 2 numbers

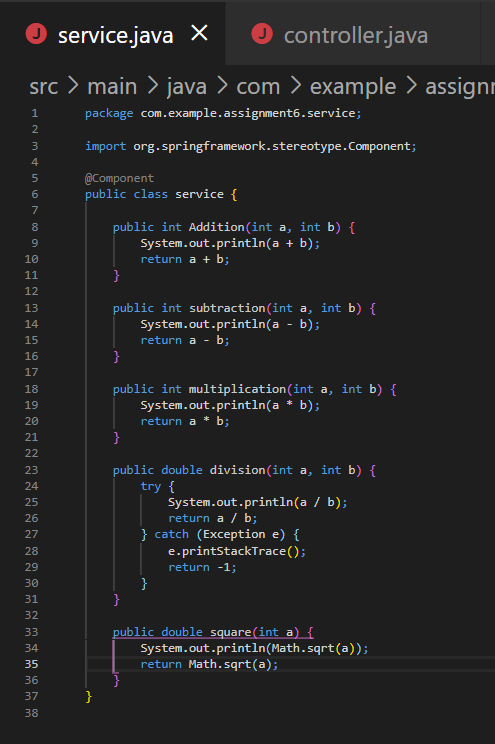
3. Multiplication of 2 numbers

4. Division of 2 numbers

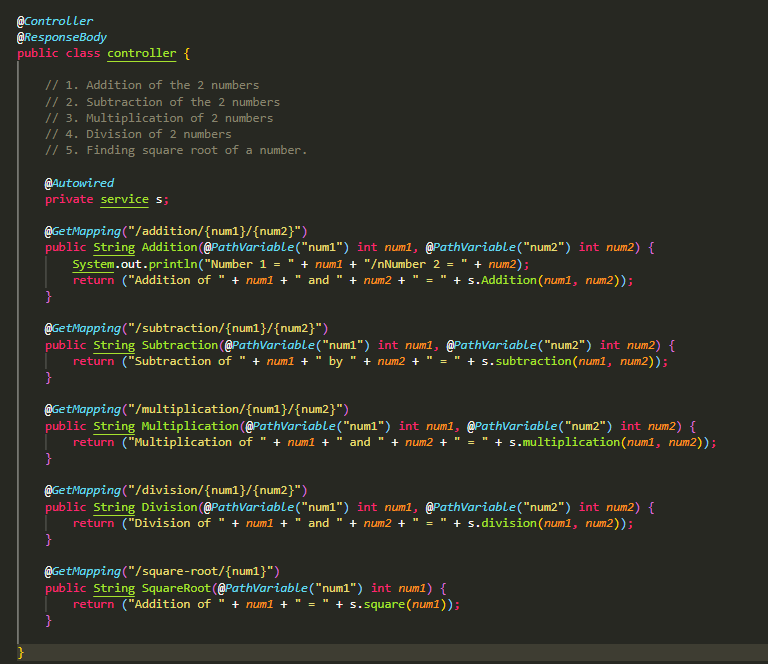
5. Finding square root of a number.

Consume the above RESTful web service by using RestTemplate.

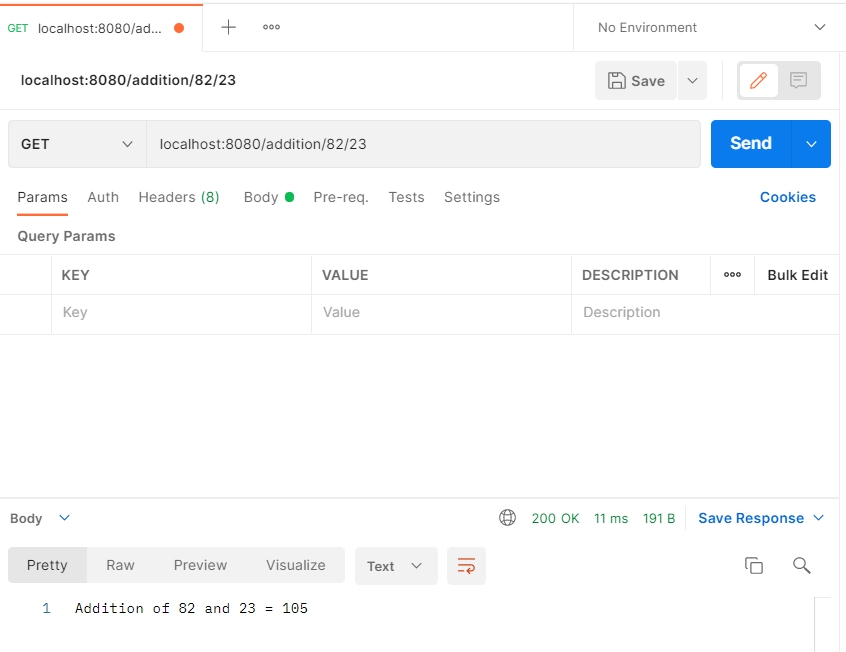
**Service.java**

****

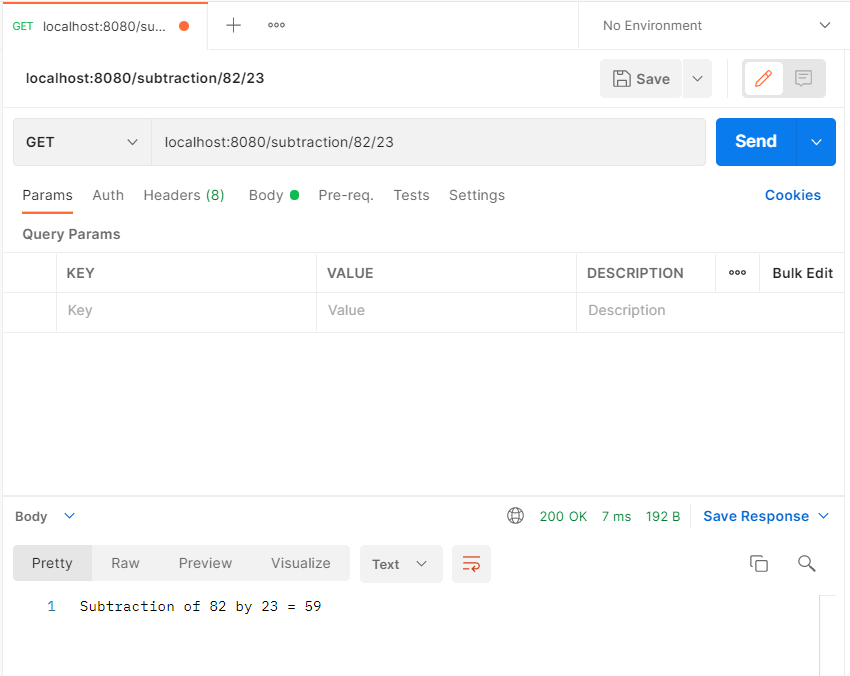
**Controller.java**



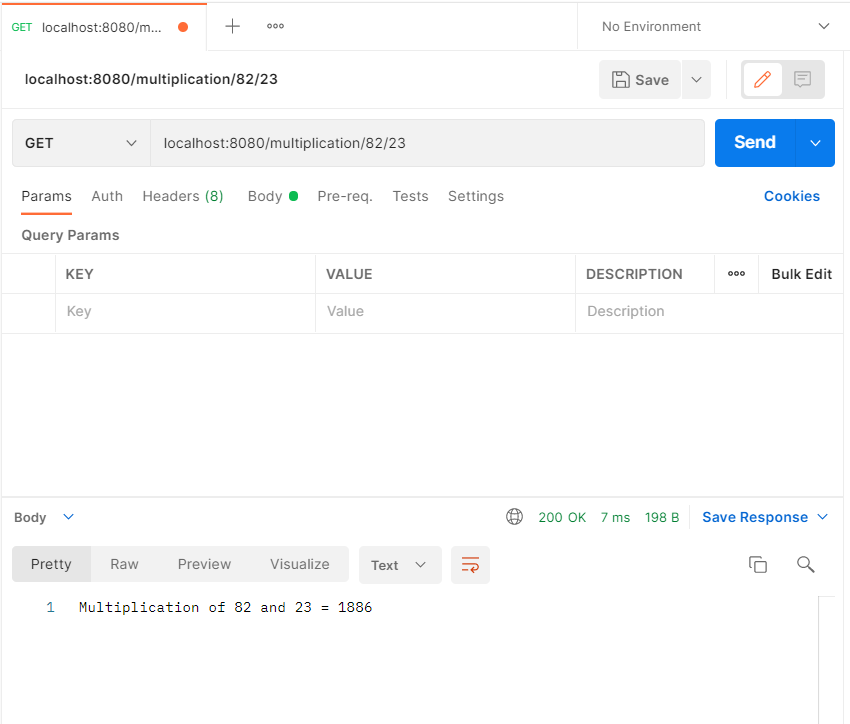
**Postman Addition:**

****

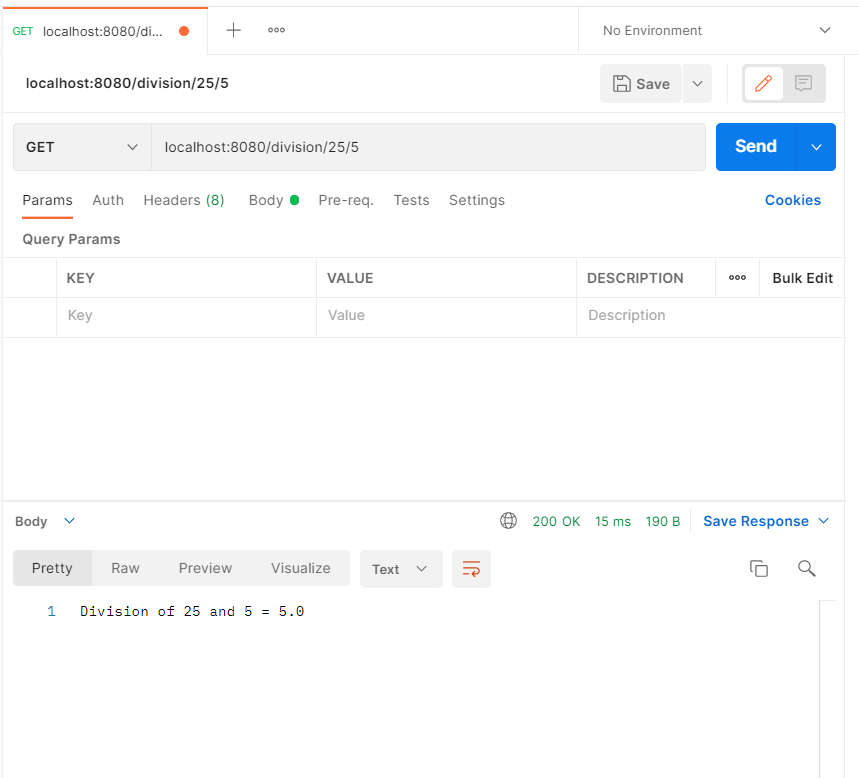
**Postman Subtraction:**

****

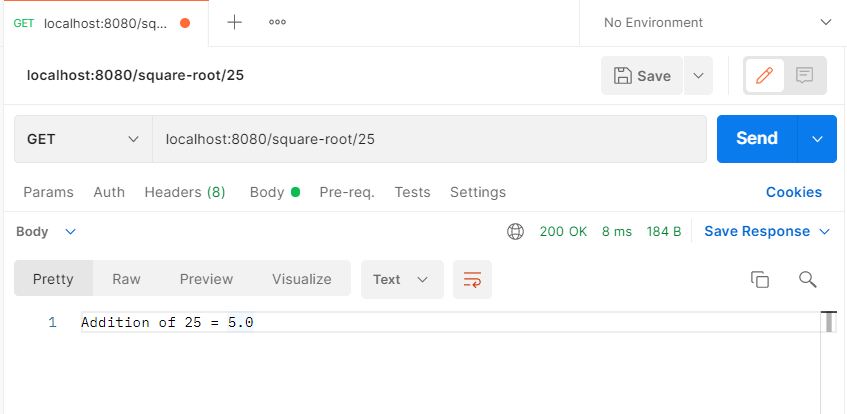
**Postman Multiplication:**

****

**POSTMAN Division:**

****

**POSTMAN Square Root:**

****

7) Design and develop RESTful web service as follows:

1. A user can place an order

2. A user can update an order

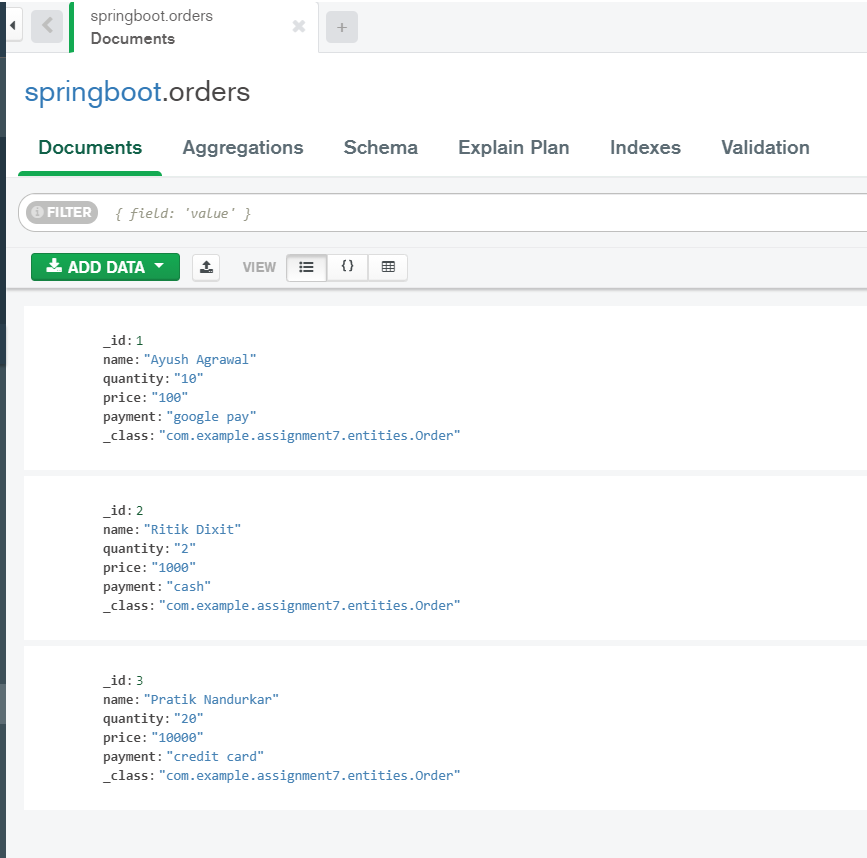
3. A user can view specific order

4. A user can view all the orders

5. A user can delete a specific order.

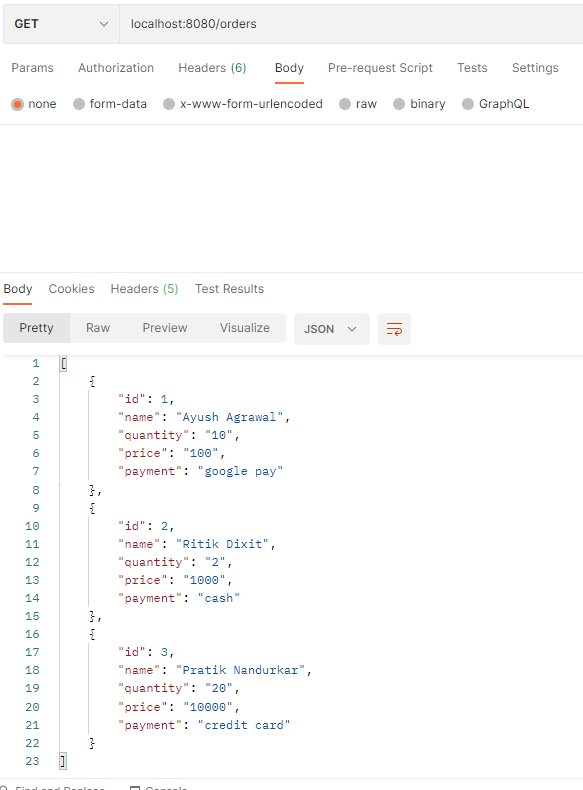
Note: Use MongoRepository of Spring data to store order details

**Mongo Compass:**

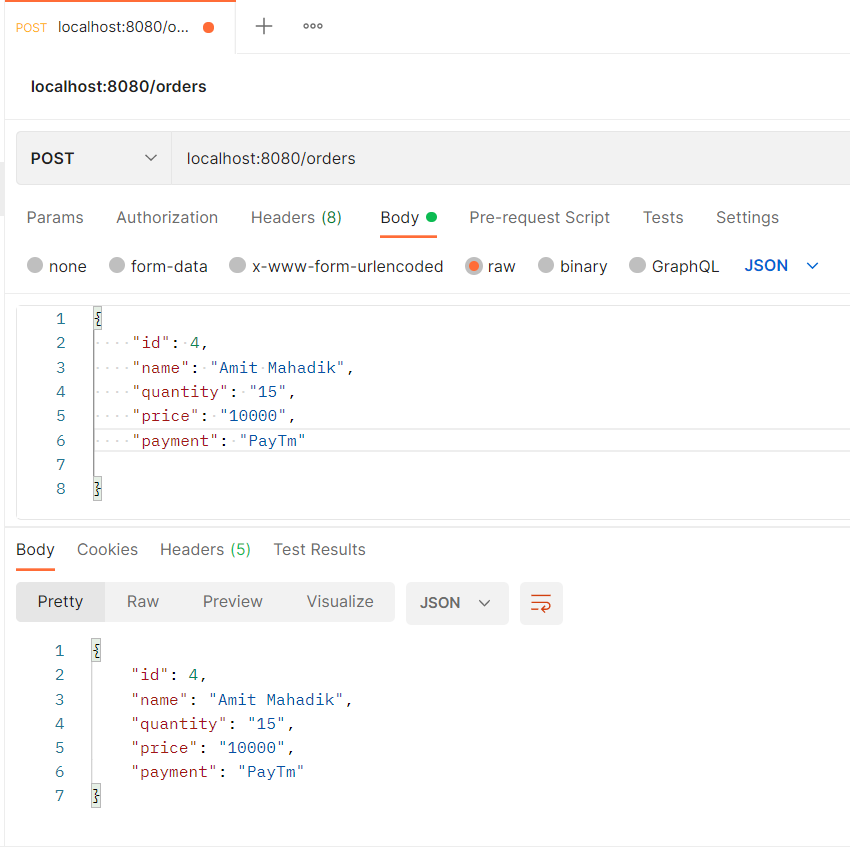
****

**POSTMAN:**

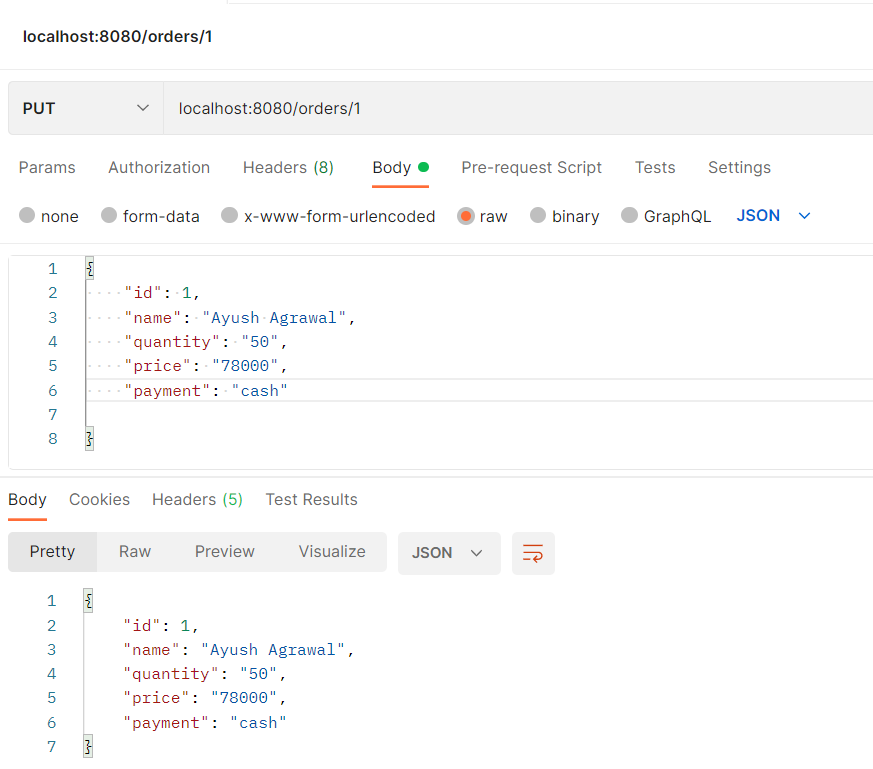
**A user can view all the orders:**

****

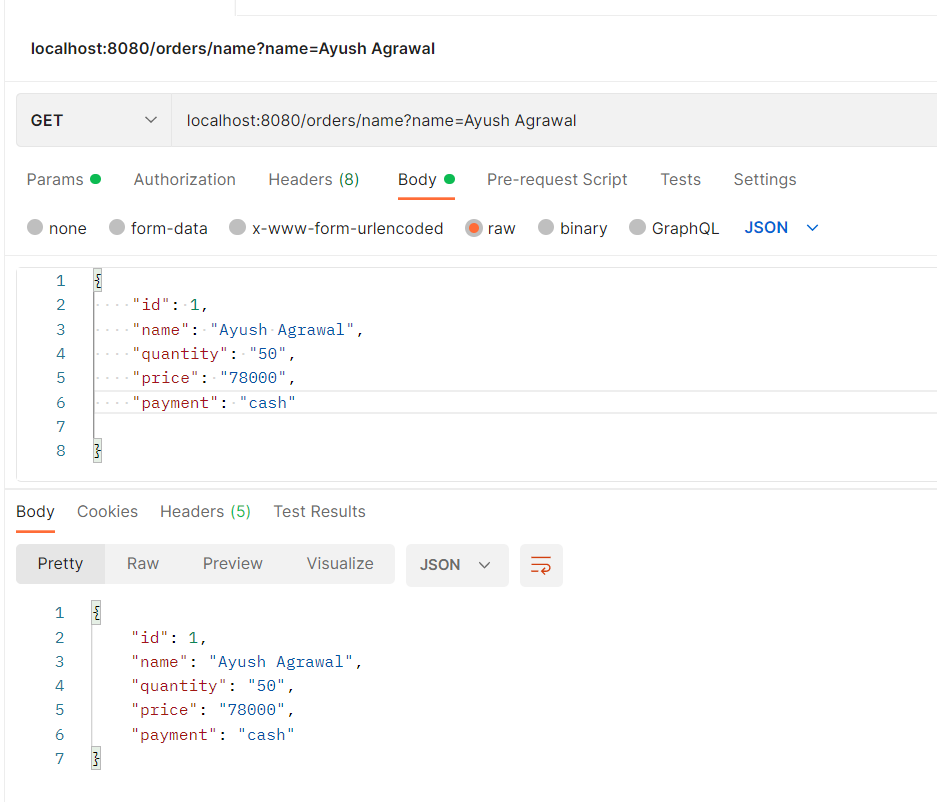
**A user can place an order :**

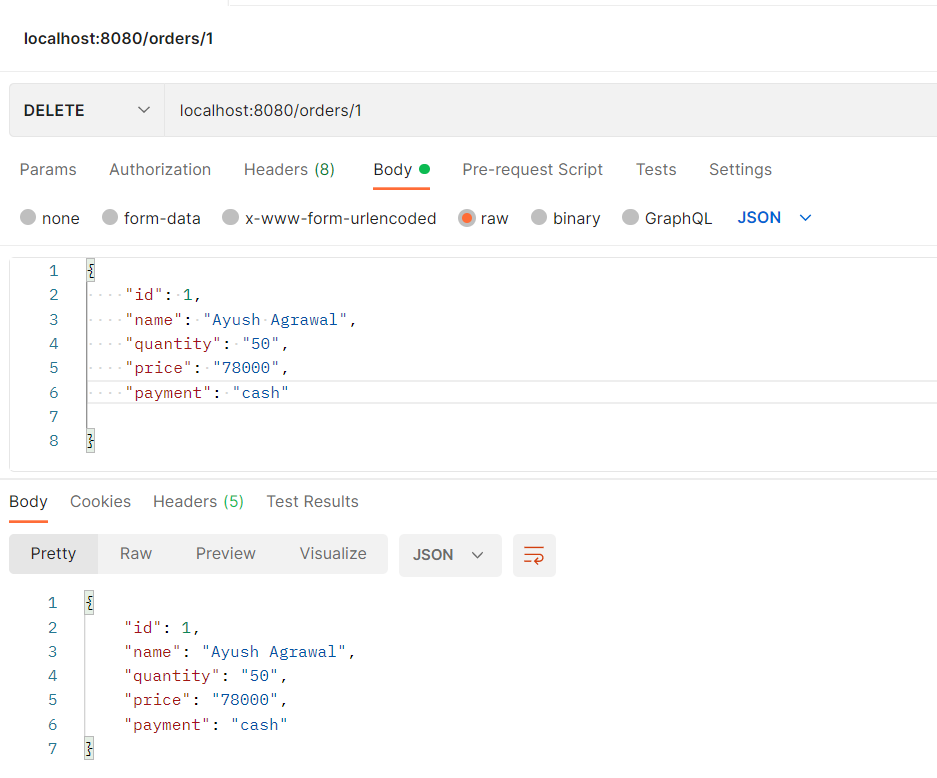
****

**A user can update an order :**

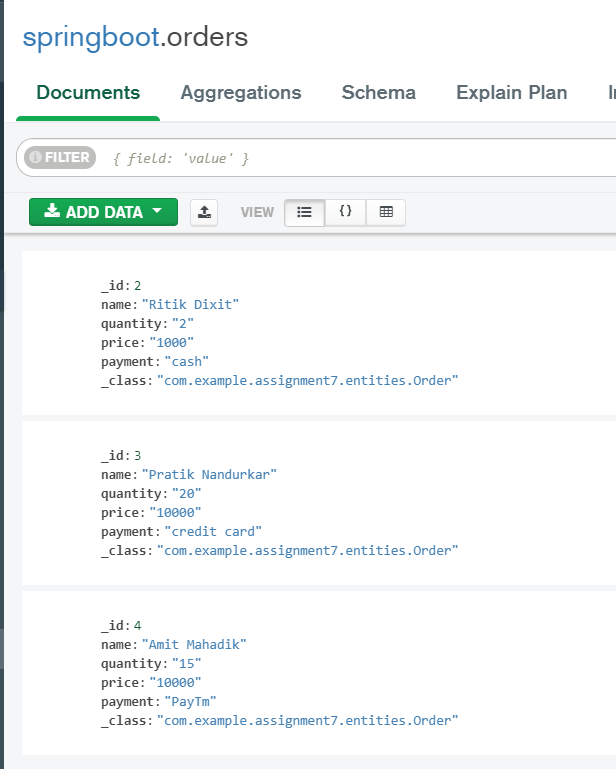
****

**A user can view specific order:**

****

****

**Mongodb Compass after performing above operations:**

****

8) Design and develop RESTful web service as follows:

1. An admin can add a new product.

2. An admin can update details of existing product.

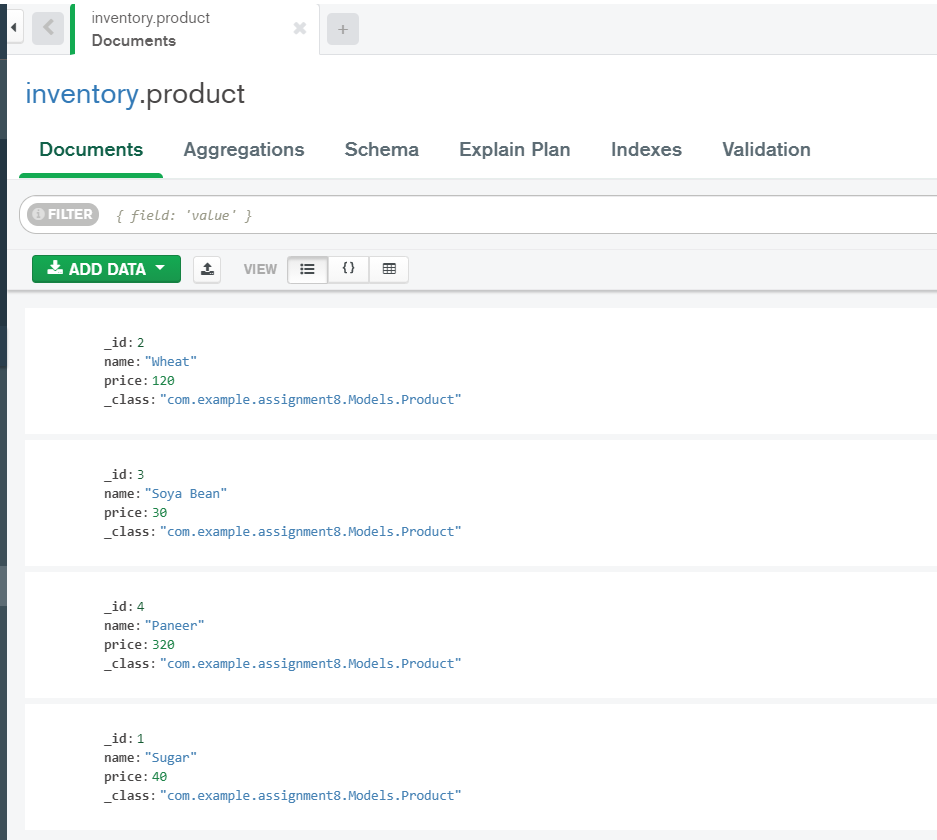
3. An admin can delete existing product

4. An admin can view specific product

5. An admin can view all the products

Note: Use MongoRepository of Spring data to store product details.

**MongoDB Database:**



9) Design and develop RESTful web service as follows:

1. Add a new customer information.

2. Update customer information.

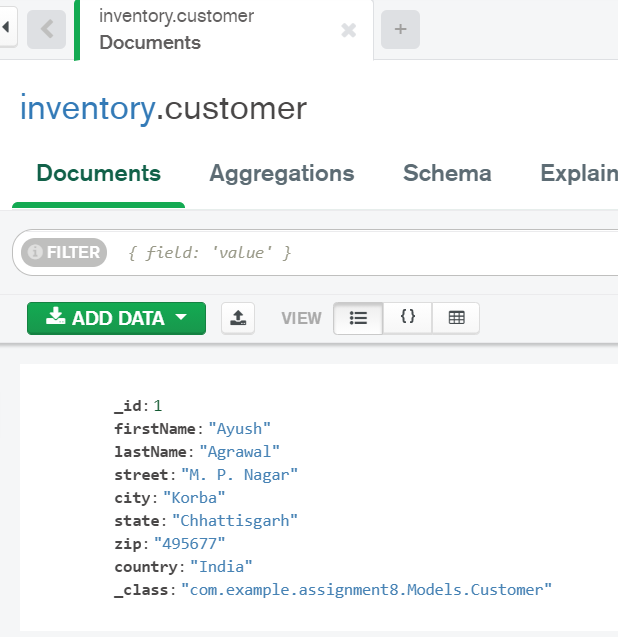
3. Delete existing customer information.

4. Fetch information of specific customer.

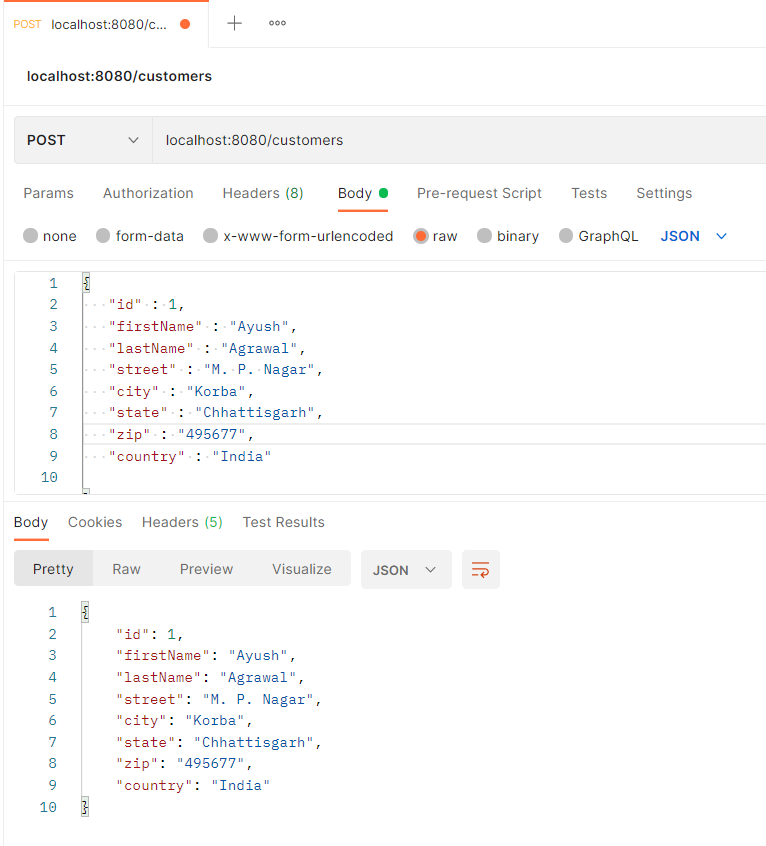
5. Fetch information of all customers.

Note: Use CrudRepository of Spring Data to store customer details.

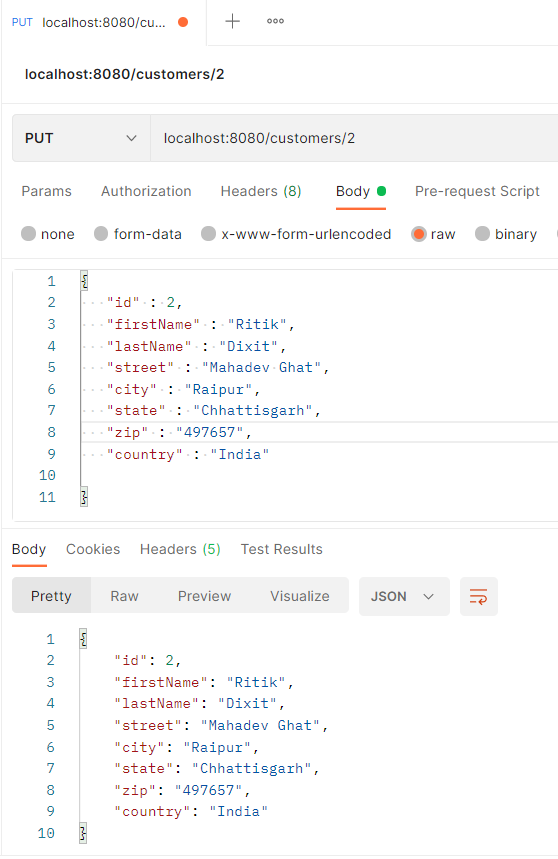
**MongoDB Database:**

****

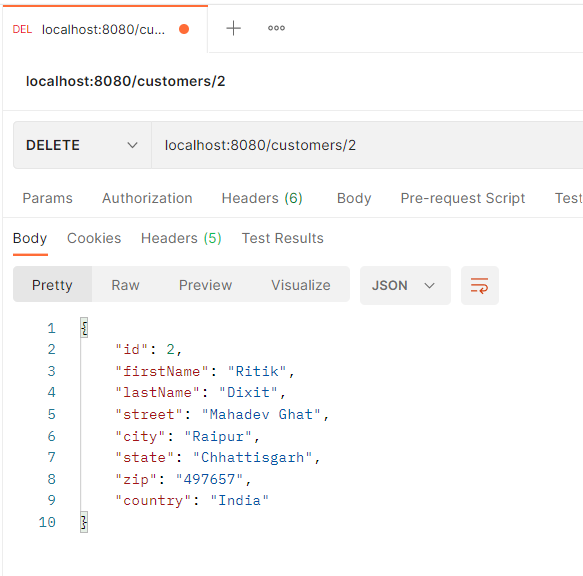
**1. Add a new customer information.**

****

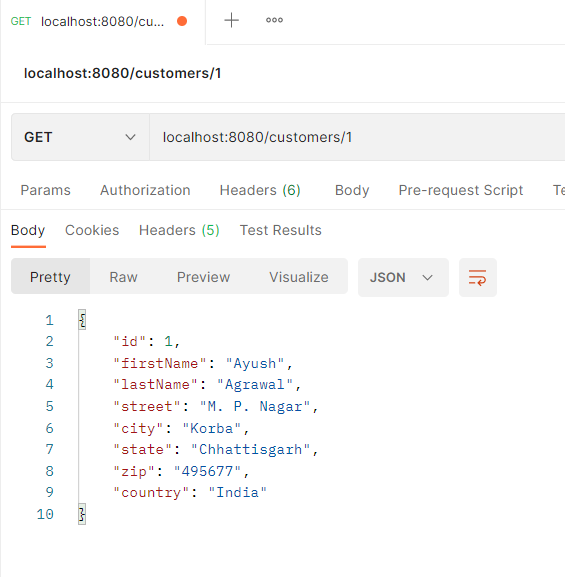
**2. Update customer information.**

****

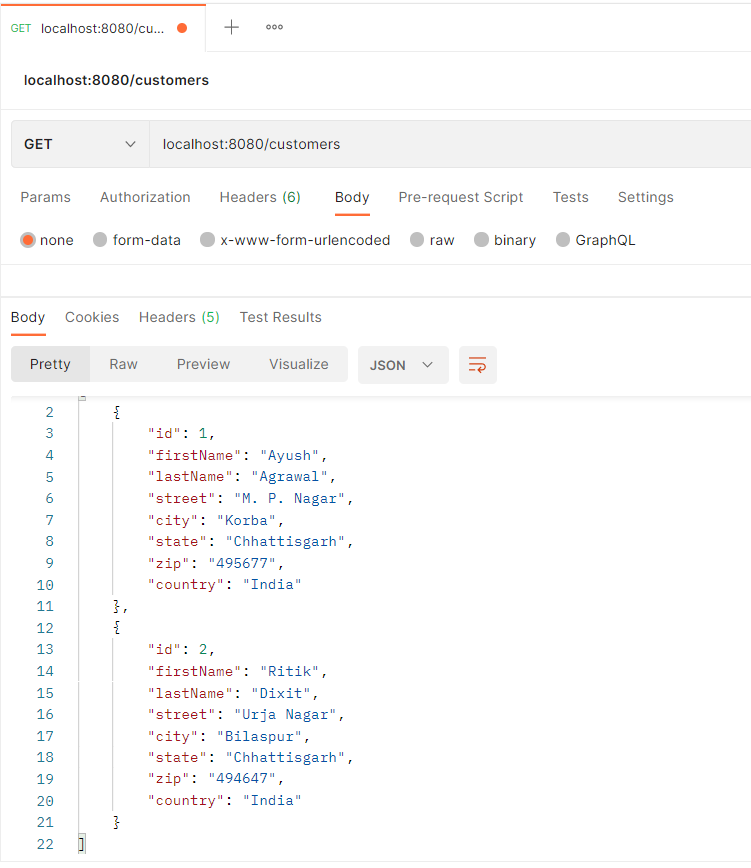
**3. Delete existing customer information.**

****

**4. Fetch information of specific customer.**

****

**5. Fetch information of all customers.**

****