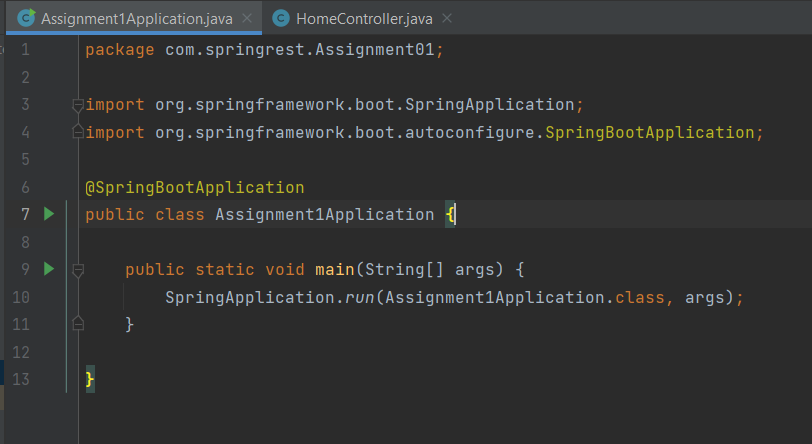
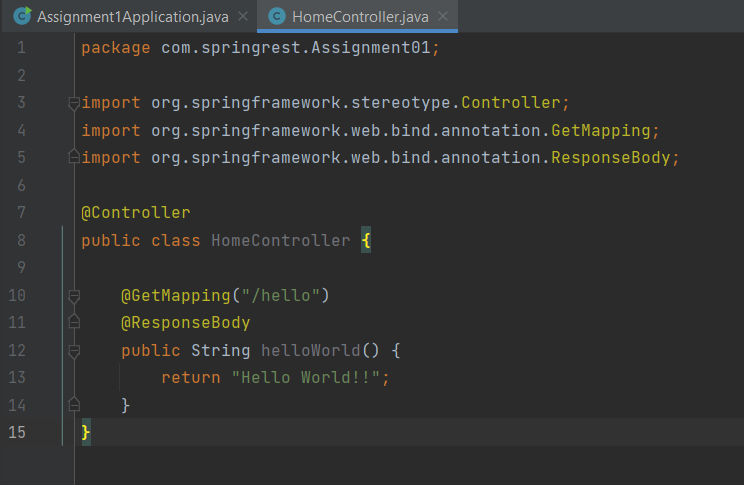
**Spring REST Assignments**

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

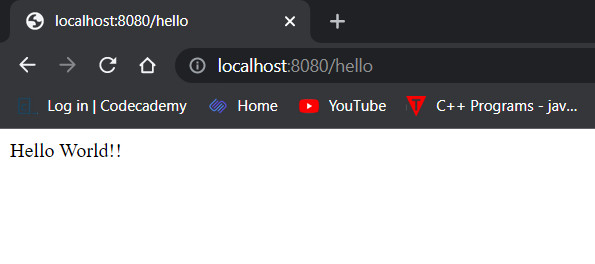
**Assignment1Application.java**



**HomeController.java**



Output :-



2) Create a RESTful web service that authenticates an 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.

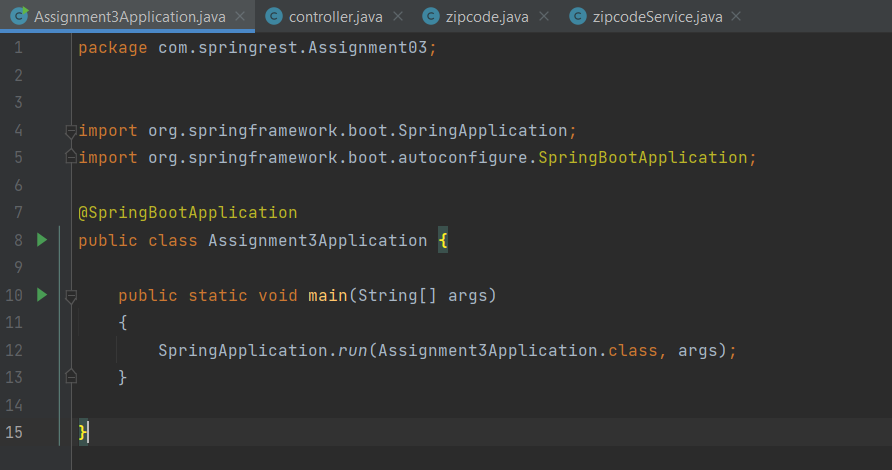
3) Create a RESTful web service that returns state, city and country information when user passes zipcode.

You can send state, city and country information in JSON format.

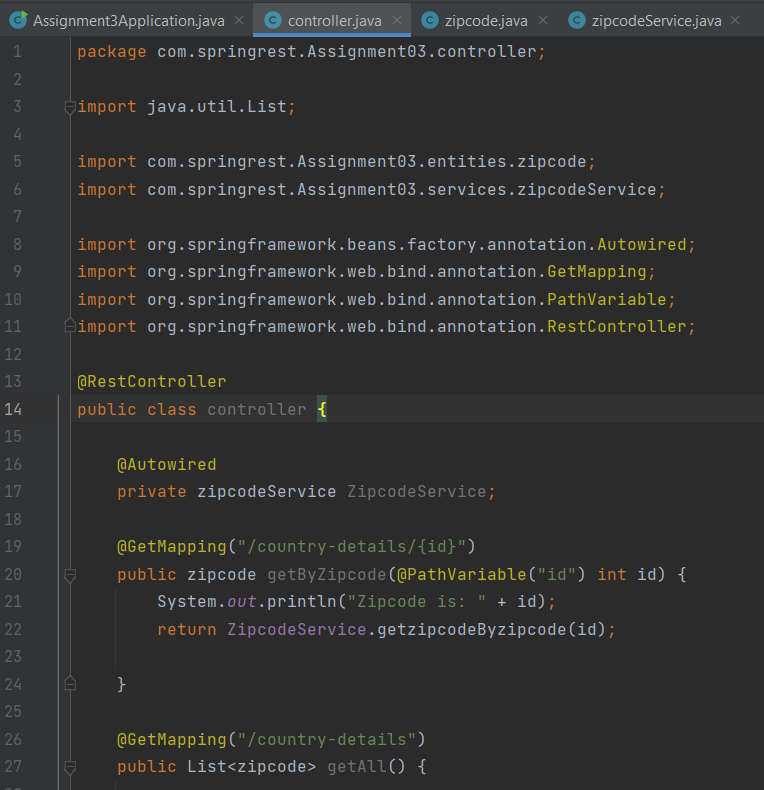
Sample Input: 99501

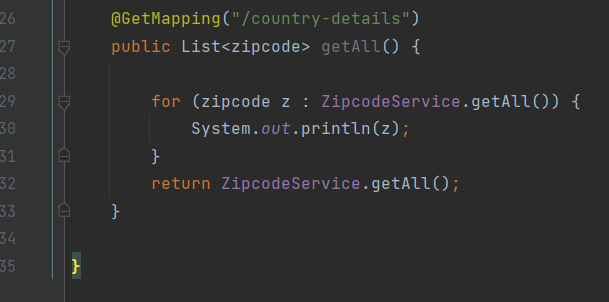
Sample output: {"state": "AK", City: "ANCHORAGE", "country: "US"}

**Assignment3Application.java**

****

**controller.java**

****

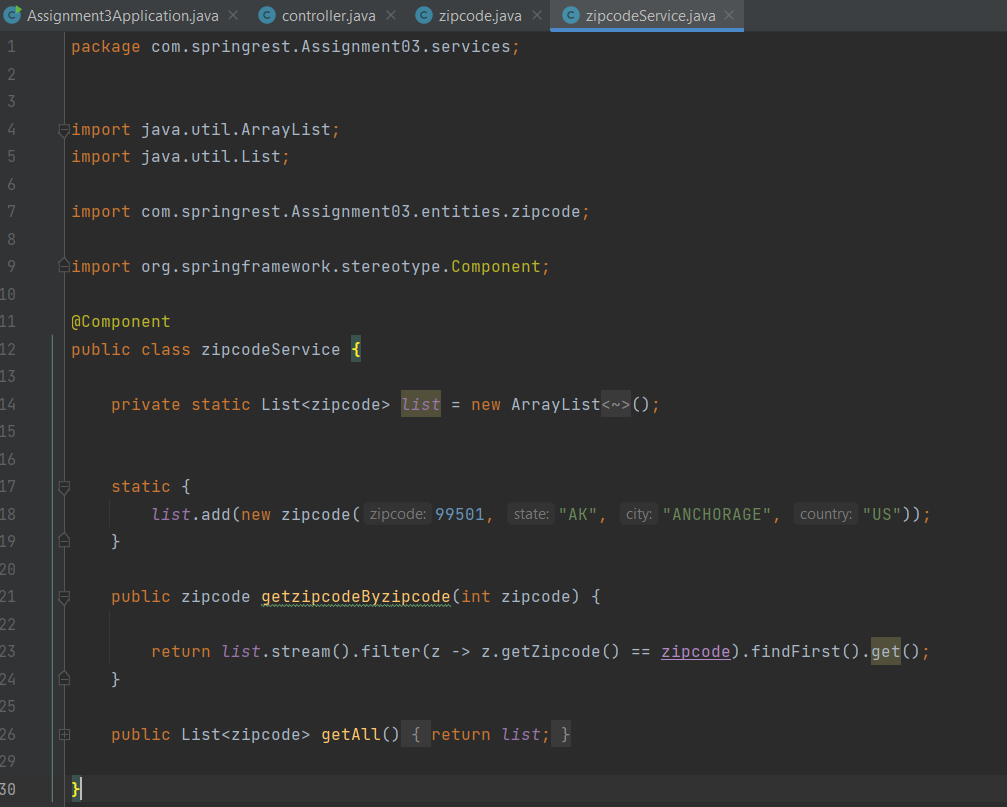
****

**zipcode.java**

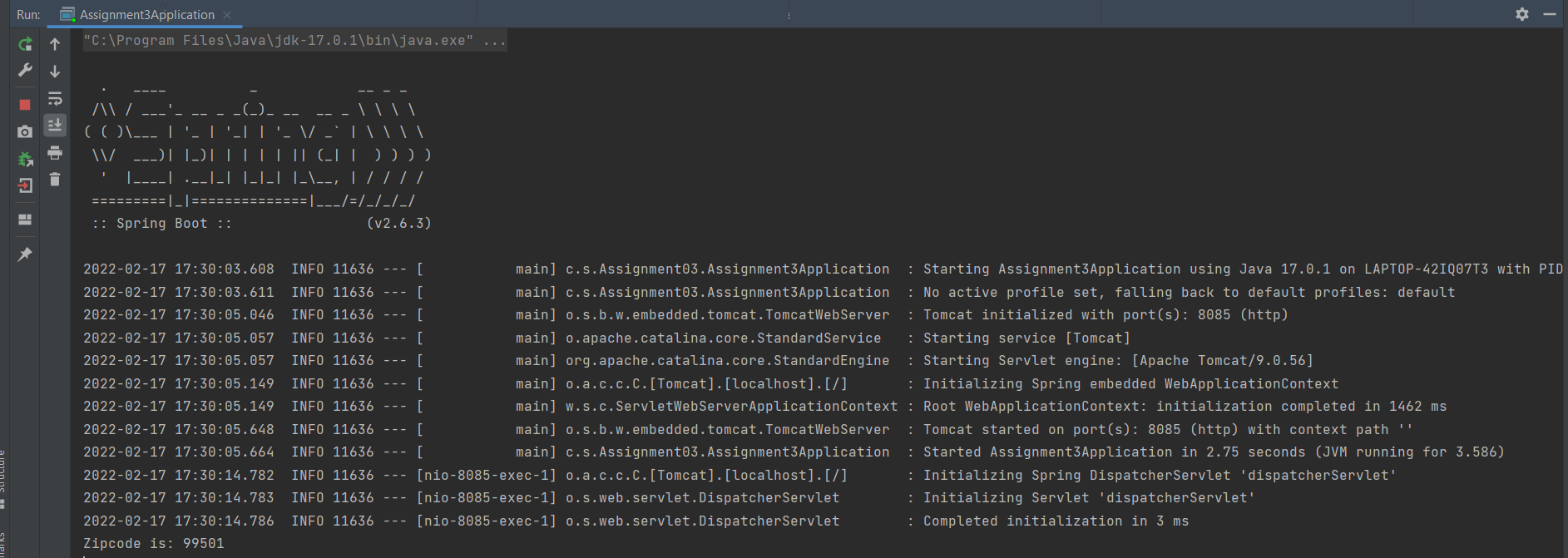
package com.springrest.Assignment03.entities;  
  
public class zipcode {  
 private int zipcode;  
 private String state;  
 private String city;  
 private String country;  
  
 public zipcode() {  
 super();  
 }  
  
 public zipcode(int zipcode, String state, String city, String country) {  
 this.zipcode = zipcode;  
 this.state = state;  
 this.city = city;  
 this.country = country;  
 }  
  
 public int getZipcode() {  
 return zipcode;  
 }  
  
 public void setZipcode(int zipcode) {  
 this.zipcode = zipcode;  
 }  
  
 public String getState() {  
 return state;  
 }  
  
 public void setState(String state) {  
 this.state = state;

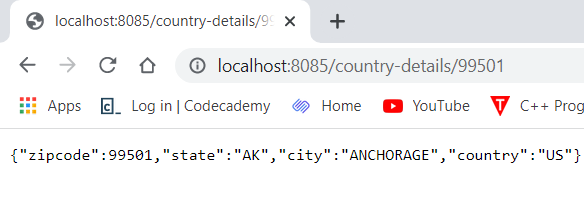
}  
  
 public String getCity() {  
 return city;  
 }  
  
 public void setCity(String city) {  
 this.city = city;  
 }  
  
 public String getCountry() {  
 return country;  
 }  
  
 public void setCountry(String country) {  
 this.country = country;  
 }  
  
 @Override  
 public String toString() {  
 return "zipcode [city=" + city + ", country=" + country + ", state=" + state + ", zipcode=" + zipcode + "]";  
 }

**zipcodeService.java**

****

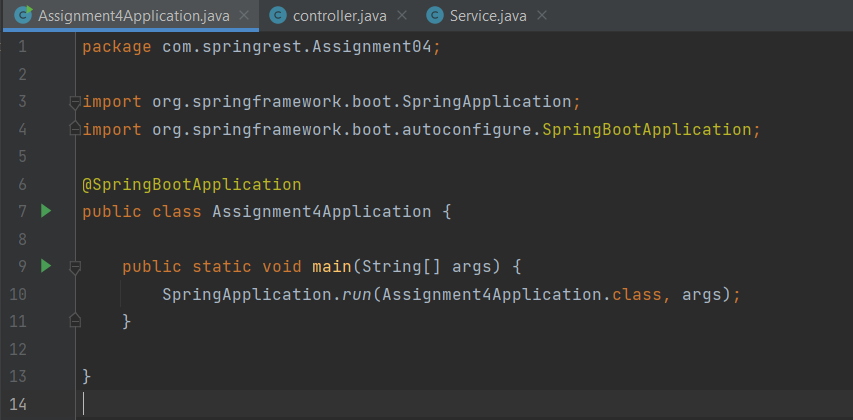
**OUTPUT :-**

****

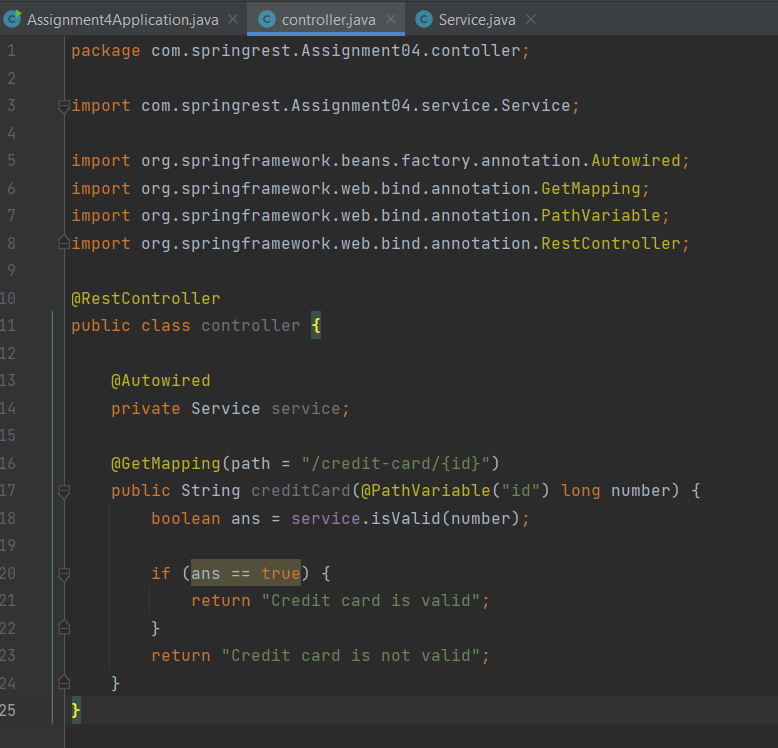
****

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,

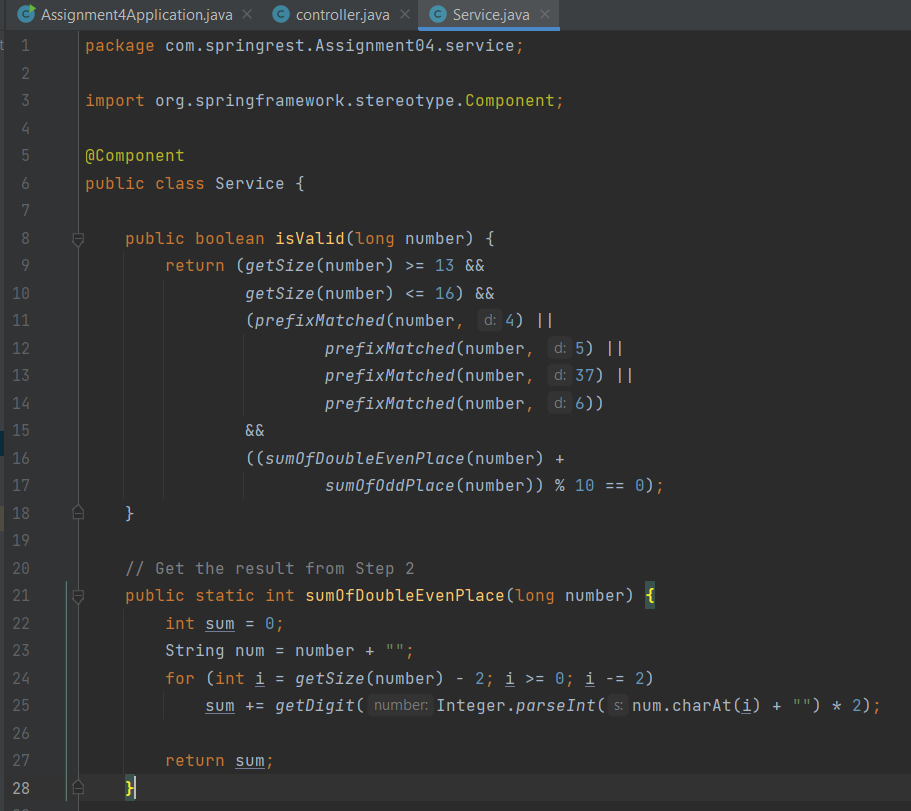
**Assignment4Application.java**

****

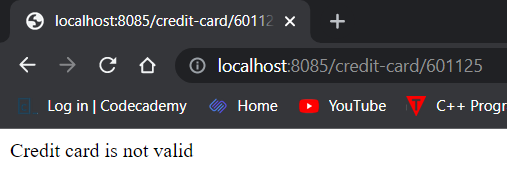
**controller.java**

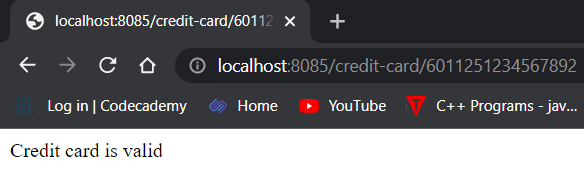
****

**Service.java**

****

**OUTPUT**

****

****

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

1. Addition of the 2 numbers

2. Subtraction of the 2 numbers

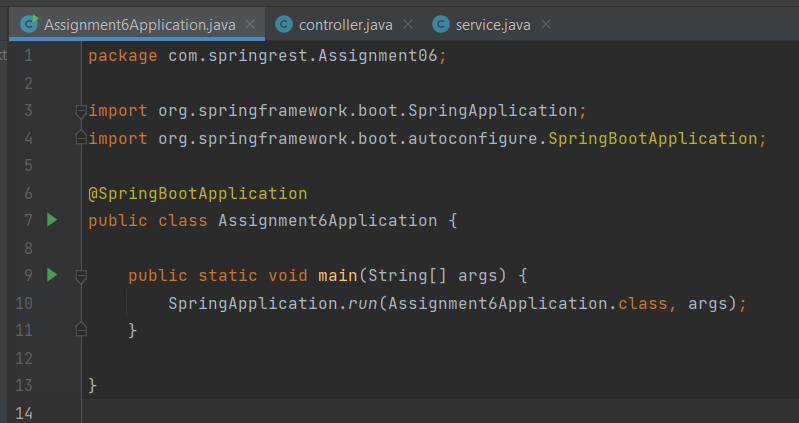
3. Multiplication of 2 numbers

4. Division of 2 numbers

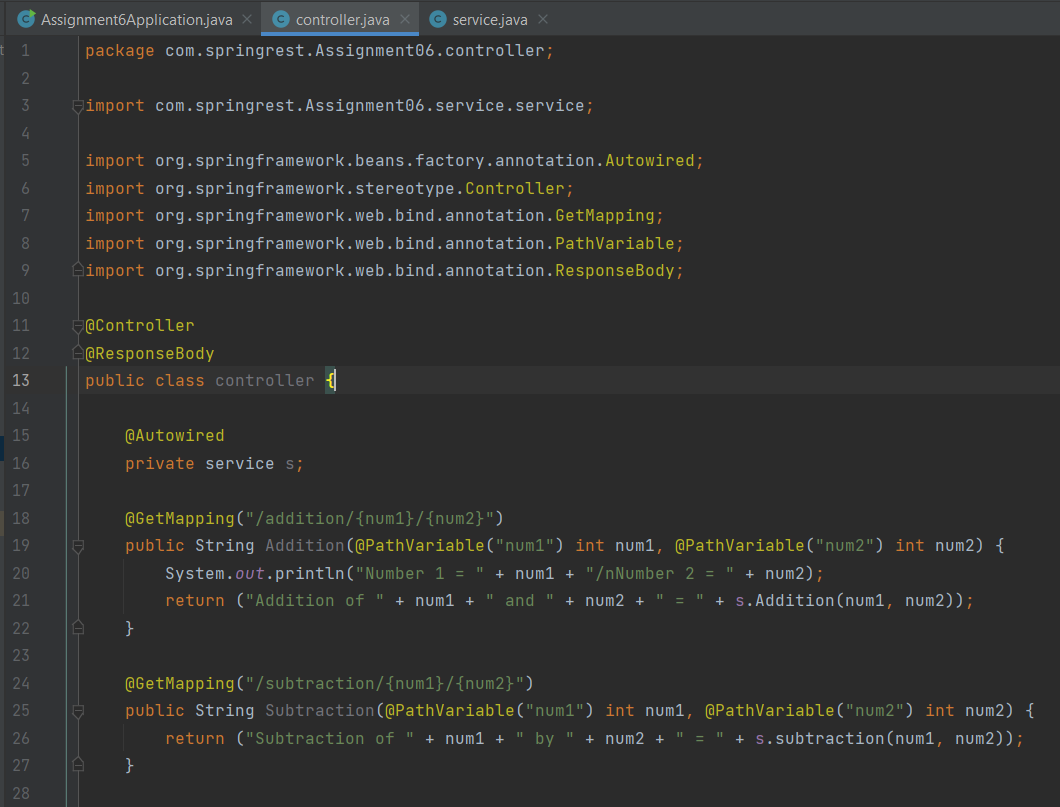
5. Finding square root of a number.

Consume the above RESTful web service by using RestTemplate.

**Assignment6Application.java**

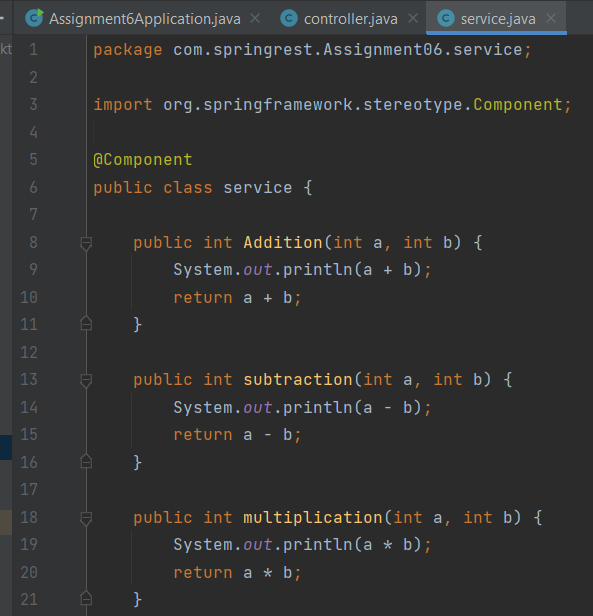
****

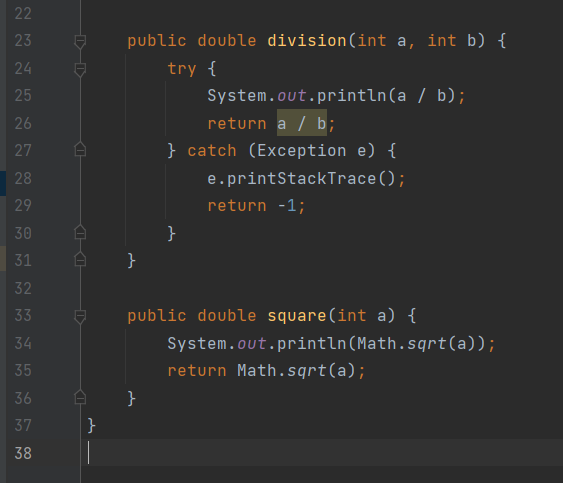
**controller.java**

****

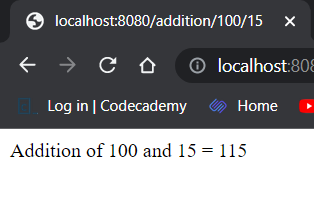
****

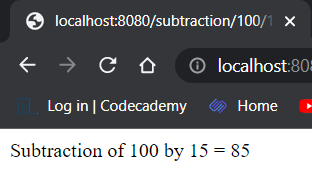
**service.java**

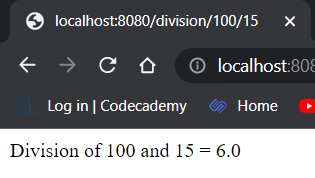
****

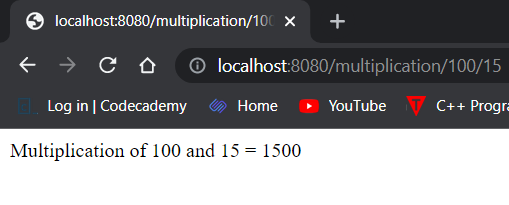
****

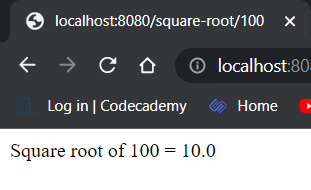
**OUTPUT –**

****

****

****

****

****

7) Design and develop RESTful web service as follows:

1. A user can place an order

2. A user can update an order

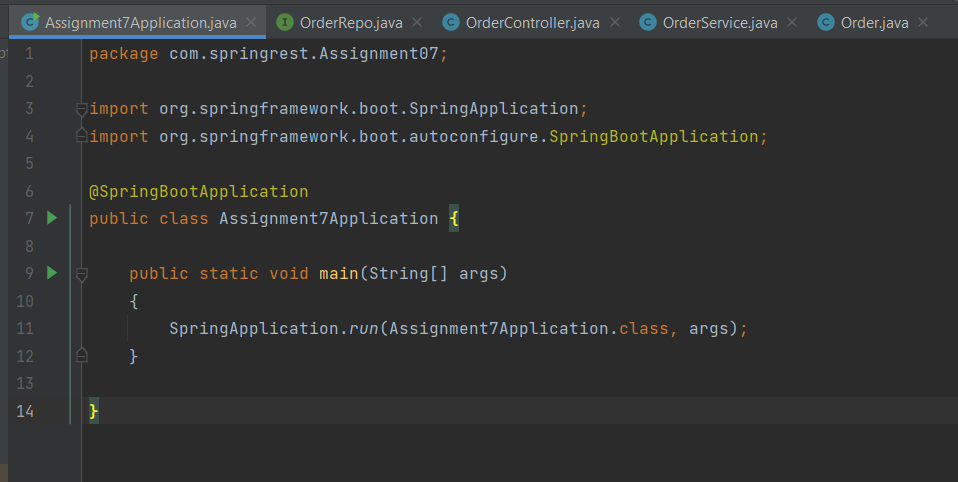
3. A user can view specific order

4. A user can view all the orders

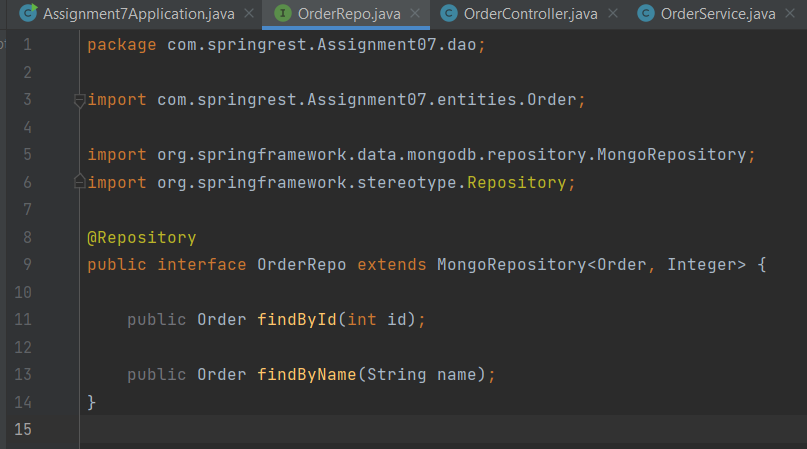
5. A user can delete a specific order.

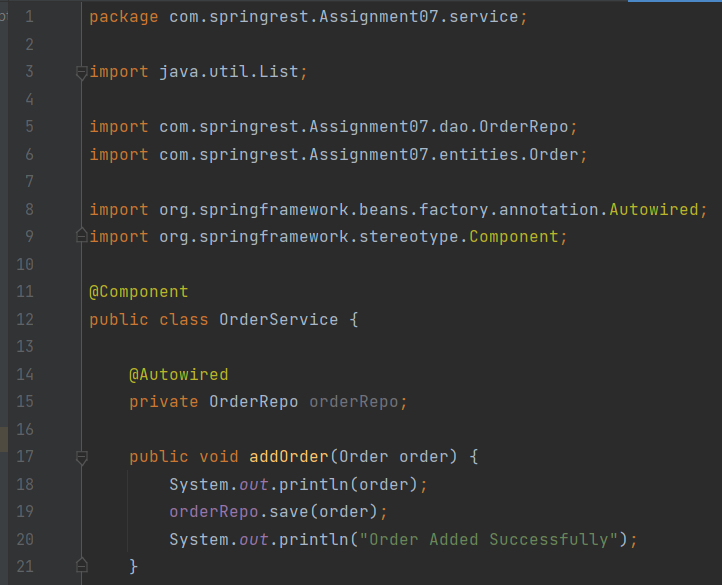
Note: Use MongoRepository of Spring data to store order details

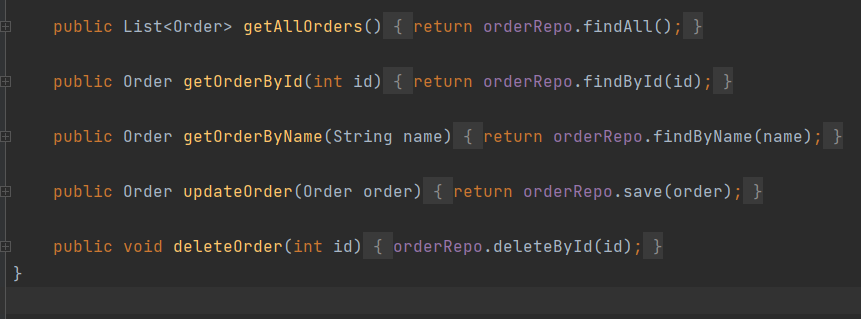
**Assignment7Application.java**



**OrderRepo.java**



**OrderService.java**

****

**Order.java**

package com.springrest.Assignment07.entities;  
  
import org.springframework.data.mongodb.core.mapping.Document;  
  
@Document(collection = "orders")  
public class Order {  
 private int id;  
 private String name;  
 private String quantity;  
 private String price;  
 private String payment;  
  
 public Order() {  
 super();  
 }  
  
 public Order(int id, String name, String quantity, String price, String payment) {  
 this.id = id;  
 this.name = name;  
 this.quantity = quantity;  
 this.price = price;  
 this.payment = payment;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getQuantity() {  
 return quantity;  
 }  
  
 public void setQuantity(String quantity) {  
 this.quantity = quantity;  
 }  
  
 public String getPrice() {  
 return price;  
 }

public void setPrice(String price) {  
 this.price = price;  
 }  
  
 public String getPayment() {  
 return payment;  
 }  
  
 public void setPayment(String payment) {  
 this.payment = payment;  
 }  
  
 @Override  
 public String toString() {  
 return "Order [id=" + id + ", name=" + name + ", payment=" + payment + ", price=" + price + ", quantity="  
 + quantity + "]";  
 }  
  
}

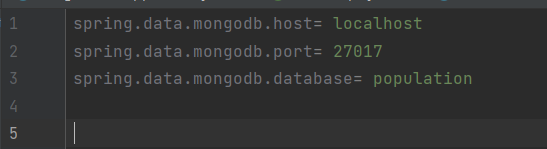
**OrderController.java**

package com.springrest.Assignment07.controller;  
  
import java.util.List;  
import java.util.Optional;  
  
import com.springrest.Assignment07.entities.Order;  
import com.springrest.Assignment07.service.OrderService;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.DeleteMapping;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.PutMapping;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.web.bind.annotation.RestController;

@RestController  
public class OrderController {  
  
 @Autowired  
 private OrderService orderService;  
  
 @PostMapping("/orders")  
 public ResponseEntity<Order> addOrder(@RequestBody Order order) {  
 try {  
 System.*out*.println(order);  
 orderService.addOrder(order);  
 return ResponseEntity.*ok*(order);  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*INTERNAL\_SERVER\_ERROR*).build();  
 }  
 }  
  
 @GetMapping("/orders")  
 public ResponseEntity<List<Order>> getAllOrder() {  
 List<Order> list = orderService.getAllOrders();  
  
 if (list.size() <= 0) {  
 return ResponseEntity.*status*(HttpStatus.*NOT\_FOUND*).build();  
 }  
 return ResponseEntity.*of*(Optional.*of*(list));  
  
 }  
  
 @GetMapping("/orders/{id}")  
 public ResponseEntity<Order> getOrderById(@PathVariable("id") int id) {  
  
 try {  
 Order order = orderService.getOrderById(id);  
 return ResponseEntity.*of*(Optional.*of*(order));  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*NOT\_FOUND*).build();  
 }  
  
 }  
  
 @GetMapping("/orders/name")  
 public ResponseEntity<Order> getOrderByName(@RequestParam(value = "name") String name) {  
 try {  
 Order order = orderService.getOrderByName(name);  
 return ResponseEntity.*of*(Optional.*of*(order));  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*NOT\_FOUND*).build();  
 }  
 }

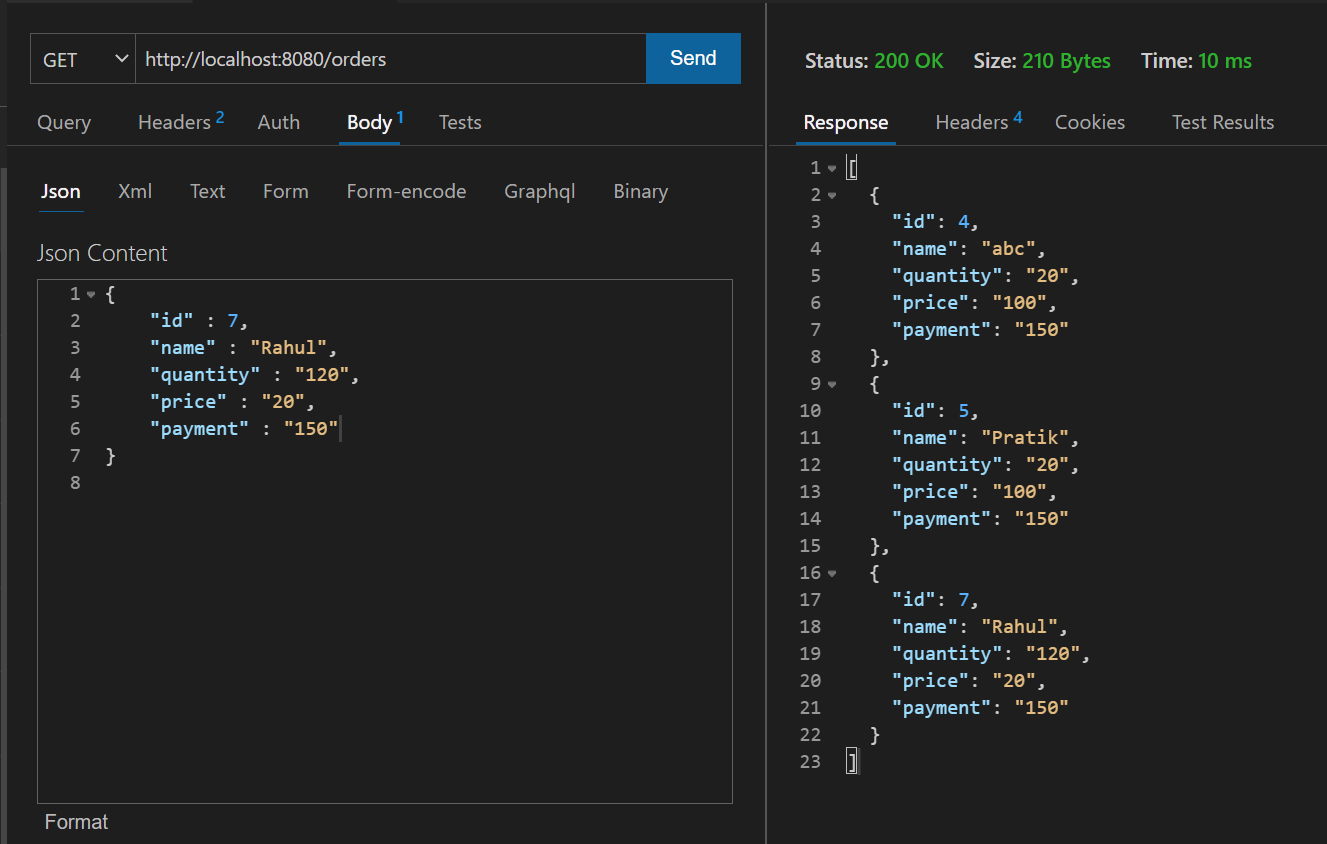
@PutMapping("/orders/{id}")  
 public ResponseEntity<Order> updateOrder(@RequestBody Order order, @PathVariable("id") int id) {  
 Order order1 = null;  
 try {  
 order1 = orderService.updateOrder(order);  
 return ResponseEntity.*ok*().body(order1);  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*INTERNAL\_SERVER\_ERROR*).build();  
 }  
 }  
  
 @DeleteMapping("/orders/{id}")  
 public ResponseEntity<Order> deleteOrder(@PathVariable("id") int id) {  
 try {  
 Order order = orderService.getOrderById(id);  
 orderService.deleteOrder(id);  
 return ResponseEntity.*ok*().body(order);  
 } catch (Exception e) {  
 e.printStackTrace();  
 return ResponseEntity.*status*(HttpStatus.*INTERNAL\_SERVER\_ERROR*).build();  
 }  
 }  
  
}

**application.properties**

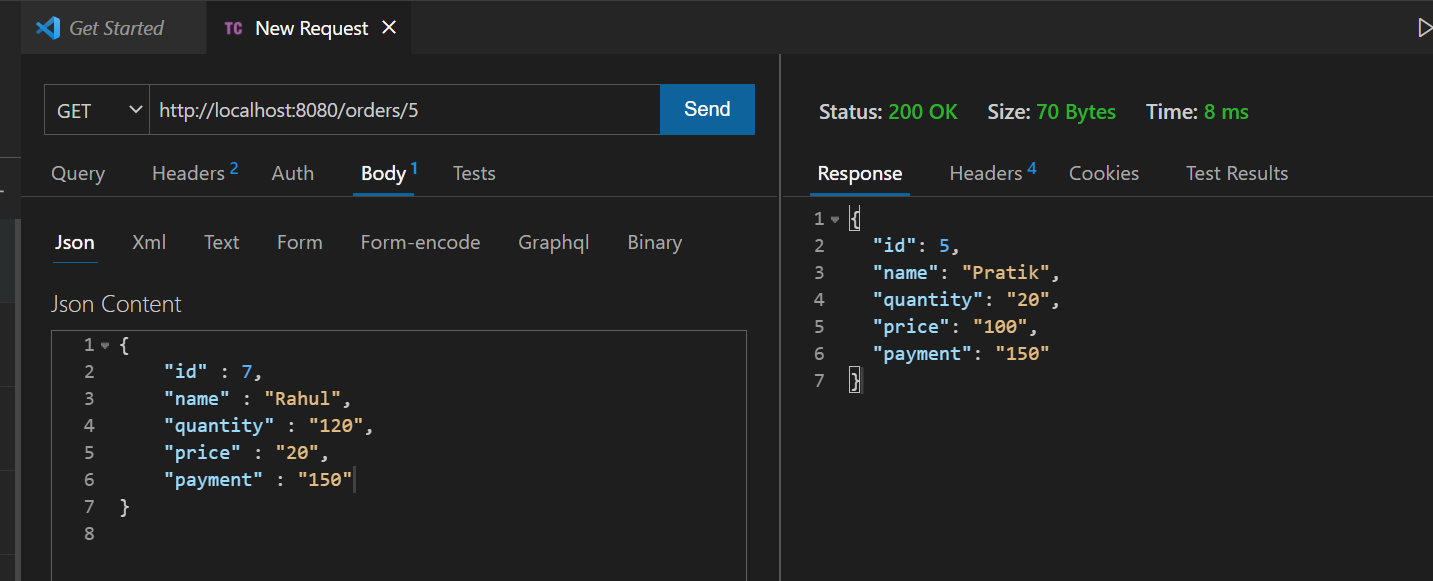
****

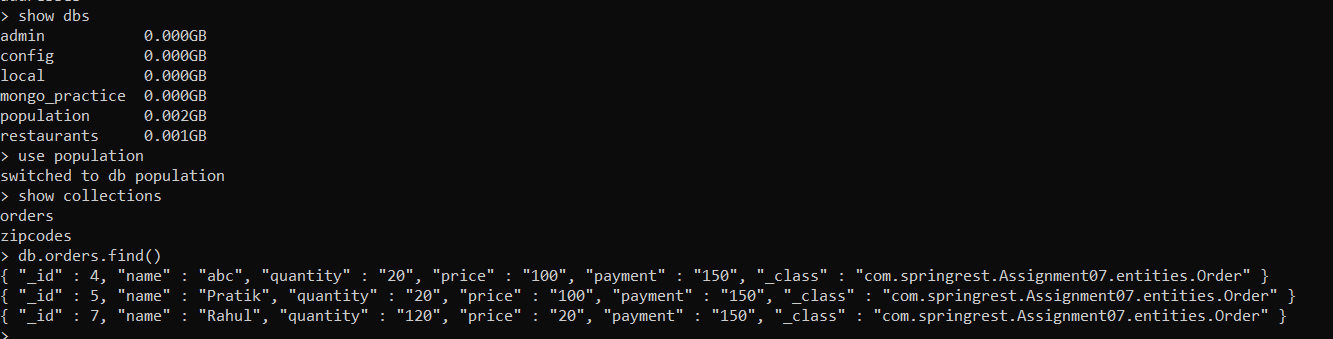
**OUTPUT :-**

**Get all orders**

****

**Get Order By Id**



****

8) Design and develop RESTful web service as follows:

1. An admin can add a new product.

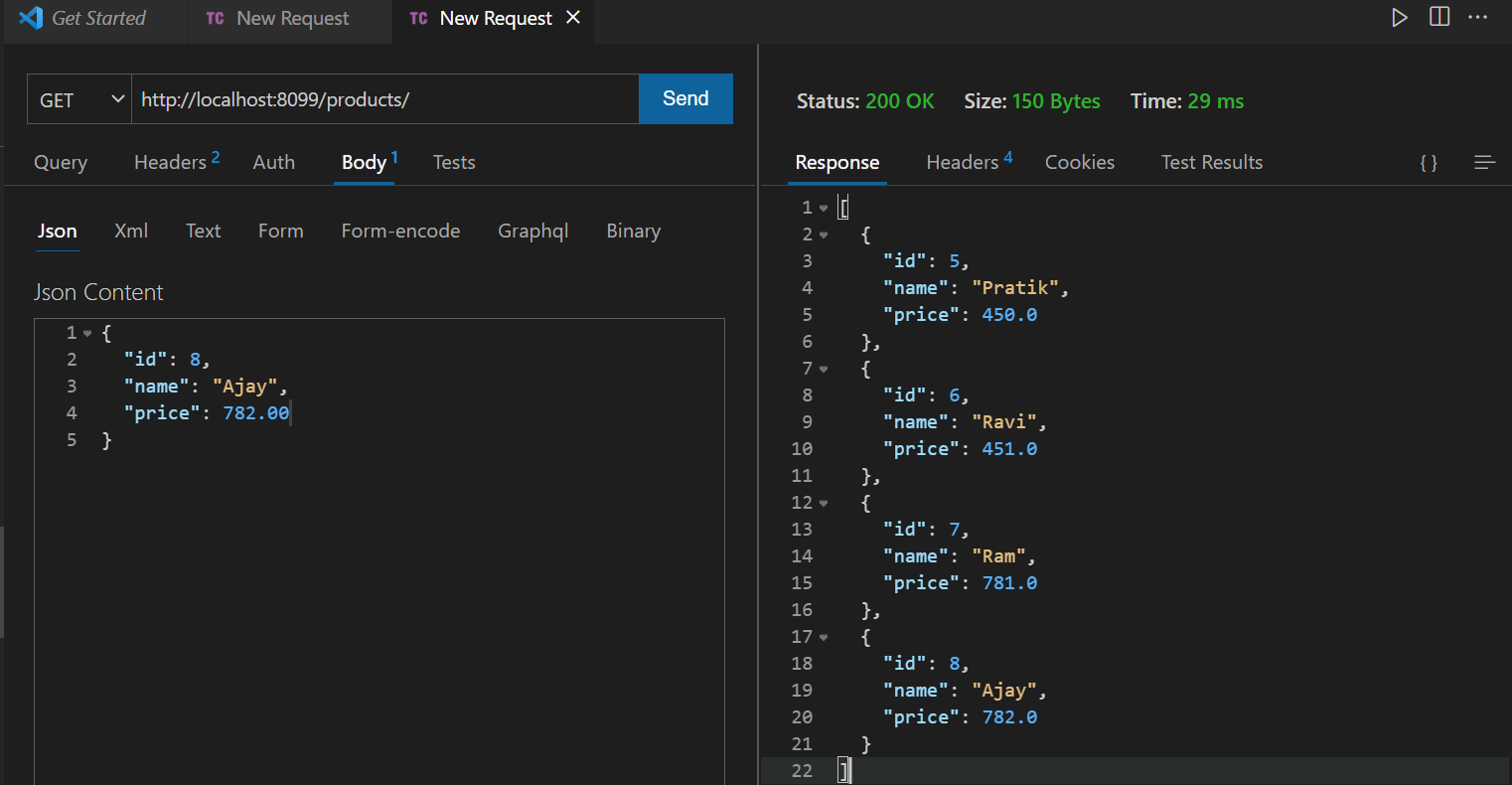
2. An admin can update details of existing product.

3. An admin can delete existing product

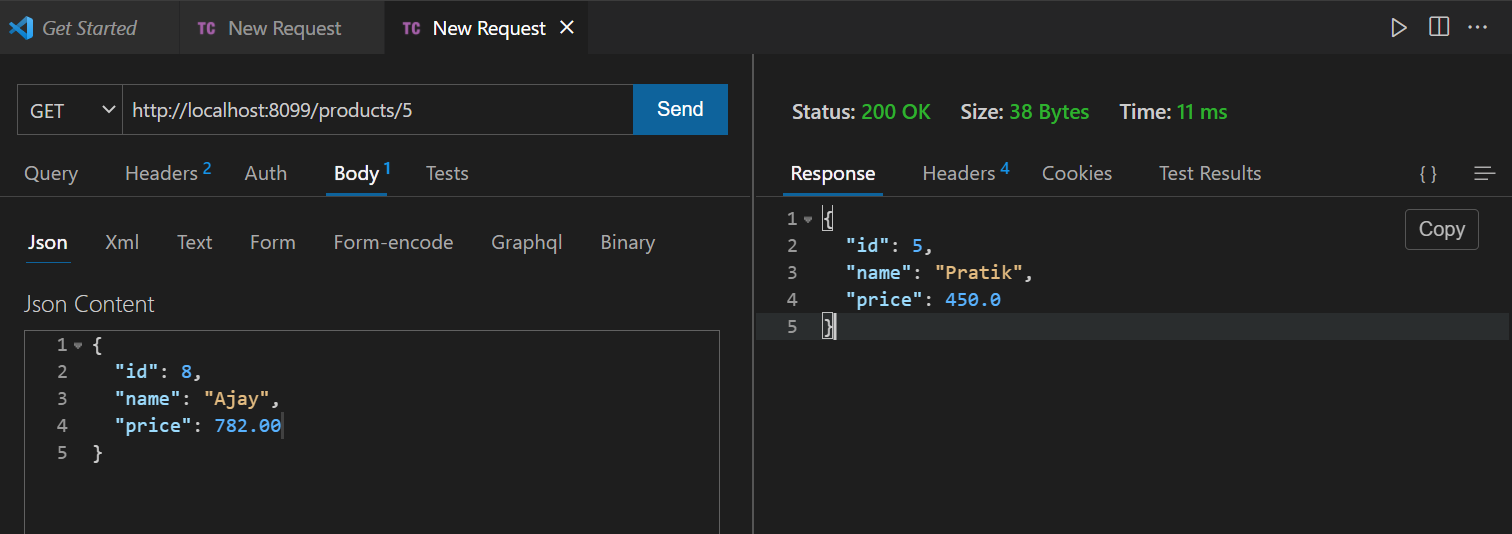
4. An admin can view specific product

5. An admin can view all the products

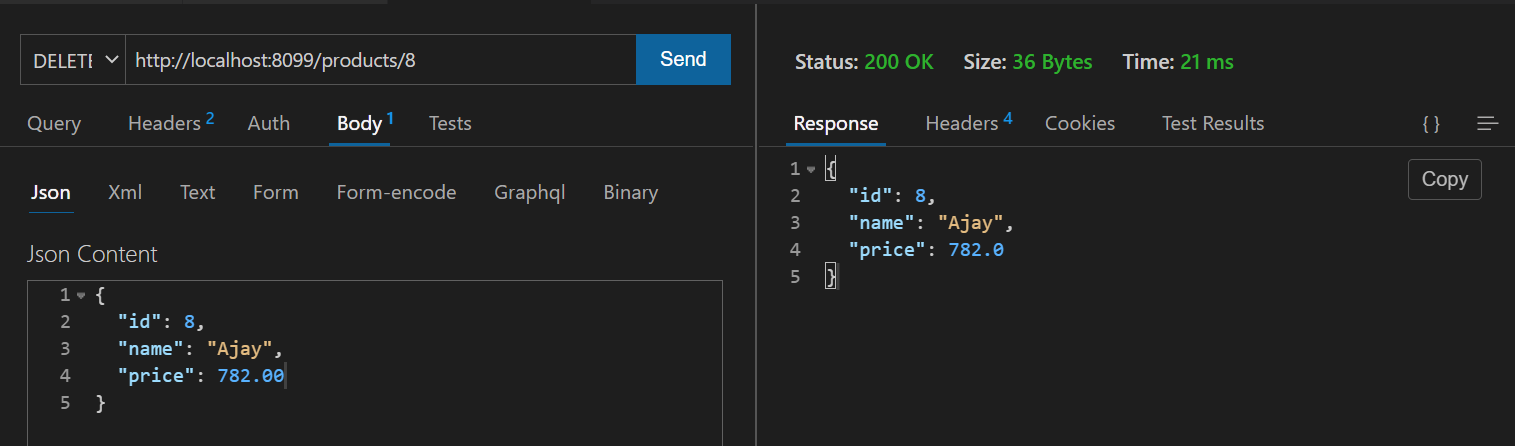
**An admin can view all the products**

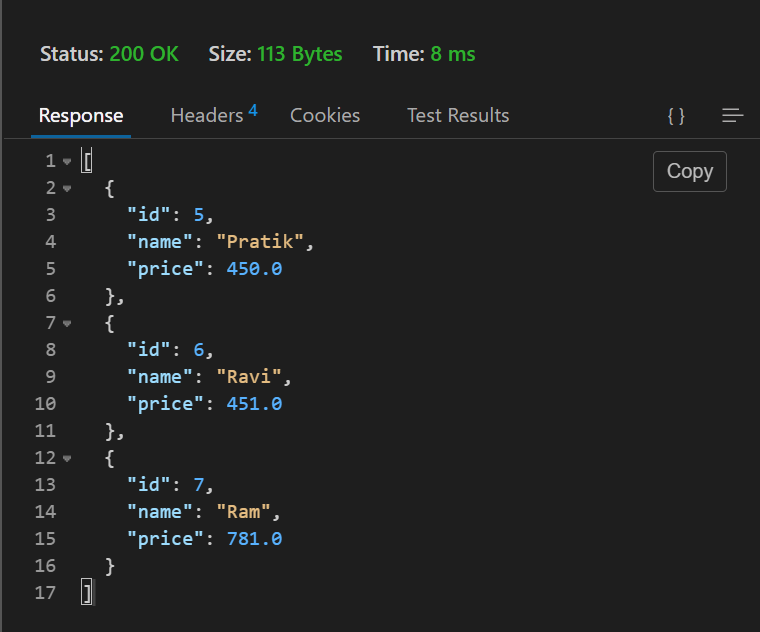


**An admin can view specific product**



**An admin can delete existing product**





9) Design and develop RESTful web service as follows:

1. Add a new customer information.

2. Update customer information.

3. Delete existing customer information.

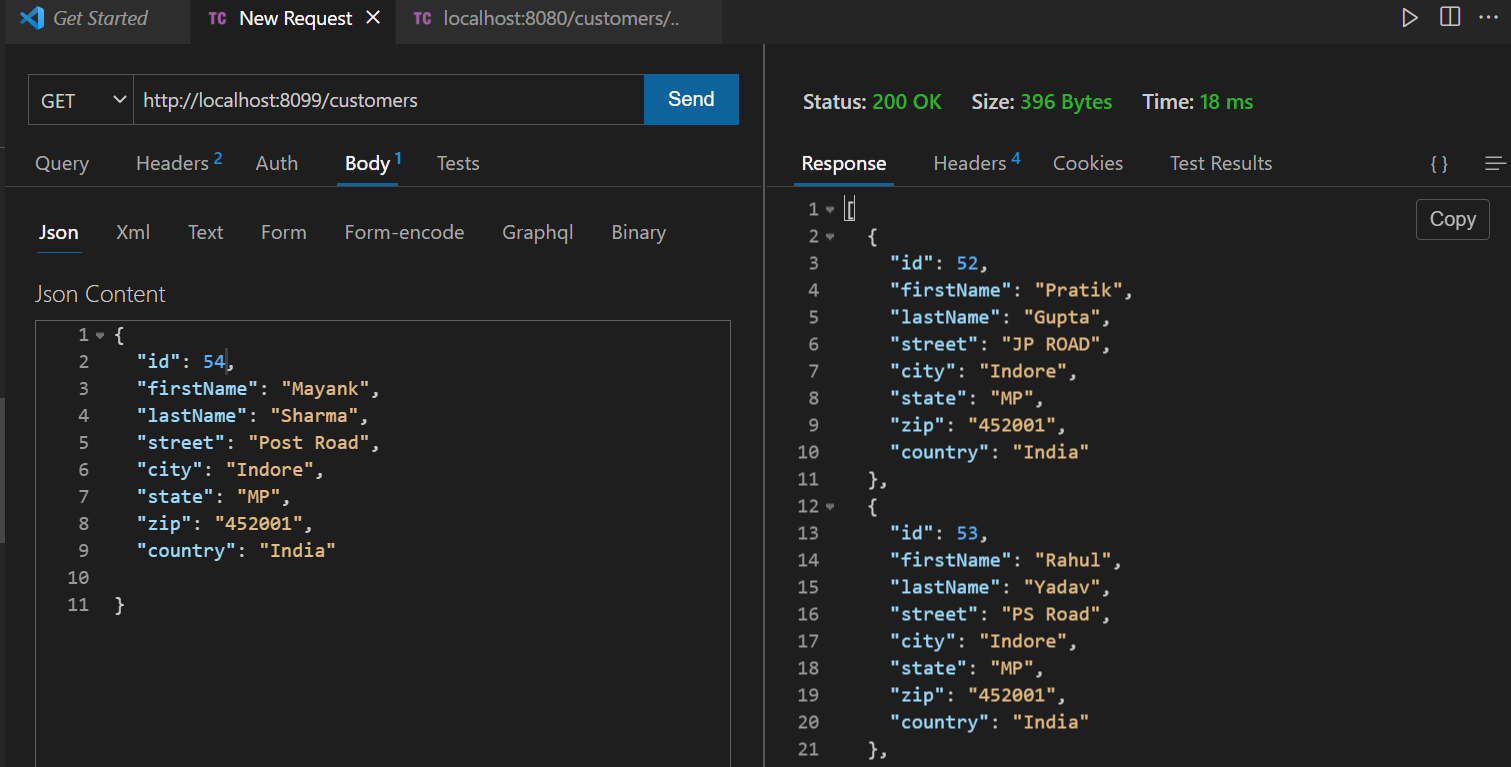
4. Fetch information of specific customer.

5. Fetch information of all customers.

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

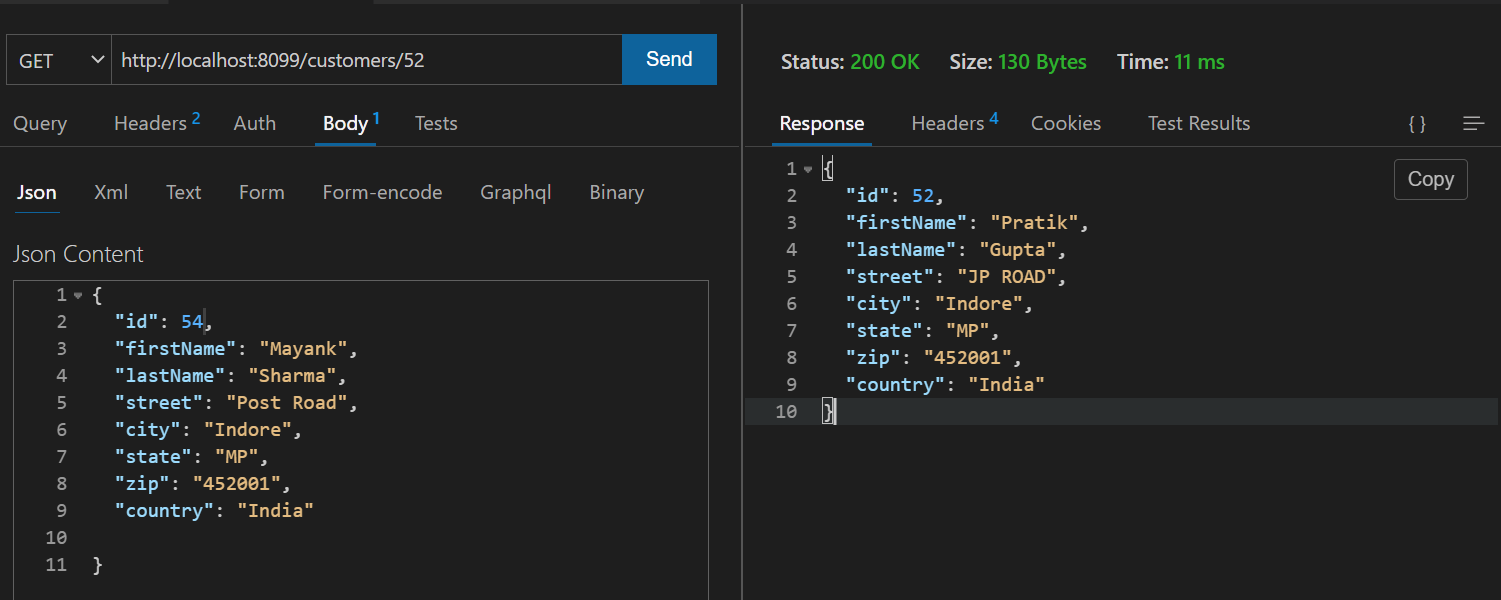
**Output :-**

**Fetch information of all customers.**

****

****

**Fetch information of specific customer.**

****