**Exercises on Microservices with Spring Boot 3.0**

1. **Build a User and Order Management System**

**Problem:** Create two microservices:

• User Service to manage users.

• Order Service to manage orders placed by users.

**Requirements:**

• Use REST APIs.

• Communicate between services using WebClient (Spring WebFlux) or OpenFeign.

• Store data in MySQL or PostgreSQL.

**Program:**

**order-service**

**OrderServiceApplication.java**

package com.example.order\_service;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.openfeign.EnableFeignClients;

@SpringBootApplication

@EnableFeignClients

public class OrderServiceApplication {

public static void main(String[] args) {

SpringApplication.*run*(OrderServiceApplication.class, args);

}

}

**UserClient.java**

package com.example.order\_service.client;

import com.example.order\_service.dto.UserDTO;

import org.springframework.cloud.openfeign.FeignClient;

import org.springframework.web.bind.annotation.\*;

@FeignClient(name = "user-service", url = "http://localhost:8081")

public interface UserClient {

@GetMapping("/users/{id}")

UserDTO getUserById(@PathVariable("id") Long id);

}

**OrderController.java**

package com.example.order\_service.controller;

import com.example.order\_service.client.UserClient;

import com.example.order\_service.dto.UserDTO;

import com.example.order\_service.model.Order;

import com.example.order\_service.service.OrderService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.\*;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

import java.util.Optional;

@RestController

@RequestMapping("/orders")

public class OrderController {

@Autowired

private OrderService orderService;

@Autowired

private UserClient userClient;

@PostMapping

public ResponseEntity<?> createOrder(@RequestBody Order order) {

try {

UserDTO user = userClient.getUserById(order.getUserId());

return ResponseEntity.*ok*(orderService.save(order));

} catch (Exception e) {

return ResponseEntity.*status*(HttpStatus.*BAD\_REQUEST*).body("User does not exist");

}

}

@GetMapping("/{id}")

public Optional<Order> getOrder(@PathVariable Long id) {

return orderService.getById(id);

}

@GetMapping("")

public List<Order> getAllOrders(){

return orderService.getAllOrders();

}

}

**UserDTO.java**

package com.example.order\_service.dto;

import lombok.\*;

@Data

@NoArgsConstructor

@AllArgsConstructor

public class UserDTO {

private Long id;

private String name;

private String email;

}

**Order.java**

package com.example.order\_service.model;

import jakarta.persistence.\*;

import lombok.\*;

@Entity

@Table(name = "orders")

public class Order {

@Id

@GeneratedValue(strategy = GenerationType.*IDENTITY*)

private Long id;

private String product;

public Order() {

}

public Order(Long id, String product, Double amount, Long userId) {

this.id = id;

this.product = product;

this.amount = amount;

this.userId = userId;

}

public Long getUserId() {

return userId;

}

public void setUserId(Long userId) {

this.userId = userId;

}

public Double getAmount() {

return amount;

}

public void setAmount(Double amount) {

this.amount = amount;

}

public String getProduct() {

return product;

}

public void setProduct(String product) {

this.product = product;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

private Double amount;

private Long userId; // External reference to user-service

}

**OrderRepository.java**

package com.example.order\_service.repository;

import com.example.order\_service.model.Order;

import org.springframework.data.jpa.repository.JpaRepository;

public interface OrderRepository extends JpaRepository<Order, Long> {

}

**OrderService.java**

package com.example.order\_service.service;

import com.example.order\_service.model.Order;

import com.example.order\_service.repository.OrderRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.List;

import java.util.Optional;

@Service

public class OrderService {

@Autowired

private OrderRepository orderRepository;

public Order save(Order order) {

return orderRepository.save(order);

}

public Optional<Order> getById(Long id) {

return orderRepository.findById(id);

}

public List<Order> getAllOrders() {

return orderRepository.findAll();

}

}

**User-service**

**UserServiceApplication.java**

package com.example.user\_service;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class UserServiceApplication {

public static void main(String[] args) {

SpringApplication.*run*(UserServiceApplication.class, args);

}

}

**UserController.java**

package com.example.user\_service.controller;

import com.example.user\_service.model.User;

import com.example.user\_service.service.UserService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

import java.util.Optional;

@RestController

@RequestMapping("/users")

public class UserController {

@Autowired

private UserService userService;

@PostMapping

public User createUser(@RequestBody User user) {

return userService.save(user);

}

@GetMapping("")

public List<User> getAllUsers(){

return userService.getAllUsers();

}

@GetMapping("/{id}")

public Optional<User> getUser(@PathVariable Long id) {

return userService.getById(id);

}

}

**User.java**

package com.example.user\_service.model;

import jakarta.persistence.\*;

import lombok.\*;

@Entity

public class User {

@Id

@GeneratedValue(strategy = GenerationType.*IDENTITY*)

private Long id;

public User(Long id, String name, String email) {

this.id = id;

this.name = name;

this.email = email;

}

public User() {

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

private String name;

private String email;

}

**UserRepository.java**

package com.example.user\_service.repository;

import com.example.user\_service.model.User;

import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepository extends JpaRepository<User, Long> {

}

**UserService.java**

package com.example.user\_service.service;

import com.example.user\_service.model.User;

import com.example.user\_service.repository.UserRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.List;

import java.util.Optional;

@Service

public class UserService {

@Autowired

private UserRepository userRepository;

public User save(User user) {

return userRepository.save(user);

}

public Optional<User> getById(Long id) {

return userRepository.findById(id);

}

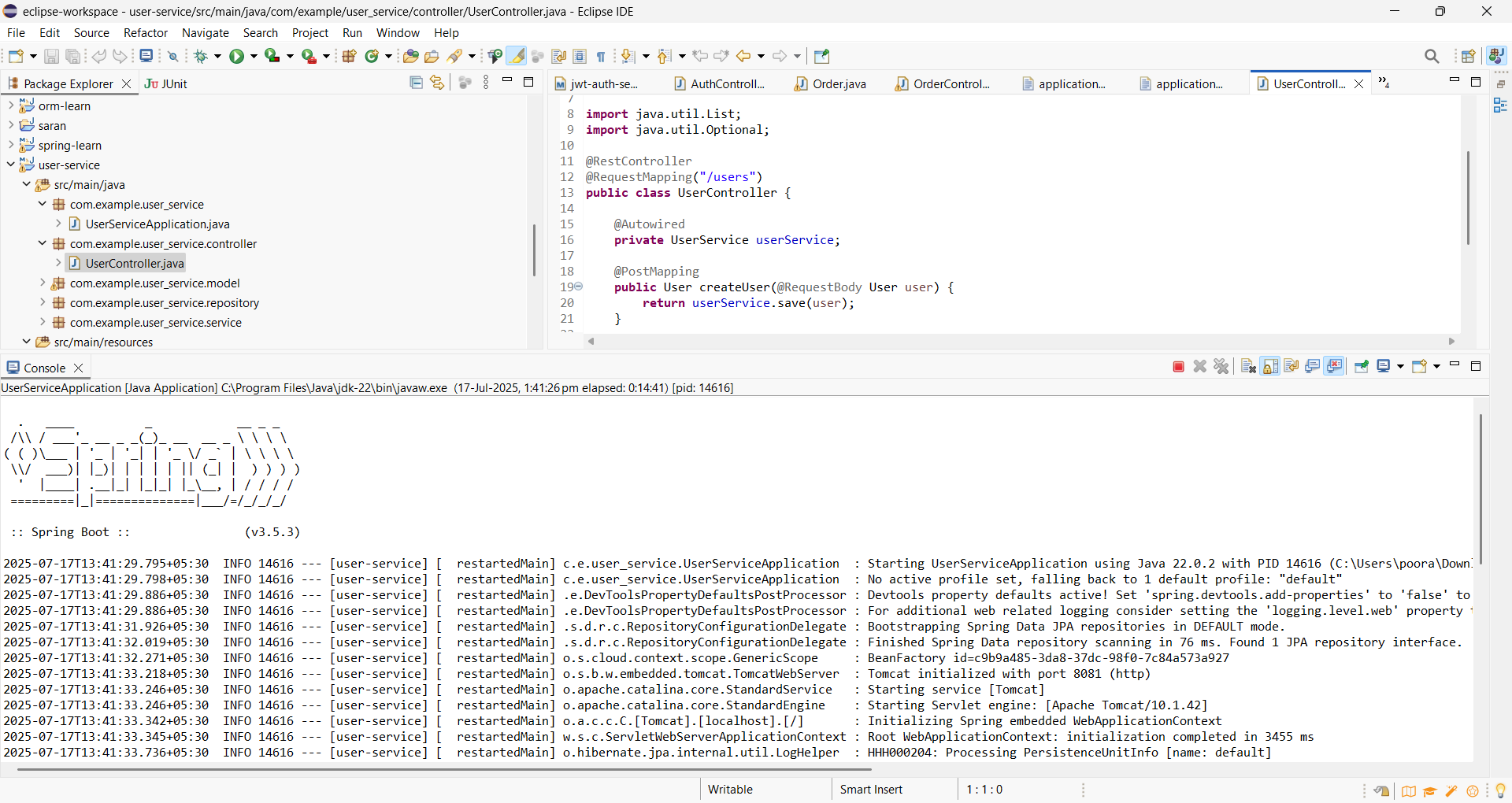
public List<User> getAllUsers() {

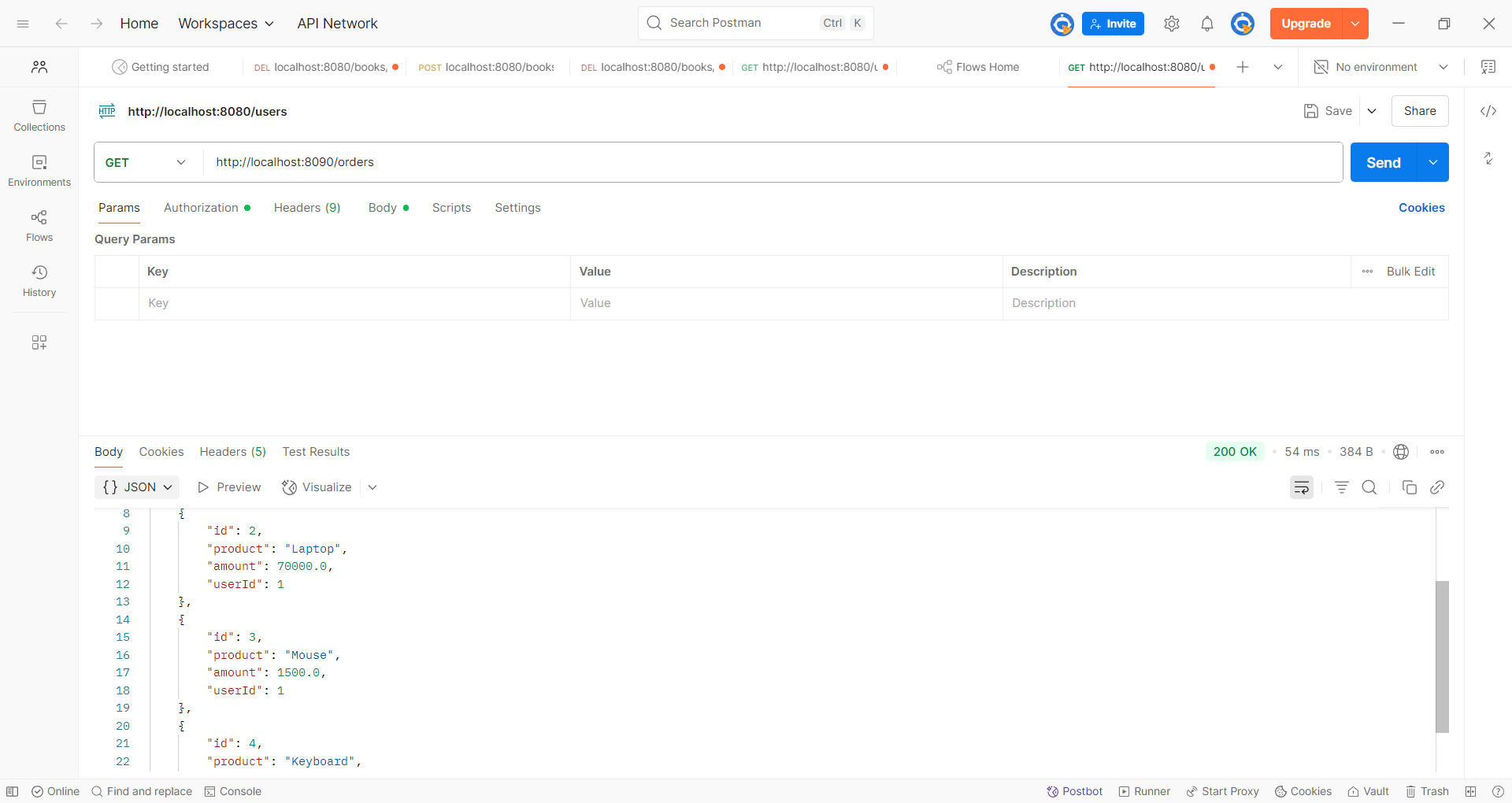
return userRepository.findAll();

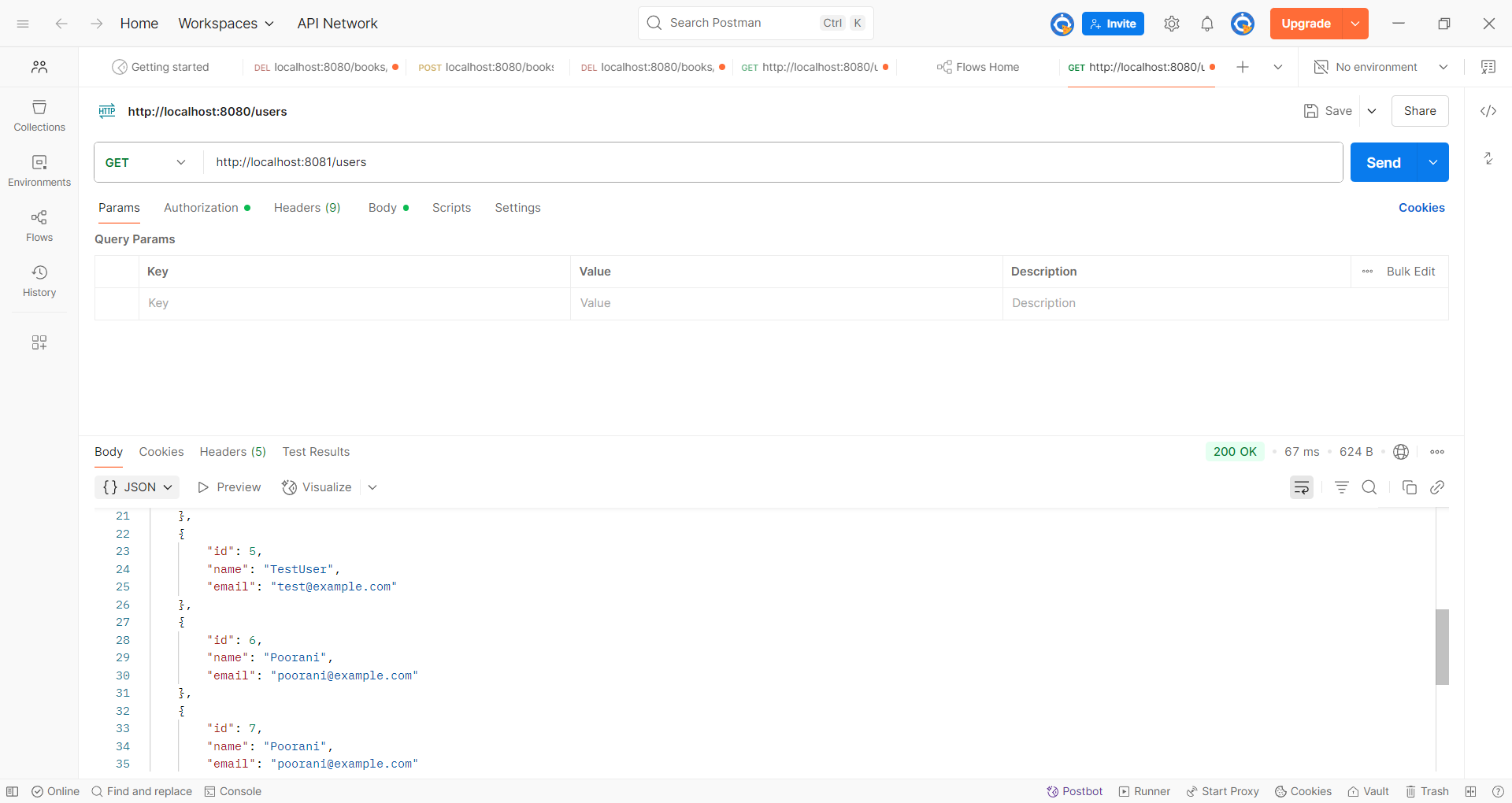
}

}

**Output:**







**2. Inventory Management System with Service Discovery** Problem: Create: • Product Service: Manage products and stock. • Inventory Service: Track stock levels for each product. Requirements: • Use Spring Cloud Netflix Eureka for service discovery. • Implement centralized configuration using Spring Cloud Config Server.

