

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**ĐỀ TÀI: XÂY DỰNG WEB SERVICE BẰNG
SPRING BOOT CHO ECOMMERCE
WEBSITE**

Sinh viên thực hiện:			
STT	Họ tên	MSSV	Ngành
1	Trần Thành Thoại	23540035	Công nghệ thông tin

TP. HỒ CHÍ MINH – 05/2025

1. GIỚI THIỆU

Đề tài “Xây dựng web service bằng Spring Boot cho ecommerce website” nhằm phát triển một hệ thống backend API phục vụ cho website thương mại điện tử. Hệ thống cung cấp API RESTful bao gồm quản lý người dùng, xác thực phân quyền, quản lý đơn hàng và địa chỉ giao hàng.

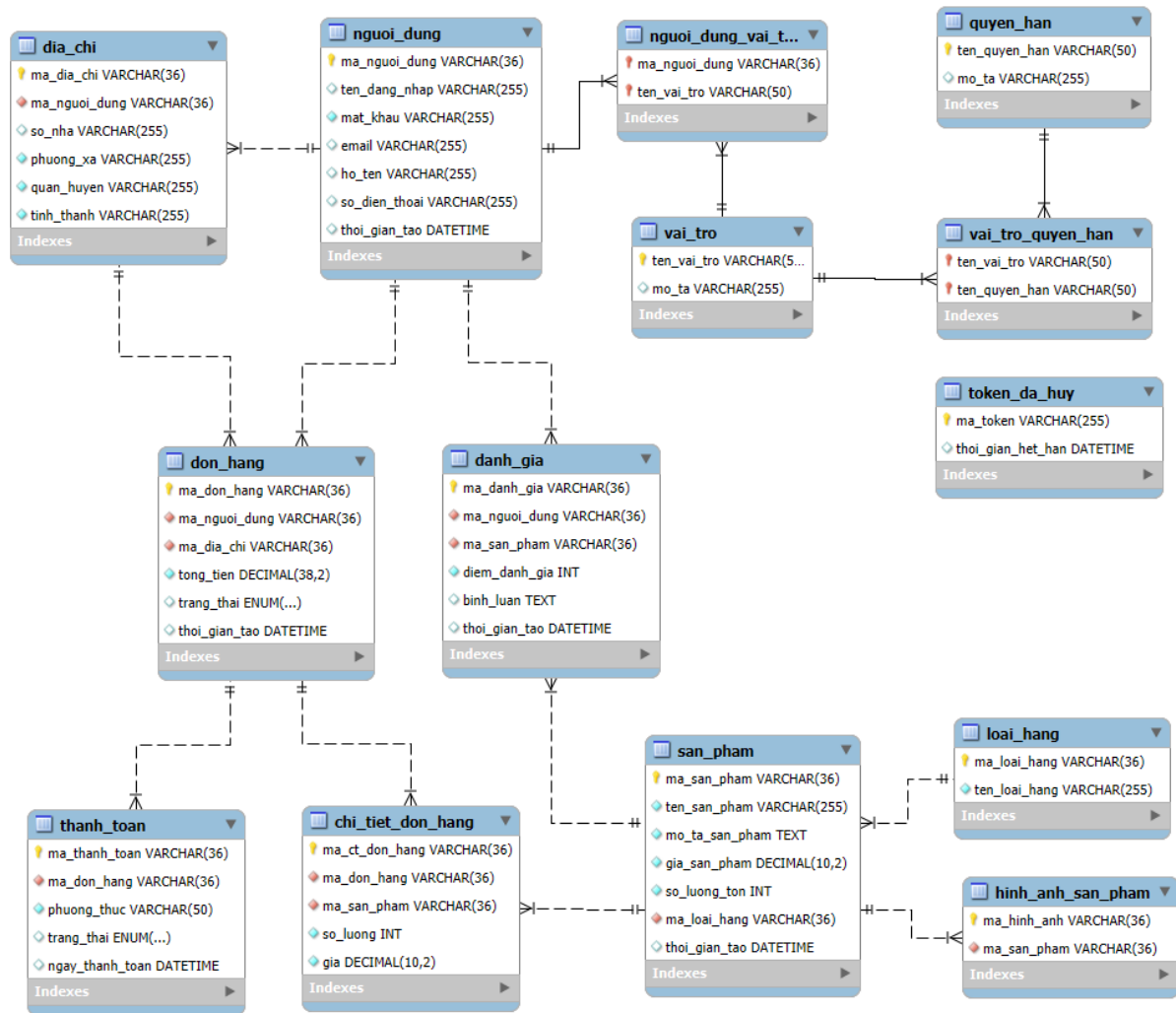
– Công nghệ sử dụng:

- + Ngôn ngữ lập trình: Java 21.
- + Framework: Spring Boot 3.4.5.
- + Cơ sở dữ liệu: MySQL.
- + Bảo mật: Spring Security với JWT (JSON Web Token).
- + ORM (Object-Relational Mapping): JPA (Java Persistence API)/Hibernate.
- + Build tool: Maven.

Cam kết: Đề tài do em tự phân tích thiết kế và không dựa trên đề tài nào khác.

– Link github của dự án: <https://github.com/thoai240699/ecommerce-service.git>

2. MÔ TẢ CƠ SỞ DỮ LIỆU



Hình 1. Mô hình ERD của cơ sở dữ liệu ecommerce_service.

Cơ sở dữ liệu được thiết kế theo mô hình quan hệ nhằm lưu trữ và quản lý dữ liệu hiệu quả cho hệ thống thương mại điện tử. Thiết kế tuân thủ nguyên tắc ACID và chuẩn hóa theo Third Normal Form (3NF) để đảm bảo tính toàn vẹn và giảm thiểu dư thừa dữ liệu. Phân quyền người dùng thiết kế theo mô hình **RBAC** (Role-Base Access Control). Trong đó quyền truy cập vào các chức năng, tài nguyên của hệ thống được gán cho các vai trò (Role). Người dùng (User) sẽ được gán một hoặc nhiều vai trò và thông qua vai trò đó thì sẽ có các quyền hạn (Permission) tương ứng.

Dữ liệu được tổ chức thành các bảng. Mỗi bảng chứa các hàng và cột đại diện cho mỗi **thực thể** (Entity). Mỗi hàng đại diện cho một đối tượng cụ thể, chứa một tập các thuộc tính duy nhất. Mỗi cột là các **thuộc tính** (Attribute) chứa các điểm dữ liệu cụ thể để mô tả mỗi thực thể. **Mối quan hệ** (relationship) giữa các thực thể được thiết lập thông qua các khóa. Cơ sở dữ liệu theo quy ước đặt tên **snake_case** để đặt tên bảng, tên thuộc tính. Sau đây là các mô tả của cơ sở dữ liệu, các thực thể khác như đơn hàng, loại hàng, sản phẩm, hình ảnh sản phẩm, chi tiết đơn hàng, thanh toán, đánh giá được mô tả tương tự và được trình bày trong **phụ lục mô tả các bảng khác** của cơ sở dữ liệu.

2.1. Thực thể người dùng

- Chức năng: Lưu trữ thông tin tài khoản của tất cả người dùng hệ thống.
- Các thuộc tính:
 - + `ma_nguoi_dung` (Khóa chính): Mã định danh duy nhất cho mỗi người dùng (UUID).
 - + `ten_dang_nhap`: Tên đăng nhập, duy nhất, dùng để đăng nhập hệ thống.
 - + `mat_khau`: Mật khẩu đã được mã hóa.
 - + `email`: Địa chỉ email, duy nhất.
 - + `ho_ten`: Họ tên đầy đủ của người dùng.
 - + `so_dien_thoai`: Số điện thoại liên hệ.
 - + `thoi_gian_tao`: Thời điểm tạo tài khoản.
- Mối quan hệ:
 - + 1-n với bảng `dia_chi`: Một người dùng có thể có nhiều địa chỉ.
 - + 1-n với bảng `don_hang`: Một người dùng có thể đặt nhiều đơn hàng.
 - + n-n với bảng `vai_tro` qua bảng `nguoi_dung_vai_tro`: Một người dùng có thể có nhiều vai trò.
 - + 1-n với bảng `danh_gia`: Một người dùng có thể đánh giá nhiều sản phẩm.

Bảng 1. `nguoi_dung`

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	<code>ma_nguoi_dung</code>	VARCHAR(36)	PRIMARY KEY
2	<code>ten_dang_nhap</code>	VARCHAR(50)	NOT NULL, UNIQUE
3	<code>mat_khau</code>	VARCHAR(255)	NOT NULL
4	<code>email</code>	VARCHAR(255)	UNIQUE
5	<code>ho_ten</code>	VARCHAR(255)	
6	<code>so_dien_thoai</code>	VARCHAR(20)	
7	<code>thoi_gian_tao</code>	DATETIME	DEFAULT CURRENT_TIMESTAMP

2.2. Thực thể vai trò

- Chức năng: Lưu trữ thông tin vai trò của người dùng, bao gồm: ADMIN, SELLER, CUSTOMER.
- Các thuộc tính:
 - + `ten_vai_tro` (Khóa chính): Tên vai trò duy nhất.
 - + `mo_ta`: Mô tả vai trò.

- Mỗi quan hệ:
 - + n-n với bảng `nguoi_dung` qua bảng `nguoi_dung_vai_tro`: Một vai trò có thể gán cho nhiều người dùng.
 - + n-n với bảng `quyen_han` qua bảng `vai_tro_quyen_han`: Một vai trò có thể có nhiều quyền hạn.

Bảng 2. vai_tro

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	<code>ten_vai_tro</code>	VARCHAR(50)	PRIMARY KEY
2	<code>mo_ta</code>	TEXT	

2.3. Thực thể quyền hạn

- Chức năng: Lưu thông tin quyền hạn của hệ thống.
- Các thuộc tính:
 - + `ten_quyen_han` (Khóa chính): Tên quyền hạn duy nhất.
 - + `mo_ta`: Mô tả quyền hạn, có khả năng thao tác các tác vụ nào đó.
- Mỗi quan hệ:
 - + n-n với bảng `vai_tro` qua bảng `vai_tro_quyen_han`: Một quyền hạn có thể thuộc nhiều vai trò.

Bảng 3. quyen_han

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	<code>ten_quyen_han</code>	VARCHAR(50)	PRIMARY KEY
2	<code>mo_ta</code>	TEXT	

2.4. Bảng phụ giữa người dùng với vai trò và giữa vai trò với quyền hạn

- Bảng `nguoi_dung_vai_tro` có chức năng tạo liên kết nhiều nhiều (n-n) giữa người dùng và vai trò. Khóa chính là (`ma_nguoi_dung`, `ten_vai_tro`).

Bảng 4. nguoi_dung_vai_tro

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	<code>ma_nguoi_dung</code>	VARCHAR(36)	NOT NULL, FOREIGN KEY
2	<code>ten_vai_tro</code>	VARCHAR(50)	NOT NULL, FOREIGN KEY

- Bảng `vai_tro_quyen_han` có chức năng tạo liên kết nhiều nhiều (n-n) giữa vai trò và quyền hạn. Khóa chính là (`ten_vai_tro`, `ten_quyen_han`).

Bảng 5. vai_tro_quyen_han

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	ten_vai_tro	VARCHAR(50)	NOT NULL, FOREIGN KEY
2	ten_quyen_han	VARCHAR(50)	NOT NULL, FOREIGN KEY

2.5. Thực thể Token đã hủy

- Chức năng: Lưu trữ các JWT token đã bị vô hiệu hóa khi logout. Khi đăng xuất thì token của người dùng vẫn còn hiệu lực cho đến khi hết hạn, khi đăng nhập nếu token có nằm trong bảng dữ liệu này thì sẽ từ chối đăng nhập.
- Các thuộc tính:
 - + ma_token (Khóa chính): Chứa mã token đã hủy.
 - + thoi_gian_het_han: Thời gian hết hạn của Token.
- Mối quan hệ: Không có liên kết với các bảng khác

Bảng 6. token_da_huy

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	ma_token	VARCHAR(50)	PRIMARY KEY
2	thoi_gian_het_han	DATETIME	

2.6. Thực thể địa chỉ

- Chức năng: Lưu trữ các địa chỉ giao hàng của người dùng.
- Các thuộc tính:
 - + ma_dia_chi (Khóa chính): Mã định danh địa chỉ.
 - + ma_nguoi_dung (Khóa ngoại): Tham chiếu đến người dùng sở hữu địa chỉ này.
 - + so_nha: Số nhà, tên đường.
 - + phuong_xa: Phường/xã.
 - + quan_huyen: Quận/huyện.
 - + tinh_thanh: Tỉnh/thành phố.
- Mối quan hệ:
 - + n-1 với bảng nguoi_dung: Mỗi địa chỉ thuộc về một người dùng.
 - + 1-n với bảng don_hang: Một địa chỉ có thể được dùng cho nhiều đơn hàng.

Bảng 7. địa_chi

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	ma_dia_chi	VARCHAR(36)	PRIMARY KEY
2	ma_nguoi_dung	VARCHAR(36)	FOREIGN KEY
3	so_nha	VARCHAR(50)	NOT NULL
4	phuong_xa	VARCHAR(255)	NOT NULL
5	quan_huyen	VARCHAR(255)	NOT NULL
6	tin_h_thanh	VARCHAR(255)	NOT NULL

3. THIẾT KẾ HỆ THỐNG

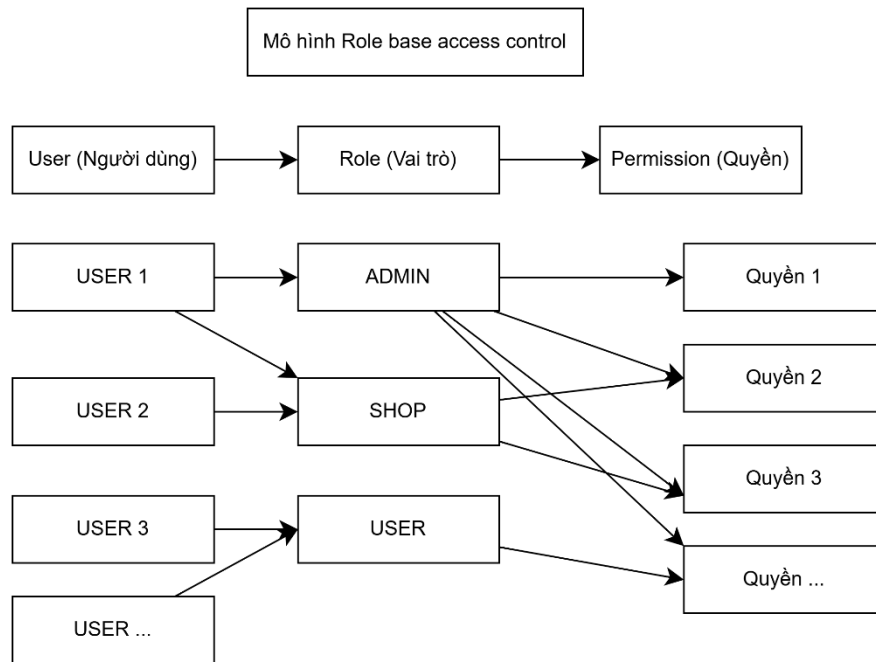
3.1. Cấu trúc thư mục dự án

ecommerce-service/	
src/	
main/	
java/com/thoai/ecommerce_service/	
configuration/	
constant/	
controller/	
dto/	
request/	
response/	
entity/	
enums/	
exception/	
mapper/	
repository/	
service/	
EcommerceServiceApplication.java	
resources/	
application.yaml	
pom.xml	

3.2. Hệ thống phân quyền

Bảng 8. Phân quyền

Vai trò	Quyền hạn chính
USER	Người dùng - Có thể mua hàng, quản lý đơn hàng và thông tin cá nhân.
SHOP	Cửa hàng - Có thể quản lý sản phẩm, xử lý đơn hàng.
ADMIN	Quản trị viên - Có toàn quyền quản lý hệ thống.



Hình 2. Mô hình RABC

3.3. Bảo mật

Mật khẩu của người dùng (User) được mã hóa khi tạo, sử dụng thuật toán BCrypt với strength = 8 thông qua Spring Security Crypto. Ví dụ: Mật khẩu “thoai123xyz” sẽ được mã hóa thành

“\$2a\$09\$0LtkcfuozCndWoHbwzmW6OG8TnKP2.h/pdMKCiOLOyqcnwB0fUTtK”.

Khi triển khai thực tế thì không thể bắt người dùng đăng nhập lại mỗi khi chuyển trang, vì vậy JWT (JSON Web Token) sinh ra để đáp ứng việc này. Phần header sử dụng thuật toán HS512. Phần Payload được thiết kế như sau:

- + sub (): Username của người dùng.
 - + iss (issuer): “thoai.com” - domain phát hành token.
 - + iat (issued at): Thời gian phát hành token.
 - + exp (expiration): Thời gian hết hạn token, 4 giờ từ lúc phát hành.
 - + jti (JWT ID): ID duy nhất của token.
 - + scope: Chuỗi chứa vai trò và quyền hạn.
- Phần Signature sử dụng HMAC (HS512) với secret key là “7OC1+qqZCfQqdQijYq1xiA6mu15k8Ho66tsGBYHmrsEMWnTzEI17L/ECLUOyLltu” được generate ngẫu nhiên.

3.4. Các API ENDPOINT

Base URL: <http://localhost:8080/ecommerce>. Các API đã được test bằng POSTMAN, kết quả được trình bày trong phụ lục hình ảnh.

Bảng 9. Các API ENDPOINT

Method	Endpoint	Mô tả	Quyền cần thiết
Authentication			
POST	/auth/token	Đăng nhập	Public
POST	/auth/introspect	Kiểm tra token	Public
POST	/auth/logout	Đăng xuất	Public
User			
POST	/users	Tạo tài khoản	Public
GET	/users	Lấy danh sách users	USER_READ_ALL
GET	/users/{userId}	Lấy thông tin user	USER_READ_ALL hoặc chính user
GET	/users/myInfo	Lấy thông tin cá nhân	USER_READ
PUT	/users/{userId}	Cập nhật user	USER_UPDATE
DELETE	/users/{userId}	Xóa user	USER_DELETE
Permission			
POST	/permissions	Tạo quyền hạn	PERMISSION_CREATE
GET	/permissions	Lấy danh sách	PERMISSION_READ
DELETE	/permissions/{permission}	Xóa quyền hạn	PERMISSION_DELETE
Role			
POST	/roles	Tạo vai trò	ROLE_CREATE
GET	/roles	Lấy danh sách	ROLE_READ
PUT	/roles/{roleName}	Cập nhật vai trò	ROLE_UPDATE
DELETE	/roles/{roleName}	Xóa vai trò	ROLE_DELETE
Address			
POST	/addresses	Tạo địa chỉ	ADDRESS_CREATE
GET	/addresses	Lấy tất cả địa chỉ	ADDRESS_READ_ALL
GET	/addresses/{userId}	Lấy địa chỉ theo userId	ADDRESS_READ_ALL
GET	/addresses/myAddresses	Lấy địa chỉ của mình	ADDRESS_READ
PUT	/addresses/{addressId}	Cập nhật địa chỉ	ADDRESS_UPDATE
DELETE	/addresses/{addressId}	Xóa địa chỉ	ADDRESS_DELETE
Order			
POST	/orders	Tạo đơn hàng	ORDER_CREATE

Method	Endpoint	Mô tả	Quyền cần thiết
GET	/orders	Lấy tất cả đơn hàng	ORDER_READ_ALL
GET	/orders/{orderId}	Lấy đơn hàng theo ID	ORDER_READ_ALL hoặc chủ đơn hàng
GET	/orders/myOrders	Lấy đơn hàng của mình	ORDER_READ
PUT	/orders/{orderId}	Cập nhật đơn hàng	ORDER_UPDATE
DELETE	/orders/{orderId}	Xóa đơn hàng	ORDER_DELETE
PUT	/orders/{orderId}/cancel	Hủy đơn hàng	ORDER_CANCEL
PUT	/orders/{orderId}/approve	Duyệt đơn hàng	ORDER_APPROVE

4. ỨNG DỤNG MÔ HÌNH MVC

Hệ thống được thiết kế theo kiến trúc 3 lớp (Three Layer Design): Controller (API) ↔ Service (Business Logic) ↔ Repository (Data Access). Service là phần trung gian giữa Repository và Controller, chứa các business logic. Controller giao tiếp với Service và Service giao tiếp với Repository. Áp dụng Three layer design đảm bảo code sạch và dễ bảo trì, mở rộng, tách biệt rõ ràng các tầng, dễ kiểm thử.

- **Model** là entity (Thực thể dữ liệu): Nằm trong package entity/, đại diện cho các bảng trong database, được ánh xạ bằng Java Persistence API.
- **Controller Layer (Presentation):** Xử lý HTTP requests và responses.
 - + UserController: Quản lý tài khoản người dùng
 - + AuthenticationController: Xác thực và JWT token
 - + RoleController: Quản lý vai trò
 - + PermissionController: Quản lý quyền hạn
 - + AddressController: Quản lý địa chỉ
 - + OrderController: Quản lý đơn hàng
- **Service Layer (Business Logic):**
 - + UserService: Logic nghiệp vụ người dùng
 - + AuthenticationService: Logic xác thực JWT
 - + RoleService: Logic quản lý vai trò
 - + PermissionService: Logic quản lý quyền hạn
 - + AddressService: Logic quản lý địa chỉ
 - + OrderService: Logic quản lý đơn hàng
- **Repository Layer (Data Access):**
 - + UserRepository: Truy xuất dữ liệu User
 - + RoleRepository: Truy xuất dữ liệu Role
 - + PermissionRepository: Truy xuất dữ liệu Permission
 - + AddressRepository: Truy xuất dữ liệu Address

- + OrderRepository: Truy xuất dữ liệu Order
- + InvalidatedTokenRepository: Quản lý token blacklist, các token đã hủy.
- Data Transfer Objects: chia thành request và response, chuyển data giữa các layer.
- Mapper (MapStruct): Chuyển đổi Entity ↔ DTO
- SecurityConfig: Cấu hình Spring Security
- GlobalExceptionHandler: Xử lý lỗi toàn cục
- CustomJwtDecoder: Tùy chỉnh JWT decoder

5. KẾT LUẬN

Đồ án được bắt đầu từ giai đoạn xác định các logic của một web thương mại điện tử, thiết kế cơ sở dữ liệu, thiết kế API, thiết kế hệ thống phân quyền, triển khai bằng framework Spring Boot, sau đó kiểm thử bằng Postman.

Đồ án đã thiết kế được **30 API** Web service sử dụng cho hệ thống thương mại điện tử, bao gồm các tính năng đăng ký, đăng nhập, sinh và xác thực JWT, phân quyền theo vai trò, quản lý người dùng, quản lý đơn hàng, quản lý địa chỉ.

Đồ án áp dụng kiến trúc Three-layer Architecture, MVC, thiết kế hệ thống phân quyền theo RBAC, tạo các RESTful API với response chuẩn, xử lý lỗi và phản hồi lỗi có ý nghĩa, thực hiện validation request. Chức năng hài lòng nhất mà dự án em triển khai là triển khai được JWT Token, tiến hành phân quyền được theo mô hình RBAC giúp bảo vệ các API hiệu quả.

TÀI LIỆU THAM KHẢO

- [1] Laurentiu Spilca, Spring Start Here - Learn what you need and learn it well, Manning, 2021.
- [2] Laurentiu Spilca, Spring Security in Action, Second edition, Manning, 2024.
- [3] Craig Walls - Spring in Action, Sixth edition, Manning, 2022.
- [4] Spring Boot Documentation. Link: <https://docs.spring.io/spring-boot/documentation.html> (02/05/2025).
- [5] Spring Security. Link: <https://docs.spring.io/spring-security/reference/index.html> (06/05/2025).
- [6] Introduction to JSON Web Tokens. Link: <https://jwt.io/introduction/> (12/05/2025).
- [7] MySQL 8.0 Reference Manual. Link: <https://dev.mysql.com/doc/refman/8.0/en/> (02/05/2025).

PHỤ LỤC PHÂN CÔNG NHIỆM VỤ

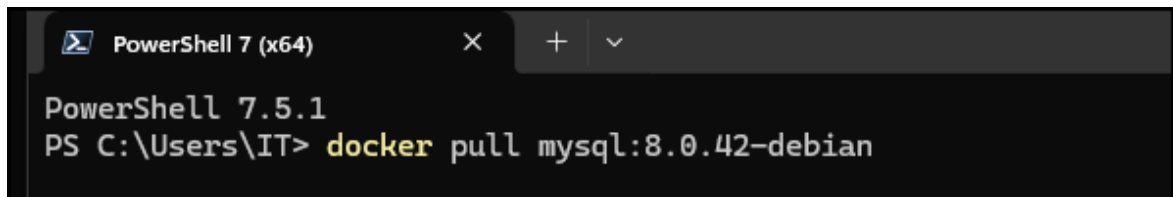
STT	Thành viên	Nhiệm vụ
1	Trần Thành Thoại	Em thực hiện hết toàn bộ sản phẩm của đồ án. Ban đầu có thêm một bạn, Tên Khánh cùng làm với em tuy nhiên tới gần cuối môn học, em không liên lạc được với bạn đó. Nên đồ án em thực hiện một mình.

PHỤ LỤC DEMO

Chạy sản phẩm các bước sau

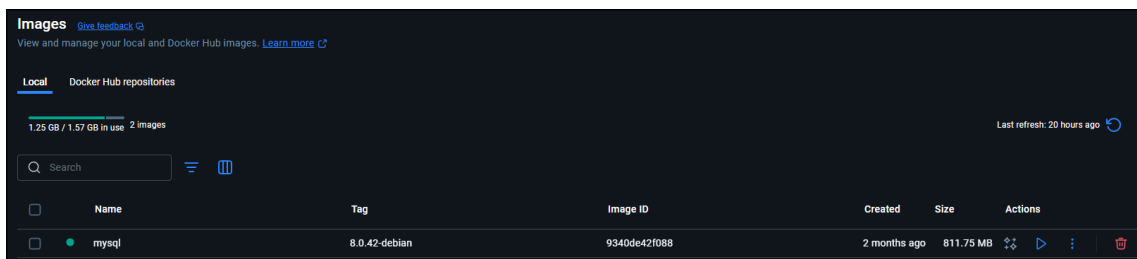
Bước 1: Thiết lập MySQL trên Docker (Vì tập sử dụng docker và quản lý tài nguyên máy dễ hơn)

- **Mở PowerShell hoặc Command line, để cài MySQL 8.0.42-debian:** `docker pull mysql:8.0.42-debian`



Hình 3. Cài MySQL

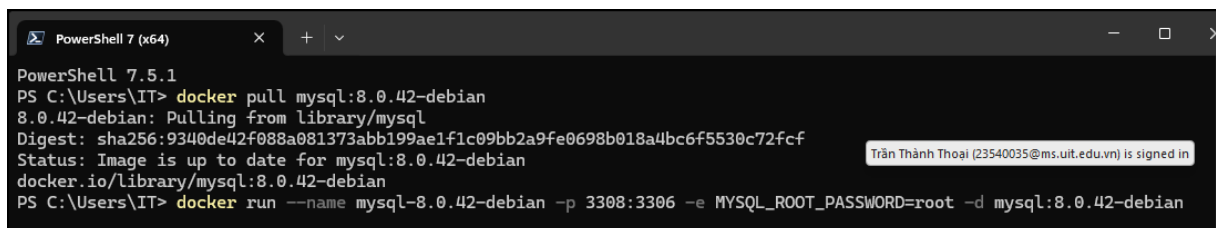
- Nếu cài thành công sẽ thấy MySQL trong Docker desktop



Hình 4. Kiểm tra MySQL

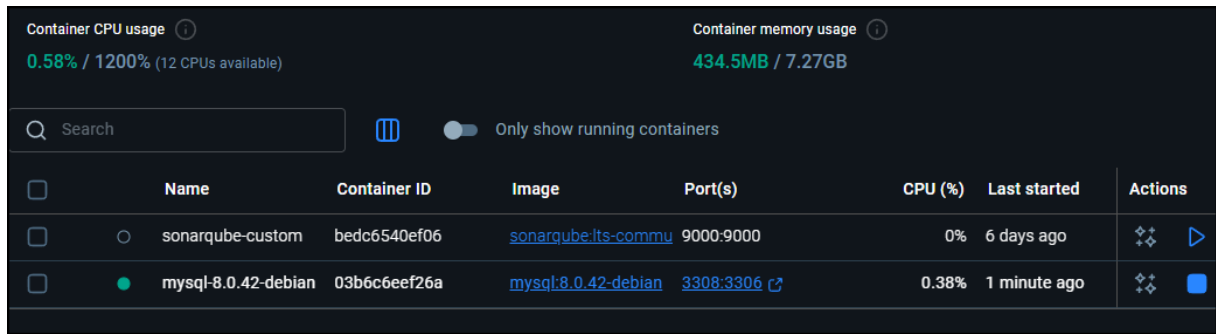
Bước 2: Chạy image, thiết lập port 3308, với password là root:

`docker run --name mysql-8.0.42-debian -p 3308:3306 -e
MYSQL_ROOT_PASSWORD=root -d mysql:8.0.42-debian`



Hình 5. Run image MySQL

- Bấm start

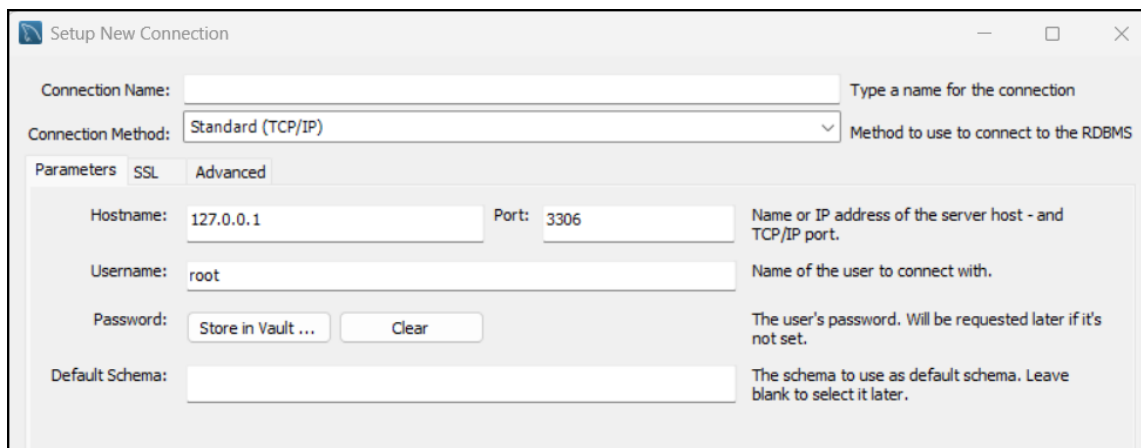


The screenshot shows the Docker Desktop interface. At the top, it displays 'Container CPU usage' as 0.58% / 1200% (12 CPUs available) and 'Container memory usage' as 434.5MB / 7.27GB. Below this is a search bar and a toggle for 'Only show running containers'. A table lists the containers:

	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	sonarqube-custom	bedc6540ef06	sonarqube:its-commu	9000:9000	0%	6 days ago	
<input checked="" type="checkbox"/>	mysql-8.0.42-debian	03b6c6eef26a	mysql:8.0.42-debian	3308:3306	0.38%	1 minute ago	

Hình 6. Start container.

Bước 3: Thiết lập theo hình sau trong MySQL workbench.



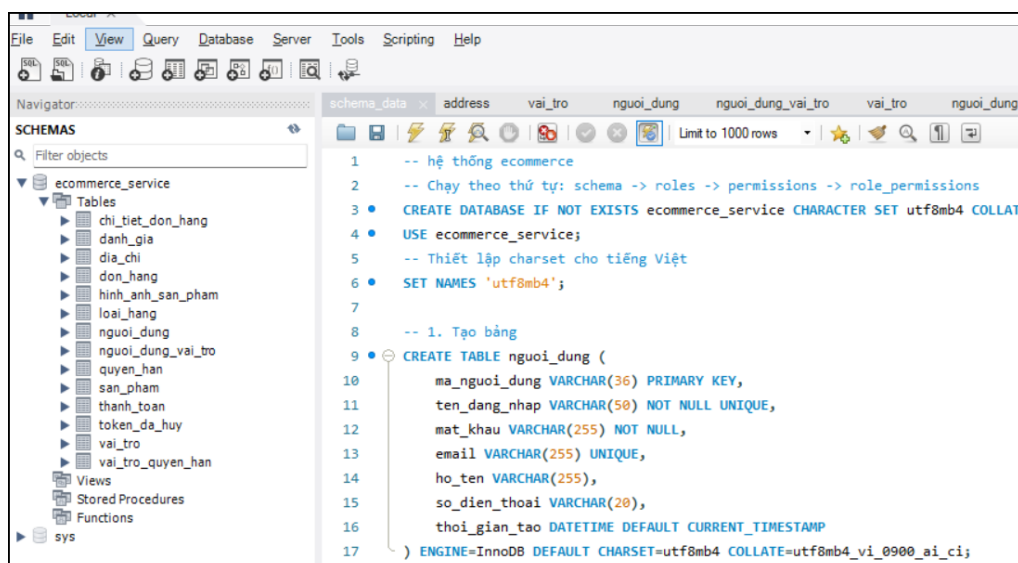
The screenshot shows the 'Setup New Connection' dialog in MySQL Workbench. The 'Connection Name' field is empty. The 'Connection Method' is set to 'Standard (TCP/IP)'. The 'Parameters' tab is selected, showing the following fields:

- Hostname: 127.0.0.1
- Port: 3306
- Username: root
- Password: (with 'Store in Vault...' and 'Clear' buttons)
- Default Schema: (empty)

Each field has a descriptive tooltip on the right.

Hình 7. Thiết lập connection trong MySQL workbench

Bước 4: Run file schema_data.sql

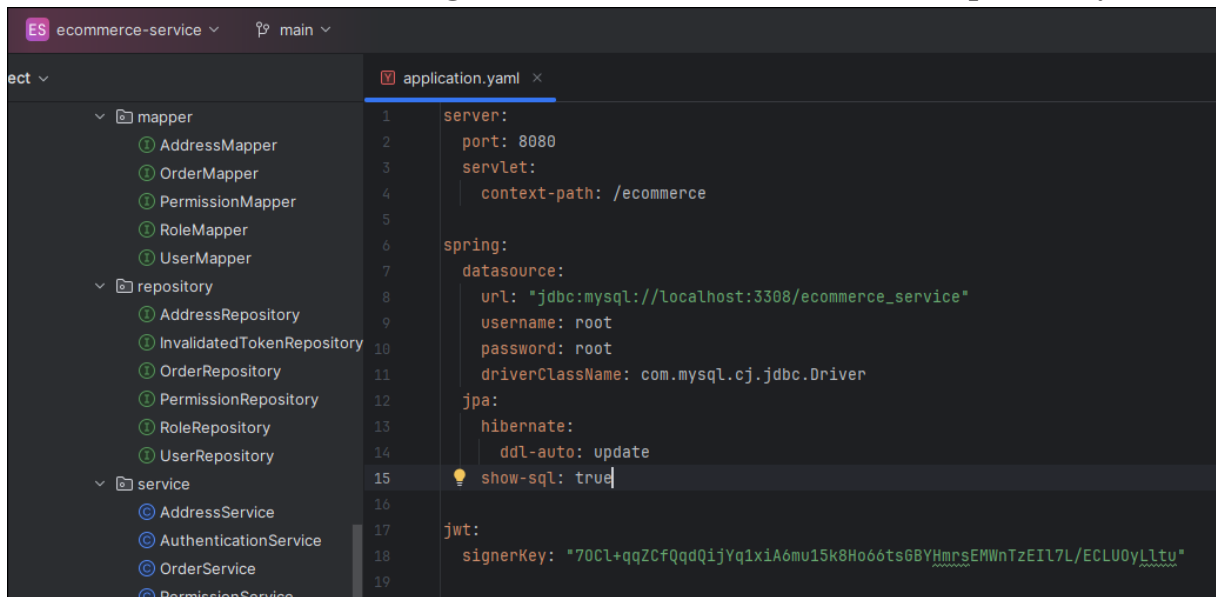


The screenshot shows the MySQL Workbench interface with a SQL script being executed. The 'Navigator' pane on the left shows the 'ecommerce_service' database selected. The 'Query' pane on the right shows the following SQL script:

```
1 -- hệ thống ecommerce
2 -- Chạy theo thứ tự: schema -> roles -> permissions -> role_permissions
3 • CREATE DATABASE IF NOT EXISTS ecommerce_service CHARACTER SET utf8mb4 COLLATE utf8mb4_vietnamese_ci;
4 • USE ecommerce_service;
5 -- Thiết lập charset cho tiếng Việt
6 • SET NAMES 'utf8mb4';
7
8 -- 1. Tạo bảng
9 • CREATE TABLE người_dùng (
10     ma_người_dùng VARCHAR(36) PRIMARY KEY,
11     ten_dang_nhap VARCHAR(50) NOT NULL UNIQUE,
12     mat_khau VARCHAR(255) NOT NULL,
13     email VARCHAR(255) UNIQUE,
14     ho_ten VARCHAR(255),
15     so_dien_thoai VARCHAR(20),
16     thoi_gian_tao DATETIME DEFAULT CURRENT_TIMESTAMP
17 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_vietnamese_ci;
```

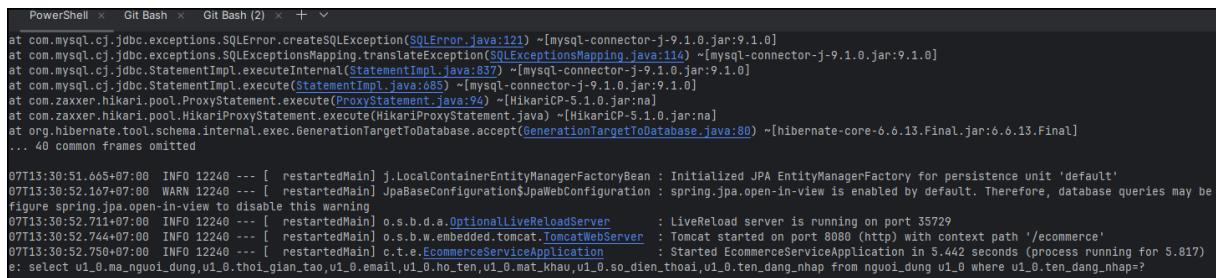
Hình 8. Tạo database.

Bước 5: Mở source code bằng IDE IntelliJ IDEA. Có thể thiết lập lại file yaml.



Hình 9. Mở code bằng IDE

Bước 6: Run bằng lệnh mvn clean spring-boot:run (Nếu có cài Maven) hoặc Run bình thường bằng IDE. Khi khởi tạo ban đầu sẽ tự tạo username admin với password admin, có thể tùy chỉnh trong code.



Hình 10. Run code.

Bước 7: Test các API bằng Postman tương tự trong Video Test API.

PHỤ LỤC MÔ TẢ CÁC BẢNG KHÁC CỦA CƠ SỞ DỮ LIỆU

1. Thực thể đơn hàng

- Các thuộc tính:
 - + `ma_don_hang` (Khóa chính): Mã định danh đơn hàng.
 - + `ma_nguoi_dung` (Khóa ngoại): Người đặt hàng.
 - + `ma_dia_chi` (Khóa ngoại): Địa chỉ giao hàng.
 - + `tong_tien`: Tổng số tiền của đơn hàng. Tính thông qua bảng chi tiết đơn hàng.
 - + `trang_thai`: Trạng thái đơn hàng (Chờ xử lý: PENDING; Đang xử lý: PROCESSING; Đã gửi: SHIPPED; Đã giao: DELIVERED; Đã hủy: CANCELLED).
 - + `thoi_gian_tao`: Thời điểm tạo đơn hàng.
- Mối quan hệ:
 - + n-1 với `nguoi_dung`: Mỗi đơn hàng thuộc về một người dùng.
 - + n-1 với `dia_chi`: Mỗi đơn hàng sử dụng một địa chỉ giao hàng.
 - + 1-n với `chi_tiet_don_hang`: Một đơn hàng có thể có nhiều chi tiết đơn hàng.
 - + 1-1 với `thanh_toan`: Mỗi đơn hàng có một thông tin thanh toán.

Bảng 10. `don_hang`

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	<code>ma_don_hang</code>	VARCHAR(36)	PRIMARY KEY
2	<code>ma_nguoi_dung</code>	VARCHAR(36)	FOREIGN KEY
3	<code>ma_dia_chi</code>	VARCHAR(36)	FOREIGN KEY
4	<code>tong_tien</code>	DECIMAL(10,2)	NOT NULL
5	<code>trang_thai</code>	ENUM('PENDING', 'PROCESSING', 'SHIPPED', 'DELIVERED', 'CANCELLED')	DEFAULT 'PENDING'
6	<code>thoi_gian_tao</code>	DATETIME	DEFAULT CURRENT_TIMESTAMP

2. Thực thể sản phẩm

- **Chức năng:** Lưu trữ thông tin các sản phẩm được bán trên website.
- Các thuộc tính:
 - + `ma_san_pham` (Khóa chính): Mã định danh sản phẩm.
 - + `ten_san_pham`: Tên sản phẩm.
 - + `mo_ta_san_pham`: Mô tả chi tiết sản phẩm.
 - + `gia_san_pham`: Giá bán.
 - + `so_luong_ton`: Số lượng tồn kho.
 - + `ma_loai_hang` (Khóa ngoại): Tham chiếu đến loại hàng.
 - + `ngay_tao`: Thời điểm thêm sản phẩm vào hệ thống.
- Mối quan hệ:
 - + n-1 với `loai_hang`: Mỗi sản phẩm thuộc một loại hàng.
 - + 1-n với `hinh_anh_san_pham`: Một sản phẩm có thể có nhiều hình ảnh.
 - + 1-n với `chi_tiet_don_hang`: Một sản phẩm có thể xuất hiện trong nhiều chi tiết đơn hàng.
 - + 1-n với `danh_gia`: Một sản phẩm có thể có nhiều đánh giá.

Bảng 11. `san_pham`

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	<code>ma_san_pham</code>	VARCHAR(36)	PRIMARY KEY
2	<code>ten_san_pham</code>	VARCHAR(255)	NOT NULL
3	<code>mo_ta_san_pham</code>	TEXT	NOT NULL
4	<code>gia_san_pham</code>	DECIMAL(10, 2)	NOT NULL
5	<code>so_luong_ton</code>	INT	NOT NULL
6	<code>ma_loai_hang</code>	VARCHAR(36)	FOREIGN KEY
7	<code>ngay_tao</code>	DATETIME	DEFAULT CURRENT_TIMESTAMP

3. Thực thể hình ảnh sản phẩm

- Chức năng: Lưu trữ các hình ảnh liên quan đến sản phẩm.
- Các thuộc tính:
 - + `ma_hinh_anh` (Khóa chính): Mã định danh hình ảnh.
 - + `ma_san_pham` (Khóa ngoại): Tham chiếu đến sản phẩm.
- Mỗi quan hệ:
 - + n-1 với `san_pham`: Mỗi hình ảnh thuộc về một sản phẩm.

Bảng 12. `hinh_anh_san_pham`

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	<code>ma_hinh_anh</code>	VARCHAR(36)	PRIMARY KEY
2	<code>ma_san_pham</code>	VARCHAR(36)	FOREIGN KEY

4. Thực thể loại hàng

- Chức năng: Lưu trữ các loại hàng hóa, phân loại sản phẩm.
- Các thuộc tính:
 - + `ma_loai_hang` (Khóa chính): Mã định danh loại hàng.
 - + `ten_loai_hang`: Tên loại hàng.
- Mỗi quan hệ:
 - + 1-n với `san_pham`: Một loại hàng có thể có nhiều sản phẩm

Bảng 13. `loai_hang`

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	<code>ma_loai_hang</code>	VARCHAR(36)	PRIMARY KEY
2	<code>ten_loai_hang</code>	VARCHAR(255)	NOT NULL

5. Thực thể chi tiết đơn hàng

- Chức năng: Lưu trữ chi tiết từng sản phẩm trong mỗi đơn hàng.
- Các thuộc tính:
 - + `ma_ct_don_hang` (Khóa chính): Mã định danh chi tiết đơn hàng.
 - + `ma_don_hang` (Khóa ngoại): Tham chiếu đến đơn hàng.
 - + `ma_san_pham` (Khóa ngoại): Tham chiếu đến sản phẩm.
 - + `so_luong`: Số lượng sản phẩm.
 - + `gia`: Giá tại thời điểm đặt hàng.

- Mỗi quan hệ:
 - + n-1 với don_hang: Mỗi chi tiết đơn hàng thuộc về một đơn hàng.
 - + n-1 với san_pham: Mỗi chi tiết đơn hàng liên kết với một sản phẩm.

Bảng 14. chi_tiet_don_hang

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	ma_ct_don_hang	VARCHAR(36)	PRIMARY KEY
2	ma_don_hang	VARCHAR(36)	FOREIGN KEY
3	ma_san_pham	VARCHAR(36)	FOREIGN KEY
4	so_luong	INT	NOT NULL
5	gia	DECIMAL(10,2)	NOT NULL

6. Thực thể thanh toán

- Chức năng: Lưu trữ thông tin thanh toán cho các đơn hàng.
- Các thuộc tính:
 - + ma_thanh_toan (Khóa chính): Mã định danh thanh toán.
 - + ma_don_hang (Khóa ngoại): Tham chiếu đến đơn hàng.
 - + phuong_thuc: Phương thức thanh toán (chuyển khoản, thẻ tín dụng, v.v.).
 - + trang_thai: Trạng thái thanh toán: Đang xử lý, hoàn tất, thất bại (PENDING, COMPLETED, FAILED)
 - + ngay_thanh_toan: Thời điểm thanh toán.
- Mỗi quan hệ:
 - + 1-1 với don_hang: Mỗi thanh toán gắn với một đơn hàng.

Bảng 15. thanh_toan

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	ma_thanh_toan	VARCHAR(36)	PRIMARY KEY
2	ma_don_hang	VARCHAR(36)	FOREIGN KEY
3	phuong_thuc	VARCHAR(50)	NOT NULL
4	trang_thai	ENUM('PENDING', 'COMPLETED', 'FAILED')	DEFAULT 'PENDING'
5	ngay_thanh_toan	DATETIME	

7. Thực thể đánh giá

- Chức năng: Lưu trữ các đánh giá, nhận xét của người dùng về sản phẩm.
- Các thuộc tính:
 - + `ma_danh_gia` (Khóa chính): Mã định danh đánh giá.
 - + `ma_nguoi_dung` (Khóa ngoại): Người đánh giá.
 - + `ma_san_pham` (Khóa ngoại): Sản phẩm được đánh giá.
 - + `diem_danh_gia`: Điểm đánh giá (1-5).
 - + `binh_luan`: Nhận xét, bình luận.
 - + `ngay_tao`: Thời điểm đánh giá.
- Mối quan hệ:
 - + n-1 với `nguoi_dung`: Mỗi đánh giá do một người dùng thực hiện.
 - + n-1 với `san_pham`: Mỗi đánh giá dành cho một sản phẩm.

Bảng 16. danh_gia

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	<code>ma_danh_gia</code>	VARCHAR(36)	PRIMARY KEY
2	<code>ma_nguoi_dung</code>	VARCHAR(36)	FOREIGN KEY
3	<code>ma_san_pham</code>	VARCHAR(36)	FOREIGN KEY
4	<code>diem_danh_gia</code>	INT	(<code>diem_danh_gia</code> BETWEEN 1 AND 5)
5	<code>binh_luan</code>	TEXT	
6	<code>ngay_tao</code>	DATETIME	DEFAULT CURRENT_TIMESTAMP

PHỤ LỤC HÌNH

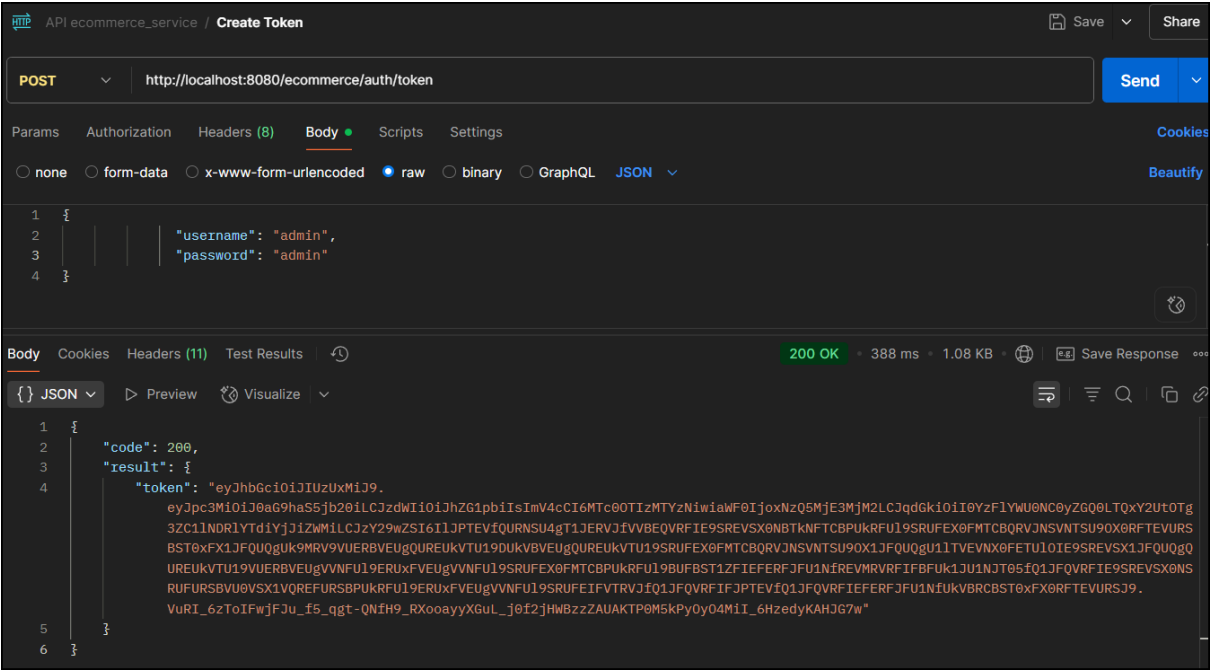
Hình 1. Mô hình ERD của cơ sở dữ liệu ecommerce_service.	2
Hình 2. Mô hình RABC.....	7
Hình 3. Cài MySQL	2
Hình 4. Kiểm tra MySQL.....	2
Hình 5. Run image MySQL.....	2
Hình 6. Start container.....	3
Hình 7. Thiết lập connection trong MySQL workbench.....	3
Hình 8. Tạo database.	3
Hình 9. Mở code bằng IDE	4
Hình 10. Run code.	4
Hình 11. Đăng nhập với admin	12
Hình 12. Đăng nhập với tài khoản thường (Role User)	13
Hình 13. Đăng nhập sai mật khẩu	14
Hình 14. Đăng nhập với username không tồn tại.....	14
Hình 15. Kiểm tra token - Introspect.....	15
Hình 16. Đăng xuất - Logout.....	16
Hình 17. Tạo người dùng - Create User (User thường, không cần đăng nhập)	17
Hình 18. Tạo admin với tài khoản admin (Chỉ có admin mới có quyền thêm role vào, còn lại mặc định sẽ thêm gán role User)	18
Hình 19. Tạo user với username đã tồn tại.....	19
Hình 20. Tạo user với email đã tồn tại	20
Hình 21. Tạo user với dữ liệu không hợp lệ - username quá ngắn	21
Hình 22. Tạo user với mật khẩu quá ngắn.....	22
Hình 23. Tạo user với email không hợp lệ	23
Hình 24. Tạo user với số điện thoại không hợp lệ	24
Hình 25. Lấy danh sách người dùng bằng tài khoản admin.....	25
Hình 26. Không lấy được danh sách người dùng bằng tài khoản user thường	26
Hình 27. Lấy thông tin người dùng theo ID (chỉ admin hoặc chính user đó)	27
Hình 28. Lấy User không tồn tại	28

Hình 29. Lấy thông tin của chính mình.....	29
Hình 30. Cập nhật người dùng (Đối với Role thì chỉ có admin mới được cập nhật)	30
Hình 31. Cập nhật người dùng và Role	31
Hình 32. Xóa người dùng với role admin.....	32
Hình 33. Tạo quyền hạn	33
Hình 34. Lấy danh sách tất cả quyền hạn (chỉ admin)	34
Hình 35. Xóa quyền hạn (chỉ admin)	35
Hình 36. Tạo vai trò mới (chỉ admin).....	36
Hình 37. Lấy danh sách vai trò.....	37
Hình 38. Cập nhật vai trò	38
Hình 39. Xóa vai trò.	39
Hình 40. Tạo địa chỉ	40
Hình 41. Lấy danh sách địa chỉ.	41
Hình 42. Lấy địa chỉ theo UserId (Chỉ có admin).....	42
Hình 43. Lấy địa chỉ của chính user	43
Hình 44. Xóa địa chỉ.....	44
Hình 45. Tạo đơn hàng.	44
Hình 46. Lấy danh sách đơn hàng.	45
Hình 47. Xem đơn hàng của chính người dùng.....	46
Hình 48. Cập nhật đơn hàng.	46
Hình 49. Xóa đơn hàng.....	47
Hình 50. Hủy đơn hàng	47
Hình 51. Duyệt đơn hàng bởi Shop hoặc admin	48

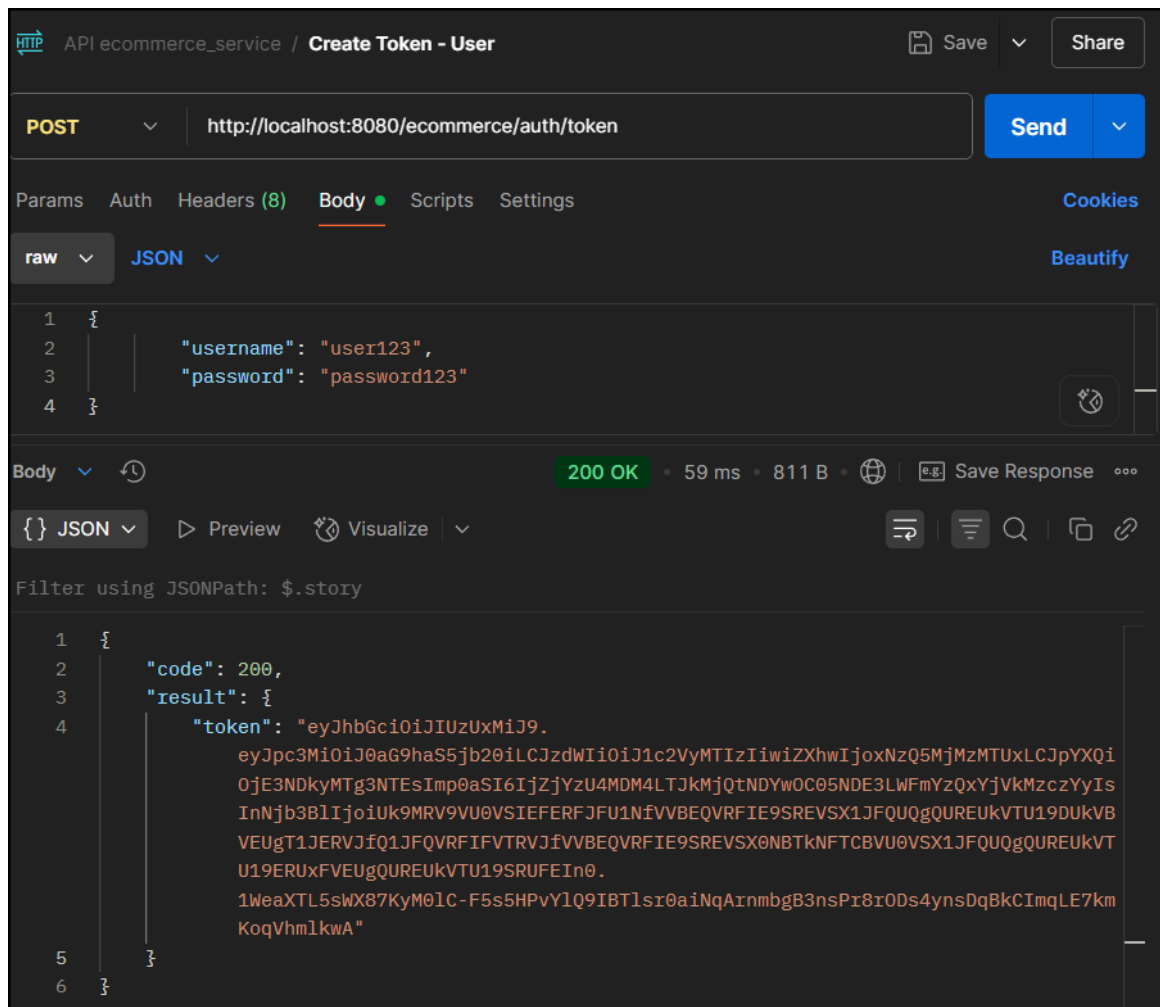
PHỤ LỤC TEST API

1. Test cases cho Authentication API (3 endpoints)

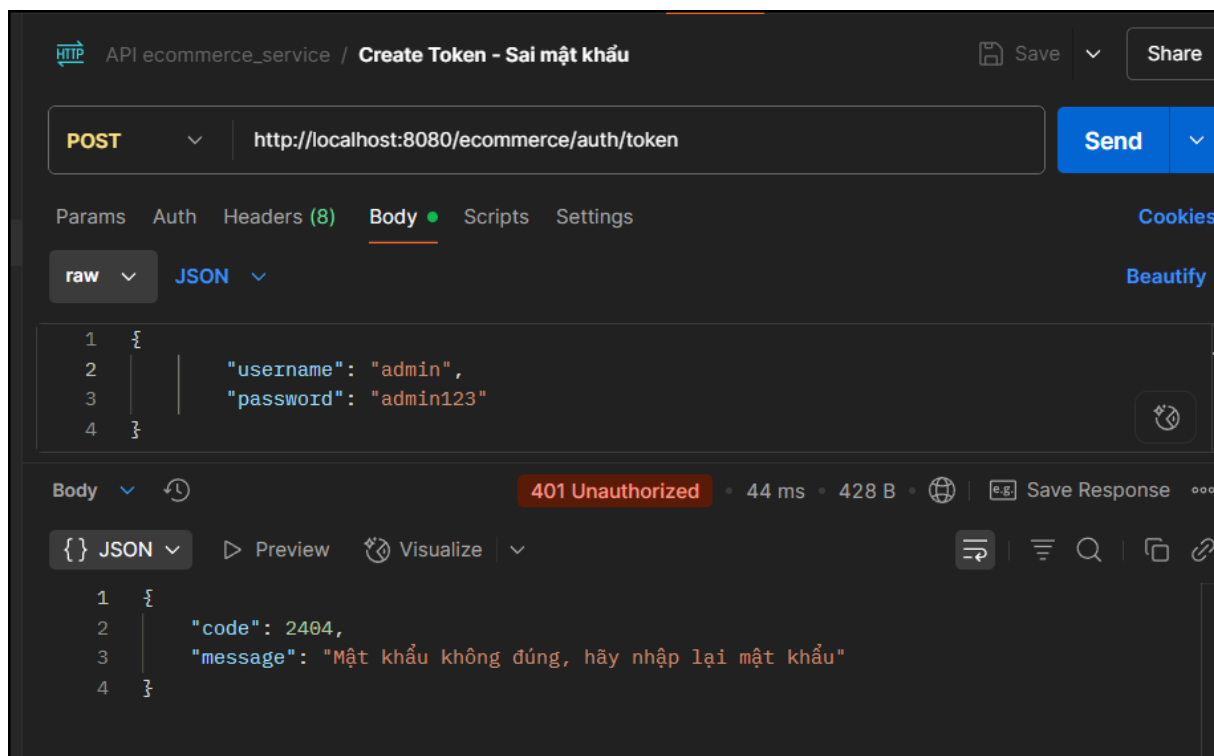
Khi chạy code sẽ kiểm tra database đã có username là admin chưa, nếu chưa có sẽ tự tạo tài khoản admin với mật khẩu “admin”. Khi triển khai thực tế sẽ chỉnh sửa không để mật khẩu trong code.



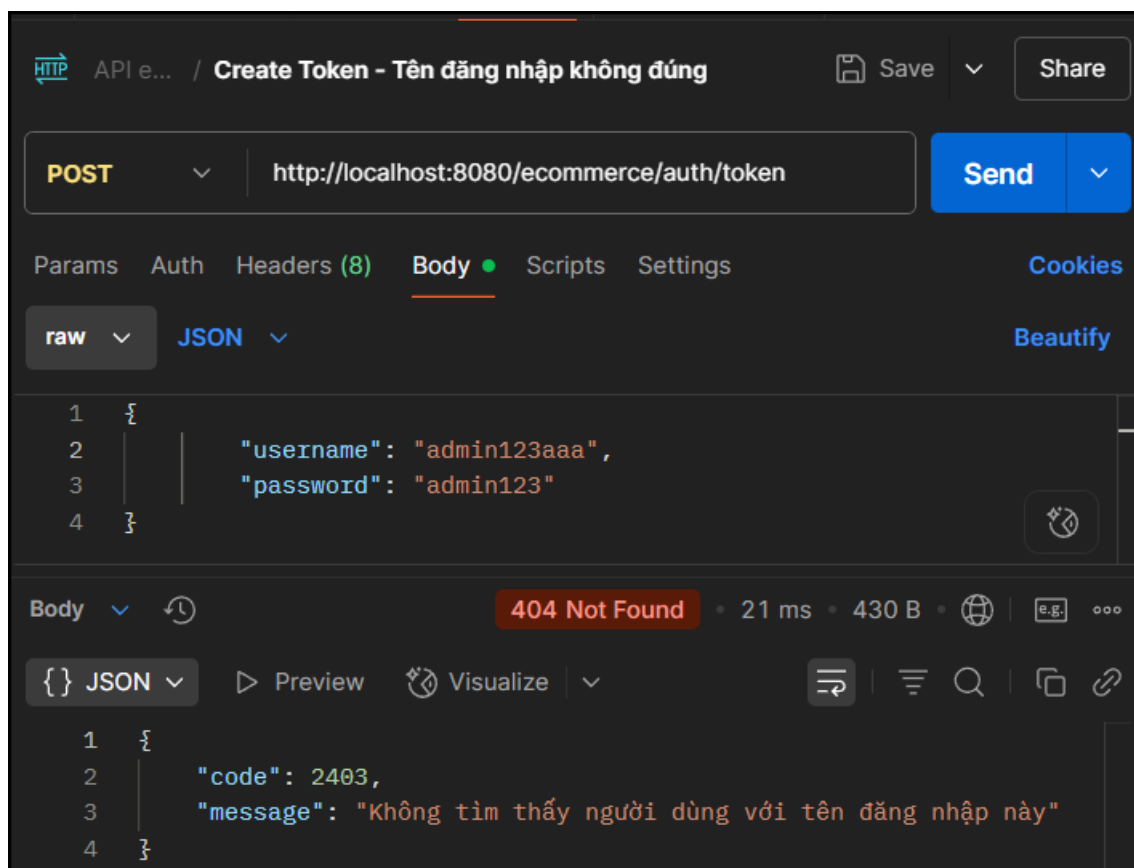
Hình 11. Đăng nhập với admin



Hình 12. Đăng nhập với tài khoản thường (Role User)

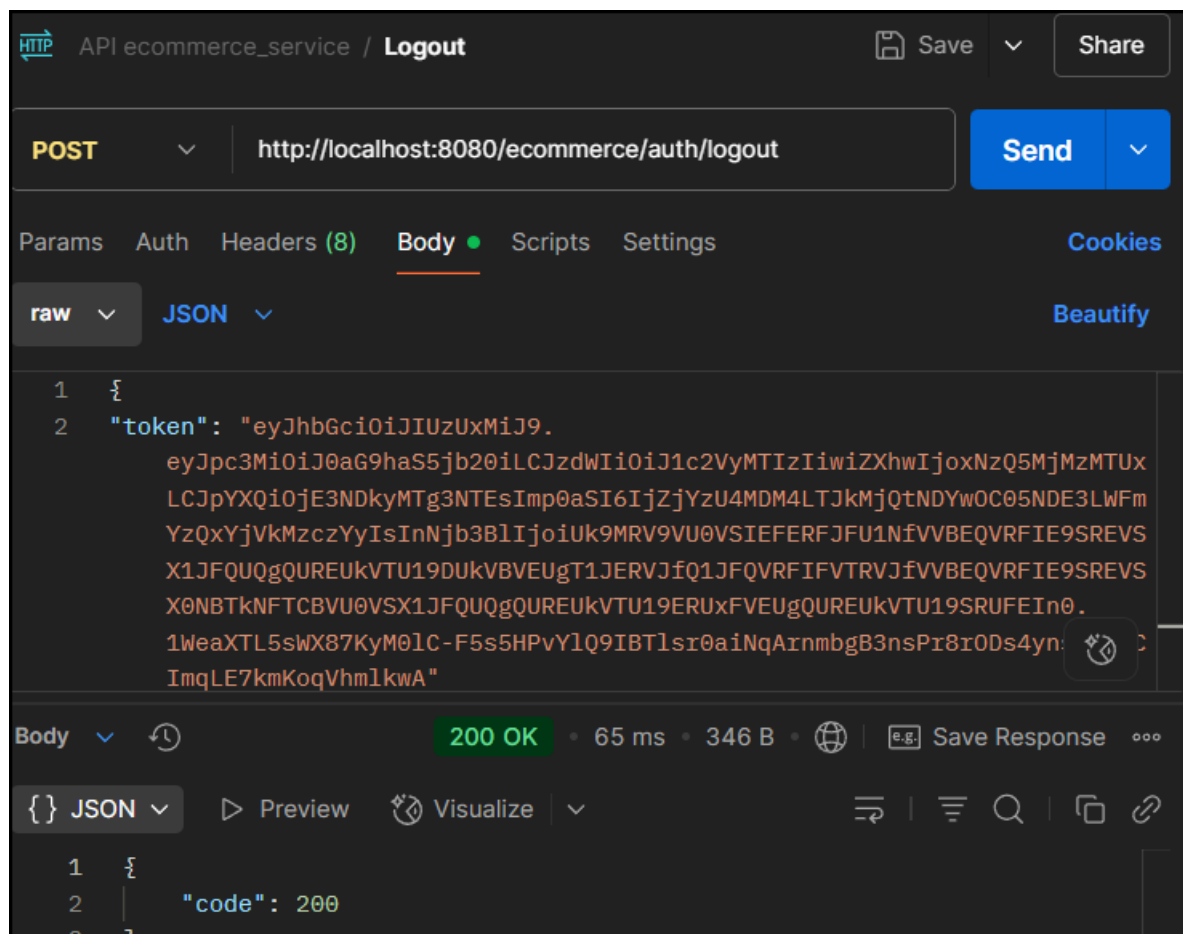


Hình 13. Đăng nhập sai mật khẩu



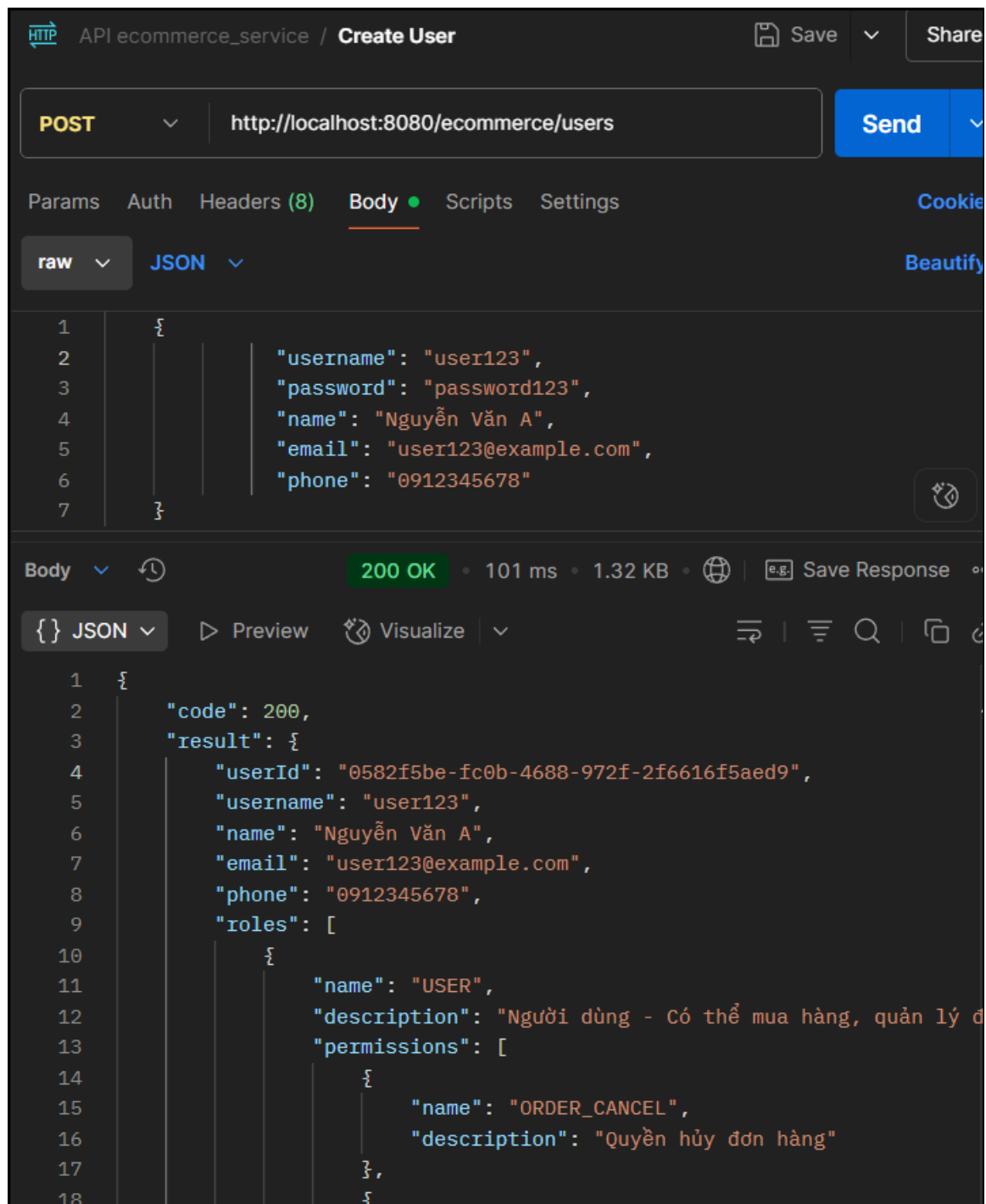
Hình 14. Đăng nhập với username không tồn tại



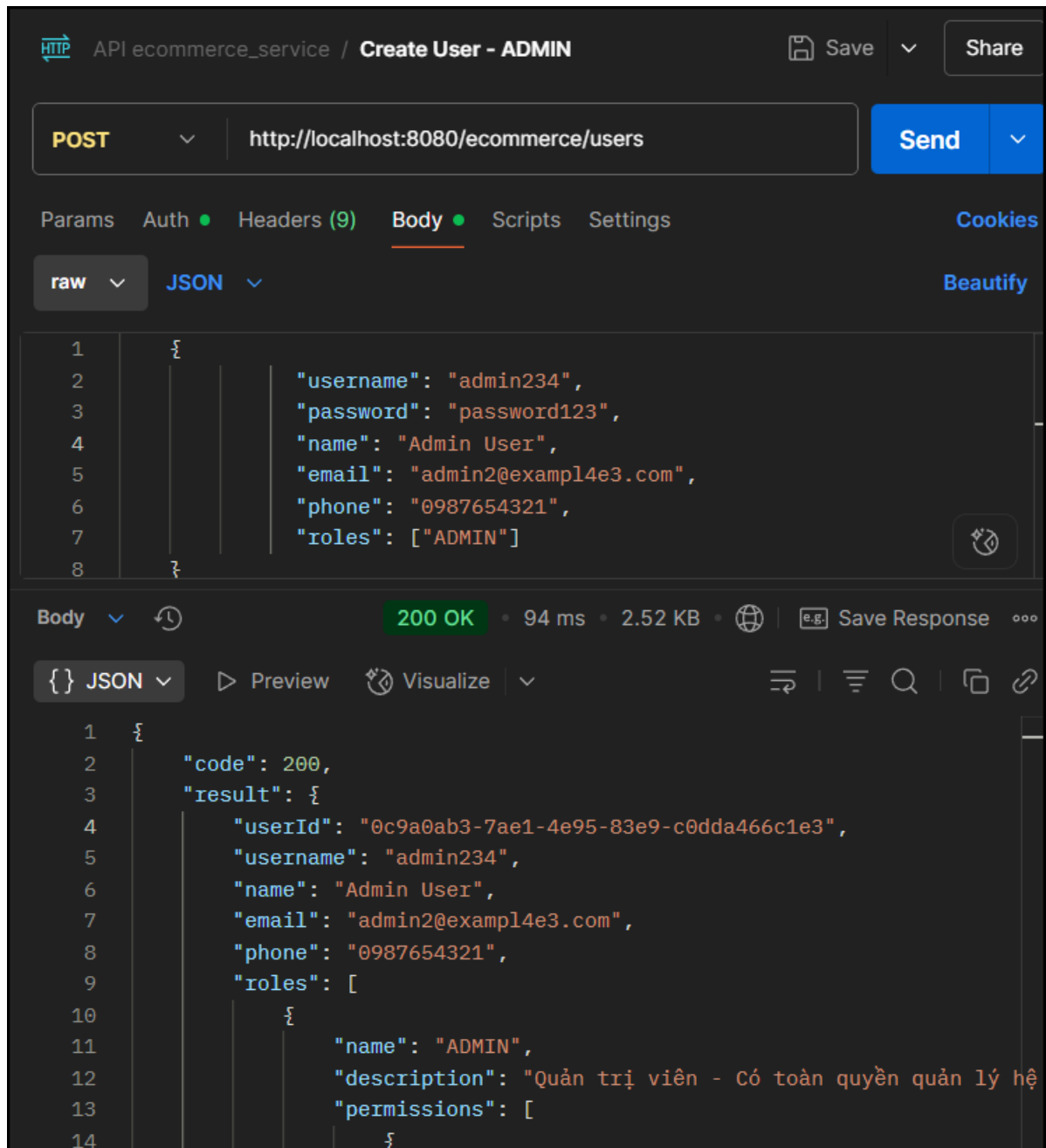


Hình 16. Đăng xuất - Logout

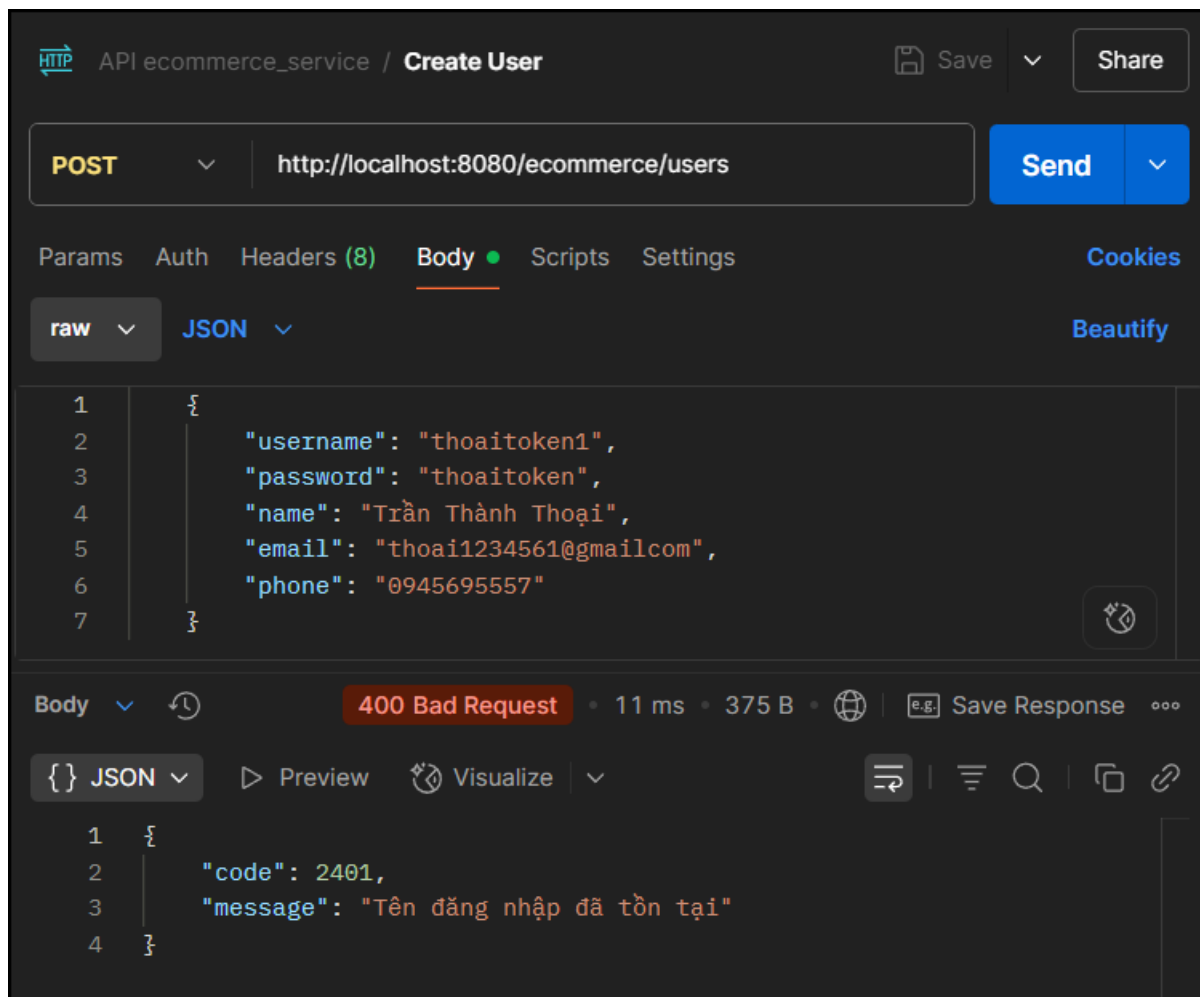
2. Test cases cho User API (6 endpoints)



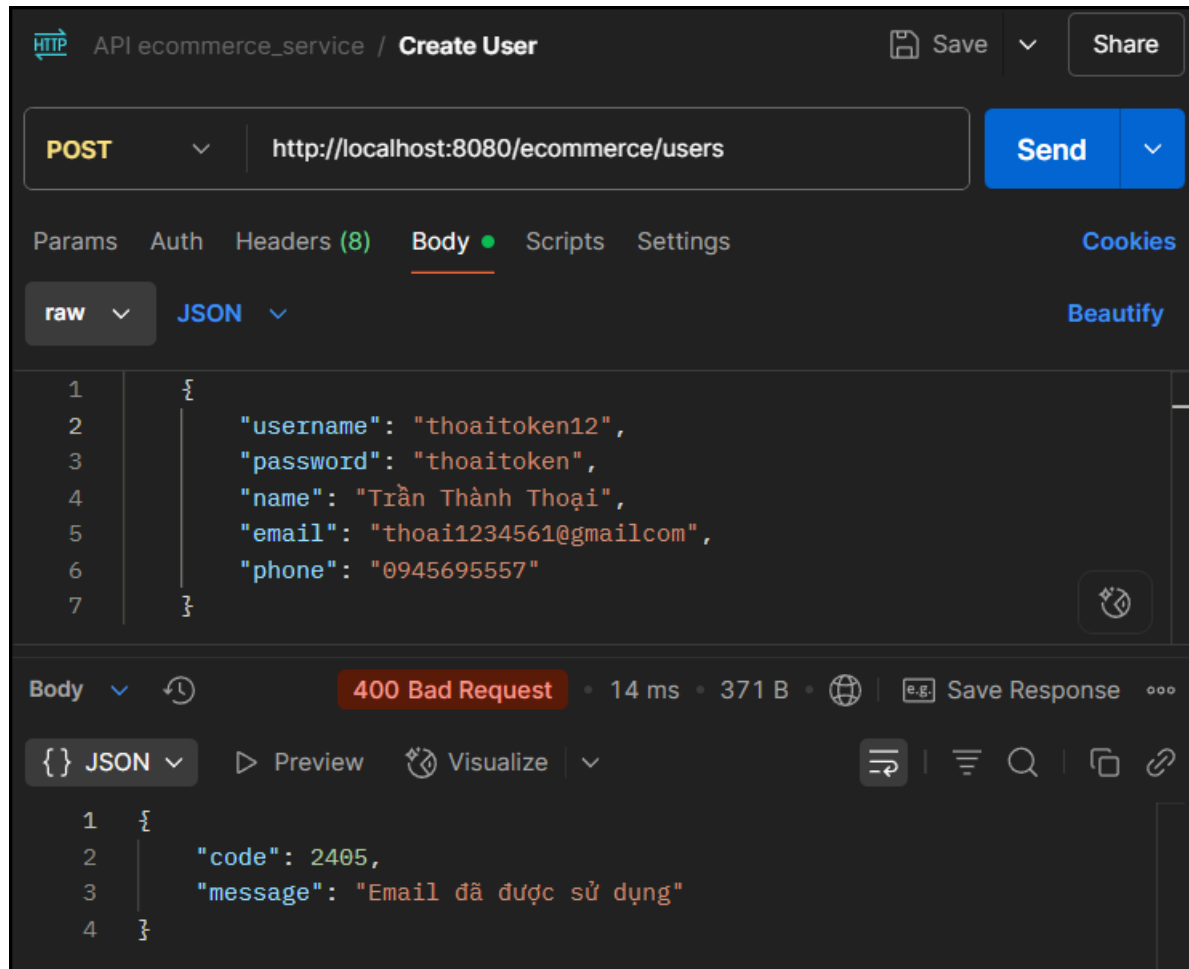
Hình 17. Tạo người dùng - Create User (User thường, không cần đăng nhập)



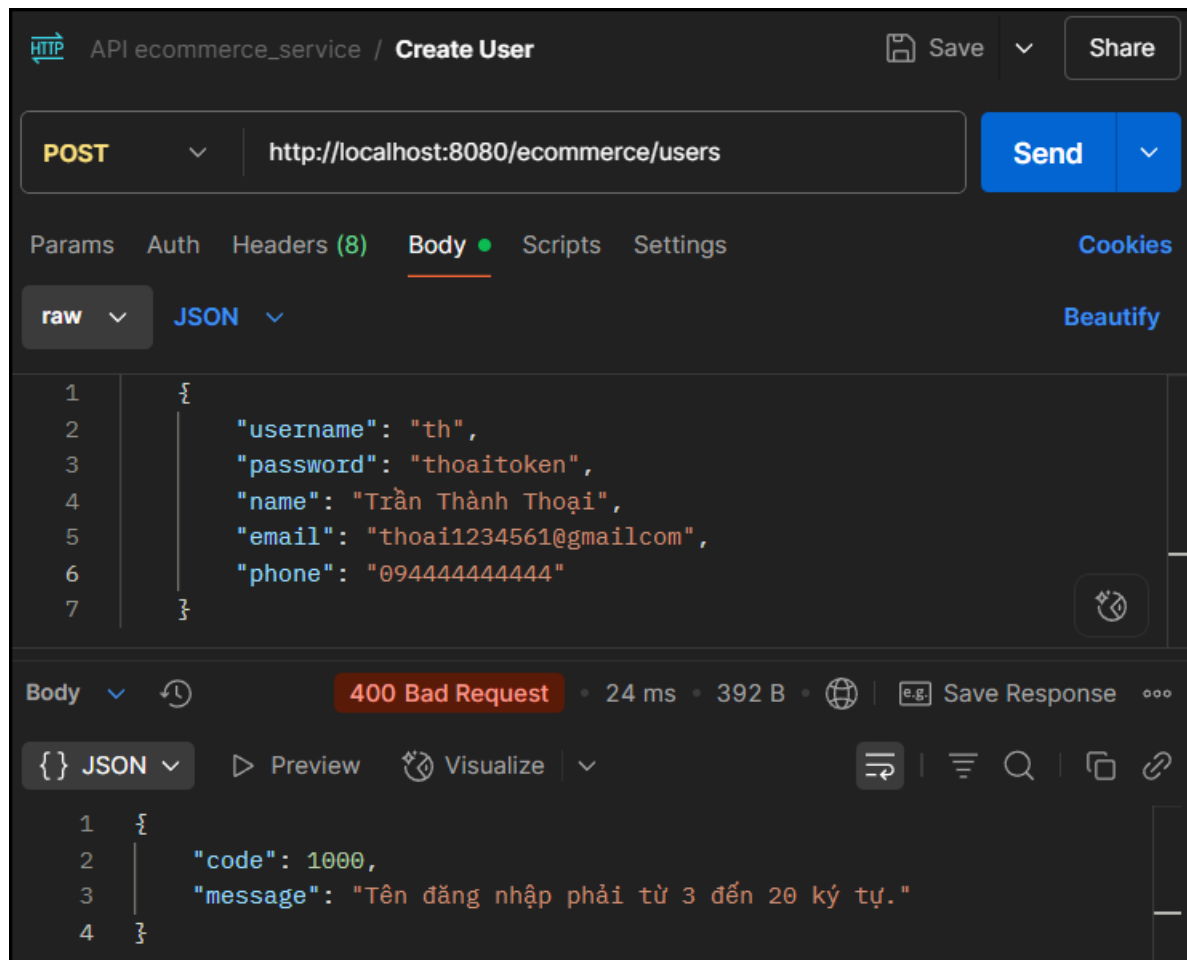
Hình 18. Tạo admin với tài khoản admin (Chỉ có admin mới có quyền thêm role vào, còn lại mặc định sẽ thêm gán role User)



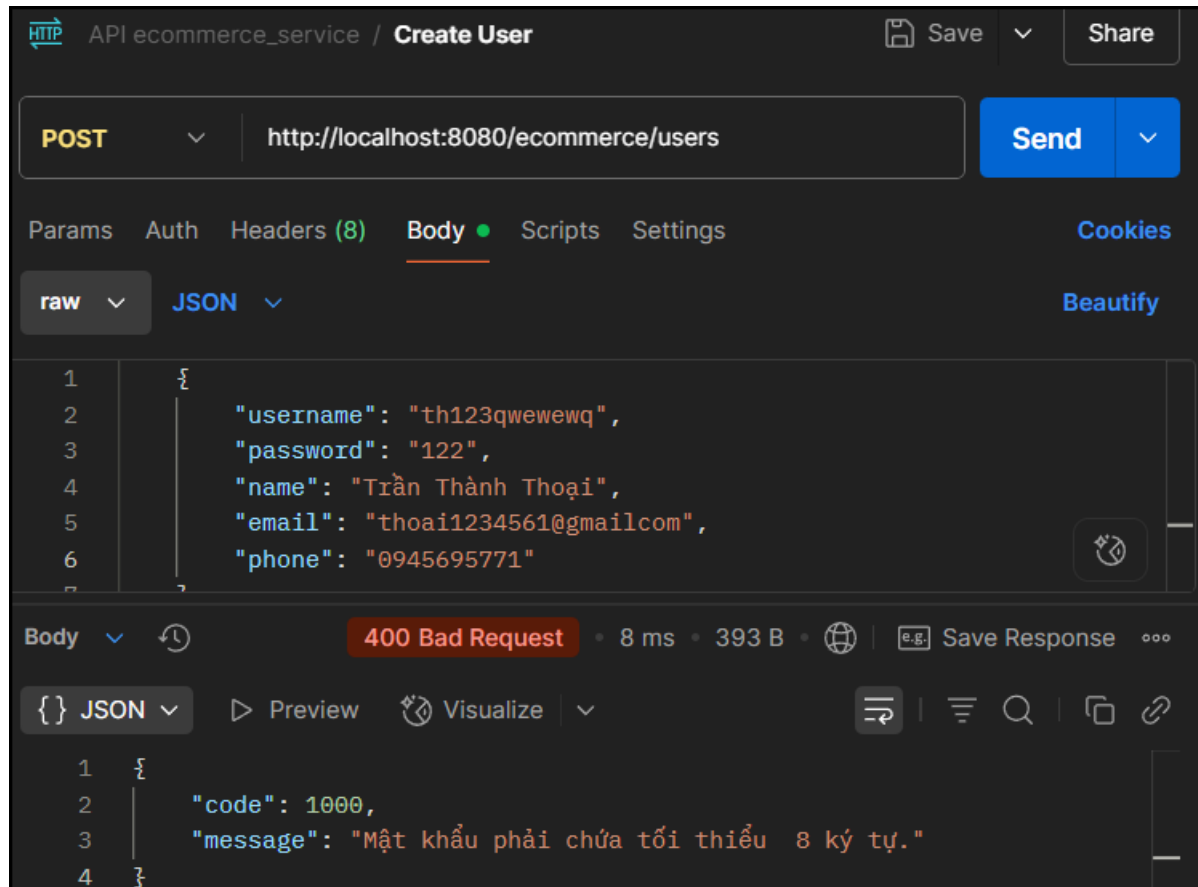
Hình 19. Tạo user với username đã tồn tại.



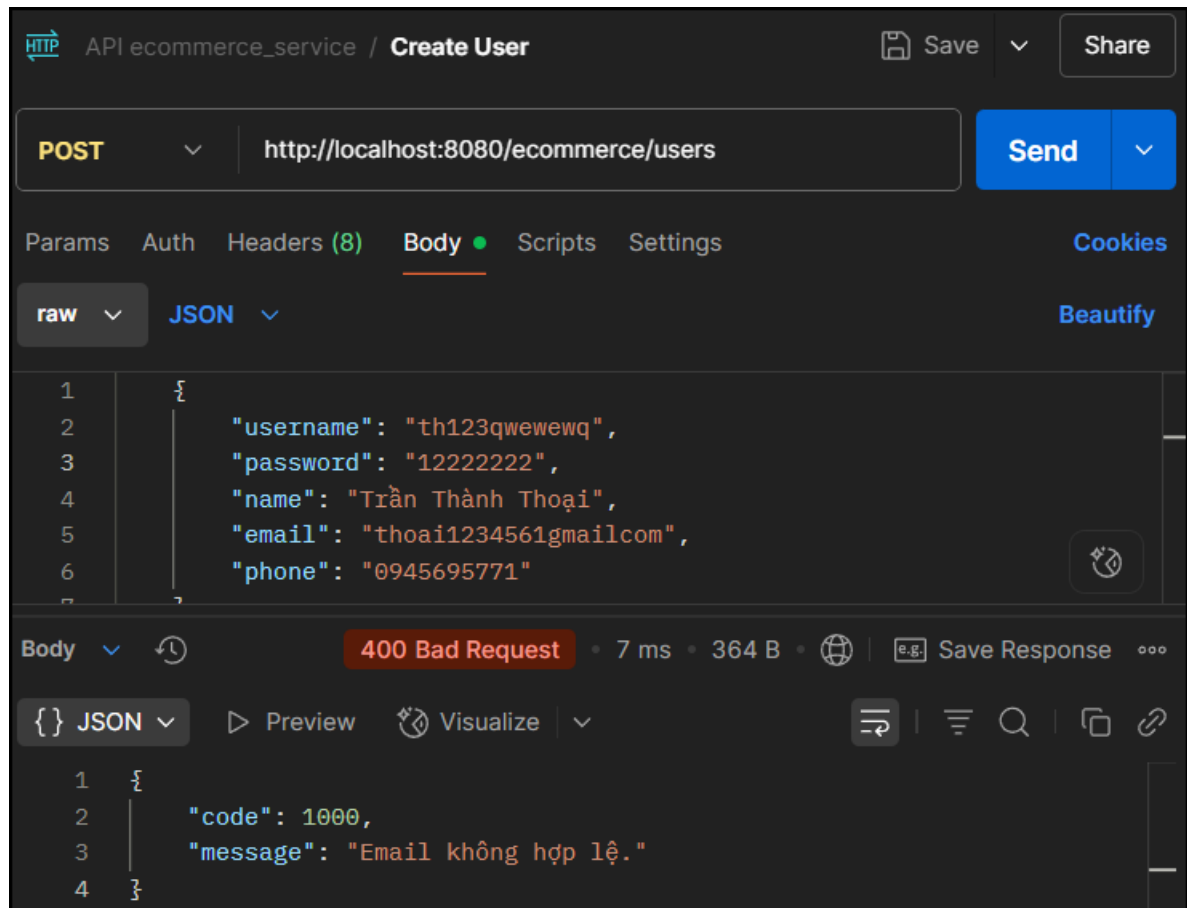
Hình 20. Tạo user với email đã tồn tại



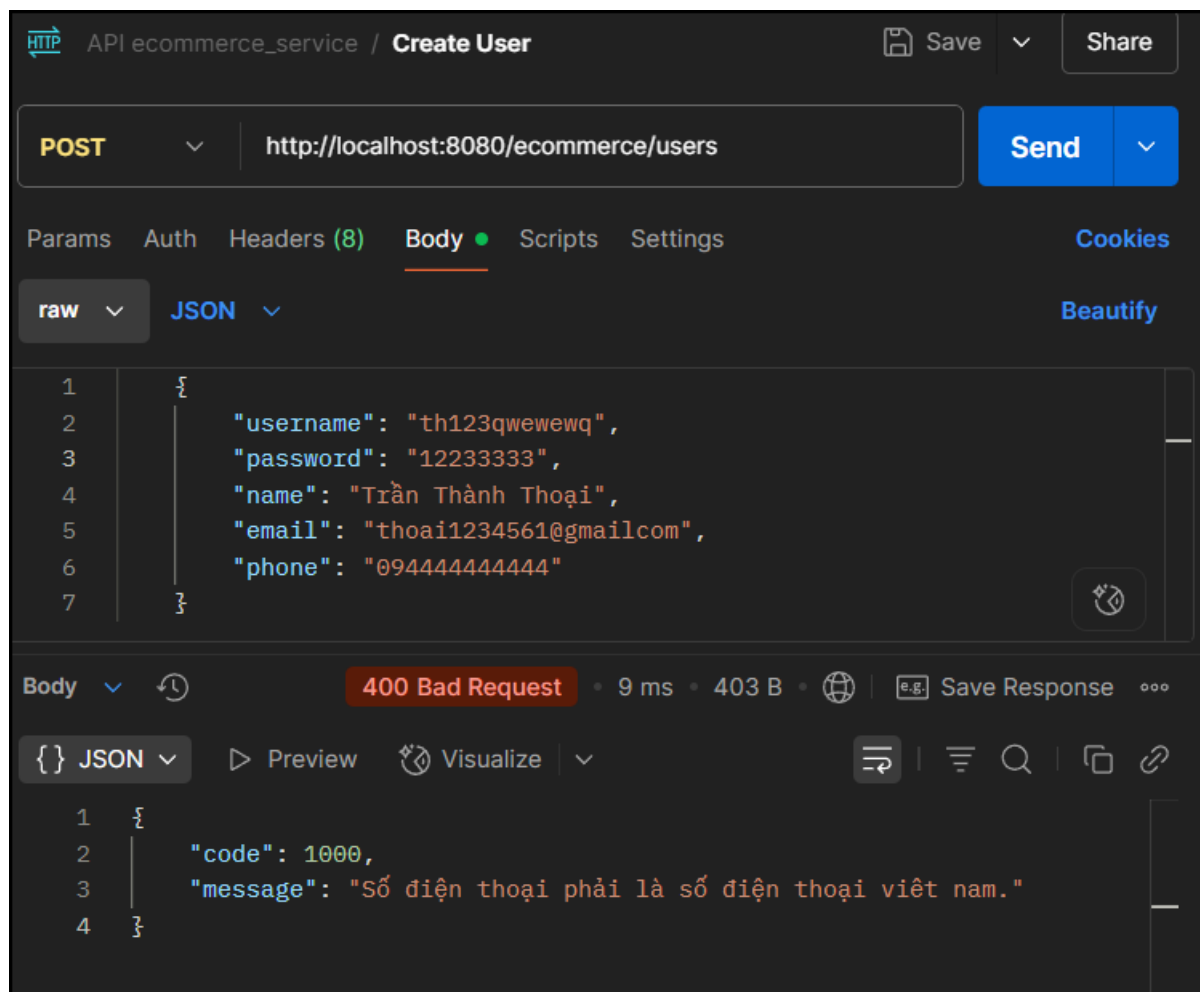
Hình 21. Tạo user với dữ liệu không hợp lệ - username quá ngắn



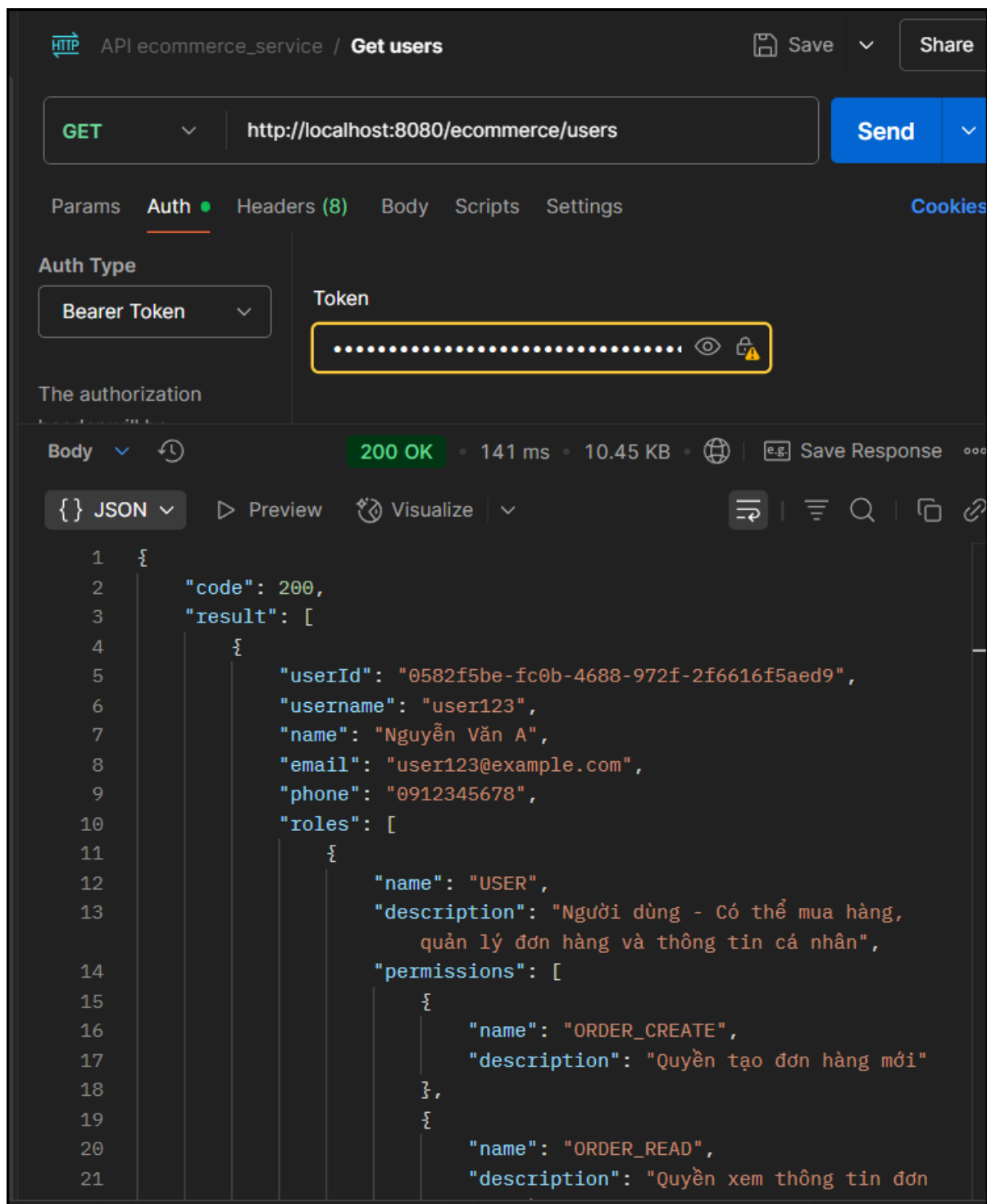
Hình 22. Tạo user với mật khẩu quá ngắn



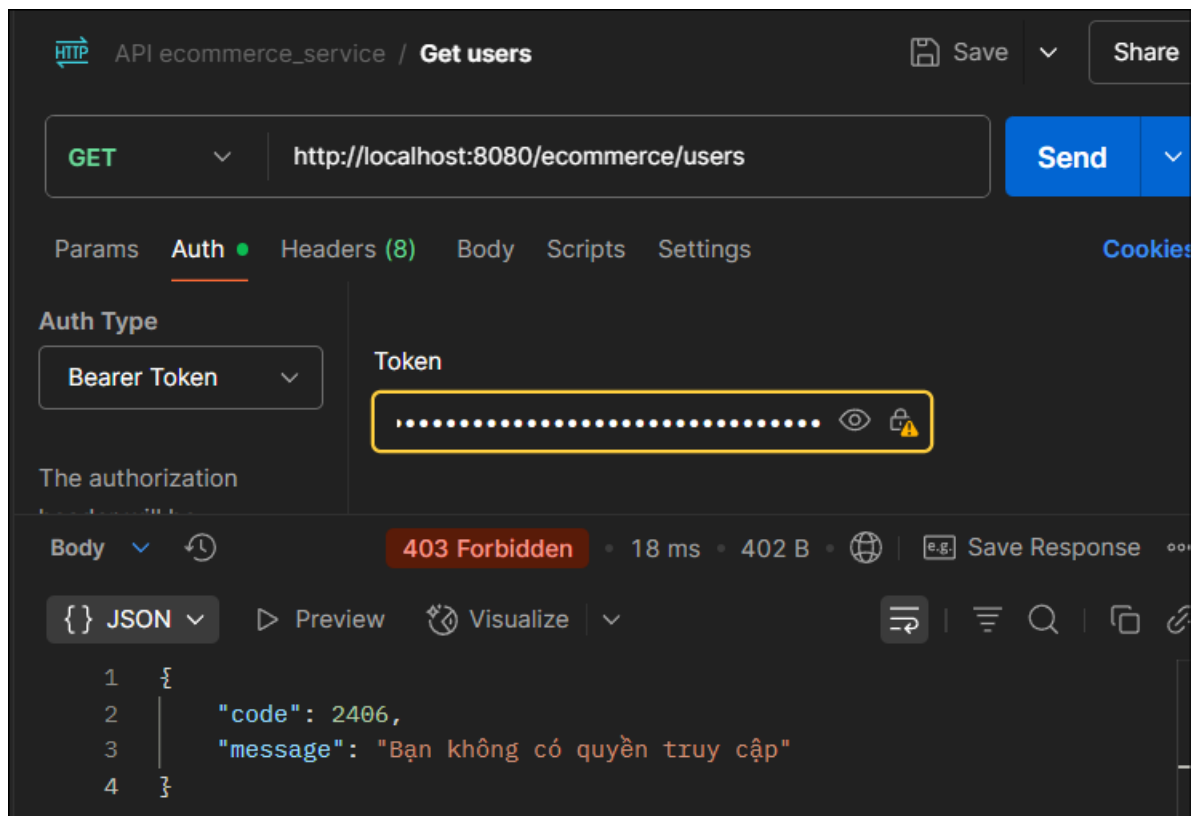
Hình 23. Tạo user với email không hợp lệ



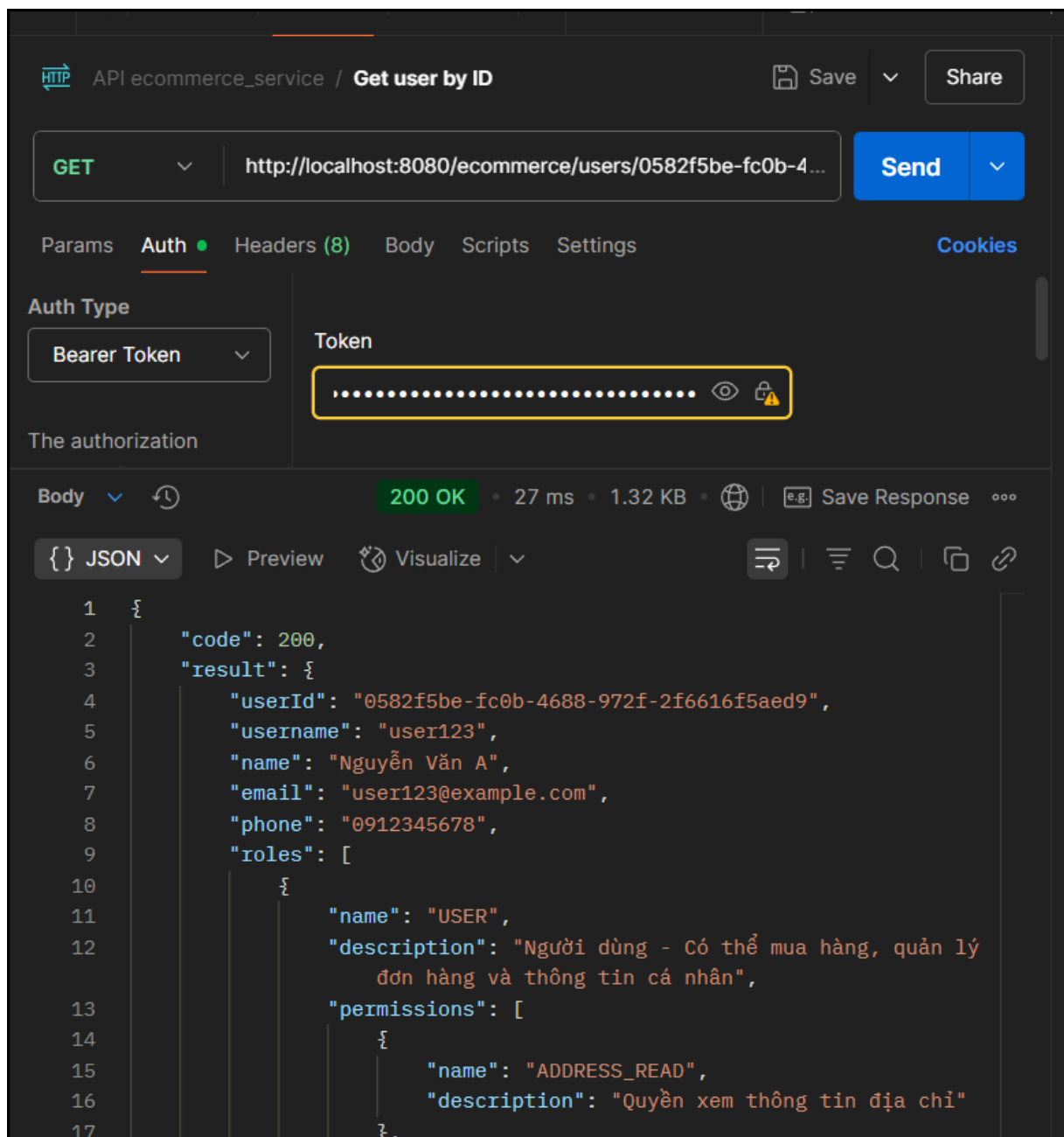
Hình 24. Tạo user với số điện thoại không hợp lệ



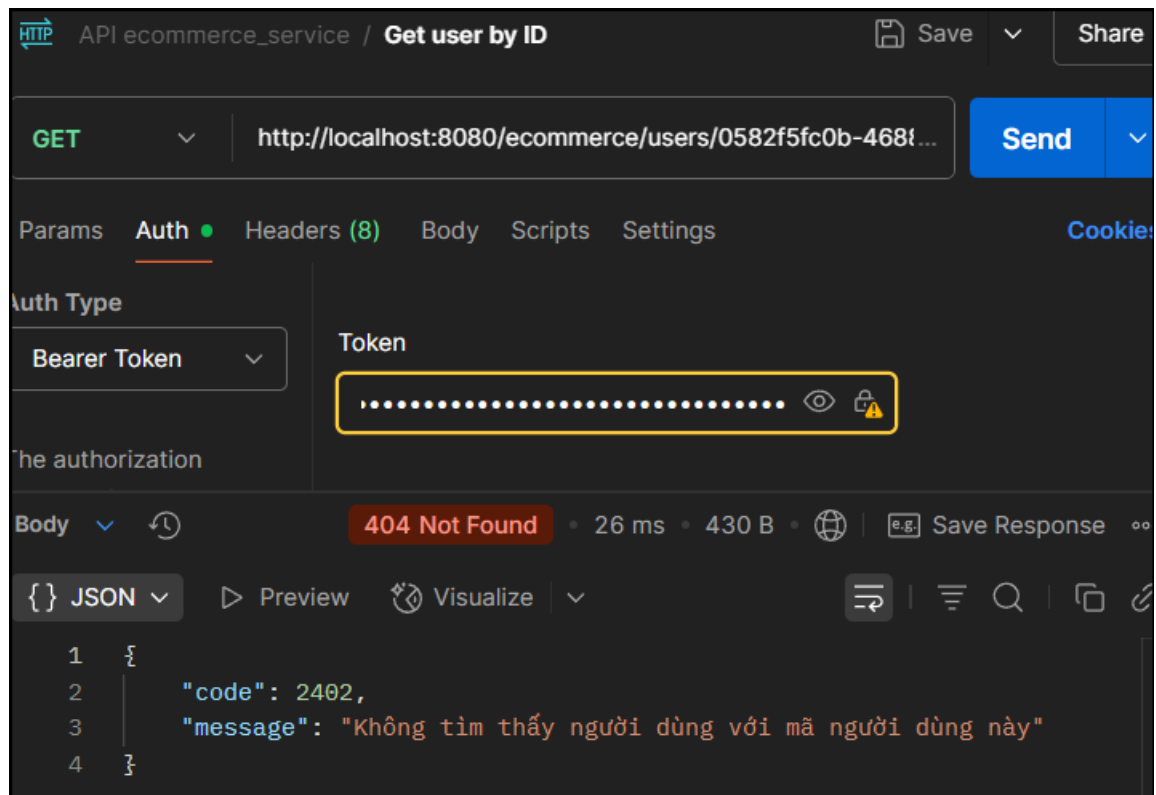
Hình 25. Lấy danh sách người dùng bằng tài khoản admin



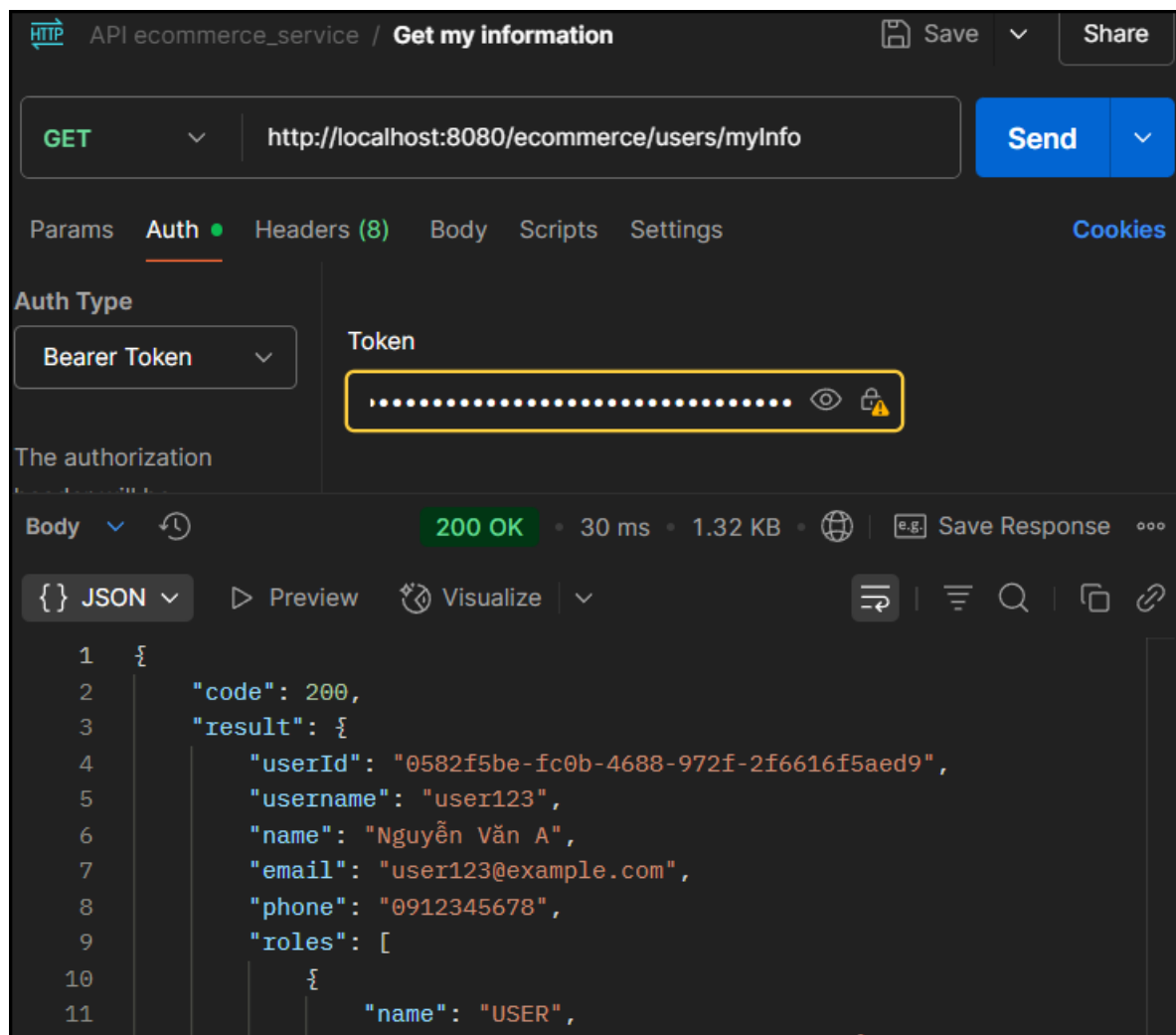
Hình 26. Không lấy được danh sách người dùng bằng tài khoản user thường



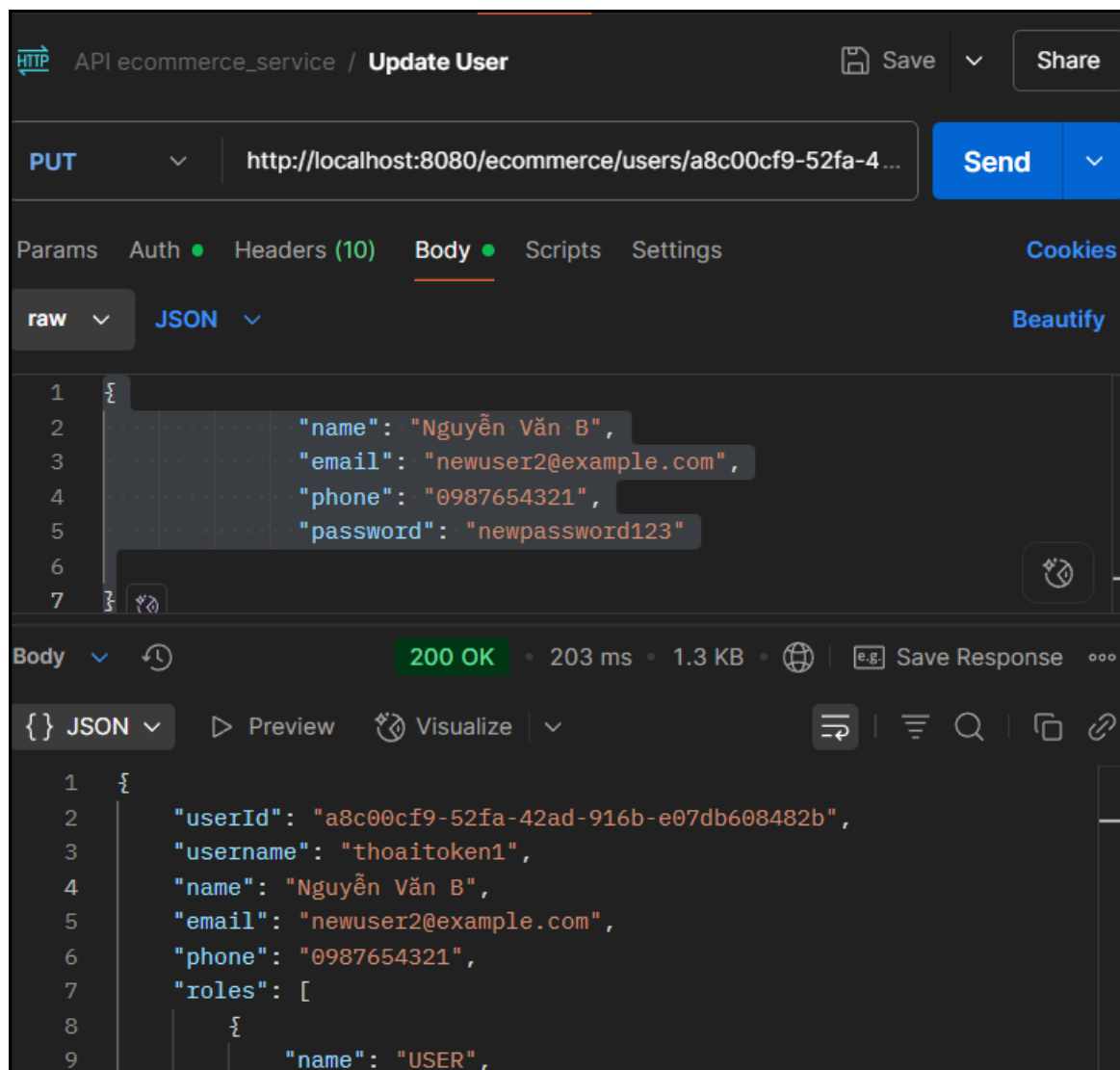
Hình 27. Lấy thông tin người dùng theo ID (chỉ admin hoặc chính user đó)



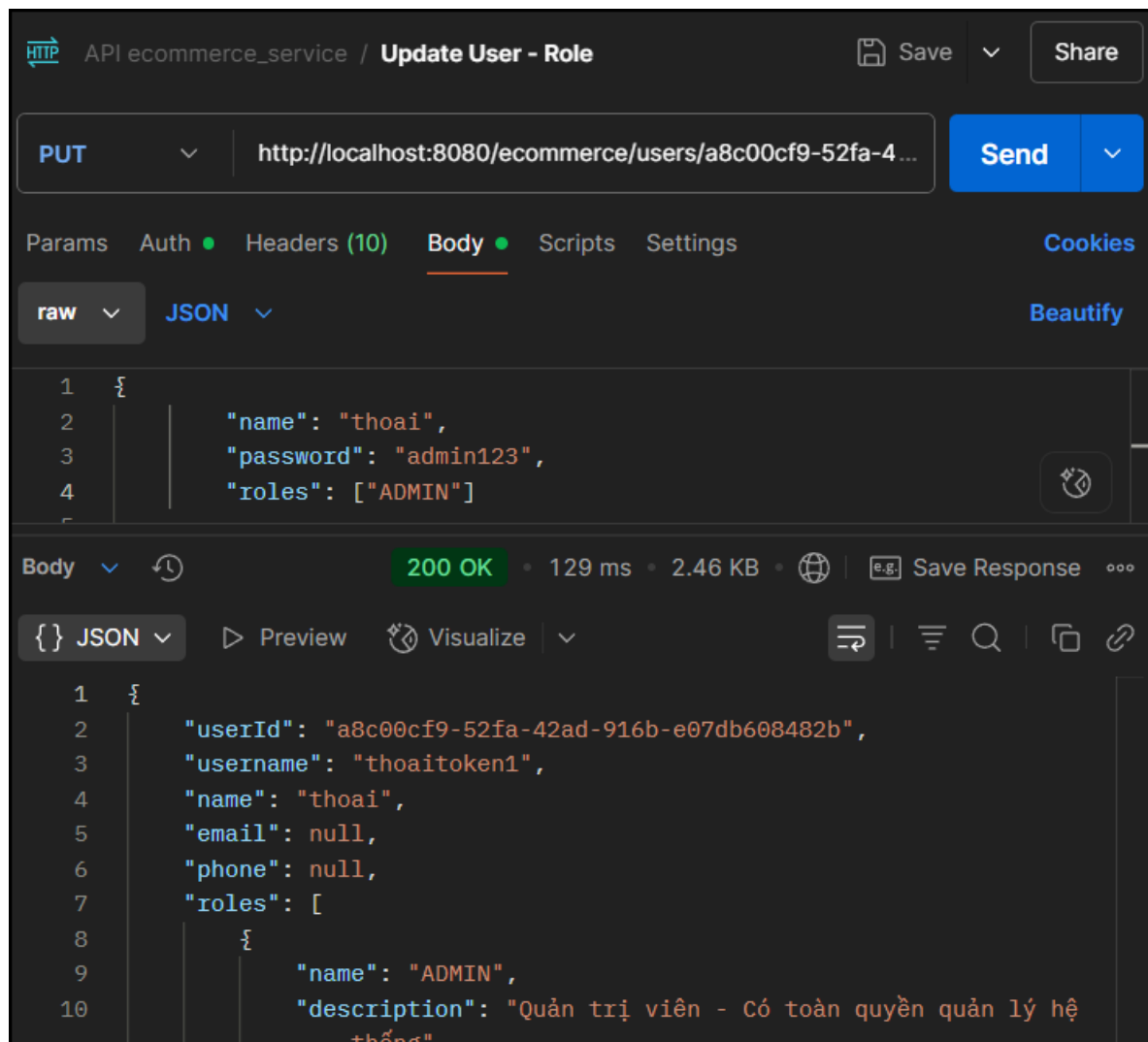
Hình 28. Lấy User không tồn tại



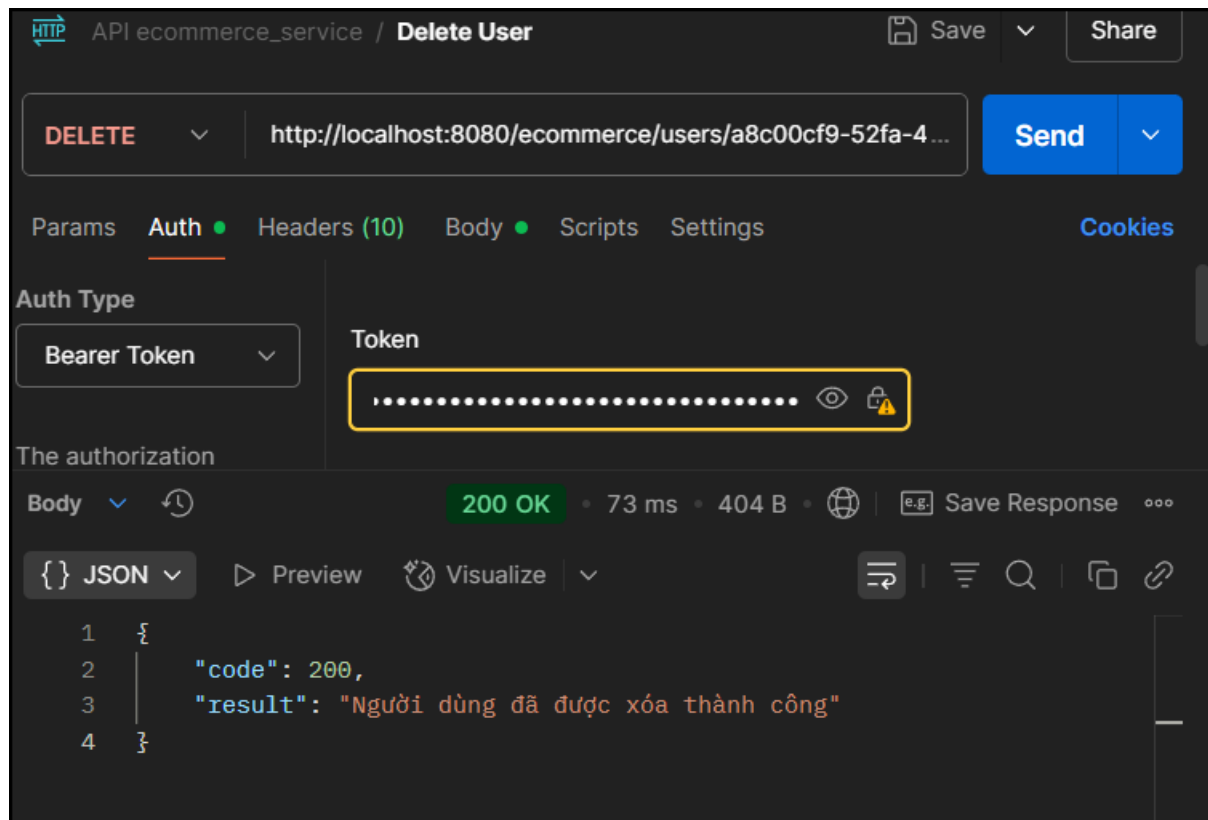
Hình 29. Lấy thông tin của chính mình.



Hình 30. Cập nhật người dùng (Đối với Role thì chỉ có admin mới được cập nhật)

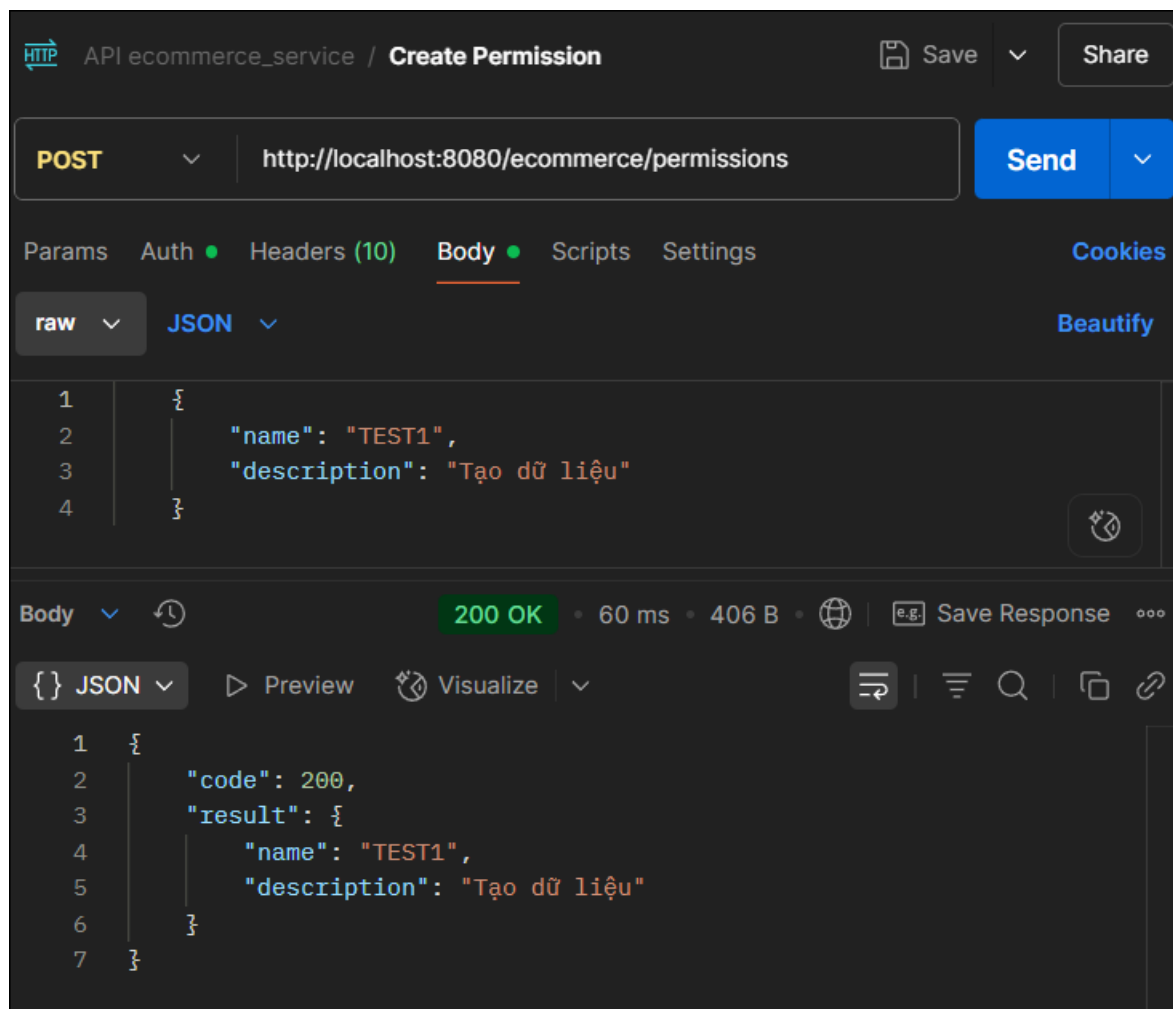


Hình 31. Cập nhật người dùng và Role

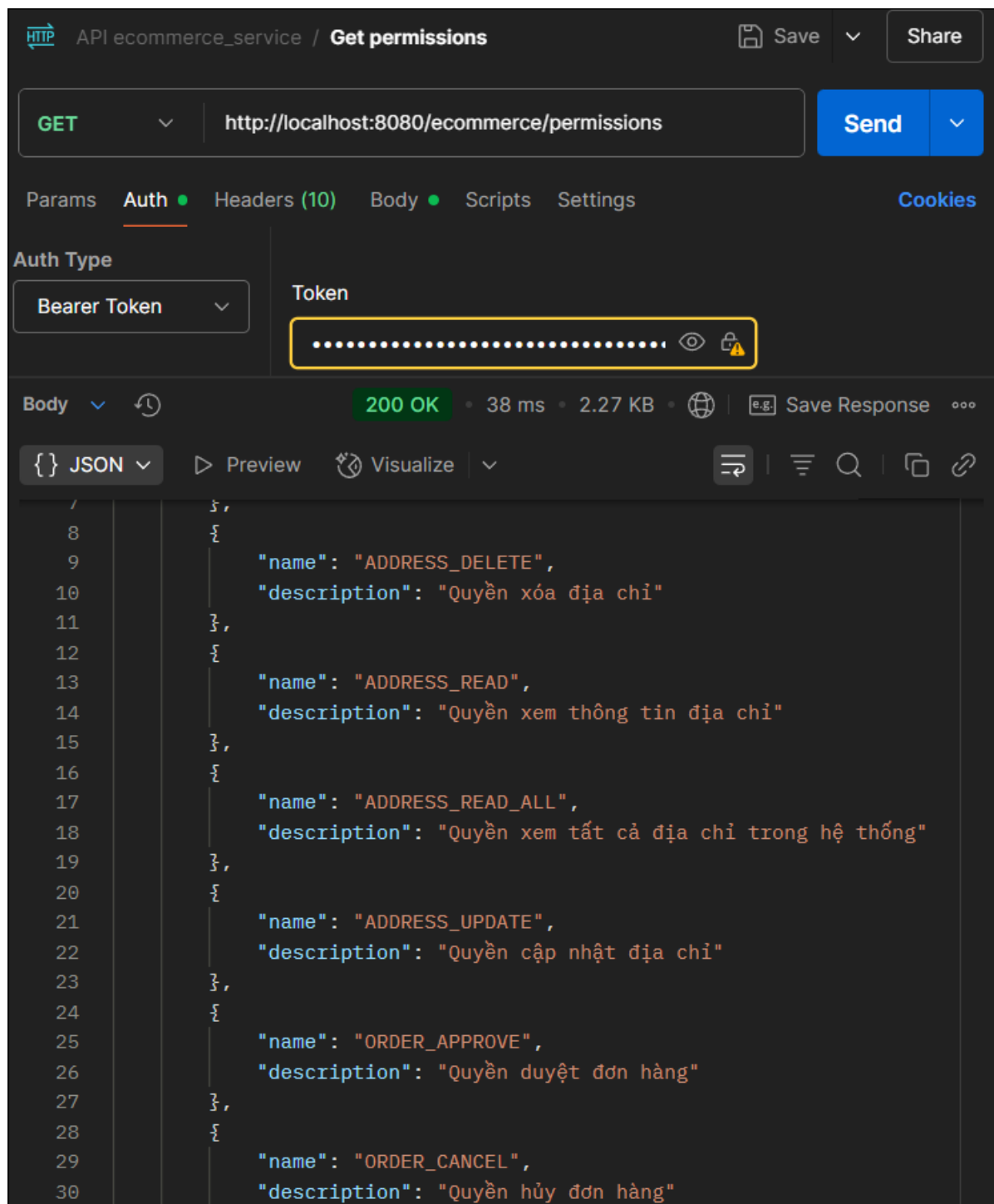


Hình 32. Xóa người dùng với role admin

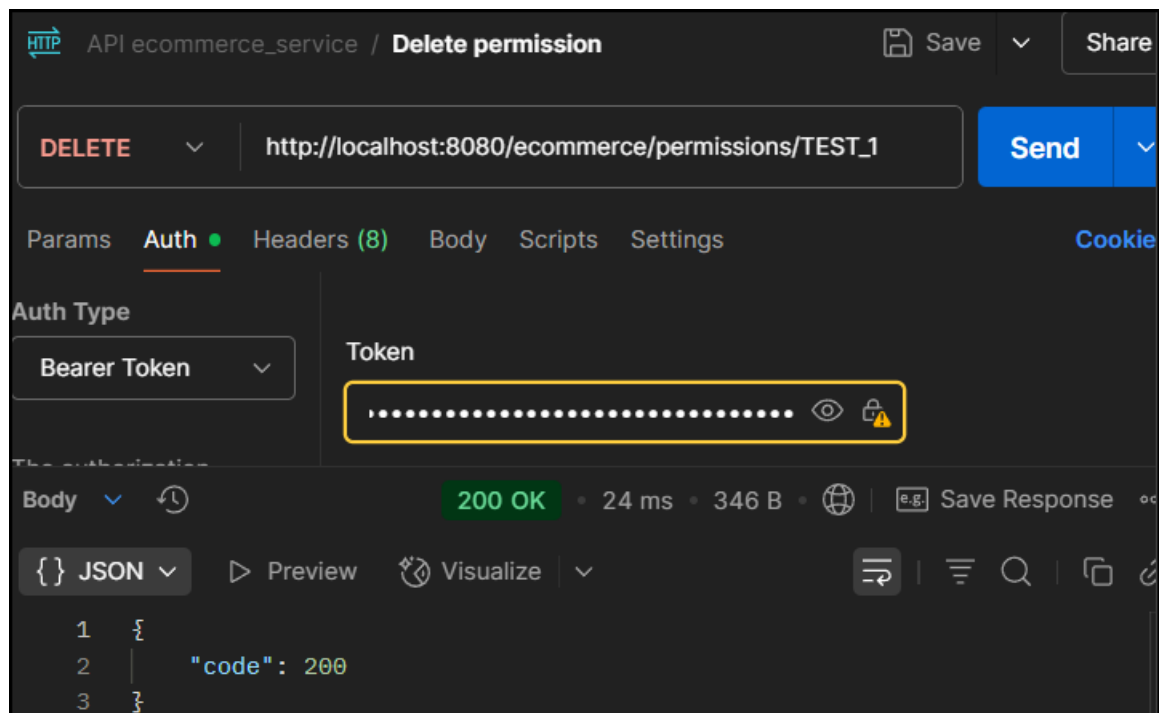
3. Test cases cho Permission API (3 endpoints)



Hình 33. Tạo quyền hạn

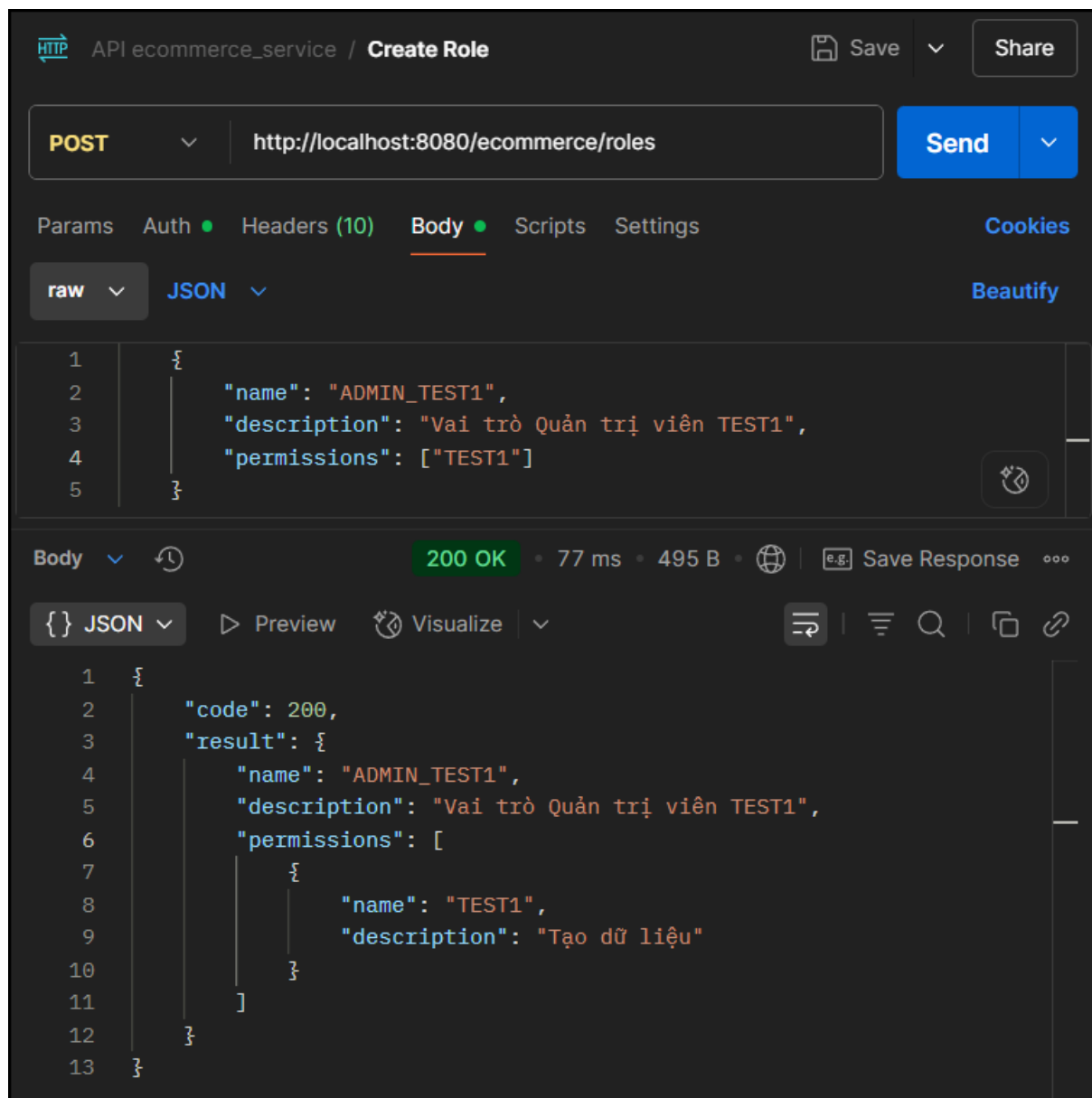


Hình 34. Lấy danh sách tất cả quyền hạn (chỉ admin)

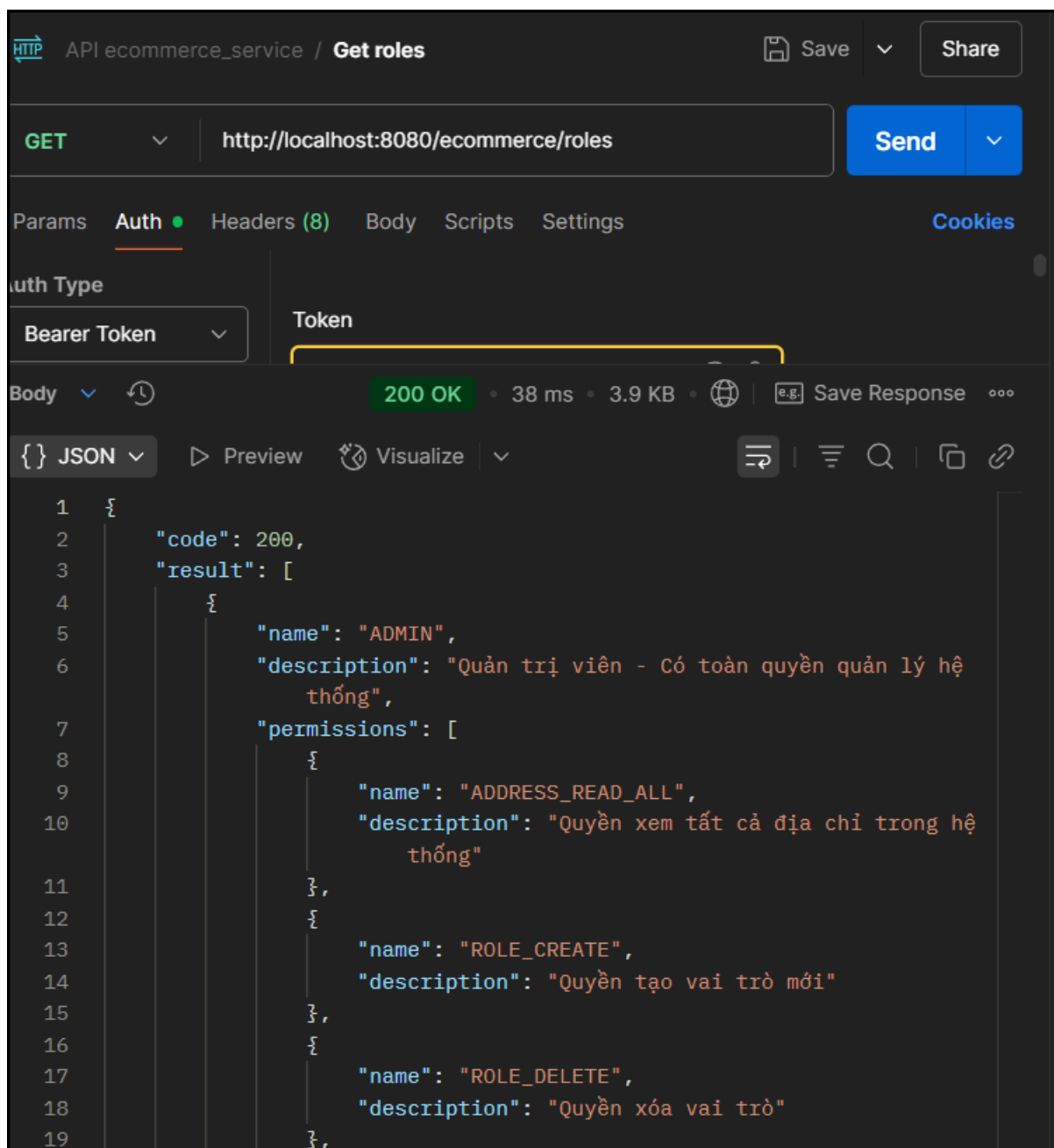


Hình 35. Xóa quyền hạn (chỉ admin)

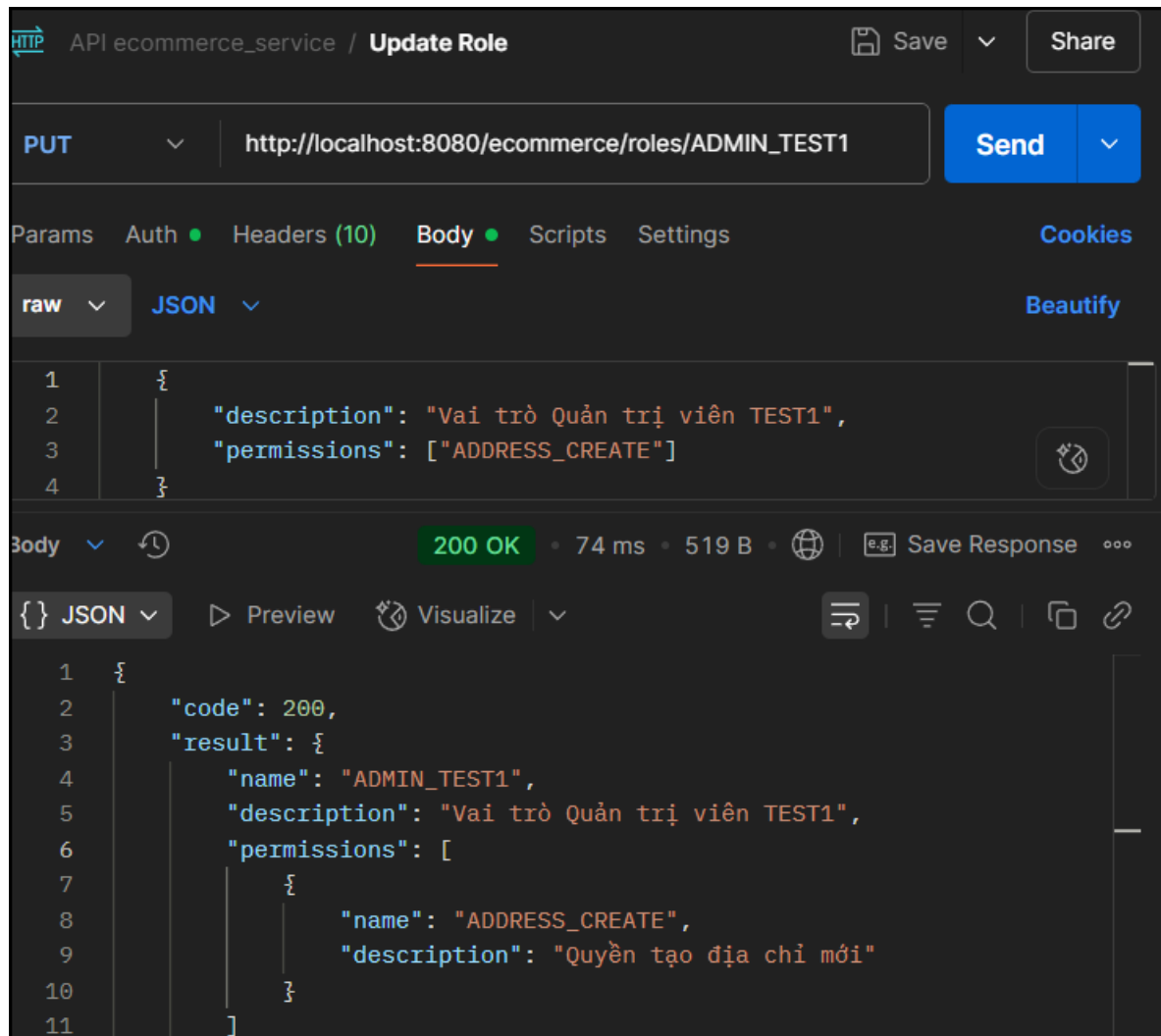
4. Test cases cho Role API (4 endpoints)



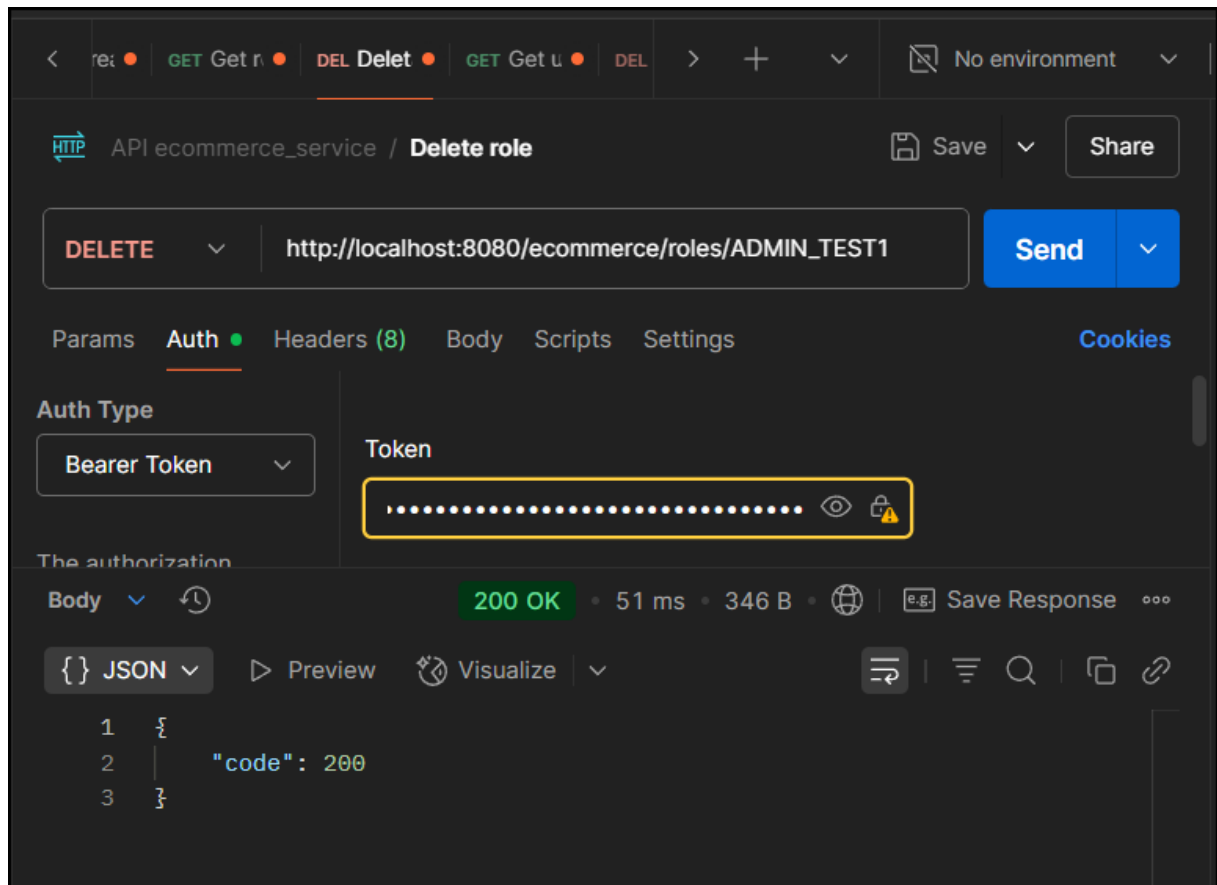
Hình 36. Tạo vai trò mới (chỉ admin)



Hình 37. Lấy danh sách vai trò

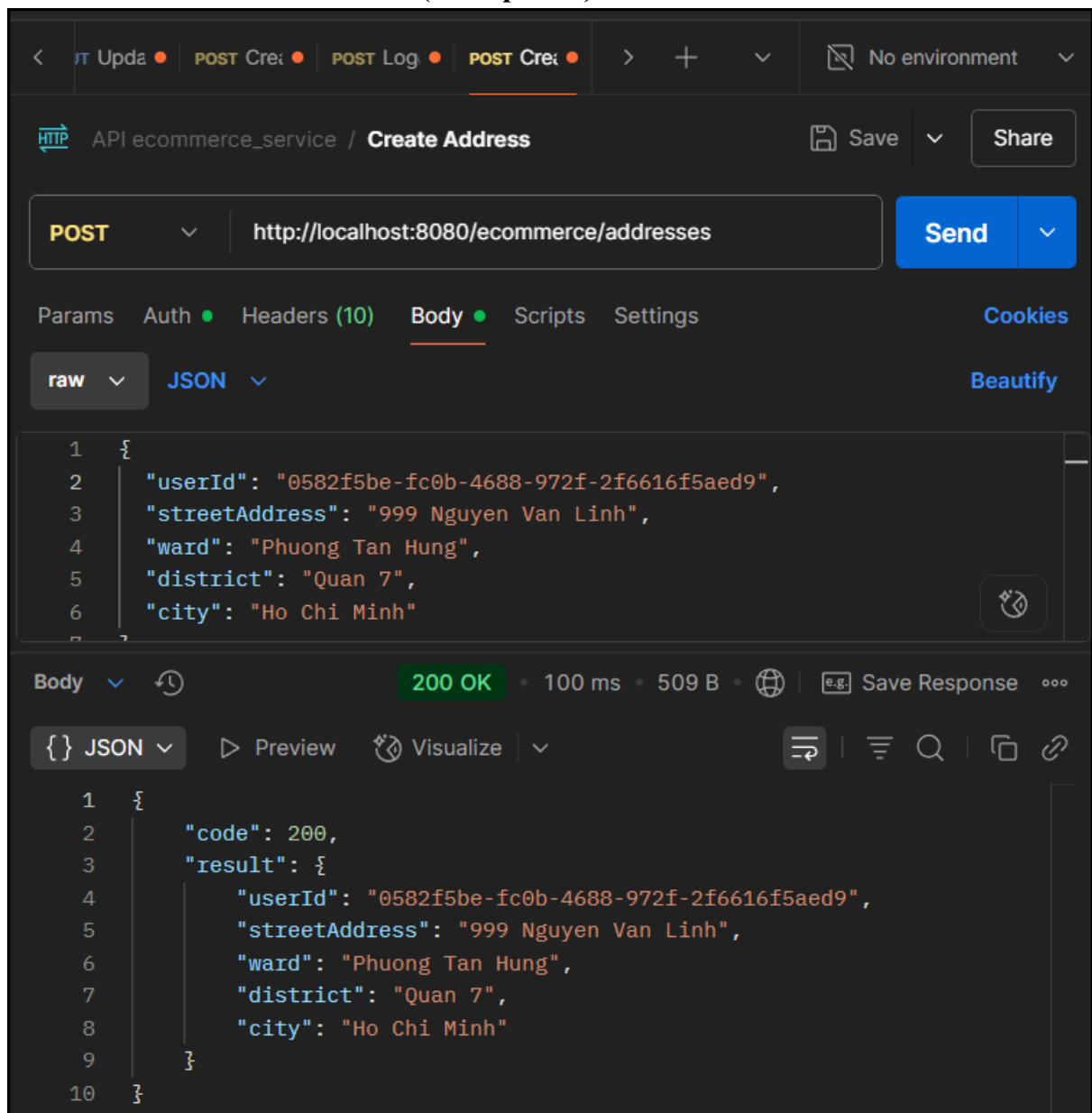


Hình 38. Cập nhật vai trò

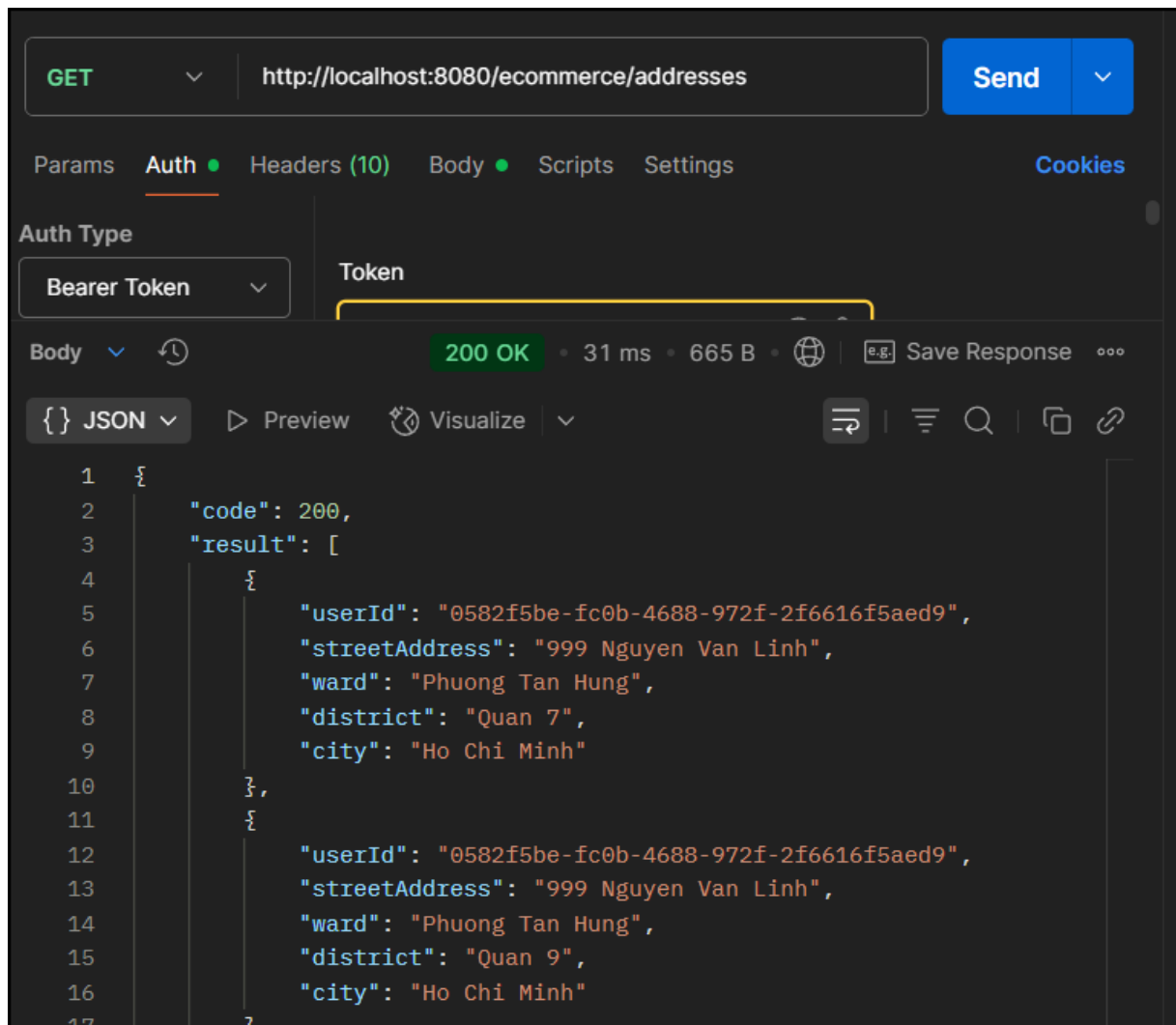


Hình 39. Xóa vai trò.

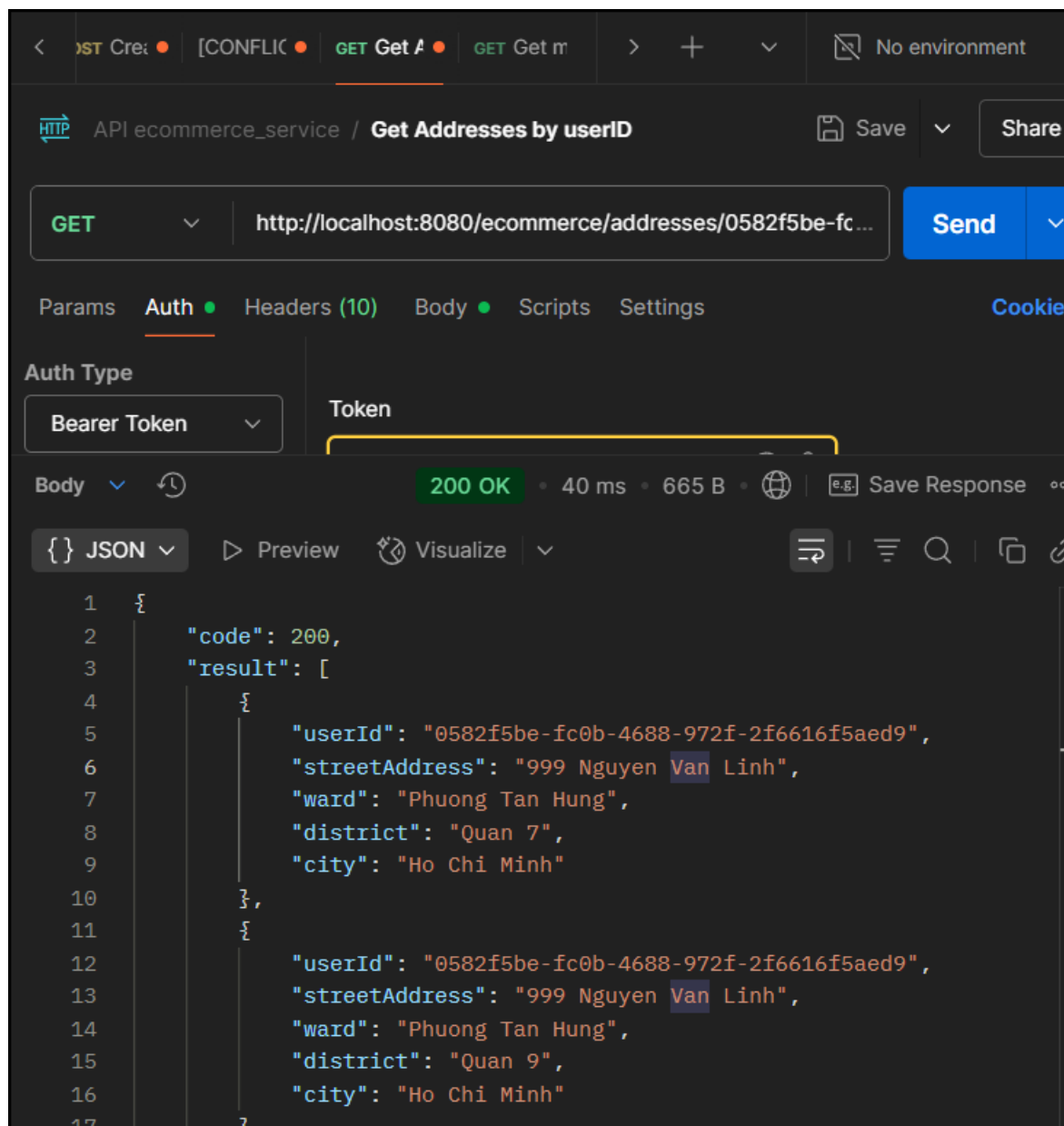
5. Test cases cho Address API (6 endpoints)



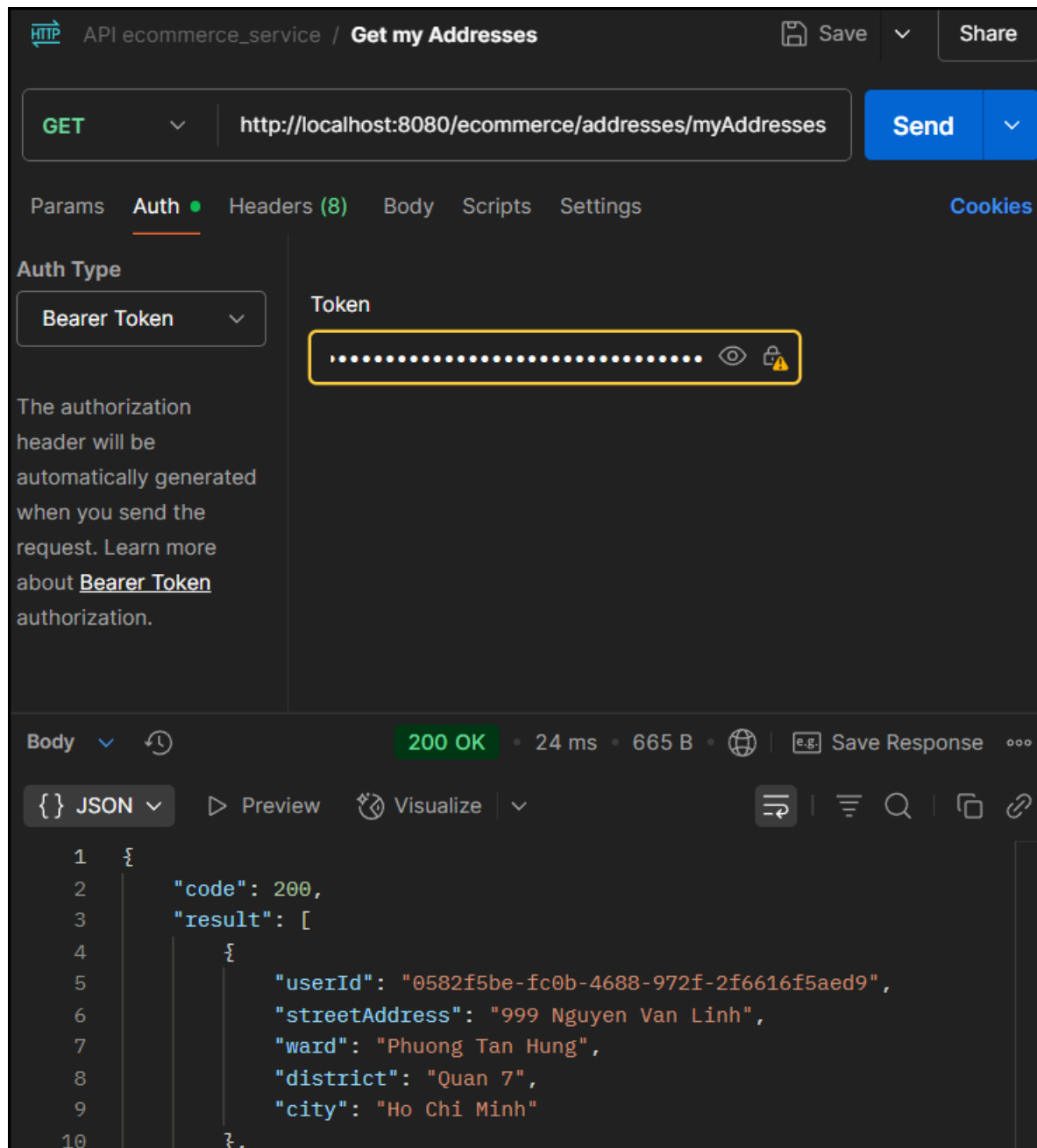
Hình 40. Tạo địa chỉ



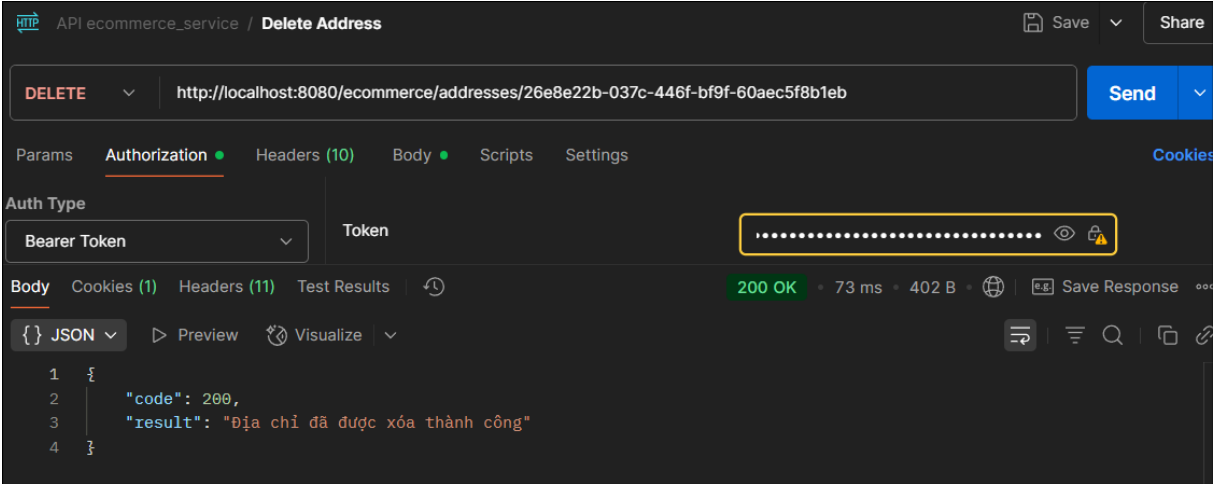
Hình 41. Lấy danh sách địa chỉ.



Hình 42. Lấy địa chỉ theo UserId (Chỉ có admin).

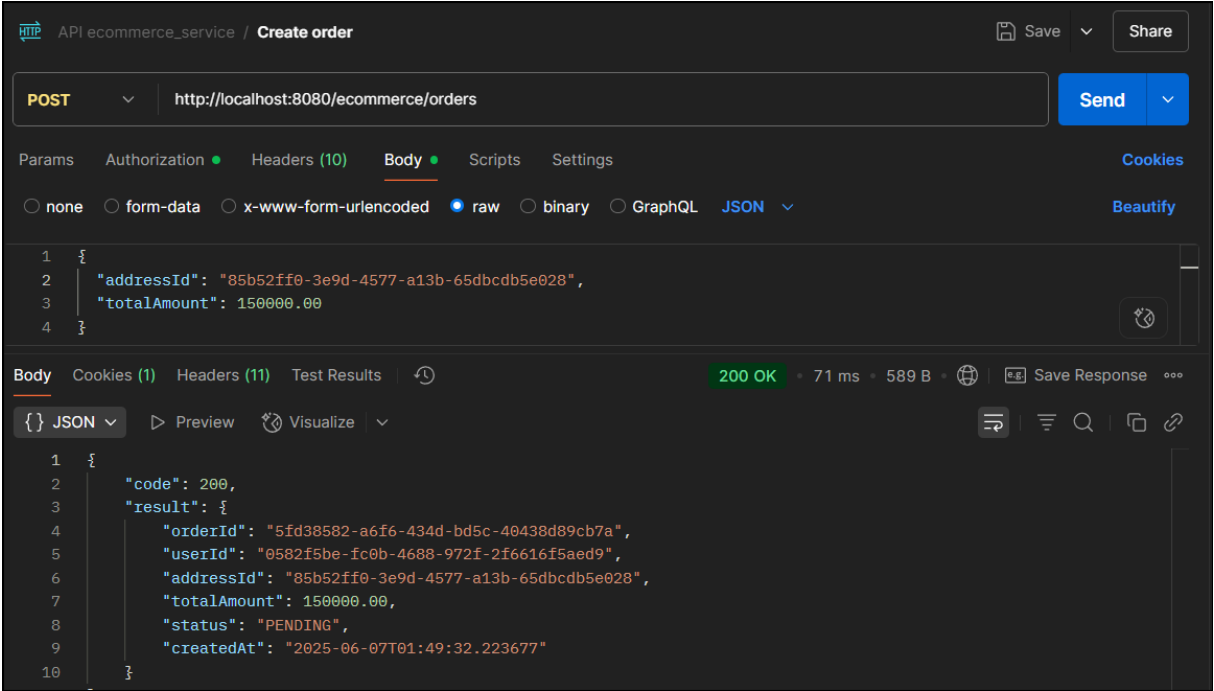


Hình 43. Lấy địa chỉ của chính user

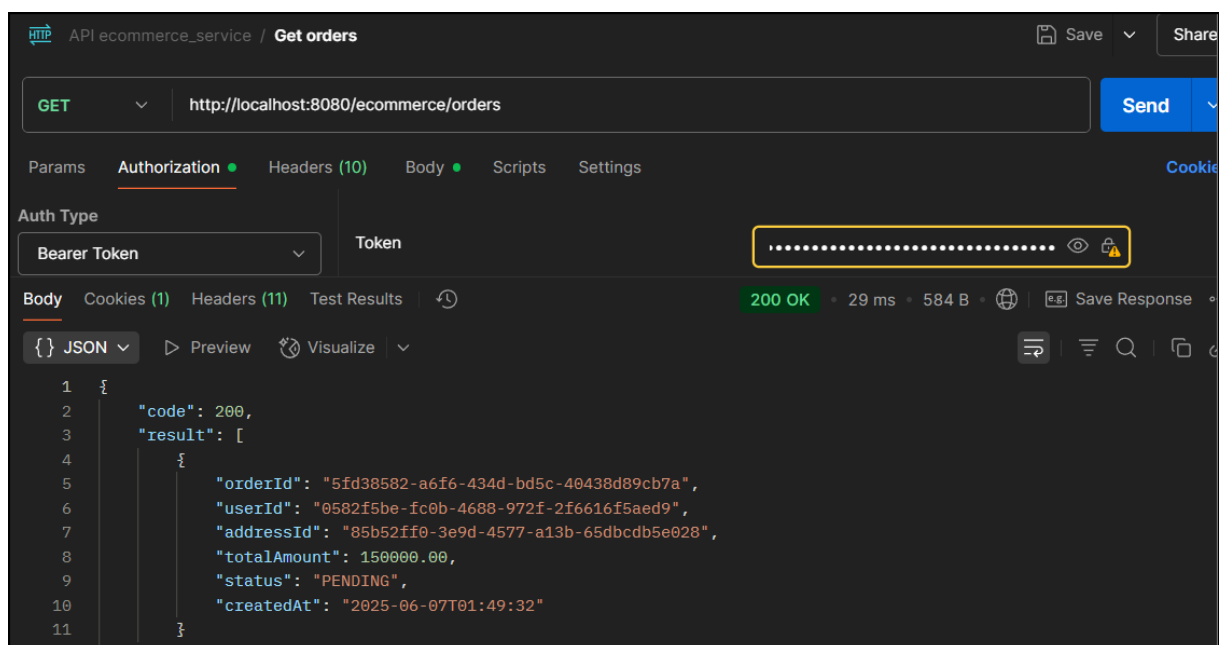


Hình 44. Xóa địa chỉ

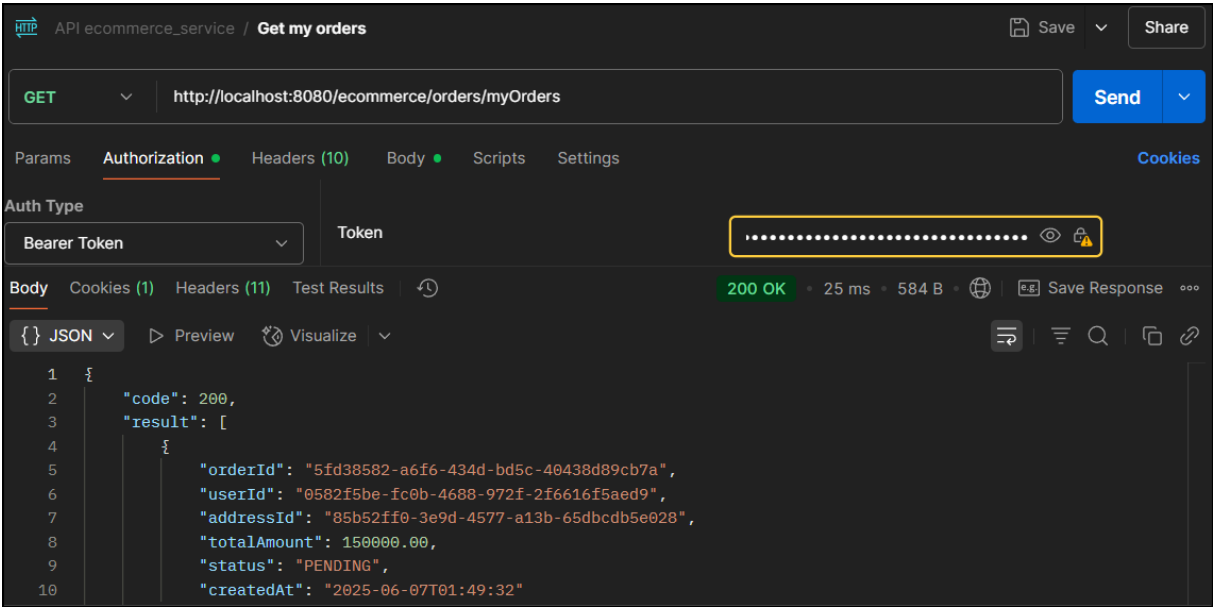
6. Test cases cho Order API (8 endpoints)



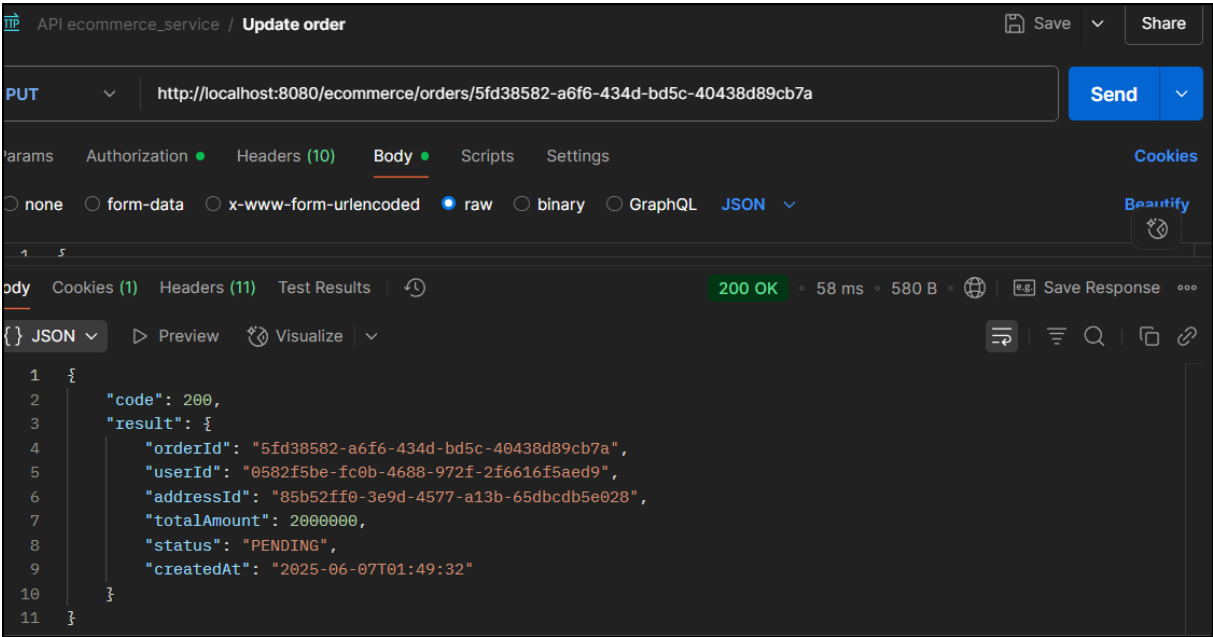
Hình 45. Tạo đơn hàng.



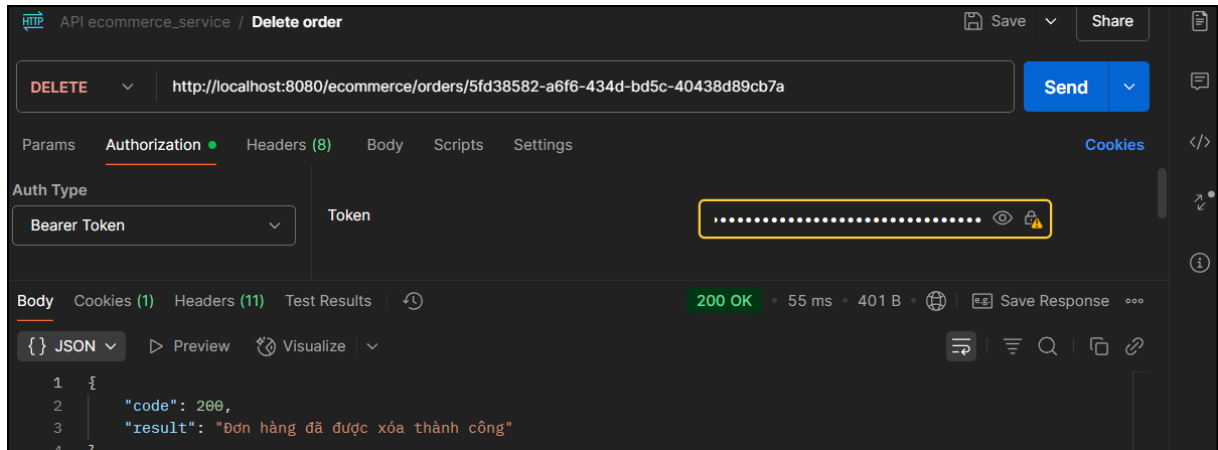
Hình 46. Lấy danh sách đơn hàng.



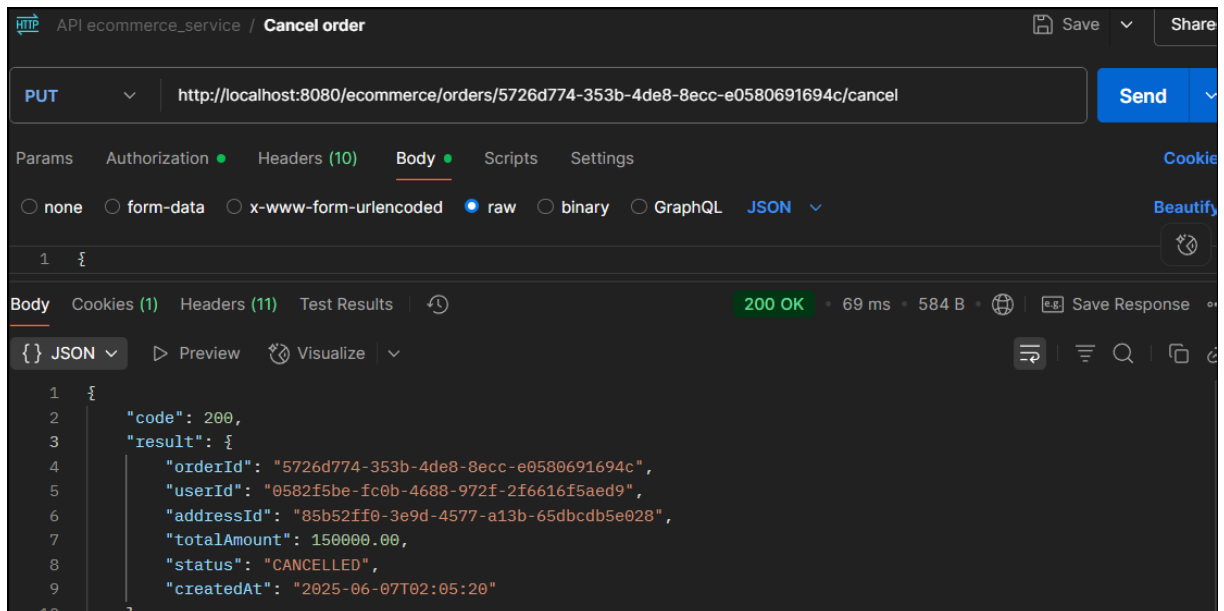
Hình 47. Xem đơn hàng của chính người dùng.



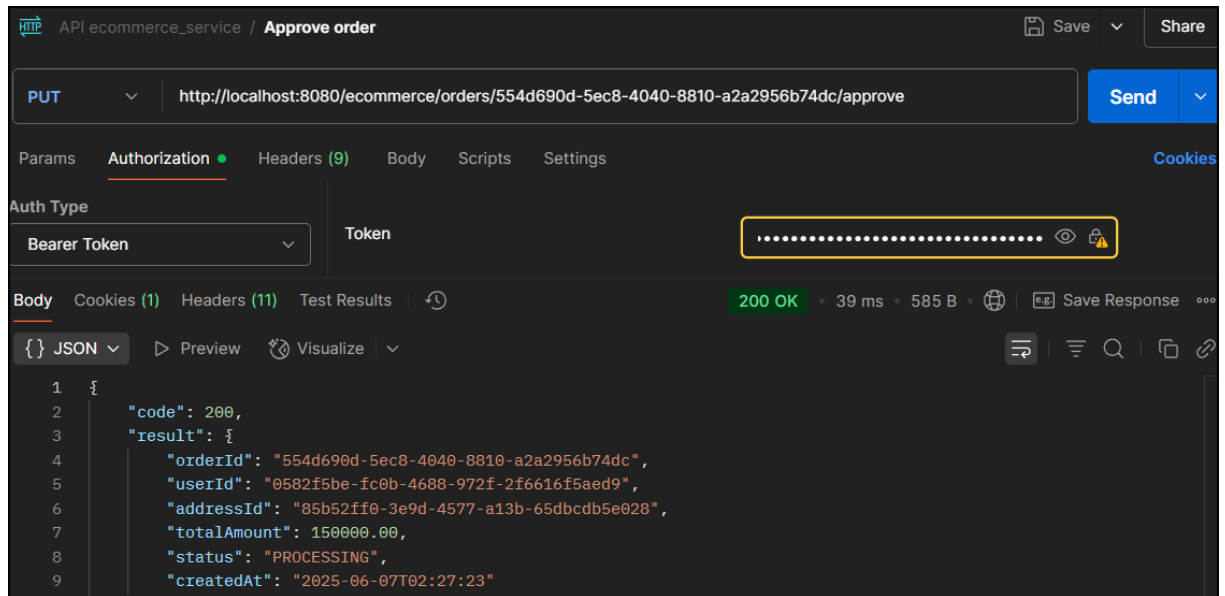
Hình 48. Cập nhật đơn hàng.



Hình 49. Xóa đơn hàng



Hình 50. Hủy đơn hàng



Hình 51. Duyệt đơn hàng bởi Shop hoặc admin