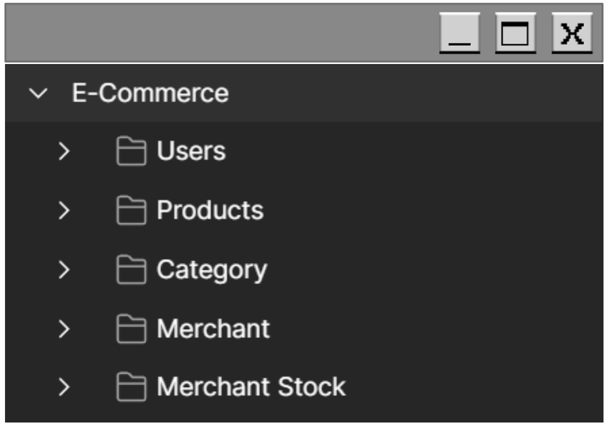
Mohammed Alherz

Capstone 1: Spring Boot API Endpoints



Exploring five advanced RESTful endpoints in a Spring Boot e-commerce project with Postman demonstrations

أكاديمية طويق
Tuwaiq Academy

Introduction

- This project extedns the core e-commerce API spring Boot
- Five additional endpoints were developed to enhance product interaction, user engagement, and merchant operations
- Each endpoint is demonstrated with Postman and explained with backend logic

أكاديمية طويق
Tuwaiq Academy

01 First Endpoint

Best-Selling Product

Best-Selling Product

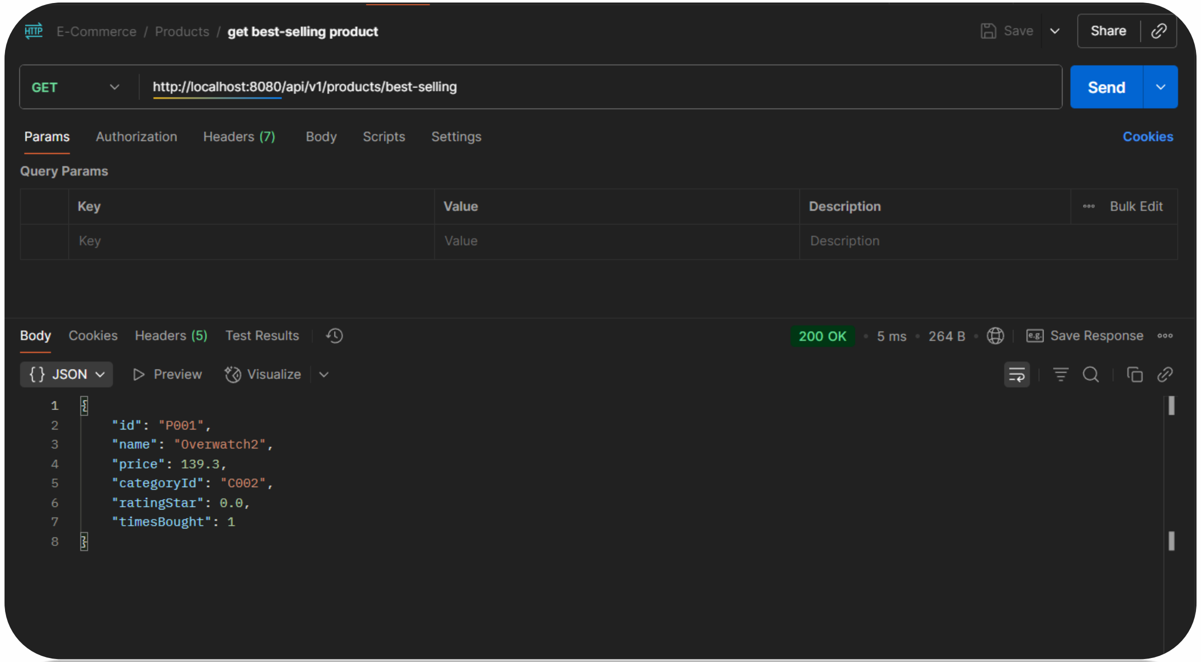
Description: Retrieves the product with the highest number of purchases.

How it works:

1. Each product has a **timesBought** counter.
2. Every purchase increments the counter.
3. The endpoint compares all values and return the product with the highest count.

أكاديمية طويق
Tuwaiq Academy

GET /products/best-selling



GET http://localhost:8080/api/v1/products/best-selling

Key	Value	Description
Key	Value	Description

200 OK • 5 ms • 264 B

```
{
    "id": "P001",
    "name": "Overwatch2",
    "price": 139.3,
    "categoryId": "C002",
    "ratingStar": 0.0,
    "timesBought": 1
}
```

أكاديمية طويق
Tuwaiq Academy

02 Second Endpoint

Products-by-merchant

Products-by-merchant

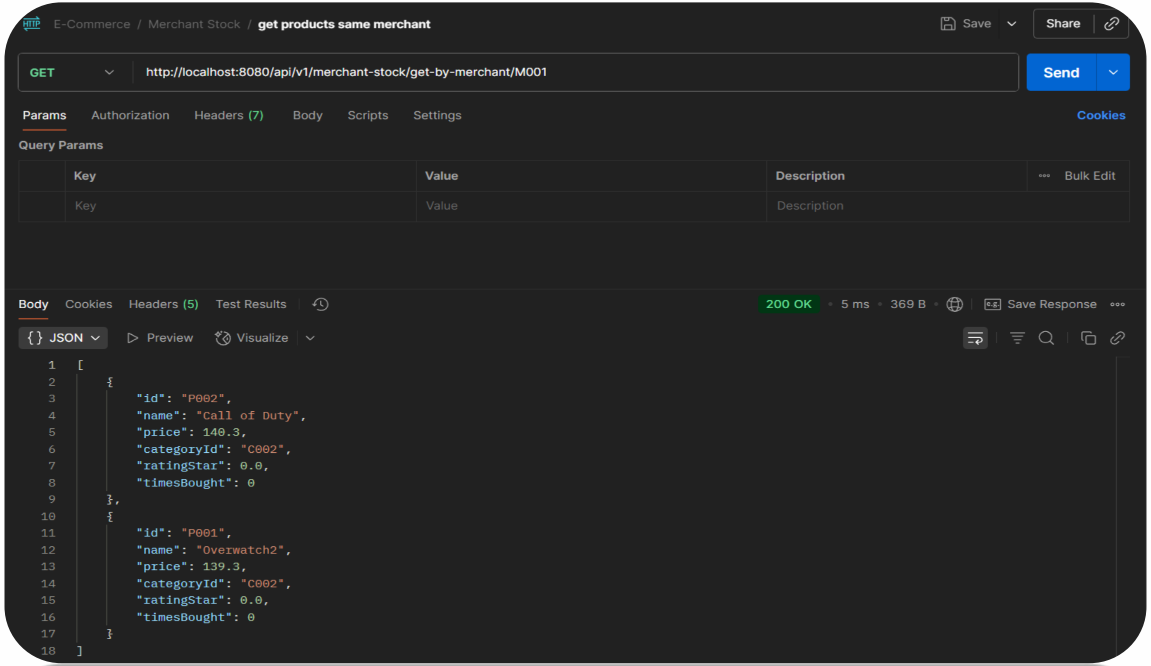
Description: Fetches all products related to a specific merchant.

How it works:

1. Accepts **merchantId** as path variable.
2. Calls **getAllProductsByMerchant(merchantId).**
3. For each product, it checks if the merchant own the product using **merchantHasProduct(merchantId, productid)**.
4. Filters and returns only the products that belong to the given merchant.

أكاديمية طويق
Tuwaiq Academy

GET /merchant-stock/get-by-merchant/M001



GET http://localhost:8080/api/v1/merchant-stock/get-by-merchant/M001

Query Params

Key	Value	Description
Key	Value	Description

200 OK • 5 ms • 369 B

```
[
    {
        "id": "P002",
        "name": "Call of Duty",
        "price": 140.3,
        "categoryId": "C002",
        "ratingStar": 0.0,
        "timesBought": 0
    },
    {
        "id": "P001",
        "name": "Overwatch2",
        "price": 139.3,
        "categoryId": "C002",
        "ratingStar": 0.0,
        "timesBought": 0
    }
]
```

أكاديمية طويق
Tuwaiq Academy

03 Third Endpoint

User's Most-purchased-product

User's Most-purchased-product

Description: Returns the product a specific user has purchased the most

How it works:

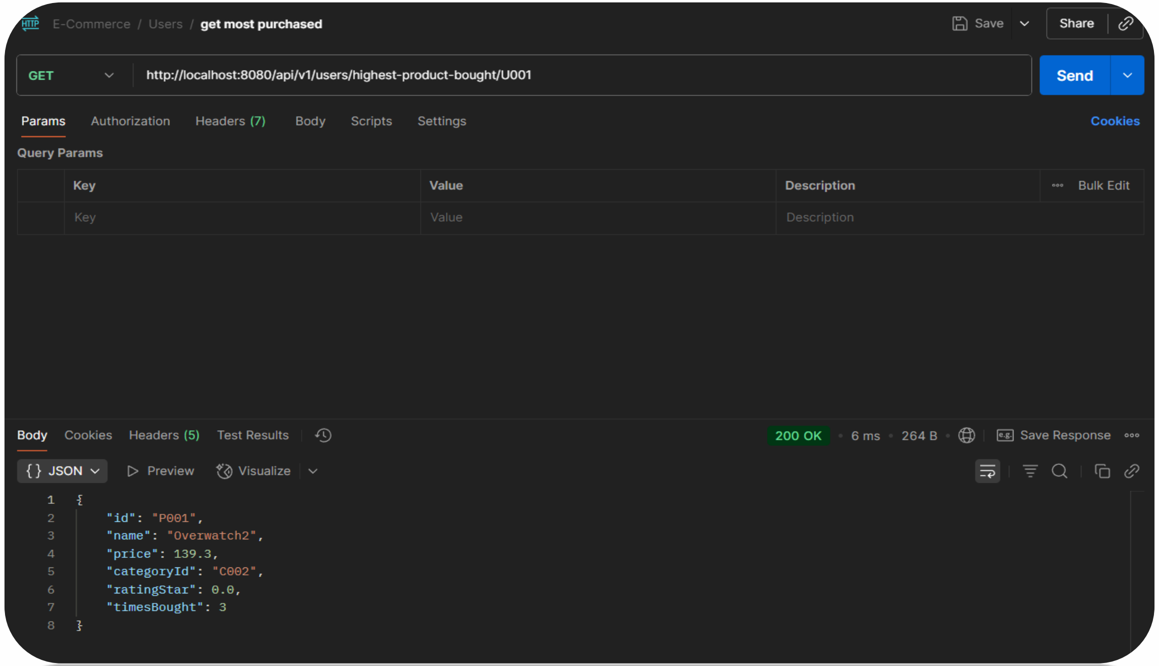
1. Uses a HashMap: userID → (productid → count).
2. On purchase, count is updated.
3. **When called:**

Validate user.

Checks purchase history.

Returns product with highest count.

GET /users/highest-product-bought/U001



GET http://localhost:8080/api/v1/users/highest-product-bought/U001

Key	Value	Description
Key	Value	Description

Body — 200 OK • 6 ms • 264 B

```
{
    "id": "P001",
    "name": "Overwatch2",
    "price": 139.3,
    "categoryId": "C002",
    "ratingStar": 0.0,
    "timesBought": 3
}
```

أكاديمية طويق Tuwaiq Academy

04 Fourth Endpoint

Rate-product-by-user

Rate-product-by-user

Description: Allos a user to rate a product they've purchased.

How it works:

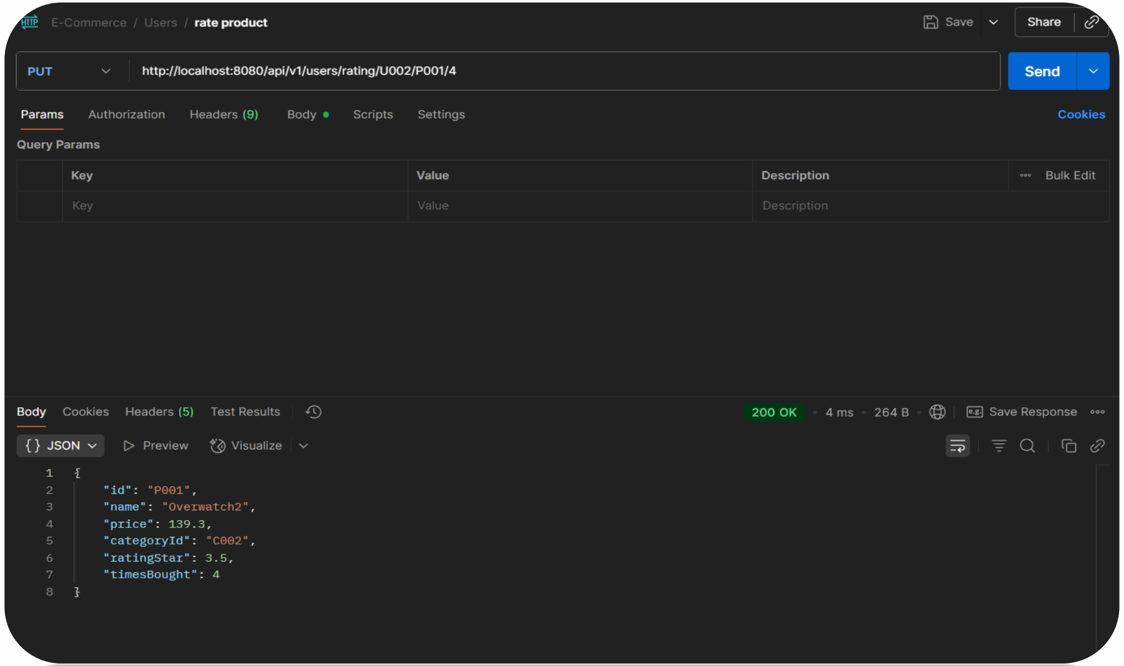
1. Confirms the user-product pair exist in **purchasedCart**.
2. Validates rating(1-5).
3. Update rating HashMap: productId → (userId → rating).
4. Recalculates average and updates product.

Validation Rules:

- Must be a valid and product.
- Purchase required.
- Rating must be 1-5.

أكاديمية طويق
Tuwaiq Academy

PUT /users/rating/U001/P001/3
PUT /users/rating/U002/P001/4


E-Commerce / Users / rate product

PUT http://localhost:8080/api/v1/users/rating/U002/P001/4

Query Params

Key	Value	Description
Key	Value	Description

200 OK · 4 ms · 264 B

```
{
    "id": "P001",
    "name": "Overwatch2",
    "price": 139.3,
    "categoryId": "C002",
    "ratingStar": 3.5,
    "timesBought": 4
}
```

05 Fifth Endpoint

Refund-product-by-user

Refund-product-by-user

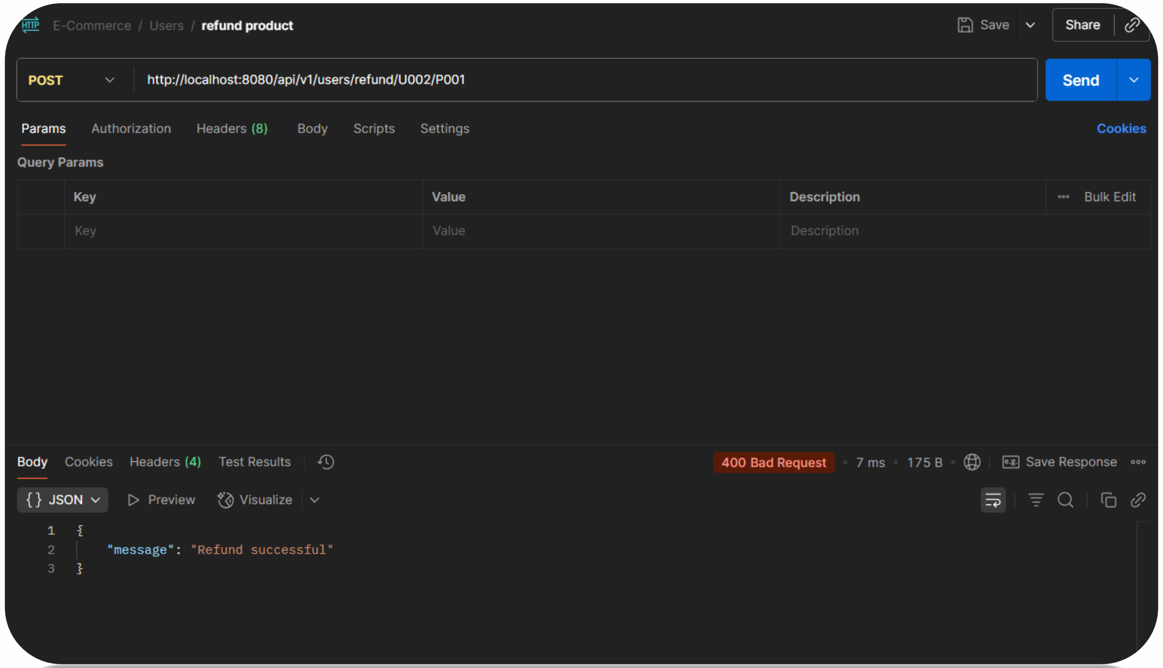
Description: Processes a refund for a purchased product.

How it works:

1. Checks if user exists.
2. Verifies product was purchased.
3. Refunds product price to user’s balance.
4. Increases merchant’s stock.
5. Remove record from **purchasedCart** and **userProductMap**.

أكاديمية طويق
Tuwaiq Academy

POST /users/refund/U002/P001



POST http://localhost:8080/api/v1/users/refund/U002/P001

Query Params

Key	Value	Description
Key	Value	Description

400 Bad Request • 7 ms • 175 B

```
{
  "message": "Refund successful"
}
```

أكاديمية طويق
Tuwaiq Academy

Conclusions

- These five endpoints add real-world functionality to the API.
- They improve the user experience (rating, refunds), support merchants (inventory), and provide analytics (top products).
- Each was tested with Postman and validated with custom logic.

أكاديمية طويق
Tuwaiq Academy

Thank you!

Do you have any questions?

CREDITS: This presentation template was created by **Slidesgo**, and includes icons, infographics & images by **Freepik**

OK Cancel Apply

أكاديمية طويق
Tuwaiq Academy