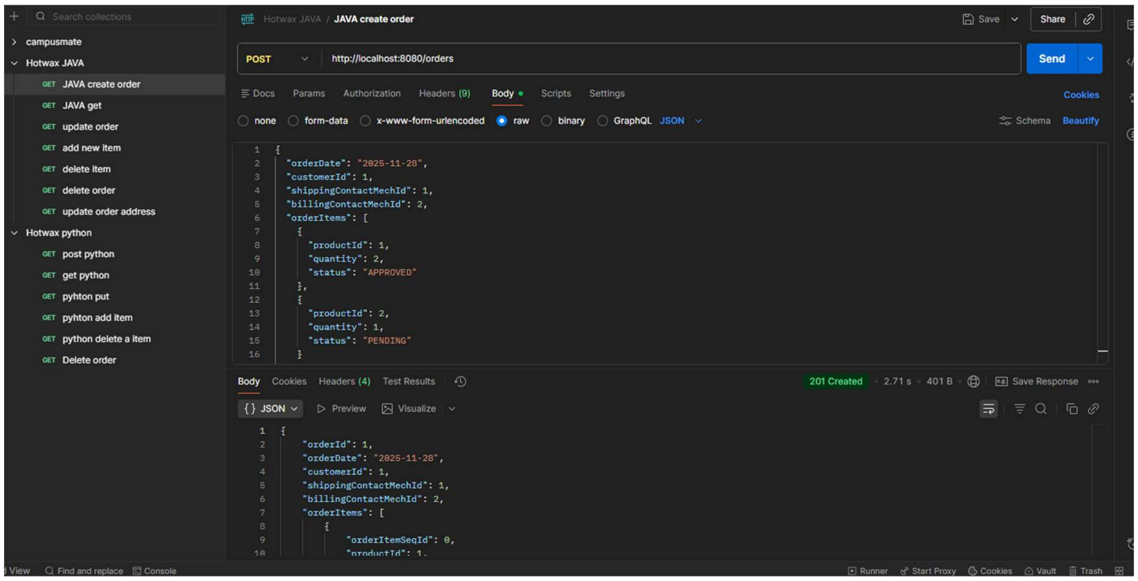
## API Execution Screenshots

Below are the screenshots from Postman demonstrating the successful execution of all required API scenarios for the Order Management System.

### Scenario 1: Create an Order

- **Endpoint:** POST /orders
- **Description:** Successfully created a new order containing multiple items (T-Shirt and Jeans) in a single atomic transaction.
- **Status Code:** 201 Created



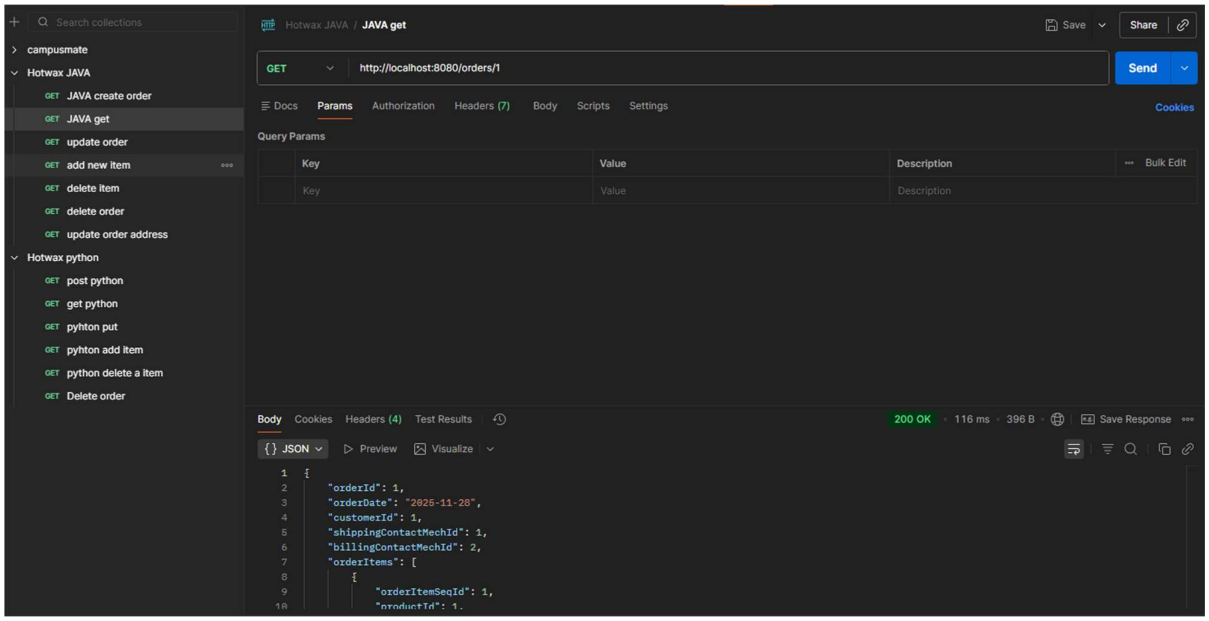
```
POST http://localhost:8080/orders
```

```
{
  "orderDate": "2025-11-28",
  "customerId": 1,
  "shippingContactMechId": 1,
  "billingContactMechId": 2,
  "orderItems": [
    {
      "productId": 1,
      "quantity": 2,
      "status": "APPROVED"
    },
    {
      "productId": 2,
      "quantity": 1,
      "status": "PENDING"
    }
```

```
{
    "orderId": 1,
    "orderDate": "2025-11-28",
    "customerId": 1,
    "shippingContactMechId": 1,
    "billingContactMechId": 2,
    "orderItems": [
        {
            "orderItemSeqId": 0,
            "productId": 1,
```

### Scenario 2: Retrieve Order Details

- **Endpoint:** GET /orders/{orderId}
- **Description:** Retrieved the full details of the created order, including customer ID, addresses, and the list of order items.
- **Status Code:** 200 OK

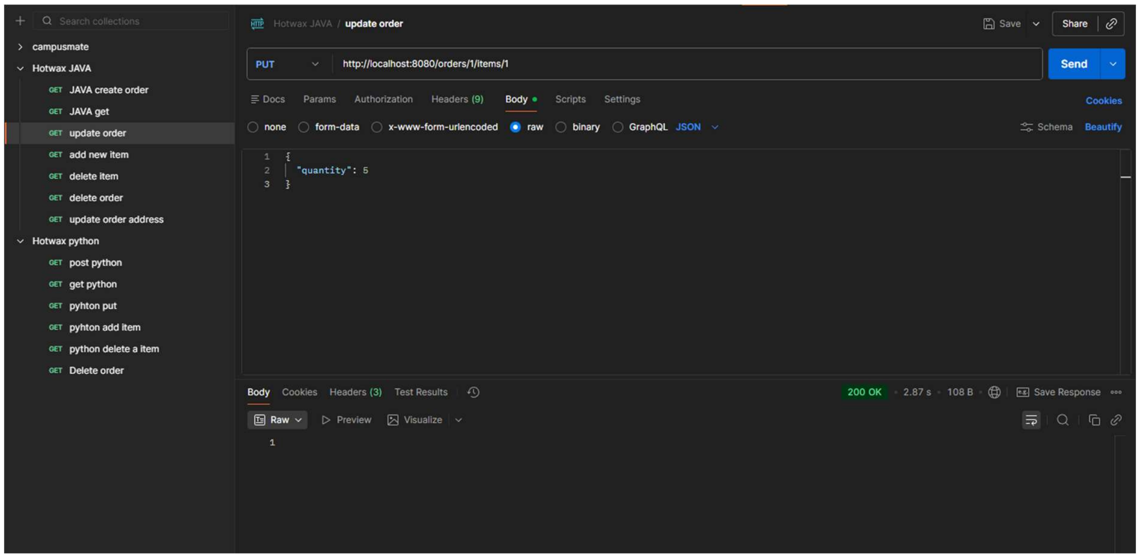


```
GET http://localhost:8080/orders/1
```

```
{
    "orderId": 1,
    "orderDate": "2025-11-28",
    "customerId": 1,
    "shippingContactMechId": 1,
    "billingContactMechId": 2,
    "orderItems": [
        {
            "orderItemSeqId": 1,
            "productId": 1,
```

### Scenario 3: Update Order (Shipping/Billing Address)

- **Endpoint:** PUT /orders/{orderId}
- **Description:** Updated the Shipping and Billing Contact Mechanism IDs for an existing order.
- **Status Code:** 200 OK



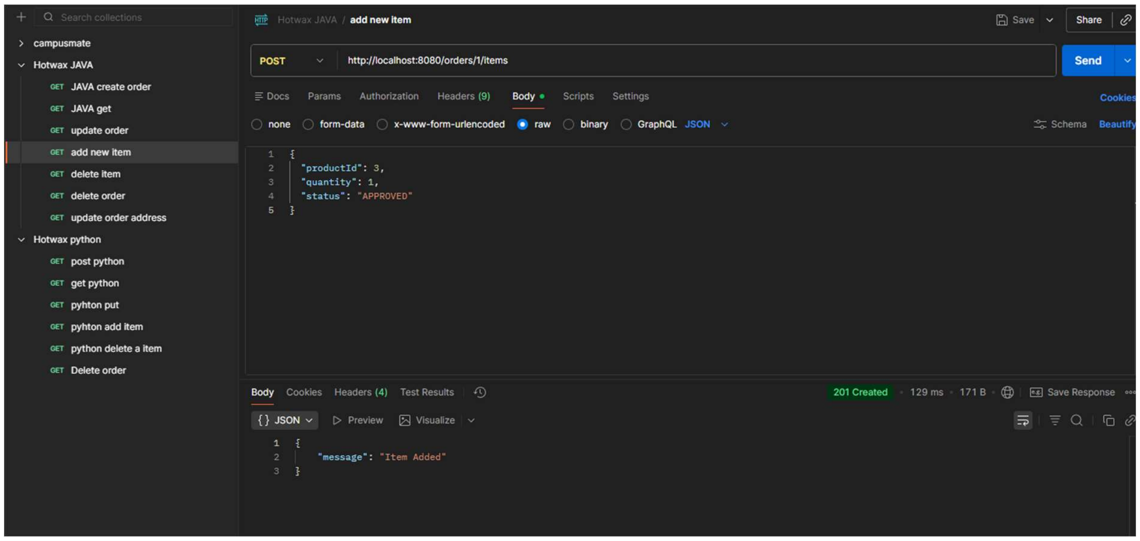
PUT http://localhost:8080/orders/1/items/1

```
{
  "quantity": 5
}
```

200 OK · 2.87 s · 108 B

### Scenario 4: Add New Item to Order

- **Endpoint:** POST /orders/{orderId}/items
- **Description:** Added a new product (e.g., Sneakers) to an existing order.
- **Status Code:** 201 Created



POST http://localhost:8080/orders/1/items

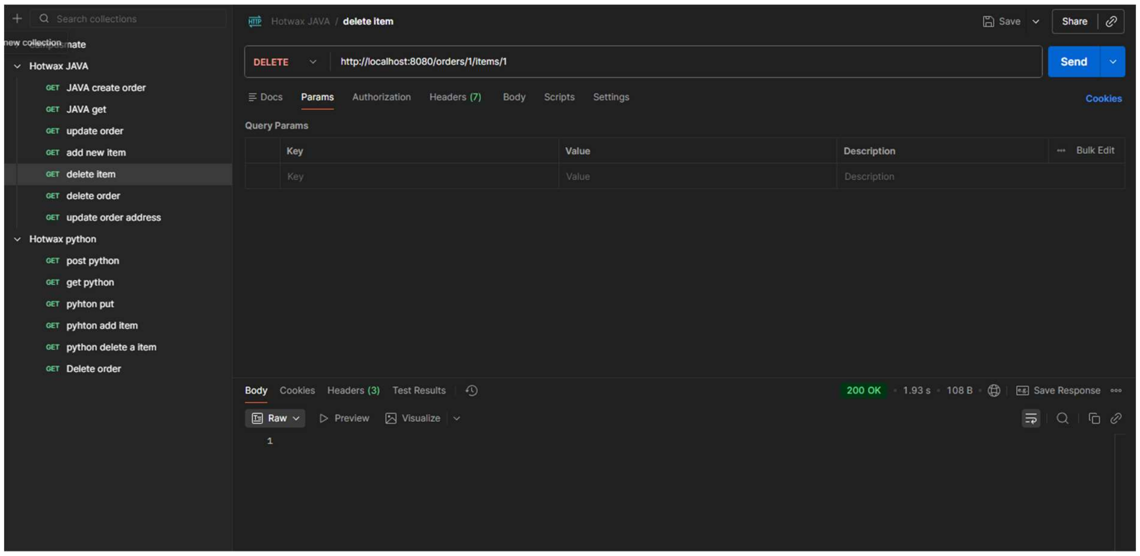
```
{
  "productId": 3,
  "quantity": 1,
  "status": "APPROVED"
}
```

201 Created · 129 ms · 171 B

```
{
    "message": "Item Added"
}
```

**Scenario 5: Delete Order Item**

- **Endpoint:** DELETE /orders/{orderId}/items/{seqId}
- **Description:** Removed a specific item from the order.
- **Status Code:** 200 OK

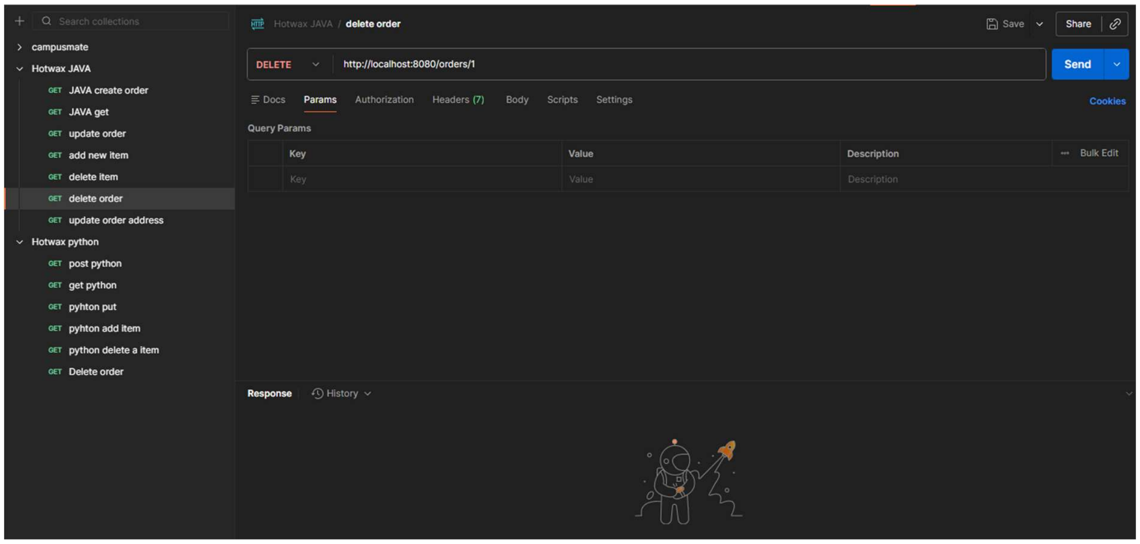

Hotwax JAVA / delete item

DELETE http://localhost:8080/orders/1/items/1

200 OK · 1.93 s · 108 B

**Scenario 6: Delete Entire Order**

- **Endpoint:** DELETE /orders/{orderId}
- **Description:** Deleted the order header. Due to ON DELETE CASCADE configuration in the database, all associated order items were automatically removed.
- **Status Code:** 200 OK


Hotwax JAVA / delete order

DELETE http://localhost:8080/orders/1