Spring Boot REST API for Product Management

Overview

This is a simple RESTful API built using Spring Boot to manage a list of products. The API allows users to perform CRUD operations on products, including creating, retrieving, updating, and deleting products.

Features

- Fetch all products
- Fetch a product by ID
- Create a new product
- Update an existing product
- Delete a product by ID
- Exception handling for invalid operations
- Input validation for product attributes

Technologies Used

- Java Openjdk 23.0.2
- Spring Boot
- Spring Web
- Spring Data JPA
- Maven Version 3.8.8
- MySQL

API Endpoints

Get all products

Endpoint: `GET /api/products`

Response:


```
[
  {
    "id": 1,
    "name": "Laptop",
    "description": "High-performance laptop",
```

```
    "price": 1200.99
  }
]
```

Get a product by ID

Endpoint: `GET /api/products/{id}`

Response:

```
{
  "id": 1,
  "name": "Laptop",
  "description": "High-performance laptop",
  "price": 1200.99
}
```

Create a new product

Endpoint: `POST /api/products`

Request Body:

```
{
  "name": "Smartphone",
  "description": "Latest model smartphone",
  "price": 799.99
}
```

Response:

```
{
  "id": 2,
  "name": "Smartphone",
  "description": "Latest model smartphone",
  "price": 799.99
}
```

Update an existing product

Endpoint: `PUT /api/products/{id}`

Request Body:

```
{
  "name": "Updated Laptop",
  "description": "Updated high-performance laptop",
  "price": 1300.99
}
```

Response:

```
{
  "id": 1,
  "name": "Updated Laptop",
  "description": "Updated high-performance laptop",
  "price": 1300.99
}
```

Delete a product by ID

Endpoint: `DELETE /api/products/{id}`

Response:

204 No Content

Exception Handling

- **404 Not Found:** When trying to retrieve or update a non-existent product.
- **400 Bad Request:** When providing invalid data (e.g., missing name, negative price).

Validation Rules

- **Name:** Cannot be null or empty.
- **Price:** Must be a positive decimal value.

Running the Application

1. Clone the repository.
2. Navigate to the project directory.
3. Run the following command:
   mvn clean package && cd target && java -jar product-api-1.0-SNAPSHOT.jar
4. Access the API at `http://localhost:8080/api/products`.

Testing the API

Using Postman:

Import the provided Postman collection and test the endpoints.

Using CURL:

```
# Create a product
curl --location 'http://localhost:8080/api/products' \
```

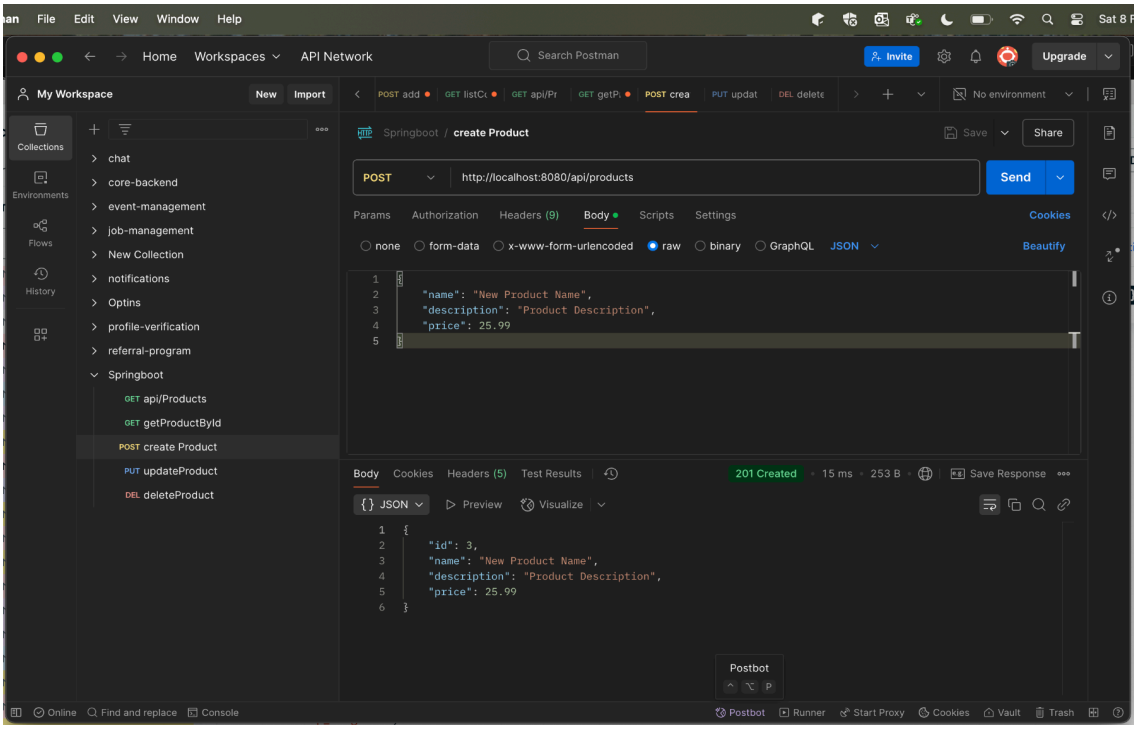
```
--header 'Content-Type: application/json' \
--data '{
"name": "New Product Name"
,
"description": "Product Description"
,
"price": 25.99
}'

# Update a product
curl --location --request PUT 'http://localhost:8080/api/products/1' \
--header 'Content-Type: application/json' \
--data '{
"name": "New CHANGED Product Name"
,
"description": "Product Description CHANGED"
,
"price": 2.99
}'

# Get all products
curl --location 'http://localhost:8080/api/products' \
--data ''

# Get a specific product
curl --location 'http://localhost:8080/api/products/2'

# Delete a product
curl --location --request DELETE 'http://localhost:8080/api/products/2'
```



Database Schema for E-Commerce Platform

User Table

```
CREATE TABLE User (
   UserID INT AUTO_INCREMENT PRIMARY KEY,
   Username VARCHAR(255) UNIQUE NOT NULL,
   Password VARCHAR(255) NOT NULL, -- In a real application, store
password hashes, not plain passwords
   Email VARCHAR(255) UNIQUE NOT NULL,
   FirstName VARCHAR(255),
   LastName VARCHAR(255),
   Address VARCHAR(255),
   IsAdmin BOOLEAN DEFAULT FALSE -- Add a column to differentiate
between normal users and admins.
);
```

Product Table

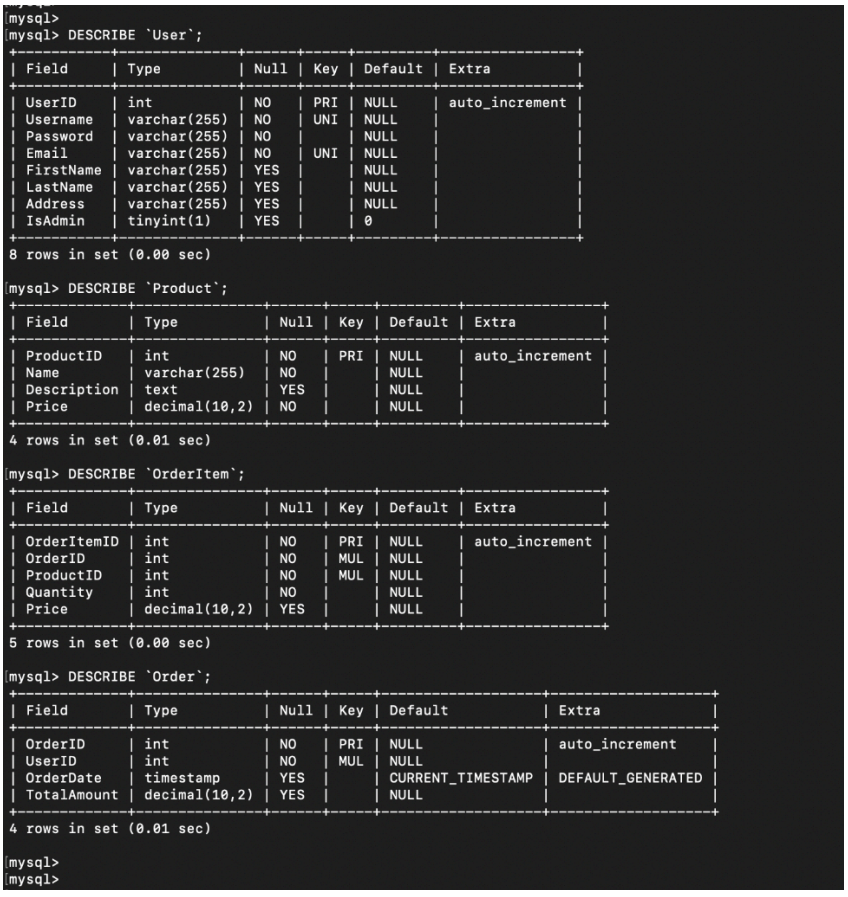
```
CREATE TABLE Product (
   ProductID INT AUTO_INCREMENT PRIMARY KEY,
   Name VARCHAR(255) NOT NULL,
   Description TEXT,
   Price DECIMAL(10, 2) NOT NULL, -- Use DECIMAL for currency
   CHECK (Price > 0) -- Constraint to ensure price is positive
);
```

Order Table

```
CREATE TABLE `Order` (
   OrderID INT AUTO_INCREMENT PRIMARY KEY,
   UserID INT NOT NULL,
   OrderDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   TotalAmount DECIMAL(10, 2),
   FOREIGN KEY (UserID) REFERENCES User(UserID)
);
```

OrderItem Table

```
CREATE TABLE OrderItem (
    OrderItemID INT AUTO_INCREMENT PRIMARY KEY,
    OrderID INT NOT NULL,
    ProductID INT NOT NULL,
    Quantity INT NOT NULL,
    Price DECIMAL(10, 2), -- Price at the time of order (might be
different from current product price)
    FOREIGN KEY (OrderID) REFERENCES `Order`(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
);
```



```
mysql>
mysql> DESCRIBE `User`;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| UserID    | int          | NO   | PRI | NULL    | auto_increment |
| Username  | varchar(255) | NO   | UNI | NULL    |                |
| Password  | varchar(255) | NO   |     | NULL    |                |
| Email     | varchar(255) | NO   | UNI | NULL    |                |
| FirstName | varchar(255) | YES  |     | NULL    |                |
| LastName  | varchar(255) | YES  |     | NULL    |                |
| Address   | varchar(255) | YES  |     | NULL    |                |
| IsAdmin   | tinyint(1)   | YES  |     | 0       |                |
+-----------+--------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)

mysql> DESCRIBE `Product`;
+-------------+---------------+------+-----+---------+----------------+
| Field       | Type          | Null | Key | Default | Extra          |
+-------------+---------------+------+-----+---------+----------------+
| ProductID   | int           | NO   | PRI | NULL    | auto_increment |
| Name        | varchar(255)  | NO   |     | NULL    |                |
| Description | text          | YES  |     | NULL    |                |
| Price       | decimal(10,2) | NO   |     | NULL    |                |
+-------------+---------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)

mysql> DESCRIBE `OrderItem`;
+-------------+---------------+------+-----+---------+----------------+
| Field       | Type          | Null | Key | Default | Extra          |
+-------------+---------------+------+-----+---------+----------------+
| OrderItemID | int           | NO   | PRI | NULL    | auto_increment |
| OrderID     | int           | NO   | MUL | NULL    |                |
| ProductID   | int           | NO   | MUL | NULL    |                |
| Quantity    | int           | NO   |     | NULL    |                |
| Price       | decimal(10,2) | YES  |     | NULL    |                |
+-------------+---------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> DESCRIBE `Order`;
+-------------+---------------+------+-----+-------------------+-------------------+
| Field       | Type          | Null | Key | Default           | Extra             |
+-------------+---------------+------+-----+-------------------+-------------------+
| OrderID     | int           | NO   | PRI | NULL              | auto_increment    |
| UserID      | int           | NO   | MUL | NULL              |                   |
| OrderDate   | timestamp     | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| TotalAmount | decimal(10,2) | YES  |     | NULL              |                   |
+-------------+---------------+------+-----+-------------------+-------------------+
4 rows in set (0.01 sec)

mysql>
mysql>
```

Sample Data

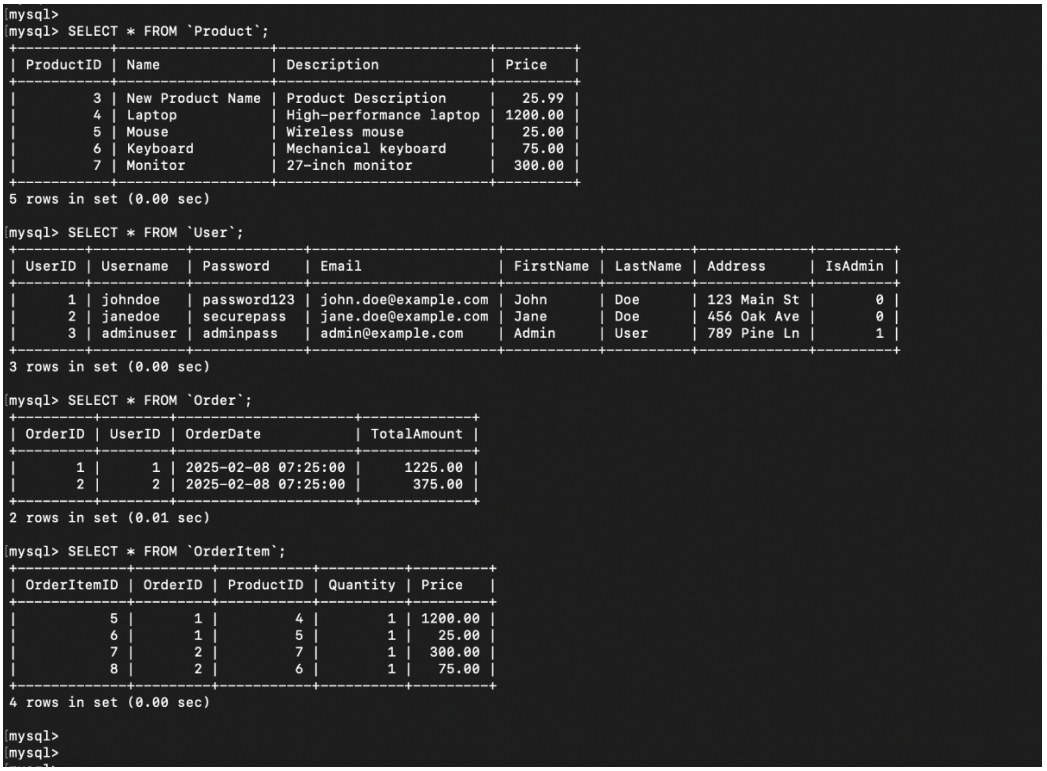
-- Insert data into the User table
INSERT INTO `User` (Username, Password, Email, FirstName, LastName, Address, IsAdmin) VALUES
('johndoe', 'password123', 'john.doe@example.com', 'John', 'Doe', '123 Main St', FALSE),
('janedoe', 'securepass', 'jane.doe@example.com', 'Jane', 'Doe', '456 Oak Ave', FALSE),

('adminuser', 'adminpass', 'admin@example.com', 'Admin', 'User', '789 Pine Ln', TRUE);

-- Insert data into the Product table
INSERT INTO `Product` (Name, Description, Price) VALUES
('Laptop', 'High-performance laptop', 1200.00),
('Mouse', 'Wireless mouse', 25.00),
('Keyboard', 'Mechanical keyboard', 75.00),
('Monitor', '27-inch monitor', 300.00);

-- Insert data into the Order table
INSERT INTO `Order` (UserID, TotalAmount) VALUES
(1, 1225.00), -- John Doe's order (Laptop + Mouse)
(2, 375.00);   -- Jane Doe's order (Monitor + Keyboard)

-- Insert data into the OrderItem table
INSERT INTO OrderItem (OrderID, ProductID, Quantity, Price) VALUES
(1, 4, 1, 1200.00), -- John Doe's order: 1 Laptop (ProductID 4)
(1, 5, 1, 25.00),   -- John Doe's order: 1 Mouse (ProductID 5)
(2, 7, 1, 300.00), -- Jane Doe's order: 1 Monitor (ProductID 7)
(2, 6, 1, 75.00);   -- Jane Doe's order: 1 Keyboard (ProductID 6)



```
mysql>
mysql> SELECT * FROM `Product`;
+-----------+------------------+-------------------------+---------+
| ProductID | Name             | Description             | Price   |
+-----------+------------------+-------------------------+---------+
|         3 | New Product Name | Product Description     |   25.99 |
|         4 | Laptop           | High-performance laptop | 1200.00 |
|         5 | Mouse            | Wireless mouse          |   25.00 |
|         6 | Keyboard         | Mechanical keyboard     |   75.00 |
|         7 | Monitor          | 27-inch monitor         |  300.00 |
+-----------+------------------+-------------------------+---------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM `User`;
+--------+----------+------------+---------------------+-----------+----------+-------------+---------+
| UserID | Username | Password   | Email               | FirstName | LastName | Address     | IsAdmin |
+--------+----------+------------+---------------------+-----------+----------+-------------+---------+
|      1 | johndoe  | password123 | john.doe@example.com | John      | Doe      | 123 Main St |       0 |
|      2 | janedoe  | securepass | jane.doe@example.com | Jane      | Doe      | 456 Oak Ave |       0 |
|      3 | adminuser | adminpass | admin@example.com   | Admin     | User     | 789 Pine Ln |       1 |
+--------+----------+------------+---------------------+-----------+----------+-------------+---------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM `Order`;
+---------+--------+---------------------+-------------+
| OrderID | UserID | OrderDate           | TotalAmount |
+---------+--------+---------------------+-------------+
|       1 |      1 | 2025-02-08 07:25:00 |     1225.00 |
|       2 |      2 | 2025-02-08 07:25:00 |      375.00 |
+---------+--------+---------------------+-------------+
2 rows in set (0.01 sec)

mysql> SELECT * FROM `OrderItem`;
+-------------+---------+-----------+----------+---------+
| OrderItemID | OrderID | ProductID | Quantity | Price   |
+-------------+---------+-----------+----------+---------+
|           5 |       1 |         4 |        1 | 1200.00 |
|           6 |       1 |         5 |        1 |   25.00 |
|           7 |       2 |         7 |        1 |  300.00 |
|           8 |       2 |         6 |        1 |   75.00 |
+-------------+---------+-----------+----------+---------+
4 rows in set (0.00 sec)

mysql>
mysql>
```