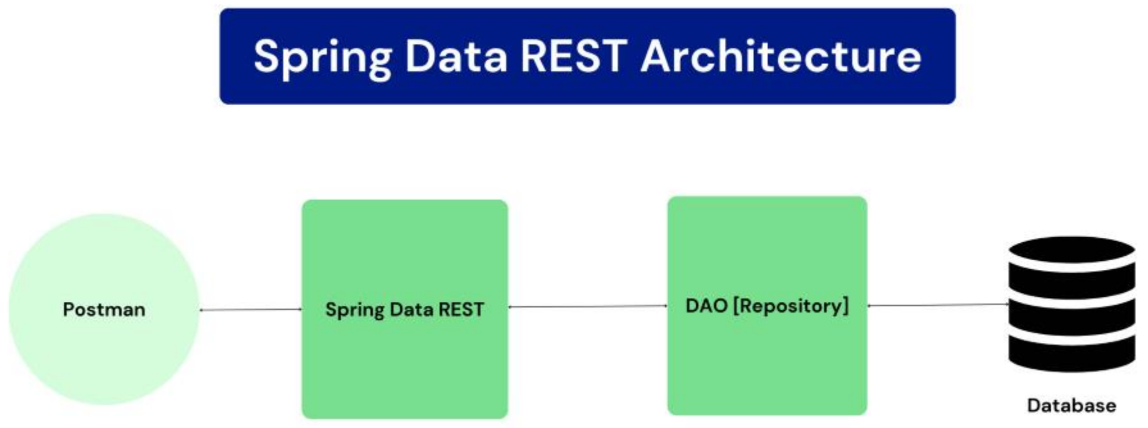
REST API-JPA

MIGUEL ANGEL RAMIREZ JUAREZ
JAVA ACADEMY

SEP/ 06/ 2024

What is a Collection in Java?

This project aims to develop an inventory management system for a logistics company with multiple warehouses. Using a **REST API** built with **Spring Boot** and **JPA**, the system allows warehouse administrators to add, update, delete, and view products in real-time. The data is stored in a **MySQL** database.



Proposed Solution:

A REST API was developed using Spring Boot that interacts with a MySQL database to manage the company's product inventory. The CRUD (Create, Read, Update, Delete) operations were designed to meet the requirements, providing endpoints that allow users to interact with the system efficiently.

System Architecture:

1-

Technologies Used:

Backend: Spring Boot, JPA (without Spring Data)

Database: MySQL

Dependencies: Spring Web, MySQL Driver

Programming Language: Java

2-

Client (Postman/cURL) <---> REST Controller (ProductoController) <---> Service (ProductoService) <---> JPA EntityManager <---> MySQL Database

Implemented Functionalities:

Add Product to Inventory:

Method: POST

Endpoint: /api/inventory

Description: Adds a new product to the inventory.

View All Products:

Method: GET

Endpoint: /api/inventory

Description: Retrieves all products from the inventory.

View Product by ID:

Method: GET

Endpoint: /api/inventory/{id}

Description: Retrieves product details based on the product ID.

Update Product Information:

Method: PUT

Endpoint: /api/inventory/{id}

Description: Updates the product's quantity and warehouse location.

Delete Product:

Method: DELETE

Endpoint: /api/inventory/{id}

Description: Deletes a product from the inventory.

CODE POM:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.3.3</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.hw1</groupId>
    <artifactId>springboot3restapijpacrud</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>springboot3restapijpacrud</name>
    <description>Demo project for Spring Boot</description>
    <url/>
    <licenses>
        <license/>
    </licenses>
    <developers>
        <developer/>
    </developers>
    <scm>
        <connection/>
        <developerConnection/>
```

```xml
        <tag/>
        <url/>
    </scm>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-devtools</artifactId>
            <scope>runtime</scope>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>com.mysql</groupId>
            <artifactId>mysql-connector-j</artifactId>
            <scope>runtime</scope>
        </dependency>
        <dependency>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
                <configuration>
```

```
                    <excludes>
                        <exclude>
                            <groupId>org.projectlombok</groupId>
                            <artifactId>lombok</artifactId>
                        </exclude>
                    </excludes>
                </configuration>
            </plugin>
        </plugins>
    </build>

</project>
```

Expected Results:

The system allows the logistics company to:

Manage its inventory across multiple warehouses.

Add and update products and their quantities in real-time.

Delete outdated products, ensuring accurate inventory management.

Retrieve detailed information on any product via the REST API.

POSTMAN:

- **GET /api/productos**:
  - Method: GET
  - URL: `http://localhost:8080/api/productos`
  - Click "Send" to retrieve all products.
- **GET /api/productos/{id}**:
  - Method: GET
  - URL: `http://localhost:8080/api/productos/{id}`
  - Replace `{id}` with an actual product ID.
  - Click "Send" to retrieve a specific product.
- **POST /api/productos**:

- Method: POST
- URL: `http://localhost:8080/api/productos`
- Body: Select "raw" and "JSON" format, then provide the product details in JSON format, e.g.:

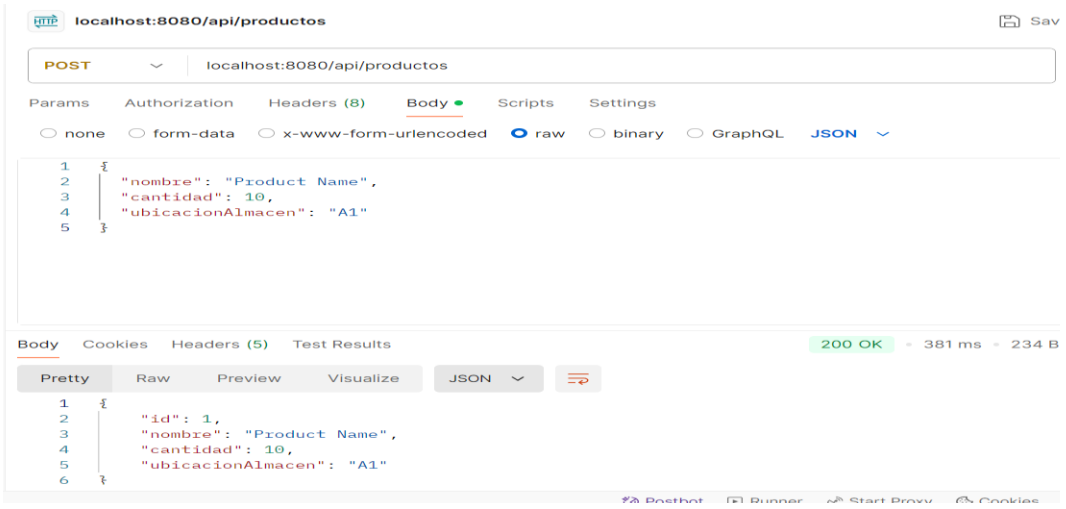
JSON

```
{
  "nombre": "Product Name",
  "cantidad": 10,
  "ubicacionAlmacen": "A1"
}
```

Click "Send" to create a new product.

- **PUT /api/productos/{id}**:
  - Method: PUT
  - URL: `http://localhost:8080/api/productos/{id}`
  - Replace `{id}` with the product ID you want to update.
  - Body: Select "raw" and "JSON" format, then provide the updated product details in JSON format.
  - Click "Send" to update the product.
- **DELETE /api/productos/{id}**:
  - Method: DELETE
  - URL: `http://localhost:8080/api/productos/{id}`
  - Replace `{id}` with the product ID you want to delete.
  - Click "Send" to delete the product.

EXAMPLE OF WORK PUT PROCESS EN POSTMAN:



POST localhost:8080/api/productos

raw JSON

```
{
  "nombre": "Product Name",
  "cantidad": 10,
  "ubicacionAlmacen": "A1"
}
```

200 OK · 381 ms · 234 B

JSON

```
{
    "id": 1,
    "nombre": "Product Name",
    "cantidad": 10,
    "ubicacionAlmacen": "A1"
}
```