Implementing a Full REST Service

Overview

In this lab you’ll enhance the REST service that you started implementing in the previous lab, so that it allows users to insert, update, and delete product suggestions.

IntelliJ starter project

If you’re happy to continue where you left off in the previous lab, use the following project:

- **`student\student-online-retailer`**

If you’d prefer a fresh start, use the solution project from the previous lab instead:

- **`solutions\solution-simple-rest-services`**

IntelliJ solution project

The solution project for this lab is located here:

- **`solutions\solution-full-rest-services`**

Roadmap

There are 3 exercises in this lab, of which the final exercise is “if time permits”. Here is a brief summary of the tasks you will perform in each exercise; more detailed instructions follow later:

1. Adding Swagger support in the pom file
2. Implementing REST service endpoints to insert and delete data
3. (If time permits) Implementing REST service endpoints to modify data

Exercise 1: Adding Swagger support in the pom file

When you implement REST endpoints to support POST, PUT, or DELETE requests, you need some way to be able to test these endpoints. Swagger is a convenient tool to help you do this, and Spring Boot makes it easy to integrate Swagger support into your application – just add the following dependency in your pom file:

```
<dependency>
  <groupId>org.springdoc</groupId>
  <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
  <version>2.0.2</version>
</dependency>
```

As it happens, you’ll also need to add the following dependency, which enables Spring Boot to validate data destined for your persistent entities:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-validation</artifactId>
</dependency>
```

Exercise 2: Implementing REST service endpoints to insert and delete data

Add endpoints in your **`ProductSuggestionController`** class to let the user perform the following tasks:

- Insert a new product suggestion
- Delete all product suggestions

Run the application, then use the Swagger UI to test these endpoints.

Exercise 3 (If time permits): Implementing REST service endpoints to modify data

Add endpoints in **`ProductSuggestionController`** to let the user modify the price and sales for a product suggestion, and to increase by 10% the price of all products that exceed a certain number of sales per year. We suggest the following endpoint URLs:

```
productSuggestions/modifyPrice/1?newPrice=10.99
productSuggestions/modifySales/1?newSales=12345
productSuggestions/increasePriceForPopularProducts
```

Run the application again, and use the Swagger UI to test these new endpoints.