Implementing a Simple REST Service

Overview

In this lab you'll implement a REST service to let the user query product suggestions in your application. Later in the course, you'll enhance the REST service to let the user modify the product suggestions as well.

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-spring-data-repositories`**

IntelliJ solution project

The solution project for this lab is located here:

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

Roadmap

There are 3 exercises in this lab. Here is a brief summary of the tasks you will perform in each exercise; more detailed instructions follow later:

1. Adding XML serialization support in the pom file
2. Defining a REST controller class
3. Implementing endpoints to query data

Exercise 1: Adding XML serialization support in the pom file

Spring Boot auto-configuration automatically supports JSON serialization in a web application, because JSON is the preferred data format for REST these days. However, many systems in the real world still need to support XML serialization…

Spring Boot auto-configuration doesn't automatically support XML serialization – you must add the following dependency to your pom file (see the demo application for details, if you need a reminder):

- **`jackson-dataformat-xml`**

Exercise 2: Defining a REST controller class

Define a REST controller class named **`ProductSuggestionController`**. Specify a base URL of **`/productSuggestions`** for the REST controller. Also enable cross-origin requests for the REST controller.

Exercise 3: Implementing endpoints to query data

Autowire a **`ProductSuggestionCrudRepository`** bean into your REST controller class, and implement two endpoints as follows:

- Get all product suggestions
- Get a product suggestion with a particular id

Run the application, then open a browser and ping the two endpoints. Verify you get the correct data back (the data that you see will depend on whether you've seeded the database with any sample data, and on the code in your **`Application`** class to create and modify some records).