**Exercise 1: Online Bookstore - Setting Up RESTful Services**

**Business Scenario:**

You have been tasked with developing a RESTful service for an online bookstore. This service will manage various entities such as books, authors, and customers.

**1. Setting Up the Spring Boot Project**

**1.1 Project Initialization:**

* **Project Name:** BookStoreAPI
* **Dependencies Added:**
  + **Spring Web:** Essential for building web applications, including RESTful services.
  + **Spring Boot DevTools:** Enhances the development experience by enabling automatic restarts, live reload, and configurations that make it easier to develop applications.
  + **Lombok:** Simplifies the coding by reducing boilerplate code like getters, setters, toString, and constructors.

**1.2 Steps to Initialize the Project:**

1. **Navigate to Spring Initializr:**
   * Go to [Spring Initializr](https://start.spring.io/).
2. **Project Configuration:**
   * **Project:** Maven Project
   * **Language:** Java
   * **Spring Boot Version:** 3.3.2
   * **Group:** com.bookstore
   * **Artifact:** BookstoreAPI
   * **Name:** BookstoreAPI
   * **Packaging:** Jar
   * **Java Version:** 17
3. **Adding Dependencies:**
   * Search for and add the following dependencies: Spring Web, Spring Boot DevTools, and Lombok.
4. **Generate the Project:**
   * Click on the "Generate" button to download the project as a zip file.
5. **Importing the Project:**
   * Extracted the zip file and imported the project into Eclipse IDE.

**1.3 Familiarizing with Project Structure:** Upon successfully importing the project, noticed the following standard directory structure:

* **src/main/java:** Contains the main application code.
* **src/main/resources:** Holds the application properties and other static resources.
* **src/test/java:** Contains the unit test cases.

Key folders/files:

* **BookstoreAPIApplication.java:** The entry point of the application.
* **application.properties:** Used for externalized configuration.
* **pom.xml:** The Maven configuration file where dependencies are managed.

**2. Exploring and Documenting New Features in Spring Boot 3**

**2.1 What's New in Spring Boot 3:**

Spring Boot 3 introduced several new features and enhancements. Some of the notable updates include:

* **Java 17 Baseline:** Spring Boot 3 requires a minimum of Java 17, ensuring modern language features and enhancements.
* **GraalVM Native Image Support:** Spring Boot 3 provides first-class support for building native images with GraalVM, offering faster startup times and lower memory consumption.
* **Enhanced Observability:** Spring Boot 3 introduced new features for better observability, including improved logging, metrics, and tracing integration.
* **Kubernetes Integration:** Out-of-the-box support for Kubernetes deployment, making it easier to run Spring Boot applications in a Kubernetes environment.
* **New Actuator Endpoints:** Additional actuator endpoints for monitoring and management tasks.
* **Better Dependency Management:** Updated dependencies and better alignment with the latest Spring Framework 6.x features.