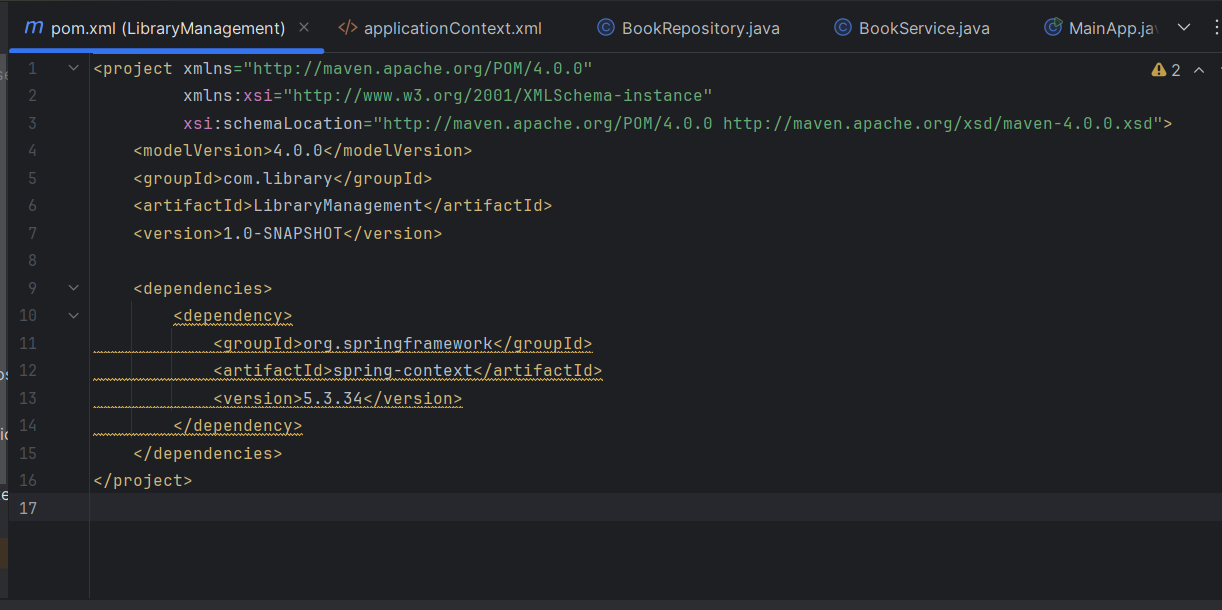
**WEEK 3**

**Spring Core and Maven**

**Exercise 1: Configuring a Basic Spring Application**

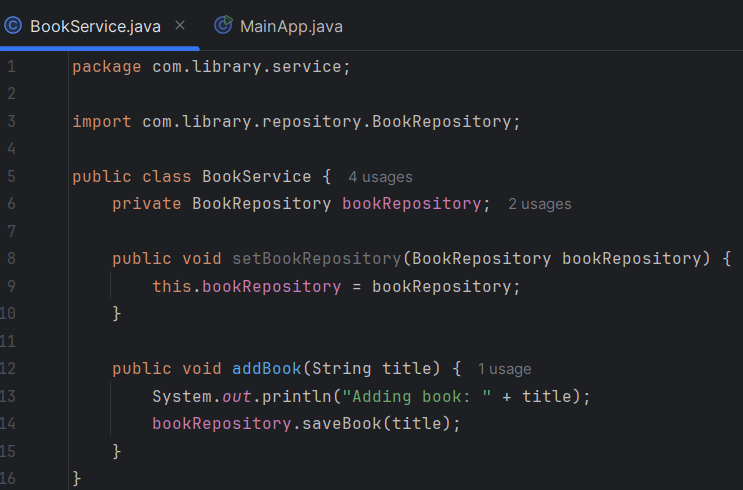
**Scenario:**

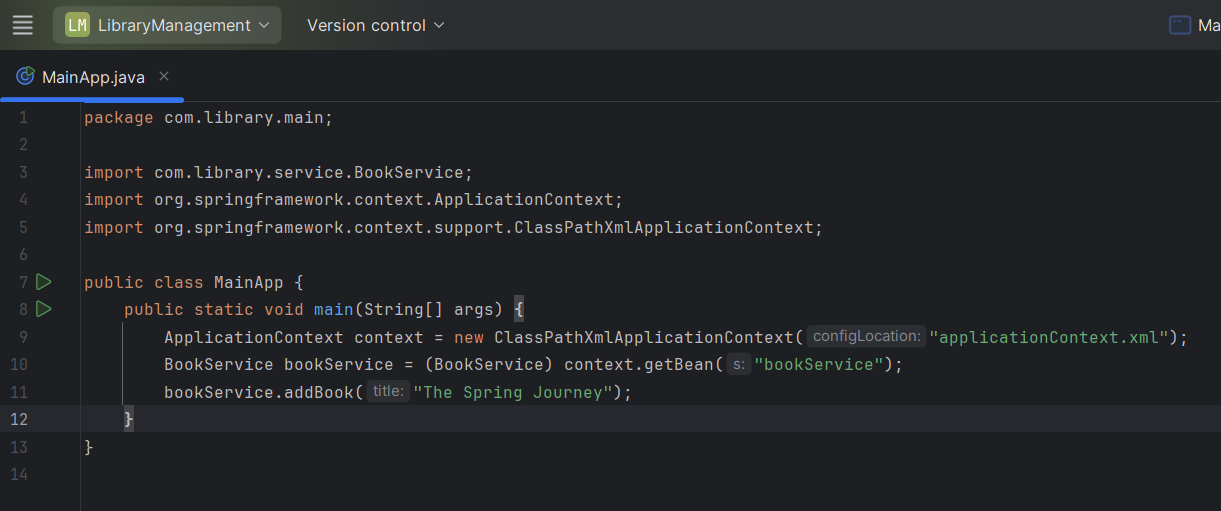
Your company is developing a web application for managing a library. You need to use the Spring Framework to handle the backend operations.

****

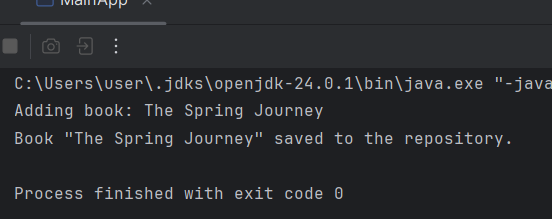
****

****

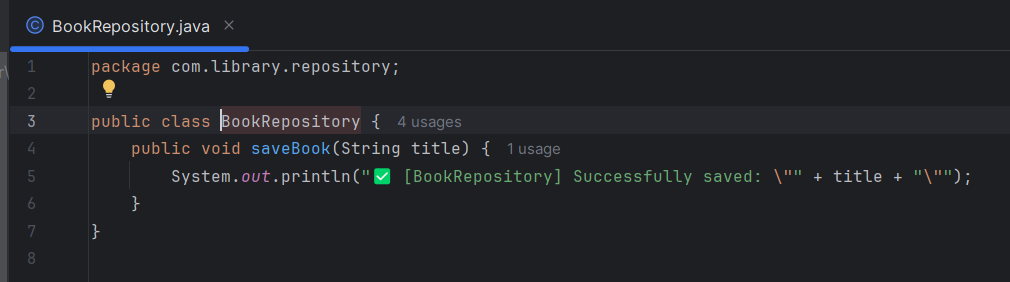
****

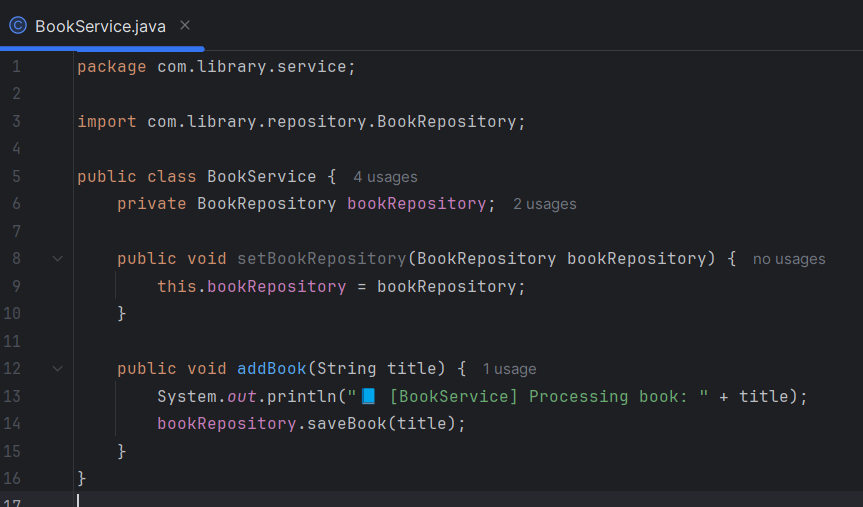
****

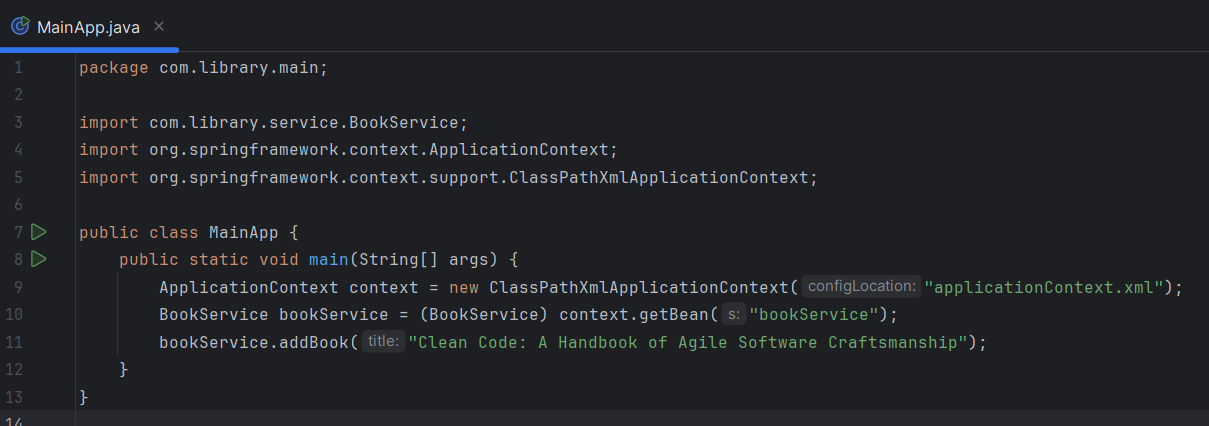
**Output**

****

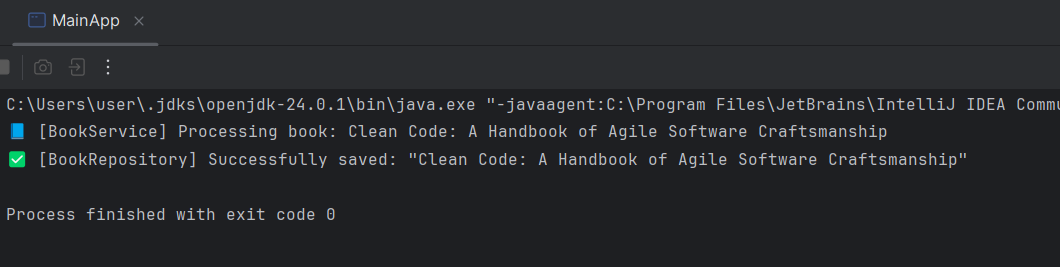
**Exercise 2: Implementing Dependency Injection**  
**Scenario:** In the library management application, you need to manage the dependencies between the BookService and BookRepository classes using Spring's IoC and DI.

****

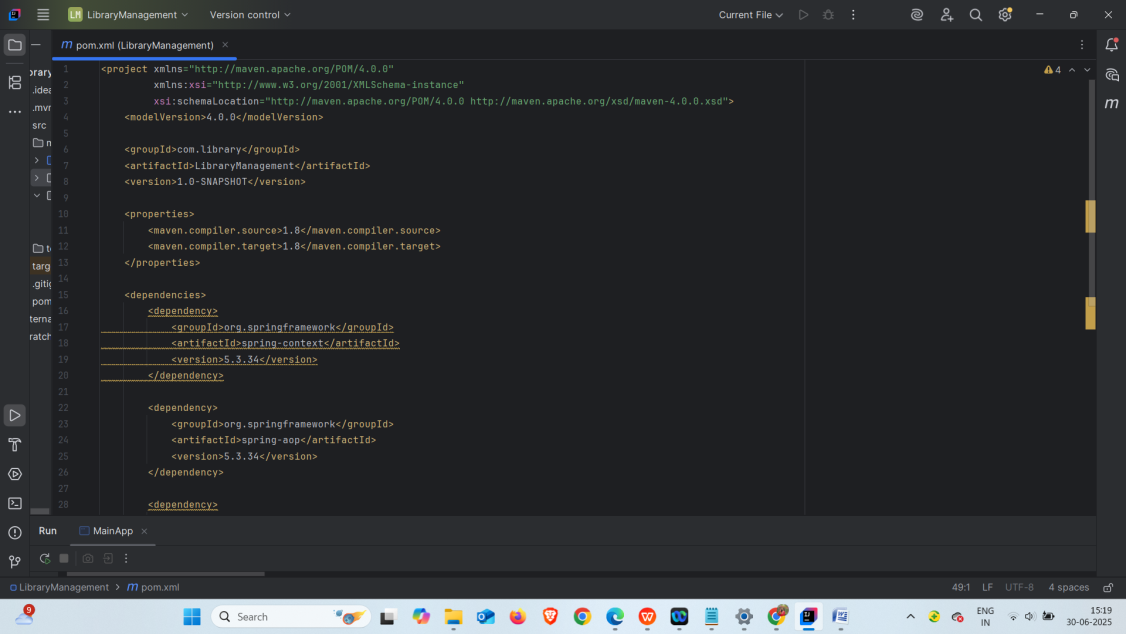
****

****

**OUTPUT**

****

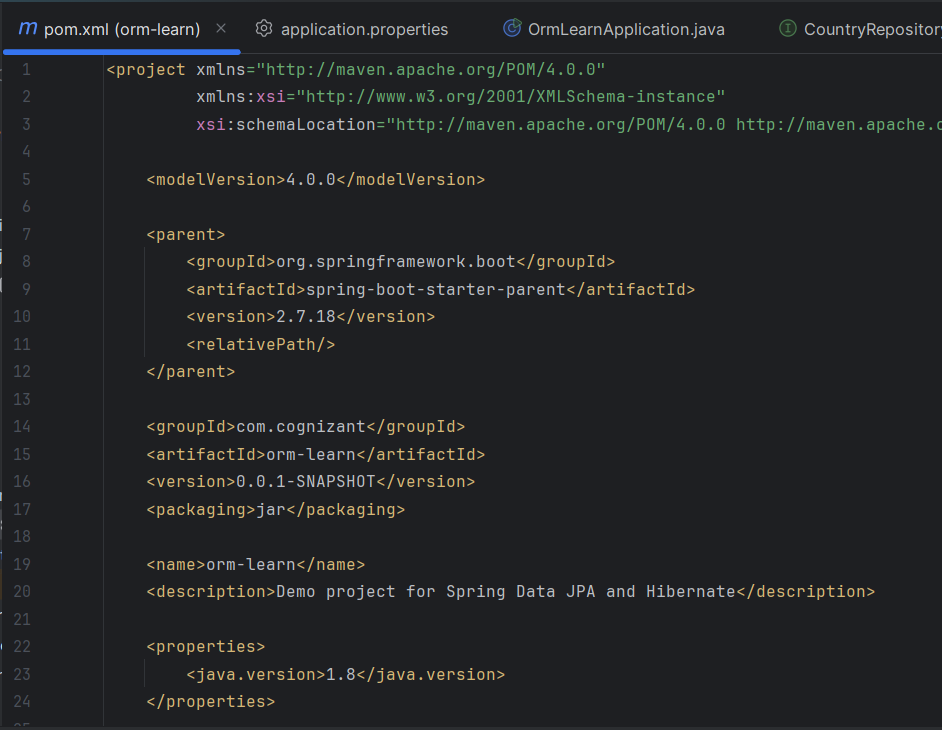
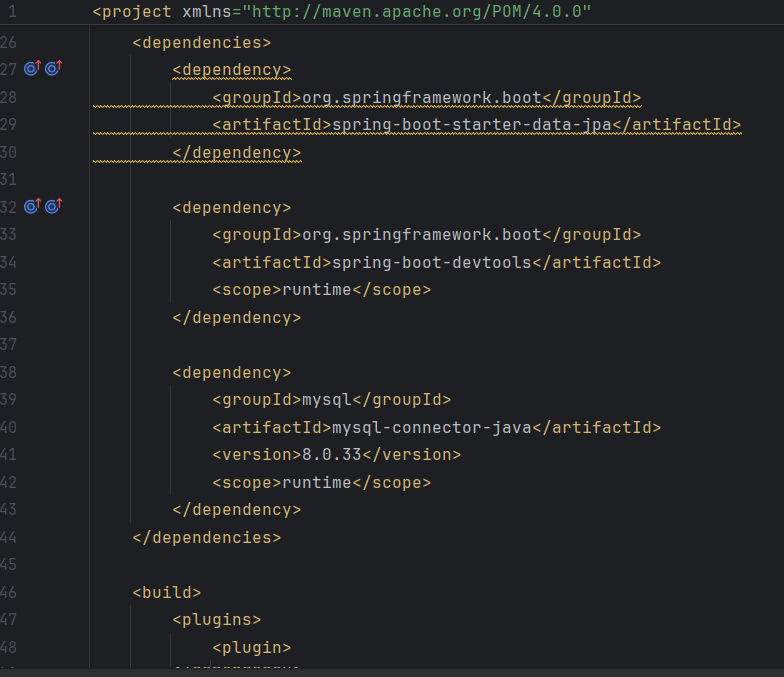
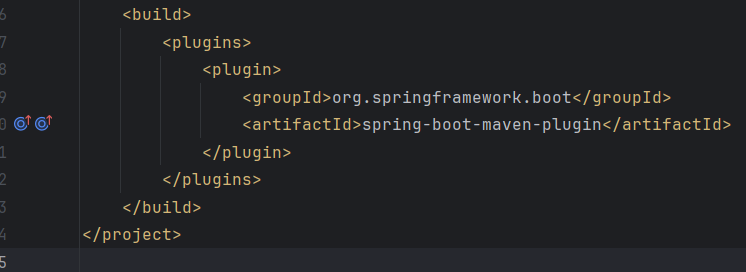
**Exercise 4: Creating and Configuring a Maven Project**  
**Scenario:** You need to set up a new Maven project for the library management application and add Spring dependencies.

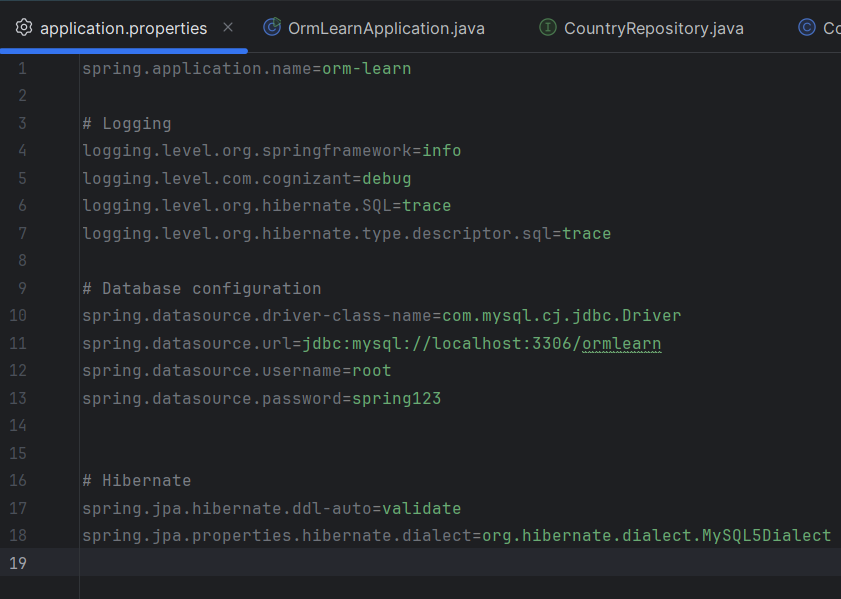
****

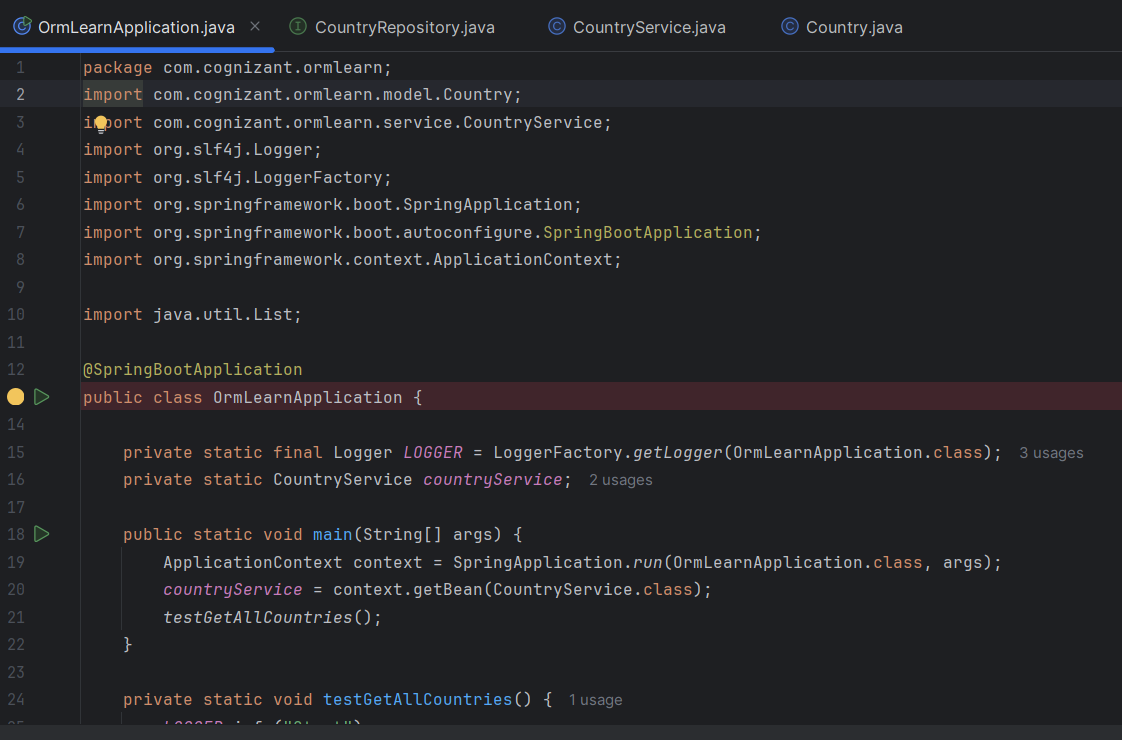
**Spring Data JPA with Spring Boot, Hibernate**

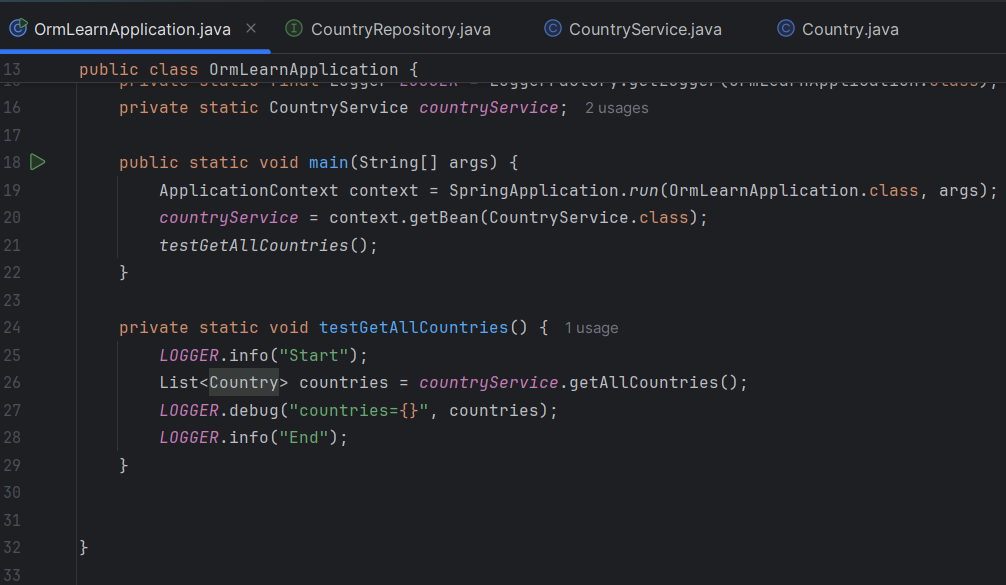
**Hands on 1**

**Spring Data JPA - Quick Example**

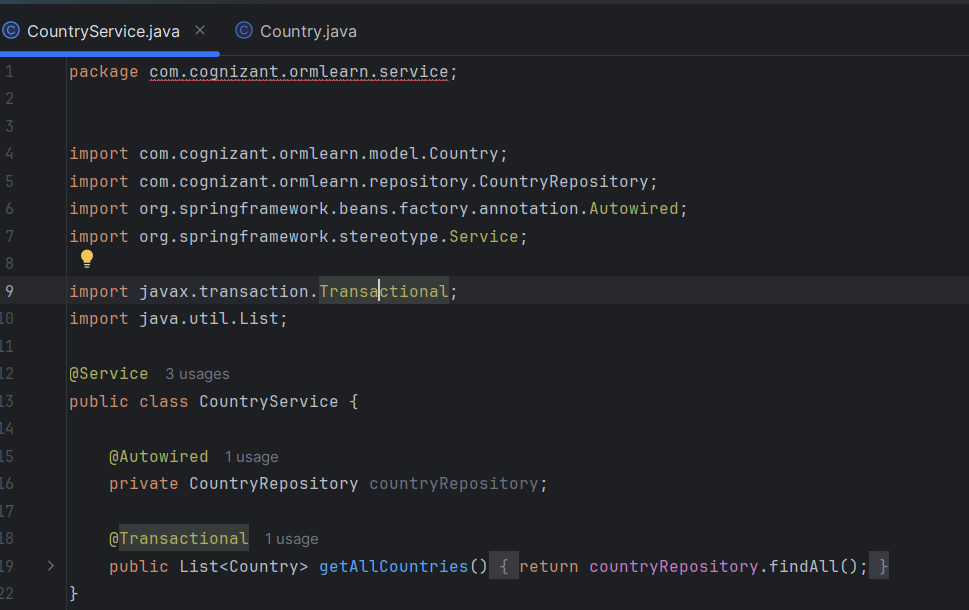
  
  




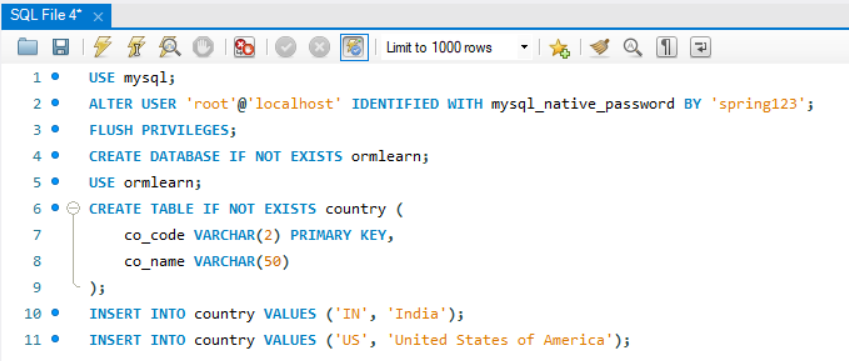


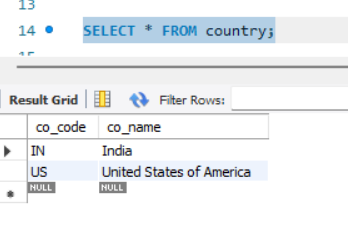


****

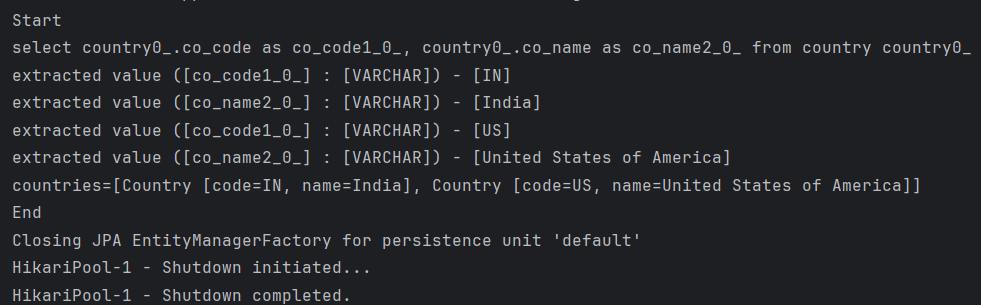
****

****

****

****

**OUTPUT**

****

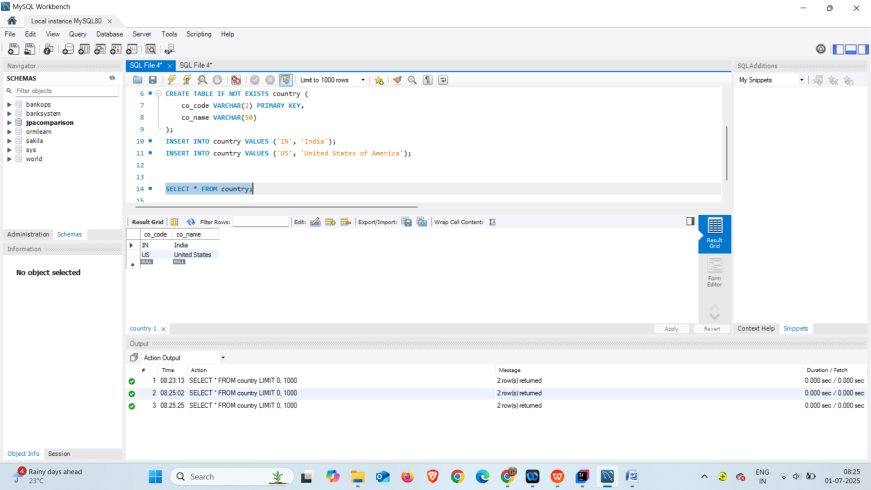
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | **Feature / Term** | **JPA** | **Hibernate** | **Spring Data JPA** | | --- | --- | --- | --- | | **Type** | Specification / Interface | Implementation (ORM framework) | Abstraction Layer built on top of JPA and Hibernate | | **What it is** | A Java specification for ORM | A popular ORM implementation of the JPA standard | Part of Spring; makes working with JPA/Hibernate easier | | **Provided by** | Java EE / Jakarta EE | Red Hat | Spring Framework | | **Configuration** | Requires manual setup and code | Requires manual configuration but richer feature set | Minimal config via application.properties | | **Boilerplate code** | Needs to write all DAO classes and queries manually | Same as JPA but allows HQL and Criteria API | Eliminates boilerplate using JpaRepository, @Query, etc. | | **Query Language** | JPQL | HQL (Hibernate Query Language) | JPQL or native SQL, with support for method name queries | | **Use case** | When you want ORM portability | When you want full control and features | When you want fast development with minimal code | |

**Hands on 4**

**Difference between JPA, Hibernate and Spring Data JPA**

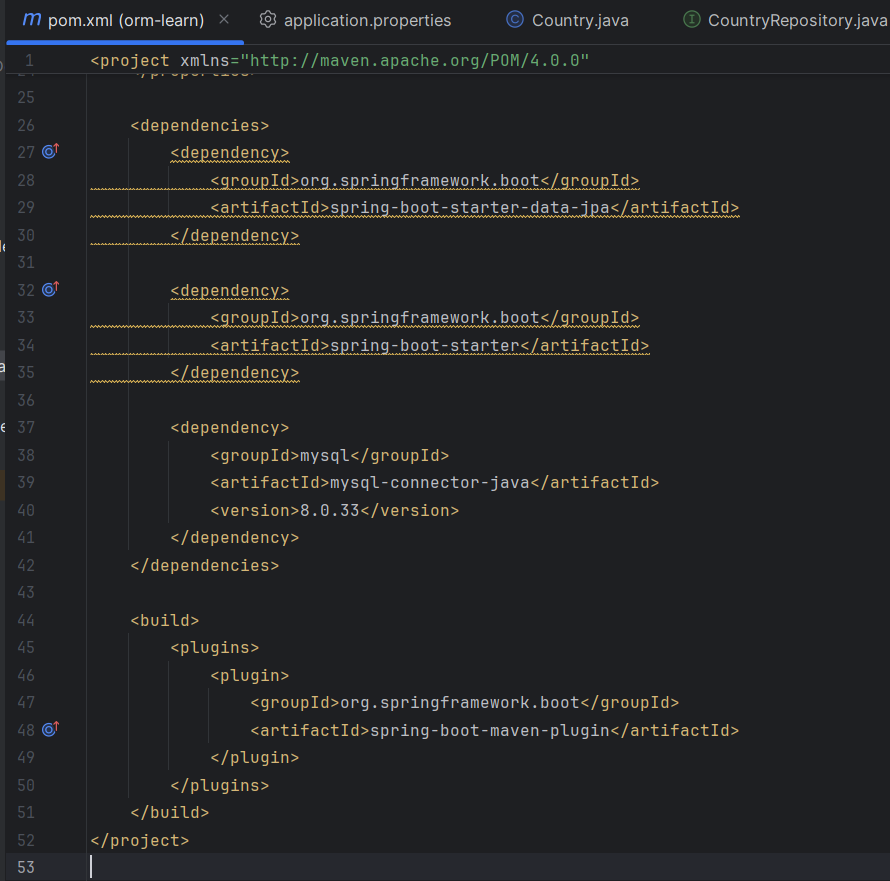
We'll create a **simple mini project** with the same functionality implemented using:

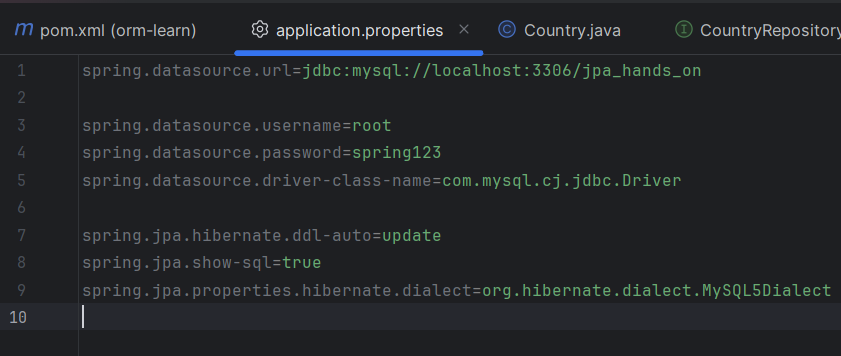
1. **Pure JPA (with EntityManager)**
2. **Pure Hibernate (with SessionFactory)**
3. **Spring Data JPA (with JpaRepository)**

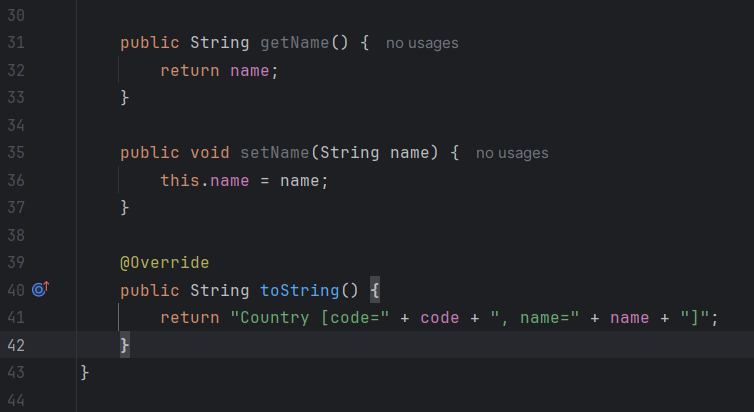


**Hands on 7**

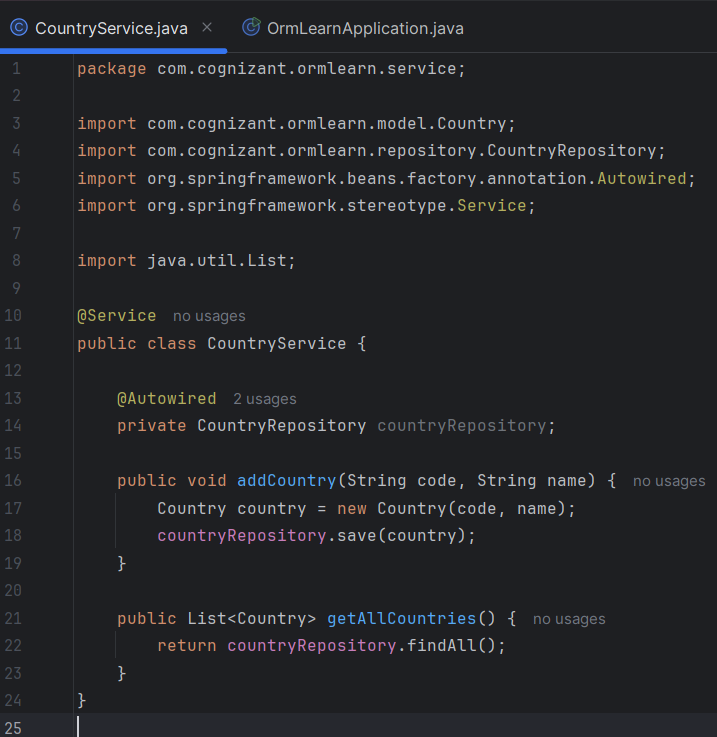
**Add a new country**

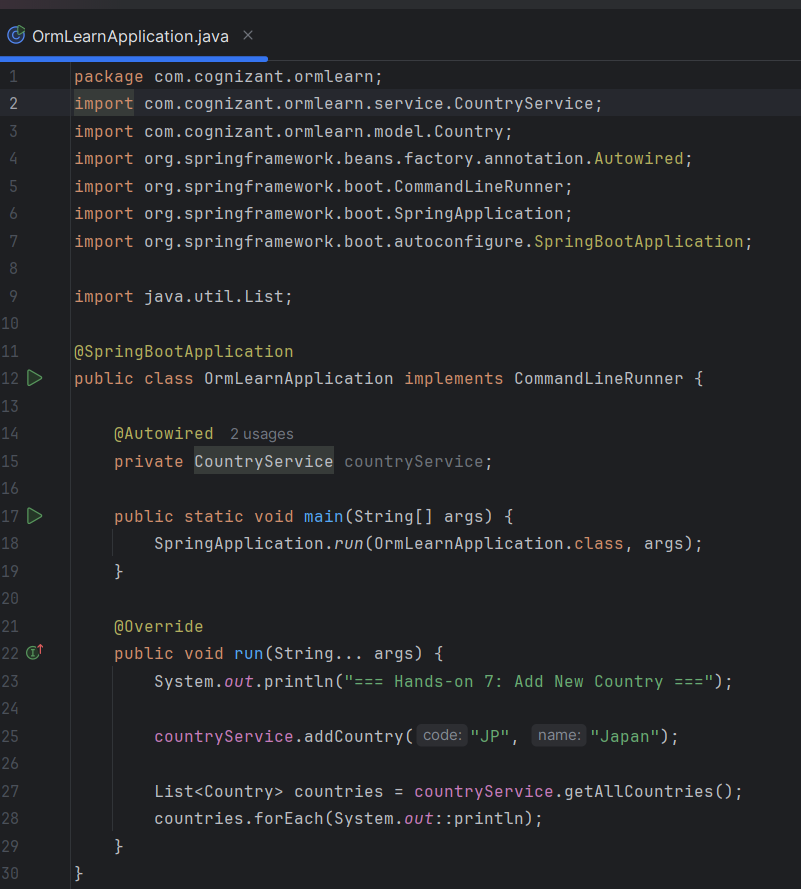
****

****

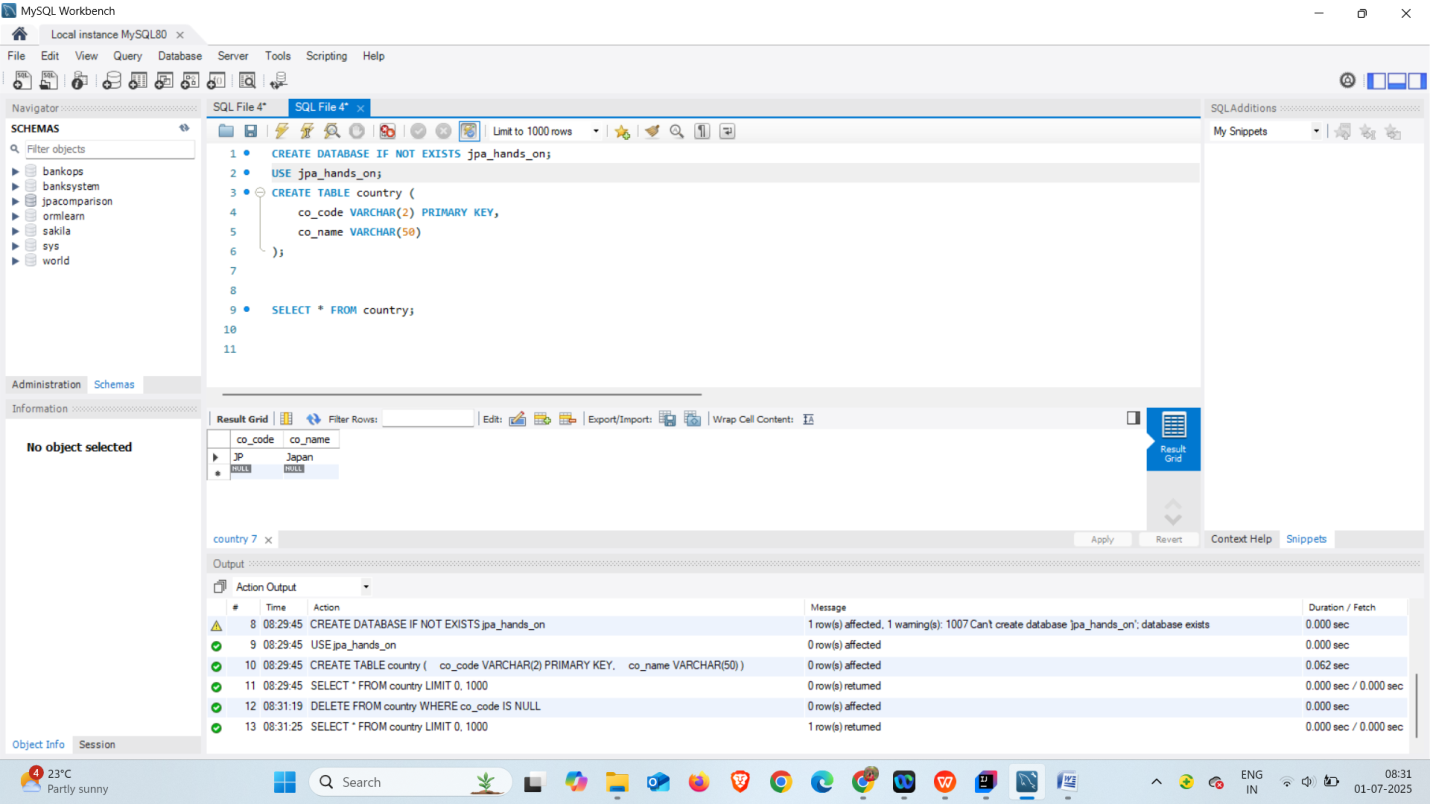
**  
**

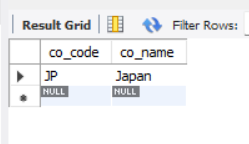
****

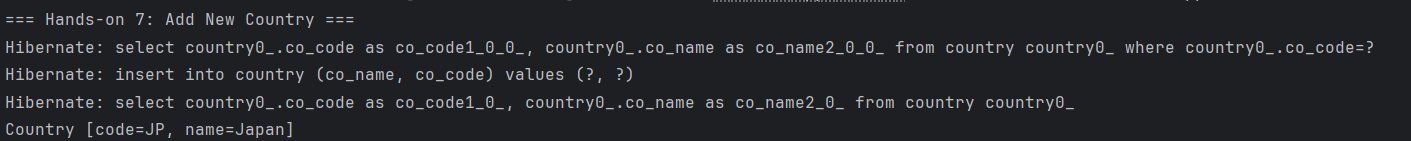
****

****

**OUTPUT**

****

****

****