**Module 2 : Spring Data JPA with Spring Boot, Hibernate**

# **Hands on 1: Spring Data JPA - Quick Example**

**Objective:**

* Understand how to implement a simple Spring Boot application with Spring Data JPA.
* Learn how to configure MySQL and interact with a database table using repository methods.

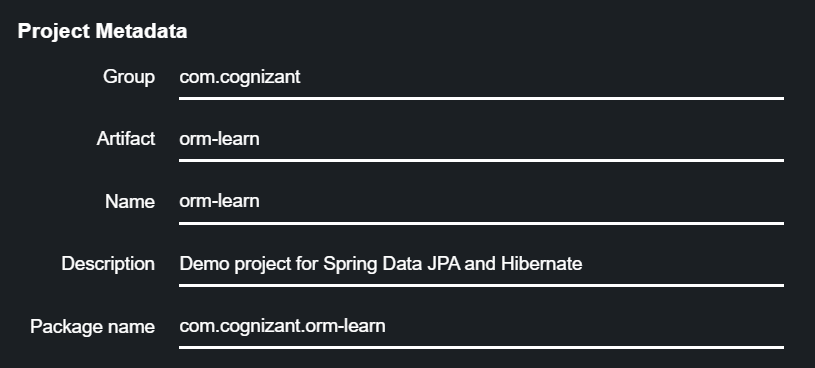
**Software Prerequisites:**

* **MySQL Server 8.0**
* **MySQL Workbench 8**
* **Eclipse IDE for Enterprise Java Developers (2019-03 R)**
* **Maven 3.6.2**

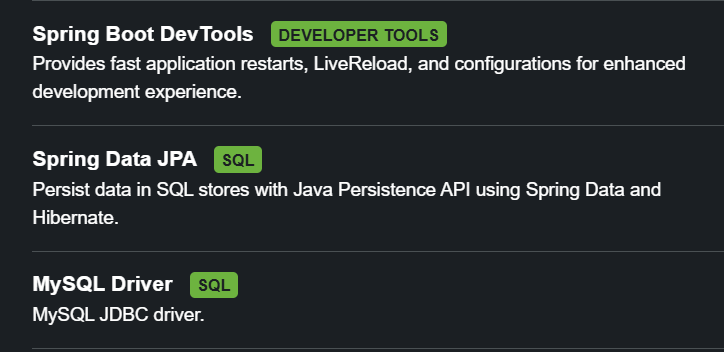
**Procedure:**

**Step 1: Spring Boot Project Creation**

* **Use Spring Initializr (**[**https://start.spring.io/**](https://start.spring.io/)**)**
* **Group:** com.cognizant
* **Artifact:** orb-learn
* **Description:** Demo project for Spring Data JPA and Hibernate



* **Dependencies:** Spring Boot DevTools, Spring Data JPA, MySQL Driver



**Step 2: Import Project in Eclipse**

* Extract and import as Maven Project**: File > Import > Maven > Existing Maven Projects.**

**Step 3: Database Setup**

* **Create schema:**

****

* **Configure database in application.properties**

**# Spring Framework and application log**

**logging.level.org.springframework=info**

**logging.level.com.cognizant=debug**

**# Hibernate logs for displaying executed SQL, input and output**

**logging.level.org.hibernate.SQL=trace**

**logging.level.org.hibernate.type.descriptor.sql=trace**

**# Log pattern**

**logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %-20.20thread %5p %-25.25logger{25} %25M %4L %m%n**

**# Database configuration**

**spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver**

**spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn**

**spring.datasource.username=root**

**spring.datasource.password=root**

**# Hibernate configuration**

**spring.jpa.hibernate.ddl-auto=validate**

**spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect**

**Step 4: Create the Country Table & Add Sample Data**

* **Add some sample data to try and execute.**

**Step 5: Build the Maven Project**

****

**Step 6: Application Code**

* **Persistence Layer (Country.java)**
  + **Annotate with @Entity, @Table, @Column, @Id**
* **Repository Interface**
  + **CountryRepository extends JpaRepository**
* **Service Layer**
  + **CountryService with @Transactional method getAllCountries()**

**Step 7: Testing in Main Class**

* **Autowire CountryService**
* **Add a method testGetAllCountries() to call service and log data.**

**Step 8: Run and Verify**

* **Execute main() and verify data is printed from MySQL.**

# **Hands on 4: Differences between JPA, Hibernate, and Spring Data JPA**

**Objective:**

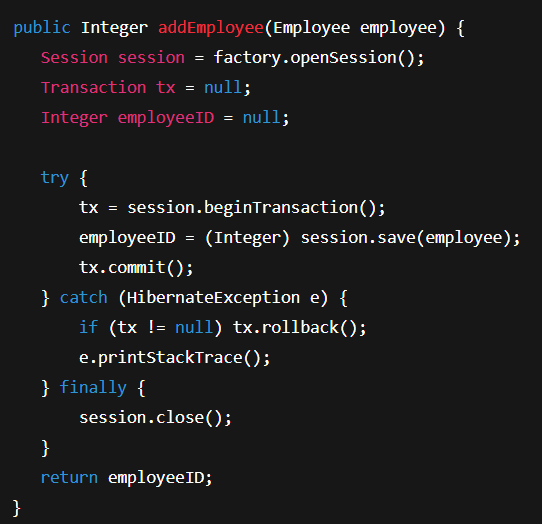
* Understand the conceptual difference between JPA, Hibernate, and Spring Data JPA.
* Compare **boilerplate code** of Hibernate and simplified implementation using Spring Data JPA.

**Conceptual Breakdown:**

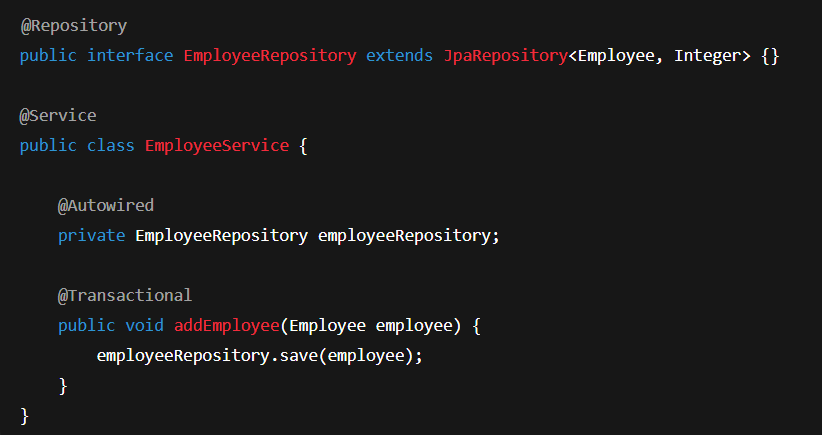
1. **JPA (Java Persistence API)**
   * **Specification (JSR 338) for database operations**
   * **No implementation (just guidelines)**
   * **Defines annotations like @Entity, @Table etc.**
2. **Hibernate**
   * **ORM tool implementing the JPA specification**
   * **Manages database connections, sessions, and transactions**
   * **Requires boilerplate code for session management**
3. **Spring Data JPA**
   * **Abstraction over JPA and Hibernate**
   * **Removes boilerplate code**
   * **Provides CRUD repositories**
   * **Spring handles transactions automatically**

**Example Comparison**

* **Hibernate Example:**

****

* **Spring Data JPA Example:**

****

**Key Differences:**

| **Feature** | **JPA** | **Hiberate** | **Spring Data JPA** |
| --- | --- | --- | --- |
| **Type** | Specification | Implementation (ORM) | Abstraction Framework |
| **Boilerplate Code** | Not applicable | YES | Minimal |
| **Transactions** | Not applicable | Manual or Automatic | Automatic |
| **API Type** | API Spec | ORM Framework | Spring Project |
| **Focus** | Defines Contract | Implements ORM | Simplified CRUD operations |
| **Example Classes** | EntityManager, Query | Session, Trasaction | JpaRepository, CrudRepository |

**Key Takeaways**

* **Hibernate implements JPA.**
* **Spring Data JPA simplifies Hibernate/JPA by removing repetitive code.**
* **Spring handles session and transaction management seamlessly.**