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

**1.Spring Data JPA - Quick Example**

What is Spring Data JPA?

Spring Data JPA is a part of the Spring Data family. It simplifies data access layers and integrates JPA (Java Persistence API) into Spring applications.

* It eliminates boilerplate code for database operations like CRUD (Create, Read, Update, Delete).
* It works on top of Hibernate, which is the most popular JPA implementation.

**Quick Example:**

**Package: src/main/java/com/example/employeeapp/**

**Class : Employee.java**

package com.example.employeeapp;

import jakarta.persistence.\*;

@Entity

public class Employee {

@Id

@GeneratedValue

private Long id;

private String name;

public Employee() {}

public Employee(Long id, String name) {

this.id = id;

this.name = name;

}

// Getters and Setters

}

**Class: EmployeeRepository.java**

package com.example.employeeapp;

import org.springframework.data.jpa.repository.JpaRepository;

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

}

**Class: EmployeeApp.java**

package com.example.employeeapp;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class EmployeeApp implements CommandLineRunner {

@Autowired

private EmployeeRepository repo;

public static void main (String[] args) {

SpringApplication.run(EmployeeApp.class, args);

}

@Override

public void run (String... args) {

repo. save (new Employee (null, "Alice"));

repo. save (new Employee (null, "Bob"));

repo. findAll (). forEach (e -> System.out.println(e. getName ()));

}

}

**File: src/main/resources/application.properties**

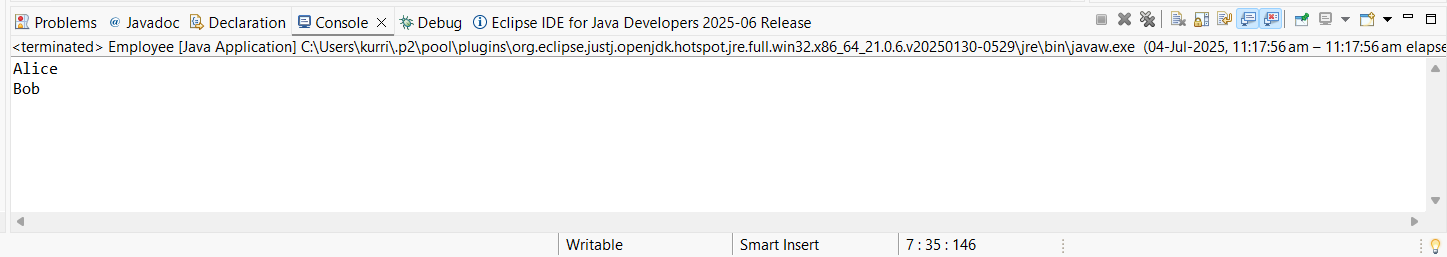
spring.datasource.url=jdbc:h2:mem:testdb

spring.jpa.hibernate.ddl-auto=create

spring.h2.console.enabled=true

spring.jpa.show-sql=true

**Output:**

****

**2. Difference between JPA, Hibernate and Spring Data JPA?**

**1. JPA (Java Persistence API)**

* JPA is a Java specification (interface) for ORM (Object Relational Mapping).
* It defines how Java objects should map to database tables.
* JPA does not provide implementation — it’s just a standard.

Example Feature:

* @Entity, @Id, @GeneratedValue, EntityManager

A set of rules or blueprint for ORM, defined by Java.

**2. Hibernate**

* Hibernate is a popular implementation of JPA.
* It provides the actual code and engine to make JPA work.
* It can also use extra features beyond JPA (like caching, criteria queries).

Example Feature:

* SessionFactory, Criteria, custom caching

A tool that follows the JPA blueprint and adds more features.

**3. Spring Data JPA**

* Spring Data JPA is a Spring Framework module that makes working with JPA (like Hibernate) easier.
* It generates JPA code for you, like basic CRUD, queries, pagination, etc.

Example Feature:

* JpaRepository, findByName(), @Query

A helper layer on top of JPA + Hibernate that reduces boilerplate code.