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

**Code:**

**pom.xml**

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

**Book.java**

package com.library.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

@Entity

public class Book {

@Id

private Long id;

private String title;

public Book() {}

public Book(Long id, String title) {

this.id = id;

this.title = title;

}

public Long getId() { return id; }

public void setId(Long id) { this.id = id; }

public String getTitle() { return title; }

public void setTitle(String title) { this.title = title; }

}

**BookRepository.java**

package com.library.repository;

import org.springframework.data.repository.CrudRepository;

import com.library.model.Book;

public interface BookRepository extends CrudRepository<Book, Long> {

}

**LibraryApp.java**

package com.library;

import com.library.model.Book;

import com.library.repository.BookRepository;

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 LibraryApp implements CommandLineRunner {

@Autowired

private BookRepository bookRepo;

public static void main(String[] args) {

SpringApplication.run(LibraryApp.class, args);

}

@Override

public void run(String... args) {

bookRepo.save(new Book(1L, "Spring in Action"));

bookRepo.save(new Book(2L, "Effective Java"));

bookRepo.findAll().forEach(book ->

System.out.println(book.getId() + " - " + book.getTitle())

);

}

}

**Application.properties**

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

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.hibernate.ddl-auto=update

**Output:**

1 - Spring in Action

2 - Effective Java

[INFO] BUILD SUCCESS

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

**JPA:**

JPA (Java Persistence API) is a specification that defines a set of interfaces and annotations for working with relational databases in Java applications. JPA provides a common API for ORM frameworks like Hibernate, EclipseLink, and OpenJPA making it easier task for us to change frameworks instead of changing code.

**Example is as follows:**

import javax.persistence.\*;

public class JPAExample {

public static void main(String[] args) {

EntityManagerFactory emf = Persistence.createEntityManagerFactory("myPU");

EntityManager em = emf.createEntityManager();

em.getTransaction().begin();

Book book = new Book(1L, "JPA in Action");

em.persist(book);

em.getTransaction().commit();

em.close();

emf.close();

}

}

**Hibernate:**

Hibernate is an Object-Relational Mapping (ORM) framework that provides a high-level API for interacting with relational databases. It allows us to map Java objects to database tables and perform database operations using a high-level API, instead of writing low-level SQL code.

**Example is**

import org.hibernate.\*;

import org.hibernate.cfg.Configuration;

public class HibernateExample {

public static void main(String[] args) {

SessionFactory factory = new Configuration().configure().buildSessionFactory();

Session session = factory.openSession();

Transaction tx = session.beginTransaction();

Book book = new Book(2L, "Hibernate in Action");

session.save(book);

tx.commit();

session.close();

factory.close();

}

}

**Spring data JPA:**

Spring Data JPA is a part of the Spring Framework that provides a higher-level, easier-to-use API for working with JPA. It reduces the amount of boilerplate code required to interact with a database using JPA

**Example is**

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.stereotype.Component;

@Component

public class SpringDataExample implements CommandLineRunner {

@Autowired

private BookRepository bookRepo;

@Override

public void run(String... args) throws Exception {

bookRepo.save(new Book(3L, "Spring Data JPA Magic"));

}

}