Difference between JPA, Hibernate and Spring Data JPA

# Difference

## JPA

JPA is short for Java Persistence Api which provides specification for persisting, reading and managing data in a java application. It does not provide the implementations.

## Hibernate

Hibernate is an ORM (Object relational mapping) which provides all the implementation for the JPA. It helps to map the java objects to the relational database tables

## Spring data JPA

Spring data JPA is the library which provides abstraction over the JPA and helps to reduce lots of boilerplate code.

# Code Comparison

## Hibernate

package com.example.hibernate;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

public class Main {

public static void main(String[] args) {

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

Student student = new Student();

student.setName("John");

Session session = factory.openSession();

session.beginTransaction();

session.save(student); // Manual persist

session.getTransaction().commit();

Student fetched = session.get(Student.class, student.getId()); // Manual fetch

System.out.println("Fetched: " + fetched.getName());

session.close();

factory.close();

}

}

## Spring Data JPA

package com.example.springdata;

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 Application implements CommandLineRunner {

@Autowired

private StudentRepository repo;

public static void main(String[] args) {

SpringApplication.run(Application.class, args);

}

@Override

public void run(String... args) {

Student student = new Student();

student.setName("John");

repo.save(student); // No session handling

Student fetched = repo.findById(student.getId()).orElse(null); // Clean API

System.out.println("Fetched: " + (fetched != null ? fetched.getName() : "Not found"));

}

}