**Difference between JPA, Hibernate and Spring Data JPA   
  
Java Persistence API (JPA)**

* **JSR 338 Specification for persisting, reading and managing data from Java objects**
* **Does not contain concrete implementation of the specification**
* **Hibernate is one of the implementation of JPA**

**Hibernate**

* **ORM Tool that implements JPA**

**Spring Data JPA**

* **Does not have JPA implementation, but reduces boiler plate code**
* **This is another level of abstraction over JPA implementation provider like Hibernate**
* **Manages transactions**

**pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.employee</groupId>

<artifactId>EmployeeSpringDataJPA</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.data</groupId>

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

<version>2.7.12</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>5.3.33</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.6.15.Final</version>

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<version>2.2.224</version>

</dependency>

<dependency>

<groupId>jakarta.persistence</groupId>

<artifactId>jakarta.persistence-api</artifactId>

<version>3.1.0</version>

</dependency>

<dependency>

<groupId>jakarta.transaction</groupId>

<artifactId>jakarta.transaction-api</artifactId>

<version>2.0.1</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>exec-maven-plugin</artifactId>

<version>3.1.0</version>

<configuration>

<mainClass>com.employee.MainApp</mainClass>

</configuration>

</plugin>

</plugins>

</build>

</project>

**applicationContext.xml**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:tx="http://www.springframework.org/schema/tx"

xmlns:jpa="http://www.springframework.org/schema/data/jpa"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd

http://www.springframework.org/schema/tx

http://www.springframework.org/schema/tx/spring-tx.xsd

http://www.springframework.org/schema/data/jpa

http://www.springframework.org/schema/data/jpa/spring-jpa.xsd">

<context:component-scan base-package="com.employee"/>

<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">

<property name="driverClassName" value="org.h2.Driver"/>

<property name="url" value="jdbc:h2:mem:testdb"/>

<property name="username" value="sa"/>

<property name="password" value=""/>

</bean>

<bean id="entityManagerFactory" class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">

<property name="dataSource" ref="dataSource"/>

<property name="packagesToScan" value="com.employee.entity"/>

<property name="jpaVendorAdapter">

<bean class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter"/>

</property>

<property name="jpaProperties">

<props>

<prop key="hibernate.hbm2ddl.auto">update</prop>

<prop key="hibernate.dialect">org.hibernate.dialect.H2Dialect</prop>

<prop key="hibernate.show\_sql">true</prop>

</props>

</property>

</bean>

<bean id="transactionManager" class="org.springframework.orm.jpa.JpaTransactionManager">

<property name="entityManagerFactory" ref="entityManagerFactory"/>

</bean>

<tx:annotation-driven transaction-manager="transactionManager"/>

<jpa:repositories base-package="com.employee.repository"/>

</beans>

**Employee.java**

package com.employee.entity;

import jakarta.persistence.\*;

public class Employee {

private int id;

private String name;

private double salary;

public Employee() {}

public Employee(String name, double salary) {

this.name = name;

this.salary = salary;

}

public int getId() {

return id;

}

public String getName() {

return name;

}

public double getSalary() {

return salary;

}

public void setName(String name) {

this.name = name;

}

public void setSalary(double salary) {

this.salary = salary;

}

@Override

public String toString() {

return "Employee{id=" + id + ", name='" + name + "', salary=" + salary + '}';

}

}

**EmployeeRepository.java**

package com.employee.repository;

import com.employee.entity.Employee;

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

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {}

**EmployeeService.java**

package com.employee.service;

import com.employee.entity.Employee;

import com.employee.repository.EmployeeRepository;

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

import org.springframework.stereotype.Service;

import jakarta.transaction.Transactional;

import java.util.List;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository repository;

@Transactional

public void saveEmployees() {

repository.save(new Employee("Alice", 55000));

repository.save(new Employee("Bob", 62000));

}

public List<Employee> getAllEmployees() {

return repository.findAll();

}

}

**MainApp.java**

package com.employee;

import com.employee.entity.Employee;

import com.employee.service.EmployeeService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.util.List;

public class MainApp {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

EmployeeService service = context.getBean(EmployeeService.class);

service.saveEmployees();

List<Employee> employees = service.getAllEmployees();

for (Employee emp : employees) {

System.out.println(emp);

}

}

}

A group of words on a white background

AI-generated content may be incorrect.