**Spring Data JPA - Quick Example**

**Configure application.properties**

logging.level.org.springframework=info

logging.level.com.cognizant=debug

logging.level.org.hibernate.SQL=trace

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

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

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

spring.jpa.hibernate.ddl-auto=validate

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

**Create Model Class**

**File: com.cognizant.ormlearn.model.Country.java**

package com.cognizant.ormlearn.model;

import javax.persistence.\*;

@Entity

@Table(name = "country")

public class Country {

@Id

@Column(name = "co\_code")

private String code;

@Column(name = "co\_name")

private String name;

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

@Override

public String toString() {

return "Country [code=" + code + ", name=" + name + "]";

}

}

**Create Repository Interface**

**File: com.cognizant.ormlearn.repository.CountryRepository.java**

package com.cognizant.ormlearn.repository;

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

import org.springframework.stereotype.Repository;

import com.cognizant.ormlearn.model.Country;

@Repository

public interface CountryRepository extends JpaRepository<Country, String> {

}

**Create Service Class**

**File: com.cognizant.ormlearn.service.CountryService.java**

package com.cognizant.ormlearn.service;

import java.util.List;

import javax.transaction.Transactional;

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

import org.springframework.stereotype.Service;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.repository.CountryRepository;

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

@Transactional

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

}

**Modify Main Class**

**File: OrmLearnApplication.java**

package com.cognizant.ormlearn;

import java.util.List;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.service.CountryService;

@SpringBootApplication

public class OrmLearnApplication {

private static final Logger LOGGER = LoggerFactory.getLogger(OrmLearnApplication.class);

private static CountryService countryService;

public static void main(String[] args) {

ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);

countryService = context.getBean(CountryService.class);

testGetAllCountries();

}

private static void testGetAllCountries() {

LOGGER.info("Start");

List<Country> countries = countryService.getAllCountries();

LOGGER.debug("countries={}", countries);

LOGGER.info("End");

}

}

**Difference between JPA, hibernate and Spring Data JPA**

**Using Hibernate:**

**Pom.xml file**

<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>org.example</groupId>  
 <artifactId>Hibernate</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <packaging>jar</packaging>  
  
 <name>Hibernate</name>  
 <url>http://maven.apache.org</url>  
  
 <properties>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>3.8.1</version>  
 <scope>test</scope>  
 </dependency>  
  
 <!-- Hibernate Core -->  
 <dependency>  
 <groupId>org.hibernate</groupId>  
 <artifactId>hibernate-core</artifactId>  
 <version>5.6.15.Final</version>  
 </dependency>  
  
 <!-- H2 Database -->  
 <dependency>  
 <groupId>com.h2database</groupId>  
 <artifactId>h2</artifactId>  
 <version>2.1.214</version>  
 </dependency>  
  
 <!-- JPA API -->  
 <dependency>  
 <groupId>javax.persistence</groupId>  
 <artifactId>javax.persistence-api</artifactId>  
 <version>2.2</version>  
 </dependency>  
  
 <!-- Logging (optional but recommended) -->  
 <dependency>  
 <groupId>org.slf4j</groupId>  
 <artifactId>slf4j-simple</artifactId>  
 <version>1.7.30</version>  
 </dependency>  
 </dependencies>  
</project>

**Employee.java**

package org.example.model;  
import javax.persistence.\*;  
@Entity  
public class Employee {  
 @Id  
 private Integer id;  
 private String name;  
 public Employee(){}  
 public Employee(String name) {  
 this.name=name;  
 }  
 public Integer getId(){ return id; }  
 public void setId(Integer id) { this.id=id; }  
 public String getName(){ return name; }  
 public void setName(String name) { this.name=name; }  
}

**hibernateUtil.java**

package org.example.util;  
import org.hibernate.SessionFactory;  
import org.hibernate.cfg.Configuration;  
public class HibernateUtil {  
 private static final SessionFactory *sessionFactory*;  
 static {  
 try {  
 *sessionFactory*=new Configuration().configure().buildSessionFactory();  
 } catch (Throwable ex) {  
 System.*err*.println("Initial SessionFactory creation failed." + ex);  
 throw new ExceptionInInitializerError(ex);  
 }  
 }  
 public static SessionFactory getSessionFactory() {  
 return *sessionFactory*;  
 }  
}

**hibernate.cfg.xml**

<!DOCTYPE hibernate-configuration PUBLIC  
 "-//Hibernate/Hibernate Configuration DTD 3.0//EN"  
 "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">  
<hibernate-configuration>  
 <session-factory>  
 <property name="hibernate.connection.driver\_class">org.h2.Driver</property>  
 <property name="hibernate.connection.url">jdbc:h2:mem:testdb</property>  
 <property name="hibernate.connection.username">sa</property>  
 <property name="hibernate.connection.password"></property>  
 <property name="connection.pool\_size">1</property>  
 <property name="hibernate.dialect">org.hibernate.dialect.H2Dialect</property>  
 <property name="current\_session\_context\_class">thread</property>  
 <property name="hibernate.hbm2ddl.auto">update</property>  
 <property name="show\_sql">true</property>  
 <mapping class="org.example.model.Employee"/>  
 </session-factory>  
</hibernate-configuration>

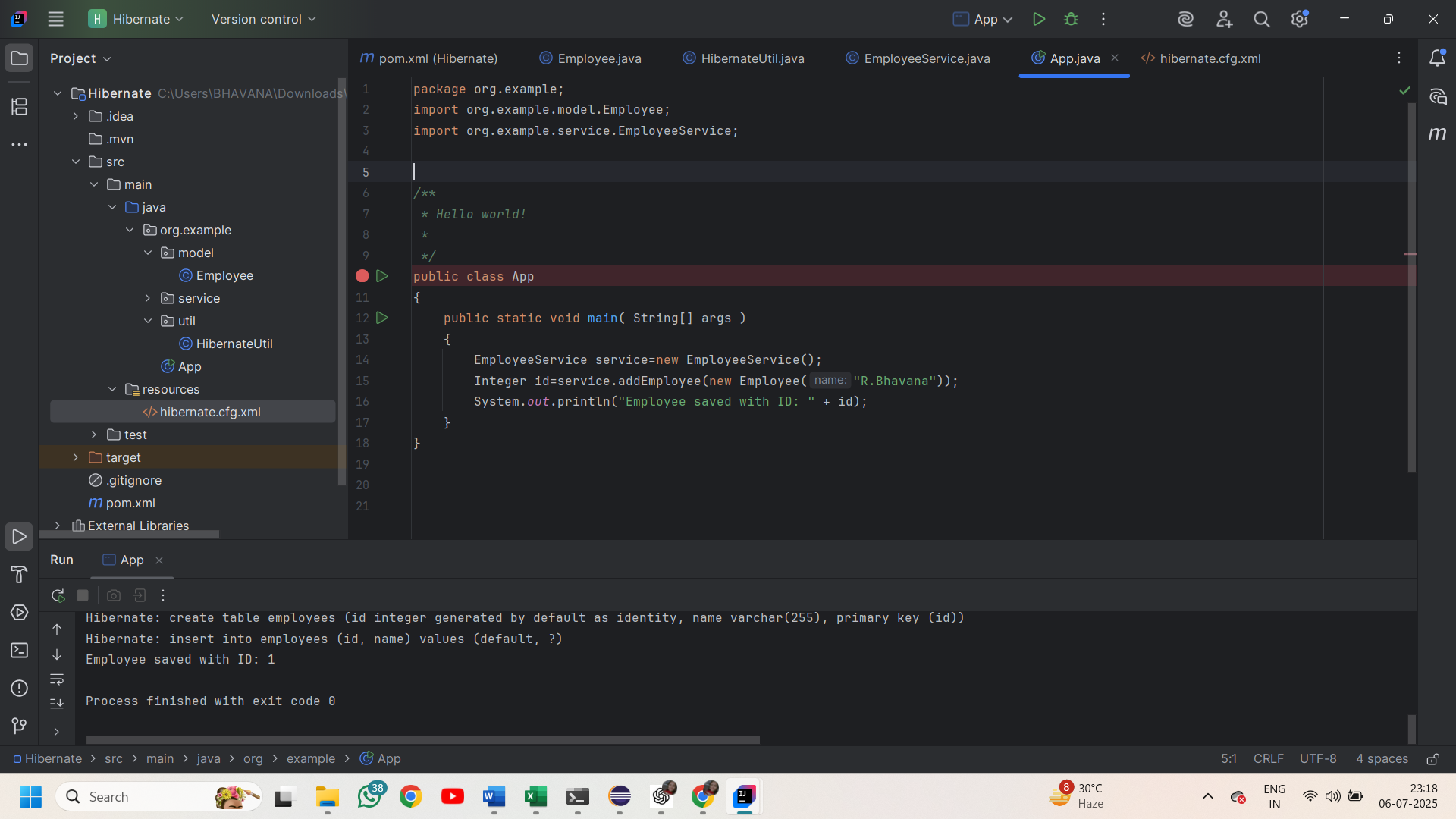
**EmployeeService.java**

package org.example.service;  
import org.example.model.Employee;  
import org.example.util.HibernateUtil;  
import org.hibernate.\*;  
public class EmployeeService {  
 public Integer addEmployee(Employee employee) {  
 Session session=HibernateUtil.*getSessionFactory*().openSession();  
 Transaction tx=null;  
 Integer employeeID=null;  
 try{  
 tx=session.beginTransaction();  
 employeeID=(Integer) session.save(employee);  
 tx.commit();  
 } catch (HibernateException e) {  
 if (tx!=null) tx.rollback();  
 e.printStackTrace();  
 } finally{  
 session.close();  
 }  
 return employeeID;  
 }  
}

**App.java**

package org.example;  
import org.example.model.Employee;  
import org.example.service.EmployeeService;  
public class App   
{  
 public static void main( String[] args )  
 {  
 EmployeeService service=new EmployeeService();  
 Integer id=service.addEmployee(new Employee("R.Bhavana"));  
 System.*out*.println("Employee saved with ID: " + id);  
 }  
}

**Output:**



**Using Spring Data JPA**

**Pom.xml file**

<?xml version="1.0" encoding="UTF-8"?>  
<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.5.3</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
 <groupId>com.example</groupId>  
 <artifactId>springdata-demo</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>springdata-demo</name>  
 <description>Demo project for Spring Boot</description>  
 <url/>  
 <licenses>  
 <license/>  
 </licenses>  
 <developers>  
 <developer/>  
 </developers>  
 <scm>  
 <connection/>  
 <developerConnection/>  
 <tag/>  
 <url/>  
 </scm>  
 <properties>  
 <java.version>21</java.version>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-jpa</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
 <dependency>  
 <groupId>com.h2database</groupId>  
 <artifactId>h2</artifactId>  
 <scope>runtime</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
</project>  
**Employee.java**

package com.example.model;  
import jakarta.persistence.\*;  
import org.springframework.stereotype.Component;  
@Component  
@Entity  
public class Employee {  
 @Id  
 private Integer id;  
 private String name;  
 public String getName() {  
 return name;  
 }  
 public void setName(String name) {  
 this.name = name;  
 }  
}

**EmployeeService.java**

package com.example.service;  
import com.example.model.Employee;  
import com.example.repository.EmployeeRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import org.springframework.transaction.annotation.Transactional;  
@Service  
public class EmployeeService {  
 @Autowired  
 private EmployeeRepository employeeRepository;  
 public void addEmployee(Employee employee) {  
 employeeRepository.save(employee);  
 }  
  
}

**EmployeeRepository.java**

package com.example.repository;  
import com.example.model.Employee;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
@Repository  
public interface EmployeeRepository extends JpaRepository<Employee,Integer> {  
  
}