**Hands on 1**

**Spring Data JPA - Quick Example**

**CODE**:

package com.cognizant.orm\_learn.model;  
  
import jakarta.persistence.Column;  
import jakarta.persistence.Entity;  
import jakarta.persistence.Id;  
import jakarta.persistence.Table;  
  
@Entity  
@Table(name = "country")  
public class Country {  
  
 @Id  
 @Column(name = "code")  
 private String code;  
  
 @Column(name = "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 + "]";  
 }  
}

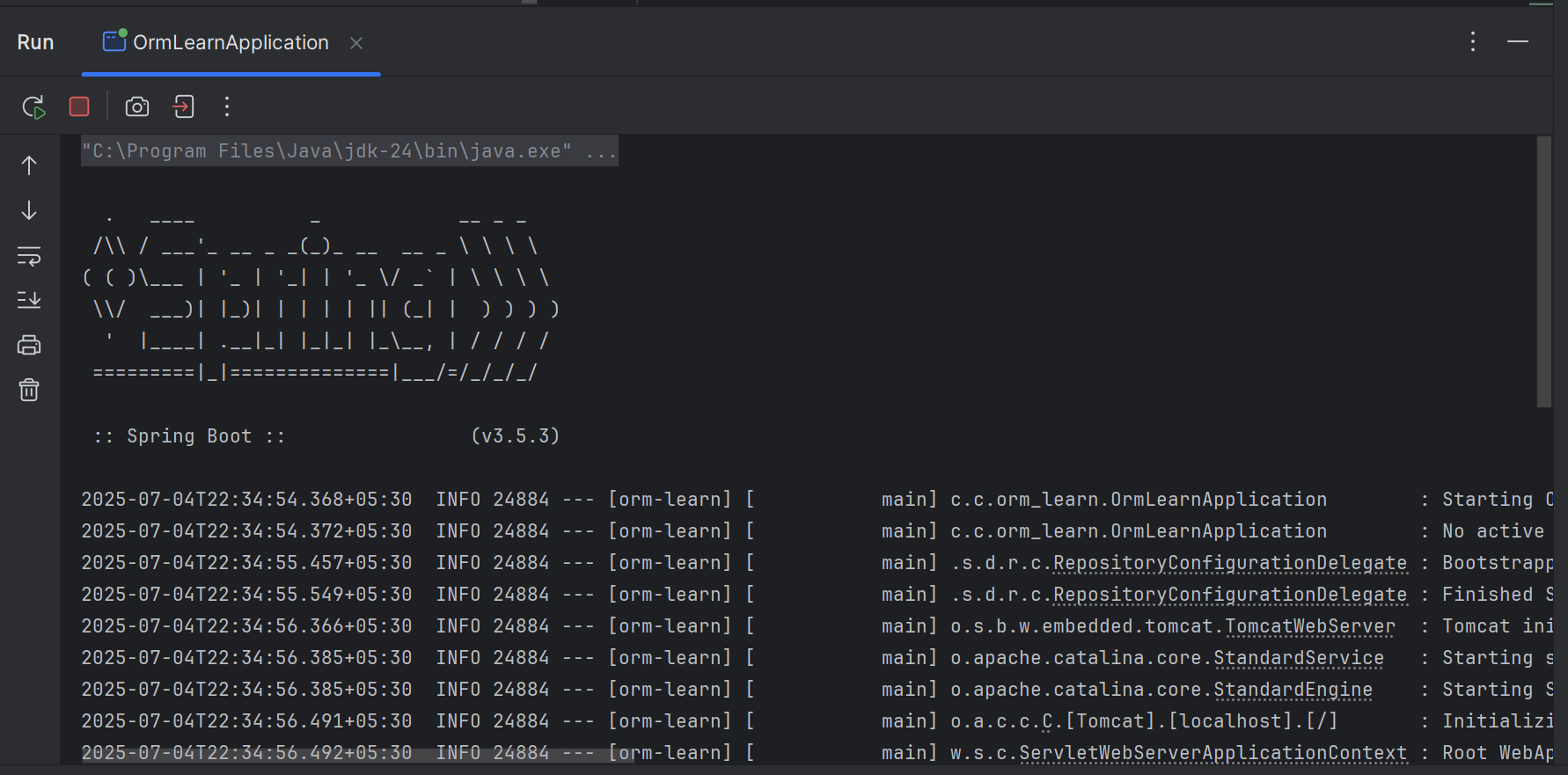
package com.cognizant.orm\_learn.repository;  
  
import com.cognizant.orm\_learn.model.Country;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface CountryRepository extends JpaRepository<Country, String> {  
}

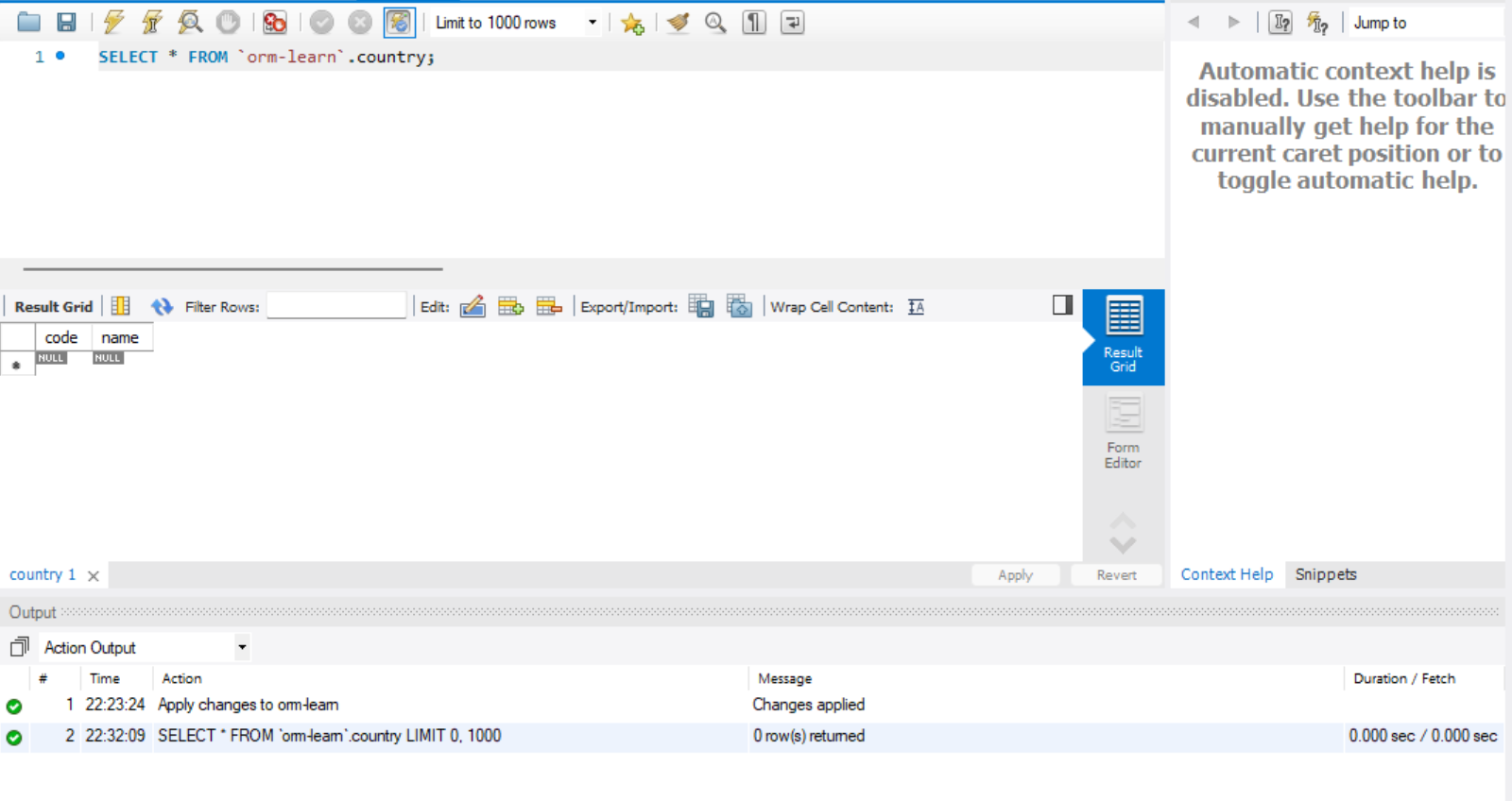
package com.cognizant.orm\_learn.service;  
  
import com.cognizant.orm\_learn.model.Country;  
import com.cognizant.orm\_learn.repository.CountryRepository;  
import jakarta.transaction.Transactional;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import java.util.List;  
  
@Service  
public class CountryService {  
  
 @Autowired  
 private CountryRepository countryRepository;  
  
 @Transactional  
 public List<Country> getAllCountries() {  
 return countryRepository.findAll();  
 }  
}

package com.cognizant.orm\_learn;  
  
import com.cognizant.orm\_learn.model.Country;  
import com.cognizant.orm\_learn.service.CountryService;  
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 java.util.List;  
  
@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);  
 *LOGGER*.info("Inside main");  
  
 *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");  
 }  
}

spring.application.name=orm-learn  
  
  
spring.datasource.url=jdbc:mysql://localhost:3306/orm-learn  
spring.datasource.username=root  
spring.datasource.password=123456  
  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect  
spring.jpa.hibernate.ddl-auto=update

**OUTPUT:**

****

****

**Hands on 4**

**Difference between JPA, Hibernate and Spring Data JPA**   
  
**Using Hibernate (Manual Session + Transaction)**

public Integer addEmployee(Employee employee){

Session session = factory.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;

}

**Using Spring Data JPA**

**EmployeeRepository.java**

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

// No need to write code implementation

}

**EmployeeService.java**

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee);

}

}

**Using JPA**

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

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>minimal-jpa</artifactId>

<version>1.0</version>

<dependencies>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.6.15.Final</version>

</dependency>

<dependency>

<groupId>javax.persistence</groupId>

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

<version>2.2</version>

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<version>2.1.214</version>

</dependency>

</dependencies>

</project>

package com.example.model;

import javax.persistence.\*;

@Entity

@Table(name = "employees")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

// Constructors

public Employee() {}

public Employee(String name) {

this.name = name;

}

// Getters & Setters

public int getId() { return id; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

}

package com.example.util;

import javax.persistence.EntityManagerFactory;

import javax.persistence.Persistence;

public class HibernateUtil {

private static final EntityManagerFactory emf =

Persistence.createEntityManagerFactory("my-pu");

public static EntityManagerFactory getEntityManagerFactory() {

return emf;

}

public static void shutdown() {

emf.close();

}

}

<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence" version="2.2">

<persistence-unit name="my-pu">

<class>com.example.model.Employee</class>

<properties>

<property name="javax.persistence.jdbc.driver" value="org.h2.Driver"/>

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

<property name="javax.persistence.jdbc.user" value="sa"/>

<property name="javax.persistence.jdbc.password" value=""/>

<property name="hibernate.dialect" value="org.hibernate.dialect.H2Dialect"/>

<property name="hibernate.hbm2ddl.auto" value="update"/>

<property name="hibernate.show\_sql" value="true"/>

</properties>

</persistence-unit>

</persistence>

package com.example;

import com.example.model.Employee;

import com.example.util.HibernateUtil;

import javax.persistence.EntityManager;

import javax.persistence.EntityTransaction;

public class MainApp {

public static void main(String[] args) {

EntityManager em = HibernateUtil.getEntityManagerFactory().createEntityManager();

EntityTransaction tx = em.getTransaction();

try {

tx.begin();

Employee emp = new Employee("Alice");

em.persist(emp);

tx.commit();

System.out.println("Employee saved with ID: " + emp.getId());

} catch (Exception e) {

if (tx.isActive()) tx.rollback();

e.printStackTrace();

} finally {

em.close();

HibernateUtil.shutdown();

}

}

}