**Hands on 5**

**Implement services for managing Country**   
**Application.properties:**

spring.datasource.url=jdbc:mysql://localhost:3306/orm\_learn

spring.datasource.username=root

spring.datasource.password=Sandhyaa\_31

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

 CountryAppRunner.java:

package com.cognizant.ormlearn;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.repository.CountryRepository;

import org.springframework.boot.CommandLineRunner;

import org.springframework.stereotype.Component;

@Component

public class CountryAppRunner implements CommandLineRunner {

    private final CountryRepository countryRepository;

    public CountryAppRunner(CountryRepository countryRepository) {

        this.countryRepository = countryRepository;

    }

    @Override

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

        Country country = new Country("IN", "India");

        countryRepository.save(country);

        System.out.println("Saved country: " + country.getName());

    }

}

Country.java

package com.cognizant.ormlearn.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

@Entity

public class Country {

    @Id

    private String code;

    private String name;

    public Country() {

    }

    public Country(String code, String name) {

        this.code = code;

        this.name = 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;

    }

}

CountryRepository.java:

package com.cognizant.ormlearn.repository;

import com.cognizant.ormlearn.model.Country;

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

import org.springframework.stereotype.Repository;

@Repository

public interface CountryRepository extends JpaRepository<Country, String> {

}

OrmLearnApplication

package com.cognizant.ormlearn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class OrmLearnApplication {

    public static void main(String[] args) {

        SpringApplication.run(OrmLearnApplication.class, args);

    }

}

CountryService

package com.cognizant.ormlearn.service;

import com.cognizant.ormlearn.model.Country;

import java.util.List;

public interface CountryService {

    Country getCountry(String code);

    List<Country> getAllCountries();

    void addCountry(Country country);

    void updateCountry(String code, String name);

    void deleteCountry(String code);

}

CountryServiceImpl.java:

package com.cognizant.ormlearn.service;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.repository.CountryRepository;

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

import org.springframework.stereotype.Service;

import java.util.List;

import java.util.Optional;

@Service

public class CountryServiceImpl implements CountryService {

    @Autowired

    private CountryRepository countryRepository;

    @Override

    public Country getCountry(String code) {

        Optional<Country> result = countryRepository.findById(code);

        return result.orElse(null);

    }

    @Override

    public List<Country> getAllCountries() {

        return countryRepository.findAll();

    }

    @Override

    public void addCountry(Country country) {

        countryRepository.save(country);

    }

    @Override

    public void updateCountry(String code, String name) {

        Country country = countryRepository.findById(code).orElse(null);

        if (country != null) {

            country.setName(name);

            countryRepository.save(country);

        }

    }

    @Override

    public void deleteCountry(String code) {

        countryRepository.deleteById(code);

    }

}

CountryController.java:

package com.cognizant.ormlearn.controller;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.service.CountryService;

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

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/countries")

public class CountryController {

    @Autowired

    private CountryService countryService;

    @GetMapping

    public List<Country> getAllCountries() {

        return countryService.getAllCountries();

    }

    @GetMapping("/{code}")

    public Country getCountry(@PathVariable String code) {

        return countryService.getCountry(code);

    }

    @PostMapping

    public void addCountry(@RequestBody Country country) {

        countryService.addCountry(country);

    }

    @PutMapping("/{code}")

    public void updateCountry(@PathVariable String code, @RequestBody Country updated) {

        countryService.updateCountry(code, updated.getName());

    }

    @DeleteMapping("/{code}")

    public void deleteCountry(@PathVariable String code) {

        countryService.deleteCountry(code);

    }

}

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.cognizant</groupId>

    <artifactId>orm-learn</artifactId>

    <version>1.0-SNAPSHOT</version>

    <packaging>jar</packaging>

    <parent>

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

        <artifactId>spring-boot-starter-parent</artifactId>

        <version>3.2.5</version> <!-- or latest compatible -->

        <relativePath/> <!-- lookup parent from repository -->

    </parent>

    <dependencies>

        <!-- Spring Boot Starter Data JPA -->

        <dependency>

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

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

        </dependency>

        <!-- Spring Boot Starter Web -->

        <dependency>

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

            <artifactId>spring-boot-starter-web</artifactId>

        </dependency>

        <!-- MySQL Driver -->

        <dependency>

            <groupId>com.mysql</groupId>

            <artifactId>mysql-connector-j</artifactId>

            <scope>runtime</scope>

        </dependency>

        <!-- Spring Boot Starter Test -->

        <dependency>

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

            <artifactId>spring-boot-starter-test</artifactId>

            <scope>test</scope>

        </dependency>

    </dependencies>

    <build>

        <plugins>

            <!-- Spring Boot Maven Plugin -->

            <plugin>

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

                <artifactId>spring-boot-maven-plugin</artifactId>

            </plugin>

        </plugins>

    </build>

</project>

**Find a country based on country code**   
CountryNotFoundException:

package com.cognizant.ormlearn.service.exception;

public class CountryNotFoundException extends Exception {

public CountryNotFoundException(String message) {

super(message);

}

}

CountryService

package com.cognizant.ormlearn.service;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.service.exception.CountryNotFoundException;

public interface CountryService {

Country findCountryByCode(String countryCode) throws CountryNotFoundException;

}

CountryServiceImpl

package com.cognizant.ormlearn.service;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.repository.CountryRepository;

import com.cognizant.ormlearn.service.exception.CountryNotFoundException;

@Service

public class CountryServiceImpl implements CountryService {

@Autowired

private CountryRepository countryRepository;

@Override

@Transactional

public Country findCountryByCode(String countryCode) throws CountryNotFoundException {

Optional<Country> result = countryRepository.findById(countryCode);

if (!result.isPresent()) {

throw new CountryNotFoundException("Country with code " + countryCode + " not found");

}

return result.get();

}

}

OrmLearnApplication.java

package com.cognizant.ormlearn;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.service.CountryService;

import com.cognizant.ormlearn.service.exception.CountryNotFoundException;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

@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);

try {

getCountryByCodeTest();

} catch (CountryNotFoundException e) {

LOGGER.error("Exception: ", e);

}

}

private static void getCountryByCodeTest() throws CountryNotFoundException {

LOGGER.info("Start");

Country country = countryService.findCountryByCode("IN");

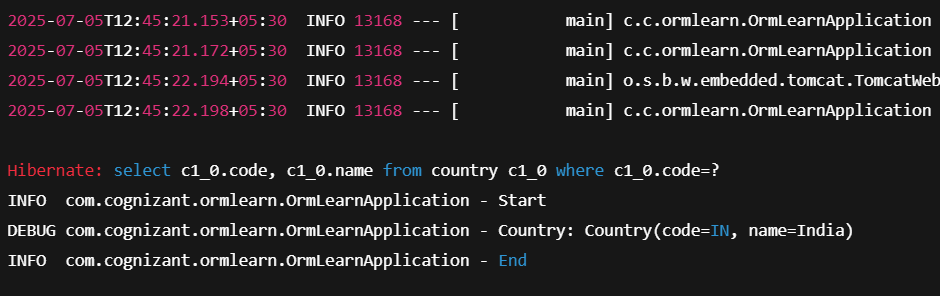
LOGGER.debug("Country: {}", country);

LOGGER.info("End");

}

}

OUTPUT:



**Add a new country**   
CountryService:

import com.cognizant.ormlearn.model.Country;

import jakarta.transaction.Transactional;

public interface CountryService {

Country findCountryByCode(String countryCode) throws CountryNotFoundException;

@Transactional

void addCountry(Country country);

}

CountryServiceImpl:

@Override

@Transactional

public void addCountry(Country country) {

countryRepository.save(country);

}

OrmLearnApplication.java:

private static void testAddCountry() {

LOGGER.info("Start");

Country country = new Country();

country.setCode("JP");

country.setName("Japan");

countryService.addCountry(country);

try {

Country addedCountry = countryService.findCountryByCode("JP");

LOGGER.debug("Added Country: {}", addedCountry);

} catch (CountryNotFoundException e) {

LOGGER.error("Country not found!", e);

}

LOGGER.info("End");

}

Main:  
public static void main(String[] args) {

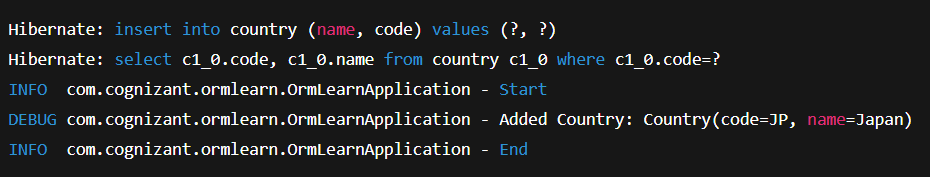
ConfigurableApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);

countryService = context.getBean(CountryService.class);

testAddCountry();

}

OUTPUT:



Demonstrate implementation of Query Methods feature of Spring Data JPA:

Country.java

@Entity

@Table(name = "country")

public class Country {

@Id

private String code;

private String name;

private int population;

private LocalDate independenceDate;

}

CountryRepository:

@Repository

public interface CountryRepository extends JpaRepository<Country, String> {

List<Country> findByNameContaining(String keyword);

List<Country> findAllByOrderByNameAsc();

List<Country> findAllByOrderByNameDesc();

List<Country> findByNameStartingWith(String prefix);

List<Country> findByIndependenceDateBetween(LocalDate start, LocalDate end);

List<Country> findByPopulationGreaterThan(int population);

List<Country> findByPopulationLessThan(int population);

List<Country> findTop3ByOrderByNameAsc();

}

OrmLearnApplication:

@SpringBootApplication

public class OrmLearnApplication {

private static CountryService countryService;

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

public static void main(String[] args) {

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

countryService = context.getBean(CountryService.class);

queryMethodsTest();

}

private static void queryMethodsTest() {

LOGGER.info("Start - Query Methods Demo");

LOGGER.info("Countries containing 'an': {}", countryService.findByNameContaining("an"));

LOGGER.info("Countries sorted ascending: {}", countryService.findAllByOrderByNameAsc());

LOGGER.info("Countries starting with 'In': {}", countryService.findByNameStartingWith("In"));

LOGGER.info("Countries independent between 1947 and 2000: {}",

countryService.findByIndependenceDateBetween(LocalDate.of(1947, 1, 1), LocalDate.of(2000, 12, 31)));

LOGGER.info("Countries with population > 1000000: {}", countryService.findByPopulationGreaterThan(1000000));

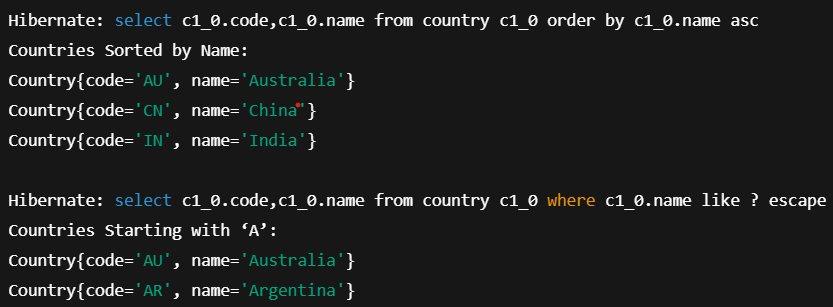
LOGGER.info("Top 3 countries: {}", countryService.findTop3ByOrderByNameAsc());

LOGGER.info("End - Query Methods Demo");

}

}

OUTPUT:



Demonstrate implementation of O/R Mapping:

Department.java:

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

import java.util.List;

@Entity

public class Department {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

@OneToMany(mappedBy = "department", fetch = FetchType.LAZY)

private List<Employee> employees;

}

Employee.java

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

import java.util.List;

@Entity

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private double salary;

@ManyToOne(fetch = FetchType.EAGER)

@JoinColumn(name = "dept\_id")

private Department department;

@ManyToMany(fetch = FetchType.LAZY)

@JoinTable(name = "employee\_skill",

joinColumns = @JoinColumn(name = "emp\_id"),

inverseJoinColumns = @JoinColumn(name = "skill\_id"))

private List<Skill> skills;

}

Skill.java:

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

import java.util.List;

@Entity

public class Skill {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

@ManyToMany(mappedBy = "skills")

private List<Employee> employees;

}

Repository Example:

package com.cognizant.ormlearn.repository;

import com.cognizant.ormlearn.model.Employee;

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

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

OrmLearnApplication.java:

package com.cognizant.ormlearn;

import com.cognizant.ormlearn.model.Employee;

import com.cognizant.ormlearn.model.Department;

import com.cognizant.ormlearn.model.Skill;

import com.cognizant.ormlearn.repository.EmployeeRepository;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import java.util.Arrays;

@SpringBootApplication

public class OrmLearnApplication implements CommandLineRunner {

@Autowired

private EmployeeRepository employeeRepository;

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

public void run(String... args) {

Employee emp = new Employee();

emp.setName("Ravi");

emp.setSalary(50000);

Department dept = new Department();

dept.setName("HR");

emp.setDepartment(dept);

Skill skill = new Skill();

skill.setName("Java");

emp.setSkills(Arrays.asList(skill));

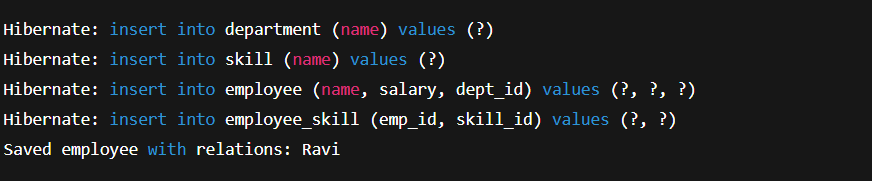
employeeRepository.save(emp);

System.out.println("Saved employee with relations: " + emp.getName());

}

}

OUTPUT:



Demonstrate writing Hibernate Query Language and Native Query:

Country.java:

@Entity

public class Country {

@Id

private String code;

private String name;

}

CountryRepository.java:

package com.cognizant.ormlearn.repository;

import com.cognizant.ormlearn.model.Country;

import org.springframework.data.jpa.repository.\*;

import org.springframework.data.repository.query.Param;

import java.util.List;

public interface CountryRepository extends JpaRepository<Country, String> {

@Query("SELECT c FROM Country c WHERE c.name LIKE %:text%")

List<Country> searchByName(@Param("text") String text);

@Query("SELECT COUNT(c) FROM Country c")

long countCountries();

@Query(value = "SELECT \* FROM country WHERE name LIKE %:text%", nativeQuery = true)

List<Country> nativeSearchByName(@Param("text") String text);

}

Service Usage Example:

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

public List<Country> searchByName(String text) {

return countryRepository.searchByName(text);

}

public long getCountryCount() {

return countryRepository.countCountries();

}

public List<Country> nativeSearch(String text) {

return countryRepository.nativeSearchByName(text);

}

}

OrmLearnApplication.java:

@SpringBootApplication

public class OrmLearnApplication implements CommandLineRunner {

@Autowired

private CountryService countryService;

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

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

List<Country> result = countryService.searchByName("in");

System.out.println("HQL Query Result:");

result.forEach(c -> System.out.println(c.getCode() + " - " + c.getName()));

long count = countryService.getCountryCount();

System.out.println("Country Count: " + count);

List<Country> nativeResult = countryService.nativeSearch("in");

System.out.println("Native Query Result:");

nativeResult.forEach(c -> System.out.println(c.getCode() + " - " + c.getName()));

}

}

OUTPUT:

