**HANDS ON 6:**

**Find a country based on country code** 

**Application.properties:**

spring.application.name=orm\_learn

# H2 DB Configuration

spring.datasource.url=jdbc:h2:mem:orm\_learn

spring.datasource.driver-class-name=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

# Show SQL logs

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format\_sql=true

# Automatically create tables from entities

spring.jpa.hibernate.ddl-auto=create

# Enable data.sql execution

spring.sql.init.mode=always

# H2 Console

spring.h2.console.enabled=true

spring.h2.console.path=/h2-console

data.sql and schema.sql:

CREATE TABLE country (

code **VARCHAR**(5) **PRIMARY KEY**,

co\_name **VARCHAR**(255)

);

insert into country (code, co\_name) values ('SS', 'South Sudan');

INSERT INTO country (code, co\_name) VALUES ('US', 'United States');

INSERT INTO country (code, co\_name) VALUES ('UK', 'United Kingdom');

**Country.java:**

**package** com.cognizant.orm\_learn.model;

**import** jakarta.persistence.Entity;

**import** jakarta.persistence.Id;

**import** jakarta.persistence.Table;

@Entity

@Table(name = "country")

**public** **class** Country {

@Id

**private** String code;

**private** String coName;

// Constructors

**public** Country() {}

**public** Country(String code, String coName) {

**this**.code = code;

**this**.coName = coName;

}

**public** String getCode() {

**return** code;

}

**public** **void** setCode(String code) {

**this**.code = code;

}

**public** String getCoName() {

**return** coName;

}

**public** **void** setCoName(String coName) {

**this**.coName = coName;

}

@Override

**public** String toString() {

**return** "Country{" +

"code='" + code + '\'' +

", coName='" + coName + '\'' +

'}';

}

}

**CountryService.java:**

**package** com.cognizant.orm\_learn.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.orm\_learn.model.Country;

**import** com.cognizant.orm\_learn.repository.CountryRepository;

**import** com.cognizant.orm\_learn.service.exception.CountryNotFoundException;

@Service

**public** **class** CountryService {

@Autowired

**private** CountryRepository countryRepository;

@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();

}

}

**CountryRepository.java:**

**package** com.cognizant.orm\_learn.repository;

**import** com.cognizant.orm\_learn.model.Country;

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

**import** java.util.Optional;

**public** **interface** CountryRepository **extends** JpaRepository<Country, String> {

Optional<Country> findByCode(String code);

}

**CountryNotFounException.java:**

**package** com.cognizant.orm\_learn.service.exception;

**public** **class** CountryNotFoundException **extends** Exception {

**public** CountryNotFoundException(String message) {

**super**(message);

}

}

**OrmLearnApplication.java:**

**package** com.cognizant.orm\_learn;

**import** com.cognizant.orm\_learn.model.Country;

**import** com.cognizant.orm\_learn.repository.CountryRepository;

**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.Optional;

@SpringBootApplication

**public** **class** OrmLearnApplication **implements** CommandLineRunner {

@Autowired

**private** CountryRepository countryRepository;

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(OrmLearnApplication.**class**, args);

}

@Override

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

String searchCode = "IN";

Optional<Country> country = countryRepository.findByCode(searchCode);

**if** (country.isPresent()) {

System.***out***.println("Found country: " + country.get().getCoName());

} **else** {

System.***out***.println("Country not found with code: " + searchCode);

}

}

}

**OUTPUT:**



