

Additional Hands-on: Implement services for managing Country

Code:

Country.java

```
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 = "co_code")
    private String code;

    @Column(name = "co_name")
    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;
    }

    @Override
    public String toString() {
```

```

        return "Country{" + "code='" + code + '\'' + ", name='" + name + '\''
+ '}' + '\n';
    }
}

```

Country Repository.java

```

package com.cognizant.orm_learn.repository;

import com.cognizant.orm_learn.model.Country;
import org.springframework.data.jpa.repository.JpaRepository;
import java.util.List;

public interface CountryRepository extends JpaRepository<Country, String> {
    List<Country> findByNameContainingIgnoreCase(String namePart);
}

```

Country Service.java

```

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 Country findCountryByCode(String code) {
        return countryRepository.findById(code)
            .orElseThrow(() -> new RuntimeException("Country not
found"));
    }

    @Transactional
    public void addCountry(Country country) {
        countryRepository.save(country);
    }

    @Transactional
    public void updateCountry(String code, String newName) {
        Country country = findCountryByCode(code);
        country.setName(newName);
    }
}

```

```

        countryRepository.save(country);
    }

    @Transactional
    public void deleteCountry(String code) {
        countryRepository.deleteById(code);
    }

    @Transactional
    public List<Country> findCountriesByPartialName(String namePart) {
        return countryRepository.findByNameContainingIgnoreCase(namePart);
    }
}

```

Main.java(OrmLearnApplication.java)

```

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);
        countryService = context.getBean(CountryService.class);

        testCountryFeatures();
    }

    private static void testCountryFeatures() {
        LOGGER.info("=== ADD ===");
        countryService.addCountry(new Country("JP", "Japan"));
        countryService.addCountry(new Country("MY", "Malaysia"));

        LOGGER.info("=== FIND BY CODE ===");
        LOGGER.info("Found: {}", countryService.findCountryByCode("JP"));
    }
}

```

```

    LOGGER.info("=== UPDATE ===");
    countryService.updateCountry("JP", "Nippon");
    LOGGER.info("Updated: {}", countryService.findCountryByCode("JP"));

    LOGGER.info("=== FIND BY PARTIAL NAME ===");
    List<Country> partial =
countryService.findCountriesByPartialName("aus");
    partial.forEach(c -> LOGGER.info("Match: {}", c));

    LOGGER.info("=== DELETE ===");
    countryService.deleteCountry("AU");

    LOGGER.info("=== FINAL LIST ===");

countryService.findCountriesByPartialName("").forEach(System.out::println);
}
}

```

OutPut:

The screenshot displays the IDE environment for the 'orm-learn' project. The 'OrmLearnApplication.java' file is open, showing the main method and a test method. The output window at the bottom shows the following log messages:

```

06-07-25 15:13:00.869 restartedMain INFO c.c.o.OrmLearnApplication testCountryFeatures 27 === ADD ===
06-07-25 15:13:00.879 restartedMain DEBUG org.hibernate.SQL logStatement 135 select c1_0.co_code,c1_0.co_name from country c1_0 where c1_0.co_c
06-07-25 15:13:00.895 restartedMain DEBUG org.hibernate.SQL logStatement 135 insert into country (co_name,co_code) values (?,?)
06-07-25 15:13:00.900 restartedMain DEBUG org.hibernate.SQL logStatement 135 select c1_0.co_code,c1_0.co_name from country c1_0 where c1_0.co_c
06-07-25 15:13:00.900 restartedMain DEBUG org.hibernate.SQL logStatement 135 insert into country (co_name,co_code) values (?,?)
06-07-25 15:13:00.902 restartedMain INFO c.c.o.OrmLearnApplication testCountryFeatures 31 === FIND BY CODE ===
06-07-25 15:13:00.904 restartedMain DEBUG org.hibernate.SQL logStatement 135 select c1_0.co_code,c1_0.co_name from country c1_0 where c1_0.co_c
06-07-25 15:13:00.906 restartedMain INFO c.c.o.OrmLearnApplication testCountryFeatures 32 Found: Country{code='JP', name='Japan'}
06-07-25 15:13:00.906 restartedMain INFO c.c.o.OrmLearnApplication testCountryFeatures 34 === UPDATE ===
06-07-25 15:13:00.906 restartedMain DEBUG org.hibernate.SQL logStatement 135 select c1_0.co_code,c1_0.co_name from country c1_0 where c1_0.co_c

```

Local Instance 3306

Administration Schemas Query 1 SQL File 1* SQL File 2*

SCHEMAS

Filter objects

- ormlearn
 - Tables
 - employee
 - Views
 - Stored Procedures
 - Functions
- sys
 - Tables
 - Views
 - Stored Procedures
 - Functions

1 use ormlearn;
2 create table country(co_code varchar(2) primary key, co_name varchar(50));
3
4 insert into country values ('IN', 'India');
5 insert into country values ('US', 'United States of America');
6
7 select * from country;
8
9

100% 1:9

Result Grid Filter Rows Search Edit Export/Import

co_code	co_name
HULL	HULL

country 2 Apply Revert

Action Output

	Time	Action	Response	Duration / Fetch Time
64	15:12:31	DROP TABLE IF EXISTS country	0 row(s) affected	0.0062 sec
65	15:12:31	CREATE TABLE country (co_code VARCHAR(2) PRIMARY KEY, co_name VARC...	0 row(s) affected	0.0050 sec
66	15:12:42	select * from country LIMIT 0, 1000	0 row(s) returned	0.0016 sec / 0.00000...

Query Completed

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Local Instance 3306

Administration Schemas Query 1 SQL File 1* SQL File 2*

SCHEMAS

Filter objects

- ormlearn
 - Tables
 - employee
 - Views
 - Stored Procedures
 - Functions
- sys
 - Tables
 - Views
 - Stored Procedures
 - Functions

1 use ormlearn;
2 create table country(co_code varchar(2) primary key, co_name varchar(50));
3
4 insert into country values ('IN', 'India');
5 insert into country values ('US', 'United States of America');
6
7 select * from country;
8
9

100% 1:9

Result Grid Filter Rows Search Edit Export/Import

co_code	co_name
JP	Nippon
MY	Malaysia
HULL	HULL

country 3 Apply Revert

Action Output

	Time	Action	Response	Duration / Fetch Time
65	15:12:31	CREATE TABLE country (co_code VARCHAR(2) PRIMARY KEY, co_name VARC...	0 row(s) affected	0.0050 sec
66	15:12:42	select * from country LIMIT 0, 1000	0 row(s) returned	0.0016 sec / 0.00000...
67	15:14:47	select * from country LIMIT 0, 1000	2 row(s) returned	0.00047 sec / 0.00000...

Query Completed

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.