Spring Data JPA - Quick Example

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>0.0.1-SNAPSHOT</version>
  <description>Demo project for Spring Data JPA and Hibernate</description>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.7.5</version>
  </parent>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-devtools</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
```

```
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <scope>runtime</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>
```

application.properties

```
# Logging
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

# DB Config
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
```

```
# Hibernate Config
spring.jpa.hibernate.ddl-auto=validate
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
```

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");
    }
}
```

model/Country.java

```
package com.cognizant.ormlearn.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.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 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 + "]";
    }
}
```

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> {
}
```

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();
    }
}
```

Difference between JPA, Hibernate and Spring Data JPA

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.example</groupId>
    <artifactId>employee-data-jpa</artifactId>
    <version>0.0.1-SNAPSHOT</version>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.7.5</version>
    </parent>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
        </dependency>
    </dependencies>
</project>
```

application.properties

```
spring.datasource.url=jdbc:mysql://localhost:3306/employeedb
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.hibernate.ddl-auto=create
spring.jpa.show-sql=true
```

model/Employee.java

```
package com.example.demo.model;

import javax.persistence.*;

@Entity
public class Employee {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    private String name;
    private String department;

    // Getters and Setters
    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

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

    public String getDepartment() {
        return department;
    }

    public void setDepartment(String department) {
        this.department = department;
    }

    @Override
    public String toString() {
        return "Employee{id=" + id + ", name='" + name + "', department='" + department + "'}";
    }
}
```

repository/EmployeeRepository.java

```
package com.example.demo.repository;

import com.example.demo.model.Employee;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
```

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

service/EmployeeService.java

```
package com.example.demo.service;

import com.example.demo.model.Employee;
import com.example.demo.repository.EmployeeRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import javax.transaction.Transactional;

@Service
public class EmployeeService {

    @Autowired
    private EmployeeRepository employeeRepository;

    @Transactional
    public void addEmployee(Employee employee) {
        employeeRepository.save(employee);
    }
}
```

DemoApplication.java

```
package com.example.demo;

import com.example.demo.model.Employee;
import com.example.demo.service.EmployeeService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class DemoApplication implements CommandLineRunner {

    @Autowired
    private EmployeeService employeeService;

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }

    @Override
    public void run(String... args) throws Exception {
        Employee emp = new Employee();
        emp.setName("John Doe");
        emp.setDepartment("Finance");

        employeeService.addEmployee(emp);
        System.out.println("Employee added using Spring Data JPA");
    }
}
```

Implement services for managing Country

pom.xml

```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.example</groupId>
    <artifactId>country-service</artifactId>
    <version>1.0</version>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.7.5</version>
    </parent>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
        </dependency>
    </dependencies>
</project>
```

application.properties

```
spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn

spring.datasource.username=root

spring.datasource.password=root

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver


spring.jpa.hibernate.ddl-auto=validate

spring.jpa.show-sql=true
```

model/Country.java

```
package com.example.demo.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;


    // Getters and Setters

    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;
    }

    // toString
    @Override
    public String toString() {
        return "Country{code='" + code + "', name='" + name + "'}";
    }
}
```

repository/CountryRepository.java

```
package com.example.demo.repository;

import com.example.demo.model.Country;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public interface CountryRepository extends JpaRepository<Country, String> {
    List<Country> findByNameContaining(String keyword);
}
```

service/CountryService.java

```
package com.example.demo.service;

import com.example.demo.model.Country;
import com.example.demo.repository.CountryRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import javax.transaction.Transactional;
import java.util.List;
import java.util.Optional;

@Service
public class CountryService {

    @Autowired
    private CountryRepository countryRepository;

    public Country findCountryByCode(String code) {
        Optional<Country> result = countryRepository.findById(code);
        return result.orElse(null);
    }

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

    @Transactional
    public void updateCountry(Country country) {
        if (countryRepository.existsById(country.getCode())) {
            countryRepository.save(country);
        }
    }

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

    public List<Country> findCountriesByName(String keyword) {
        return countryRepository.findByNameContaining(keyword);
    }
}
```

DemoApplication.java

```
package com.example.demo;

import com.example.demo.model.Country;
import com.example.demo.service.CountryService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.CommandLineRunner;

import java.util.List;

@SpringBootApplication
public class DemoApplication implements CommandLineRunner {
    @Autowired
    private CountryService countryService;

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }

    @Override
    public void run(String... args) throws Exception {
        System.out.println("Testing Country Services:");

        Country country = new Country();
        country.setCode("XX");
        country.setName("Testland");

        countryService.addCountry(country);
        System.out.println("Added: " + countryService.findCountryByCode("XX"));

        country.setName("Updated Testland");
        countryService.updateCountry(country);
        System.out.println("Updated: " + countryService.findCountryByCode("XX"));

        List<Country> matched = countryService.findCountriesByName("land");
        System.out.println("Countries matching 'land': " + matched);

        countryService.deleteCountry("XX");
        System.out.println("Deleted: XX");
    }
}
```

Find a country based on country code

com.cognizant.springlearn.service.exception.CountryNotFoundException.java

```
package com.cognizant.springlearn.service.exception;


public class CountryNotFoundException extends Exception {
    public CountryNotFoundException(String message) {
        super(message);
    }
}
```

com.cognizant.springlearn.service.CountryService.java

```
package com.cognizant.springlearn.service;


import com.cognizant.springlearn.model.Country;
import com.cognizant.springlearn.repository.CountryRepository;
import com.cognizant.springlearn.service.exception.CountryNotFoundException;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;


import javax.transaction.Transactional;
import java.util.List;
import java.util.Optional;


@Service
public class CountryService {


    @Autowired
    private CountryRepository countryRepository;


    public List<Country> getAllCountries() {
        return countryRepository.findAll();
    }

    @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();
    }
}
```

com.cognizant.springlearn.OrmLearnApplication.java

```
package com.cognizant.springlearn;

import com.cognizant.springlearn.model.Country;
import com.cognizant.springlearn.service.CountryService;
import com.cognizant.springlearn.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 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);

        testFindCountryByCode();
    }

    private static void testFindCountryByCode() {
        LOGGER.info("Start");

        try {
            Country country = countryService.findCountryByCode("IN");
            LOGGER.debug("Country: {}", country);
        } catch (CountryNotFoundException e) {
            LOGGER.error("Exception: {}", e.getMessage());
        }

        LOGGER.info("End");
    }
}
```

Add a new country

Update CountryService

```
package com.cognizant.springlearn.service;


import com.cognizant.springlearn.model.Country;
import com.cognizant.springlearn.repository.CountryRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;


import javax.transaction.Transactional;


@Service
public class CountryService {


    @Autowired
    private CountryRepository countryRepository;


    // other methods …


    /** Insert—or overwrite—one Country row */
    @Transactional
    public void addCountry(Country country) {
        countryRepository.save(country);
    }
}
```

Add the Test Method in OrmLearnApplication

```
package com.cognizant.springlearn;


import com.cognizant.springlearn.model.Country;
import com.cognizant.springlearn.service.CountryService;
import com.cognizant.springlearn.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);

        testAddCountry();          // <-- new test
        // testFindCountryByCode(); // previous tests (optional)
    }

    /** Hands-on 7: create, insert, then retrieve a country */
    private static void testAddCountry() {
        LOGGER.info("Start – add country");

        //1️⃣ construct a brand-new Country object
        Country newCountry = new Country();
        newCountry.setCode("TP");          // use a 2-letter code not in the table
        newCountry.setName("Testopia");
        // 2️⃣ add it via the service
        countryService.addCountry(newCountry);
        LOGGER.debug("Inserted TP : {}", newCountry);

        // 3️⃣ read it back to verify
        try {
            Country fetched = countryService.findCountryByCode("TP");
            LOGGER.debug("Fetched  TP : {}", fetched);
        } catch (CountryNotFoundException e) {
            LOGGER.error("Country not found after insert! {}", e.getMessage());
        }

        LOGGER.info("End – add country");
    }
}
```