

Hands on 1

Spring Data JPA - Quick Example

Pom.xml:

```
<dependencies>

  <!-- Spring Boot -->

  <dependency>

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

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

  </dependency>

  <!-- MySQL Driver -->

  <dependency>

    <groupId>com.mysql</groupId>

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

  </dependency>

  <!-- Logging (optional) -->

  <dependency>

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

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

  </dependency>

</dependencies>

application.properties

spring.datasource.url=jdbc:mysql://localhost:3306/orm_learn

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=none

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

Country.java

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

```
import jakarta.persistence.Column;
```

```
import jakarta.persistence.Entity;
```

```
import jakarta.persistence.Id;
```

```
@Entity
```

```
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 + "];"  
}  
}
```

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> {  
}
```

CountryService.java

```
package com.cognizant.ormlearn.service;
```

```
import java.util.List;  
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;
```

```
    public List<Country> getAllCountries() {
```

```
        return countryRepository.findAll();
    }
}

OrmLearnApplication.java

package com.cognizant.ormlearn;

import java.util.List;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ComponentScan;

import com.cognizant.ormlearn.model.Country;
import com.cognizant.ormlearn.service.CountryService;

@SpringBootApplication
@ComponentScan(basePackages = "com.cognizant.ormlearn")
public class OrmLearnApplication {

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

    @Autowired
    private CountryService countryService;

    public static void main(String[] args) {
        SpringApplication.run(OrmLearnApplication.class, args);
        System.out.println("Inside main");
    }
}
```

@Autowired

```
public void testGetAllCountries() {
```

```
    System.out.println("Start");
```

```
    List<Country> countries = countryService.getAllCountries();
```

```
    LOGGER.debug("countries={}", countries);
```

```
    System.out.println("countries=" + countries);
```

```
    System.out.println("End");
```

```
}
```

```
}
```

SQL for Country Table

CREATE TABLE country (

code CHAR(2) PRIMARY KEY,

name VARCHAR(50)

);

INSERT INTO country VALUES ('IN', 'India');

INSERT INTO country VALUES ('US', 'United States of America');

The screenshot shows a SQL IDE interface with a query editor on the left and an output pane at the bottom. The query editor contains the following SQL commands:

```
1  
2  
3 CREATE DATABASE ormlearn;  
4  
5 USE ormlearn;  
6  
7 CREATE TABLE country (  
8     code VARCHAR(2) PRIMARY KEY,  
9     name VARCHAR(50)  
10 );  
11  
12 INSERT INTO country VALUES ('IN', 'India');  
13 INSERT INTO country VALUES ('US', 'United States of America');  
14
```

The output pane at the bottom displays the results of the executed commands:

#	Time	Action	Message	Duration / Fetch
✓ 2	21:33:51	USE ormlearn	0 row(s) affected	0.000 sec
✓ 3	21:33:55	CREATE TABLE country (code VARCHAR(2) PRIMARY KEY, name VARCHAR(50))	0 row(s) affected	0.109 sec
✓ 4	21:33:59	INSERT INTO country VALUES ('IN', 'India')	1 row(s) affected	0.031 sec
✓ 5	21:33:59	INSERT INTO country VALUES ('US', 'United States of America')	1 row(s) affected	0.000 sec

