

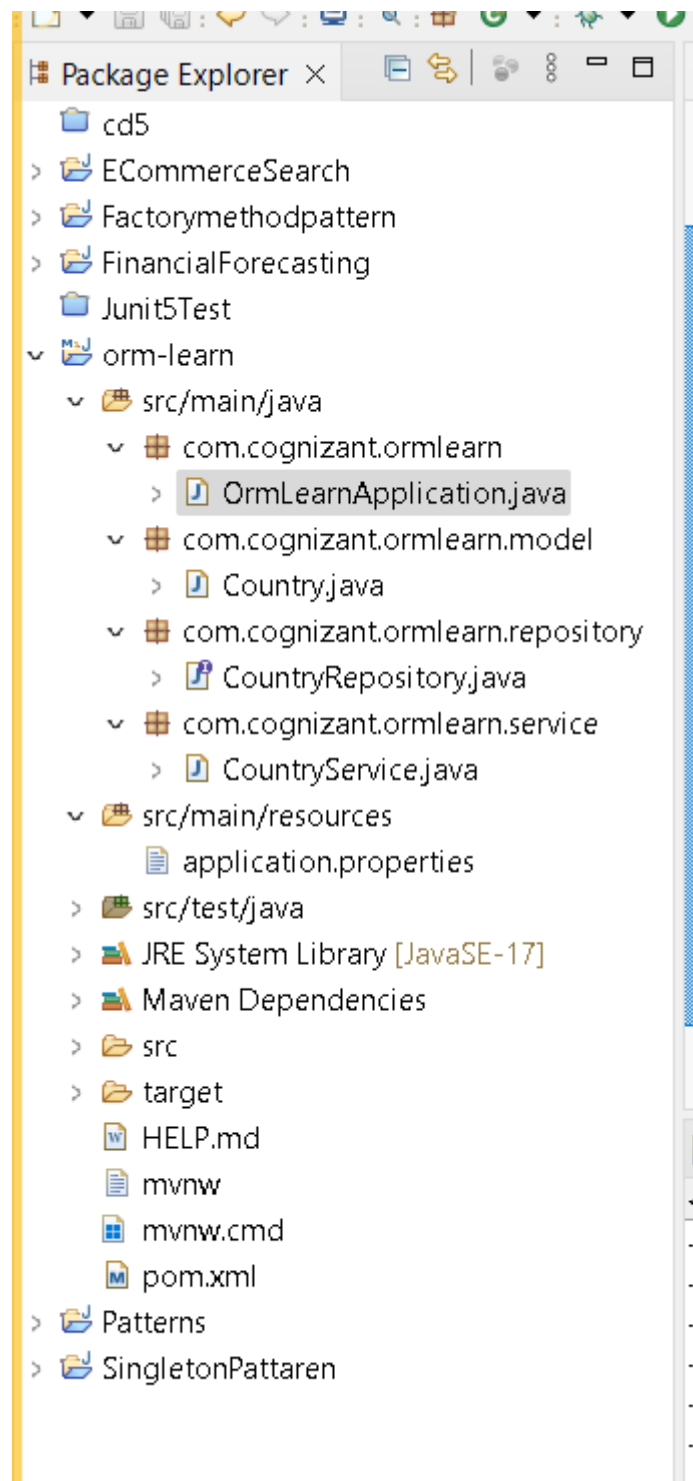
## Spring Data JPA - Quick Example

From File 1. spring-data-jpa-handson :

---

Create a Eclipse Project using Spring Initializr

Pic from Eclipse IDE:



**Country.java class from package com.cognizant.ormlearn.model:**

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

```
import jakarta.persistence.*;
```

```
@Entity
```

```
@Table(name = "country")
```

```
public class Country {
```

```
    @Id
```

```
    @Column(name = "code")
```

```
    private String code;
```

```
    @Column(name = "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() {

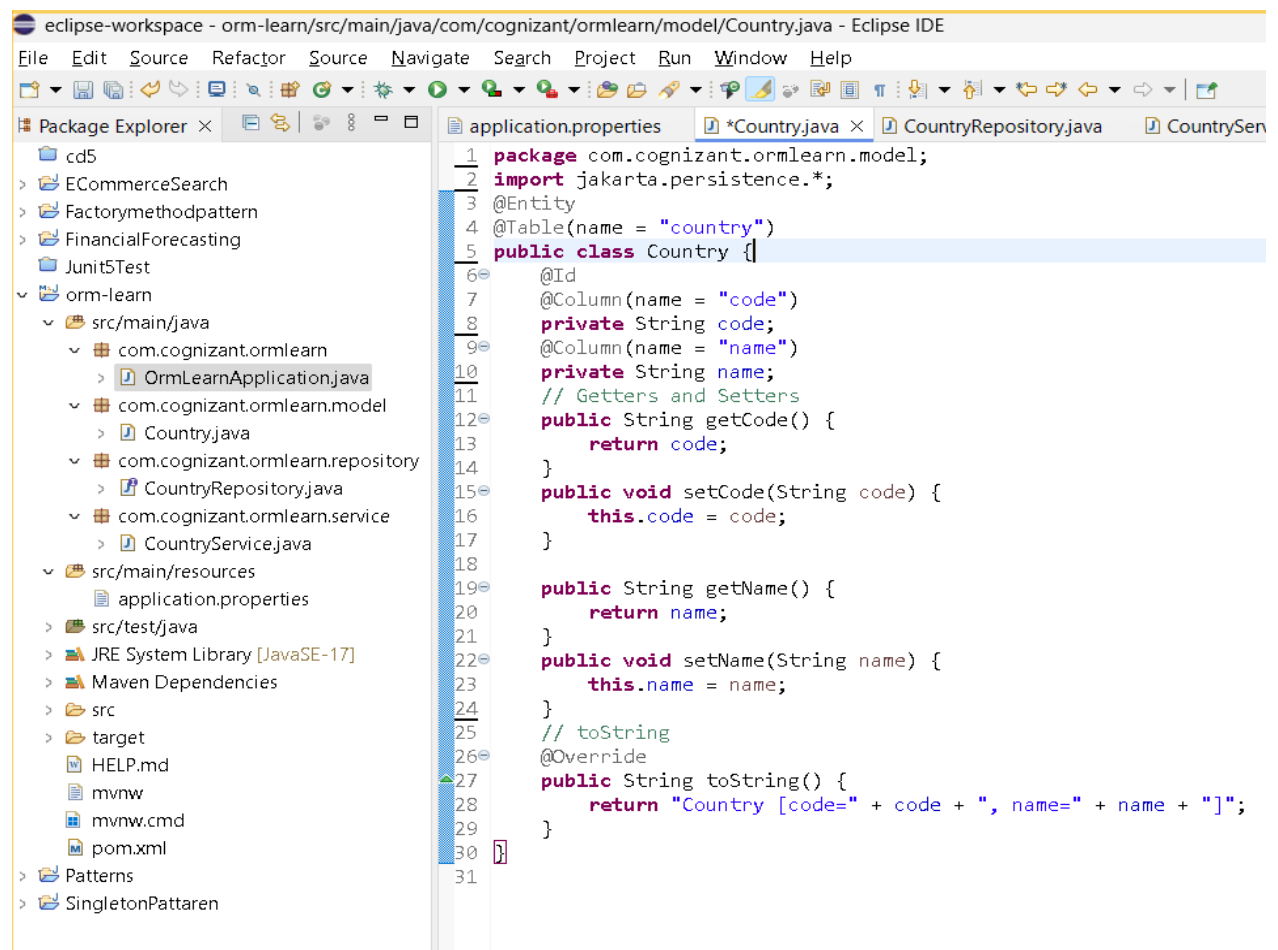
        return "Country [code=" + code + ", name=" + name + "];"

    }

}

```

### Pic from Eclipse IDE:



## Country.Repository.java from packagecom.cognizant.ormlearn.repository:

```
package com.cognizant.ormlearn.repository;

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

import org.springframework.stereotype.Repository;

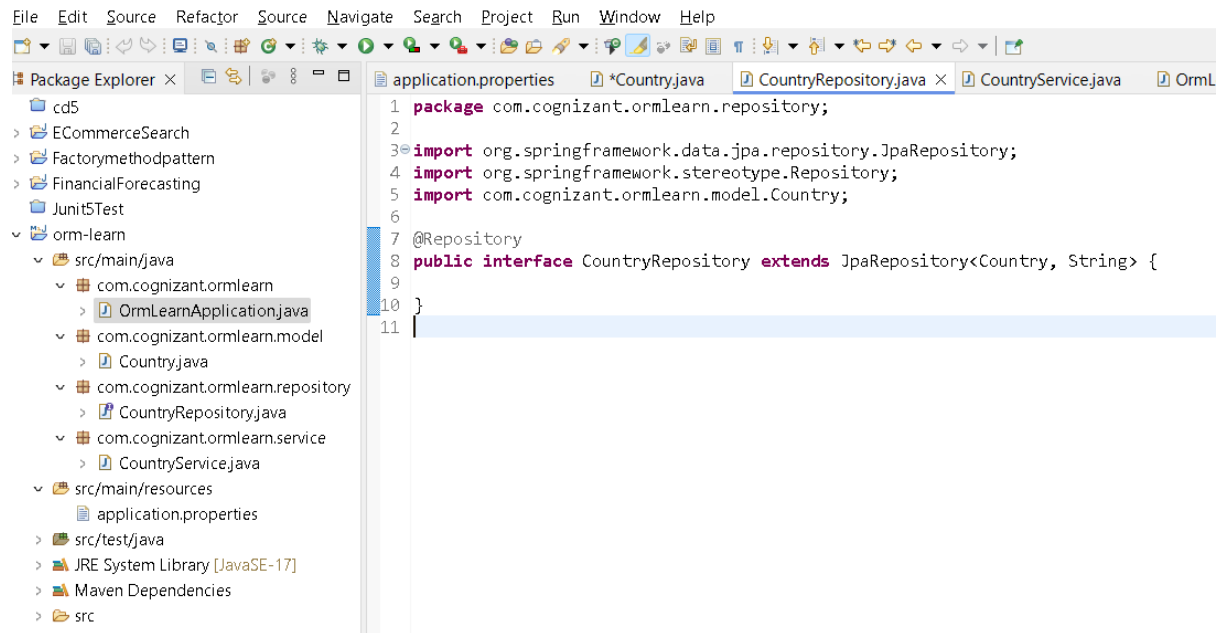
import com.cognizant.ormlearn.model.Country;

@Repository

public interface CountryRepository extends JpaRepository<Country,
String> {

}
```

### Pic From Eclipse IDE:



**CountryService.java class from package  
com.cognizant.ormlearn.service:**

```
package com.cognizant.ormlearn.service;  
  
import java.util.List;  
  
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;
```

@Service

```
public class CountryService {
```

@Autowired

```
private CountryRepository countryRepository;
```

@Transactional

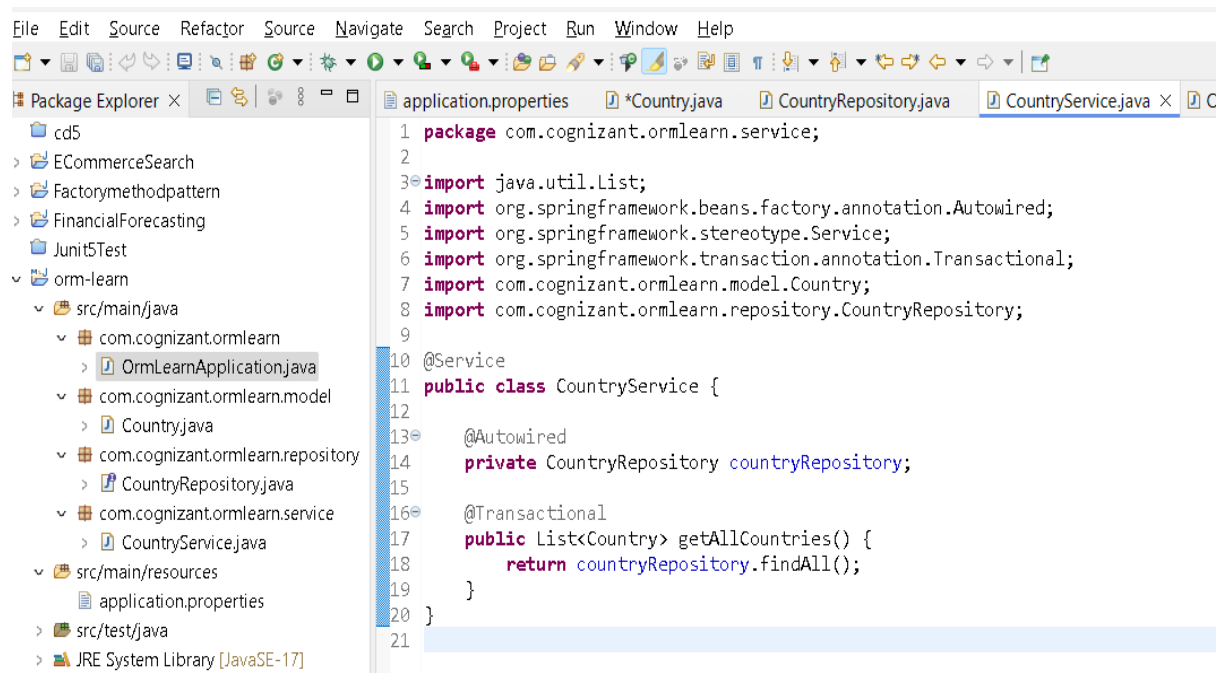
```
public List<Country> getAllCountries() {
```

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

```
}
```

```
}
```

## Pic From Eclipse IDE:



## OrmLearnApplication.java class

### Code:

```
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 com.cognizant.ormlearn.model.Country;
```

```
import com.cognizant.ormlearn.service.CountryService;

import jakarta.annotation.PostConstruct;

@SpringBootApplication
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);
    }

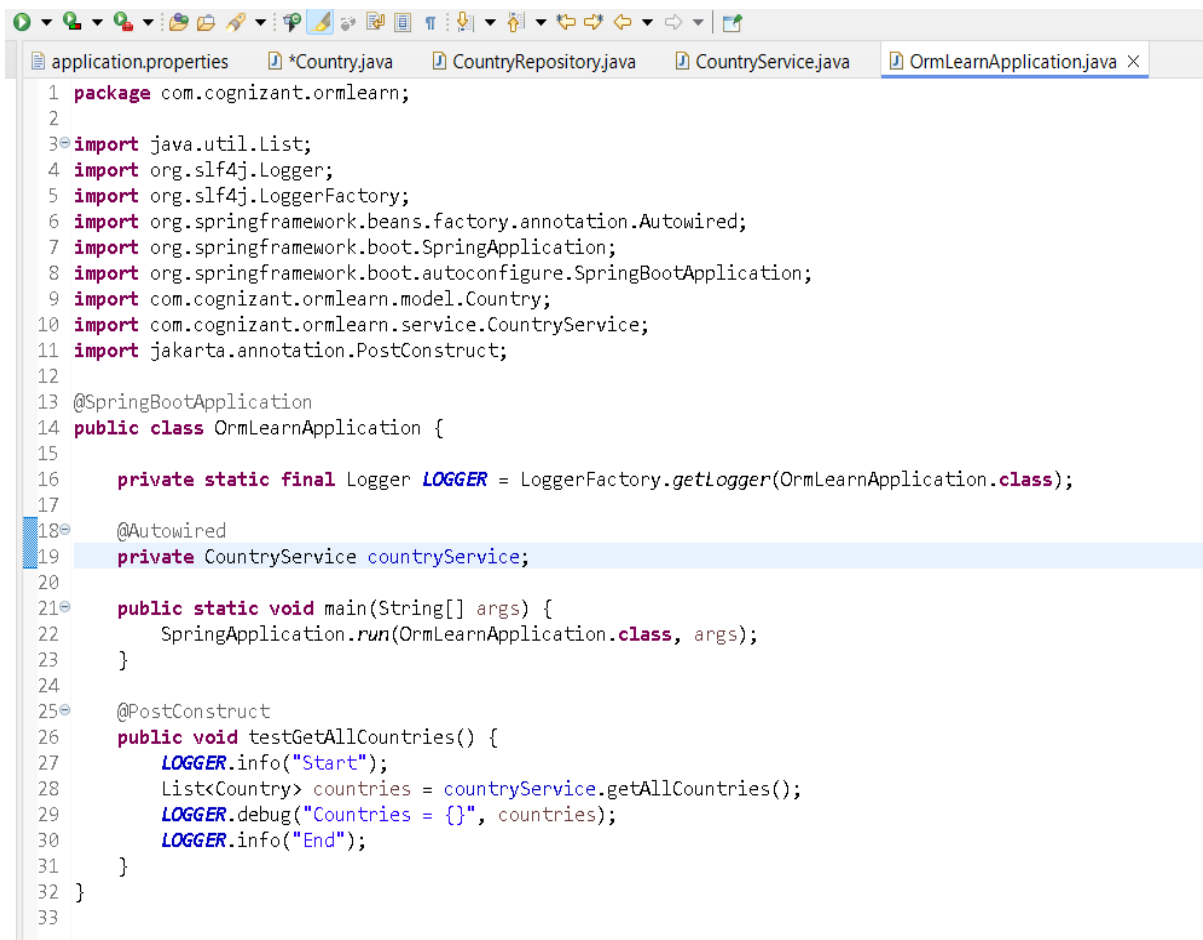

    @PostConstruct
    public void testGetAllCountries() {
        LOGGER.info("Start");

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

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

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

## Pic From Eclipse IDE:



```
1 package com.cognizant.ormLearn;
2
3 import java.util.List;
4 import org.slf4j.Logger;
5 import org.slf4j.LoggerFactory;
6 import org.springframework.beans.factory.annotation.Autowired;
7 import org.springframework.boot.SpringApplication;
8 import org.springframework.boot.autoconfigure.SpringBootApplication;
9 import com.cognizant.ormlearn.model.Country;
10 import com.cognizant.ormlearn.service.CountryService;
11 import jakarta.annotation.PostConstruct;
12
13 @SpringBootApplication
14 public class OrmLearnApplication {
15
16     private static final Logger LOGGER = LoggerFactory.getLogger(OrmLearnApplication.class);
17
18     @Autowired
19     private CountryService countryService;
20
21     public static void main(String[] args) {
22         SpringApplication.run(OrmLearnApplication.class, args);
23     }
24
25     @PostConstruct
26     public void testGetAllCountries() {
27         LOGGER.info("Start");
28         List<Country> countries = countryService.getAllCountries();
29         LOGGER.debug("Countries = {}", countries);
30         LOGGER.info("End");
31     }
32 }
33
```

## Application.properties code:

# Spring Framework and application log

logging.level.org.springframework=info

logging.level.com.cognizant=debug

# Hibernate logs

logging.level.org.hibernate.SQL=trace

logging.level.org.hibernate.type.descriptor.sql=trace

# Log pattern

logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %-  
20.20thread %5p %-25.25logger{25} %25M %4L %m%n



## # Database configuration

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=32190

## # Hibernate configuration

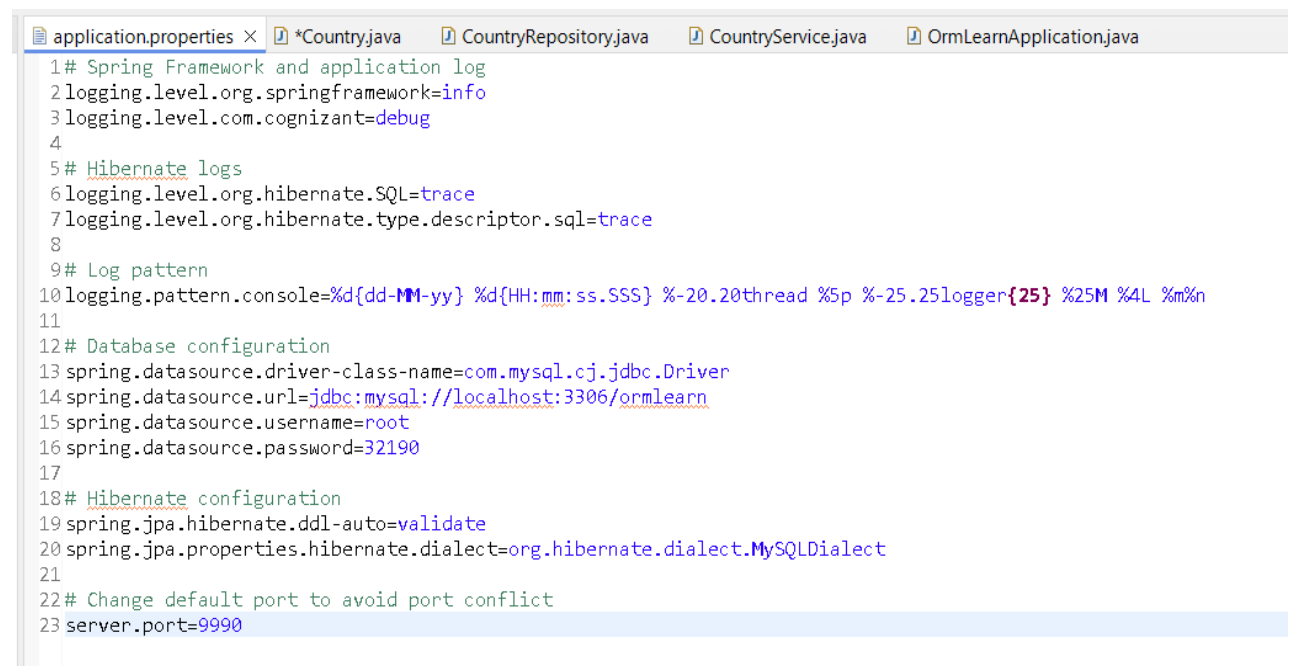
spring.jpa.hibernate.ddl-auto=validate

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL  
Dialect

## # Change default port to avoid port conflict

server.port=9990

## Pic From Eclipse IDE:

A screenshot of the Eclipse IDE interface. The top toolbar shows several open files: 'application.properties', '\*Country.java', 'CountryRepository.java', 'CountryService.java', and 'OrmLearnApplication.java'. The 'application.properties' file is selected and its content is displayed in the editor. The code is color-coded and includes comments for different configuration sections. The last line of the configuration, 'server.port=9990', is highlighted with a light blue background.

```
1# Spring Framework and application log
2logging.level.org.springframework=info
3logging.level.com.cognizant=debug
4
5# Hibernate logs
6logging.level.org.hibernate.SQL=trace
7logging.level.org.hibernate.type.descriptor.sql=trace
8
9# Log pattern
10logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %-20.20thread %5p %-25.25logger{25} %25M %4L %m%n
11
12# Database configuration
13spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
14spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn
15spring.datasource.username=root
16spring.datasource.password=32190
17
18# Hibernate configuration
19spring.jpa.hibernate.ddl-auto=validate
20spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
21
22# Change default port to avoid port conflict
23server.port=9990
```

## Pom.xml code:

```
<?xml version="1.0" encoding="UTF-8"?>

<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
    https://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <parent>

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

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

    <version>3.5.3</version>

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

  </parent>

  <groupId>com.cognizant</groupId>

  <artifactId>orm-learn</artifactId>

  <version>0.0.1-SNAPSHOT</version>

  <name>orm-learn</name>

  <description>Demo project for Spring Data JPA and
  Hibernate</description>

  <url/>

  <licenses>

    <license/>

  </licenses>
```

```
<developers>
  <developer/>
</developers>

<scm>
  <connection/>
  <developerConnection/>
  <tag/>
  <url/>
</scm>

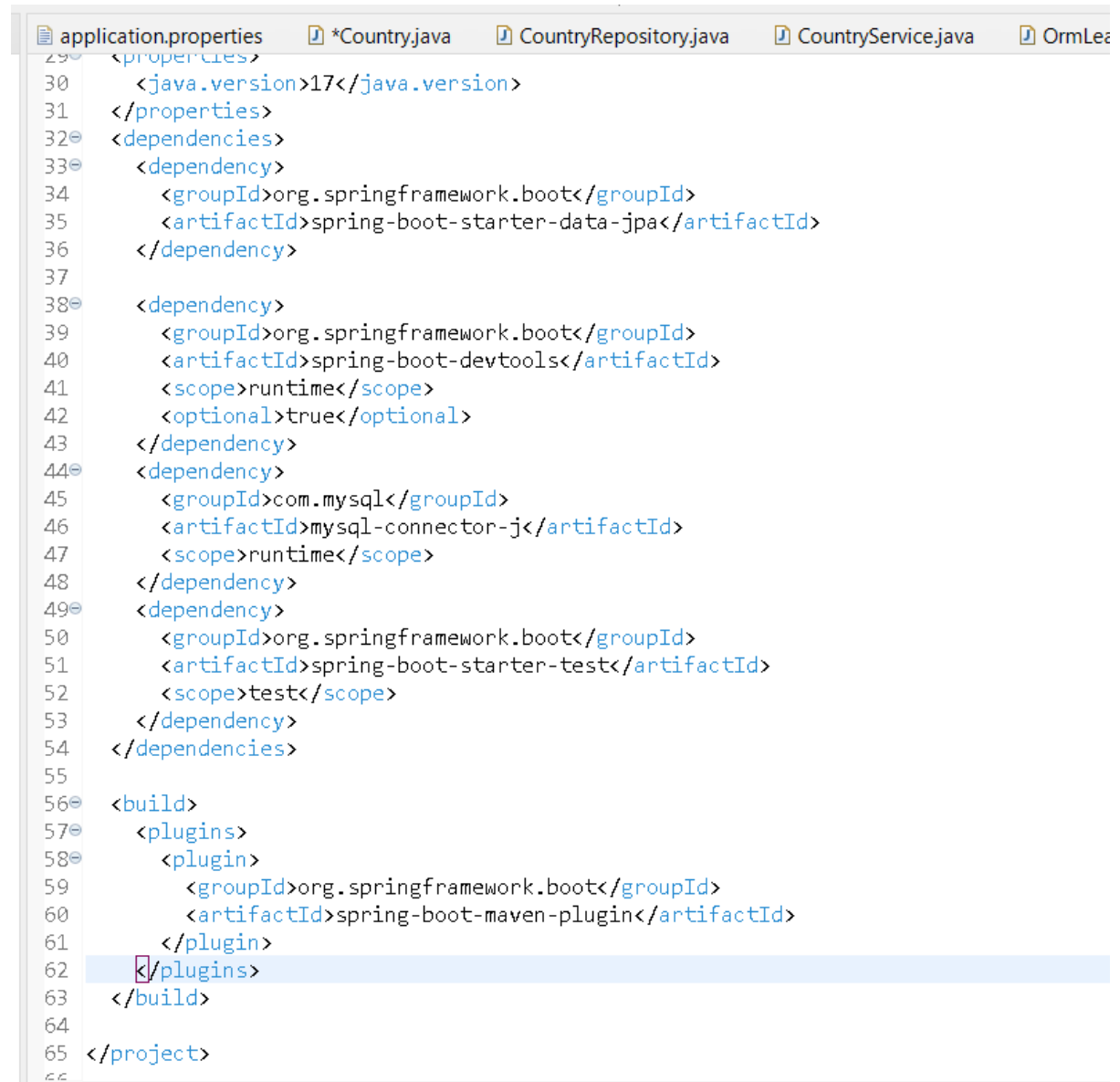
<properties>
  <java.version>17</java.version>
</properties>

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

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
```

```
</dependency>
<dependency>
  <groupId>com.mysql</groupId>
  <artifactId>mysql-connector-j</artifactId>
  <scope>runtime</scope>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
</project>
```

## PIC From Eclipse IDE:



```
29<properties>
30  <java.version>17</java.version>
31</properties>
32<dependencies>
33  <dependency>
34    <groupId>org.springframework.boot</groupId>
35    <artifactId>spring-boot-starter-data-jpa</artifactId>
36  </dependency>
37
38  <dependency>
39    <groupId>org.springframework.boot</groupId>
40    <artifactId>spring-boot-devtools</artifactId>
41    <scope>runtime</scope>
42    <optional>true</optional>
43  </dependency>
44  <dependency>
45    <groupId>com.mysql</groupId>
46    <artifactId>mysql-connector-j</artifactId>
47    <scope>runtime</scope>
48  </dependency>
49  <dependency>
50    <groupId>org.springframework.boot</groupId>
51    <artifactId>spring-boot-starter-test</artifactId>
52    <scope>test</scope>
53  </dependency>
54</dependencies>
55
56<build>
57  <plugins>
58    <plugin>
59      <groupId>org.springframework.boot</groupId>
60      <artifactId>spring-boot-maven-plugin</artifactId>
61    </plugin>
62  </plugins>
63</build>
64
65</project>
66
```

## SQL Command for Creating Database :

## PIC From Command prompt:

```
Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create schema ormlearn;
Query OK, 1 row affected (0.06 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| campus_portal |
| deep_skilling |
| employee_management_system |
| information_schema |
| jfsd |
| jfsd1 |
| klu |
| mysql |
| onlinesdp |
| ormlearn |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
14 rows in set (0.00 sec)

mysql> use ormlearn;
Database changed
mysql> create table country (
->     code varchar(2) primary key,
->     name varchar(50)
-> );
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> use ormlearn;
Database changed
mysql> create table country (
->     code varchar(2) primary key,
->     name varchar(50)
-> );
Query OK, 0 rows affected (0.08 sec)

mysql> select * from country;
Empty set (0.01 sec)

mysql> insert into country values ('IN', 'India');
Query OK, 1 row affected (0.01 sec)

mysql> insert into country values ('US', 'United States of America');
Query OK, 1 row affected (0.01 sec)

mysql> select * from country;
+-----+
| code | name |
+-----+
| IN   | India |
| US   | United States of America |
+-----+
2 rows in set (0.00 sec)

mysql> |
```

## OUTPUT AFTER RUNING OrmApplication.java

testGetAllCountries 29 Countries = [Country [code=IN, name=India],  
Country [code=US, name=United States of America]]

### Pic From Eclipse IDEL:

```
DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(-3, org.hibernate.type.descriptor.sql.internal.CapacityDependentDdlType@1bcfa0df) replaced pr
DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(4003, org.hibernate.type.descriptor.sql.internal.DdlTypeImpl@5162a961) replaced previous regi
DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(4001, org.hibernate.type.descriptor.sql.internal.DdlTypeImpl@6f0ba18f) replaced previous regi
DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(4002, org.hibernate.type.descriptor.sql.internal.DdlTypeImpl@4fb469d0) replaced previous regi
DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(2004, org.hibernate.type.descriptor.sql.internal.CapacityDependentDdlType@5504ce7b) replaced
DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(2005, org.hibernate.type.descriptor.sql.internal.CapacityDependentDdlType@7d8c2782) replaced
DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(2011, org.hibernate.type.descriptor.sql.internal.CapacityDependentDdlType@61304817) replaced
INFO .p.i.JtaPlatformInitiator initiateService 59 HHH000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integ
INFO nEntityManagerFactoryBean buildNativeEntityManagerFactory 447 Initialized JPA EntityManagerFactory for persistence unit 'default'
INFO c.c.o.OrmLearnApplication testGetAllCountries 27 Start
DEBUG org.hibernate.SQL logStatement 135 select c1_0.code,c1_0.name from country c1_0
testGetAllCountries 29 Countries = [Country [code=IN, name=India], Country [code=US, name=United States of America]]
INFO c.c.o.OrmLearnApplication testGetAllCountries 30 End
INFO .OptionalLiveReloadServer startServer 59 LiveReload server is running on port 35729
INFO c.c.o.OrmLearnApplication logStarted 59 Started OrmLearnApplication in 5.562 seconds (process running for 6.317)
ook INFO nEntityManagerFactoryBean destroy 660 Closing JPA EntityManagerFactory for persistence unit 'default'
ook INFO c.z.h.HikariDataSource close 349 HikariPool-1 - Shutdown initiated...
ook INFO c.z.h.HikariDataSource close 351 HikariPool-1 - Shutdown completed.
```