

Spring Data JPA - Quick Example

Application.properties:

```
spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn1
spring.datasource.username=root
spring.datasource.password=Gayatri@81$
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.hibernate.ddl-auto=none
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
server.port=8086
```

ormLearnApplication.java:

```
package com.cognizant.ormlearn;

import com.cognizant.ormlearn.model.Country;
import com.cognizant.ormlearn.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);

LOGGER.info("Inside main");

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");

}
}

```

Country.java:

```

package com.cognizant.ormlearn.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", nullable = false, length = 2)

    private String code;

    @Column(name = "co_name", nullable = false, length = 50)

    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 + "]";
}
}

```

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 org.springframework.transaction.annotation.Transactional;

import com.cognizant.ormlearn.model.Country;
import com.cognizant.ormlearn.repository.CountryRepository;

@Service

public class CountryService {

    private final CountryRepository countryRepository;

    @Autowired

    public CountryService(CountryRepository countryRepository) {

        this.countryRepository = countryRepository;

    }

    @Transactional(readOnly = true)
```

```

public List<Country> getAllCountries() {

    return countryRepository.findAll();

}

}

```

OUTPUT:

The screenshot shows the Eclipse IDE with the following components:

- Package Explorer:** Shows the project structure with packages like `cognizant`, `demo`, `demo_demo`, `unit_demo`, `LibraryManagement`, and `orm-learn`.
- Editor:** Displays the `OrmLearnApplication.java` file. The code includes imports for `com.cognizant.ormlearn.model.Country`, `com.cognizant.ormlearn.service.CountryService`, `org.slf4j.Logger`, `org.slf4j.LoggerFactory`, `org.springframework.boot.SpringApplication`, `org.springframework.boot.autoconfigure.SpringBootApplication`, `org.springframework.context.ApplicationContext`, and `java.util.List`. The `main` method uses `SpringApplication.run` and `context.getBean` to start the application.
- Console:** Shows the output of the application. It includes logs for the `BytecodeProviderInitiator`, `JtaPlatformInitiator`, `LocalContainerEntityManagerFactoryBean`, `JpaBaseConfiguration$JpaWebConfiguration`, `OptionalLiveReloadServer`, `TomcatWebServer`, and the `OrmLearnApplication` itself. The logs indicate that the application started successfully on port 8086.

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

Navigator

MANAGEMENT

Server Status

Client Connections

Users and Privileges

Status and System Variables

Data Export

Data Import/Restore

INSTANCE

Startup / Shutdown

Server Logs

Options File

PERFORMANCE

Dashboard

Performance Reports

Performance Schema Setup

AdministrationSchemas

Information

SQL File 2*

Limit to 1000 rows

1 create schema omlearn1;

2

3 use omlearn1;

4

5 create table country (

6 co_code varchar(2) primary key,

7 co_name varchar(50)

8);

9

10 insert into country values ('IN', 'India');

11 insert into country values ('US', 'United States of America');

12

SQLAdditions

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

Context HelpSnippets

No object selected

Output

Action Output

#	Time	Action	Message	Duration / Fetch
13	10:24:15	create schema omlearn1	1 row(s) affected	0.000 sec
14	10:24:23	use omlearn1	0 row(s) affected	0.000 sec
15	10:24:23	create table country (co_code varchar(2) primary key, co_name varchar(50))	0 row(s) affected	0.125 sec
16	10:24:29	insert into country values ('IN', 'India')	1 row(s) affected	0.015 sec
17	10:24:29	insert into country values ('US', 'United States of America')	1 row(s) affected	0.016 sec

Object InfoSession