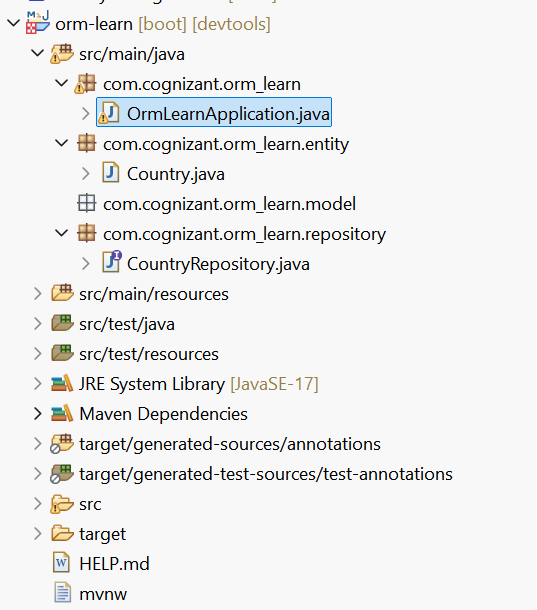
**Spring Data JPA - Quick Example**

**File Structure:**  
  
**Pom.xml:** <?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 Boot</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>

<dependency>

<groupId>jakarta.persistence</groupId>

<artifactId>jakarta.persistence-api</artifactId>

<version>3.1.0</version>

<dependency>

<dependency>

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

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

</dependency>

<!-- Spring Boot Starter JDBC -->

<dependency>

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

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

</dependency>

<dependency>

<groupId>com.mysql</groupId>

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

<scope>runtime</scope>

</dependency>

<!-- Spring Boot Starter -->

<dependency>

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

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

</dependency>

<groupId>com.mysql</groupId>

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

<version>8.0.33</version> <!-- Or latest version -->

</dependency>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**Country.java:  
package** com.cognizant.orm\_learn.entity;

**import** jakarta.persistence.Entity;

**import** jakarta.persistence.Id;

**import** jakarta.persistence.Table;

**import** jakarta.persistence.Column;

@Entity

@Table(name = "country")

**public** **class** Country {

@Id

@Column(name = "code")

**private** String code;

@Column(name = "name")

**private** String name;

// --- Getters ---

**public** String getCode() {

**return** code;

}

**public** String getName() {

**return** name;

}

// --- Setters ---

**public** **void** setCode(String code) {

**this**.code = code;

}

**public** **void** setName(String name) {

**this**.name = name;

}

// --- Optional toString ---

@Override

**public** String toString() {

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

}

}

**MainApplication:  
package** com.cognizant.orm\_learn;

**import** org.springframework.boot.CommandLineRunner;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** com.cognizant.orm\_learn.entity.Country;

**import** com.cognizant.orm\_learn.repository.CountryRepository;

**import** java.util.List;

@SpringBootApplication

**public** **class** OrmLearnApplication **implements** CommandLineRunner {

@Autowired

**private** CountryRepository countryRepository;

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(OrmLearnApplication.**class**, args);

}

@Override

**public** **void** run(String... args) **throws** Exception {

testDatabaseConnection();

}

**private** **void** testDatabaseConnection() {

System.***out***.println("=== All Countries from DB ===");

List<Country> countries = countryRepository.findAll();

**for** (Country country : countries) {

System.***out***.printf("Code: %-5s | Name: %-20s%n", country.getCode(), country.getName());

}

System.***out***.println("=== End of List ===");

}

}

**Application.Propertires:**spring.application.name=orm-learn

# 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

# Console 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 connection

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=password

# Hibernate

spring.jpa.hibernate.ddl-auto=validate

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect

**CountryRepository.java:**

**package** com.cognizant.orm\_learn.repository;

**import** org.springframework.stereotype.Repository;

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

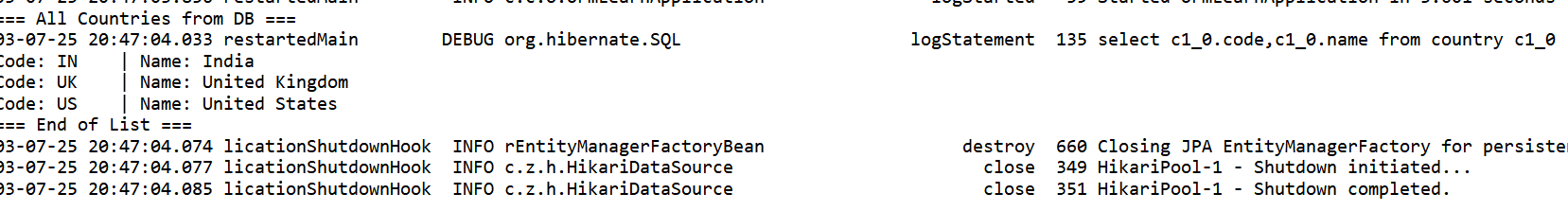
**import** com.cognizant.orm\_learn.entity.Country;

@Repository

**public** **interface** CountryRepository **extends** JpaRepository<Country, String> {

// No extra code needed for now

}

**Ouptut:**

**Difference between JPA, Hibernate and Spring Data JPA**   
  
Java Persistence API (JPA)

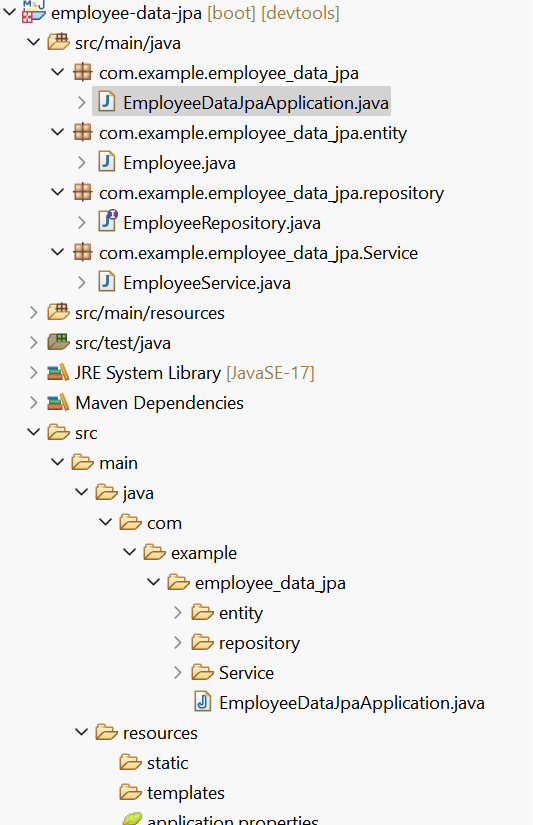
* JSR 338 Specification for persisting, reading and managing data from Java objects
* Does not contain concrete implementation of the specification
* Hibernate is one of the implementation of JPA

Hibernate

* ORM Tool that implements JPA

Spring Data JPA

* Does not have JPA implementation, but reduces boiler plate code
* This is another level of abstraction over JPA implementation provider like Hibernate
* Manages transactions

**File Structure:**

**MainApplication.java:  
package** com.example.employee\_data\_jpa;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.boot.CommandLineRunner;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** com.example.employee\_data\_jpa.entity.Employee;

**import** com.example.employee\_data\_jpa.Service.EmployeeService; // Corrected import: 'Service' with capital 'S'

@SpringBootApplication

**public** **class** EmployeeDataJpaApplication **implements** CommandLineRunner {

@Autowired

**private** EmployeeService employeeService;

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(EmployeeDataJpaApplication.**class**, args);

}

@Override

**public** **void** run(String... args) {

Employee e = **new** Employee();

e.setName("John Doe");

e.setDepartment("HR");

e.setSalary(45000.0);

employeeService.save(e);

employeeService.getAllEmployees().forEach(emp ->

System.***out***.println(emp.getId() + " | " + emp.getName() + " | " + emp.getDepartment())

);

}

}

**Service.java:  
package** com.example.employee\_data\_jpa.Service;

**import** java.util.List;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** com.example.employee\_data\_jpa.entity.Employee;

**import** com.example.employee\_data\_jpa.repository.EmployeeRepository;

@Service

**public** **class** EmployeeService {

@Autowired

**private** EmployeeRepository employeeRepository;

**public** Employee save(Employee emp) {

**return** employeeRepository.save(emp);

}

**public** List<Employee> getAllEmployees() {

**return** employeeRepository.findAll();

}

}

**Pom.xml:**<?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.example</groupId>

<artifactId>employee-data-jpa</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>employee-data-jpa</name>

<description>Demo project for Spring Boot</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-starter-web</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>

**Employee.java:  
package** com.example.employee\_data\_jpa.entity;

**import** jakarta.persistence.\*;

@Entity

@Table(name = "employee")

**public** **class** Employee {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

**private** **int** id;

**private** String name;

**private** String department;

**private** **double** salary;

// 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;

}

**public** **double** getSalary() {

**return** salary;

}

**public** **void** setSalary(**double** salary) {

**this**.salary = salary;

}

}

**Repo.java:  
package** com.example.employee\_data\_jpa.repository;

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

**import** com.example.employee\_data\_jpa.entity.Employee;

**public** **interface** EmployeeRepository **extends** JpaRepository<Employee, Integer> {

}

**Application.properties:**

spring.application.name=employee-data-jpa

# MySQL Configuration

spring.datasource.url=jdbc:mysql://localhost:3306/employee\_db

spring.datasource.username=root

spring.datasource.password=password

# JPA & Hibernate Settings

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

# spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect # This line can be removed as per the previous warning

**Output:**. \_\_\_\_ \_ \_\_ \_ \_

/\\ / \_\_\_'\_ \_\_ \_ \_(\_)\_ \_\_ \_\_ \_ \ \ \ \

( ( )\\_\_\_ | '\_ | '\_| | '\_ \/ \_` | \ \ \ \

\\/ \_\_\_)| |\_)| | | | | || (\_| | ) ) ) )

' |\_\_\_\_| .\_\_|\_| |\_|\_| |\_\\_\_, | / / / /

=========|\_|==============|\_\_\_/=/\_/\_/\_/

:: Spring Boot :: (v3.5.3)

Database driver: undefined/unknown

Database version: 8.0.42

Autocommit mode: undefined/unknown

Isolation level: undefined/unknown

Minimum pool size: undefined/unknown

Maximum pool size: undefinedHibernate: insert into employee (department,name,salary) values (?,?,?)

Hibernate: select e1\_0.id,e1\_0.department,e1\_0.name,e1\_0.salary from employee e1\_0

1 | John Doe | HR

