**Spring Data JPA - Quick Example**   
  
**Software Pre-requisites**

* MySQL Server 8.0
* MySQL Workbench 8
* Eclipse IDE for Enterprise Java Developers 2019-03 R
* Maven 3.6.2

**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>ormlearn</artifactId>

<version>1.0-SNAPSHOT</version>

<packaging>jar</packaging>

<parent>

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

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

<version>2.5.4</version>

</parent>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

</dependency>

<dependency>

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

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

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

**application.properties:**

spring.h2.console.enabled=true

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driver-class-name=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.show-sql=true

spring.jpa.hibernate.ddl-auto=update

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

**Country.java:**

package com.cognizant.ormlearn.model;

import javax.persistence.\*;

@Entity

@Table(name = "country")

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 com.cognizant.ormlearn.model.Country;

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

import org.springframework.stereotype.Repository;

@Repository

public interface CountryRepository extends JpaRepository<Country, String>

{

}

**CountryService.java:**

package com.cognizant.ormlearn.service;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.repository.CountryRepository;

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

import org.springframework.stereotype.Service;

import javax.transaction.Transactional;

import java.util.List;

@Service

public class CountryService

{

@Autowired

private CountryRepository countryRepository;

@Transactional

public List<Country> getAllCountries()

{

return countryRepository.findAll();

}

}

**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);

countryService = context.getBean(CountryService.class);

LOGGER.info("Inside main");

testGetAllCountries();

}

private static void testGetAllCountries()

{

LOGGER.info("Start");

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

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

LOGGER.info("End");

}

}

**.replit:**

run = "mvn clean spring-boot:run"

entrypoint = "src/main/java/com/cognizant/ormlearn/OrmLearnApplication.java"

[packager]

language = "java"

[languages.java]

pattern = "\*\*/\*.java"

**replit.nix:**

{ pkgs }:

{

deps = [

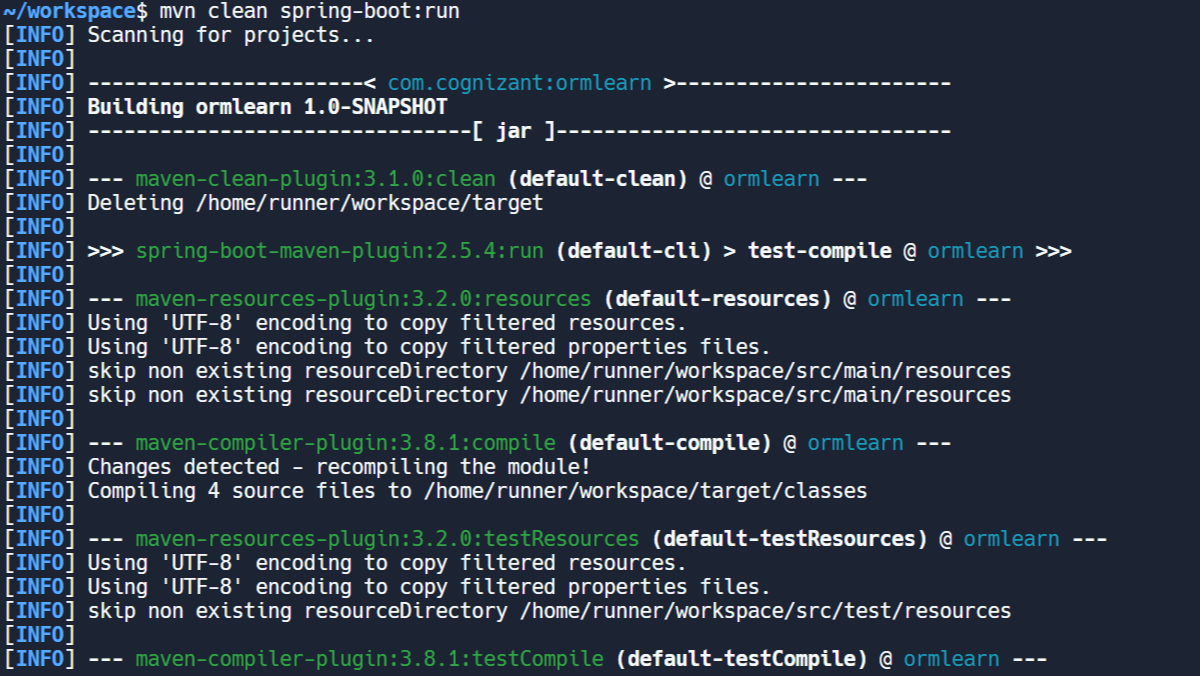
pkgs.maven

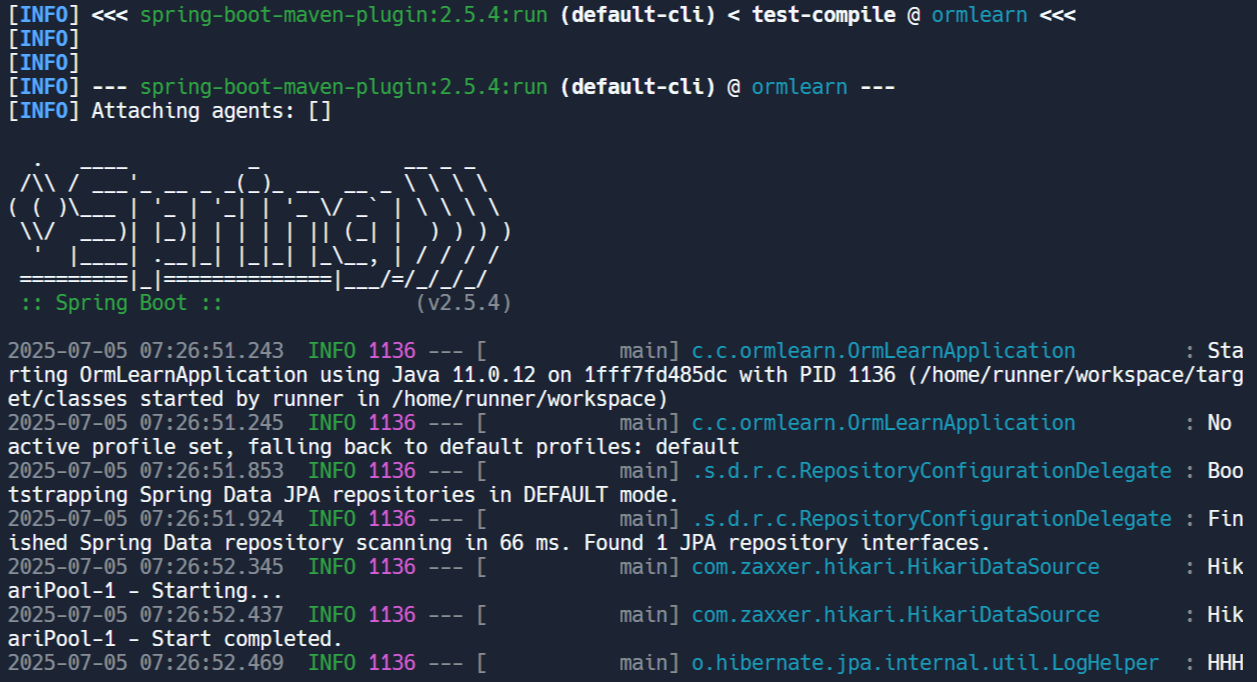
pkgs.openjdk11

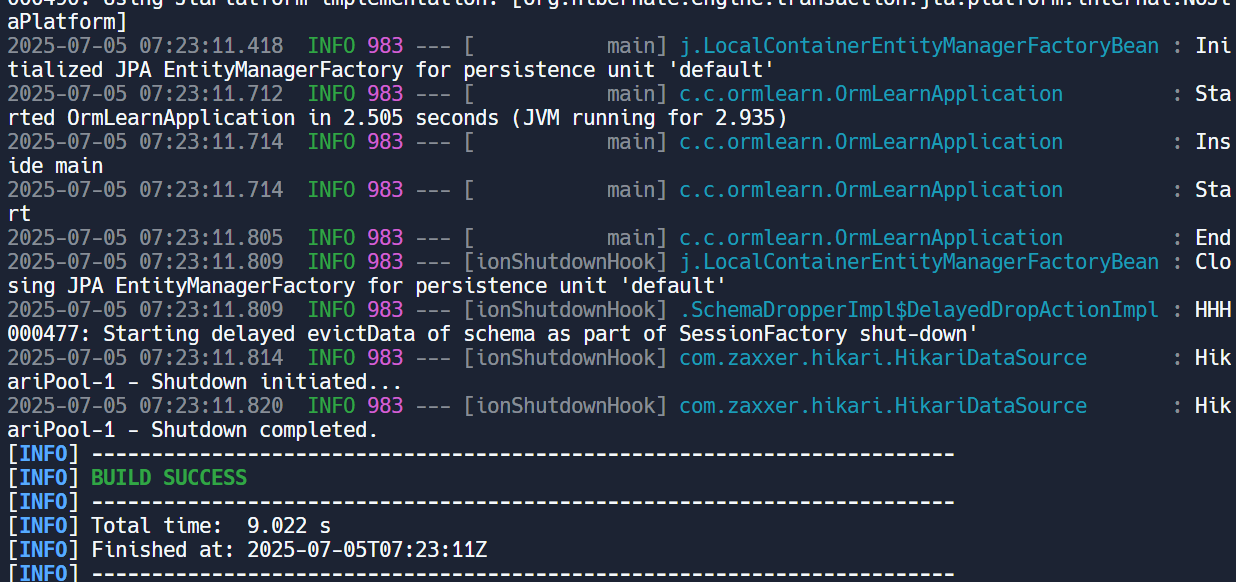
];

}

**Result:**







**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

**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>ormlearn</artifactId>

<version>1.0-SNAPSHOT</version>

<parent>

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

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

<version>2.5.4</version>

</parent>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

</dependency>

</dependencies>

</project>

**application.properties:**

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driver-class-name=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

**Employee.java:**

package com.cognizant.ormlearn.model;

import javax.persistence.\*;

@Entity

@Table(name = "employee")

public class Employee

{

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Integer id;

@Column

private String name;

@Column

private String department;

public Integer getId()

{

return id;

}

public void setId(Integer 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 + "'}";

}

}

**EmployeeRepository.java:**

package com.cognizant.ormlearn.repository;

import com.cognizant.ormlearn.model.Employee;

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

import org.springframework.stereotype.Repository;

@Repository

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

package com.cognizant.ormlearn.service;

import com.cognizant.ormlearn.model.Employee;

import com.cognizant.ormlearn.repository.EmployeeRepository;

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

import org.springframework.stereotype.Service;

**EmployeeService.java:**

import javax.transaction.Transactional;

import java.util.List;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee)

{

employeeRepository.save(employee);

}

@Transactional

public List<Employee> getAllEmployees()

{

return employeeRepository.findAll();

}

}

**OrmLearnApplication.java:**

package com.cognizant.ormlearn;

import com.cognizant.ormlearn.model.Employee;

import com.cognizant.ormlearn.service.EmployeeService;

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

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 EmployeeService employeeService;

public static void main(String[] args)

{

ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);

employeeService = context.getBean(EmployeeService.class);

testAddEmployee();

testGetAllEmployees();

}

private static void testAddEmployee()

{

Employee emp1 = new Employee();

emp1.setName("Alice");

emp1.setDepartment("HR");

Employee emp2 = new Employee();

emp2.setName("Bob");

emp2.setDepartment("IT");

employeeService.addEmployee(emp1);

employeeService.addEmployee(emp2);

}

private static void testGetAllEmployees()

{

List<Employee> employees = employeeService.getAllEmployees();

for (Employee emp : employees)

{

System.out.println(emp);

}

}

}

**.replit:**

run = "mvn clean spring-boot:run"

entrypoint = "src/main/java/com/cognizant/ormlearn/OrmLearnApplication.java"

[packager]

language = "java"

[languages.java]

pattern = "\*\*/\*.java"

**replit.nix:**

{ pkgs }: {

deps = [

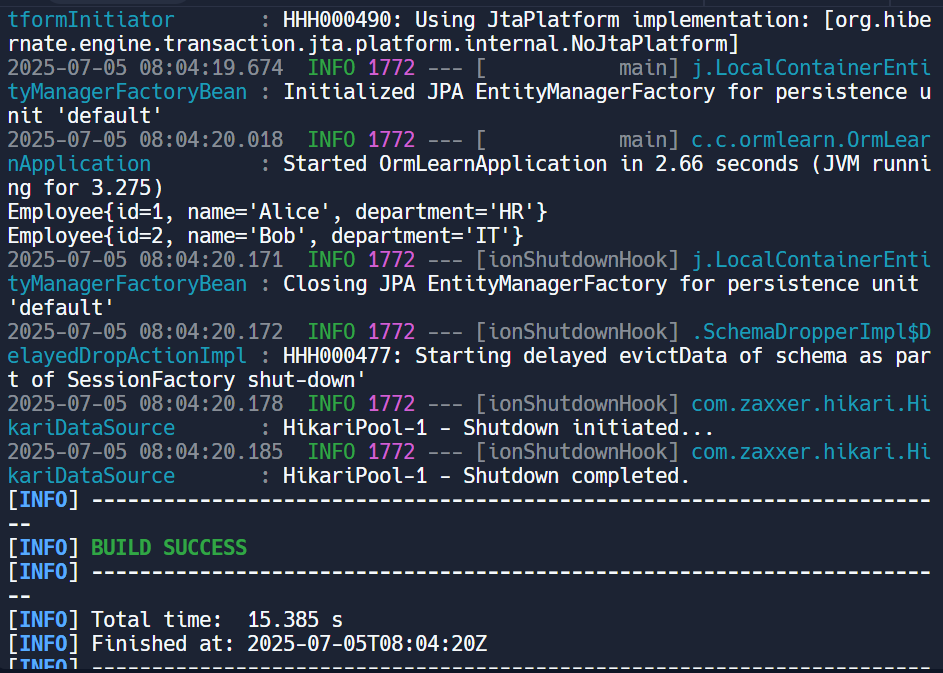
pkgs.maven

pkgs.openjdk11

];

}

**Result:**



**Implement services for managing Country**   
  
An application requires for features to be implemented with regards to country. These features needs to be supported by implementing them as service using 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>country-demo</artifactId>

<version>1.0.0</version>

<packaging>jar</packaging>

<parent>

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

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

<version>3.2.0</version>

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

</parent>

<properties>

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

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>com.mysql</groupId>

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

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<optional>true</optional>

</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>

**application.properties:**

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

spring.datasource.username=root

spring.datasource.password=your\_password

spring.jpa.hibernate.ddl-auto=validate

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

**CountryController.java:**

package com.example.demo.controller;

import com.example.demo.model.Country;

import com.example.demo.service.CountryService;

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

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/countries")

public class CountryController

{

@Autowired

private CountryService countryService;

@GetMapping("/{code}")

public ResponseEntity<Country> getCountryByCode(@PathVariable String code) {

return countryService.findCountryByCode(code)

.map(ResponseEntity::ok)

.orElse(ResponseEntity.notFound().build());

}

@PostMapping

public Country addCountry(@RequestBody Country country)

{

return countryService.addCountry(country);

}

@PutMapping

public Country updateCountry(@RequestBody Country country)

{

return countryService.updateCountry(country);

}

@DeleteMapping("/{code}")

public ResponseEntity<Void> deleteCountry(@PathVariable String code)

{

countryService.deleteCountry(code);

return ResponseEntity.ok().build();

}

@GetMapping("/search")

public List<Country> searchByPartialName(@RequestParam String name)

{

return countryService.findByPartialName(name);

}

@GetMapping

public List<Country> getAllCountries()

{

return countryService.getAllCountries();

}

}

**Country.java:**

package com.example.demo.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

@Entity

@Table(name = "country")

public class Country

{

@Id

private String coCode;

private String coName;

public String getCoCode()

{

return coCode;

}

public void setCoCode(String coCode)

{

this.coCode = coCode;

}

public String getCoName()

{

return coName;

}

public void setCoName(String coName)

{

this.coName = coName;

}

}

**CountryRepository.java:**

package com.example.demo.repository;

import com.example.demo.model.Country;

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

import java.util.List;

public interface CountryRepository extends JpaRepository<Country, String>

{

List<Country> findByCoNameContainingIgnoreCase(String coName);

}

**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 java.util.List;

import java.util.Optional;

@Service

public class CountryService

{

@Autowired

private CountryRepository countryRepo;

public Optional<Country> findCountryByCode(String coCode)

{

return countryRepo.findById(coCode);

}

public Country addCountry(Country country)

{

return countryRepo.save(country);

}

public Country updateCountry(Country country)

{

return countryRepo.save(country);

}

public void deleteCountry(String coCode)

{

countryRepo.deleteById(coCode);

}

public List<Country> findByPartialName(String partialName)

{

return countryRepo.findByCoNameContainingIgnoreCase(partialName);

}

public List<Country> getAllCountries()

{

return countryRepo.findAll();

}

}

**CountryDemoApplication.java:**

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class CountryDemoApplication

{

public static void main(String[] args)

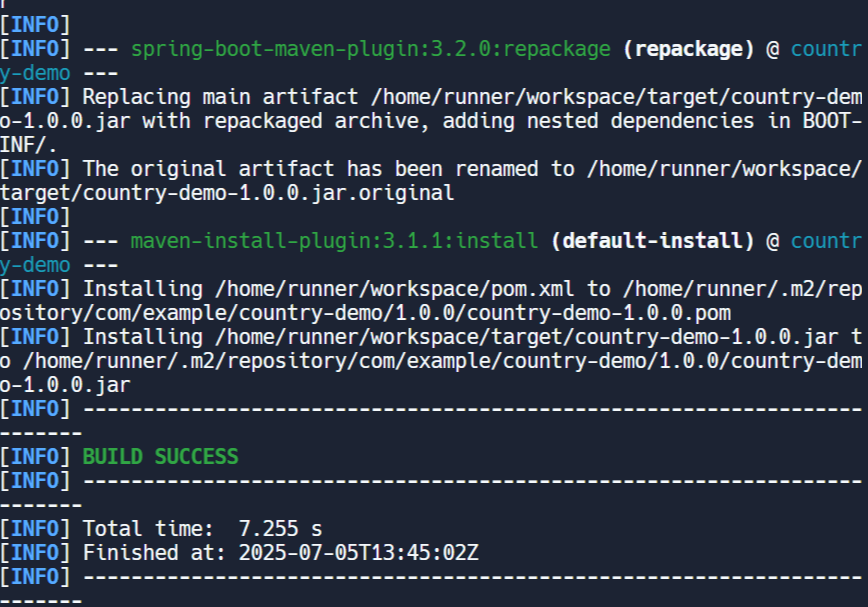
{

SpringApplication.run(CountryDemoApplication.class, args);

}

}

**Result:**



**Add a new country**

**CountryDemoApplication.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.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class CountryDemoApplication implements CommandLineRunner

{

@Autowired

private CountryService countryService;

public static void main(String[] args)

{

SpringApplication.run(CountryDemoApplication.class, args);

}

@Override

public void run(String... args)

{

testAddCountry();

}

private void testAddCountry()

{

Country country = new Country();

country.setCode("XY");

country.setName("Xyland");

countryService.addCountry(country);

Country saved = countryService.findCountryByCode("XY");

System.out.println("Saved Country: " + saved.getCode() + " - " + saved.getName());

}

}

**Country.java:**

package com.example.demo.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

@Entity

@Table(name = "country")

public class Country

{

@Id

private String code;

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;

}

}

**CountryRepository.java:**

package com.example.demo.repository;

import com.example.demo.model.Country;

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

public interface CountryRepository extends JpaRepository<Country, String> {

}

**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 org.springframework.transaction.annotation.Transactional;

@Service

public class CountryService

{

@Autowired

private CountryRepository countryRepo;

public Country findCountryByCode(String code)

{

return countryRepo.findById(code).orElse(null);

}

@Transactional

public void addCountry(Country country)

{

countryRepo.save(country);

}

}

**application.properties:**

spring.datasource.url=jdbc:mysql://localhost:3306/orm\_learn

spring.datasource.username=root

spring.datasource.password=your\_password

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

**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>country-demo</artifactId>

<version>1.0.0</version>

<packaging>jar</packaging>

<name>Country Demo</name>

<description>Spring Boot project for Country CRUD operations</description>

<parent>

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

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

<version>3.2.0</version>

<relativePath/>

</parent>

<properties>

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

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>com.mysql</groupId>

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

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>jakarta.persistence</groupId>

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

<version>3.1.0</version>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>1.18.30</version>

<scope>provided</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>

<version>3.2.0</version>

</plugin>

</plugins>

</build>

</project>

**Result:**

