**Spring Data JPA - Quick Example**

**Pom.xml**

<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>

**Application.properties:**

spring.application.name=spring-data-jpa-demo

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/spring\_data\_jpa\_demo

spring.datasource.username=root

spring.datasource.password=02July@2005

spring.jpa.hibernate.ddl-auto=validate

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

logging.level.org.springframework=info

logging.level.com.example=debug

logging.level.org.hibernate.SQL=trace

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

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

**Country.java:**

**package** com.example.spring\_data\_jpa\_demo.model;

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

@Entity

@Table(name = "country")

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

@Id

@Column(name = "co\_code")

**private** String code;

@Column(name = "co\_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; }

@Override

**public** String toString() {

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

}

}

**CountryRepository.java:**

**package** com.example.spring\_data\_jpa\_demo.repository;

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

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

**import** com.example.spring\_data\_jpa\_demo.model.Country;

@Repository

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

}

**CountryService.java:**

**package** com.example.spring\_data\_jpa\_demo.service;

**import** java.util.List;

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

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

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

**import** com.example.spring\_data\_jpa\_demo.model.Country;

**import** com.example.spring\_data\_jpa\_demo.repository.CountryRepository;

@Service

**public** **class** CountryService {

@Autowired

**private** CountryRepository countryRepository;

@Transactional

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

**return** countryRepository.findAll();

}

}

**SpringDataJpaDemoApplication.java:**

**package** com.example.spring\_data\_jpa\_demo;

**import** java.util.List;

**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** com.example.spring\_data\_jpa\_demo.model.Country;

**import** com.example.spring\_data\_jpa\_demo.service.CountryService;

@SpringBootApplication

**public** **class** SpringDataJpaDemoApplication {

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

**private** **static** CountryService *countryService*;

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

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

*countryService* = context.getBean(CountryService.**class**);

***LOGGER***.info("Inside main");

*testGetAllCountries*(); // make sure this method is declared

System.***out***.println("----- Country List Start -----");

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

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

System.***out***.println("Code: " + c.getCode() + ", Name: " + c.getName());

}

System.***out***.println("----- Country List End -----");

}

**private** **static** **void** testGetAllCountries() {

***LOGGER***.info("Start");

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

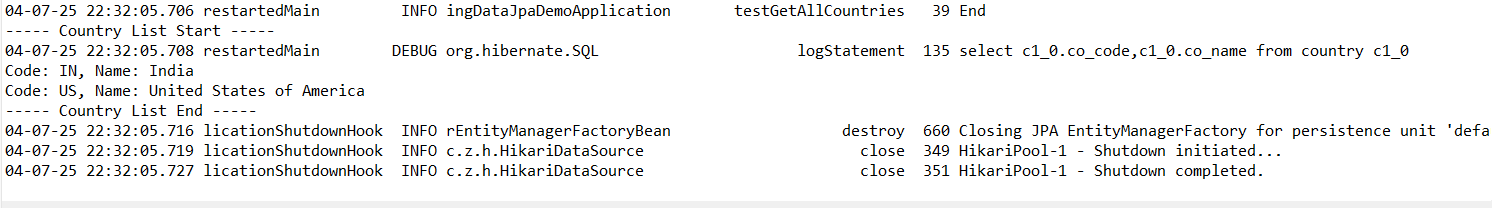
***LOGGER***.debug("countries={}", countries);

***LOGGER***.info("End");

}

}

**OUTPUT;**

****

**2.Difference Between JPA, Hibernate, and Spring Data JPA**

Java Persistence API (JPA)

* **Standard specification** (JSR 338) for object-relational mapping (ORM) in Java
* Defines interfaces and annotations for persisting Java objects to relational databases
* **Not an implementation** - just provides the API specification
* Requires an implementation provider (like Hibernate, EclipseLink, etc.)

Hibernate

* **Most popular implementation** of the JPA specification
* Provides additional features beyond JPA specification
* Handles all the low-level database interactions
* Manages database connections, transactions, and SQL generation
* Requires more boilerplate code compared to Spring Data JPA

Spring Data JPA

* **Abstraction layer** on top of JPA providers (like Hibernate)
* Reduces boilerplate code through repository interfaces
* Provides powerful features like:
  + Derived query methods (automatically generates queries from method names)
  + Pagination and sorting support
  + Custom repository implementations
* Still relies on a JPA provider (typically Hibernate) under the hood

**Hibernate Example:**

public Integer addEmployee(Employee employee) {

Session session = factory.openSession();

Transaction tx = null;

Integer employeeID = null;

try {

tx = session.beginTransaction();

employeeID = (Integer) session.save(employee);

tx.commit();

} catch (HibernateException e) {

if (tx != null) tx.rollback();

e.printStackTrace();

} finally {

session.close();

}

return employeeID;

}

**Spring Data JPA Example (More Concise)**

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

// No implementation needed for basic CRUD operations

}

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee);

}

}

Additional hands-on:

**Implement services for managing Country**

**Pom.xml**

<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>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>

**Application.properties:**

spring.application.name=countryservice

# Database config

spring.datasource.url=jdbc:mysql://localhost:3306/spring\_data\_jpa\_demo

spring.datasource.username=root

spring.datasource.password=02July@2005

# JPA config

spring.jpa.hibernate.ddl-auto=validate

spring.jpa.show-sql=true

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

server.port=8082

**country.java:**

**package** com.example.country;

**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;

// Getters and Setters

**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;

}

}

**CountryController.java:**

**package** com.example.country;

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

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

**import** java.util.List;

@RestController

@RequestMapping("/countries")

**public** **class** CountryController {

@Autowired

**private** CountryService countryService;

@GetMapping("/{code}")

**public** Country getCountryByCode(@PathVariable String code) {

**return** countryService.getCountryByCode(code);

}

@GetMapping("/search")

**public** List<Country> getCountriesByPartialName(@RequestParam String name) {

**return** countryService.getCountriesByPartialName(name);

}

@PostMapping

**public** Country addCountry(@RequestBody Country country) {

**return** countryService.addCountry(country);

}

@PutMapping("/{code}")

**public** Country updateCountry(@PathVariable String code, @RequestBody Country updatedCountry) {

**return** countryService.updateCountry(code, updatedCountry);

}

@DeleteMapping("/{code}")

**public** **void** deleteCountry(@PathVariable String code) {

countryService.deleteCountry(code);

}

}

**CountryRepository.java:**

**package** com.example.country;

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

**import** java.util.List;

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

// Find countries by partial name (case-insensitive)

List<Country> findByCoNameContainingIgnoreCase(String partialName);

}

**CountryService.java:**

**package** com.example.country;

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

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

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

**import** java.util.List;

**import** java.util.Optional;

@Service

**public** **class** CountryService {

@Autowired

**private** CountryRepository countryRepository;

**public** Country getCountryByCode(String code) {

**return** countryRepository.findById(code).orElse(**null**);

}

**public** List<Country> getCountriesByPartialName(String partialName) {

**return** countryRepository.findByCoNameContainingIgnoreCase(partialName);

}

@Transactional

**public** Country addCountry(Country country) {

**return** countryRepository.save(country);

}

@Transactional

**public** Country updateCountry(String code, Country updatedCountry) {

Optional<Country> optionalCountry = countryRepository.findById(code);

**if** (optionalCountry.isPresent()) {

Country country = optionalCountry.get();

country.setCoName(updatedCountry.getCoName());

**return** countryRepository.save(country);

}

**return** **null**;

}

@Transactional

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

countryRepository.deleteById(code);

}

}

**CountryserviceApplication.java:**

**package** com.example.country;

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

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

@SpringBootApplication

**public** **class** CountryserviceApplication {

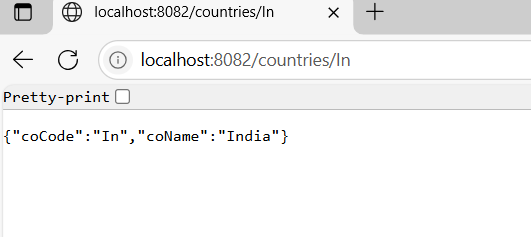
**public** **static** **void** main(String[] args) {

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

}

}

**OUTPUT:**

****