**Mandatory HandsOn - 01:**

Spring Data JPA - Quick Example

* MySQL database [in workbench]

*CREATE DATABASE countrydb;*

*USE countrydb;*

*CREATE TABLE country (*

*co\_code VARCHAR(2) PRIMARY KEY,*

*co\_name VARCHAR(50)*

*);*

*INSERT INTO country VALUES ('IN', 'India'), ('US', 'United States');*

**applicationProperties**

*spring.application.name=countrydemo*

*# MySQL DB Connection*

*spring.datasource.url=jdbc:mysql://localhost:3306/countrydb*

*spring.datasource.username=root*

*spring.datasource.password=B@blu7612*

*# JPA*

*spring.jpa.hibernate.ddl-auto=none*

*spring.jpa.show-sql=true*

*spring.jpa.properties.hibernate.format\_sql=true*

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

**Country.java**

*package com.example.countrydemo.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")*

*private String code;*

*@Column(name = "co\_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.example.countrydemo.repository;*

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

*import org.springframework.stereotype.Repository;*

*import com.example.countrydemo.model.Country;*

*@Repository*

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

*}*

**CountrydemoApplication.java**

*package com.example.countrydemo;*

*import java.util.List;*

*import org.springframework.boot.SpringApplication;*

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

*import org.springframework.context.ApplicationContext;*

*import com.example.countrydemo.model.Country;*

*import com.example.countrydemo.repository.CountryRepository;*

*@SpringBootApplication*

*public class CountrydemoApplication {*

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

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

*CountryRepository repo = context.getBean(CountryRepository.class);*

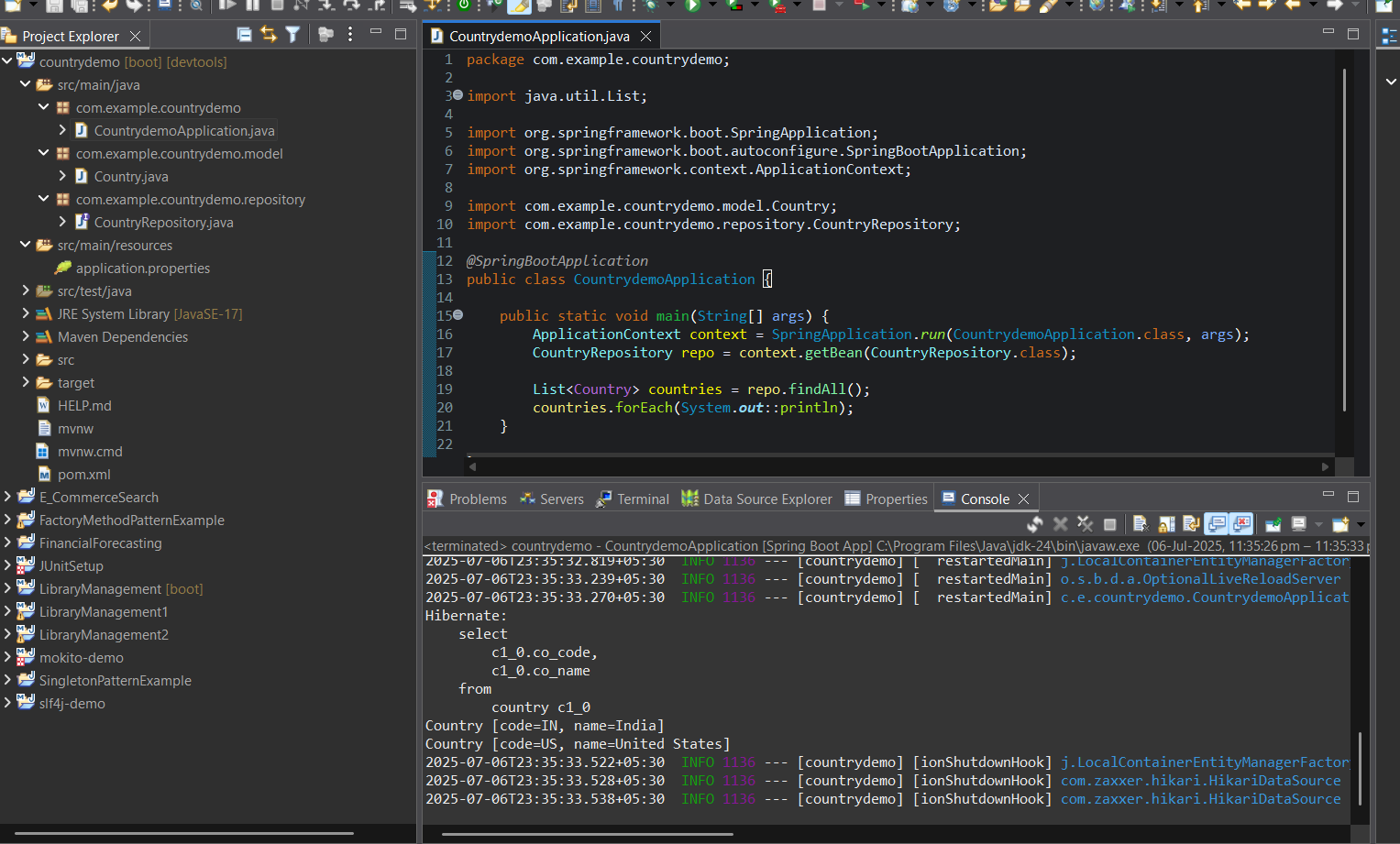
*List<Country> countries = repo.findAll();*

*countries.forEach(System.****out****::println);*

*}*

*}*

**OUTPUT:**



**Mandatory HandsOn-2:**

**Book.java**

*package com.example.JPAdemos.model;*

*import jakarta.persistence.\*;*

*@Entity*

*public class Book {*

*@Id*

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

*private Long id;*

*private String title;*

*private String author;*

*public Book() {}*

*public Book(String title, String author) {*

*this.title = title;*

*this.author = author;*

*}*

*public Long getId() {*

*return id;*

*}*

*public void setId(Long id) {*

*this.id = id;*

*}*

*public String getTitle() {*

*return title;*

*}*

*public void setTitle(String title) {*

*this.title = title;*

*}*

*public String getAuthor() {*

*return author;*

*}*

*public void setAuthor(String author) {*

*this.author = author;*

*}*

*}*

**JPAService.java**

*package com.example.JPAdemos.service;*

*import org.springframework.stereotype.Service;*

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

*import com.example.JPAdemos.model.Book;*

*import jakarta.persistence.EntityManager;*

*import jakarta.persistence.PersistenceContext;*

*@Service*

*public class JPAService {*

*@PersistenceContext*

*private EntityManager entityManager;*

*@Transactional*

*public void saveBookWithJPA(String title, String author) {*

*Book book = new Book(title, author);*

*entityManager.persist(book); // Pure JPA*

*}*

*}*

**HibernateService.java**

*package com.example.JPAdemos.service;*

*import org.hibernate.Session;*

*import org.springframework.stereotype.Service;*

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

*import com.example.JPAdemos.model.Book;*

*import jakarta.persistence.EntityManager;*

*@Service*

*public class HibernateService {*

*private final EntityManager entityManager;*

*public HibernateService(EntityManager entityManager) {*

*this.entityManager = entityManager;*

*}*

*@Transactional*

*public void saveBookWithHibernate(String title, String author) {*

*Session session = entityManager.unwrap(Session.class);*

*Book book = new Book(title, author);*

*session.save(book); // Hibernate-specific method*

*}*

*}*

**SpringDataService.java**

*package com.example.JPAdemos.service;*

*import org.springframework.stereotype.Service;*

*import com.example.JPAdemos.model.Book;*

*import com.example.JPAdemos.repository.BookRepository;*

*@Service*

*public class SpringDataService {*

*private final BookRepository bookRepository;*

*public SpringDataService(BookRepository bookRepository) {*

*this.bookRepository = bookRepository;*

*}*

*public void saveBook(String title, String author) {*

*Book book = new Book(title, author);*

*bookRepository.save(book); // Abstracted JPA operation*

*}*

*}*

**BookController.java**

*package com.example.JPAdemos.controller;*

*import com.example.JPAdemos.service.\*;*

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

*@RestController*

*@RequestMapping("/books")*

*public class BookController {*

*private final JPAService jpaService;*

*private final HibernateService hibernateService;*

*private final SpringDataService springDataService;*

*public BookController(JPAService jpaService, HibernateService hibernateService, SpringDataService springDataService) {*

*this.jpaService = jpaService;*

*this.hibernateService = hibernateService;*

*this.springDataService = springDataService;*

*}*

*@PostMapping("/jpa")*

*public void saveWithJPA(@RequestParam String title, @RequestParam String author) {*

*jpaService.saveBookWithJPA(title, author);*

*}*

*@PostMapping("/hibernate")*

*public void saveWithHibernate(@RequestParam String title, @RequestParam String author) {*

*hibernateService.saveBookWithHibernate(title, author);*

*}*

*@PostMapping("/springdata")*

*public void saveWithSpringData(@RequestParam String title, @RequestParam String author) {*

*springDataService.saveBook(title, author);*

*}*

*}*

**BookRepository.java**

*package com.example.JPAdemos.repository;*

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

*import com.example.JPAdemos.model.Book;*

*public interface BookRepository extends JpaRepository<Book, Long> {*

*}*

**application.properties**

*spring.application.name=JPAdemos*

*spring.datasource.url=jdbc:h2:mem:jpademodb*

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

*spring.datasource.username=sa*

*spring.datasource.password=*

*spring.jpa.hibernate.ddl-auto=update*

*spring.jpa.show-sql=true*

*spring.h2.console.enabled=true*

**JpAdemosApplication.java**

*package com.example.JPAdemos;*

*import org.springframework.boot.SpringApplication;*

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

*@SpringBootApplication*

*public class JpAdemosApplication {*

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

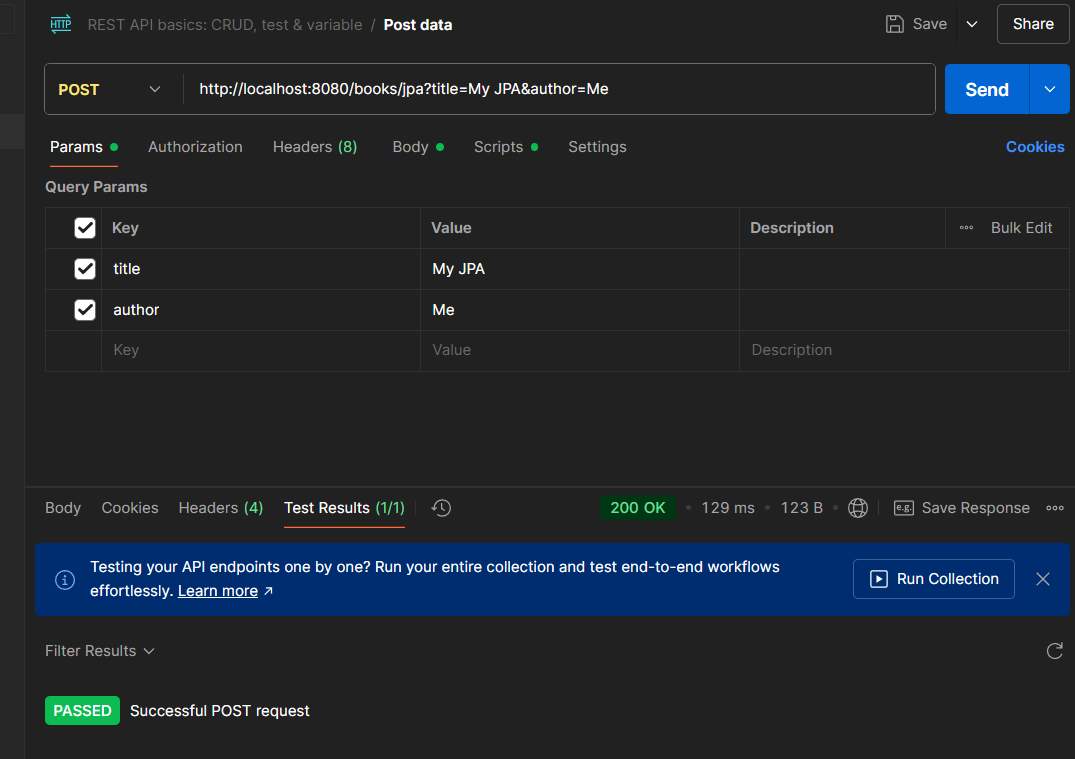
*SpringApplication.run(JpAdemosApplication.class, args);*

*}*

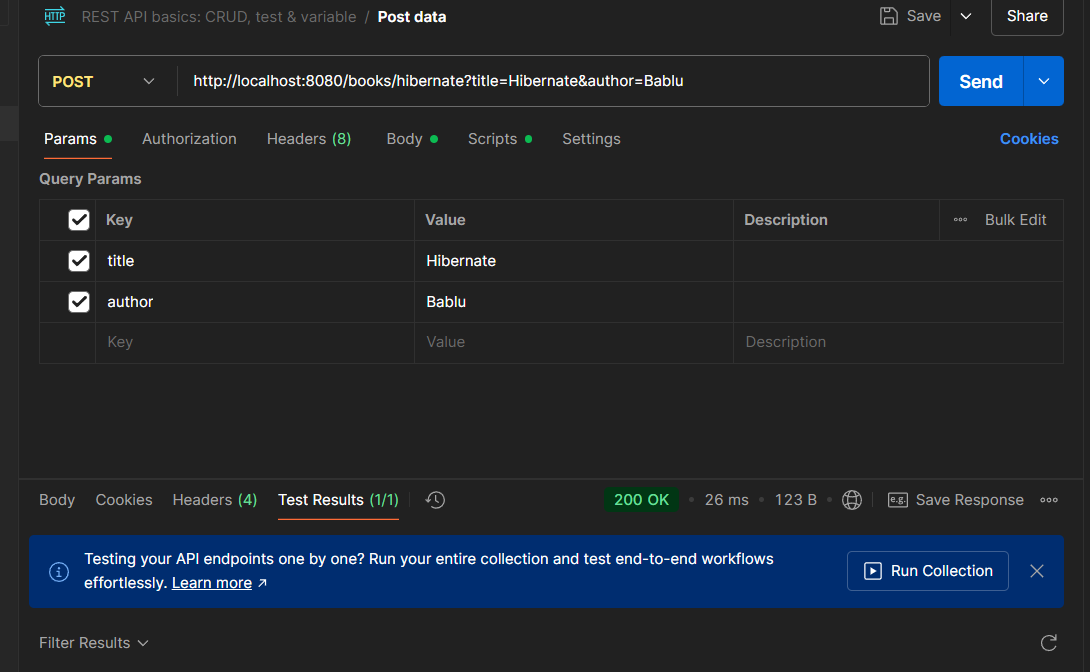
*}*

**Testing using Postman**

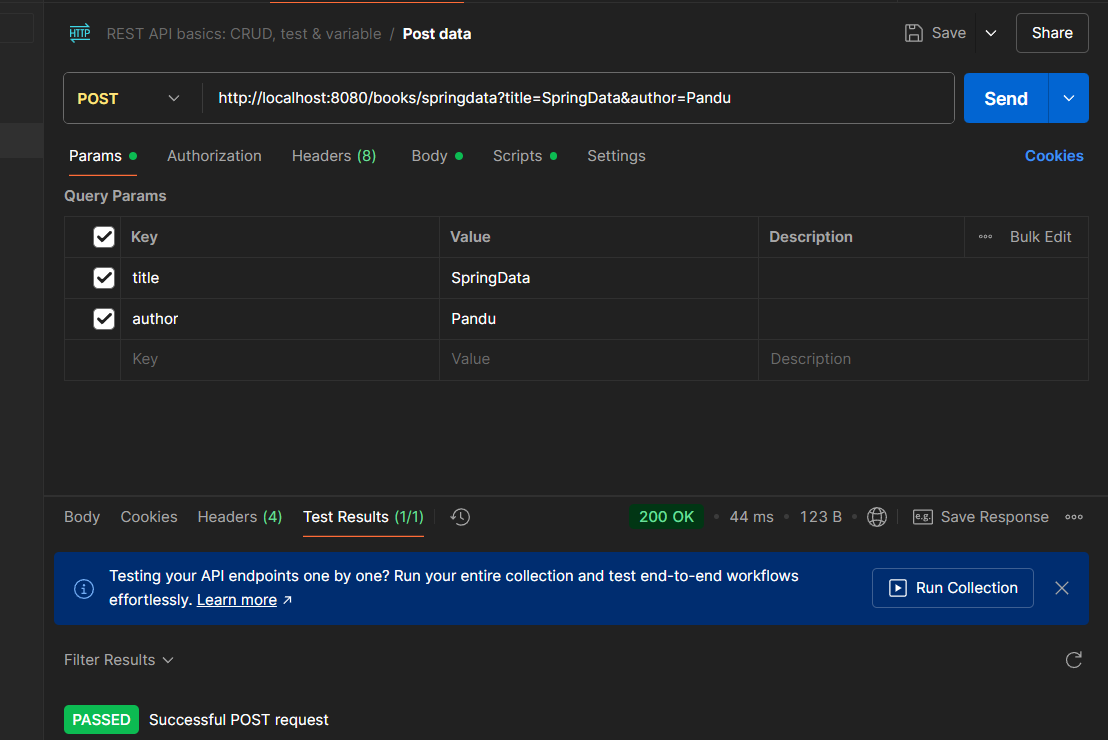
1. **Save with JPA**

****

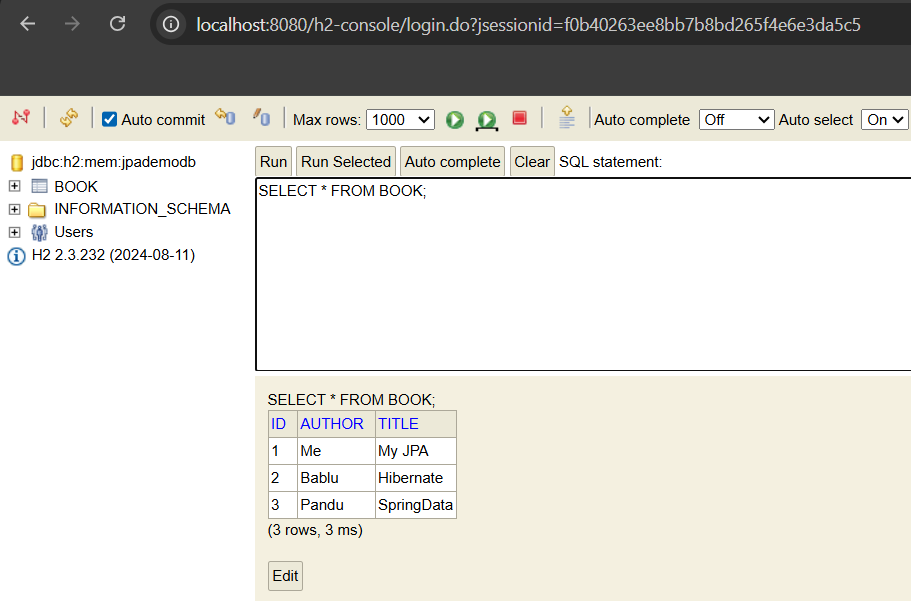
1. **Save with Hibernate**

****

1. **Save using Spring Data JPA**

****

**Verification in H2 Console.**

****