**Assignment Documentation: Spring Data JPA - Quick Example**

**Objective:**

Demonstrate a working Spring Boot + Spring Data JPA application with a simple Book entity and full CRUD operations using an in-memory H2 database.

**Project Structure:**

pom.xml

src/

└── main/

├── java/

│ └── com.example.demo/

│ ├── DemoApplication.java

│ ├── model/

│ │ └── Book.java

│ ├── repository/

│ │ └── BookRepository.java

│ └── controller/

│ └── BookController.java

└── resources/

├── application.properties

**Step 1: Entity - Book.java**

package com.example.demo.model;  
  
import javax.persistence.\*;  
  
@Entity  
@Table(name = "book")  
public class Book {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Long id;  
  
 private String title;  
 private String author;  
  
 // Constructors  
 public Book() {  
 }  
  
 public Book(String title, String author) {  
 this.title = title;  
 this.author = author;  
 }  
  
 // Getters and Setters  
 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; }  
}

**Step 2: Repository - BookRepository.java**

package com.example.demo.repository;  
  
import com.example.demo.model.Book;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface BookRepository extends JpaRepository<Book, Long> {  
}

**Step 3: Controller - BookController.java**

package com.example.demo.controller;  
  
import com.example.demo.model.Book;  
import com.example.demo.repository.BookRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
import java.util.Optional;  
  
@RestController  
@RequestMapping("/books")  
public class BookController {  
  
 @Autowired  
 private BookRepository bookRepository;  
  
 @PostMapping  
 public Book addBook(@RequestBody Book book) {  
 return bookRepository.save(book);  
 }  
  
 @GetMapping  
 public List<Book> getAllBooks() {  
 return bookRepository.findAll();  
 }  
  
 @GetMapping("/{id}")  
 public Optional<Book> getBookById(@PathVariable Long id) {  
 return bookRepository.findById(id);  
 }  
  
 @DeleteMapping("/{id}")  
 public void deleteBook(@PathVariable Long id) {  
 bookRepository.deleteById(id);  
 }  
}

**Step 4: application.properties**

# Enable H2 console  
spring.h2.console.enabled=true  
spring.h2.console.path=/h2-console  
  
# Use in-memory database  
spring.datasource.url=jdbc:h2:mem:testdb  
spring.datasource.driverClassName=org.h2.Driver  
spring.datasource.username=sa  
spring.datasource.password=  
  
# JPA and Hibernate settings  
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect  
spring.jpa.hibernate.ddl-auto=update  
spring.jpa.show-sql=true

**Step 5: DemoApplication.java**

package com.example.demo;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class DemoApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(DemoApplication.class, args);  
 }  
}

