**Hands on: Demonstrate implementation of O/R Mapping**

* + @ManyToOne, @JoinColumn, @OneToMany, FetchType.EAGER, FetchType.LAZY, @ManyToMany, @JoinTable, mappedBy
    - Relationships reference - https://www.baeldung.com/spring-data-rest-relationships

**Solution code:**

**Application.properties:**

# H2 in-memory DB  
spring.datasource.url=jdbc:h2:mem:testdb  
spring.datasource.driver-class-name=org.h2.Driver  
spring.datasource.username=sa  
spring.datasource.password=  
  
# Hibernate schema generation  
spring.jpa.hibernate.ddl-auto=create  
  
# Run schema and data SQL  
spring.sql.init.mode=always  
spring.jpa.defer-datasource-initialization=true  
  
# Enable H2 Console  
spring.h2.console.enabled=true  
spring.h2.console.path=/h2-console

**File name Student.java:**

package com.example.world\_learn.model;  
  
import com.fasterxml.jackson.annotation.JsonBackReference;  
import com.fasterxml.jackson.annotation.JsonManagedReference;  
import jakarta.persistence.\*;  
import java.util.List;  
@Entity  
public class Student {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Long id;  
  
 private String name;  
  
 @ManyToOne(fetch = FetchType.*EAGER*)  
 @JoinColumn(name = "country\_code")  
 @JsonBackReference  
 private Country country;  
  
 @ManyToMany  
 @JoinTable(  
 name = "student\_book",  
 joinColumns = @JoinColumn(name = "student\_id"),  
 inverseJoinColumns = @JoinColumn(name = "book\_id")  
 )  
 @JsonBackReference  
 private List<Book> books;  
  
 @ManyToMany  
 @JoinTable(  
 name = "student\_course",  
 joinColumns = @JoinColumn(name = "student\_id"),  
 inverseJoinColumns = @JoinColumn(name = "course\_id")  
 )  
 private List<Course> courses;  
  
 // Getters and setters  
 public Long getId() { return id; }  
  
 public String getName() { return name; }  
 public void setName(String name) { this.name = name; }  
  
 public Country getCountry() { return country; }  
 public void setCountry(Country country) { this.country = country; }  
  
 public List<Book> getBooks() { return books; }  
 public void setBooks(List<Book> books) { this.books = books; }  
  
 public List<Course> getCourses() { return courses; }  
 public void setCourses(List<Course> courses) { this.courses = courses; }  
}

**file name is Course.java:**

package com.example.world\_learn.model;  
  
import jakarta.persistence.\*;  
import java.util.Set;  
  
@Entity  
public class Course {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Long id;  
  
 private String name;  
  
 @ManyToMany(mappedBy = "courses")  
 private Set<Student> students;  
  
 // Getter and Setter for 'id'  
 public Long getId() {  
 return id;  
 }  
  
 public void setId(Long id) {  
 this.id = id;  
 }  
  
 // Getter and Setter for 'name'  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 // Getter and Setter for 'students'  
 public Set<Student> getStudents() {  
 return students;  
 }  
  
 public void setStudents(Set<Student> students) {  
 this.students = students;  
 }  
}

**file name is Book.java:**

package com.example.world\_learn.model;  
  
import com.fasterxml.jackson.annotation.JsonBackReference;  
import com.fasterxml.jackson.annotation.JsonManagedReference;  
import jakarta.persistence.\*;  
import java.util.List;  
  
@Entity  
public class Book {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Long id;  
  
 private String title;  
  
 @ManyToMany(mappedBy = "books")  
 @JsonBackReference  
 private List<Student> students;  
  
 // Getters and Setters  
 public Long getId() { return id; }  
  
 public String getTitle() { return title; }  
 public void setTitle(String title) { this.title = title; }  
  
 public List<Student> getStudents() { return students; }  
 public void setStudents(List<Student> students) { this.students = students; }  
}

**file name StudentRepository.java:**

package com.example.world\_learn.repository;  
  
import com.example.world\_learn.model.Student;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface StudentRepository extends JpaRepository<Student, Long> {}

**file name CourseRepository.java:**

package com.example.world\_learn.repository;  
  
import com.example.world\_learn.model.Course;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface CourseRepository extends JpaRepository<Course, Long> {  
}

**file name BookRepository.java:**

package com.example.world\_learn.repository;  
  
import com.example.world\_learn.model.Book;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface BookRepository extends JpaRepository<Book, Long> {}

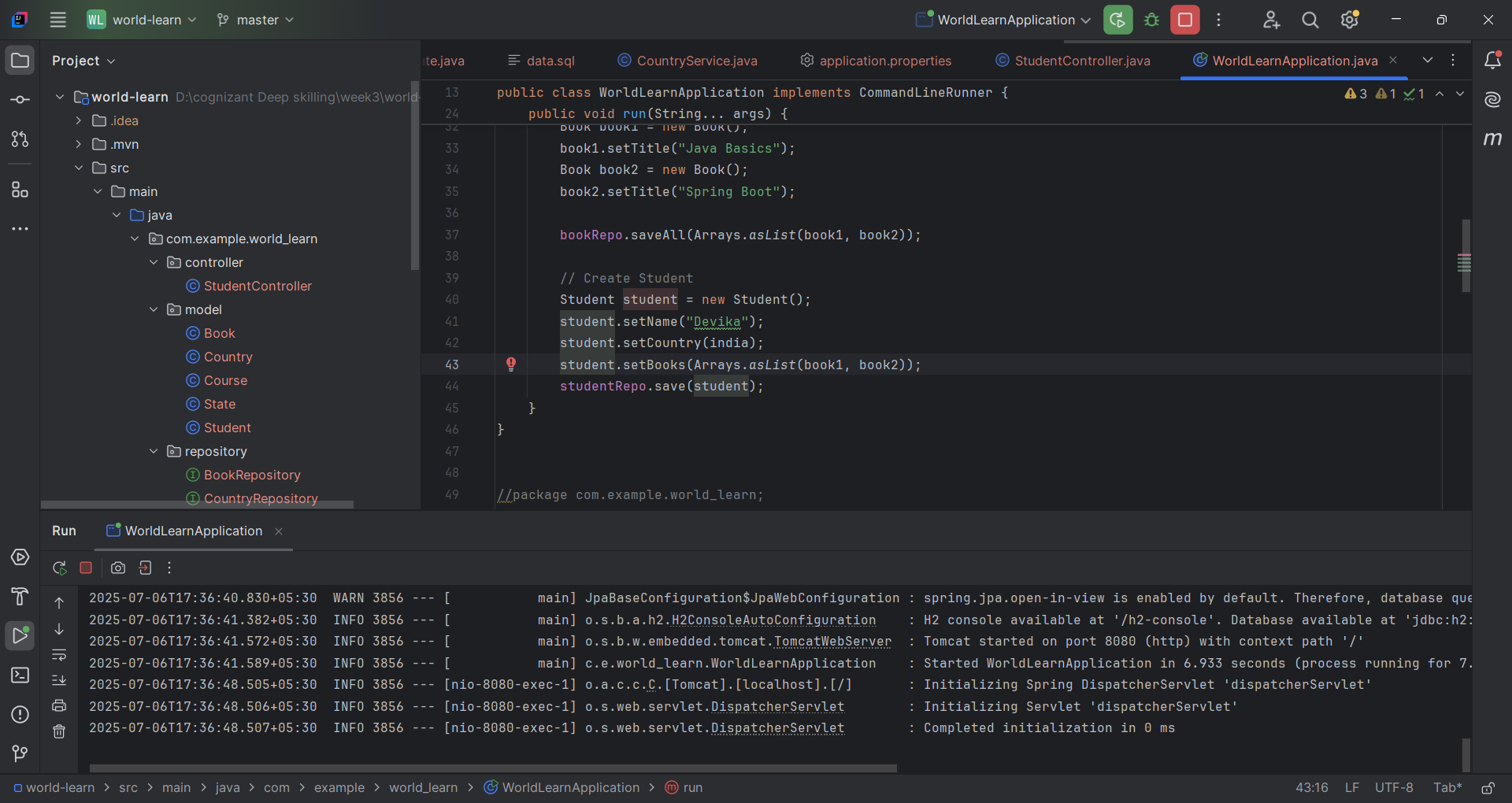
**file name StudentController.java:**

package com.example.world\_learn.controller;  
  
import com.example.world\_learn.model.Student;  
import com.example.world\_learn.repository.StudentRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
@RequestMapping("/students")  
public class StudentController {  
  
 @Autowired  
 private StudentRepository studentRepository;  
  
 @GetMapping  
 public List<Student> getAllStudents() {  
 return studentRepository.findAll();  
 }  
  
 @PostMapping  
 public Student createStudent(@RequestBody Student student) {  
 return studentRepository.save(student);  
 }  
  
 @GetMapping("/{id}")  
 public Student getStudentById(@PathVariable Long id) {  
 return studentRepository.findById(id).orElse(null);  
 }  
  
 @PutMapping("/{id}")  
 public Student updateStudent(@PathVariable Long id, @RequestBody Student updatedStudent) {  
 Student student = studentRepository.findById(id).orElse(null);  
 if (student != null) {  
 student.setName(updatedStudent.getName());  
 student.setCountry(updatedStudent.getCountry());  
 student.setBooks(updatedStudent.getBooks());  
 return studentRepository.save(student);  
 }  
 return null;  
 }  
  
 @DeleteMapping("/{id}")  
 public void deleteStudent(@PathVariable Long id) {  
 studentRepository.deleteById(id);  
 }  
}

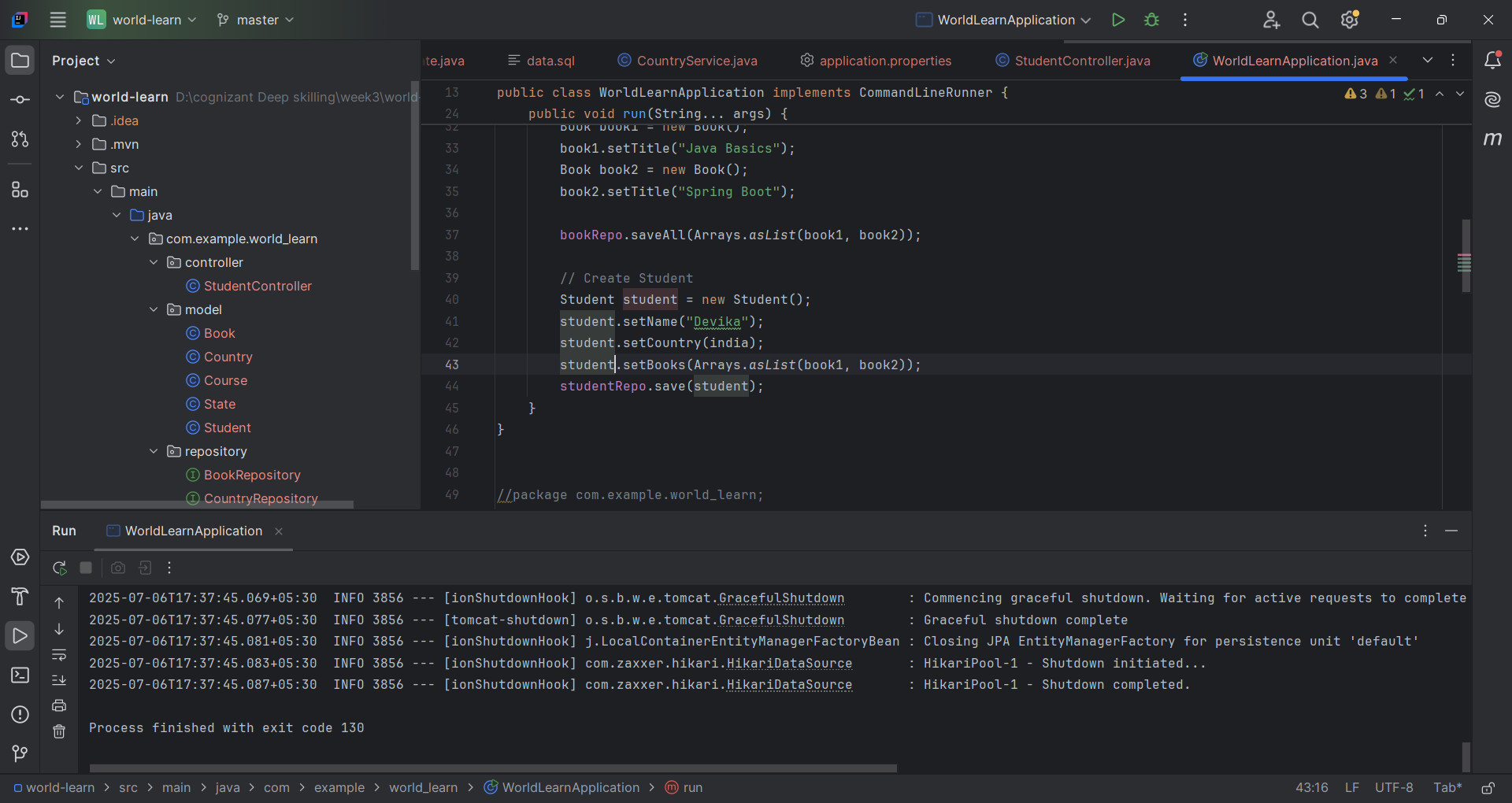
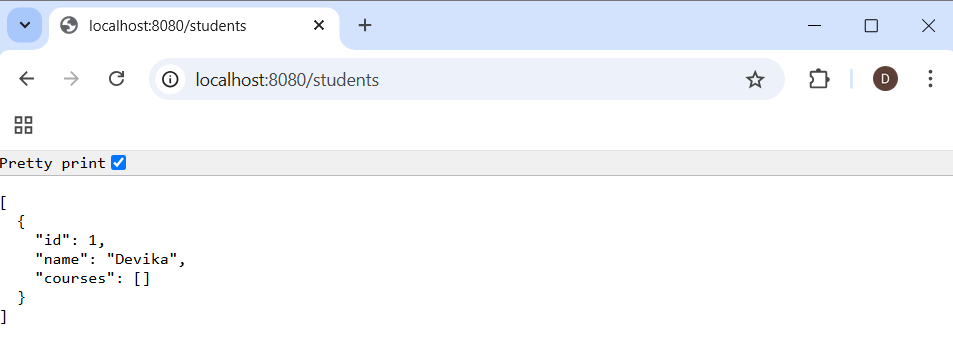
**Testing file name WorldLearnApplication:**

package com.example.world\_learn;  
  
import com.example.world\_learn.model.\*;  
import com.example.world\_learn.repository.\*;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.boot.CommandLineRunner;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
import java.util.Arrays;  
  
@SpringBootApplication  
public class WorldLearnApplication implements CommandLineRunner {  
  
 @Autowired private CountryRepository countryRepo;  
 @Autowired private StudentRepository studentRepo;  
 @Autowired private BookRepository bookRepo;  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(WorldLearnApplication.class, args);  
 }  
  
 @Override  
 public void run(String... args) {  
 // Create Country  
 Country india = new Country();  
 india.setCode("IN");  
 india.setName("India");  
 countryRepo.save(india);  
  
 // Create Books  
 Book book1 = new Book();  
 book1.setTitle("Java Basics");  
 Book book2 = new Book();  
 book2.setTitle("Spring Boot");  
  
 bookRepo.saveAll(Arrays.*asList*(book1, book2));  
  
 // Create Student  
 Student student = new Student();  
 student.setName("Devika");  
 student.setCountry(india);  
 student.setBooks(Arrays.*asList*(book1, book2));  
 studentRepo.save(student);  
 }  
}

**OUTPUT:**

****

Since there is no course data, output will be as follows:

****