Spring Boot-based Web Application for a Simplified Online School System

Type: Projects

Skill: Java

Database MappingEntity RelationshipsJPARESTful EndpointsSpring Boot

Medium

Develop a Spring Boot-based web application for a simplified Online School System that includes a many-to-one relationship between students and courses.

Utilize Spring Boot and JPA to manage these entities and establish a seamless many-to-one association, where multiple students can be enrolled in a single course, but each student can only be enrolled in one course at a time.

**Functional Requirements:**

1.Student Entity: Create a Student class with attributes such as

* studentId (type Long)
* firstName (type String)
* lastName (type String)
* email (type String)
* course (type Course)
* Use Spring JPA annotations to map the Student class to a corresponding table in the database.

2.Course Entity: Create a Course class with attributes such as

* courseId (type Long)
* courseName (type String)
* courseDescription (type String)
* credits (type int)
* students (type List<Student>)
* Use Spring JPA annotations to map the Course class to a corresponding table in the database.
* Implement Getters and Setters for all attributes in both the Student and Course classes as per standard java conventions.

3.Many-to-One Relationship:

* Establish a many-to-one relationship between the Student and Course entities. Each student is enrolled in exactly one course, while a single course can have multiple students enrolled. The Student entity should have a reference to the Course it is enrolled in.

4.Course Creation:

* Implement a RESTful endpoint /courses with a @PostMapping annotation to allow users to add new courses to the system. The creation request should include the course's details (courseName, courseDescription, credits).
* The Api should accept CourseRequest object as request body and return CourseResponse object as response body.

5.Student Creation:

* Implement a RESTful endpoint /students/{courseId} with a @PostMapping annotation to allow users to add new students to the system. The creation request should include the student's details (first name, last name, email) and the corresponding course ID.
* The Api should accept StudentRequest object, courseId as path variable and return Student object as response body with the course details.

6.Student Retrieval:

* Implement a RESTful endpoint /students/{studentId} with a @GetMapping annotation to retrieve details of a specific student, including the course they are enrolled in.

**Input:**

1.Course Creation: Input fields for the course name, description, and credits.

2.Student Creation: Input fields for the student's first name, last name, email, and course ID.

3.Student Retrieval: Input fields to retrieve a student's information based on their ID.

**Output:**

1.Course Creation: Display the created course with a success message.

2.Student Creation: Display the created student with a success message, including the course they are enrolled in.

3.Student Retrieval: Display the details of a specific student, including their associated course.

**Note:**

Your task is to implement the missing code, missing annotations, and missing methods in the provided Spring Boot project to fulfill the requirements mentioned above.