Learning Platform Microservice Documentation

Introduction

The Learning Platform Microservice is responsible for managing students, courses, and enrollments within the learning platform system. This documentation provides a guide on how to use and interact with the microservice.

we only have two microservices that communicate with only REST APIs .

first microservice:(microserviceone)

uses EJBs , wildfly application server , jakarta ee and in memory database (h2)

second microservice:(microservicetwo)

uses spring boot and MySQL database .

--------------------------------------

how to create the database :

--------------------------------------

CREATE TABLE Course (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

duration INT NOT NULL,

category VARCHAR(255) NOT NULL,

rating DOUBLE NOT NULL,

capacity INT NOT NULL,

numberOfEnrolledStudents INT NOT NULL,

status VARCHAR(255) NOT NULL

);

CREATE TABLE Enrollment (

id INT AUTO\_INCREMENT PRIMARY KEY,

student\_id BIGINT NOT NULL,

course\_id BIGINT NOT NULL,

enrollment\_status VARCHAR(50),

FOREIGN KEY (student\_id) REFERENCES Student(id),

INDEX fk\_student\_idx (student\_id),

INDEX fk\_course\_idx (course\_id)

);

CREATE TABLE Student (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

password VARCHAR(100) NOT NULL,

affiliation VARCHAR(100),

bio TEXT

);

--------------------------------------

the process

--------------------------------------

first of all you need to download the wildfly application server , also make

sure of the maven dependencies .

secondly Deploying the Microservices

Deploy your EJB-based microservice (microserviceone) to the WildFly application server.

Ensure that the microservice is running correctly and is accessible via its assigned endpoint URLs.

Build and package your Spring Boot application (microservicetwo) into a JAR file.

Deploy the JAR file to your preferred environment, such as a cloud platform or a local server.

Start the microservice and verify that it's successfully connected to the MySQL database.

Endpoints:

Admin Endpoints

api/admins/login (POST)

api/admins/users (GET)

api/admins/viewAll/courses (GET) : show courses details

admins/publishCheck/courses/2/status (put) : change status

Students Endpoints

/api/students/register (POST): Register a new student.

/api/students/login (POST): Login an existing student.

/api/students/{studentId} (GET): Retrieve information about a specific student.

/api/students/{studentId}/enrollments (GET): Retrieve enrollments for a specific student.

/api/students/{studentId}/courses (GET): Retrieve courses for a specific student.

/api/students/{studentId}/courses/{courseId}/enroll (POST): Enroll a student in a course.

/api/students/{studentId}/courses/{courseId}/cancel-enrollment (DELETE): Cancel enrollment in a course.

Courses Endpoints

/api/courses (GET): Retrieve all courses.

/api/courses/{courseId} (GET): Retrieve information about a specific course.

/api/courses/search (GET): Search for courses by name, category, or other criteria.

Instructor Endpoints

Instructor Registration and Login

POST /api/instructors/register: Register a new instructor account.

POST /api/instructors/login: Log in an existing instructor account.

POST /api/instructors/logout: Log out the currently authenticated instructor.

Course Information Endpoints

GET /api/students/courses/{courseId}: Retrieve details of a specific course.

GET /api/students/courses/search: Search for courses by name, category, etc.

examples on some endpoints :

Student Registration (POST /api/students/register)

Request Body:

{

"name": "John Doe",

"email": "john.doe@example.com",

"password": "password123",

"affiliation": "University XYZ",

"bio": "Computer Science student"

}

Response:

{

"id": 1,

"name": "John Doe",

"email": "john.doe@example.com",

"affiliation": "University XYZ",

"bio": "Computer Science student"

}

notes :

the first microservice (microserviceone) uses in memory database ,

the second microservice (microservictwo) uses centralized MySQL database ,