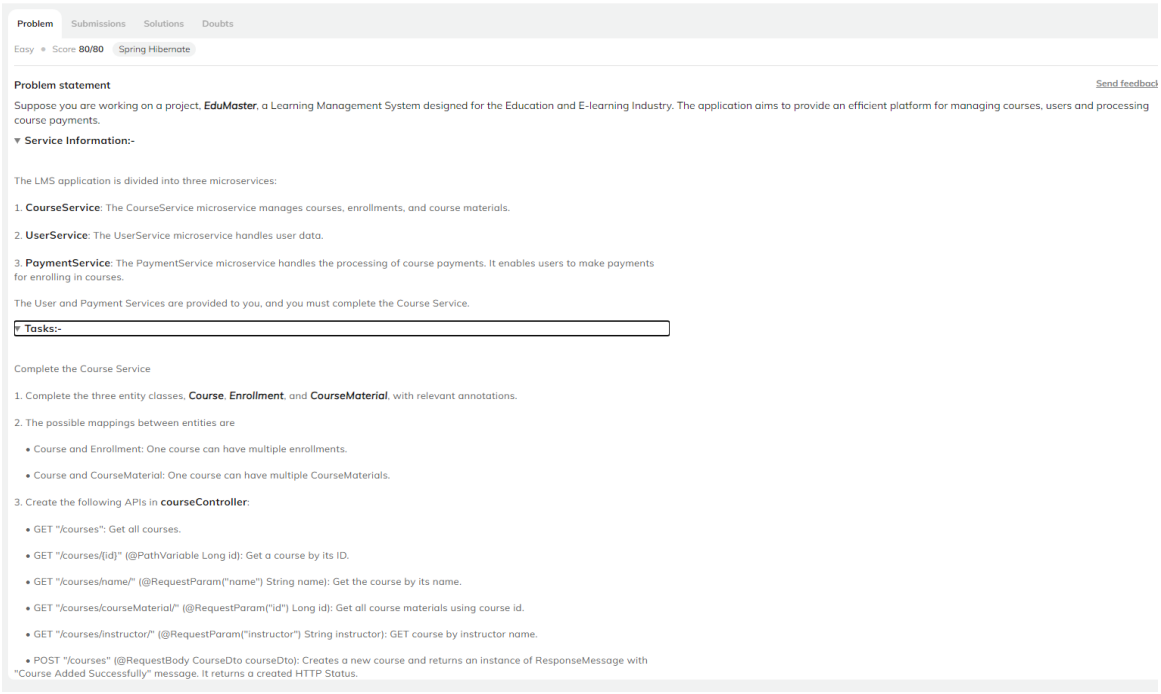
Edu Master Application

Problem Statement



Problem Submissions Solutions Doubts

Easy • Score 80/80 Spring Hibernate

Problem statement Send feedback

Suppose you are working on a project, ***EduMaster***, a Learning Management System designed for the Education and E-learning Industry. The application aims to provide an efficient platform for managing courses, users and processing course payments.

▼ **Service Information:-**

The LMS application is divided into three microservices:

1. **CourseService**: The CourseService microservice manages courses, enrollments, and course materials.

2. **UserService**: The UserService microservice handles user data.

3. **PaymentService**: The PaymentService microservice handles the processing of course payments. It enables users to make payments for enrolling in courses.

The User and Payment Services are provided to you, and you must complete the Course Service.

▼ **Tasks:-**

Complete the Course Service

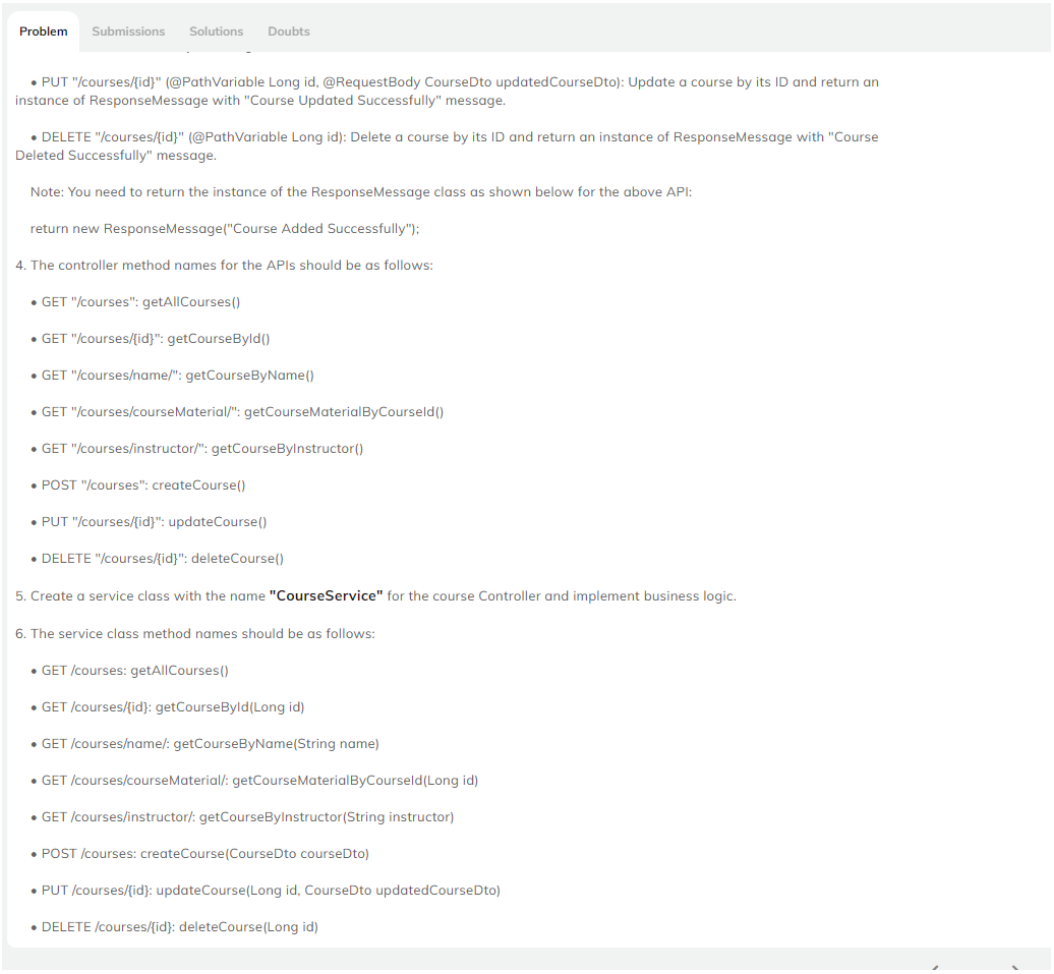
1. Complete the three entity classes, ***Course***, ***Enrollment***, and ***CourseMaterial***, with relevant annotations.

2. The possible mappings between entities are

- Course and Enrollment: One course can have multiple enrollments.
- Course and CourseMaterial: One course can have multiple CourseMaterials.

3. Create the following APIs in **courseController**:

- GET "/courses": Get all courses.
- GET "/courses/{id}" (@PathVariable Long id): Get a course by its ID.
- GET "/courses/name/" (@RequestParam("name") String name): Get the course by its name.
- GET "/courses/courseMaterial/" (@RequestParam("id") Long id): Get all course materials using course id.
- GET "/courses/instructor/" (@RequestParam("instructor") String instructor): GET course by instructor name.
- POST "/courses" (@RequestBody CourseDto courseDto): Creates a new course and returns an instance of ResponseMessage with "Course Added Successfully" message. It returns a created HTTP Status.



Problem Submissions Solutions Doubts

- PUT "/courses/{id}" (@PathVariable Long id, @RequestBody CourseDto updatedCourseDto): Update a course by its ID and return an instance of ResponseMessage with "Course Updated Successfully" message.

- DELETE "/courses/{id}" (@PathVariable Long id): Delete a course by its ID and return an instance of ResponseMessage with "Course Deleted Successfully" message.

Note: You need to return the instance of the ResponseMessage class as shown below for the above API:

return new ResponseMessage("Course Added Successfully");

4. The controller method names for the APIs should be as follows:

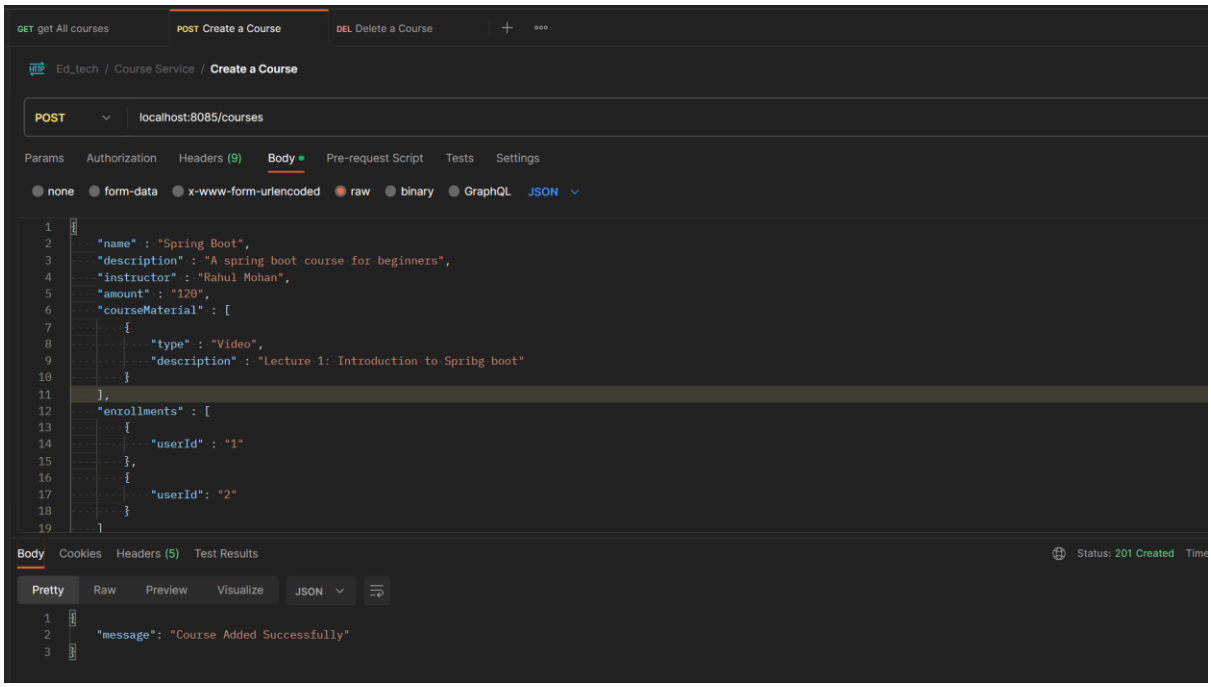
- GET "/courses": getAllCourses()
- GET "/courses/{id}": getCourseById()
- GET "/courses/name/": getCourseByName()
- GET "/courses/courseMaterial/": getCourseMaterialByCourseId()
- GET "/courses/instructor/": getCourseByInstructor()
- POST "/courses": createCourse()
- PUT "/courses/{id}": updateCourse()
- DELETE "/courses/{id}": deleteCourse()

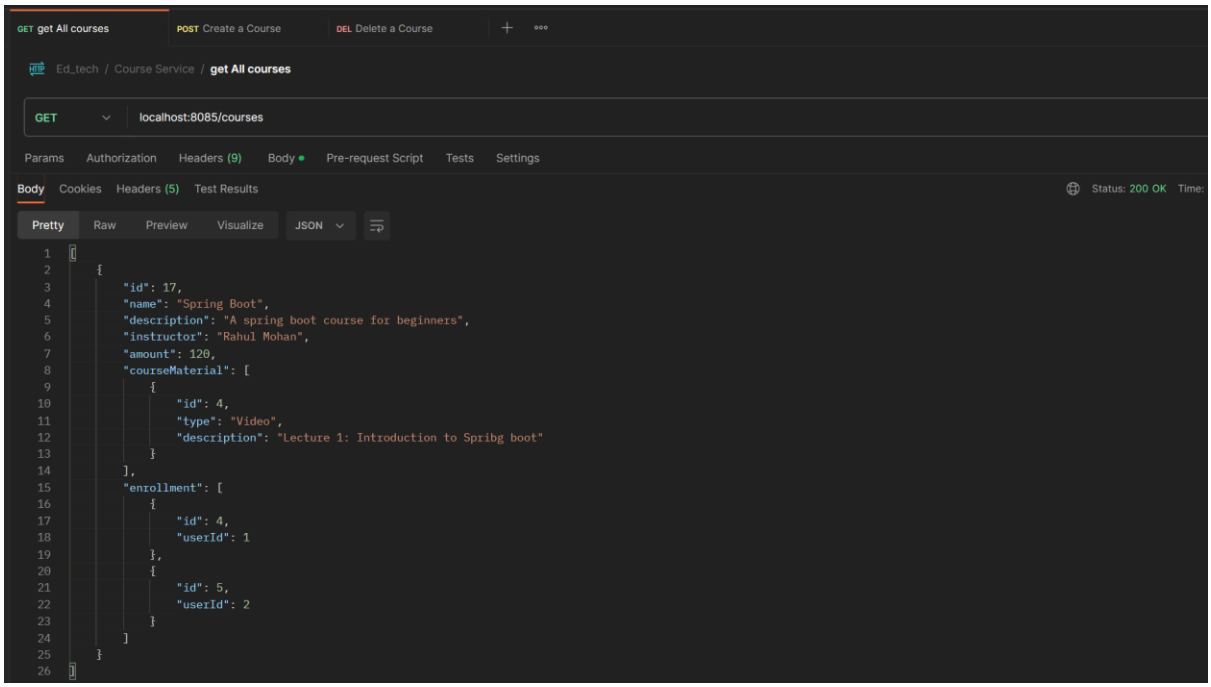
5. Create a service class with the name **"CourseService"** for the course Controller and implement business logic.

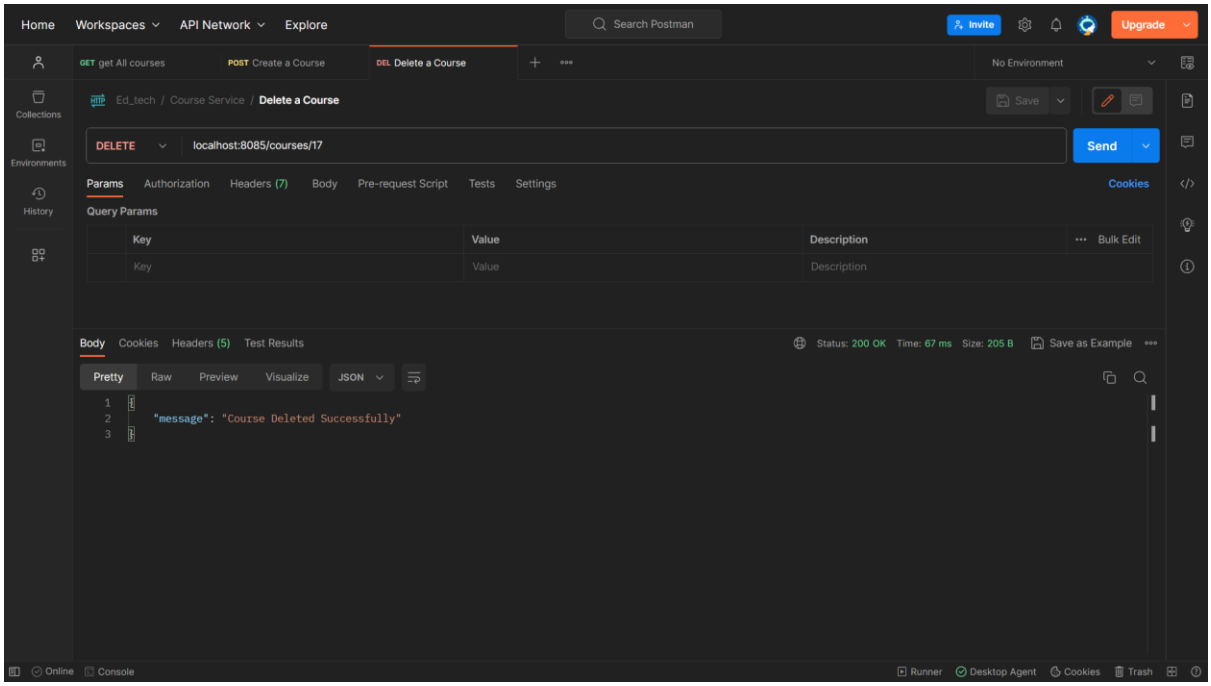
6. The service class method names should be as follows:

- GET /courses: getAllCourses()
- GET /courses/{id}: getCourseById(Long id)
- GET /courses/name/: getCourseByName(String name)
- GET /courses/courseMaterial/: getCourseMaterialByCourseId(Long id)
- GET /courses/instructor/: getCourseByInstructor(String instructor)
- POST /courses: createCourse(CourseDto courseDto)
- PUT /courses/{id}: updateCourse(Long id, CourseDto updatedCourseDto)
- DELETE /courses/{id}: deleteCourse(Long id)

Output:


GET get All courses
POST Create a Course
DEL Delete a Course
Ed_tech / Course Service / Create a Course
POST
localhost:8085/courses
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
{
"name" : "Spring Boot",
"description" : "A spring boot course for beginners",
"instructor" : "Rahul Mohan",
"amount" : "120",
"courseMaterial" : [
{
"type" : "Video",
"description" : "Lecture 1: Introduction to Spribg boot"
}
],
"enrollments" : [
{
"userId" : "1"
},
{
"userId": "2"
}
]
Body
Cookies
Headers (5)
Test Results
Status: 201 Created
Pretty
Raw
Preview
Visualize
JSON
{
"message": "Course Added Successfully"
}


GET get All courses
POST Create a Course
DEL Delete a Course
Ed_tech / Course Service / get All courses
GET
localhost:8085/courses
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Body
Cookies
Headers (5)
Test Results
Status: 200 OK
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": 17,
"name": "Spring Boot",
"description": "A spring boot course for beginners",
"instructor": "Rahul Mohan",
"amount": 120,
"courseMaterial": [
{
"id": 4,
"type": "Video",
"description": "Lecture 1: Introduction to Spribg boot"
}
],
"enrollment": [
{
"id": 4,
"userId": 1
},
{
"id": 5,
"userId": 2
}
]
}
]


Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
GET get All courses
POST Create a Course
DEL Delete a Course
No Environment
Collections
Environments
History
Ed_tech / Course Service / Delete a Course
Save
DELETE
localhost:8085/courses/17
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Body
Cookies
Headers (5)
Test Results
Status: 200 OK
Time: 67 ms
Size: 205 B
Save as Example
Pretty
Raw
Preview
Visualize
JSON
{
"message": "Course Deleted Successfully"
}
Online
Console
Runner
Desktop Agent
Cookies
Trash