Scheduler API

# Prerequisites

- Java 11 or higher  
- Maven  
- Spring Boot  
- MySQL or H2 Database (for development)  
- Postman (for API testing)

# Getting Started

## 1. Clone the Repository

git clone https://github.com/your-repository/scheduler-api.git  
cd scheduler-api

## 2. Configure the Database

Update the `application.properties` file located in `src/main/resources/` with your database configuration:

# For MySQL  
spring.datasource.url=jdbc:mysql://localhost:3306/scheduler\_db  
spring.datasource.username=root  
spring.datasource.password=yourpassword  
spring.jpa.hibernate.ddl-auto=update  
  
# For H2 Database (development)  
# spring.datasource.url=jdbc:h2:mem:testdb  
# spring.datasource.driverClassName=org.h2.Driver  
# spring.datasource.username=sa  
# spring.datasource.password=password  
# spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

## 3. Build and Run the Application

mvn clean install  
mvn spring-boot:run

## 4. Access the API

The API will be running on `http://localhost:8080/api`.

# API Endpoints

## 1. Mentor APIs

List all mentors: `GET /api/mentors`  
Returns a list of all mentors.

Get mentor by ID: `GET /api/mentors/{id}`  
Retrieves details of a specific mentor by ID.

Create a new mentor: `POST /api/mentors`  
Creates a new mentor.  
  
Example Request Body:

{  
 "name": "John Doe",  
 "areaOfInterest": "Java Development"  
}

Update mentor details: `PUT /api/mentors/{id}`  
Updates details of an existing mentor.  
  
Example Request Body:

{  
 "name": "John Smith",  
 "areaOfInterest": "Web Development"  
}

Delete a mentor: `DELETE /api/mentors/{id}`  
Deletes a mentor by ID.

## 2. Student APIs

List all students: `GET /api/students`  
Returns a list of all students.

Get student by ID: `GET /api/students/{id}`  
Retrieves details of a specific student by ID.

Create a new student: `POST /api/students`  
Creates a new student.  
  
Example Request Body:

{  
 "name": "Jane Doe",  
 "email": "jane.doe@example.com"  
}

Update student details: `PUT /api/students/{id}`  
Updates details of an existing student.  
  
Example Request Body:

{  
 "name": "Jane Smith",  
 "email": "jane.smith@example.com"  
}

Delete a student: `DELETE /api/students/{id}`  
Deletes a student by ID.

## 3. Session APIs

List all sessions: `GET /api/sessions`  
Returns a list of all sessions.

Get session by ID: `GET /api/sessions/{id}`  
Retrieves details of a specific session by ID.

Create a new session: `POST /api/sessions`  
Creates a new session.  
  
Example Request Body:

{  
 "studentId": 1,  
 "mentorId": 2,  
 "startTime": "2024-08-15T10:00:00",  
 "endTime": "2024-08-15T11:00:00",  
 "duration": 60,  
 "fee": 6000  
}

Update session details: `PUT /api/sessions/{id}`  
Updates details of an existing session.  
  
Example Request Body:

{  
 "studentId": 1,  
 "mentorId": 2,  
 "startTime": "2024-08-15T12:00:00",  
 "endTime": "2024-08-15T13:00:00",  
 "duration": 60,  
 "fee": 6000  
}

Delete a session: `DELETE /api/sessions/{id}`  
Deletes a session by ID.

## 4. Booking and Scheduling APIs

Book a 1x1 session: `POST /api/sessions/book`  
Books a new 1x1 session between a student and a mentor.  
  
Example Request Body:

{  
 "studentId": 1,  
 "mentorId": 2,  
 "preferredDuration": 60  
}

Get available mentors for a student: `GET /api/mentors/available`  
Retrieves a list of mentors available for a specific area of interest.  
  
Example Request Parameter:  
  
```  
/api/mentors/available?areaOfInterest=Java Development  
```

Calculate session fee: `GET /api/sessions/fee`  
Calculates the fee for a session based on its duration and mentor.  
  
Example Request Parameters:  
  
```  
/api/sessions/fee?duration=60&mentorId=2  
```

# Testing the API

You can use Postman or any other API testing tool to interact with the API endpoints. Import the provided endpoints into your API testing tool and perform CRUD operations on mentors, students, and sessions.