

- You have to solve the problem described below:
 - Room Booking System
- You have to use SpringBoot to create the REST APIs as described under each problem statement
- You need to use the H2 database to store any persistent data. Please use

```
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
spring.h2.console.enabled=true
```

properties file. *Do not change the username, password and DB name in the properties file, or your submission will not run on our test platform.* You may add other configuration parameters if required.

- The base URL for the application is: <http://127.0.0.1:8080>
- You are advised to use IntelliJ IDEA for development and Postman for API Testing
- What to submit:
 - You will be required to submit a jar of your application
 - And, the source code for the application
- Evaluation:
 - The source code will be compiled to check that it generates the same jar that you submitted. If it does not, your submission will be rejected
 - The app will be run on port 8080 and some test API calls will be made such that at a given instant it will return a unique response
 - You will be marked for the number of test cases correctly passed and the quality of your app design
- Please note:
 - All requests and responses are in JSON format
 - All dates are in the yyyy-MM-dd format
 - All times are in the hh:mm:ss format

Problem : Room Booking Portal

LOGIN: Endpoint to login an existing user. If the user does not exist or the credentials are incorrect, a relevant error is thrown

URL: /login

Method: POST

Body: email <str>
password <str>

Response:<One of the following Messages>

- Login Successful
- Username/Password Incorrect
- User does not exist

SIGNUP: Endpoint to register a new user. If the account already exists, relevant error should be returned

URL: /signup

Method: POST

Request Body: email <str>

name <str>

password <str>

Response: <One of the following Messages>

- Account Creation Successful
- Forbidden, Account already exists

USER DETAIL: Endpoint to retrieve details about a user. Relevant error is returned if the user does not exist

URL: /user

Query Parameter: userID

Method: GET

Response Body: One of:

- name <str>
userID <int>
email <str>
- User does not exist

ROOM BOOKING HISTORY: Endpoint to retrieve the room booking history for a particular user **URL:** /history

Query Parameter: userID

Method: GET

Response Body: <objects>

-room <str>
-roomID<int>
-bookingID<int>
-dateOfBooking<date>
-timeFrom<str>
- timeTo<str>
-purpose<str>

Error:

Room does not exist

User does not exist

UPCOMING ROOM BOOKING: Endpoint to retrieve the upcoming room bookings for a particular user

URL: /upcoming

Query Parameter: userID

Method: GET

Response Body: <object>

- room <str>
- roomID<int>
- bookingID<int>
- dateOfBooking<date>
- timeFrom<str>
- timeTo<str>
- purpose<str>

Error:

User does not exist

ROOMS: Endpoints related to rooms

URL: /rooms

Method: GET

Query Parameters: Filter on Date, time, capacity

Response Body: room <object>

- roomID<int>
- capacity<int>
- booked<object>
 - bookingID<int>
 - dateOfBooking<date>
 - timeFrom<str>
 - timeTo<str>
 - purpose<str>
 - userID<int>

Error:

Invalid parameters

To add a new room:

Method: POST

Request Body:

roomName<str>

roomCapacity<unsigned int>

Response: One of:

- Room created successfully
- Room already exists
- Invalid capacity

To edit a room:

Method: PATCH

Request Body:

roomID<int>

roomName<str>

roomCapacity<unsigned int>

Response: One of:

- Room edited successfully
- Room does not exist
- Invalid capacity

To delete a room:

Method: DELETE

Request Body:

roomID<int>

Response: One of:

- Room deleted successfully
- Room does not exist

BOOKING: Endpoints related to bookings

URL: /book

To create a new Booking:

Method: POST

Body: userID<int>

roomID<int>

dateOfBooking<date>

-timeFrom<str>

-timeTo<str>

-purpose<str>

Response: (Any of the following)

- Booking created successfully
- Room unavailable
- Room does not exist
- User does not exist
- Invalid date/time

To edit an existing booking:

Method: PATCH

Body: -userID<int>
-roomID<int>
-bookingID<int>
-dateOfBooking<date>
-timeFrom<str>
-timeTo<str>
-purpose<str>

Response: (Any of the following) ●
Booking modified successfully ●
Room unavailable
● Room does not exist
● User does not exist
● Invalid date/time

Method: DELETE

Body: bookingID<int>

Response: (One of the following) ●
Booking deleted successfully ●
Booking does not exist