**Managing Hotel Bookings with help of DB**

**Requirements**

You are provided with models - Booking, Guest and Room and dtos BookingRequestDto , BookingResponseDto, RoomRequestDto

1. You need to implement following APIs in BookingController

* GET /booking/guest/{guestEmail} which will return all Bookings done by guest taking guestEmail as input and returning output in form of List<BookingResponseDto>
* GET /booking/{bookingId} which will return Booking detail corresponding to a bookingId in form of BookingResponseDto
* POST /booking to create a booking which will take body in form of BookingRequestDto and return output in form of BookingResponseDto
* PUT /booking/{bookingId} to replace a booking having particular bookingId, taking body in form of BookingRequestDto and return output in form of BookingResponseDto
* DELETE /booking to delete a booking having particular bookingId and return Boolean result
* All these APIs will call methods mentioned in IBookingService

1. You also need to add implementation in methods present in StorageBookingService

* getBooking will get booking from DB using bookingId, if not found return null
* getAllBookingsPerGuest will get all Bookings for a particular Guest from DB using GuestRepo and BookingRepo.
* createBooking will create a new Booking using bookingRequestDto. You need to save Booking and Room info into DB. Please use given private methods to create Room object from RoomRequestDto and Date object from date given in String form in bookingRequestDto. Total Bill will be calculated by Adding (Room Rent of specific type) \* (Room count of specific type) for each Room Type for total number of days person is staying. In case person is checking out on same day, we will consider his stay duration as 1 day.Make sure all fields are set up in Booking before persisting into DB.
* replaceBooking will delete Booking with particular BookingId if it exists and then create a new Booking having new BookingId with help of bookingRequestDto. If Booking with given BookingId doesn't exist in DB, then don't create a new Booking and return null.
* deleteBooking will delete Booking with particular bookingId if it exists and return true. If booking is not there, then return false

1. You also need to define cardinality between models and make sure tables are created for them. Fields are already present, no new field need to be added, no field need to be removed or modified. Just annotate with relevant annotations. Apart from Cardinalities, you may also need to add Cascade , JsonManagedReference , JsonBackReference

**Testing**

You can also check which tables with what fields are created in H2 by running Application in IntellIJ and opening http://localhost:8080/h2-console on browser and put values as below

* Saved Settings: Generic H2(Embedded)
* Setting Name: Generic H2(Embedded)
* Driver Class: org.h2.Driver
* JDBC URL: jdbc:h2:mem:class9\_ques4
* User Name: sa
* Password: password
* click Connect

**Hints**

* Nothing is needed from your side in pom.xml or application.properties
* No new file need to be created.
* No new field need to be added, No field need to be removed or modified. No change is needed in dtos , IBookingService repos.
* If you will try to run testcases without defining relations, all Testcases will fail.