

Further Improvements Needed

- Instead of returning Database models to UI, response classes should be used. I would use AutoMapper to map between the type. Although manual mapping also possible.
- When handling API requests, request classes should be used instead of Database model classes.
- Need more unit test and more detailed assertions.
- To meet the (Booking numbers should be unique. not overlap) requirement, I am using the primary key of the Booking table, this will ensure uniqueness. Another option was to use a GUID but I felt that would make it too long. We can always set the starting point of the Booking Id to an arbitrary higher number.
- No validation of customer is added, to add a booking, a customer with the customer id in the request must exist.
- No validation on Bookings (it is possible to book a room in the past)
- Assumption: When checking availability, I have allowed searching across all available hotels. Also, the booking controller will book the first available room of the correct capacity.
- This behaviour can easily be changed by adding Hotel Id to get rooms and book room requests.

Testing

Setup:

1. Run the HotelApp project
2. Swagger should load at <https://localhost:7222/swagger/index.html>
3. Execute the Post endpoint under Seed Controller

This will create the database and add some test data:

Hotel Table

Id	Name
1	Hilton
2	Marriott
3	Ibis

Customer Table

Id	Name
1	John Doe
2	tony Stark

Room Table

Id	Capacity	RoomType	HotelId
1	1	0	1

2	2	1	1
3	4	2	1
4	1	0	1
5	2	1	1
6	4	2	1
7	1	0	2
8	2	1	2
9	3	2	2
10	2	1	2
11	2	1	2
12	2	1	2
13	1	0	3
14	2	1	3
15	5	2	3
16	2	1	3
17	5	2	3
18	5	2	3

Room Type Key (0 = Single, 1 = Double, 2 = Deluxe)

Use case: Find a hotel based on its name

1. Call the Get endpoint for Hotels using "Hilton", response should be 200 and hotel entity should be returned.
2. Call the Get endpoint for Hotels using "Novotel", response should be 404 (not found).

Use case: Find available rooms between two dates for a given number of people

1. Call the Get endpoint for Rooms using: 5 as value for occupants, and any two dates (2022-07-28T12:49:48.504Z and 2022-07-29T12:49:48.504Z) > 3 rooms should return

Use case: Book Room

1. Call Post end point For Booking Controller with request:

```
{
  "checkInDate": "2022-07-28T12:58:47.264Z",
  "checkOutDate": "2022-07-28T12:58:47.264Z",
  "customerId": 1,
```

```
"occupants": 5
```

```
}
```

Response should be 200 (success) and Booking with room Id is returned.

2. (Repeat previous step) Call the Get endpoint for Rooms using: 5 as value for occupants, and any two dates (2022-07-28T12:49:48.504Z and 2022-07-29T12:49:48.504Z) > only 2 rooms should return

Use case: Find booking details based on booking ref number

1. Call the Get endpoint for booking controller using 1 as bookingId, the booking should return.

Once finished with testing, delete endpoint of Seed controller can be used to clear all the data.