

:

## Integration Tests – Hotel Reservation System

**Name:** Pooja Nadagouda

### Steps to run the tests:

1. Install maven if not present on your machine.
2. Unzip IntegrationTest.zip file
3. Open command prompt and change directory path to IntegrationTest folder by using following command:  
If the folder is on desktop, then using the command below:  
***cd Desktop/IntegrationTest***
4. Now run the following command:  
***mvn clean***
5. Once BUILD SUCCESS is obtained from above command execution, Run the following command to execute the integration tests:  
***mvn test***
6. Once BUILD SUCCESS is obtained the tests execution is complete and you can see the test cases passed or failed status.
7. To view full report of test cases passed/failed, open the target folder which is created inside the IntegrationTest folder.
8. Now inside the target folder open surefire-reports folder. Search for ***emailable-report.html*** and ***index.html*** files. Open these files to view full report. Also, another detailed report is present in ***All Test Suites*** folder.

Paths for reports:

***.../IntegrationTest/target/surefire-reports/emailable-report.html***

***.../IntegrationTest/target/surefire-reports/index.html***

***.../IntegrationTest/target/surefire-reports/All Test Suite/\_IntegrationTest.html***

### Starting and stopping the middle tier:

This application is hosted on AWS hence it is always running. To stop the application, we should delete the deployment through the AWS management console using admin credentials.

### Application Services and corresponding testcases:

For each of the following interface 1 positive and 1 negative scenario test case is added for testing.

Here TestNG framework is used for writing integration tests for the public interfaces.

:

### 1. RoomBookingService:

#### Interfaces:

- /Bookings/POST – adds the new booking details record to the database
- /Bookings/GET – gets booking details from database for given booking ID

#### TESTCASES:

##### RoomBookingServiceTest.java – Class for testing the above interfaces

1. testGetBookingPositiveCase
2. testGetBookingNegativeCase
3. testAddBookingPositiveCase
4. testAddBookingNegativeCase

### 2. RoomAvailabilityService:

#### Interfaces:

- /Availability/POST – adds room availability details for a date to database
- /Availability/GET – gets room availability details for a date from database
- /Availability/PUT – updates room availability details for a date in database

#### TESTCASES:

##### RoomAvailabilityServiceTest.java - Class for testing the above interfaces

1. testGetAvailabilityPositiveCase
2. testGetAvailabilityNegativeCase
3. testUpdateAvailabilityPositiveCase
4. testUpdateAvailabilityNegativeCase
5. testAddAvailabilityPositiveCase
6. testAddAvailabilityNegativeCase

### 3. RoomService:

#### Interfaces:

- /Room/POST – adds new room-type to the database
- /Room/GET – gets details of a room-type from the database
- /Rooms/GET – gets details of all room-types from the database

#### TESTCASES:

##### RoomServiceTest.java - Class for testing the above interfaces

1. testGetAllRooms
2. testGetRoomPositiveCase
3. testGetRoomNegativeCase
4. testAddRoomPositiveCase
5. testAddRoomNegativeCase