**This service contains three servers (All are separate Main classes can run independently)**

* com.prabathj.bookinggw.AppServer.class
* com.prabathj.bookingsrv.BookingService.class
* com.prabathj.hotelavailabilitysrv.AvailabilityService.class

**\*AppServer** – Behave as the API GW and configuration management server. Configurations are stored in **data/configdata.json** file this will be loaded as per the incoming requests. Any Request will be directed accordingly to relevant services

**URI- http://localhost:5030/api/booking**

**\*BookingService**- Handle booking process

**URI- http://localhost:5031/api/booking**

**\*AvailabilityService** - Contain the persisted data of bookings, read configuration data at start-up so **AppServe** should be up and running before start this service

**URI-** [**http://localhost:5031/api/book**](http://localhost:5031/api/book)

1. Booking Request – POST request

**Booking Request Path**

http://localhost:5030/api/booking

http://localhost:5031/api/booking

BookingService

APPSERVER

http://localhost:5032/api/book/prebook

http://localhost:5032/api/book/confirmbook

AvailabilityService

**Booking Data Model -** **com.prabathj.hotelavailabilitysrv.model.BookingInfo**

**class** BookingInfo{

**private** String name;

**private** Integer room;

@JsonFormat(pattern = "yyyy-MM-dd")

**private** LocalDate bookingDate;

}

**Response**

**Booking confirm – 201**

**Error codes- 400**

\*Pre-Booking request sent to check room availability and temporary lock the booked room. This is an asynchronized request

\*After pre-booking response customer get the booking success response and a separate request send to confirm the booking. Platform can perform any booking related task at this stage, and it is time consumption does not affect to the customer.

1. Search Requests – GET request

**Search Request Path**

http://localhost:5030/api/booking?date=2023-03-09

**R**

http://localhost:5032/api/book?name=xyz

http://localhost:5032/api/book?date=2023-03-09

AvailabilityService

APPSERVER

http://localhost:5030/api/booking?name=xyz

**Response**

Search Available Rooms on a given date

Response Model - com.prabathj.hotelavailabilitysrv.model.AvailableRooms

**class** AvailableRooms {

**private** List<Integer> rooms;

}

Search Booked Rooms of a Customer name

Response Model - com.prabathj.hotelavailabilitysrv.model.Bookings

**class** Bookings {

**private** List<BookingInfo> bookings;

}

**Configuration Request -** GET Request

If a service required config data, it could request data from the AppServer at the start-up. Config data are stored in **data/configdata.json**

AvailabilityService

http://localhost:5030/api/booking?config

APPSERVER

**Response**

com.prabathj.hotelavailabilitysrv.model.ConfigData

**class** ConfigData {

**private** **int** no\_rooms;

}

**Assumptions**

Number of rooms do not configure frequently. When configure all bookings will be cancelled and AvilabilityService required a reboot.

Dynamic Service registration is not available, static values are used for service URIs

Any issue in the booking confirmation process need to forward for manual consideration

**Test cases**

\*Test cases are written using **Junit5 & rest-assured** frameworks and libraries.

\*Test cases are written for manual Testing and automated Testing

com.prabathj.bookinggw.BookingCompleteProcessAuoServerStartTest - Automated Testing and sample data for the Booking and search process use **maven test/install** command

\*The following Test cases are for manual testing (Servers need to start separately) and they are disabled from the class (Remove @Disabled for manual testing)

com.prabathj.bookinggw.BookingCompleteProcessMannualServerStartTest

com.prabathj.bookinggw.BookingSearchByDayTest

com.prabathj.bookinggw.BookingSearchPerPersonTest