**Sweet-Home Application**

The Sweet-Home app is a **Hotel room booking application** that allows people book hotel rooms and pay for their bookings.

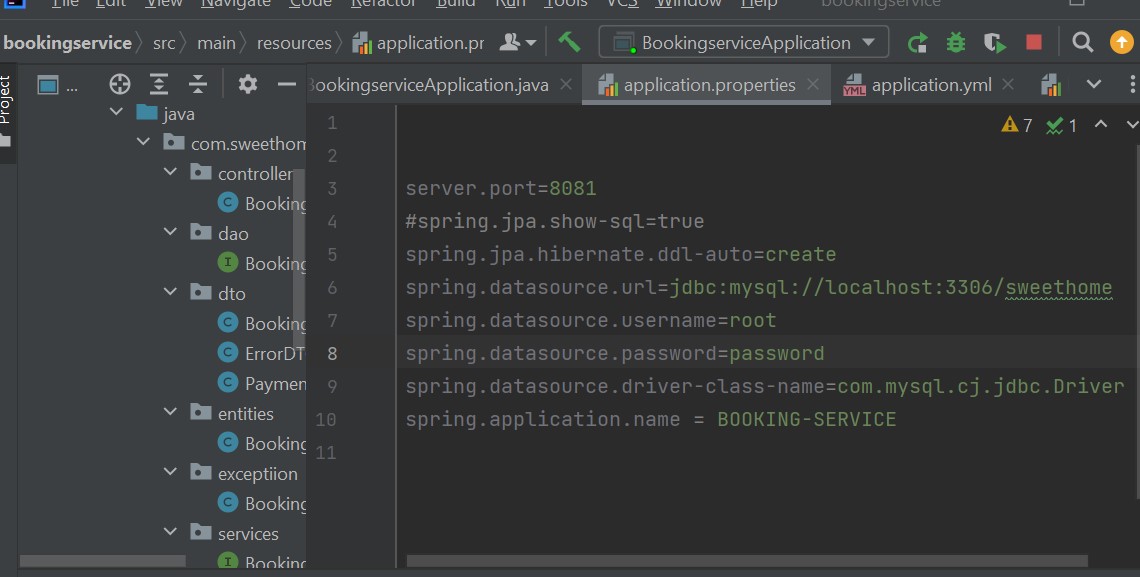
The microservices of this app are **Booking service and Payment service** and Eureka server service. The **Booking service and Payment service** are registered as clients to the **Eureka-server.**

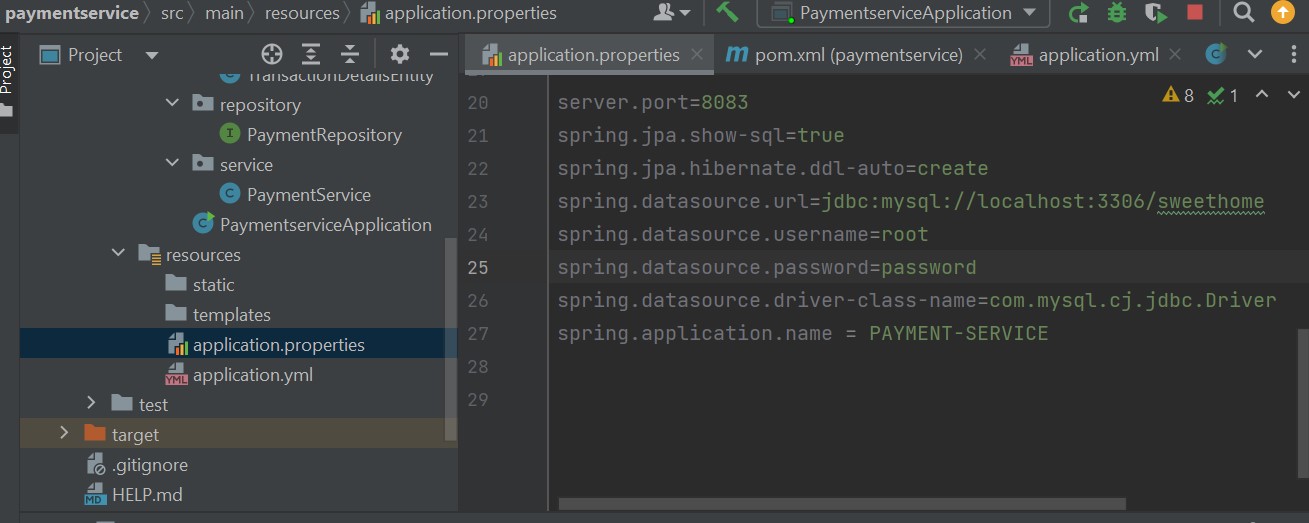
The project structure was created using Spring Initializer and the microservices- 'Booking', 'Payment', and 'Eureka server' were created inside “Sweet-Home”. The database used is MySQL.

Synchronous communication between booking and payment service was set up using the REST template of Spring Boot.

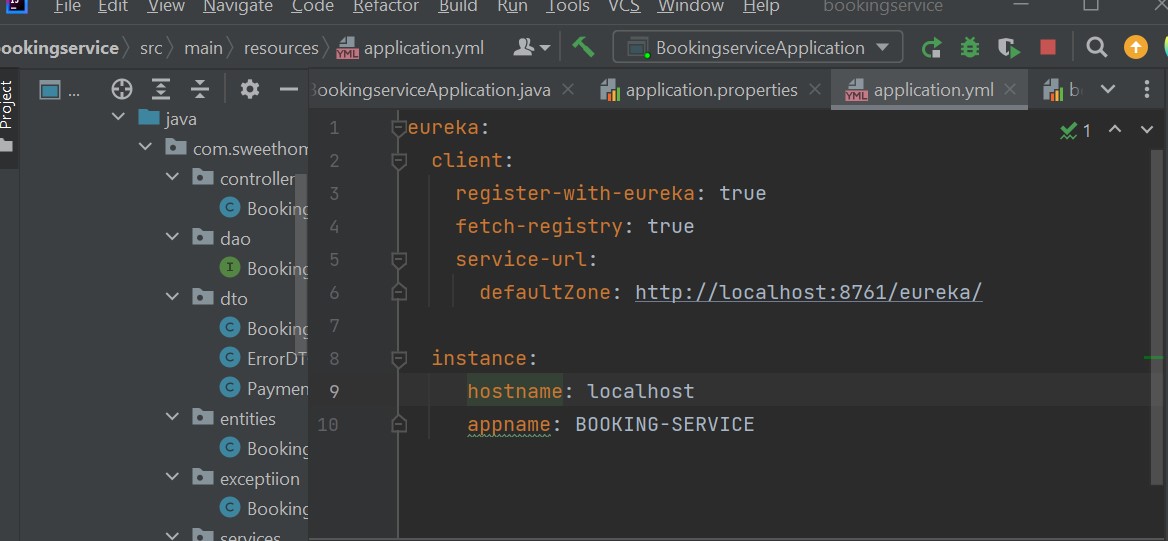
The steps used to run the project are as follows:

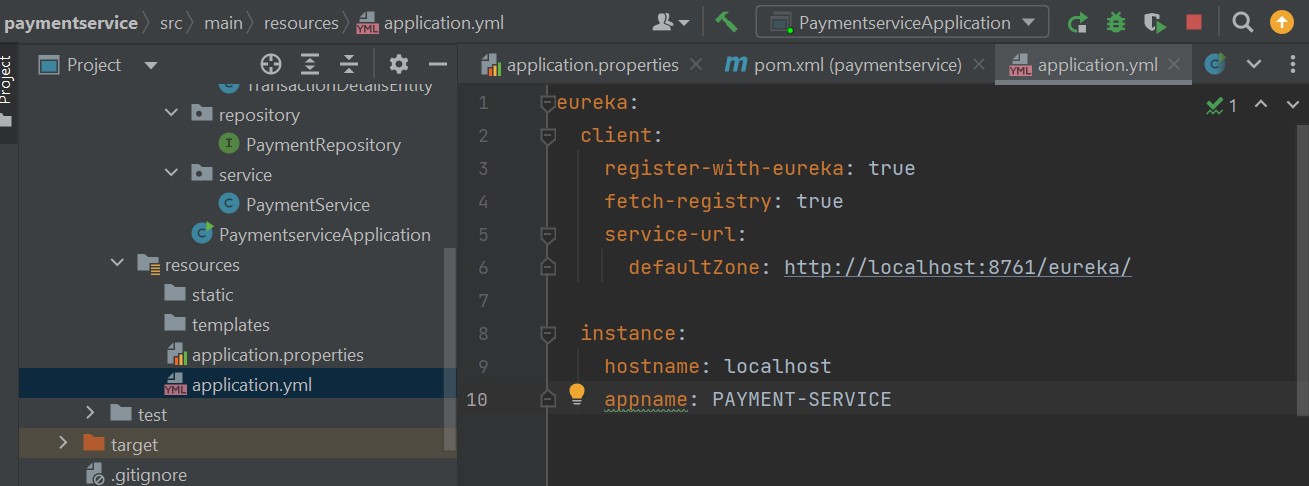
* MySQL Workbench was used to set up our MySQL and schema named sweethome was created.
* The different configuration which is required to run the application was added to the application.properties file in both booking & payment services.





* The application.yml file shows the configuration of the eureka instance URL





* The Booking Service takes inputs from users like- toDate, fromDate, aadharNumber and the number of rooms required (numOfRooms) and were saved it the database.

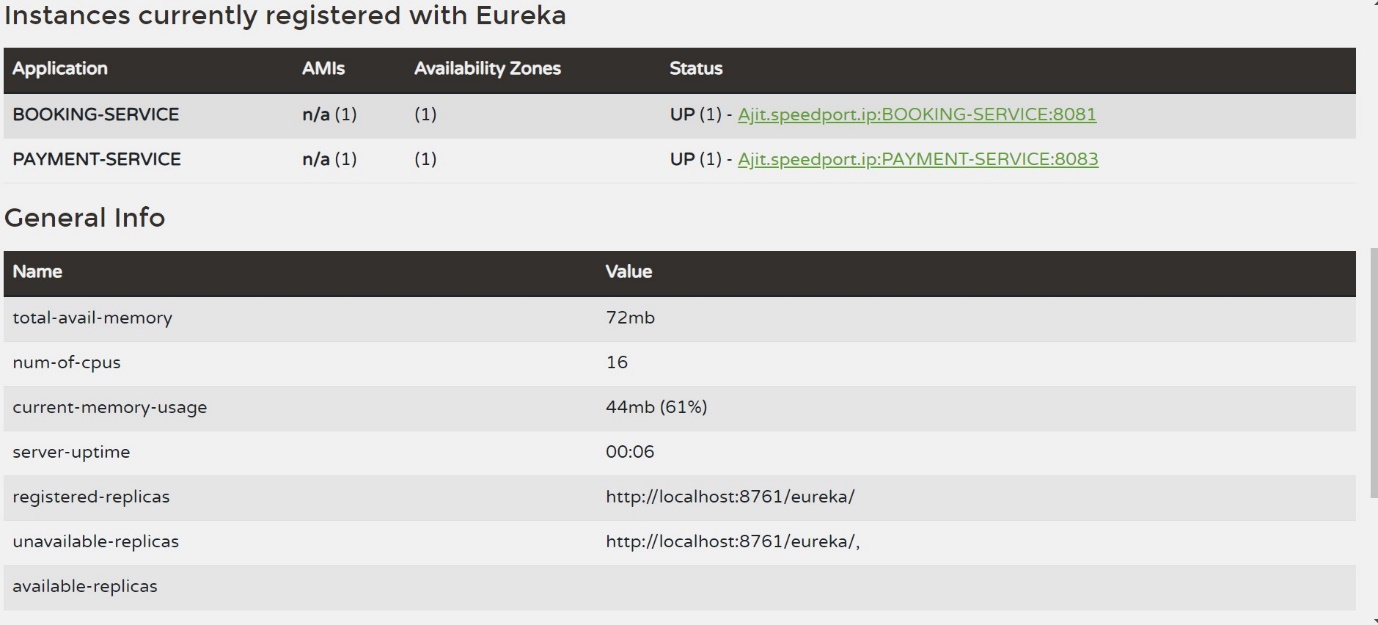
The logic used to generate a random list of room numbers depending on 'numOfRooms' requested by the user is :

public ArrayList<String> getRandomNumbers(int count){  
 Random rand = new Random();  
 int upperBound = 100;  
 ArrayList<String>numberList = new ArrayList<String>();  
  
 for (int i=0; i<count; i++){  
 numberList.add(String.*valueOf*(rand.nextInt(upperBound)));  
 }  
  
 return numberList;  
}

The logic to calculate room price is :

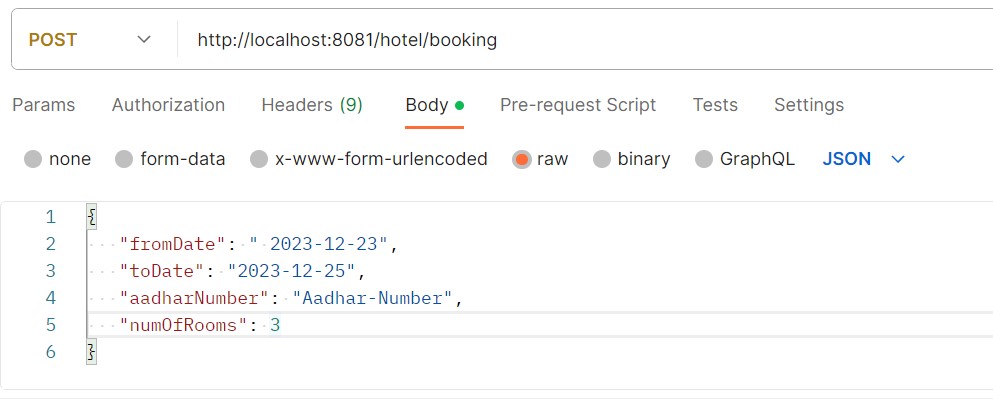
bookingInfo.setRoomPrice(Math.*toIntExact*((1000 \* bookingInfo.getNumOfRooms() \* (ChronoUnit.*DAYS*.between((Temporal) bookingInfo.getFromDate(), (Temporal) bookingInfo.getToDate())))));

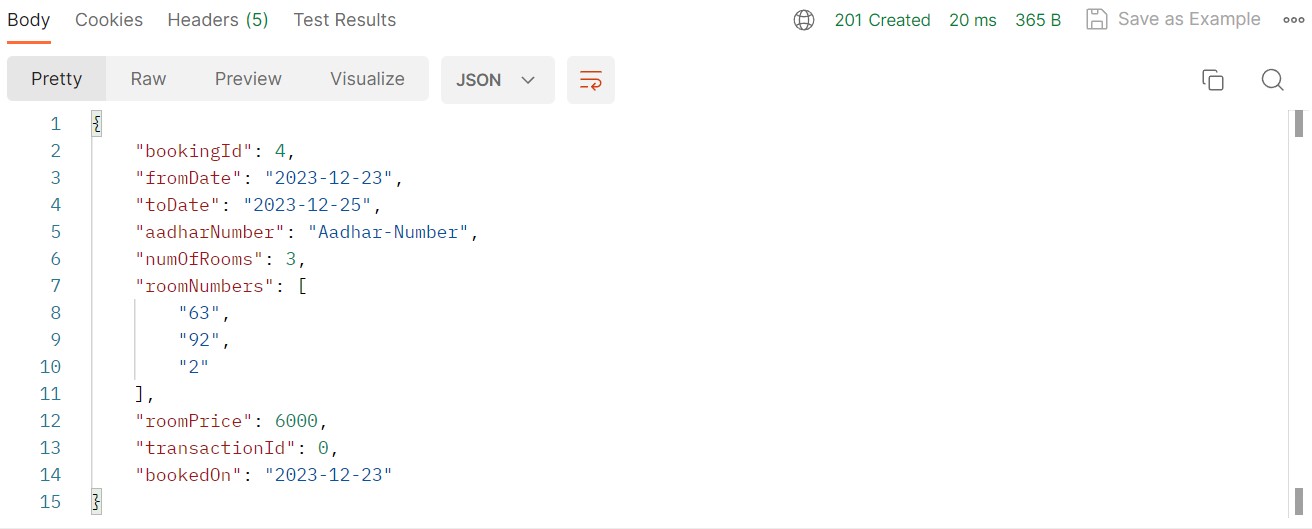
* The Eureka server service was created and the booking and payment services are registered to the server. the records for both the booking service and payment service are seen when we go to <http://localhost:8761>



* After we run the booking service, payment service and eureka server service simultaneously, the Postman endpoint tests show the results as follows:

1. localhost:8081/booking





1. localhost:8081/booking/1/transaction
2. localhost:8083/transaction
3. localhost:8083/transaction/1