

Kora Sree Satya([sreesatya505@gmail.com](mailto:sreesatya505@gmail.com))

Akash Kesarwani([akashkesarwani66@gmail.com](mailto:akashkesarwani66@gmail.com))

### Note:

- a. Create database *bookings* and *transactions* before running the application.
- b. Set username and password in the *application.properties* file

Following APIs are implemented. There are two microservices which provide these 4 APIs.

1. *booking* which provide API 1 & 2.
2. *payment* which provide API 3 & 4.

### APIs:

- 1) **POST:** <http://localhost:9191/hotel/booking>  

```
{  
  "fromDate": "2021-06-10",  
  "toDate": "2021-06-15",  
  "aadharNumber": "Sample-Aadhar-Number",  
  "numOfRooms": 3  
}
```
- 2) **POST** <http://localhost:9191/hotel/booking/1/transaction>  

```
{  
  "paymentMode": "UPI",  
  "bookingId": 1,  
  "upiId": "UPI details",  
  "cardNumber": ""  
}
```
- 3) **POST** <http://localhost:9191/payment/transaction>  

```
{  
  "paymentMode": "UPI",  
  "bookingId": 1,  
  "upiId": "UPI details",  
  "cardNumber": ""  
}
```
- 4) **GET** <http://localhost:9191/payment/transaction/1>

Both the microservices, *booking* and *payment* are registered on eureka server with name as BOOKING-SERVICE & PAYMENT-SERVICE respectively. There is also API-GATEWAY which is hosted on port 9191.

## Details Description of the APIs.

1. **POST:** <http://localhost:9191/hotel/booking>

Database table: **booking**

Entity class: BookingInfoEntity

DTO: BookingDTO

This service saves data, provided in post request body, in the booking table. Its also return BookingDTO object in the response body.

2. **POST:** <http://localhost:9191/hotel/booking/1/transaction>

Database table: **booking**

Entity class: BookingInfoEntity

DTO: BookingDTO

This service internally calls API #3 to save the transaction data related to booking id and thus return BookingDTO with updated transaction id.

3. **POST** <http://localhost:9191/payment/transaction>

Database table: **transaction**

Entity class: TransactionDetailsEntity

DTO: TransactionDTO

This service store transaction data related to booking id and thus return transaction id. This service is used by API #2.

4. **GET** <http://localhost:9191/payment/transaction/1>

Database table: **transaction**

Entity class: TransactionDetailsEntity

DTO: TransactionDTO

This service returns transaction details for the particular transaction id.