

Database that is used for this project is MY-SQL

```
spring:
  datasource:
    url: jdbc:mysql://localhost:3306/SweetHome
    username: root
    password: admin
    driver-class-name: com.mysql.cj.jdbc.Driver
```

Create the database named SweetHome in My-SQL workbench
And change the username and Password accordingly

Microservices that we used in this project are:

Name	Port
Service registry	8761
Api gateway	9191
Booking	8081
Payment	8083

ServiceRegistry : This is our Eureka Discovery Server, all other micro services are discovery client , which are registered to this Service [Service Registry]

Api-gateway : Added this optional service to our project , you can access Booking and Payment service using this api gateway, this service is registered with eureka server and in application.yml file we defined the routes for Booking service and for Payment service

Booking Service and Payment Service :

Booking service is responsible for taking input from users

Sample input:

```
{
  "fromDate": "2021-06-10",
  "toDate": "2021-06-15",
  "aadharNumber": "Sample-Aadhar-Number",
  "numOfRooms": 3
}
```

Booking Service has 3 layer : controller , service , Repo

1. GET : <http://localhost:9191/hotel/booking> this endpoint will display the all the booking details [extra feature]
2. POST : <http://localhost:9191/hotel/booking> : this is one of the important endpoint, Upon providing the user input, this endpoint will hit the controller layer,

Where in controller layer we have mapper , which converts DTO to entity

After that our request will reach to service layer of BOOKING-SERVICE

In service layer we are calculating the price and available rooms using Helper class

And then save the result into the database

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

This end point is for making the transaction , this endpoint is defined in booking service

Payload

```
{
  "paymentMode": "UPI",
  "bookingId": 1,
  "upiId": "upi details",
  "cardNumber": ""
}
```

Upon submitting this request it will hit booking service controller , and in controller mapping happens between DTO and Entity , after that it will hit service layer makePayment() method, In that method , we do some validation , before saving into database

We check Boking Id is already present in the database, if not we are going to raise the custom exception and this exception will get caught by our Global Exception handler

If bookingId in payload is not matching in what we passed in request, then also we are throwing custom exception

We are taking only two mode of payment UPI and CARD (lowercase and Uppercase, both are allowed)

If we pass Other mode of payment , then we are throwing the custom exception, which is then handled by our Global Exception Handler

If it passes all the validation, then we are communicating with our payment Service using rest template and we are creating the bean in MyConfig class

To get hold of PAYMENT-SERVICE, instance from the Eureka, we are making use of load balancer

```
ServiceInstance paymentServiceInstance = loadBalancerClient.choose("PAYMENT-SERVICE");
```

If it is not present in the loadbalancer then we are throwing the `IllegalStateException`

Else we are communicating with this end point : **`http://localhost:9191/payment/transaction`**

Which is defined in the payment service , this endpoint will generate the transactionId for the payment, after receiving the response from this endpoint and we are updating the Booking-service table entry with transaction Id

After successful transaction we are displaying the successful booking confirmation message with aadhar number in console

4. Post <http://localhost:9191/payment/transaction> this endpoint is defined in payment service, and it generates the transactionId for the specified bookingId

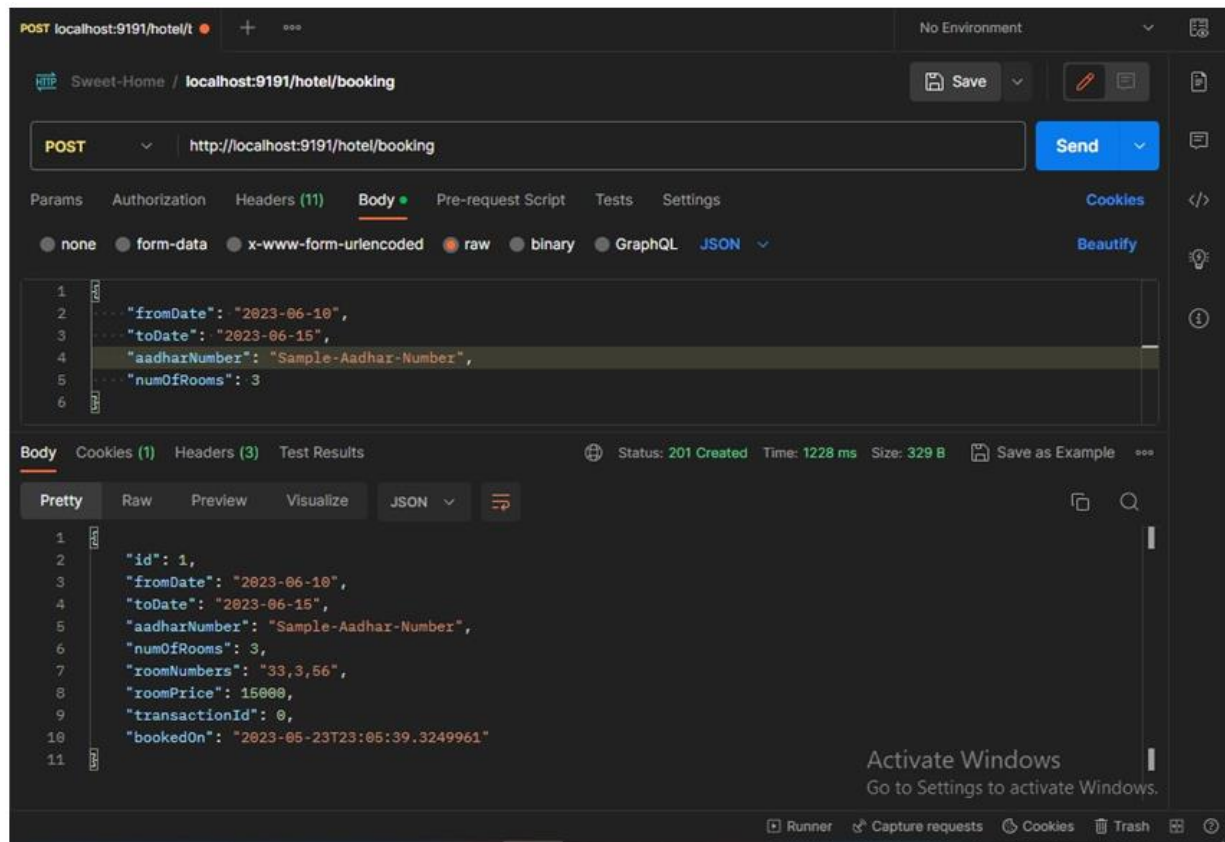
5. Get <http://localhost:9191/payment/transaction> this end point is defined in payment service , and it gives the transaction details based on transaction Id

Providing the screenshot for all of these:

Registered the Api-gateway, booking service, payment service with eureka client

Data center	default	Uptime	00:03
		Lease expiration enabled	false
		Renews threshold	6
		Renews (last min)	4
DS Replicas			
Instances currently registered with Eureka			
Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - DESKTOP-IPTLVPP-API-GATEWAY-9191
BOOKING-SERVICE	n/a (1)	(1)	UP (1) - DESKTOP-IPTLVPP-BOOKING-SERVICE-8081
PAYMENT-SERVICE	n/a (1)	(1)	UP (1) - DESKTOP-IPTLVPP-PAYMENT-SERVICE-8083
General Info			
Name	Value		
total-avail-memory	69mb		
num-of-cpus	4		

Accessing the booking service using api gateway and making the booking



Making the valid payment for the booking:

POST localhost:9191/hotel/t POST localhost:9191/hotel/bo... No Environment

Sweet-Home / localhost:9191/hotel/booking/1/transaction Save

POST http://localhost:9191/hotel/booking/1/transaction Send

Params Authorization Headers (11) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   ... "paymentMode": "UPI",
3   ... "bookingId": 1,
4   ... "upiId": "upi_details",
5   ... "cardNumber": ""
6 }
```

Body Cookies (1) Headers (3) Test Results Status: 201 Created Time: 1009 ms Size: 328 B Save as Example

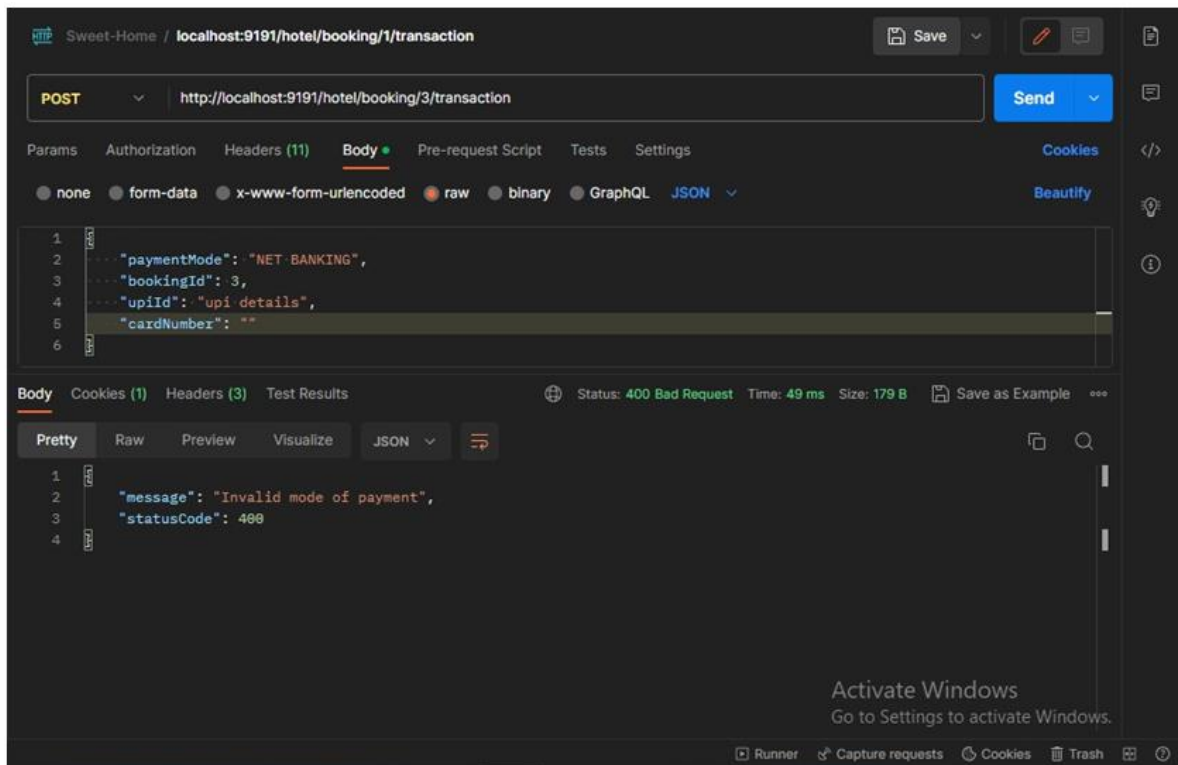
Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "fromDate": "2023-06-10",
4   "toDate": "2023-06-15",
5   "aadharNumber": "Sample-Aadhar-Number",
6   "numOfRooms": 3,
7   "roomNumbers": "33,3,56",
8   "roomPrice": 15000,
9   "transactionId": 2,
10  "bookedOn": "2023-05-23T23:05:39.324996"
11 }
```

Activate Windows
Go to Settings to activate Windows.

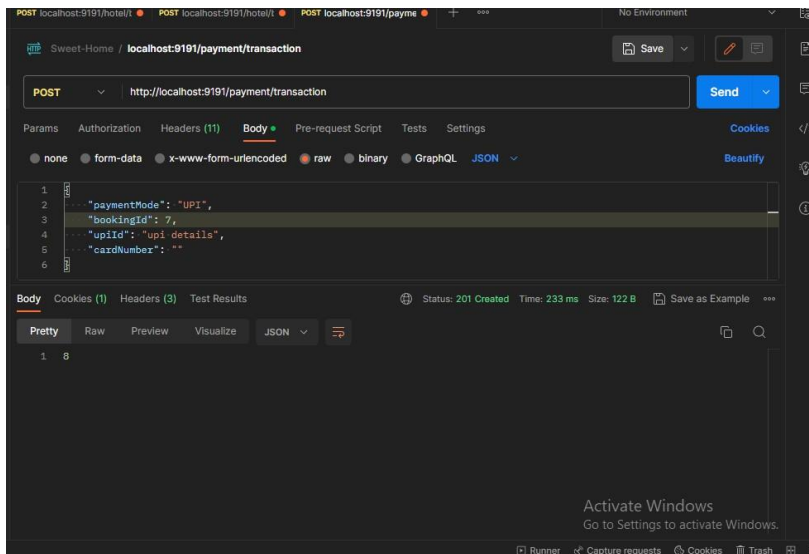
Runner Capture requests Cookies Trash

Invalid mode of payment

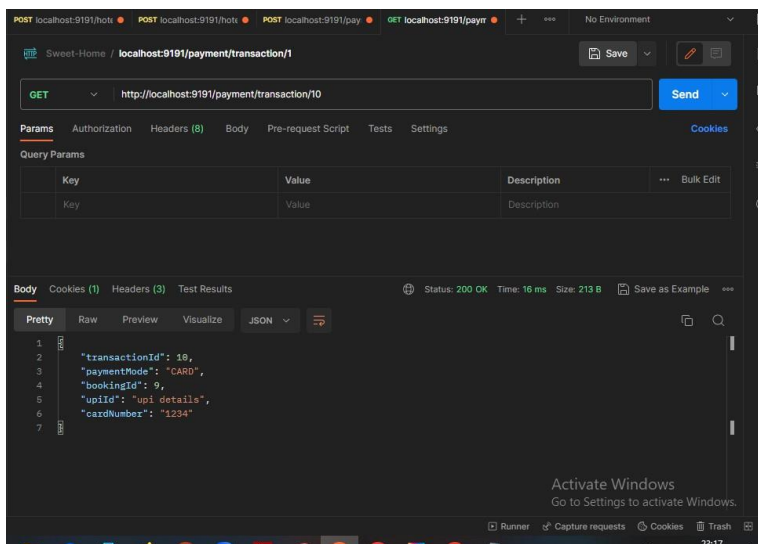


Making the payment for Invalid booking id:

Making the payment from payment end point:



Fetching the transaction:



Data from database :

Sweet-Home / localhost:9191/hotel/booking/1/transaction

POST

http://localhost:9191/hotel/booking/6/transaction

Send

Params

Authorization

Headers (11)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1  {
2    "paymentMode": "CARD",
3    "bookingId": 6,
4    "upiId": "upi details",
5    "cardNumber": ""
6  }
```

Body

Cookies (1)

Headers (3)

Test Results

Status: 400 Bad Request

Time: 19 ms

Size: 174 B

Save as Example

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "message": "Invalid Booking Id",
3    "statusCode": 400
4  }
```

Activate Windows

Go to Settings to activate Windows.

Runner

Capture requests

Cookies

Trash

Query | x micro_users booking

1 • select * from booking

2

Result Grid

id	aadhar_number	booked_on	from_date	num_of_room	room_numbers	room_price	to_date	transaction_id
1	Sample-Aadhar-Number	2023-05-23 23:05:39.324996	2023-06-10	3	33,3,56	15000	2023-06-15	2
3	Sample-Aadhar-Number	2023-05-23 23:09:39.362092	2023-06-10	3	80,33,54	15000	2023-06-15	4
5	Sample-Aadhar-Number	2023-05-23 23:12:07.139218	2023-06-10	4	11,89,38,83	20000	2023-06-15	6
7	Sample-Aadhar-Number	2023-05-23 23:14:02.449911	2023-06-10	4	35,47,86,24	20000	2023-06-15	0
9	Sample-Aadhar-Number	2023-05-23 23:16:39.703612	2023-06-10	3	65,22,67	24000	2023-06-18	10
11	1234	2023-05-23 23:18:48.667297	2023-07-12	3	73,74,51	18000	2023-07-18	12
13	1234	2023-05-23 23:19:21.330920	2023-09-12	3	38,34,62	18000	2023-09-18	14
NOTE	NOTE	NOTE	NOTE	NOTE	NOTE	NOTE	NOTE	NOTE

booking 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	23:21:03	select * from transaction LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clicki...	0.250 sec
2	23:21:15	use sweethome	0 row(s) affected	0.000 sec
3	23:21:31	select * from booking LIMIT 0, 1000	7 row(s) returned	0.016 sec / 0.000 sec

Activate Windows
Go to Settings to activate Windows.

Query | x micro_users booking

1 • select * from transaction

2

Result Grid

transaction_id	booking_id	card_number	payment_mode	upi_id
2	1		UPI	upi details
4	3		CARD	upi details
6	5	1234	CARD	upi details
8	7		UPI	upi details
10	9	1234	CARD	upi details
12	11	1234	CARD	upi details
14	13	123456666	UPI	upi details
NOTE	NOTE	NOTE	NOTE	NOTE

transaction 2 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	23:21:03	select * from transaction LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clicki...	0.250 sec
2	23:21:15	use sweethome	0 row(s) affected	0.000 sec
3	23:21:31	select * from booking LIMIT 0, 1000	7 row(s) returned	0.016 sec / 0.000 sec
4	23:22:53	select * from transaction LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

Activate Windows
Go to Settings to activate Windows.