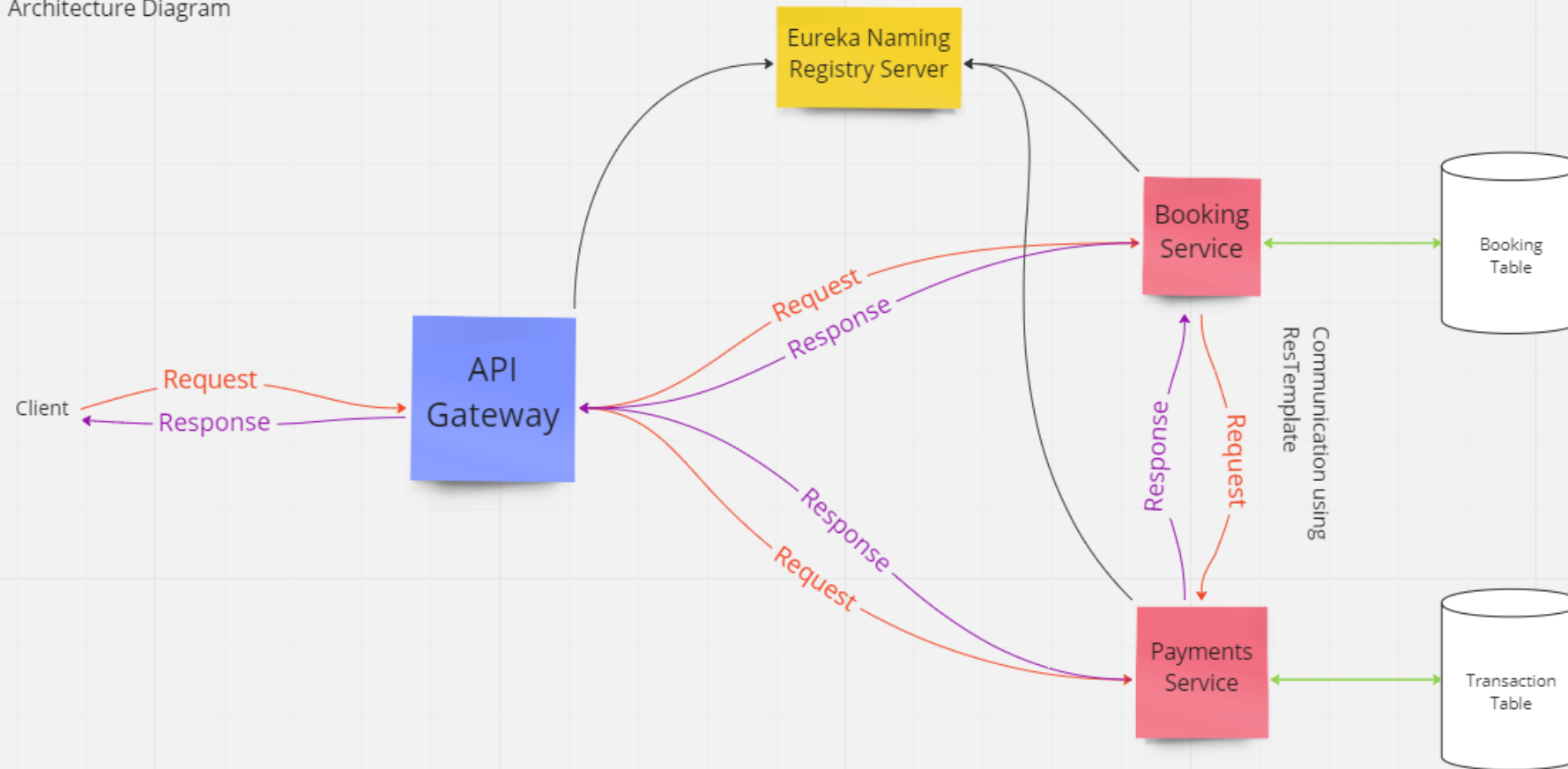


Architecture Diagram



— API Requests —>

— API Response —>

— Naming Registry —>

- DB Communications ->

## DS Replicas

### Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - <a href="http://localhost:api-gateway:9191">localhost:api-gateway:9191</a>
BOOKING-SERVICE	n/a (1)	(1)	UP (1) - <a href="http://localhost:booking-service:8080">localhost:booking-service:8080</a>
PAYMENT-SERVICE	n/a (1)	(1)	UP (1) - <a href="http://localhost:payment-service:8081">localhost:payment-service:8081</a>

API calls through Gateway

### General Info

sweet-home-booking / 3\_PaymentTransaction

POST http://localhost:9191/payment/transaction

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

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

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

Body Cookies Headers (3) Test Results

Status: 201 Created Time: 44 ms Size: 122 B Save Response

Pretty Raw Preview Visualize JSON

1 4

sweet-home-booking / getPaymentTransaction

GET http://localhost:9191/payment/transaction/2

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies Beautify

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

1

Body Cookies Headers (3) Test Results

Status: 200 OK Time: 56 ms Size: 210 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "bookingId": 1,
3   "transactionId": 2,
4   "paymentMode": "UPI",
5   "upiId": "upi_details",
6   "cardNumber": "abc"
7 }
```

## application.yml files for Gateway Settings

Gateway

Booking Service

Payment Service

```
server:
  port: 9191

spring:
  application:
    name: api-gateway

  cloud:
    gateway:
      routes:
        - id: BOOKING-SERVICE
          uri: lb://BOOKING-SERVICE
          predicates:
            - Path=/hotel/**

        - id: PAYMENT-SERVICE
          uri: lb://PAYMENT-SERVICE
          predicates:
            - Path=/payment/**

      discovery:
        enabled: true

eureka:
  client:
    register-with-eureka: true
    fetch-registry: true
    service-url:
      defaultZone: http://localhost:8761/eureka/

instance:
  hostname: localhost
```

http://localhost:9191/hotel/booking

sweet-home-booking / 1\_booking

POST http://localhost:9191/hotel/booking

Params Authorization Headers (8) Body Pre-request Script Test Results

none form-data x-www-form-urlencoded raw binary

```
1 {
2   "fromDate": "2022-11-27",
3   "toDate": "2022-11-29",
4   "aadhaarNumber": "122334252",
5   "numOfRooms": 2
6 }
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "bookingId": 1,
3   "fromDate": "2022-11-27",
4   "toDate": "2022-11-29",
5   "aadhaarNumber": "122334252",
6   "numOfRooms": 2
7 }
```

```
service > booking > Controller > BookingsController > getOrder
ServiceBookingApplication.java x BookingsController.java x application.properties x Booking.java x
17
18 @RestController
19 @Validated
20 @RequestMapping("/hotel")
21 public class BookingsController {
22     1 usage
23     Logger logger = LoggerFactory.getLogger(BookingsController.class);
24     6 usages
25     BookingsService bookingsService;
26
27     gitcrumbs
28     public BookingsController(BookingsService bookingsService) { this.bookingsService = bookingsService; }
29
30     gitcrumbs
31     @GetMapping("/status")
32     public ResponseEntity getOrder() {
33         return ResponseEntity.ok(bookingsService.getAllBookings());
34     }
35
36     gitcrumbs
37     @PostMapping("/booking")
38     public ResponseEntity<Booking> createBooking(@RequestBody Booking trxVo) {
39
40         return new ResponseEntity<>(bookingsService.createBooking(trxVo), HttpStatus.CREATED);
41     }
42
43     gitcrumbs *
44     @PostMapping("/booking/{id}/transaction")
45     public ResponseEntity getBookingStatus(@Valid @RequestBody Transaction trxVo, @PathVariable Integer id) {
46
47         Booking bookingentry = bookingsService.getBookingStatus(id).orElseThrow(() -> new RecordNotFoundException("Booking not found for the given id "+id));
48         Booking bookingentryservice = new Booking();
49         if(bookingentry.getBookingId()!=null){
50             System.out.println("Booking id found "+bookingentry.getBookingId() );
51             bookingentryservice = bookingsService.createTrx(trxVo, id,bookingentry);
52         }
53
54         logger.info("Booking confirmed for user with aadhaar number: "
55             + bookingsService.getAadhaarNumber());
56     }
57 }
```

```

33 }
34
35 /* The requested post call for booking brings the code flow here
36 * and after formatting the data, the entry is saved in the booking repository*/
37 1 usage gitcrumbs*
38 public Booking createBooking(Booking booking) throws Exception {
39     Booking bookedItem = bookingRepository.save(formatDataEntry(booking));
40     return bookedItem;
41 }
42
43 1 usage gitcrumbs
44 @ public Booking createTrx(Transaction trxVo, Integer id, Booking bookingentry){ //
45     trxVo.setBookingId(id);
46
47     Integer item= restTemplate.postForObject(transactionAppUrl, trxVo, Integer.class);
48     System.out.println("Generated Transaction id is"+item);
49     bookingentry.setTransactionId(item);
50
51     bookingRepository.save(bookingentry);
52
53     return bookingentry;
54 }

```

```

77 /* This code block formats the data as per the given format in
78 the problem statement*/
79 1 usage gitcrumbs*
80 @ public Booking formatDataEntry(Booking booking) throws Exception{
81     int numofRooms = booking.getNumofRooms();
82     String s= getRandomNumbers(numofRooms).toString();
83     String requiredString = s.substring(s.indexOf("[")+1, s.indexOf("]"));
84     booking.setRoomNumbers(requiredString);
85     try{
86         //Setting the booking date to current date
87         System.out.println("The LocalDateTime format is : "+LocalDateTime.now());
88         booking.setBookedOn(java.sql.Timestamp.valueOf(LocalDateTime.now()));
89
90         //Calculate differences between date and calculate the room price
91         Date firstDate = booking.getFromDate();
92         Date secondDate = booking.getToDate();
93         long diffInMillies = Math.abs(secondDate.getTime() - firstDate.getTime());
94         long daysBetween = TimeUnit.DAYS.convert(diffInMillies, TimeUnit.MILLISECONDS);
95         long roomPrice = 1000 * numofRooms*(daysBetween);
96
97         booking.setFromDate(booking.getFromDate());
98         booking.setToDate(booking.getToDate());
99         booking.setRoomPrice((int)roomPrice);
100     }catch(Exception e){
101         throw new Exception("Exception in Data Manipulations"+e.getMessage());

```

Activate Windows

## Code Flow: Booking Service

```
service > booking > entities > Booking > fromDate
Booking.java x
12 @Id
13 @GeneratedValue(strategy = GenerationType.AUTO)
14 private Integer bookingId;
15 @DateTimeFormat(pattern = "yyyy-MM-dd")
16 @NotNull(message = "Please provide a From date.")
17 private Date fromDate;
18 @DateTimeFormat(pattern = "yyyy-MM-dd")
19 @NotNull(message = "Please provide a To date.")
20 private Date toDate;
21 private String aadharNumber;
22 private Integer numOfRooms;
23 @Column(nullable = false)
24 private String roomNumbers;
25 private int roomPrice;
26 @GeneratedValue(strategy = GenerationType.AUTO)
27 private int transactionId;
28 private Date bookedOn;
29 public Integer getBookingId() { return bookingId; }
```

Booking Entity ↔ Schema

Service	Table Name	Columns	Datatypes	Constraints
Booking Service	booking	bookingId	int (auto generated)	PRIMARY KEY
		fromDate	Date	NULL
		toDate	Date	NULL
		aadharNumber	String	NULL
		numOfRooms	int	
		roomNumbers	String	
		roomPrice	int	NOT NULL
		transactionId	int	DEFAULT=0
		bookedOn	date	NULL

```

Transaction.java x
1 package com.example.payments.entities;
2
3 import com.sun.istack.NotNull;
4
5 import javax.persistence.*;
6 import javax.validation.constraints.NotBlank;
7
8 @Entity
9 @Table(name = "Transaction")
10 public class Transaction {
11     @Column(nullable = false)
12     private int bookingId;
13     @Id
14     @GeneratedValue(strategy = GenerationType.AUTO)
15     private int transactionId;
16     private String paymentMode;
17     private String upiId;
18     @NotBlank(message = "Invalid mode of payment")
19     private String cardNumber;
20
21     public int getTransactionId() { return transactionId; }

```

Transaction Entity  $\leftrightarrow$  Schema

Service	Table Name	Columns	Datatypes	Constraints	
Payment Service	transaction	transactionId	int( auto generated)	PRIMARY KEY	It re
		paymentMode	String		It re
		booking Id	int	NOT NULL	It re is c;
		upi Id	String	NULL	If th
		cardNumber	String	NULL	If th

```
gitcrumbs
public BookingsController(BookingsService bookingsService) { this.bookingsService = bookingsService; }

gitcrumbs
@GetMapping("/status")
public ResponseEntity getOrder() {

    return ResponseEntity.ok(bookingsService.getAllBookings());
}

/* The post mapping call with the booking endpoint passes the booking object as payload
to the createBooking method defined in the BookingService layer and it in returns the
generates the booking entry in the database .
* */
gitcrumbs *
@PostMapping("/booking")
public ResponseEntity<Booking> createBooking(@RequestBody Booking trxVo) throws Exception {

    return new ResponseEntity<>(bookingsService.createBooking(trxVo), HttpStatus.CREATED);
}

/* The post mapping call with the booking id to the transaction service
```

Schema Maps

POST http://localhost:9191/hotel/booking

Params Authorization Headers (8) Body Pre-request Script Tests Settings

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

1

Body Cookies Headers (3) Test Results

Status: 201 Created Time: 1878 ms Size: 361 B

Pretty Raw Preview Visualize JSON

1 "bookingId": 1,  
2 "fromDate": "2022-11-27T00:00:00.000+05:30",  
3 "toDate": "2022-11-29T00:00:00.000+05:30",  
4 "aadharNumber": "122334252",  
5 "numOfRooms": 2,  
6 "roomNumbers": "35, 93",  
7 "roomPrice": 4000,  
8 "transactionId": 0,  
9 "bookedOn": "2022-11-27T19:55:35.2906605"  
10  
11



3 usages

```
@NotBlank(message = "Invalid mode of payment")
private String cardNumber;
```

```
1 spring.application.name=booking-service
2 spring.jpa.hibernate.ddl-auto=create
3 spring.datasource.initialization-mode=always
4 spring.datasource.platform=postgres
5 spring.datasource.url=jdbc:postgresql://localhost:5432/myDB
6 spring.datasource.username=postgres
7 spring.datasource.password=postgres
8 spring.jpa.properties.hibernate.jdbc.lob.non_contextual_creation=true
9
10
11 transactionApp.url=http://localhost:8081/payment/transaction
```

```
@Value("${transactionApp.url}")
private String transactionAppUrl;
```

application.properties

```
/* This method makes an internal call to the payments service,
 * when the user requests the call on booking/{id}/transaction
 * The call is made using the rest template and the parameters are
 * injected from the constants file application.properties
 * The url is injected with the @Value("${transactionApp.url}") annotation
 * */
1 usage gitcrumbs
public Booking createTrx(Transaction trxVo, Integer id, Booking bookingentry){
    trxVo.setBookingId(id);

    Integer item= restTemplate.postForObject(transactionAppUrl, trxVo, Integer.class);

    bookingentry.setTransactionId(item);

    bookingRepository.save(bookingentry);

    return bookingentry;
```

sweet-home-booking / 2\_booking\_Transaction

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

Params Authorization Headers (8) Body Pre-request Script Tests Settings

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

```
1
2 ... "paymentMode": "UPI",
3 ... "bookingId": 1,
4 ... "upiId": "upi details",
5 ... "cardNumber": ""
6
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize JSON

```
1
2 "message": "Invalid mode of payment",
3 "statusCode": 400
4
```

Status: 400 Bad Request Time: 158 ms Size: 179 B Save Response

POST http://localhost:9191/hotel/booking/2/transaction

Params Authorization Headers (8) Body Pre-request Script Tests Settings

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

```
1
2 ... "paymentMode": "UPI",
3 ... "bookingId": 2,
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize JSON

```
1
2 "bookingId": 2,
3 "fromDate": "2022-11-27T00:00:00.000+05:30",
4 "toDate": "2022-11-29T00:00:00.000+05:30",
5 "aadharNumber": "122334252",
6 "numOfRooms": 2,
7 "roomNumbers": "79, 74",
8 "roomPrice": 4000,
9 "transactionId": 3,
10 "bookedOn": "2022-11-27T22:24:22.232142"
11
```

Status: 200 OK Time: 590 ms Size: 355 B

\*Booking Service -> Payments Service

```
booking > Controller > BookingsController > getBookingStatus

ServiceBookingApplication.java x BookingsController.java x BookingsService.java x application.properties x Booking.java x T

if(bookingEntry.getBookingId()!=null){
    System.out.println("Booking id found "+bookingEntry.getBookingId() );
    bookingEntry = bookingsService.createTrx(trxVo, id,bookingEntry);
}

logger.info("Booking confirmed for user with aadhaar number: "
    + bookingEntry.getAadhaarNumber()
    + " | "
    + "Here are the booking details:  "+ bookingEntry.toString());
return ResponseEntity.ok(bookingEntry);
}
```

```
application.yml
> test
> target
.gitignore
mvnw
mvnw.cmd
pom.xml
service-booking.iml
External Libraries
Scratches and Consoles

74
75
76
77
78
79

return response.orElseThrow(()-> new RecordNotFoundException("Booking not generated for the id :"+id));
}

Run: ServiceBookingApplication x
: Completed initialization in 3 ms
., booking0_.booked_on as booked_o3_0_0_, booking0_.from_date as from_dat4_0_0_, booking0_.num_of_rooms as num_of_r5_0_0_, booking0_.room_numbers as room_num6_0_0_, booking0_.room_price as room_pri7_0_0_, booking0_.to_date as :
., booking0_.booked_on as booked_o3_0_0_, booking0_.from_date as from_dat4_0_0_, booking0_.num_of_rooms as num_of_r5_0_0_, booking0_.room_numbers as room_num6_0_0_, booking0_.room_price as room_pri7_0_0_, booking0_.to_date as :
om_price, to_date, transaction_id, booking_id) values (?, ?, ?, ?, ?, ?, ?, ?)
., booking0_.booked_on as booked_o3_0_0_, booking0_.from_date as from_dat4_0_0_, booking0_.num_of_rooms as num_of_r5_0_0_, booking0_.room_numbers as room_num6_0_0_, booking0_.room_price as room_pri7_0_0_, booking0_.to_date as :
., booking0_.booked_on as booked_o3_0_0_, booking0_.from_date as from_dat4_0_0_, booking0_.num_of_rooms as num_of_r5_0_0_, booking0_.room_numbers as room_num6_0_0_, booking0_.room_price as room_pri7_0_0_, booking0_.to_date as :
rs=?, room_price=?, to_date=?, transaction_id=? where booking_id=?
: Booking confirmed for user with aadhaar number: 122334252 | Here are the booking details: Booking{bookingId=3, fromDate=2022-11-27T00:00:00.000+05:30, toDate=2022-11-29T00:00:00.000+05:30, aadhaarNumber='122334252',

Git Run TODO Problems Terminal Services Build Dependencies
Build completed successfully in 1 min 17 sec (3 minutes ago)
79:1 CRIF UTF-8 4 spaces Implementation Fixes
```

Project

src

main

java

com

example

service

booking

service

BookingsService

createTx

BookingsController.java

LoggingAspect.java

BookingNotFoundException.java

ErrorResponse.java

CustomExceptionHandler.java

BookingsService.java

Query

Query History

Scratch Pad

service-booking

D:\Upgrad\Projects\Sweet-home\service-booking

src

main

java

com.example.service.booking

aspect

LoggingAspect

Controller

Dao

entities

Booking

Transaction

exceptions

BookingNotFoundException

CustomExceptionHandler

ErrorResponse

RecordNotFoundException

repository

service

ServiceBookingApplication

40

}

41

}

42

/\* This method makes an internal call to the payments service,

43

\* when the user requests the call on booking/{id}/transaction

44

\* The call is made using the rest template and the parameters are

45

\* injected from the constants file application.properties

46

\* The url is injected with the @Value("\${transactionApp.url}") annotation

47

\*/

48

@Usage(gitcrumbs = "1 usage")

49

public Booking createTx(Transaction trxVo, Integer id, Booking bookingentry){

50

trxVo.setBookingId(id);

51

52

Integer item= restTemplate.postForObject(transactionAppUrl, trxVo, Integer.class);

53

bookingentry.setTransactionId(item);

54

55

bookingRepository.save(bookingentry);

56

57

return bookingentry;

58

}

POST

http://localhost:9191/hotel/booking/4/transaction

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

2

3

4

5

6

body

Cookies

Headers (3)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

Activate Wir

Go to Settings to

1

2

3

select \* from booking

Data output

Messages

Notifications

booking\_id

aadhar\_number

booked\_on

from\_date

num\_of\_rooms

room\_numbers

room\_price

to\_date

transaction\_id

[PK] integer

character varying (255)

timestamp without time zone

character varying (255)

integer

character varying (255)

integer

character varying (255)

integer

1

2

4

454654874674

122334252

122334252

2022-11-27 22:59:24.604248

2022-11-27 23:00:08.488001

2022-11-27 23:00:24.323422

2022-11-27T22:59:24.6...

2022-11-27T00:00:00.0...

2022-11-27T00:00:00.0...

[null]

2

2

502

58, 85

28, 18

2000

4000

4000

2022-11-27T22:59:24.6...

2022-11-29T00:00:00.0...

2022-11-29T00:00:00.0...

1234

0

0

Query

Query History

Scratch Pad

1

2

3

select \* from booking

Data output

Messages

Notifications

Lid

aadhar\_number

booked\_on

from\_date

num\_of\_rooms

room\_numbers

room\_price

to\_date

transaction\_id

integer

character varying (255)

timestamp without time zone

character varying (255)

integer

character varying (255)

integer

character varying (255)

integer

1

2

4

454654874674

122334252

122334252

2022-11-27 22:59:24.604248

2022-11-27 23:00:08.488001

2022-11-27 23:00:24.323422

2022-11-27T22:59:24.604248...

2022-11-27T00:00:00.000+05:30...

2022-11-27T00:00:00.000+05:30...

[null]

2

2

502

58, 85

28, 18

2000

4000

4000

2022-11-27T22:59:24.604248...

2022-11-29T00:00:00.000+05:30...

2022-11-29T00:00:00.000+05:30...

1234

0

6

Bookings table before  
Transaction creation

Bookings table after  
Transaction creation,  
transaction id is updated

\*Booking Service -> Payments Service

http://localhost:9191/payment/transaction

```
36
37 @PostMapping("/transaction")
38 public ResponseEntity<Integer> createTransaction(@Valid @RequestBody Transaction trxVo){
39
40     Integer created = trxService.createTransaction(trxVo);
41     return new ResponseEntity<>(created, HttpStatus.CREATED);
42
43
44 }
```

Working locally in Scratch Pad. Switch to a Workspace

New Import

GET getPaymentTransaction POST 2\_booking\_Transaction POST 1\_booking POST 3\_PaymentTransaction

...

sweet-home-booking / 3\_PaymentTransaction

Save

POST http://localhost:9191/payment/transaction ← Payments Service

Params Authorization Headers (8) Body Pre-request Script Tests Settings

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

1 {  
2 "paymentMode": "UPI",  
3 "bookingId": 4,  
4 "upiId": "upi details for bookingId 4",  
5 "cardNumber": "abc"  
6 }

Body Cookies Headers (3) Test Results

● Status: 201 Created Time: 429 ms Size:

Pretty Raw Preview Visualize JSON

1 5

pgAdmin 4

myDB/postgres@PostgreSQL 14\*

Query

```
select * from transaction
```

Data output

transaction_id [PK] integer	booking_id integer	card_number character varying (255)	payment_mode character varying (255)	upi_id character varying (255)
1	1	1234567	Card	12345@upi

pgAdmin 4

myDB/postgres@PostgreSQL 14\*

Query

```
select * from transaction
```

Data output

transaction_id [PK] integer	booking_id integer	card_number character varying (255)	payment_mode character varying (255)	upi_id character varying (255)
1	1	1234567	Card	12345@upi
2	5	abc	UPI	upi details for bookingId 4

Import

GET getPaymentTransaction

POST 2\_booking\_Transaction

POST 1\_booking

POST 3\_PaymentTransac

No Environr

sweet-home-booking / getPaymentTransaction

GET

http://localhost:9191/payment/transaction/5

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

Body

Cookies

Headers (3)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "bookingId": 4,
3   "transactionId": 5,
4   "paymentMode": "UPI",
5   "upiId": "upi details for bookingId 4",
6   "cardNumber": "abc"
7 }
```

Status: 200 OK Time: 69 ms Size: 226 B

New Import

GET getPaymentTransaction

POST 2\_booking\_Transaction

POST 1\_booking

POST 3\_PaymentTransac

No Environment

sweet-home-booking / getPaymentTransaction

GET

http://localhost:9191/payment/transaction/4

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

Body

Cookies

Headers (3)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "No Transaction Information found for :4"
3 }
```

Status: 400 Bad Request Time: 103 ms Size: 168 B Save Response

pgAdmin 4

pgAdmin

File Object Tools Help

Dashboard Properties SQL Statistics myDB/postgres@PostgreSQL 14\*

myDB/postgres@PostgreSQL 14

Query Query History

```
1 select * from transaction
2
3
```

Data output Messages Notifications

	transaction_id [PK] integer	booking_id integer	card_number character varying (255)	payment_mode character varying (255)	upi_id character varying (255)
1	1	1234	1234567	Card	12345@upi
2	5	4	abc	UPI	upi details for bookingId 4

GET getPaymentTransaction POST 2\_booking\_Transaction POST 1\_booking POST 3\_PaymentTransaction + ... No Environment

sweet-home-booking / 2\_booking\_Transaction Save

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

Params Authorization Headers (8) Body Pre-request Script Tests Settings

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

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

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Invalid mode of payment",
3   "statusCode": 400
4 }
```

Status: 400 Bad Request Time: 158 ms Size: 179 B Save

Activate Windows  
Go to Settings to activate Windows.



```

^1.137 INFO 23264 --- [nio-8081-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
^1.141 INFO 23264 --- [nio-8081-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 4 ms
^1.303 INFO 23264 --- [nio-8081-exec-1] c.example.payments.aspect.LoggingAspect : In TransactionController entering createTransaction
^1.339 INFO 23264 --- [nio-8081-exec-1] c.example.payments.aspect.LoggingAspect : In TransactionService entering createTransaction
^1.368 INFO 23264 --- [nio-8081-exec-1] c.example.payments.aspect.LoggingAspect : In CrudRepository entering save
nextval ('hibernate_sequence')
into transaction (booking_id, card_number, payment_mode, upi_id, transaction_id) values (?, ?, ?, ?, ?)
^1.472 INFO 23264 --- [nio-8081-exec-1] c.example.payments.aspect.LoggingAspect : In CrudRepository exiting save
^1.472 INFO 23264 --- [nio-8081-exec-1] c.example.payments.aspect.LoggingAspect : In TransactionService exiting createTransaction
^1.474 INFO 23264 --- [nio-8081-exec-1] c.example.payments.aspect.LoggingAspect : In TransactionController exiting createTransaction
^1.374 INFO 23264 --- [nio-8081-exec-3] c.example.payments.aspect.LoggingAspect : In TransactionController entering createTransaction
^1.375 INFO 23264 --- [nio-8081-exec-3] c.example.payments.aspect.LoggingAspect : In TransactionService entering createTransaction
^1.376 INFO 23264 --- [nio-8081-exec-3] c.example.payments.aspect.LoggingAspect : In CrudRepository entering save
nextval ('hibernate_sequence')
into transaction (booking_id, card_number, payment_mode, upi_id, transaction_id) values (?, ?, ?, ?, ?)
^1.383 INFO 23264 --- [nio-8081-exec-3] c.example.payments.aspect.LoggingAspect : In CrudRepository exiting save
^1.383 INFO 23264 --- [nio-8081-exec-3] c.example.payments.aspect.LoggingAspect : In TransactionService exiting createTransaction
^1.383 INFO 23264 --- [nio-8081-exec-3] c.example.payments.aspect.LoggingAspect : In TransactionController exiting createTransaction

```

```

>-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
>-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
>-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 3 ms
>-8080-exec-1] c.e.s.booking.aspect.LoggingAspect : In BookingsController entering createBooking
>-8080-exec-1] c.e.s.booking.aspect.LoggingAspect : In BookingsService entering createBooking
>-8080-exec-1] c.e.s.booking.aspect.LoggingAspect : In CrudRepository entering save

```

```

nce')
er, booked_on, from_date, num_of_rooms, room_numbers, room_price, to_date, transaction_id, booking_id) values (?, ?, ?, ?, ?, ?, ?, ?, ?)
-8080-exec-1] c.e.s.booking.aspect.LoggingAspect : In CrudRepository existing save
-8080-exec-1] c.e.s.booking.aspect.LoggingAspect : In BookingsService existing createBooking
-8080-exec-1] c.e.s.booking.aspect.LoggingAspect : In BookingsController existing createBooking
-8080-exec-2] c.e.s.booking.aspect.LoggingAspect : In BookingsController entering createBooking
-8080-exec-2] c.e.s.booking.aspect.LoggingAspect : In BookingsService entering createBooking
-8080-exec-2] c.e.s.booking.aspect.LoggingAspect : In CrudRepository entering save

```

```

nce')
er, booked_on, from_date, num_of_rooms, room_numbers, room_price, to_date, transaction_id, booking_id) values (?, ?, ?, ?, ?, ?, ?, ?, ?)
-8080-exec-2] c.e.s.booking.aspect.LoggingAspect : In CrudRepository existing save
-8080-exec-2] c.e.s.booking.aspect.LoggingAspect : In BookingsService existing createBooking
-8080-exec-2] c.e.s.booking.aspect.LoggingAspect : In BookingsController existing createBooking

```

```

import org.springframework.stereotype.Component;

10
11 @Aspect
12 @Component
13 public class LoggingAspect {
14     2 usages
15     Logger logger = LoggerFactory.getLogger(LoggingAspect.class);
16     = gitcrumbs*
17     @Around("execution(* com.example.payments..*(..))")
18     public Object applyLogging(ProceedingJoinPoint joinPoint) throws Throwable{
19         MethodSignature signature = (MethodSignature) joinPoint.getSignature();
20         String className = signature.getDeclaringType().getSimpleName();
21         String methodName = signature.getName();
22
23         logger.info("In "+className+" entering "+methodName);
24         Object result= joinPoint.proceed();
25     }
26 }

```

```

import org.aspectj.lang.reflect.MethodSignature;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Component;

10
11 @Aspect
12 @Component
13 public class LoggingAspect {
14     2 usages
15     Logger logger = LoggerFactory.getLogger(LoggingAspect.class);
16     = gitcrumbs*
17     @Around("execution(* com.example.service.booking..*(..))")
18     public Object applyLogging(ProceedingJoinPoint joinPoint) throws Throwable{
19         MethodSignature signature = (MethodSignature) joinPoint.getSignature();
20         String className = signature.getDeclaringType().getSimpleName();
21         String methodName = signature.getName();
22     }
23 }

```

## AOP Logging Aspects

## Database Entries for Booking and Transactions

Query History

```
1 select * from booking
```

Data output

booking_id [PK] integer	aadhar_number character varying (255)	booked_on timestamp without time zone	from_date character varying (255)	num_of_rooms integer	room_numbers character varying (255)	room_price integer	to_date character varying (255)	transaction_id integer
1	122334252	2022-11-27 19:55:35.290661	2022-11-27T00:00:00.000+05:30	2	35, 93	4000	2022-11-29T00:00:00.000+05:30	2

Total rows: 1 of 1 Query complete 00:00:00.129

Successfully run. Total query runtime: 129 msec. 1 rows affected.

Server myDB/postgres@PostgreSQL 14

Query History

```
1 select * from transaction
```

Data output

transaction_id [PK] integer	booking_id integer	card_number character varying (255)	payment_mode character varying (255)	upi_id character varying (255)	
1	2	1	abc	UPI	upi details

et-home-booking / 2\_booking\_Transaction

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

ns Authorization Headers (8) Body Pre-request Script Tests Settings

one form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ... "paymentMode": "UPI",
3   ... "bookingId": 1,
4   ... "upiId": "upi details",
5   ... "cardNumber": "abc"
}
```

id Cookies Headers (3) Test Results

Status: 200 OK

Pretty Raw Preview Visualize JSON

```
1 {
2   "bookingId": 1,
3   "fromDate": "2022-11-27T00:00:00.000+05:30",
4   "toDate": "2022-11-29T00:00:00.000+05:30",
5   "aadharNumber": "122334252",
6   "numOfRooms": 2,
7   "roomNumbers": "50, 28",
8   "roomPrice": 4000,
9   "transactionId": 2,
10  "bookedOn": "2022-11-27T19:26:09.341474"
11 }
```



getPaymentTransaction POST 2\_booking\_transaction POST 1\_booking POST 3\_PaymentTransaction

sweet-home-booking / 2\_booking\_Transaction

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

Params Authorization Headers (8) Body Pre-request Script Tests Settings

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

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

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "bookingId": 1,
3   "fromDate": "2022-11-27T00:00:00.000+05:30",
4   "toDate": "2022-11-29T00:00:00.000+05:30",
5   "aadharNumber": "122334252",
6   "numOfRooms": 2,
7   "roomNumbers": "50, 28",
8   "roomPrice": 4000,
9   "transactionId": 2,
10  "bookedOn": "2022-11-27T10:36:00.244474"
```

pgAdmin 4

Schema Maps

pgAdmin 4 File Object Tools Help

Dashboard Properties SQL Statistics myDB/postgres@PostgreSQL 14\*

myDB/postgres@PostgreSQL 14

Query Query History

```
1 select * from booking
2
3
```

Data output Messages Notifications

booking_id [PK] integer	aadhar_number character varying (255)	booked_on timestamp without time zone	from_date character varying (255)	num_of_rooms integer	room_numbers character varying (255)	room_price integer	to_date character varying (255)	transaction_id integer
1	1	122334252	2022-11-27 19:55:35.290661	2022-11-27T00:00:00.000+05:...	2	35, 93	4000	2022-11-29T00:00:00.000+05:...

New Import GET getPaymentTransaction POST 2\_booking\_Transaction POST 1\_booking POST 3\_PaymentTransaction

sweet-home-booking / getPaymentTransaction

GET http://localhost:9191/payment/transaction/2

Params Authorization Headers (6) Body Pre-request Script Tests Settings

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

```
1 {
2   "bookingId": 1,
3   "transactionId": 2,
4   "paymentMode": "UPI",
5   "upiId": "upi details",
6   "cardNumber": "abc"
7 }
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "bookingId": 1,
3   "transactionId": 2,
4   "paymentMode": "UPI",
5   "upiId": "upi details",
6   "cardNumber": "abc"
7 }
```

pgAdmin 4

myDB/postgres@PostgreSQL 14

Query Query History

```
1 select * from transaction
2
3
```

Data output Messages Notifications

transaction_id [PK] integer	booking_id integer	card_number character varying (255)	payment_mode character varying (255)	upi_id character varying (255)
1	2	1	abc	UPI

Schema Maps

**Thank You**