

Booking- Service

Database: H2 database(Change the username and password)

EndPoint 1: /hotel/booking

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 {
2   "fromDate": "2021-06-20",
3   "toDate": "2021-08-25",
4   "aadharNumber": "abc",
5   "numOfRooms": 3
6 }
```

Body Cookies Headers (3) Test Results

Status: 201 Created Time: 13 ms Size: 354 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "fromDate": "2021-06-20T00:00:00.000+00:00",
3   "toDate": "2021-08-25T00:00:00.000+00:00",
4   "aadharNumber": "abc",
5   "numOfRooms": 3,
6   "roomNumbers": "51,96,76",
7   "roomPrice": 198000,
8   "transactionId": 0,
9   "bookedOn": "2021-09-12T21:12:30.283+00:00",
10  "id": 2
11 }
```

EndPoint 2: /hotel/booking/{bookingId}/transaction

Case 1:

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

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": "CARD",
3   "bookingId": 3,
4   "upId": "",
5   "cardNumber": "12345"
6 }
```

Body Cookies Headers (3) Test Results

Status: 201 Created Time: 350 ms Size: 297 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "fromDate": "2021-06-20",
3   "toDate": "2021-08-25",
4   "aadharNumber": "abc",
5   "numOfRooms": 3,
6   "roomNumbers": "17,98,89",
7   "roomPrice": 198000,
8   "transactionId": 1,
9   "bookedOn": "2021-09-12",
10  "id": 2
11 }
```

Case 2: Payment Mode = "SOME"

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

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": "SOME",
3   "bookingId": 3,
4   "upiId": "",
5   "cardNumber": "12345"
6 }
```

Body Cookies Headers (3) Test Results

Status: 404 Not Found Time: 18 ms Size: 177 B Save Response

Pretty Raw Preview Visualize JSON

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

Case 3: If no transaction exist for the Booking Id passed to the endpoint.

POST http://localhost:9191/hotel/booking/10/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": "SOME",
3   "bookingId": 3,
4   "upiId": "",
5   "cardNumber": "12345"
6 }
```

Body Cookies Headers (3) Test Results

Status: 404 Not Found Time: 30 ms Size: 174 B Save Response

Pretty Raw Preview Visualize JSON

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

BookServiceImpl:

Methods used in BookServiceImpl:

getRandomNumbers(int noOfRooms):

Generates the random room numbers between 1 to 100.

addBooking(BookingInfoEntity bookingInfoEntity):

Calculates the room price based on numberOfdays and saves the data in the repository.

getPaymentDetails(int bookingId, TransactionDetailsEntity transactionDetailsEntity):

Checks whether the booking Id is available or not, if it's not available, then the method throw "RecordNotFoundException". Also checks the payment mode, if the payment mode is not "UPI" or "CARD" it throws "RecordNotFoundException" and calls makePayment() method from payment-service, the return value of makePayment() is greater than 0, updates the transactionId and save the BookingInfoDetails.

Payment- Service

Database: H2 database

EndPoint 1: /payment/transaction

The screenshot shows a REST client interface with a POST request to `http://localhost:9191/payment/transaction`. The request body is a JSON object with the following fields:

```
1 {
2   "paymentMode": "UPI",
3   "bookingId": 1,
4   "upiId": "",
5   "cardNumber": "1994S"
6 }
```

The response status is 201 Created, with a time of 13 ms and a size of 122 B. The response is saved.

EndPoint 2: /payment/transaction/{transactionId}

Case 1: If transaction Id available then return the details

The screenshot shows a REST client interface with a GET request to `http://localhost:9191/payment/transaction/1`. The response status is 200 OK, with a time of 38 ms and a size of 191 B. The response is saved.

The response body is a JSON object with the following fields:

```
1 {
2   "bookingId": 3,
3   "paymentMode": "CARD",
4   "upiId": "",
5   "cardNumber": "1234S",
6   "id": 1
7 }
```

Case 2: If transaction Id is not available then throws TransactionNotFoundException.

The screenshot shows a REST client interface with a GET request to `http://localhost:9191/payment/transaction/10`. The response status is 404 Not Found, with a time of 146 ms and a size of 178 B. The response body is displayed in JSON format:

```
1 {
2   "message": "Transaction Id Not Found",
3   "statusCode": 400
4 }
```

PaymentServiceImpl:

Methods in the PaymentServiceImpl:

getTransactionId(TransactionDetailsEntity transactionDetailsEntity):

Saves the transactionDetailsEntity and returns the transactionId.

getTransactionDetails(int id):

It checks whether transactionDetailsEntity is present or not using the `findById(int bookingId)` method, if it is present then it returns the transactionDetailsEntity else it throws TransactionNotFoundException.