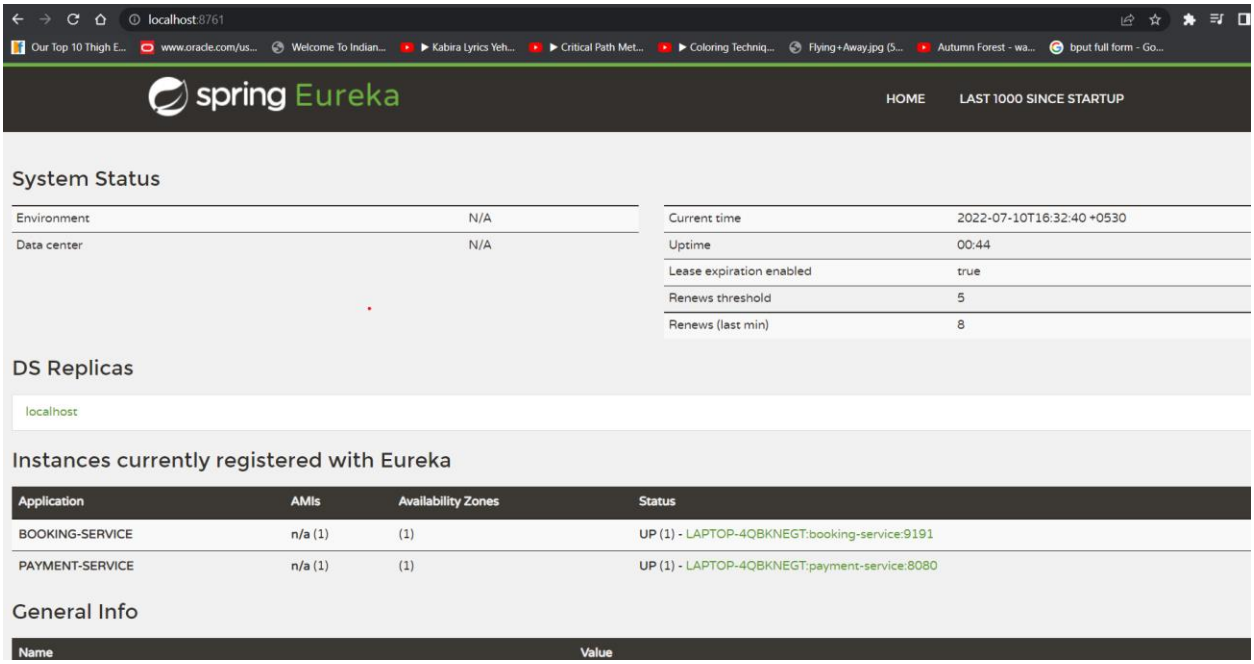
Sweet-Home documentation

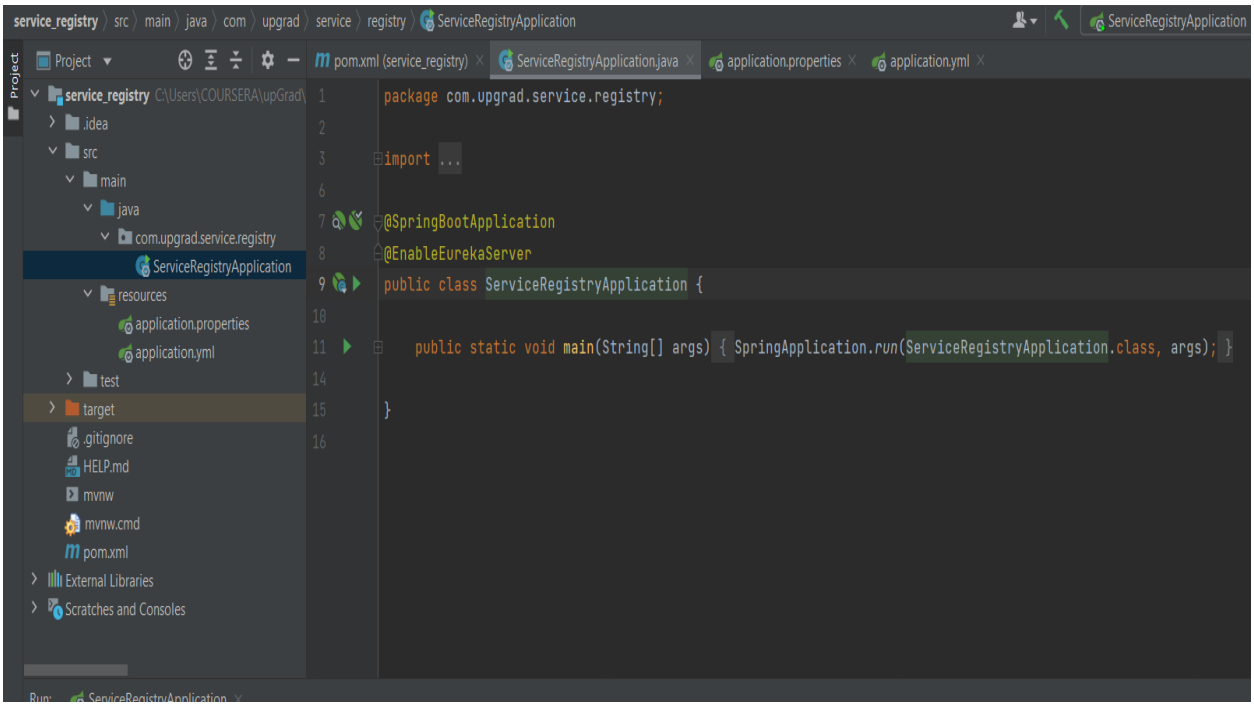
Eureka server and client setup

Booking and Payment service are registered here at port no 8761

http://localhost:8761/

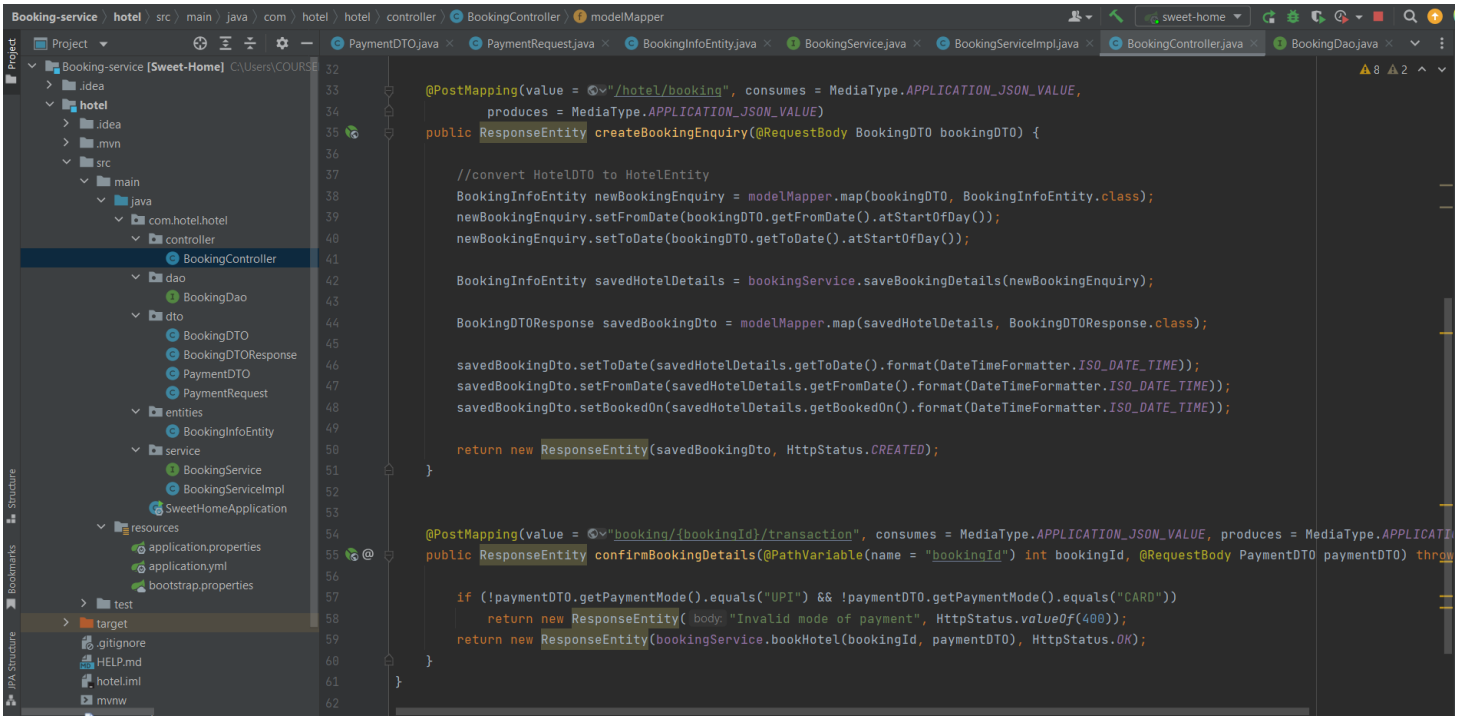


Eureka Service registry project structure



Booking-Service

Project structure



BookingController -

POST http://localhost:9191/hotel/booking url will be called to save data in the database using the method

```
public ResponseEntity createBookingEnquiry(@RequestBody BookingDTO bookingDTO)
```

which has its implementation in the service Class **BookingServiceImpl (**interface used **BookingService)** in the below method.

```
public BookingInfoEntity saveBookingDetails(BookingInfoEntity bookingInfo);
```

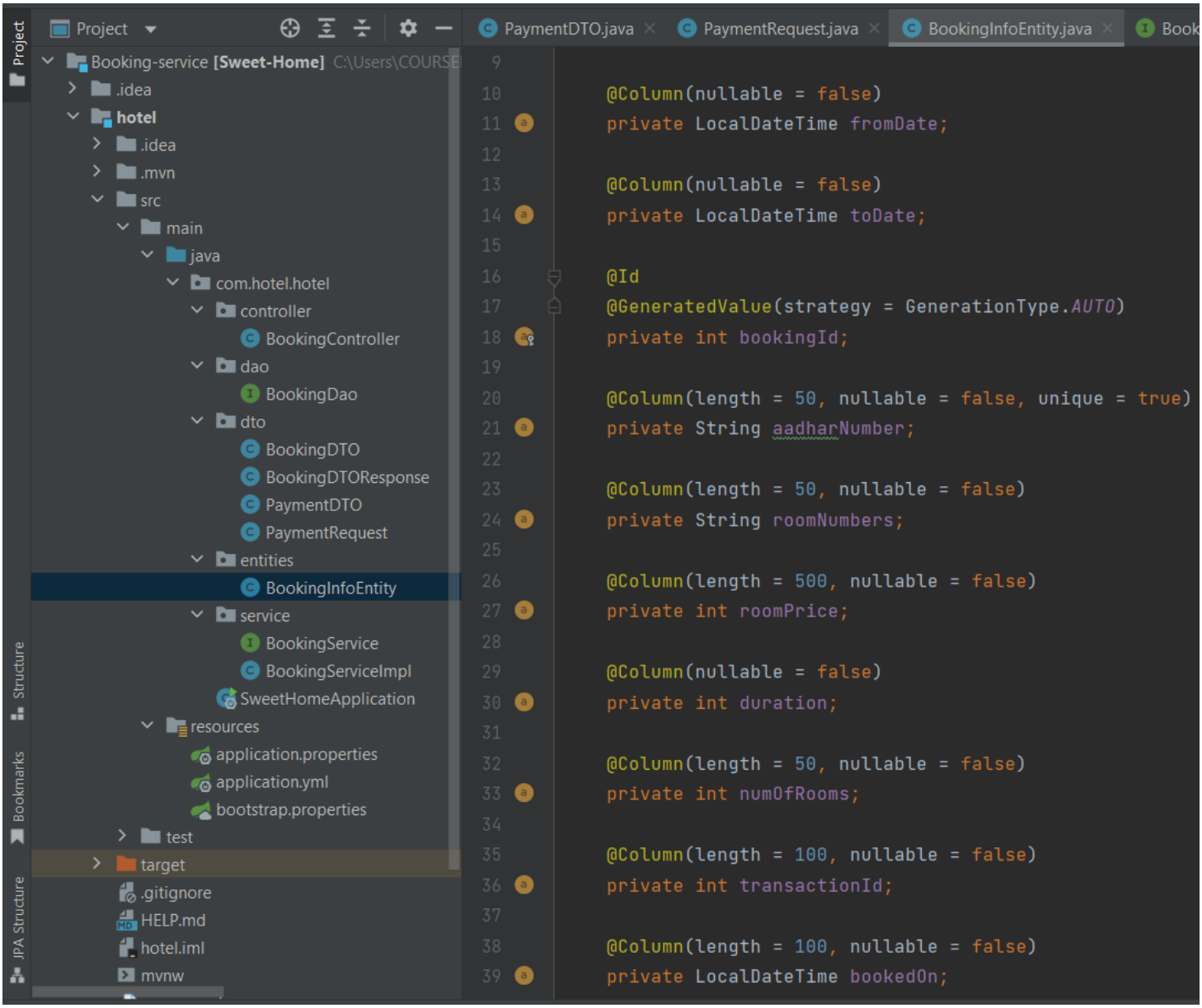
POST http://localhost:9191/booking/{{bookingId}}/transaction API call is made to confirm the booking using the method

```
public ResponseEntity confirmBookingDetails(@PathVariable(name = "bookingId") int
bookingId, @RequestBody PaymentDTO paymentDTO) throws Exception
```

which has its implementation in the service Class **BookingServiceImpl (**interface used **BookingService)** in the below method.

```
public BookingInfoEntity bookHotel(int bookingId, PaymentDTO paymentDTO) throws Exception;
```

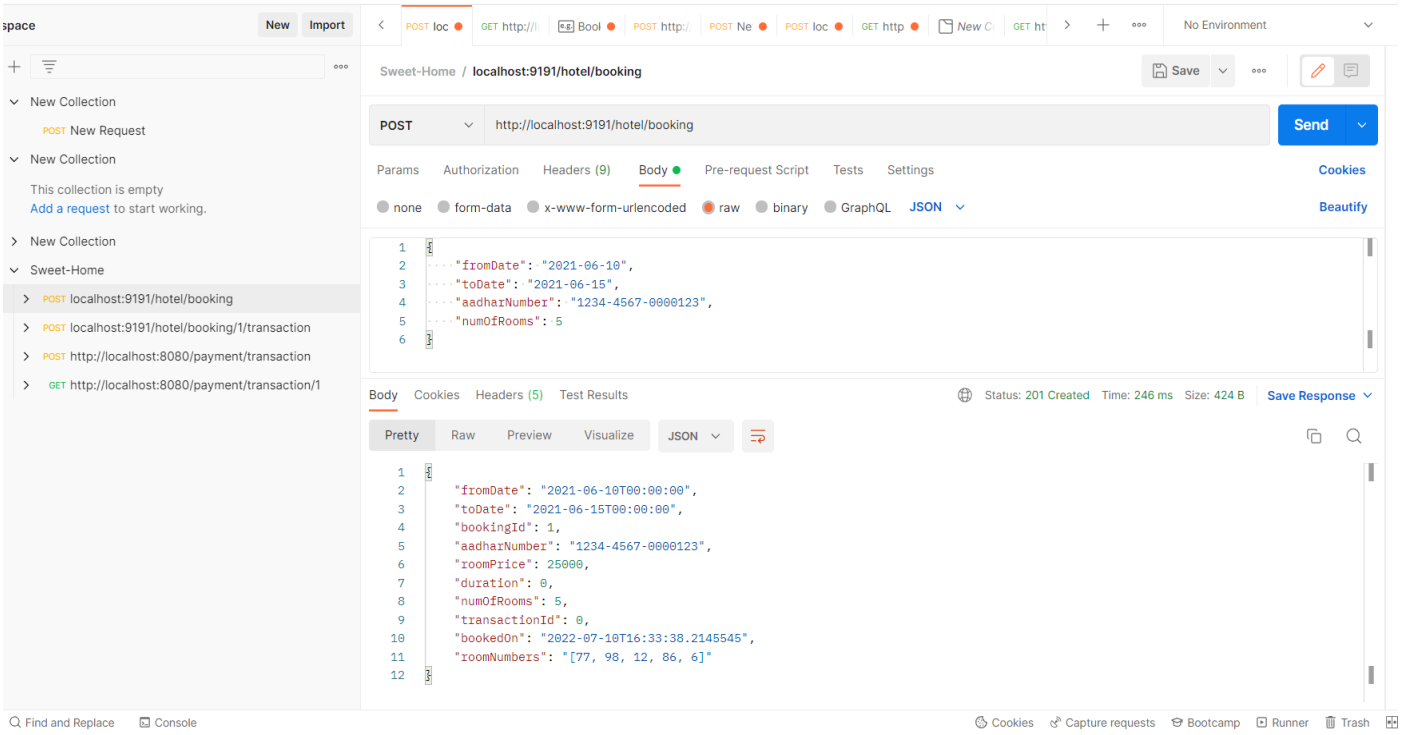
Entity class with the field mapping



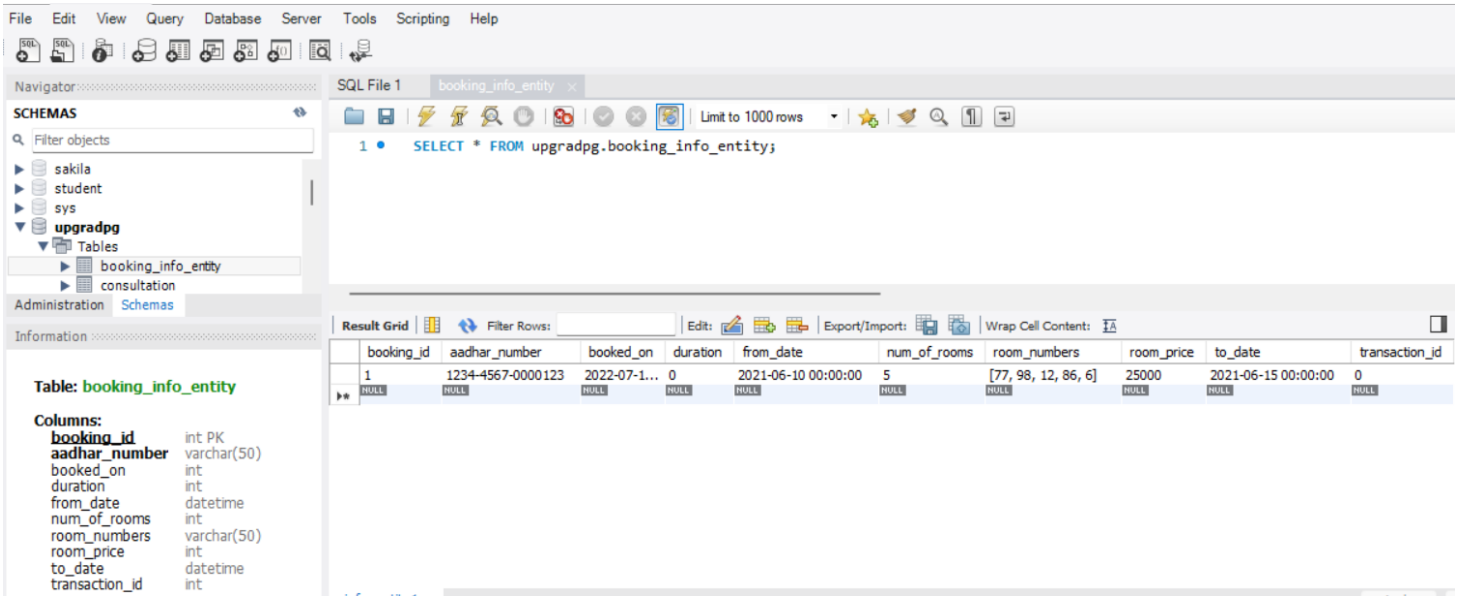
API calls

1. When the user enquires about the booking in the hotel , call done using postman , below is the request and response JSON

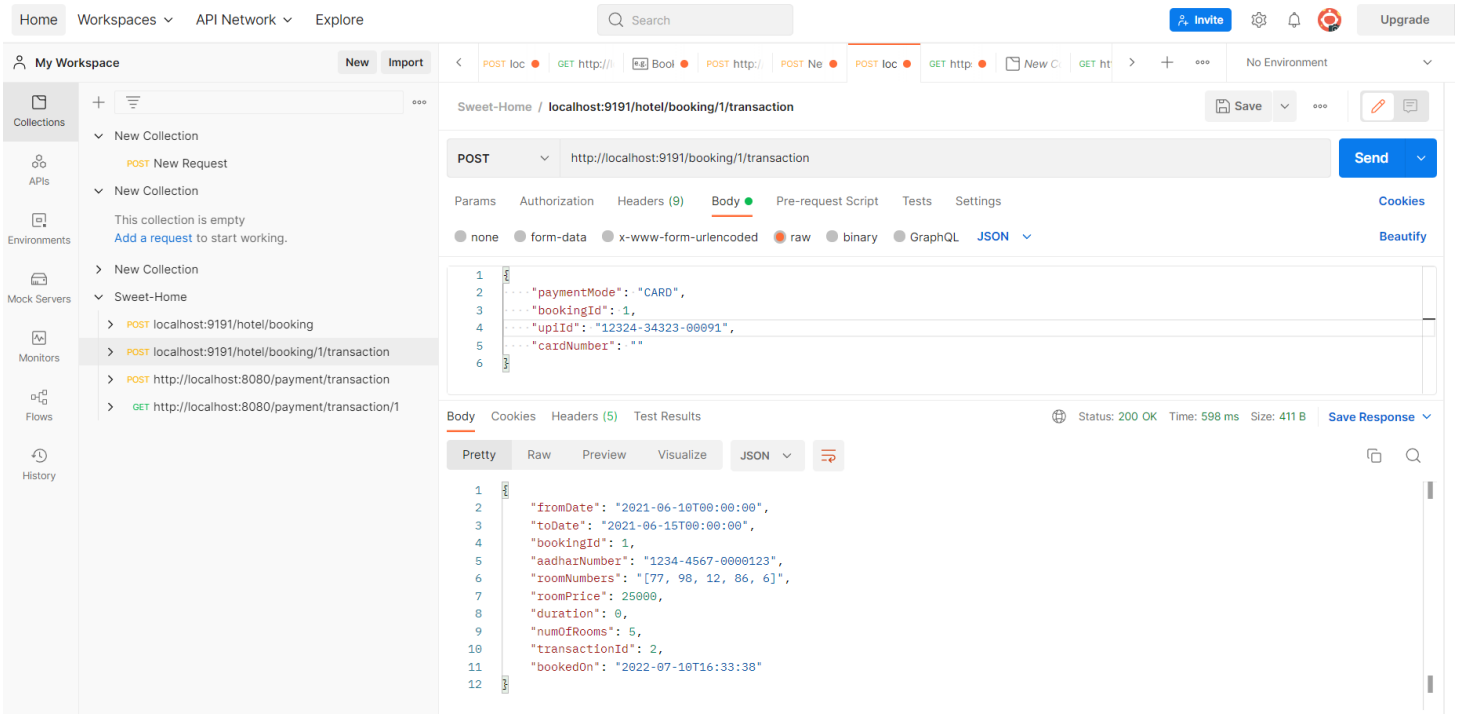
   http://localhost:9191/hotel/booking



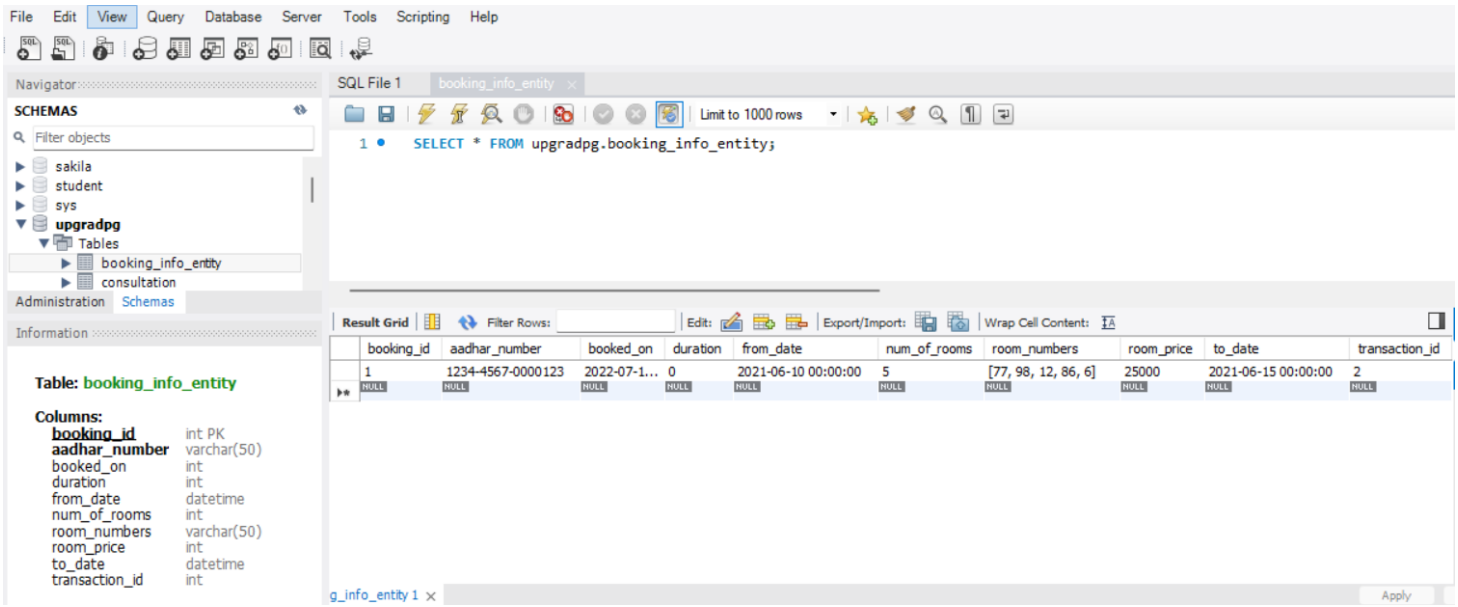
Data saved in the database



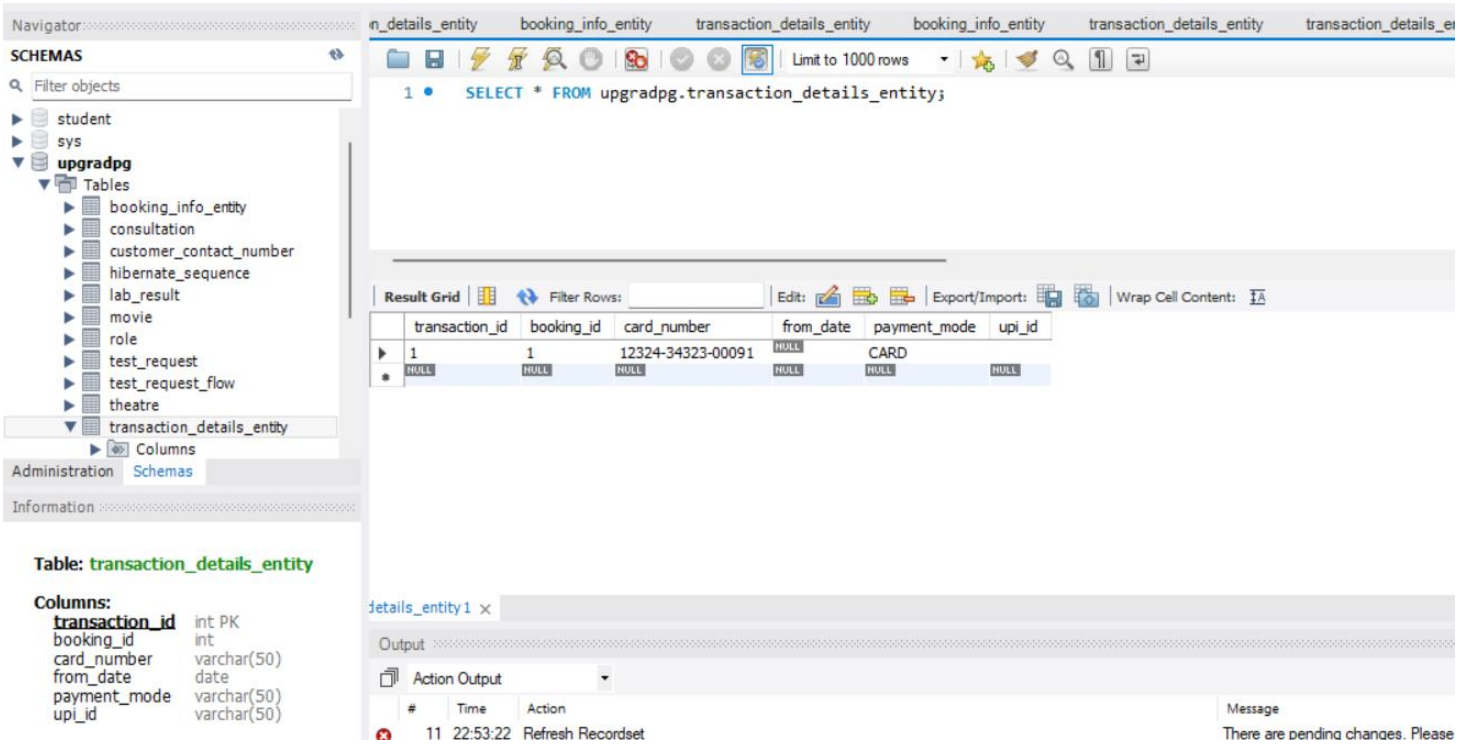
2. Request with the booking id and fetch details from the payment service and save it in the booking table and display it in the response.
   http://localhost:9191/booking/1/transaction



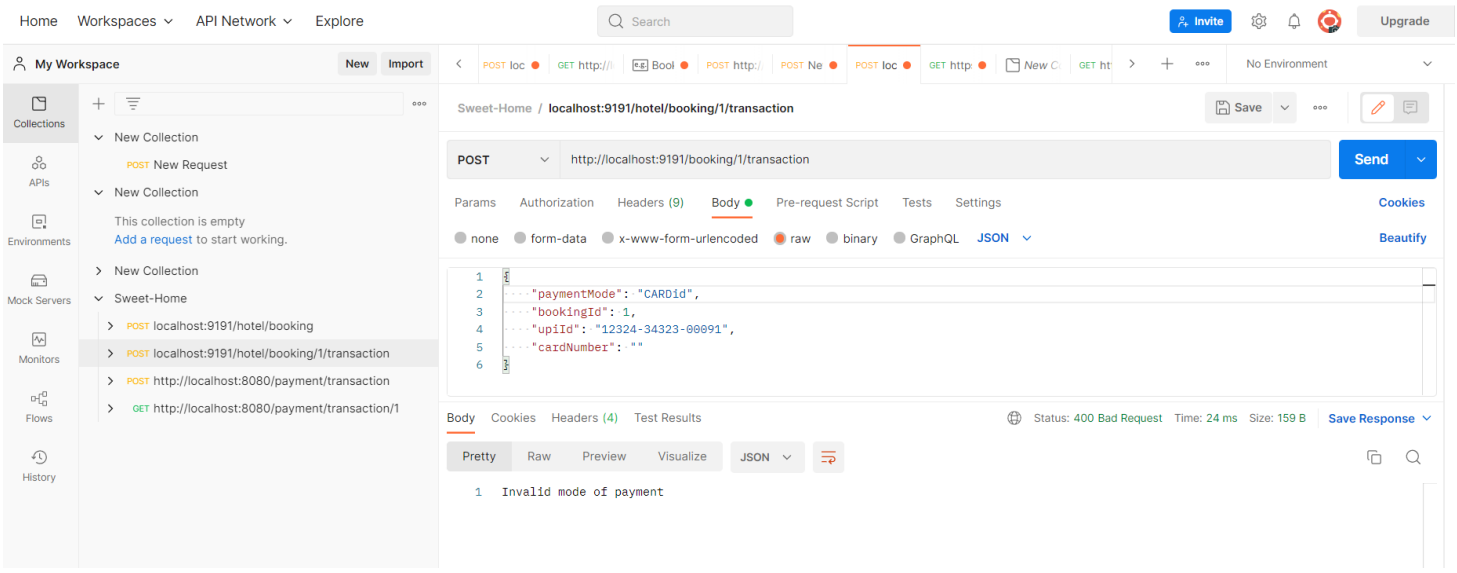
Record updated the booking table



Record updated in the payment table

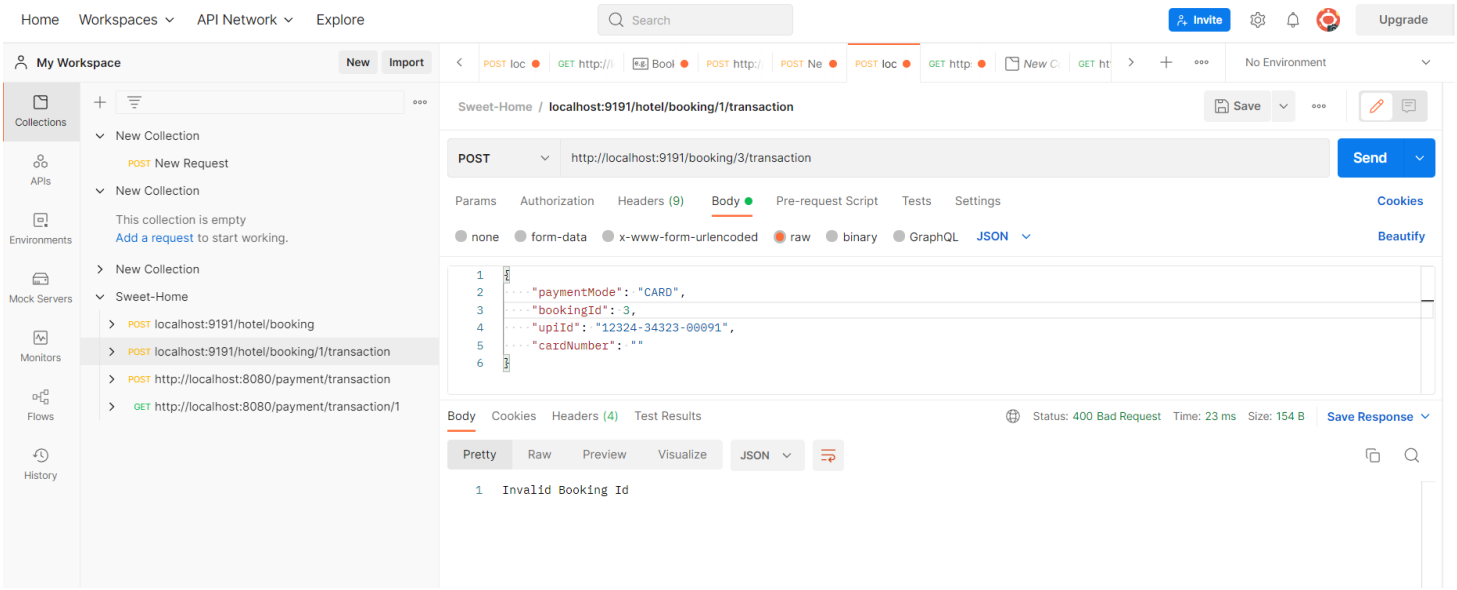


If the payment mode is not UPI or CARD , it throws bad request 400



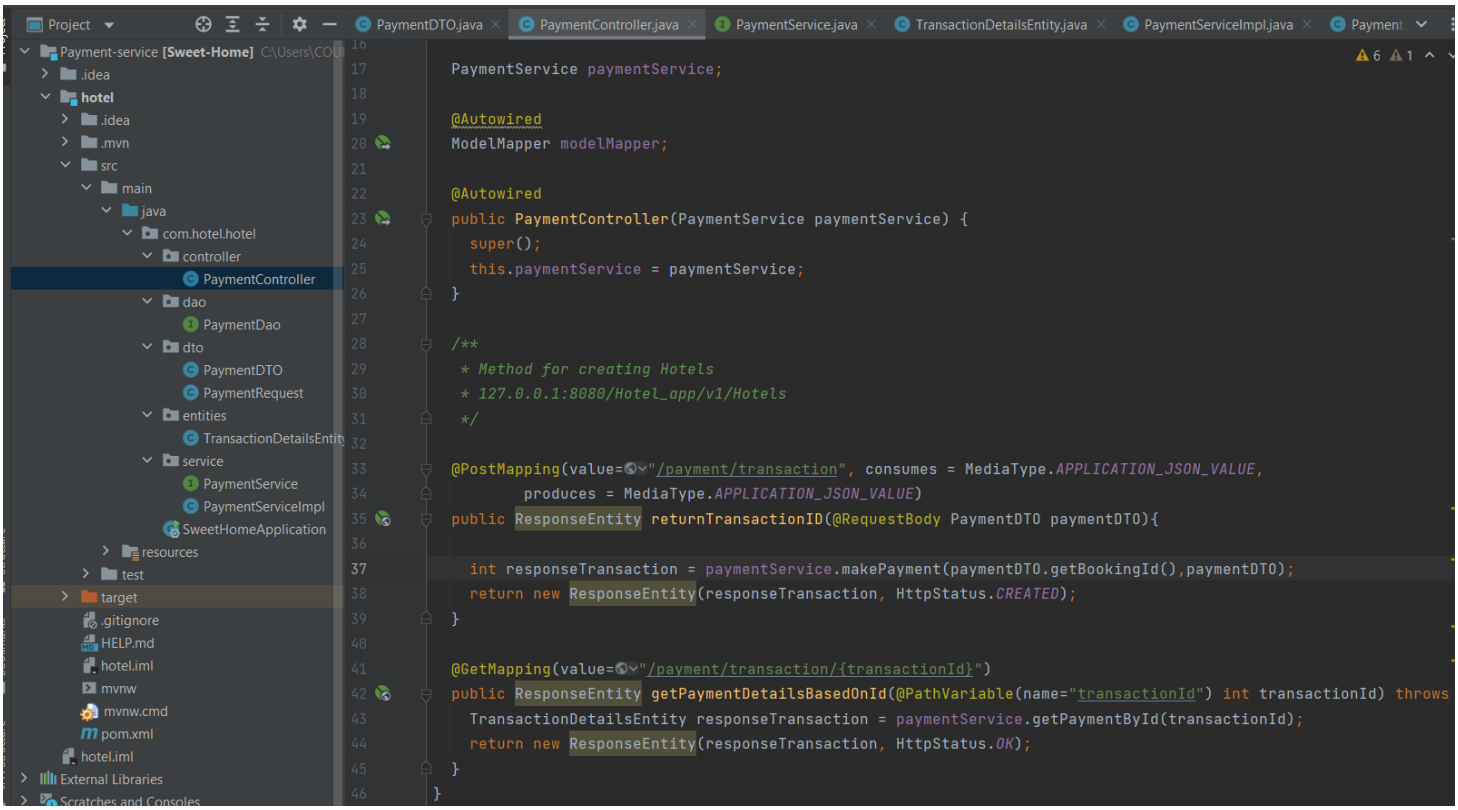
If the booking id is not present in the database , it throws bad request 400

http://localhost:9191/booking/3/transaction



Payment-Service

Project structure



BookingController -

POST http://localhost:8080/payment/transaction url will be called to save data in the database using the method

```
public ResponseEntity returnTransactionID(@RequestBody PaymentDTO paymentDTO)
```

which has its implementation in the service Class **PaymentServiceImpl (**interface used **PaymentService)** in the below method.

```
public BookingInfoEntity saveBookingDetails(BookingInfoEntity bookingInfo);
```

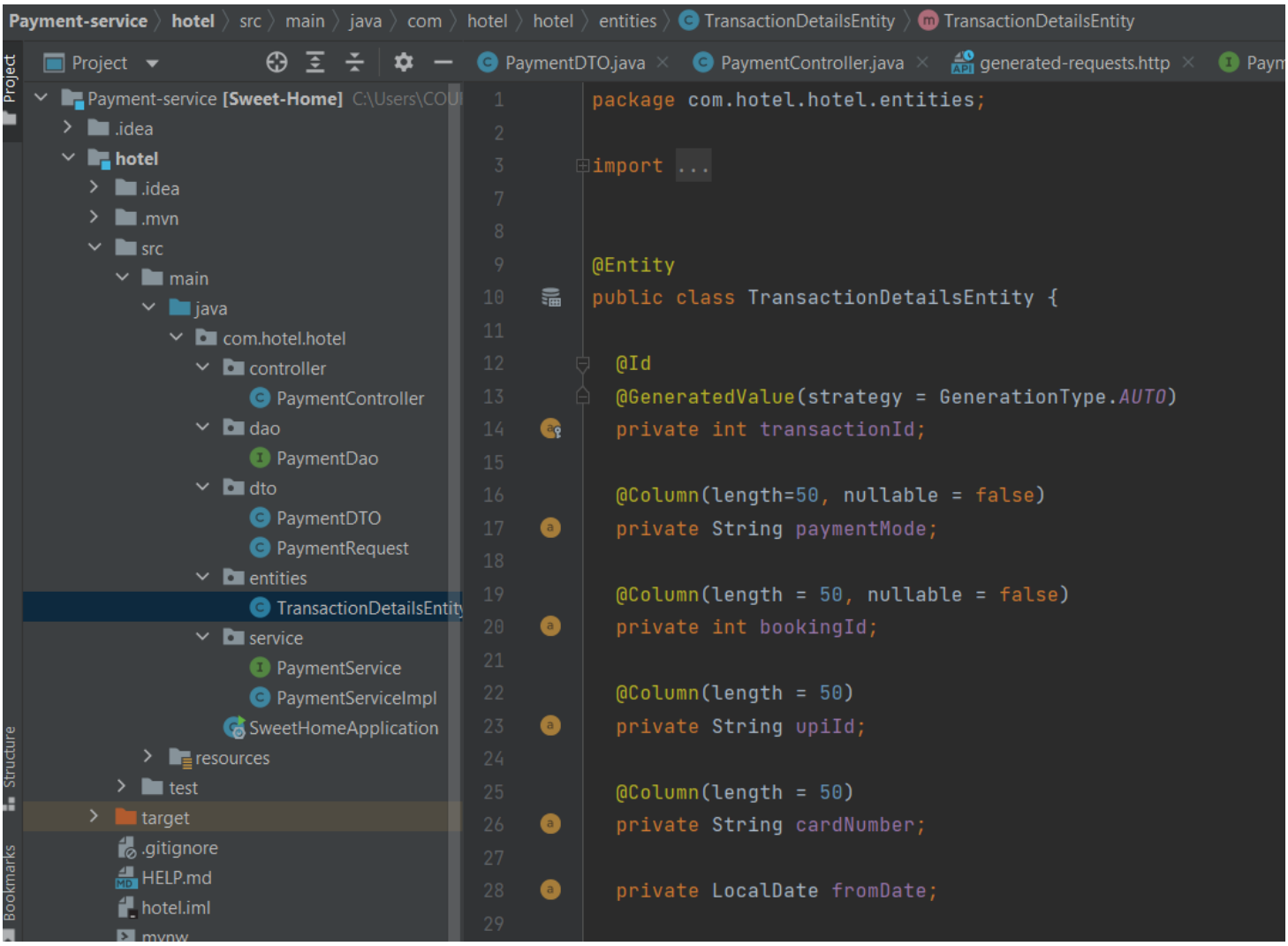
GET http://localhost:8080/payment/transaction/{{transactionId}} API call is made to fetch the payment details based on the transaction id using the method

```
public ResponseEntity getPaymentDetailsBasedOnId(@PathVariable(name="transactionId") int
transactionId) throws Exception
```

which has its implementation in the service Class **PaymentServiceImpl (**interface used **PaymentService)** in the below method.

```
public TransactionDetailsEntity getPaymentById(int transactionId) throws Exception
```

Entity class with the field mapping

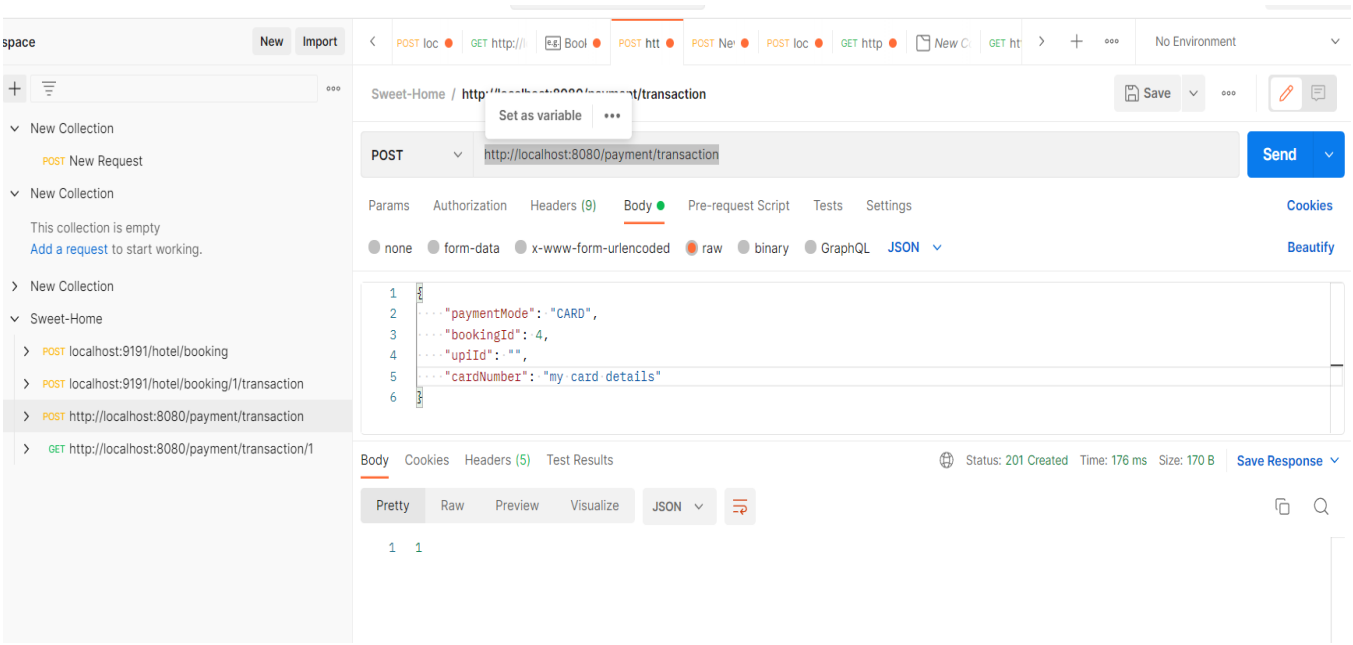


API calls

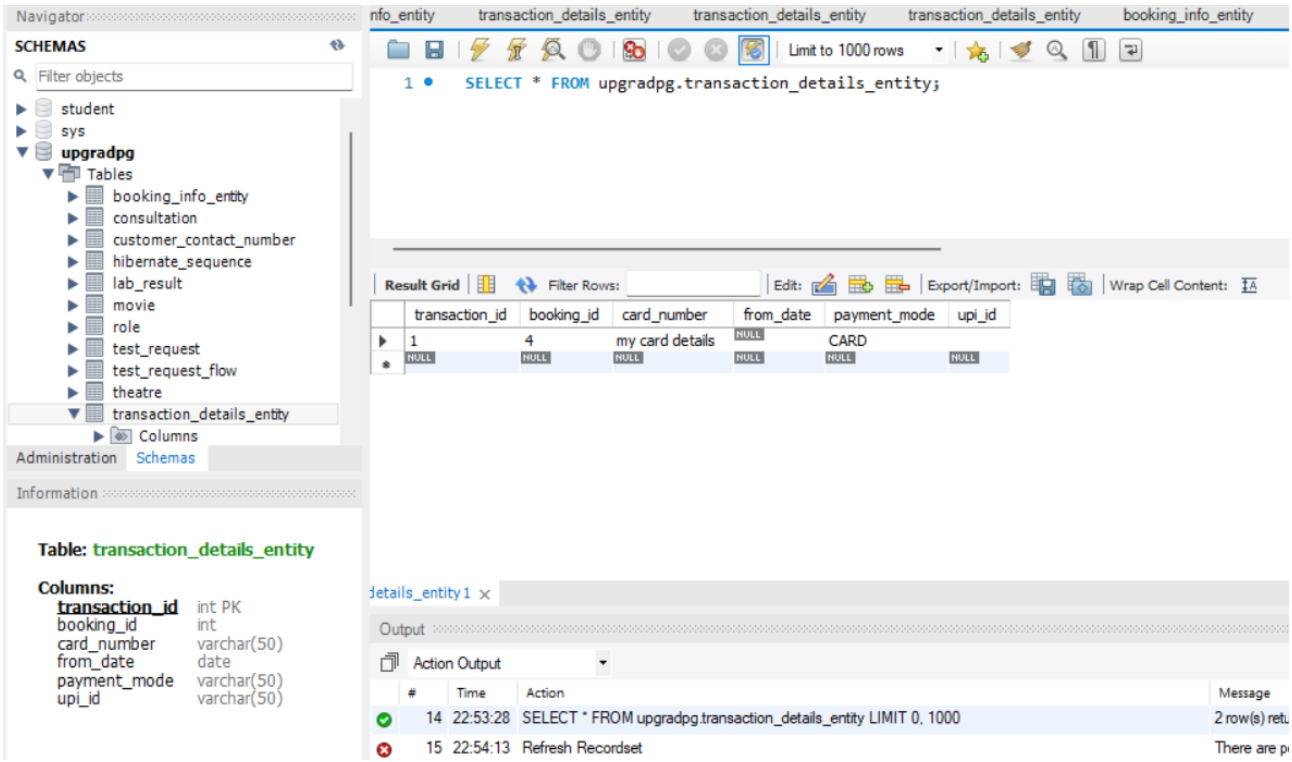
(please use port 8080 to make the API calls for this service)

1. Using this API call the payment details is saved the database and the transaction id is returned int he response , call done using postman , below is the request and response JSON in the screenshot.

http://localhost:8080/payment/transaction

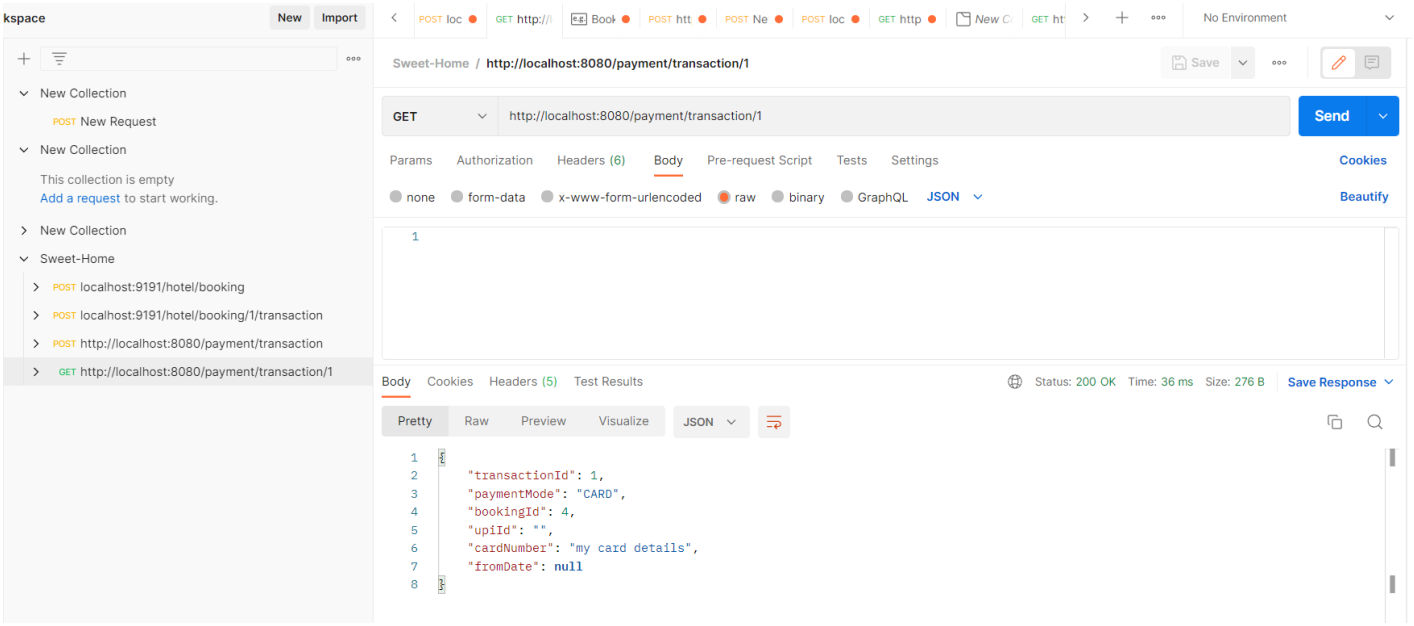


Data saved in the database



2. Request with the transaction id and get the payment details from the database and display it in the response.

   http://localhost:8080/payment/transaction/1



(Note : database details used is as provided in the assignment details)