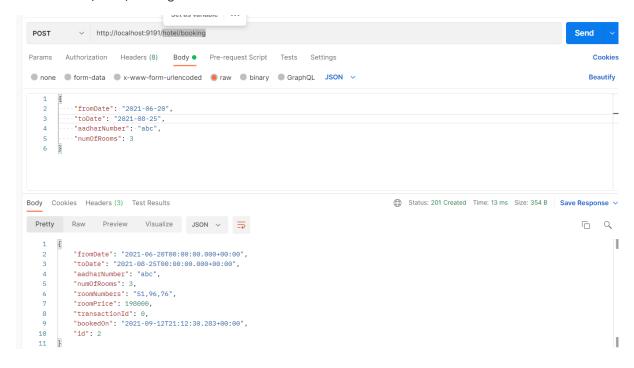
Booking-Service

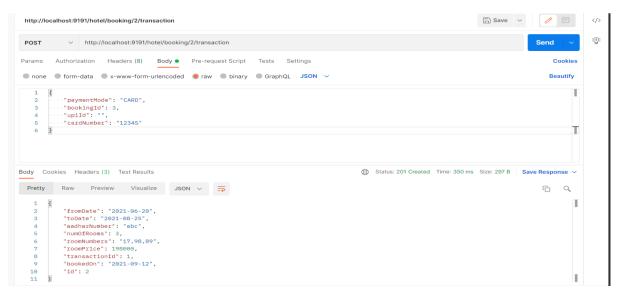
Database: H2 database(Change the username and password)

EndPoint 1: /hotel/booking

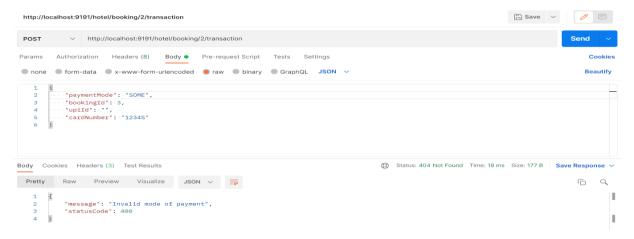


EndPoint 2: /hotel/booking/{bookingId}/transaction

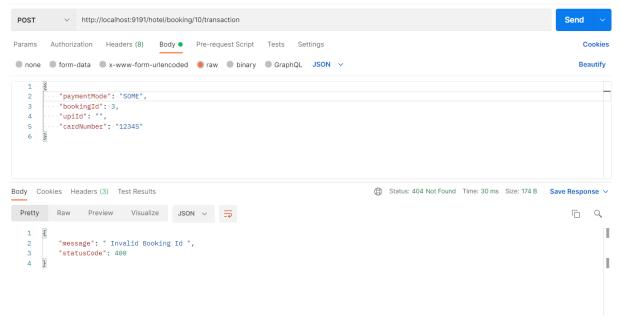
Case 1:



Case 2: Payment Mode = "SOME"



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



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.

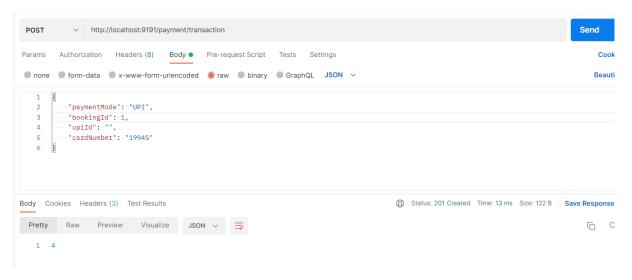
getPaymentDetails(int bookingId, TransactionDetailsEntity transactionDetailsEntity):

Checks weather 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

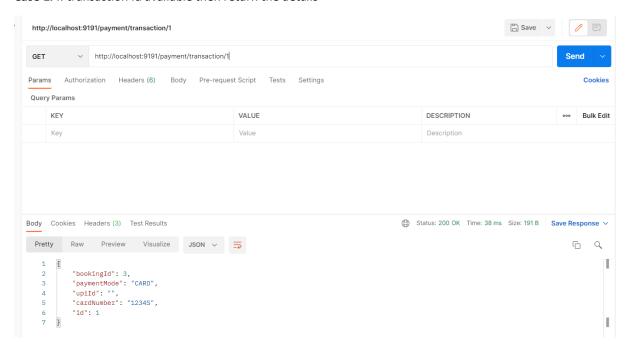
Database: H2 database

EndPoint 1: /payment/transaction

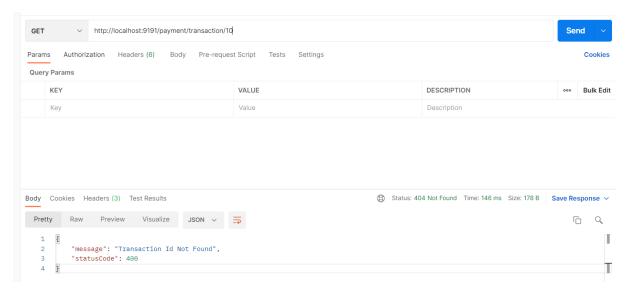


EndPoint 2: /payment/transaction/{transactionId}

Case 1: If transaction Id available then return the details



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



PaymentServiceImpl:

Methods in the PaymentServiceImpl:

getTransactionId(TransactionDetailsEntity transactionDetailsEntity):

Saves the transactionDetailsEntity and returns the transactionId.

getTransactionDetails(int id):

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