HOTEL ROOM BOOKING

Booking Service :

This service is responsible for taking input from users like- toDate, fromDate, aadharNumber and the number of rooms required (numOfRooms) and save it in its database. If the user wishes to go ahead with the booking, they can provide the payment related details like bookingMode, upiId / cardNumber, which will be further sent to the payment service to retrieve the transactionId. This transactionId then gets updated in the Booking table created in the database of the Booking Service and a confirmation message is printed on the console.

This service contains following folder :

Entity : It contains BookingInfoEntity .java file , which is used to create necessary columns , getter setter , to string method for database .

DAO : It contains BookingDao.java file , which used jpa repository to perform CRUD operation on db .

DTO : It contains BookingDTO , ErrorDTO and PaymentDTO , where PaymentDTO is used to maps the API request and fetch data from payment service

Service : It contains BookingService.java and BookingServiceImpl.java where BookingService.java is used to declare all necessary method to perform the business logic required by end points and BookingServiceImpl.java contains all implementation of this methods.

Controller : It contains BookingController.java , which is used create all necessary end points and further implementation is perform by service file .

Exception : It handles all necessary exception for the endpoints.

Pom.xml : It contains all required dependencies .

Application.properties : It contains all necessary configuration like database , port number .

BookingApplication.java : It is the starting trigger of the application .

Database : H2

Port Number : 8080

Payment Service :

This service is responsible for taking payment-related information- paymentMode, upiId or cardNumber, bookingId and returns a unique transactionId to the booking service. It saves the data in its database and returns the transactionId as a response.

This service contains following folder :

Entity : It contains TransactionDetaolsEntity .java file , which is used to create necessary columns , getter setter , to string method for database .

DAO : It contains TransactionDao.java file , which used jpa repository to perform CRUD operation on db .

DTO : It contains TransactionDTO , where PaymentDTO is used to maps the API request and fetch data from payment service

Service : It contains PaymentService.java and PaymentServiceImpl.java where PaymentService.java is used to declare all necessary method to perform the business logic required by end points and PaymentServiceImpl.java contains all implementation of this methods.

Controller : It contains PaymentController.java , which is used create all necessary end points and further implementation is perform by service file .

Exception : It handles all necessary exception for the endpoints.

Pom.xml : It contains all required dependencies .

Application.properties : It contains all necessary configuration like database , port number .

BookingApplication.java : It is the starting trigger of the application .

Database : H2

Port Number : 8081

Eureka server running on port Number : 8761

Steps To Run Project :

- Download the zip file and import in IntelliJ .
- Open Pom.xml in all 3 folder and set the environment .
- Run EurekaApplication.java from Eureka Folder .
- Run BookingApplication.java from Booking Folder .

- Run PaymentApplcation.java from Payment Folder.
- Run following APIs in PostMan:

Eureka Server   http://localhost:8761/

Booking service : POST :http://localhost:8080/hotel/booking

POST : http://localhost:8080/hotel/booking/1/transaction

Payment Service :   POST : http://localhost:8081/payment/transaction

GET : http://localhost:8081/payment/transaction/1

Note :

- ✓ Eureka Port Number : 8761
- ✓ Booking Service Port Number : 8080
- ✓ Payment Service Port Number : 8081