RestEasy Proj :

Application Overview :

This application is created using Spring boot and mysql.

Based on details provided required tables are created . Below are the database queries.

DataBase :

1. CREATE DATABASE restaurant;
2. CREATE TABLE login

(

userid VARCHAR(150) NOT NULL, # Name of the user

password VARCHAR(150) NOT NULL, # password

userrole VARCHAR(50) NOT NULL, # userRole

PRIMARY KEY (userid) # Make the id the primary key

);

1. CREATE TABLE dish

(

dishid INT unsigned NOT NULL AUTO\_INCREMENT, # Unique ID for the record

dishname VARCHAR(150) NOT NULL, # Name of the dish

dishcalories VARCHAR(150) NOT NULL, # calories in dish

PRIMARY KEY (dishid) # Make the id the primary key

);

1. CREATE TABLE vendor

(

vendorid INT unsigned NOT NULL AUTO\_INCREMENT, # Unique ID for the record

vendorname VARCHAR(150) NOT NULL, # Name of the dish

dishid int(10) unsigned, # calories in dish

dishprice FLOAT(10,2) NOT NULL,

PRIMARY KEY (vendorid), # Make the id the primary key

FOREIGN KEY (dishid) REFERENCES dish(dishid)

);

1. CREATE TABLE foodorder

(

orderid INT unsigned NOT NULL ,

dishname VARCHAR(150) NOT NULL,

vendorname VARCHAR(150) NOT NULL, # Name of the dish

qty INT unsigned NOT NULL, # calories in dish

unitprice FLOAT(10,2) NOT NULL,

totalprice FLOAT(10,2) NOT NULL,

orderdate date

);

mysql> show tables;

+----------------------+

| Tables\_in\_restaurant |

+----------------------+

| dish |

| foodorder |

| login |

| vendor |

+----------------------+

1. rows in set (0.00 sec)

Application Flow :

The application is designed in Spring Boot and uses CLI as UI , hence scanner is used for taking user inputs. Due to this Junits cannot be written as Scanners are Final (we can’t mock). I have designed the application on REST and each functionality works as an API.

1. ResteasyApplication :

Application is started from here.

1. SpringBootJdbcController :

This class acts as controller and service both. Based on URL path here we decide which API to call. This class also contains business logics.

1. DatabaseHandler :

This class helps to create and close Db connections.

1. AdminDao :

Used for admin related queries.

1. Logindao :

Used for Login in and getting user role , based on user role we decide if few of the admin queries/APIs should run or not.

1. QueryDao : Holds all the queries related to different APIs

Apart for these we have 2 beans acting like pojo for holding data. One pojo named Orderlist.java is used for holding order details.

API List and uses:

URL : <http://localhost:8080/loginDBCall>

Above URL will ask for login and will determine the user role.

URL: <http://localhost:8080/showallVendors>

Above URL will show all the details of Menu for all Vendor

URL: <http://localhost:8080/showJustVendors>

Shows only vendors list , without food details

URL: http://localhost:8080/addVendors

Can be used only by admin to enter food and vendor details.

URL: http://localhost:8080/showByDish

Shows menu details based on dish name entered.

URL: http://localhost:8080/showByVendor

Shows menu details based on Vendor name entered.

URL: http://localhost:8080/createOrder

Allows users to create Orders.

URL: http://localhost:8080/getAllOrders

Only admin can use this and get all orders details.

URL: http://localhost:8080/printByOrderAmt

Only admin can use this and get all orders details order by Order Amount.

Admin needs to give input as asc or desc.

URL: http://localhost:8080/printByOrderDate

Only admin can use this and get all orders details order by Order Date.

Admin needs to give input as asc or desc.

URL: http://localhost:8080/showByPrice

Users can use this to get menu details order by Price.

Admin needs to give input as asc or desc.