RESTFul ASSIGNMENT Day_2

Harshit Kushmakar| 16896

1. In the same RestService add one method to test Post method mapping. Create a form to accept username and send it to the service with the help of Post method. In the service retrieve this username with the help of @FormParam annotation and send a message “Hello “ + username back to the client application. Use Postman API to test your service. Make use of proper @Consumes and @Produces annotation.

```
<form action="rest/Day2Q1/service" method="post">
        Enter Name : <input type="text" name="name"/><br/><br/>
<input type="submit" value="SUBMIT"/>
</form>
```

```
package com.rest.assignment02;
        import javax.ws.rs.*;
        import javax.ws.rs.core.MediaType;
        import javax.ws.rs.core.Response;
@Path("/Day2Q1")
public class Question1 {
    @POST
    @Path("/service")
    @Produces(MediaType.TEXT_PLAIN)
    public Response service(
            @FormParam("name") String name
    ){
        return Response.status(200)
                .entity("Hello " + name)
                .build();
    }
}
```

2. Update the Question – 4 of Day 1 to accept username and password. Authenticate the user.

```
package com.rest.assignment01;
        import com.rest.entity.LoanDetails;
        import com.rest.entity.LoanDetailsDAO;
        import javax.ws.rs.*;
        import javax.ws.rs.core.MediaType;
        import java.util.HashMap;
        import java.util.List;
@Path("/Question4")
public class Question4 {
    HashMap<Integer, String> hashMap = new HashMap<>();
    public void insertCredentials(){
        hashMap.put(1, "Harshit");
```

```
        hashMap.put(2, "Aman");
    }
    @GET
    @Path("/read")
    @Produces(MediaType.TEXT_PLAIN)
    public String getMessage(
            @QueryParam("ID") int customerID,
            @QueryParam("Pass") String password
    ){
        insertCredentials();
        if(!hashMap.get(customerID).equals(password))
            return "Password not found.";
        LoanDetailsDAO obj = new LoanDetailsDAO();
        List<LoanDetails> customerDetails = obj.read(customerID);
        if(customerDetails.size() == 0)
            return "The Customer ID does not exists.";
        String str = "Customer ID : " + customerID + "\n";
        str += "Customer Name : " +
customerDetails.get(0).getCustomerName();
        str += "\nCustomer Number : " +
                customerDetails.get(0).getContactNumber();
        str += "\nCustomer Email : " + customerDetails.get(0).getEmail();
        for(int i=0; i<customerDetails.size(); i++)
            str += customerDetails.get(i).toString();
        return str;
    }
}
```

3. Create an application using the RESTful Web Services. The application needs to use GET, POST, DELETE methods to perform the following operations: a. GET/customers - Get the list of all the registered customers in form of JSON object from the database. b. GET/customer/{customerId} - Get the Customer details given the customer Id. If the customer is not available, it send null object back. c. POST – register a customer from the details given in the form. The customer class can be used from Java Day-3 assignment. d. DELETE – given a customer id, delete the customer details from the database.

```
package com.rest.assignment02;
        import com.rest.entity.Customer;
        import com.rest.entity.CustomerDAO;
        import javax.ws.rs.*;
        import javax.ws.rs.core.MediaType;
        import java.util.List;
@Path("/Day2Q3")
public class Question3 {
    @GET
    @Path("/customers") // Q3A
    @Produces(MediaType.TEXT_PLAIN)
    public String getAll(){
        String str = "Here are the list of contents : ";
        CustomerDAO obj = new CustomerDAO();
        List<Customer> customerList = obj.readAll();
        if(customerList.size() == 0)
```

```
            return "The List is empty.";
        for(int i=0; i< customerList.size(); i++)
            str += customerList.get(i).toString();
        return str;
    }
    @GET
    @Path("/customers/{param}") //Q3B
    @Produces(MediaType.TEXT_PLAIN)
    public String getCustomer(@PathParam("param") int customerId){
        String str = "Data of Customer : " + customerId;
        CustomerDAO obj = new CustomerDAO();
        List<Customer> customerList = obj.readAll();
        if(customerList.size() == 0)
            return "The List is empty.";
        boolean get = false;
        for(int i=0; i< customerList.size(); i++) {
            if(customerList.get(i).getCustomerId() == customerId) {
                get = true;
                str += customerList.get(i).toString();
            }
        }
        if(get)
            return str;
        return null;
    }
    @POST
    @Path("/insertCustomer") //Q3C
    public String insertCustomer(
            @FormParam("custId") int customerId,
            @FormParam("custName") String name,
            @FormParam("custDob") String date,
            @FormParam("custEmail") String email,
            @FormParam("custPhone") long phone,
            @FormParam("custIncome") long monthlyIncome,
            @FormParam("custExpense") long monthyExpense
    ){
        Customer obj = new Customer(customerId, name, date, email, phone,
                monthlyIncome, monthyExpense);
        //System.out.println(obj);
        CustomerDAO dao = new CustomerDAO();
        return dao.insert(obj);
    }
    @DELETE
    @Path("/deleteCustomer")
    public String deleteCustomer(
            @QueryParam("custDelId") int customerId
    ){
        CustomerDAO obj = new CustomerDAO();
        List<Customer> customerList = obj.readAll();
        if(customerList.size() == 0)
            return "The Customer List is Empty.";
        boolean get = false;
        for(int i=0; i< customerList.size(); i++) {
            if(customerList.get(i).getCustomerId() == customerId) {
                get = true;
                break;
            }
        }
        if(get == false)
            return "The customer ID doesn't exists.";
        return obj.delete(customerId);
```

```
    }
}
```

```
package com.rest.entity;
public class Customer {
    private int customerId;
    private String name;
    private String dob;
    private String email;
    private long number;
    private long monthyIncome;
    private long monthlyExpense;
    public Customer(int customerId, String name, String dob, String email,
long
            number, long monthyIncome, long monthlyExpense) {
        this.customerId = customerId;
        this.name = name;
        this.dob = dob;
        this.email = email;
        this.number = number;
        this.monthyIncome = monthyIncome;
        this.monthlyExpense = monthlyExpense;
    }
    public int getCustomerId() {
        return customerId;
    }
    public void setCustomerId(int customerId) {
        this.customerId = customerId;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getDob() {
        return dob;
    }
    public void setDob(String dob) {
        this.dob = dob;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public long getNumber() {
        return number;
    }
    public void setNumber(long number) {
        this.number = number;
    }
    public long getMonthyIncome() {
        return monthyIncome;
    }
    public void setMonthyIncome(long monthyIncome) {
        this.monthyIncome = monthyIncome;
    }
    public long getMonthlyExpense() {
```

```
        return monthlyExpense;
    }
    public void setMonthlyExpense(long monthlyExpense) {
        this.monthlyExpense = monthlyExpense;
    }
    @Override
    public String toString() {
        return "\nCustomer{" +
                "customerId=" + customerId +
                ", name='" + name + '\'' +
                ", dob='" + dob + '\'' +
                ", email='" + email + '\'' +
                ", number=" + number +
                ", monthyIncome=" + monthyIncome +
                ", monthlyExpense=" + monthlyExpense +
                '}';
    }
}
```

```
package com.rest.entity;
        import java.sql.*;
        import java.util.ArrayList;
        import java.util.List;
public class CustomerDAO {
    Connection con = null;
    Statement stmt = null;
    private static String connectionParameters = "training";
    public CustomerDAO(){}
    public void conn() {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            con = DriverManager.getConnection(
                    "jdbc:oracle:thin:training@//10.1.50.198:1535/nsbt19c",
                    connectionParameters, connectionParameters);
            stmt = con.createStatement();
        } catch (SQLException | ClassNotFoundException e) {
            e.getMessage();
        }
    }
    public List<Customer> readAll(){
        conn();
        List<Customer> customerDetails = null;
        try {
            ResultSet rs = stmt.executeQuery("SELECT * FROM
                    CUSTOMERDETAILS16920");
                    customerDetails = new ArrayList<>();
            while (rs.next()) {
                Customer obj = new Customer(rs.getInt(1), rs.getString(2),
                        rs.getString(3), rs.getString(4), rs.getLong(5),
rs.getLong(6), rs.getLong(7));
                customerDetails.add(obj);
            }
            return customerDetails;
        } catch (SQLException e) {
            e.getMessage();
        }
        return null;
    }
    public String insert(Customer obj){
        conn();
```

```
        try {
            String query = "INSERT into CUSTOMERDETAILS16920 VALUES(" +
                    obj.getCustomerId() + ", '" + obj.getName() + "', '" +
obj.getDob() +
                    "', '" + obj.getEmail() + "', " + obj.getNumber() + ","
+
                    obj.getMonthyIncome() + "," + obj.getMonthlyExpense() +
")";
            //System.out.println(query);
            stmt.executeUpdate(query);
            return "Successful Insertion";
        } catch (NumberFormatException | SQLException e){
            System.out.println(e.getMessage());
            return "Cannot Insert due to Exception";
        }
    }
    public String delete(int customerId){
        conn();
        try {
            String query = "DELETE FROM CUSTOMERDETAILS16920 WHERE
CUSTOMERID = "
                    + customerId;
            stmt.executeUpdate(query);
            return "Successful Deletion";
        } catch (NumberFormatException | SQLException e){
            System.out.println(e.getMessage());
            return "Cannot Delete due to Exception";
        }
    }
}
```

```
<br>
<a href="rest/Day2Q3/customers">Customer Get All</a>
<br>
<br>
<a href="rest/Day2Q3/customers/1">Get Customer 1</a>
<br>
<form method="post" action="rest/Day2Q3/insertCustomer">
        Enter Customer ID : <input type="number" name="custId">
        Enter Customer Name : <input type="text" name="custName">
        Enter Customer Dob : <input type="text" name="custDob">
        Enter Customer Email : <input type="text" name="custEmail">
        Enter Customer Number : <input type="number" name="custPhone">
        Enter Customer Monthly Income : <input type="number"
name="custIncome">
        Enter Customer Monthly Expense : <input type="number"
name="custExpense">
<button type="submit">Register</button>
</form>
<br>
<form action="rest/Day2Q3/deleteCustomer" method="post">
<input type="hidden" name="_method" value="delete">
        Enter Customer ID for Deletion : <input type="number"
name="custDelId">
<button type="submit" value="Delete">Delete</button>
</form>
<br>
```

4. Create a RESTful web service to find out whether mentioned applicant is eligible for loan and for what amount. The customer object needs to be passed to the method of the webservice.

```
package com.rest.assignment02;
        import com.rest.entity.Customer;
        import com.rest.entity.CustomerDAO;
        import javax.ws.rs.*;
        import javax.ws.rs.core.MediaType;
        import java.util.List;
@Path("/Day2Q4")
public class Question4 {
    @GET
    @Path("/loan/{param}") //Q4
    @Produces(MediaType.TEXT_PLAIN)
    public String getCustomer(@PathParam("param") int customerId){
        String str = "";
        CustomerDAO obj = new CustomerDAO();
        List<Customer> customerList = obj.readAll();
        if(customerList.size() == 0)
            return "The List is empty.";
        boolean get = false;
        long salary = 0;
        long expense = 0;
        for(int i=0; i< customerList.size(); i++) {
            if(customerList.get(i).getCustomerId() == customerId) {
                get = true;
                salary = customerList.get(i).getMonthyIncome();
                expense = customerList.get(i).getMonthlyExpense();
            }
        }
        if(get && salary > expense){
            str += "Eligible for Loan.\n Max Loan Amount : " + (2 *
salary);
            return str;
        }
        return "Customer Doesn't Exists.";
    }
}
```

5. Create a RESTful web service to get the details of all the loans whose emis are overdue.

```
package com.rest.assignment02;
        import com.rest.entity.EmiDAO;
        import javax.ws.rs.GET;
        import javax.ws.rs.Path;
        import javax.ws.rs.Produces;
        import javax.ws.rs.core.MediaType;
@Path("/Day2Q5")
public class Question5 {
    @GET
    @Path("/dueEmi")
    @Produces(MediaType.TEXT_PLAIN)
    public String getDueEmi(){
```

```
        EmiDAO obj = new EmiDAO();
        return obj.readDueEmi();
    }
}
```

```
       package com.rest.entity;
        import java.sql.*;
        import java.util.ArrayList;
        import java.util.List;
public class EmiDAO {
    Connection con = null;
    Statement stmt = null;
    private static String connectionParameters = "training";
    public EmiDAO(){}
    public void conn() {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            con = DriverManager.getConnection(
                    "jdbc:oracle:thin:training@//10.1.50.198:1535/nsbt19c",
                    connectionParameters, connectionParameters);
            stmt = con.createStatement();
        } catch (SQLException | ClassNotFoundException e) {
            e.getMessage();
        }
    }
    public String readDueEmi(){
        conn();
        List<Emi> customerDetails = null;
        String str = "";
        try{
            ResultSet rs = stmt.executeQuery("SELECT * FROM
                    CUSTOMEREMISTATUS16920 WHERE UPPER(CustomerEMIStatus) =
'DUE'");
                    customerDetails = new ArrayList<>();
            while (rs.next()) {
                Emi obj = new Emi(rs.getInt(1), rs.getString(2),
                        rs.getString(3));
                customerDetails.add(obj);
            }
            for(int i=0; i<customerDetails.size(); i++) {
                System.out.println(customerDetails.get(i).toString());
                str += customerDetails.get(i).toString();
            }
            return str;
        } catch (NumberFormatException | SQLException e){
            System.out.println(e.getMessage());
            return "Cannot Insert due to Exception";
        }
    }
}
```

```
       package com.rest.entity;
public class Emi {
    private int customerId;
    private String customerLoanId;
    private String customerEmiStatus;
```

```
    public Emi(int customerId, String customerLoanId, String
customerEmiStatus) {
        this.customerId = customerId;
        this.customerLoanId = customerLoanId;
        this.customerEmiStatus = customerEmiStatus;
    }
    public int getCustomerId() {
        return customerId;
    }
    public void setCustomerId(int customerId) {
        this.customerId = customerId;
    }
    public String getCustomerLoanId() {
        return customerLoanId;
    }
    public void setCustomerLoanId(String customerLoanId) {
        this.customerLoanId = customerLoanId;
    }
    public String getCustomerEmiStatus() {
        return customerEmiStatus;
    }
    public void setCustomerEmiStatus(String customerEmiStatus) {
        this.customerEmiStatus = customerEmiStatus;
    }
    @Override
    public String toString() {
        return "\nEmi{" +
                "customerId=" + customerId +
                ", customerLoanId='" + customerLoanId + '\'' +
                ", customerEmiStatus='" + customerEmiStatus + '\'' +
                '}';
    }
}
```