DEVELOP CRUD PROGRAM FOR MEMBER TABLE

Stella Kim
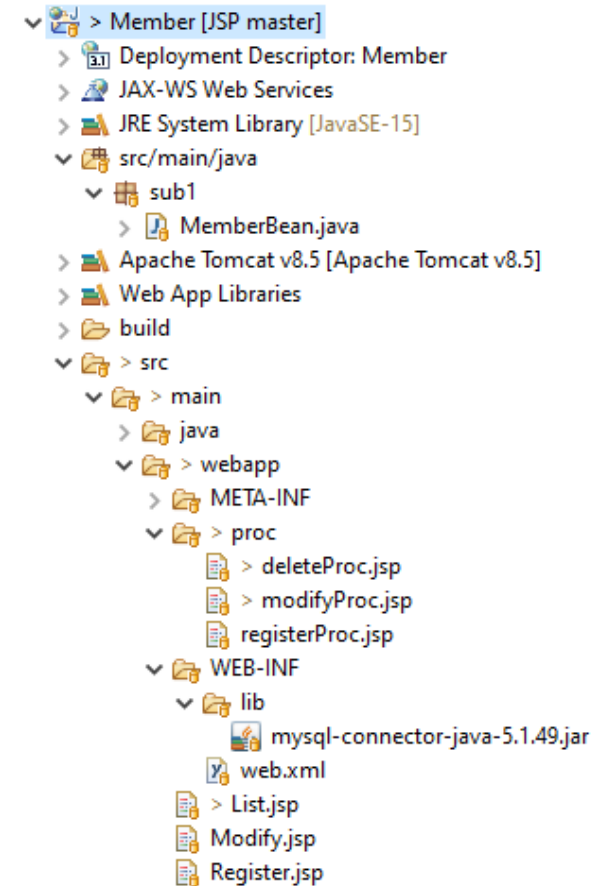
March 29th, 2021

INDEX

- Set up Eclipse
- INSERT operation
- SELECT operation
- UPDATE operation
- DELETE operation

GENERAL SETUP ON ECLIPSE

- Add mysql-connector-java-5.1.49.jar file to lib folder under src-main-webapp-WEB-INF.

- Create List, Modify, Register JSP files under webapp.

- Make Class MemberBean as Java file under src/main/java.

- Create a new folder as proc under webapp and add registerProc, modifyProc, deleteProc JSP files.

1. INSERT

Components

o Register.JSP

o registerProc.JSP

Register an employee

ID	Enter your ID
Name	Enter your name
Phone	Enter your phone number
Position	associate ⌄
Department	1 ⌄
	submit

1. INSERT - REGISTER.JSP

- Create a html file for registration page.

- Make a table for entering data of ID, name, phone number, position, department number with a submit button.

```jsp
1  <%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"%>
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <meta charset="UTF-8">
6      <title>Register</title>
7  </head>
8  <body>
9      <h3>Register an employee</h3>
10
11     <form action="./proc/registerProc.jsp" method="post">
12         <table border="1">
13             <tr>
14                 <td>ID</td>
15                 <td><input type="text" name="uid" placeholder="Enter your ID"/></td>
16             </tr>
17             <tr>
18                 <td>Name</td>
19                 <td><input type="text" name="name" placeholder="Enter your name"/></td>
20             </tr>
21             <tr>
22                 <td>Phone</td>
23                 <td><input type="tel" name="hp" placeholder="Enter your phone number"/></td>
24             </tr>
25             <tr>
26                 <td>Position</td>
27                 <td>
28                     <select name="pos">
29                         <option>associate</option>
30                         <option>manager class2</option>
31                         <option>manager class1</option>
32                         <option>deputy manager</option>
33                         <option>head manager</option>
34                     </select>
35                 </td>
36             </tr>
```

1. INSERT
- REGISTER.JSP

- Place the table under form tag which leads to RegisterProc JSP file as POST method.

```
37    <tr>
38        <td>Department</td>
39        <td>
40            <select name="dep">
41                <option value="101">1</option>
42                <option value="102">2</option>
43                <option value="103">3</option>
44                <option value="104">4</option>
45                <option value="105">5</option>
46                <option value="106">6</option>
47                <option value="107">7</option>
48            </select>
49        </td>
50    </tr>
51    <tr>
52        <td colspan="2" align="right">
53            <input type="submit" value="submit">
54        </td>
55    </tr>
56    </table>
57    </form>
58
59 </body>
60 </html>
```

1. INSERT
- REGISTERPROC.JSP

- Processing SQL statements with JDBC
  1. Encoding data as UTF-8.
  2. Receiving data as variables.
  3. Establishing a connection with HeidiSQL.
  4. Create a statement to insert values into member table.
  5. Update the query.
  6. Close the connection.
  7. Redirect to List page.

```jsp
1  <%@page import="java.sql.Connection"%>
2  <%@page import="java.sql.Statement"%>
3  <%@page import="java.sql.DriverManager"%>
4  <%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"%>
5  <%
6      //encoding data
7      request.setCharacterEncoding("utf-8");
8
9      //receiving data
10     String uid = request.getParameter("uid");
11     String name = request.getParameter("name");
12     String hp = request.getParameter("hp");
13     String pos = request.getParameter("pos");
14     String dep = request.getParameter("dep");
15
16     //database
17     String host = "jdbc:mysql://192.168.10.114:3306/kye";
18     String user = "kye";
19     String pass = "1234";
20
21     //1.load JDBC driver
22     Class.forName("com.mysql.jdbc.Driver");
23
24     //2.access to the dadtabase
25     Connection conn = DriverManager.getConnection(host,user,pass);
26
27     //3.make SQL object
28     Statement stmt = conn.createStatement();
29
30     //4.SQL
31     String sql = "INSERT INTO `MEMBER` VALUES('"+uid+"','"+name+"','"+hp+"','"+pos+"','"+dep+"', NOW());";
32     stmt.executeUpdate(sql);
33
34     //5.process the result(if SELECT)
35     //6.exit database
36     stmt.close();
37     conn.close();
38
39     //redirect
40     response.sendRedirect("../List.jsp");
41 %>
```

2. SELECT

Components

o List.JSP

o MemberBean.JAVA

List of employees

ID	Name	Phone	Position	Department	Registered date	Modify	
a101	Stella Kim	010-1234-1111	associate	101	03/24/21	Edit	Delete
a102	김유신	010-1234-1002	차장	101	21-03-24	Edit	Delete
a103	김춘추	010-1234-1003	사원	101	21-03-24	Edit	Delete
a104	장보고	010-1234-1004	대리	102	21-03-24	Edit	Delete
a105	강감찬	010-1234-1005	과장	102	21-03-24	Edit	Delete
a106	이성게	010-1234-1006	차장	103	21-03-24	Edit	Delete
a107	정철	010-1234-1007	차장	103	21-03-24	Edit	Delete
a108	이순신	010-1234-1008	부장	104	21-03-24	Edit	Delete
a109	허균	010-1234-1009	부장	104	21-03-24	Edit	Delete
a110	정약용	010-1234-1010	사원	105	21-03-24	Edit	Delete
a111	박지원	010-1234-1011	사원	105	21-03-24	Edit	Delete
b101	을지문덕	010-5555-1234	사장	107	21-03-24	Edit	Delete

1. INSERT

- MEMBERBEAN.JAVA

- Imported to List.JSP

- Set columns of Member table as variables

- Generate Getter and Setter for each variable

```java
1  package sub1;
2
3  public class MemberBean {
4
5      private String uid;
6      private String name;
7      private String hp;
8      private String pos;
9      private int dep;
10     private String rdate;
11
12     public String getUid() {
13         return uid;
14     }
15     public void setUid(String uid) {
16         this.uid = uid;
17     }
18     public String getName() {
19         return name;
20     }
21     public void setName(String name) {
22         this.name = name;
23     }
24     public String getHp() {
25         return hp;
26     }
27     public void setHp(String hp) {
28         this.hp = hp;
29     }
30     public String getPos() {
31         return pos;
32     }
33     public void setPos(String pos) {
34         this.pos = pos;
35     }
36     public int getDep() {
37         return dep;
38     }
39     public void setDep(int dep) {
40         this.dep = dep;
41     }
42     public String getRdate() {
43         return rdate;
44     }
45     public void setRdate(String rdate) {
46         this.rdate = rdate;
47     }
48 }
```

2. SELECT - LIST.JSP

- Processing SQL statements with JDBC

  1. Encoding data as UTF-8.

  2. Receiving data as variables.

  3. Establishing a connection with HeidiSQL.

  4. Create a statement to select all from the member table.

  5. Execute the query.

  6. Process the ResultSet objects to move the cursor.

  7. Set MemberBean as an array list and enter the data.

  8. Close the connection.

```jsp
1  <%@page import="sub1.MemberBean"%>
2  <%@page import="java.util.ArrayList"%>
3  <%@page import="java.util.List"%>
4  <%@page import="java.sql.ResultSet"%>
5  <%@page import="java.sql.Statement"%>
6  <%@page import="java.sql.DriverManager"%>
7  <%@page import="java.sql.Connection"%>
8  <%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"%>
9  <%
10     //database
11     String host = "jdbc:mysql://192.168.10.114:3306/kye";
12     String user = "kye";
13     String pass = "1234";
14
15     //1.load JDBC driver
16     Class.forName("com.mysql.jdbc.Driver");
17
18     //2.access to the dadtabase
19     Connection conn = DriverManager.getConnection(host,user,pass);
20
21     //3.make SQL object
22     Statement stmt = conn.createStatement();
23
24     //4.SQL
25     String sql = "SELECT * FROM `MEMBER`;";
26     ResultSet rs = stmt.executeQuery(sql);
27
28     //5.process the result(if SELECT)
29     List<MemberBean> mbList = new ArrayList<>();
30
31     while(rs.next()){
32         MemberBean mb = new MemberBean();
33
34         mb.setUid(rs.getString(1));
35         mb.setName(rs.getString(2));
36         mb.setHp(rs.getString(3));
37         mb.setPos(rs.getString(4));
38         mb.setDep(rs.getInt(5));
39         mb.setRdate(rs.getString(6));
40         mbList.add(mb);
41     }
42
43     //6.exit database
44     rs.close();
45     stmt.close();
46     conn.close();
47  %>
```

2. SELECT - LIST.JSP

- Make a table for displaying the list of employees from the database.

- Print the table composed of data from the list

- Links attached:

  - Register: Register.jsp for INSERT operation

  - Edit: Modify.jsp through uid, name, hp

  - Delete: deleteProc.jsp through uid

```html
49  <!DOCTYPE html>
50  <html>
51  <head>
52      <meta charset="UTF-8">
53      <title>List</title>
54  </head>
55  <body>
56      <h3>List of employees</h3>
57
58      <a href="./Register.jsp">Register</a>
59      <table border="1">
60          <tr>
61              <th>ID</th>
62              <th>Name</th>
63              <th>Phone</th>
64              <th>Position</th>
65              <th>Department</th>
66              <th>Registered date</th>
67              <th>Modify</th>
68          </tr>
69          <tr>
70              <td>a101</td>
71              <td>Stella Kim</td>
72              <td>010-1234-1111</td>
73              <td>associate</td>
74              <td>101</td>
75              <td>03/24/21</td>
76              <td>
77                  <a href="./Modify.jsp?uid=a101&name=Stella Kim&hp=010-1234-1111">Edit</a>
78                  <a href="./proc/deleteProc.jsp?uid=a101">Delete</a>
79              </td>
80          </tr>
81          <%for(MemberBean mb:mbList){ %>
82          <tr>
83              <td><%=mb.getUid() %></td>
84              <td><%=mb.getName() %></td>
85              <td><%=mb.getHp() %></td>
86              <td><%=mb.getPos() %></td>
87              <td><%=mb.getDep() %></td>
88              <td><%=mb.getRdate().substring(2,10) %></td>
89              <td>
90                  <a href="./Modify.jsp?uid=<%=mb.getUid() %>&name=<%=mb.getName() %>&hp=<%=mb.getHp() %>">Edit</a>
91                  <a href="./proc/deleteProc.jsp?uid=<%=mb.getUid() %>">Delete</a>
92              </td>
93          </tr>
94          <%} %>
95      </table>
96
97  </body>
98  </html>
```

3. UPDATE

Components

○ Modify.JSP

○ modifyProc.JSP

Modify employee information

ID	null
Name	null
Phone	null
Position	associate ∨
Department	1 ∨
	modify

3. UPDATE - MODIFY.JSP

- Make the same table as registering to update the data selected from the list with a modify button.

- Place the table under form tag which leads to modifyProc JSP file as POST method.

```jsp
1  <%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"%>
2  <%
3      //encoding data
4      request.setCharacterEncoding("utf-8");
5
6      //receiving data
7      String uid = request.getParameter("uid");
8      String name = request.getParameter("name");
9      String hp = request.getParameter("hp");
10 %>
11 <!DOCTYPE html>
12 <html>
13 <head>
14     <meta charset="UTF-8">
15     <title>Modify</title>
16 </head>
17 <body>
18     <h3>Modify employee information</h3>
19
20     <form action="./proc/modifyProc.jsp" method="post">
21         <table border="1">
22             <tr>
23                 <td>ID</td>
24                 <td><input type="text" name="uid" readonly value="<%=uid%>"/></td>
25             </tr>
26             <tr>
27                 <td>Name</td>
28                 <td><input type="text" name="name" value="<%=name%>"/></td>
29             </tr>
30             <tr>
31                 <td>Phone</td>
32                 <td><input type="tel" name="hp" value="<%=hp%>"/></td>
33             </tr>
34             <tr>
35                 <td>Position</td>
36                 <td>
37                     <select name="pos">
38                         <option>associate</option>
39                         <option>manager class2</option>
40                         <option>manager class1</option>
41                         <option>deputy manager</option>
42                         <option>head manager</option>
43                     </select>
44                 </td>
45             </tr>
46             <tr>
47                 <td>Department</td>
48                 <td>
49                     <select name="dep">
50                         <option value="101">1</option>
51                         <option value="102">2</option>
52                         <option value="103">3</option>
53                         <option value="104">4</option>
54                         <option value="105">5</option>
55                         <option value="106">6</option>
56                         <option value="107">7</option>
57                     </select>
58                 </td>
59             </tr>
60             <tr>
61                 <td colspan="2" align="right">
62                     <input type="submit" value="modify">
63                 </td>
64             </tr>
65         </table>
66     </form>
67 </body>
68 </html>
```

3. UPDATE

- MODIFYPROC.JSP

- Processing SQL statements with JDBC

  1. Encoding data as UTF-8.

  2. Receiving data as variables.

  3. Establishing a connection with HeidiSQL.

  4. Create a statement to update the data of member table.

  5. Update the query.

  6. Close the connection.

```jsp
1  <%@page import="java.sql.Connection"%>
2  <%@page import="java.sql.Statement"%>
3  <%@page import="java.sql.DriverManager"%>
4  <%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"%>
5  <%
6      //encoding data
7      request.setCharacterEncoding("utf-8");
8
9      //receiving data
10     String uid = request.getParameter("uid");
11     String name = request.getParameter("name");
12     String hp = request.getParameter("hp");
13     String pos = request.getParameter("pos");
14     String dep = request.getParameter("dep");
15
16     //database
17     String host = "jdbc:mysql://192.168.10.114:3306/kye";
18     String user = "kye";
19     String pass = "1234";
20
21     //1.load JDBC driver
22     Class.forName("com.mysql.jdbc.Driver");
23
24     //2.access to the dadtabase
25     Connection conn = DriverManager.getConnection(host,user,pass);
26
27     //3.make SQL object
28     Statement stmt = conn.createStatement();
29
30     //4. sql
31     String sql = "UPDATE `MEMBER` SET ";
32         sql += "`name`='"+name+"', ";
33         sql += "`hp`='"+hp+"', ";
34         sql += "`pos`='"+pos+"', ";
35         sql += "`dep`="+dep+" ";
36         sql += "WHERE `uid`='"+uid+"', ";
37
38     stmt.executeUpdate(sql);
39
40     //5.process the result(if SELECT)
41     //6.exit database
42     stmt.close();
43     conn.close();
44
45
46     //redirect
47     response.sendRedirect("../List.jsp");
```

4. DELETE

Components

o Modify.JSP

o deleteProc.JSP

4. DELETE
- DELETEPROC.JSP

- Processing SQL statements with JDBC
  1. Encoding data as UTF-8.
  2. Receiving data as variables.
  3. Establishing a connection with HeidiSQL.
  4. Create a statement to delete the data of the specific uid retrieved from List.JSP.
  5. Update the query.
  6. Close the connection.
  7. Redirect to List page.

```jsp
1  <%@page import="java.sql.Connection"%>
2  <%@page import="java.sql.Statement"%>
3  <%@page import="java.sql.DriverManager"%>
4  <%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"%>
5  <%
6      //encoding data
7      request.setCharacterEncoding("utf-8");
8
9      //receiving data
10     String uid = request.getParameter("uid");
11
12     //database
13     String host = "jdbc:mysql://192.168.10.114:3306/kye";
14     String user = "kye";
15     String pass = "1234";
16
17     //1.load JDBC driver
18     Class.forName("com.mysql.jdbc.Driver");
19
20     //2.access to the dadtabase
21     Connection conn = DriverManager.getConnection(host,user,pass);
22
23     //3.make SQL object
24     Statement stmt = conn.createStatement();
25
26     //4. sql
27     String sql = "DELETE FROM `MEMBER` WHERE `uid`= '"+uid+"';";
28     stmt.executeUpdate(sql);
29
30     //5.process the result(if SELECT)
31     //6.exit database
32     stmt.close();
33     conn.close();
34
35     //redirect
36     response.sendRedirect("../List.jsp");
37
38  %>
```