

```
1  <%--
2    Created by IntelliJ IDEA.
3    User: mhm85
4    Date: 10/14/2019
5    Time: 3:24 PM
6    To change this template use File | Settings |
   File Templates.
7  --%>
8  <%@ page contentType="text/html; charset=UTF-8"
   language="java" %>
9  <html>
10 <head>
11     <title>$Title$</title>
12 </head>
13 <body>
14 <form action="doSomething" method="post">
15     <input type="hidden" name="fromPage" value="
   page1">
16     <label>First name:</label>
17     <input type="text" name="fn" required><br>
18     <label>Last name:</label>
19     <input type="text" name="ln" required><br>
20     <label>Username:</label>
21     <input type="text" name="un" required><br>
22     <label>Password:</label>
23     <input type="password" name="pw" required><br>
24     <input type="submit" value="Sign Up">
25 </form>
26 </body>
27 </html>
28
```

```
1 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/
  jstl/core" %>
2 <%--
3   Created by IntelliJ IDEA.
4   User: mhm85
5   Date: 11/20/2019
6   Time: 2:23 PM
7   To change this template use File | Settings |
  File Templates.
8 --%>
9 <%@ page contentType="text/html; charset=UTF-8"
  language="java" %>
10 <html>
11 <head>
12     <title>Index 2</title>
13 </head>
14 <body>
15 <table border="1">
16     <c:forEach items="${users}" var="user"
  varStatus="loopCounter">
17         <tr>
18             <td>
19                 <form action="doSomething" method="
  post">
20                     <input type="hidden" name="
  fromPage" value='<c:out value="${user.idUser}"/>'>
21                     <label>First name:</label><c:
  out value="${user.fnUser}"/><br>
22                     <label>Last name:</label><c:out
  value="${user.lnUser}"/><br>
23                     <label>Username:</label><c:out
  value="${user.unUser}"/><br>
24                     <label>Password:</label><c:out
```

```
24 value="${user.pwUser}" /><br>
25         <input type="submit" value="
    Send">
26             </form>
27         </td>
28         <c:if test="${loopCounter.count % 2 ==
    0}">
29             <tr></tr>
30         </c:ifc:forEach
```

```
1 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/
  jstl/core" %>
2 <%--
3   Created by IntelliJ IDEA.
4   User: mhm85
5   Date: 10/16/2019
6   Time: 3:00 PM
7   To change this template use File | Settings |
  File Templates.
8 --%>
9 <%@ page contentType="text/html; charset=UTF-8"
  language="java" %>
10 <html>
11 <head>
12     <title>Title</title>
13 </head>
14 <body>
15 ${aDude.fnUser} ${aDude.lnUser} <br>
16 <table border="1">
17     <tr>
18         <th>Id</th>
19         <th>First Name</th>
20         <th>Last Name</th>
21         <th>Username</th>
22         <th>Password</th>
23     </tr>
24     <c:forEach items="${users}" var="user">
25         <tr>
26             <td><c:out value="${user.idUser}"/></td>
27             <td><c:out value="${user.fnUser}"/></td>
28             <td><c:out value="${user.lnUser}"/></td>
```

```
28 >
29         <td><b:c:out value="\${user.unUser}"/></td>
30         >
31         <td><b:c:out value="\${user.pwUser}"/></td>
32         >
33     </tr>
34 </c:forEach>
35 </table>
36 </body>
37 </html>
38
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3         xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
4         xsi:schemaLocation="http://maven.apache.
org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0
.0.xsd">
5     <modelVersion>4.0.0</modelVersion>
6
7     <groupId>groupId</groupId>
8     <artifactId>maven_testing</artifactId>
9     <version>1.0-SNAPSHOT</version>
10
11     <dependencies>
12         <!--<dependency>
13             <groupId>javax.servlet</groupId>
14             <artifactId>javax.servlet-api</
artifactId>
15             <version>3.1.0</version>
16         </dependency>-->
17         <dependency>
18             <groupId>mysql</groupId>
19             <artifactId>mysql-connector-java</
artifactId>
20             <version>5.1.48</version>
21         </dependency>
22         <dependency>
23             <groupId>jstl</groupId>
24             <artifactId>jstl</artifactId>
25             <version>1.2</version>
26         </dependency>
27     </dependencies>
28
```

```
29      <!--<packaging>war</packaging>-->  
30 </project>
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
3         xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
4         xsi:schemaLocation="http://xmlns.jcp.org/
xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/
web-app_4_0.xsd"
5         version="4.0">
6
7     <servlet>
8         <servlet-name>MyServlet</servlet-name>
9         <servlet-class>MyServlet</servlet-class>
10    </servlet>
11    <servlet-mapping>
12        <servlet-name>MyServlet</servlet-name>
13        <url-pattern>/</url-pattern>
14    </servlet-mapping>
15
16    <listener>
17        <listener-class>MyListener</listener-class>
18    </listener>
19
20 </web-app>
```



```
1 import java.io.Serializable;
2
3 public class User implements Serializable {
4     private int idUser;
5     private String fnUser;
6     private String lnUser;
7     private String unUser;
8     private String pwUser;
9
10    public User(int idUser, String fnUser, String
    lnUser, String unUser, String pwUser) {
11        this.idUser = idUser;
12        this.fnUser = fnUser;
13        this.lnUser = lnUser;
14        this.unUser = unUser;
15        this.pwUser = pwUser;
16    }
17
18    public User() {
19        idUser = 0;
20        fnUser = "";
21        lnUser = "";
22        unUser = "";
23        pwUser = "";
24    }
25
26    public int getIdUser() {
27        return idUser;
28    }
29
30    public void setIdUser(int idUser) {
31        this.idUser = idUser;
32    }
```

```
33
34     public String getFnUser() {
35         return fnUser;
36     }
37
38     public void setFnUser(String fnUser) {
39         this.fnUser = fnUser;
40     }
41
42     public String getLnUser() {
43         return lnUser;
44     }
45
46     public void setLnUser(String lnUser) {
47         this.lnUser = lnUser;
48     }
49
50     public String getUnUser() {
51         return unUser;
52     }
53
54     public void setUnUser(String unUser) {
55         this.unUser = unUser;
56     }
57
58     public String getPwUser() {
59         return pwUser;
60     }
61
62     public void setPwUser(String pwUser) {
63         this.pwUser = pwUser;
64     }
65 }
```

```
1 import java.sql.*;
2 import java.util.ArrayList;
3
4 public class DatabaseTools {
5     public static Connection getConnected() {
6         Connection conn = null;
7
8         try {
9             //Load MySQL database driver
10             Class.forName("com.mysql.jdbc.Driver");
11
12             // Connection string
13             String dbURL = "jdbc:mysql://localhost:
14 3306/max";
15
16             // Credentials
17             String username = "root";
18             String password = "mysql";
19
20             // Set connection to database to
21             Connection object
22             conn = DriverManager.getConnection(
23 dbURL, username, password);
24         } catch (ClassNotFoundException e) {
25             e.printStackTrace();
26         } catch (SQLException e) {
27             e.printStackTrace();
28         }
29
30         return conn;
31     }
32
33     public static void insert(String fn, String ln,
```

```
30 String un, String pw) {
31     // Connection to database;
32     Connection myConn = DatabaseTools.
    getConnected();
33
34     PreparedStatement ps = null;
35
36     // Insert query
37     String queryInsert = "INSERT INTO tbl_users
    (user_fn, user_ln, user_un, user_pw) VALUES
    (?, ?, ?, ?);";
38
39     try {
40         // Set PreparedStatement object to
    instance with query
41         ps = myConn.prepareStatement(
    queryInsert);
42
43         // Set values for parameters
44         ps.setString(1, fn);
45         ps.setString(2, ln);
46         ps.setString(3, un);
47         ps.setString(4, pw);
48
49         // Execute insert on database
50         ps.execute();
51     } catch (SQLException e) {
52         e.printStackTrace();
53     } finally {
54         DatabaseTools.closePreparedStatement(ps
    );
55         DatabaseTools.closeConnection(myConn);
56     }
```

```
57     }
58
59     public static ArrayList<User> selectAllUsers()
60     {
61         // Connection to database
62         Connection conn = DatabaseTools.
63         getConnected();
64
65         PreparedStatement ps = null;
66         ResultSet rs = null;
67         ArrayList<User> myUserList = new ArrayList<
68         User>();
69
70         // Select statement
71         String selectQuery = "SELECT user_id,
72         user_fn, user_ln, user_un, user_pw FROM tbl_users;"
73         ;
74
75         try {
76             // Create prepared statement object
77             ps = conn.prepareStatement(selectQuery)
78             ;
79
80             // Execute query and return result set
81             rs = ps.executeQuery();
82
83             // Loop through all rows in result set
84             while (rs.next()) {
85                 // Create user object to add to
86                 list.
87                 User user = new User();
88
89                 // Add values for row of result set
```

```
82  to properties of User object
83      user.setIdUser(rs.getInt("user_id"
   ));
84      user.setFnUser(rs.getString("
   user_fn"));
85      user.setLnUser(rs.getString("
   user_ln"));
86      user.setUnUser(rs.getString("
   user_un"));
87      user.setPwUser(rs.getString("
   user_pw"));
88
89      // Add user to list
90      myUserList.add(user);
91  }
92  } catch (SQLException e) {
93      e.printStackTrace();
94  } finally {
95      DatabaseTools.closePreparedStatement(
   ps);
96      DatabaseTools.closeResultSet(rs);
97      DatabaseTools.closeConnection(conn);
98  }
99
100     return myUserList;
101 }
102
103     public static void closePreparedStatement(
   Statement ps) {
104         try {
105             if (ps != null) {
106                 ps.close();
107             }
```

```
108         } catch (SQLException e) {
109             e.printStackTrace();
110         }
111     }
112
113     public static void closeConnection(Connection
conn) {
114         try {
115             if (conn != null) {
116                 conn.close();
117             }
118         } catch (SQLException e) {
119             e.printStackTrace();
120         }
121     }
122
123     public static void closeResultSet(ResultSet rs
) {
124         try {
125             if (rs != null) {
126                 rs.close();
127             }
128         } catch (SQLException e) {
129             e.printStackTrace();
130         }
131     }
132 }
133
```

```
1 import java.security.MessageDigest;
2 import java.security.NoSuchAlgorithmException;
3
4 public class SecurePassword {
5     public static String lockPassword(String pw) {
6         String newPw = null;
7
8         try {
9             MessageDigest md = MessageDigest.
10 getInstance("MD5");
11
12             md.update(pw.getBytes());
13
14             byte[] bytes = md.digest();
15
16             StringBuilder sb = new StringBuilder();
17
18             for(int i = 0; i < bytes.length; i++) {
19                 sb.append(Integer.toString((bytes[i]
20 ] & 0xff) + 0x100, 16).substring(1));
21
22                 newPw = sb.toString();
23             } catch (NoSuchAlgorithmException e) {
24                 e.printStackTrace();
25             }
26
27             return newPw;
28         }
29     }
```



```
1 import javax.servlet.ServletContextEvent;
2 import javax.servlet.ServletContextListener;
3 import javax.servlet.annotation.WebListener;
4 import javax.servlet.http.
  HttpSessionAttributeListener;
5 import javax.servlet.http.HttpSessionEvent;
6 import javax.servlet.http.HttpSessionListener;
7 import javax.servlet.http.HttpSessionBindingEvent;
8 import java.util.ArrayList;
9
10 @WebListener()
11 public class MyListener implements
  ServletContextListener,
12      HttpSessionListener,
  HttpSessionAttributeListener {
13
14      // Public constructor is required by servlet
  spec
15      public MyListener() {
16      }
17
18
19      // -----
20      // ServletContextListener implementation
21
22      // -----
23      public void contextInitialized(
  ServletContextEvent sce) {
24          /* This method is called when the servlet
  context is
25          initialized(when the Web application is
```

```
23 deployed) .
24         You can initialize servlet context related
        data here.
25         */
26
27         // Get users from database.
28         ArrayList<User> users = DatabaseTools.
selectAllUsers();
29
30         // Get servlet context from event object and
set array list of users as attribute.
31         sce.getServletContext().setAttribute("users",
        users);
32
33     }
34
35     public void contextDestroyed(
ServletContextEvent sce) {
36         /* This method is invoked when the Servlet
Context
37         (the Web application) is undeployed or
38         Application Server shuts down.
39         */
40     }
41
42
43     // -----
44     // HttpSessionListener implementation
45     // -----
46
47     public void sessionCreated(HttpSessionEvent se)
```

```
45  {
46      /* Session is created. */
47  }
48
49  public void sessionDestroyed(HttpSessionEvent
se) {
50      /* Session is destroyed. */
51  }
52
53
54  // -----
55  // HttpSessionAttributeListener implementation
56  // -----
57
58  public void attributeAdded(
HttpSessionBindingEvent sbe) {
59      /* This method is called when an attribute
60      is added to a session.
61      */
62  }
63
64  public void attributeRemoved(
HttpSessionBindingEvent sbe) {
65      /* This method is called when an attribute
66      is removed from a session.
67      */
68  }
69
70  public void attributeReplaced(
HttpSessionBindingEvent sbe) {
```

```
70      /* This method is invoked when an attribute
71          is replaced in a session.
72      */
73  }
74 }
75
```

```
1  import javax.servlet.ServletException;
2  import javax.servlet.annotation.WebServlet;
3  import javax.servlet.http.HttpServlet;
4  import javax.servlet.http.HttpServletRequest;
5  import javax.servlet.http.HttpServletResponse;
6  import java.io.IOException;
7  import java.util.ArrayList;
8
9  @WebServlet(name = "MyServlet")
10 public class MyServlet extends HttpServlet {
11     protected void doPost(HttpServletRequest
        request, HttpServletResponse response) throws
        ServletException, IOException {
12
13         String url = "";
14         String action = request.getParameter("
        fromPage");
15
16         System.out.println(action);
17
18         if (action.equals("page1")) {
19             String fn = request.getParameter("fn");
20             String ln = request.getParameter("ln");
21             String un = request.getParameter("un");
22             String pw = request.getParameter("pw");
23
24             User u1 = new User(111, fn, ln, un, pw)
        ;
25
26             request.setAttribute("aDude", u1);
27
28             // Encrypt password
29             pw = SecurePassword.lockPassword(pw);
```

```
30
31         // Insert user into database
32         DatabaseTools.insert(fn, ln, un, pw);
33
34         // Get all users from database
35         ArrayList<User> myData = DatabaseTools.
selectAllUsers();
36
37         // Set list of users as attribute of
request
38         request.setAttribute("users", myData);
39
40         url = "/page2.jsp";
41     }
42     else if (action.equals("1")) {
43
44     }
45
46     getServletContext().getRequestDispatcher(
url).forward(request, response);
47
48     }
49
50     protected void doGet(HttpServletRequest request
, HttpServletResponse response) throws
ServletException, IOException {
51         doPost(request, response);
52     }
53 }
54
```