

```
1 import java.io.*;
2 import java.sql.*;
3 import javax.servlet.*;
4 import javax.servlet.http.*;
5
6 public class LuoRyhma extends HttpServlet {
7
8     final String dbDriver="org.postgresql.Driver";
9     final String dbServer ="jdbc:postgresql://localhost:10388/tsoha";
10    final String dbUser= "avtanska"; // replace with your db user account
11    final String dbPassword ="postgres"; // replace with your password
12
13
14    public void service(HttpServletRequest req, HttpServletResponse res)
15        throws ServletException, IOException {
16        ServletOutputStream out;
17        res.setContentType("text/html");
18        out= res.getOutputStream();
19
20        String submit = req.getParameter("submit");
21        HttpSession session = req.getSession(false);
22        String sessionTunnus = (String)session.getValue("sessionTunnus");
23        String[] jäsenet = req.getParameterValues("jasenet");
24        String ryhmä = req.getParameter("ryhma");
25        String virhe;
26
27
28        /*
29         * Tarkistetaan onko tarvittavat tiedot täytetty
30         */
31
32        if (submit != null && (jäsenet == null || ryhmä.length() == 0)) {
33            virhe = "Anna ryhmän nimi ja valitse jäsenet!";
34        }
35        else {
36            virhe = null;
37        }
38
39        out.println("<html><head><title></title>" +
40            "<link rel='stylesheet' type='text/css' " +
41            "href='http://db.cs.helsinki.fi/u/avtanska/tsoha/' +
42            "perus.css' /><body>");
43
44        /*
45         * Jos tullaan sivulle 'Luo ryhmä'-linkin kautta tai lähetetään
46         * virheellinen lomake.
47         */
48
49        if (submit == null || (submit != null && virhe != null)) {
50
51            out.println("<form method='post' action='http://db.cs.helsinki.fi/" +
52                "s/avtanska/LuoRyhma'>");
53
54            if (virhe != null) {
55                out.println("<p><b>" + virhe + "</b></p>");
56            }
57
58            out.println("<h3>Ryhmän luonti:</h3>");
59            out.println("<table><tr><td>Ryhmän nimi:</td><td><input name='ryhma'" +
60                " type='text'"></td></tr>");
61            out.println("<tr><td valign='top'>Valitse jäsenet:</td><td><select " +
62                "multiple name='jasenet'">");
63
64            Connection con=null;
65            con= createDbConnection(dbDriver,dbServer,dbUser,dbPassword,out);
66            if (con==null) {
67                out.println("</body></html>");
68                return;
69            }
70        }
71    }
```

```
70
71     /*
72     * Haetaan tietokannasta kaikki henkilöt, admin-käyttäjää
73     * ei tulosteta listaan
74     */
75
76     Statement stmt = null;
77     ResultSet rs = null;
78     try {
79         stmt = con.createStatement();
80         rs = stmt.executeQuery("SELECT * FROM henkilö ORDER BY tunnus");
81         while(rs.next()) {
82             if(!(rs.getString("tunnus")).equals("admin")) {
83                 out.println("<option value='" + rs.getString("tunnus") + "'>" +
84                     rs.getString("etunimi") + " " + rs.getString("sukunimi"));
85             }
86         }
87     } catch (SQLException ee) {
88         out.println("Tietokantavirhe "+ee.getMessage());
89     } finally {
90         try {
91             if (rs!=null) rs.close();
92             if (stmt!=null) stmt.close();
93             con.close();
94         } catch (SQLException e) {
95             out.println("An SQL Exception was thrown.");
96         }
97     }
98 }
99
100 out.println("</td></tr><tr><td>&nbsp;</td></tr><tr><td colspan='2'>" +
101     "<input type='submit' name='submit' value='Luo ryhmä'>" +
102     "<input type='reset' name='reset' value='Tyhjennä'>" +
103     "</td></tr></table></form></body></html>");
104
105 } // loppu if submit null
106
107
108     /*
109     * Jos lomake lähetetty
110     */
111
112     else {
113
114         Connection con = null;
115         con= createDbConnection(dbDriver,dbServer,dbUser,dbPassword,out);
116
117         if (con==null) {
118             out.println("</body></html>");
119             return;
120         }
121
122         Statement stmt = null;
123         ResultSet rs = null;
124         try {
125             stmt = con.createStatement();
126
127
128             /*
129             * Tarkistetaan, onko halutunnimistä ryhmää jo olemassa
130             */
131
132             rs = stmt.executeQuery("SELECT nimi FROM ryhmä WHERE nimi='" +
133                 ryhmä + "'");
134
135             if (rs.next()) {
136                 out.println("<span class='virhe'>Ryhmä on jo olemassa</b>");
137             }
138             else {
```

```
139
140      /*
141      * Luodaan ryhmä ja jäsenyydet halutuille jäsenille
142      */
143
144      String insertRyhmä =
145          "INSERT INTO ryhmä VALUES ('" + ryhmä +
146          "','" + sessionTunnus + "')";
147
148      int count = stmt.executeUpdate(insertRyhmä);
149
150      String insertJäsenyys = "";
151
152      for (int i = 0; i < jäsenet.length; i++) {
153          insertJäsenyys =
154              "INSERT INTO jäsenyys VALUES ('" + jäsenet[i] +
155              "','" + ryhmä + "')";
156
157          count = stmt.executeUpdate(insertJäsenyys);
158      }
159
160      out.println("<h3>Ryhmän luonti onnistui.</h3>" +
161          "<p>Ryhmää ei vielä näy omien ryhmiesi " +
162          "listassa.<br />Pääset käsiksi ryhmän tietoihin, " +
163          "kun kirjoittaudut<br />sisään seuraavan kerran.</p>");
164
165      }
166
167      } catch (SQLException ee) {
168          out.println("Tietokantavirhe "+ee.getMessage());
169      } finally {
170          try {
171              if (rs!=null) rs.close();
172              if (stmt!=null) stmt.close();
173              con.close();
174          } catch (SQLException e) {
175              out.println("An SQL Exception was thrown.");
176          }
177      }
178
179      } // loppu else jos lomake lähetetty
180
181  }
182
183
184
185  private Connection createDbConnection(
186      String dbDriver, String dbServer, String dbUser, String dbPassword,
187      ServletOutputStream out) throws IOException {
188
189      // establish a database connection
190      try{
191          Class.forName(dbDriver);           // load driver
192      } catch (ClassNotFoundException e) {
193          out.println("Couldn't find driver "+dbDriver);
194          return null;
195      }
196      Connection con=null;
197      try {
198          con = DriverManager.getConnection(dbServer,dbUser,dbPassword);
199      } catch (SQLException se) {
200          out.println("Couldn't get connection to "+dbServer+
201              " for "+ dbUser+"<br>");
202          out.println(se.getMessage());
203      }
204      return con;
205  }
206
207  }
```