Fpt University

Session Management

Sessions and Listeners
Technique: Error Handling in Servlets

Review

- **Web Applications**
  - File and Directory Structures (war file)
- **The Web Container Model**
  - The Servlet Container
  - The Servlet Context
  - The Servlet Config
  - Attributes
    - Request, Session, and Context Scope
  - Request Dispatcher (forward, include)
  - Filter (Filter Chain, Wrapper class)
    - Web Application → Servlet, HTML, … → Filter → Process → …

Objectives

- **Sessions and Listeners**
  - Session Life Cycle
  - Session Management
  - Request and Context Listeners
  - Session Listeners
- **Techniques: Error Handling in Servlets**
  - Reporting Errors
  - Logging Errors

Sessions & Listeners

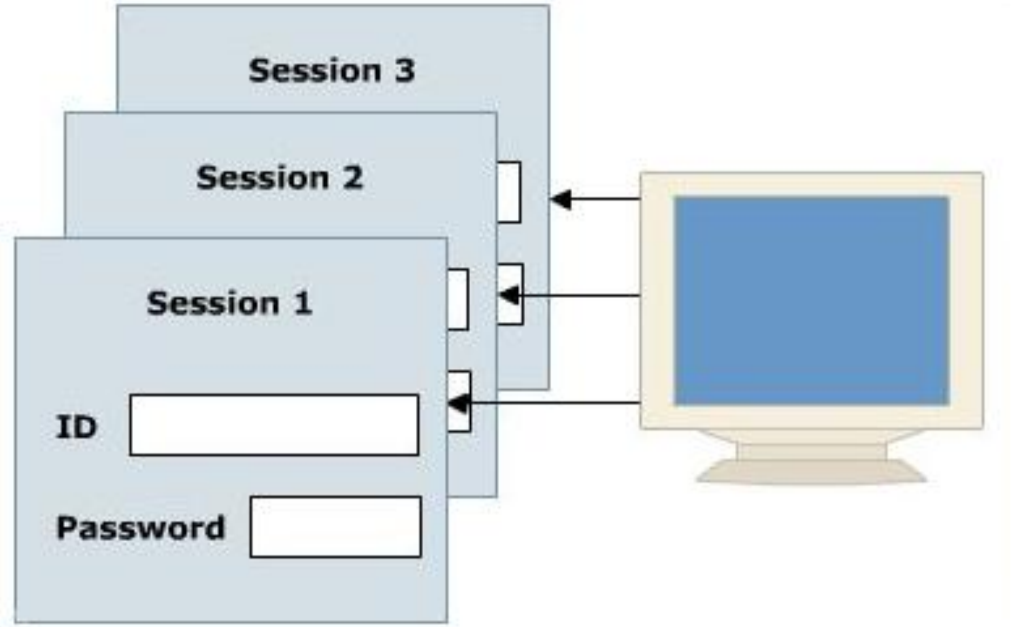
Session

- Is the **period of connection** between client and server
- Is a group of activities that are performed by a user while accessing a particular web site
- HttpSession are **virtual connection** between client and server
- Web container reserves **an individual memory block for storing information about each session → Session objects.**
- **The session tracking** (mechanism)
  - Serves the purpose **tracking** the client identity and other state information required throughout the session
  - Allows the **server to keep a track of successive requests** made by same client
  - Allows **the customer to maintain the information with the server** as long as the customer does not log out from the website

Sessions & Listeners

Session Tracking Techniques

- URL Rewriting
- Hidden form field
- Cookies
- HttpSession interface



Session Tracking

Sessions & Listeners

URL Rewriting

- **Maintains** the **state** of end user by **modifying the URL.**
- **Adds some extra data** at the end of the URL
- Is **used** when the **information to be transferred** is not critical.
- **Syntax**: **url?query_string**
- **Ex**
  - <a href=“http://localhost:8080/Books?category=java”> Java Books </a>
  - <form action=“http://localhost:8080/UpdateProfile?uid=123” method=“get”> ---------- </form>
- A "**pseudo-parameter**" called **jsessionid** is placed in the URL between the servlet name (with path information, if present) and the query string (encodeURL() or encodeRedirectURL() or response)
- **Disadvantages**:
  - Server side **processing is tedious**.
  - Every URL that is **returned** to the **user** should have **additional information appended** to it.
  - If the user **leaves the session** and **opens** the **Web page using a link or bookmark** then the session information is lost.
  - The **query string is limited**

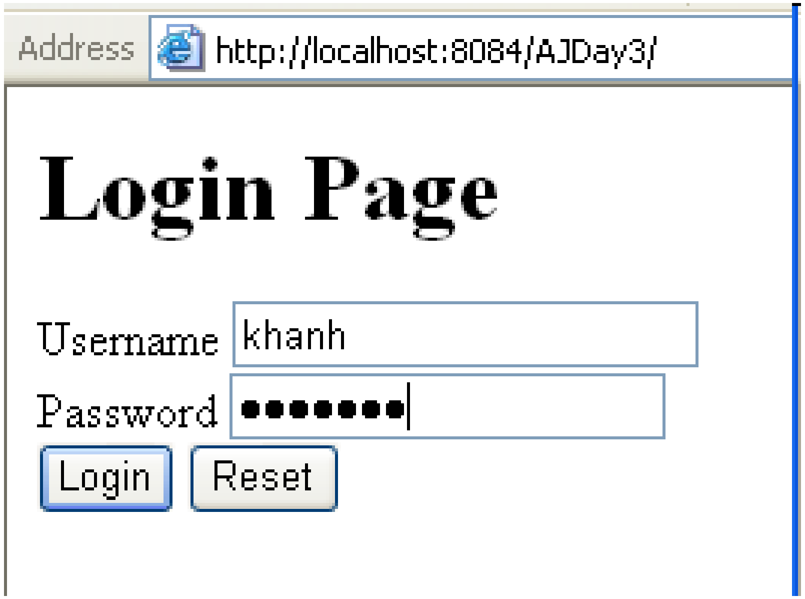
Sessions & Listeners

URL Rewriting – Example

- Building the web application can do some following functions as
  - The user **must be authenticated** before they want to use this web site **using the DB**
  - If the user is invalid, the **message "Invalid username and password" is presented**, then the **link "Click here to try again" is shown** that **redirect the user to the login page**
  - Otherwise, **the search page** is redirected. This page allows user **search appropriate the last name of users**
  - **The result** of searching is **shown in the data grid**. In each row, the **information about ordinary number, username, and last name is shown**
  - Besides, the **data grid allows the user delete the selected row**. After delete action is complete, **the data grid is updated**
- The GUI of web application is present as following

Sessions & Listeners

URL Rewriting – Example

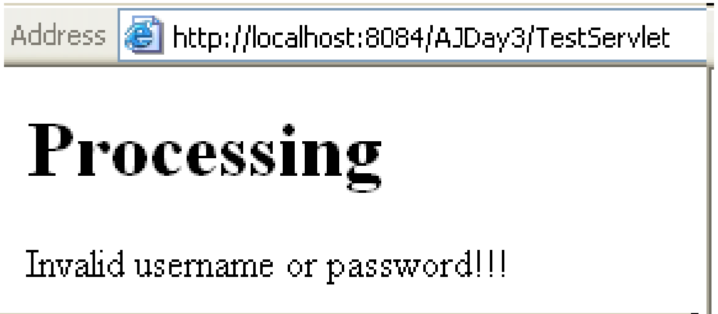


http://localhost:8084/AJDay3/

Login Page

Username: khanh

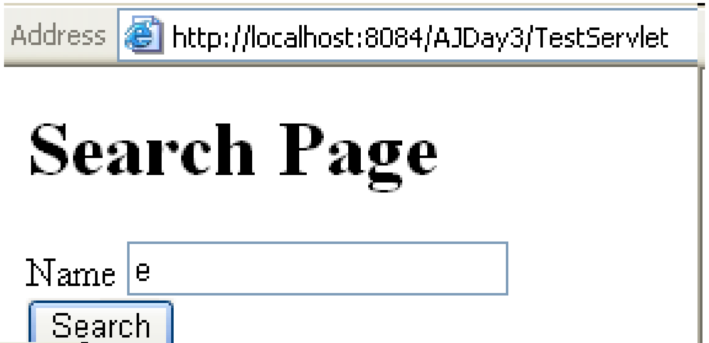
Password: ●●●●●●●●



http://localhost:8084/AJDay3/TestServlet

Processing

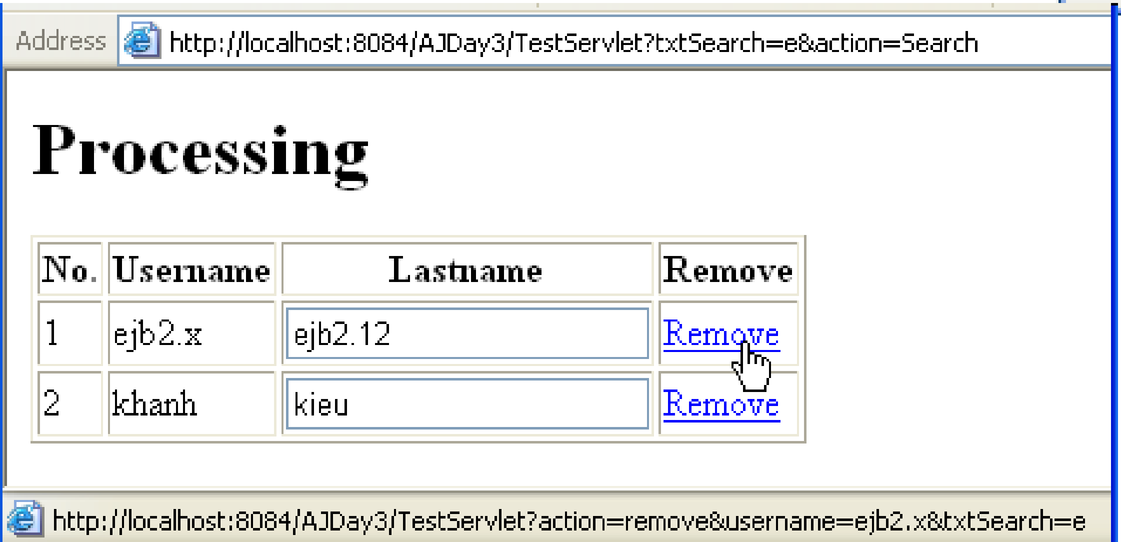
Invalid username or password!!!



http://localhost:8084/AJDay3/TestServlet

Search Page

Name: e



http://localhost:8084/AJDay3/TestServlet?txtSearch=e&action=Search

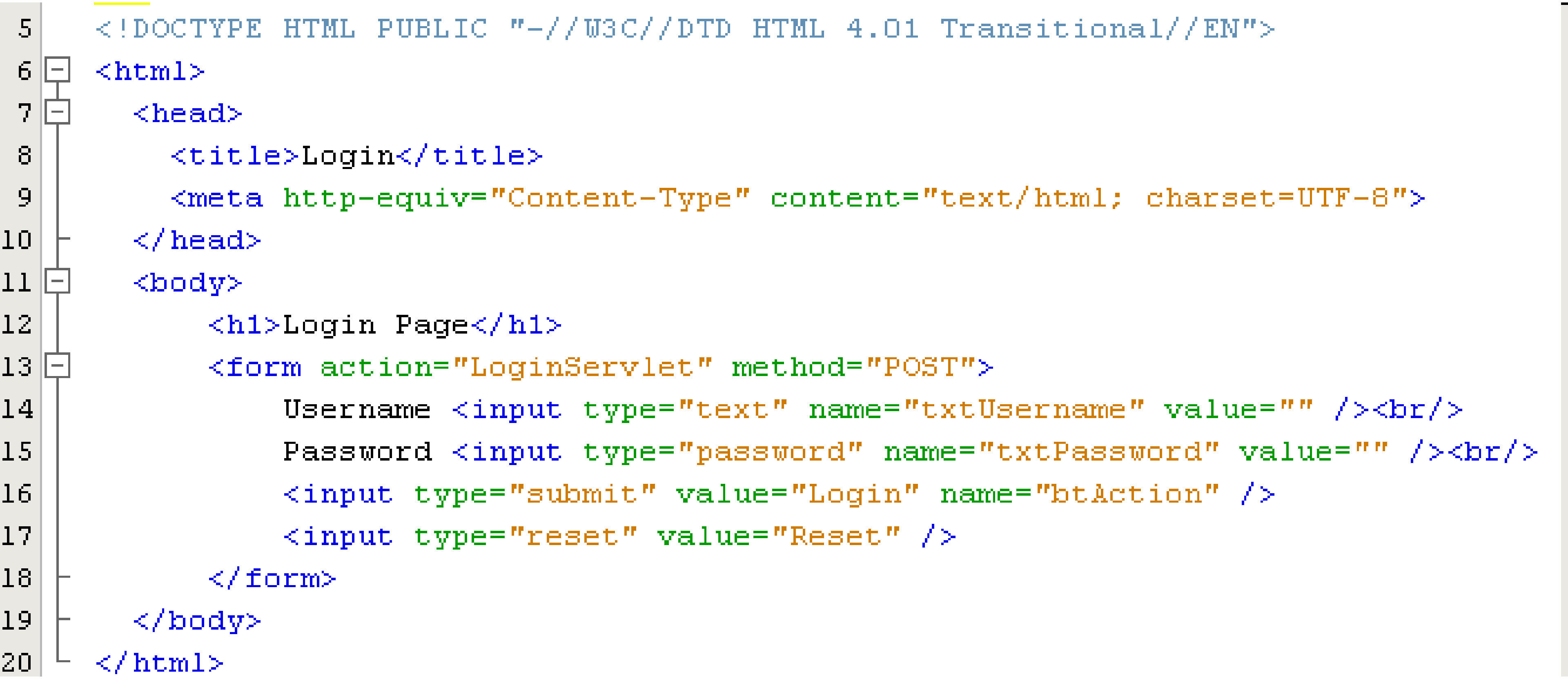
Processing

No.	Username	Lastname	Remove
1	ejb2.x	ejb2.12	Remove
2	khanh	kieu	Remove

http://localhost:8084/AJDay3/TestServlet?action=remove&username=ejb2.x&txtSearch=e

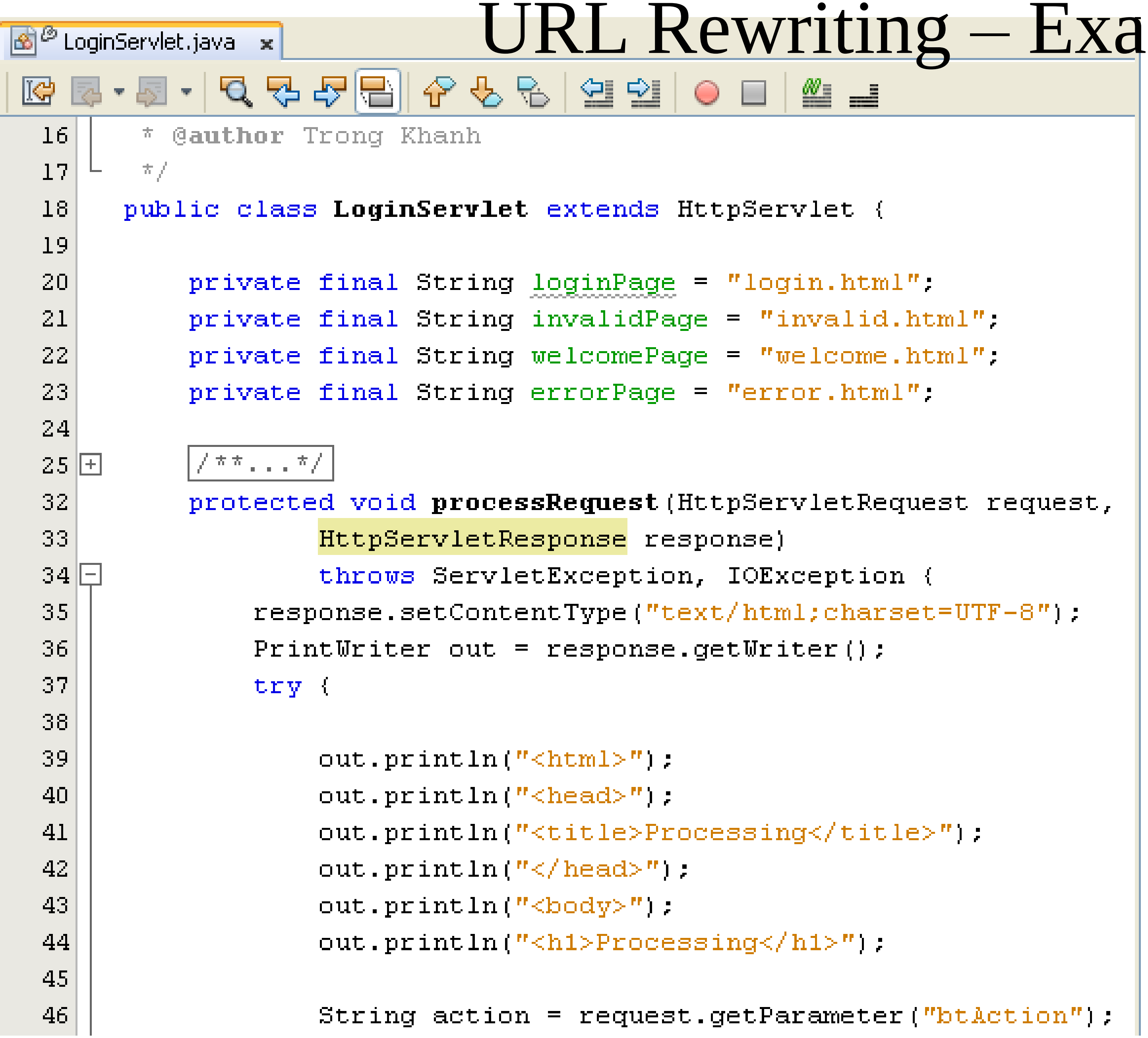
Sessions & Listeners

URL Rewriting – Example



```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Login</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
      <h1>Login Page</h1>
      <form action="LoginServlet" method="POST">
          Username <input type="text" name="txtUsername" value="" /><br/>
          Password <input type="password" name="txtPassword" value="" /><br/>
          <input type="submit" value="Login" name="btAction" />
          <input type="reset" value="Reset" />
      </form>
  </body>
</html>
```

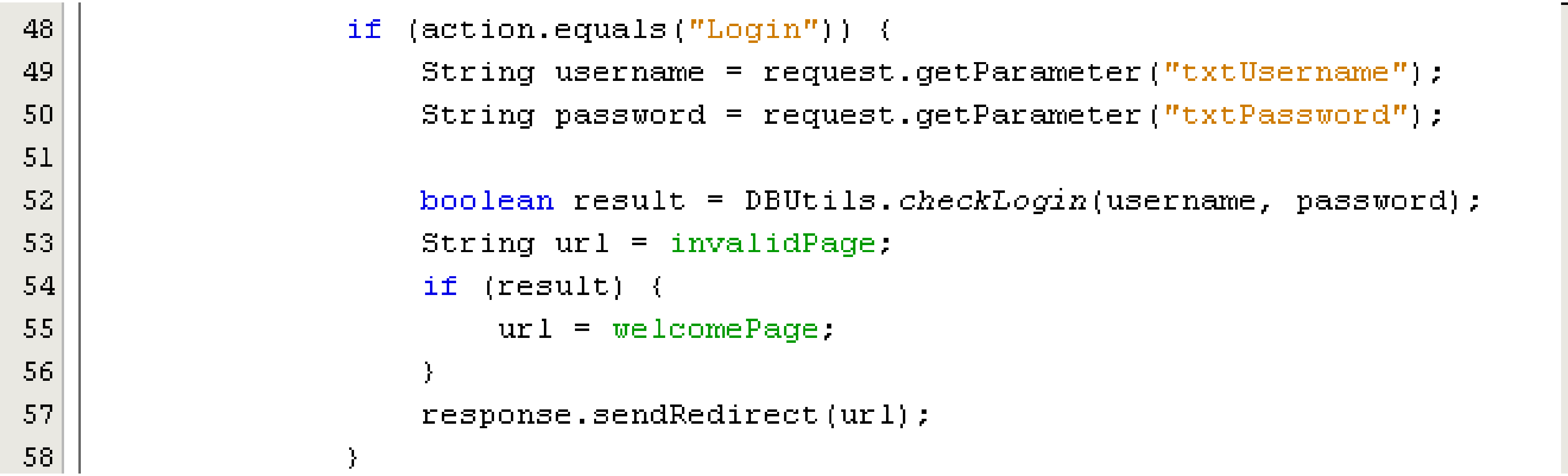
URL Rewriting – Example



```
 * @author Trong Khanh
 */
public class LoginServlet extends HttpServlet {

    private final String loginPage = "login.html";
    private final String invalidPage = "invalid.html";
    private final String welcomePage = "welcome.html";
    private final String errorPage = "error.html";

    /**...*/
    protected void processRequest(HttpServletRequest request,
            HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {

            out.println("<html>");
            out.println("<head>");
            out.println("<title>Processing</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Processing</h1>");

            String action = request.getParameter("btAction");
```

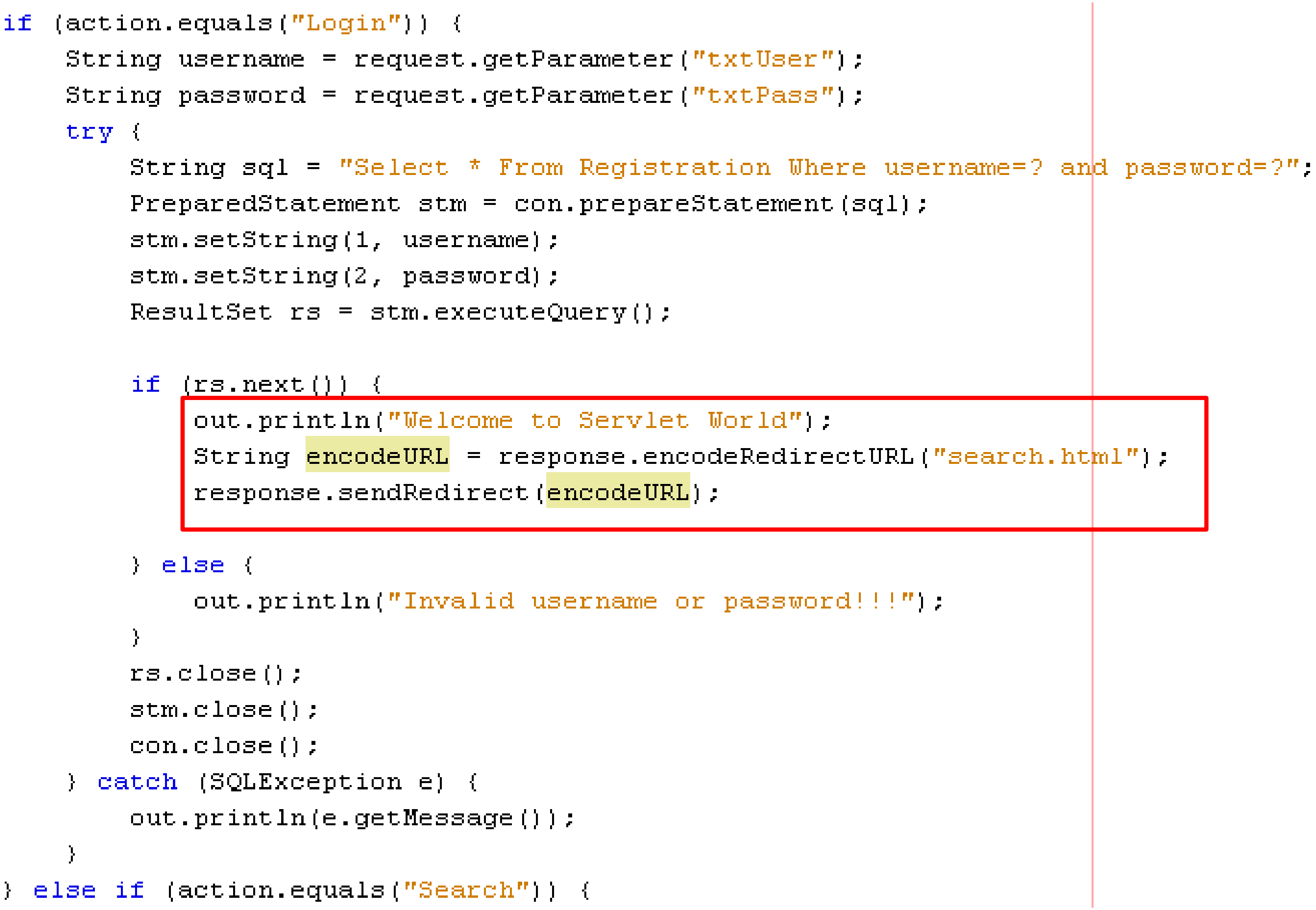
Sessions & Listeners

URL Rewriting – Example – First way



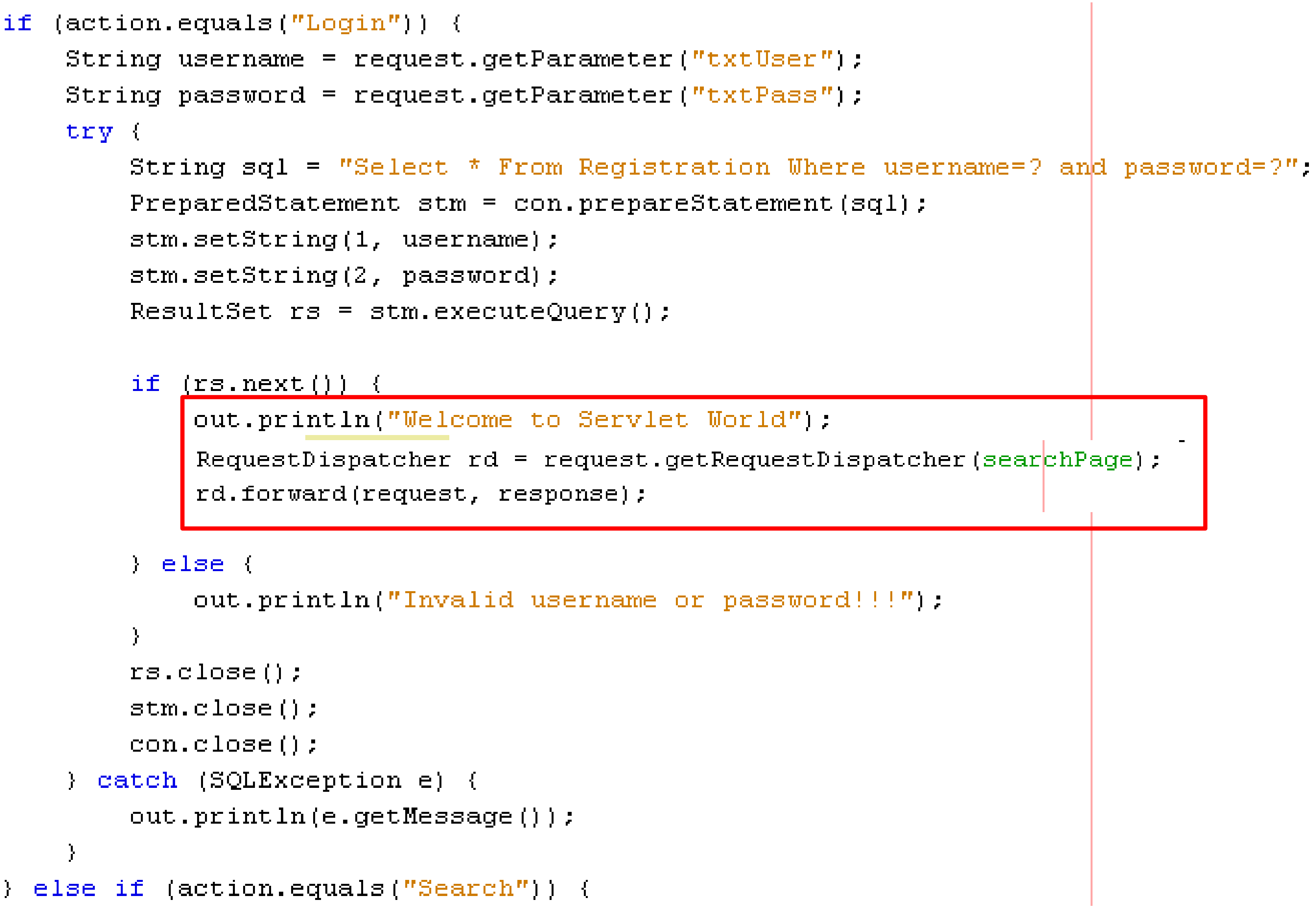
```
if (action.equals("Login")) {
    String username = request.getParameter("txtUsername");
    String password = request.getParameter("txtPassword");

    boolean result = DBUtils.checkLogin(username, password);
    String url = invalidPage;
    if (result) {
        url = welcomePage;
    }
    response.sendRedirect(url);
}
```

URL Rewriting – Example – Second



```
if (action.equals("Login")) {
    String username = request.getParameter("txtUser");
    String password = request.getParameter("txtPass");
    try {
        String sql = "Select * From Registration Where username=? and password=?";
        PreparedStatement stm = con.prepareStatement(sql);
        stm.setString(1, username);
        stm.setString(2, password);
        ResultSet rs = stm.executeQuery();

        if (rs.next()) {
            out.println("Welcome to Servlet World");
            String encodeURL = response.encodeRedirectURL("search.html");
            response.sendRedirect(encodeURL);

        } else {
            out.println("Invalid username or password!!!");
        }
        rs.close();
        stm.close();
        con.close();
    } catch (SQLException e) {
        out.println(e.getMessage());
    }
} else if (action.equals("Search")) {
```

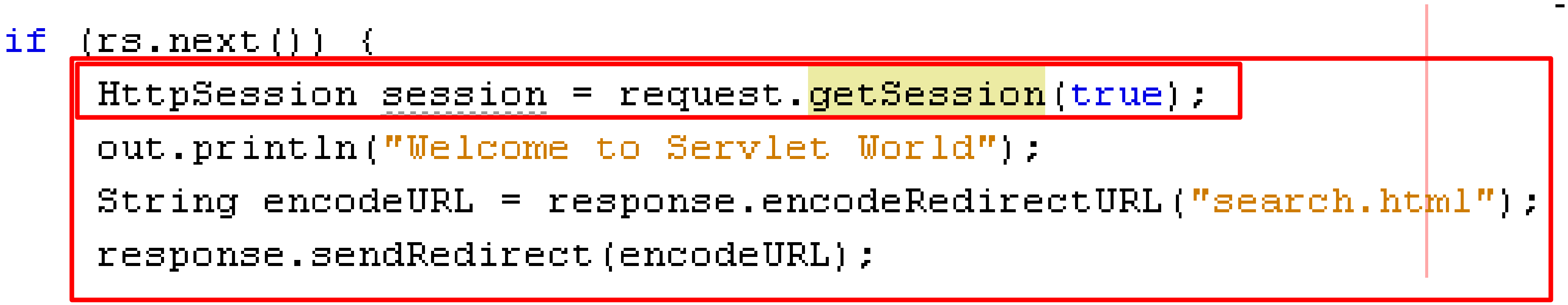
URL Rewriting – Example – Second



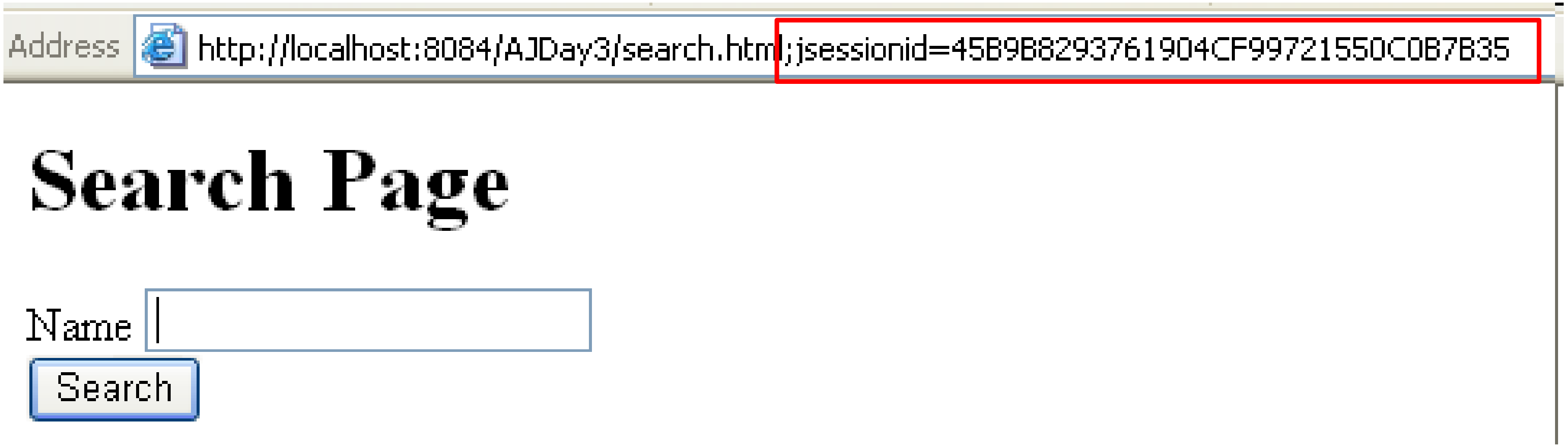
```
if (action.equals("Login")) {
    String username = request.getParameter("txtUser");
    String password = request.getParameter("txtPass");
    try {
        String sql = "Select * From Registration Where username=? and password=?";
        PreparedStatement stm = con.prepareStatement(sql);
        stm.setString(1, username);
        stm.setString(2, password);
        ResultSet rs = stm.executeQuery();

        if (rs.next()) {
            out.println("Welcome to Servlet World");
            RequestDispatcher rd = request.getRequestDispatcher(searchPage);
            rd.forward(request, response);

        } else {
            out.println("Invalid username or password!!!");
        }
        rs.close();
        stm.close();
        con.close();
    } catch (SQLException e) {
        out.println(e.getMessage());
    }
} else if (action.equals("Search")) {
```

Sessions & Listeners

URL Rewriting – Example – Others



```
if (rs.next()) {
    HttpSession session = request.getSession(true);
    out.println("Welcome to Servlet World");
    String encodeURL = response.encodeRedirectURL("search.html");
    response.sendRedirect(encodeURL);
```

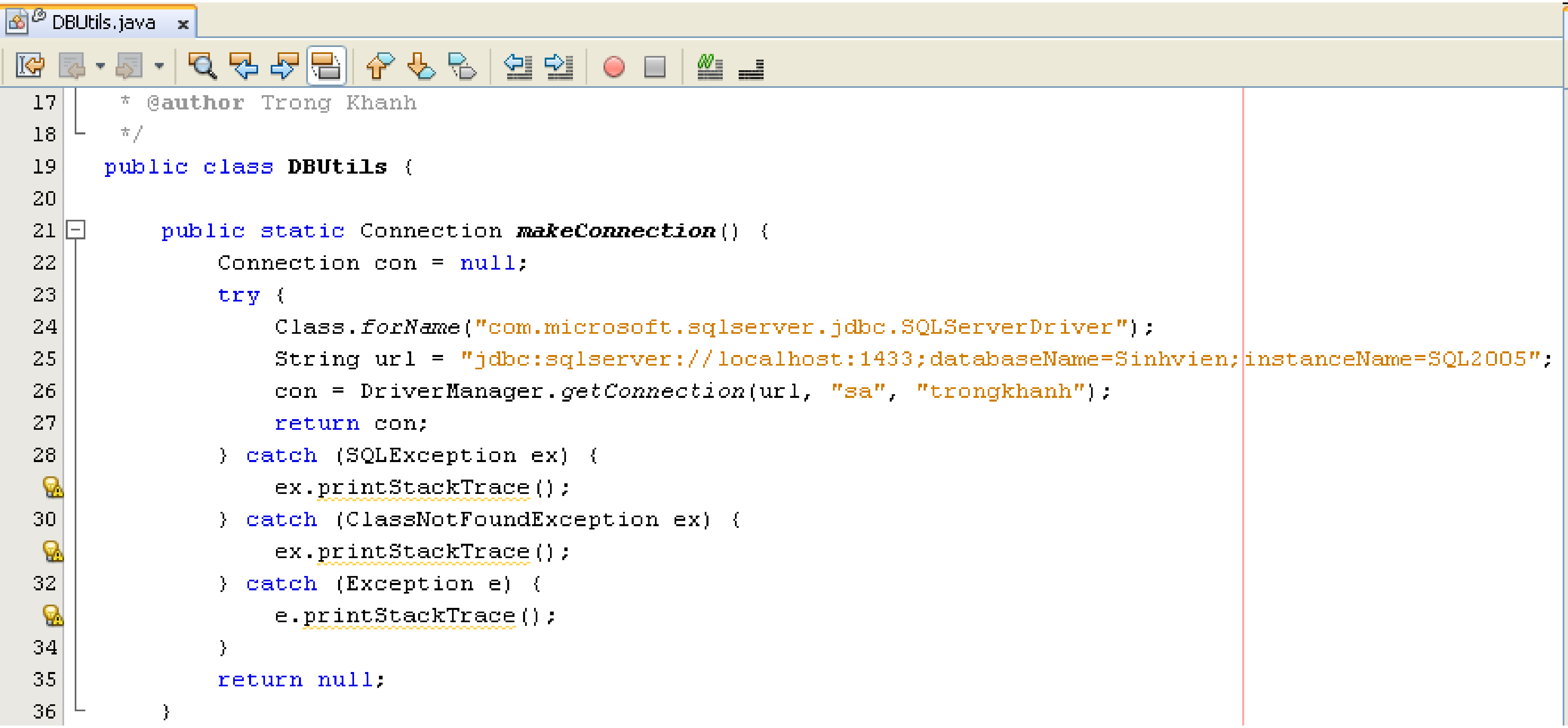


Address http://localhost:8084/AJDay3/search.html;jsessionid=45B9B8293761904CF99721550C0B7B35

Search Page

Sessions & Listeners

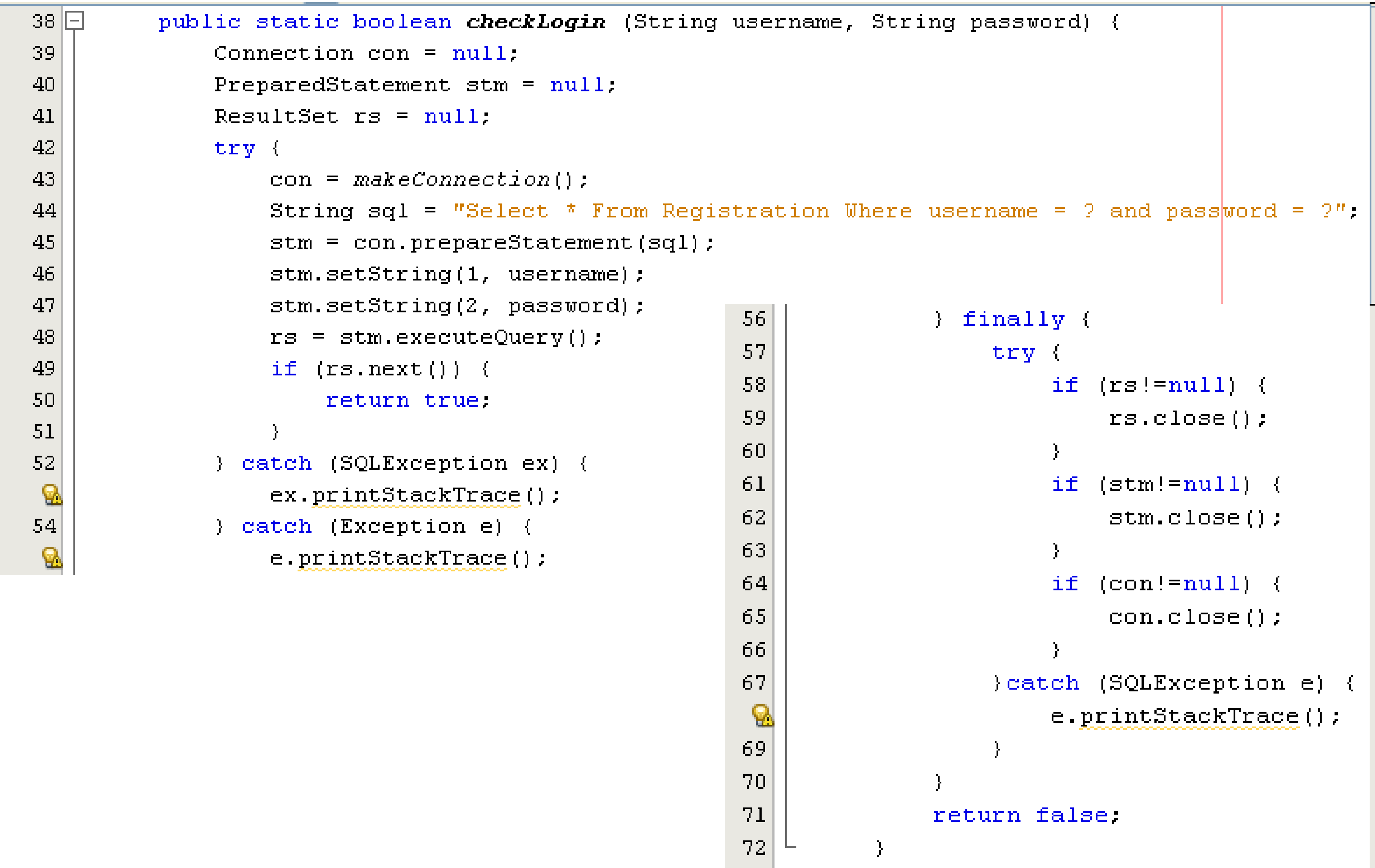
URL Rewriting – Example



```java
 * @author Trong Khanh
 */
public class DBUtils {

    public static Connection makeConnection() {
        Connection con = null;
        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
            String url = "jdbc:sqlserver://localhost:1433;databaseName=Sinhvien;instanceName=SQL2005";
            con = DriverManager.getConnection(url, "sa", "trongkhanh");
            return con;
        } catch (SQLException ex) {
            ex.printStackTrace();
        } catch (ClassNotFoundException ex) {
            ex.printStackTrace();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }
```

Sessions & Listeners

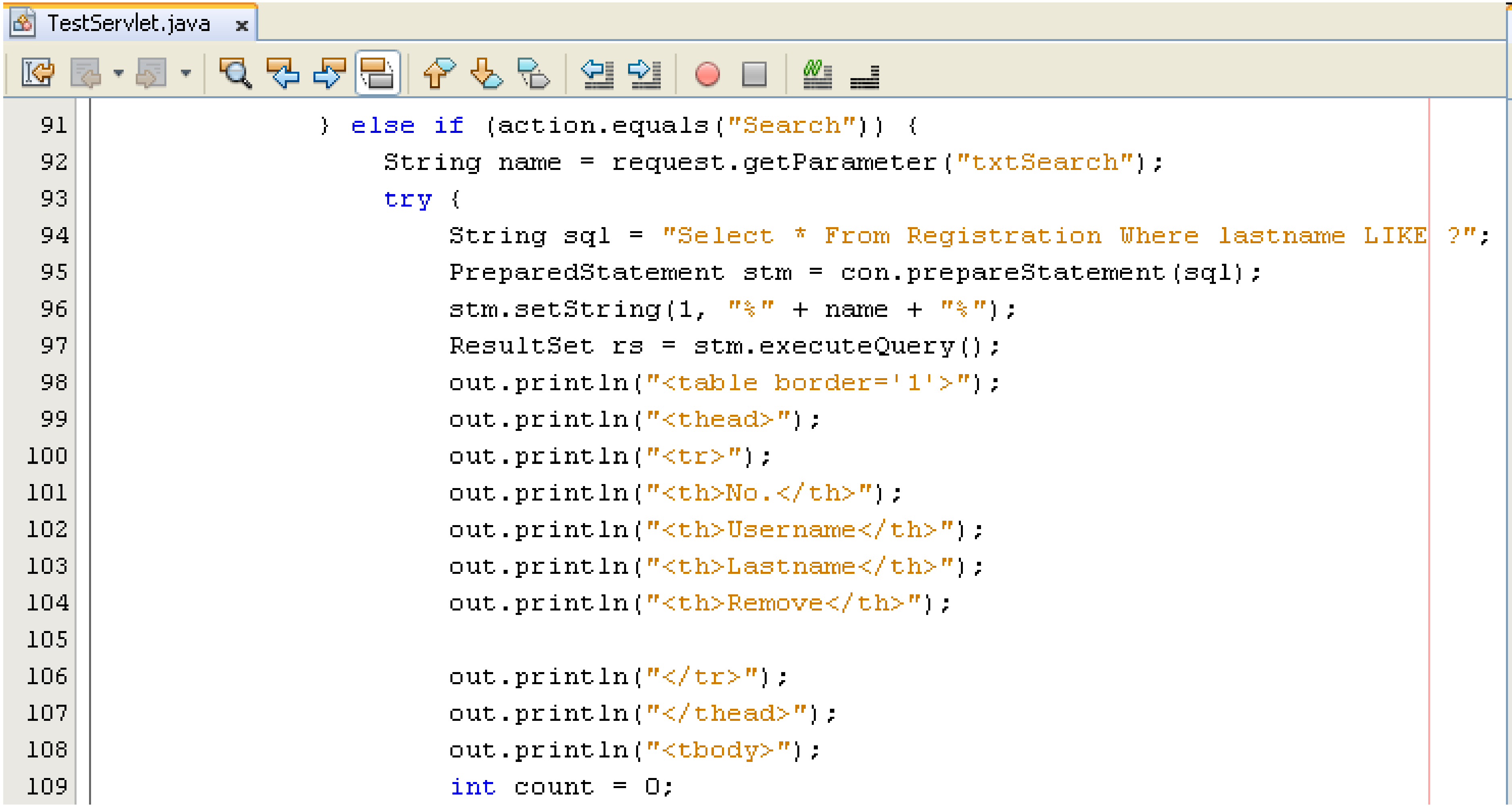
URL Rewriting – Example



```java
public static boolean checkLogin (String username, String password) {
    Connection con = null;
    PreparedStatement stm = null;
    ResultSet rs = null;
    try {
        con = makeConnection();
        String sql = "Select * From Registration Where username = ? and password = ?";
        stm = con.prepareStatement(sql);
        stm.setString(1, username);
        stm.setString(2, password);
        rs = stm.executeQuery();
        if (rs.next()) {
            return true;
        }
    } catch (SQLException ex) {
        ex.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            if (rs!=null) {
                rs.close();
            }
            if (stm!=null) {
                stm.close();
            }
            if (con!=null) {
                con.close();
            }
        }catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return false;
}
```

Sessions & Listeners

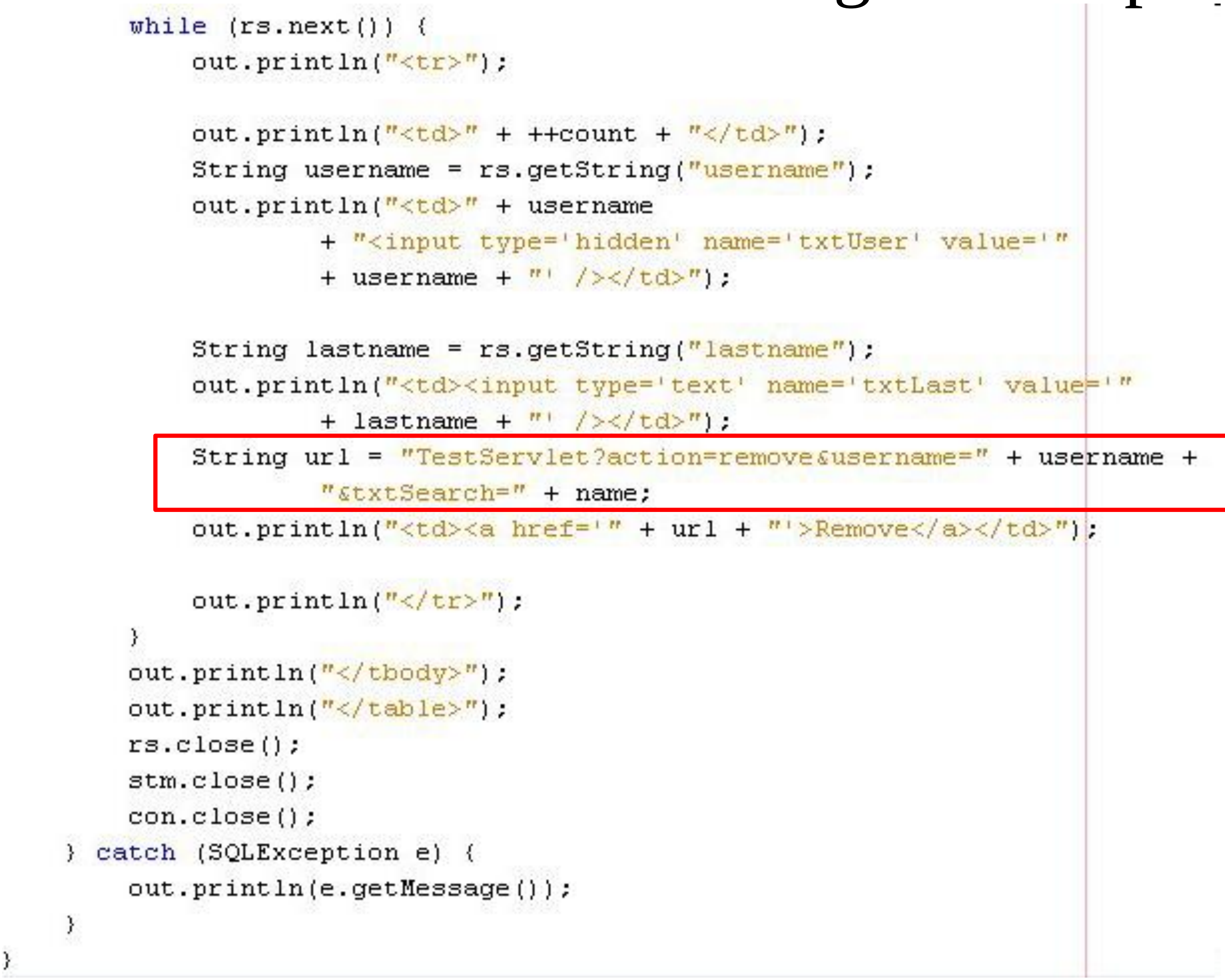
URL Rewriting – Example – First



```
            } else if (action.equals("Search")) {
                String name = request.getParameter("txtSearch");
                try {
                    String sql = "Select * From Registration Where lastname LIKE ?";
                    PreparedStatement stm = con.prepareStatement(sql);
                    stm.setString(1, "%" + name + "%");
                    ResultSet rs = stm.executeQuery();
                    out.println("<table border='1'>");
                    out.println("<thead>");
                    out.println("<tr>");
                    out.println("<th>No.</th>");
                    out.println("<th>Username</th>");
                    out.println("<th>Lastname</th>");
                    out.println("<th>Remove</th>");

                    out.println("</tr>");
                    out.println("</thead>");
                    out.println("<tbody>");
                    int count = 0;
```

Sessions & Listeners

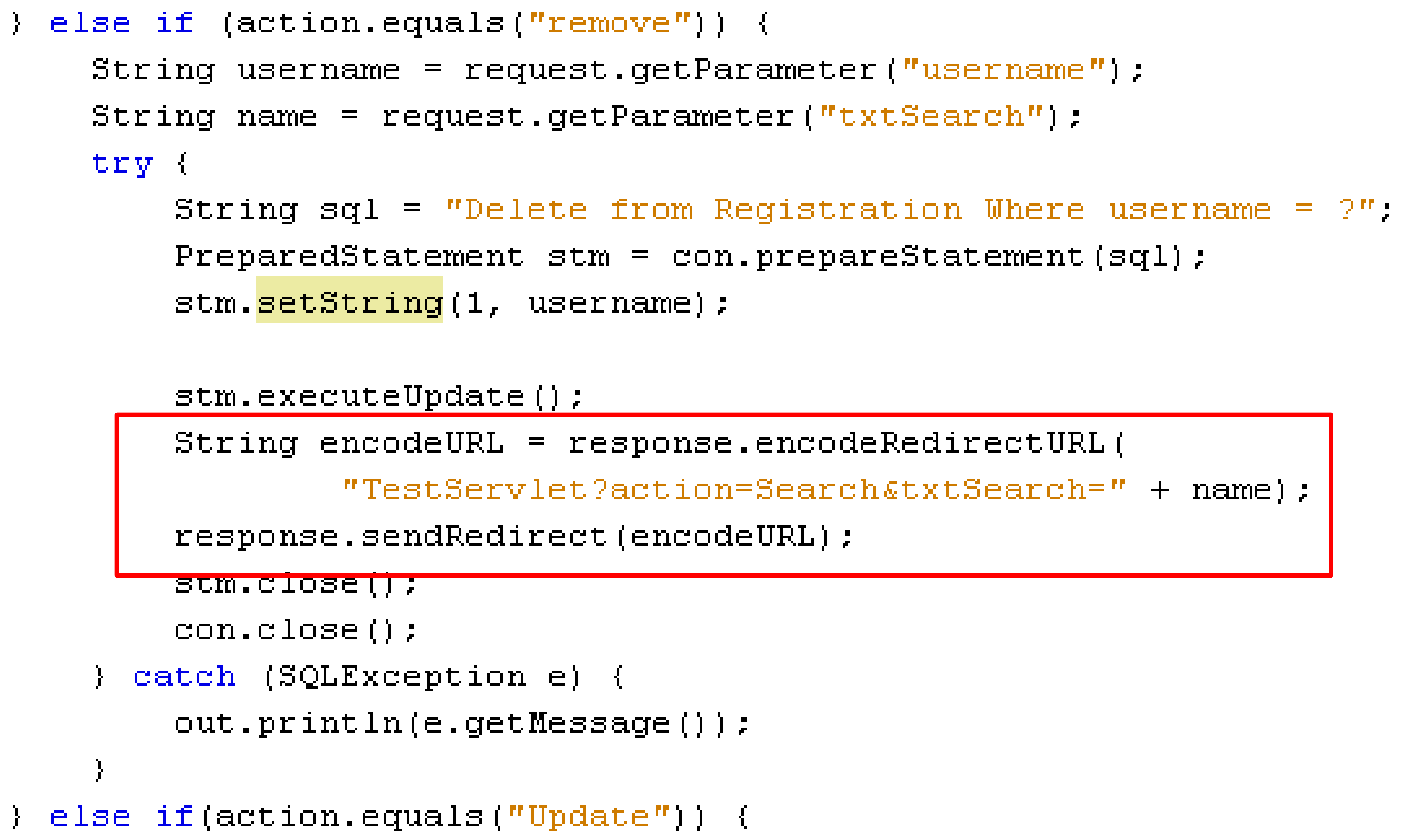
URL Rewriting – Example



```
        while (rs.next()) {
            out.println("<tr>");

            out.println("<td>" + ++count + "</td>");
            String username = rs.getString("username");
            out.println("<td>" + username
                    + "<input type='hidden' name='txtUser' value='"
                    + username + "' /></td>");

            String lastname = rs.getString("lastname");
            out.println("<td><input type='text' name='txtLast' value='"
                    + lastname + "' /></td>");
            String url = "TestServlet?action=remove&username=" + username +
                    "&txtSearch=" + name;
            out.println("<td><a href='" + url + "'>Remove</a></td>");

            out.println("</tr>");
        }
        out.println("</tbody>");
        out.println("</table>");
        rs.close();
        stm.close();
        con.close();
    } catch (SQLException e) {
        out.println(e.getMessage());
    }
}
```

Sessions & Listeners

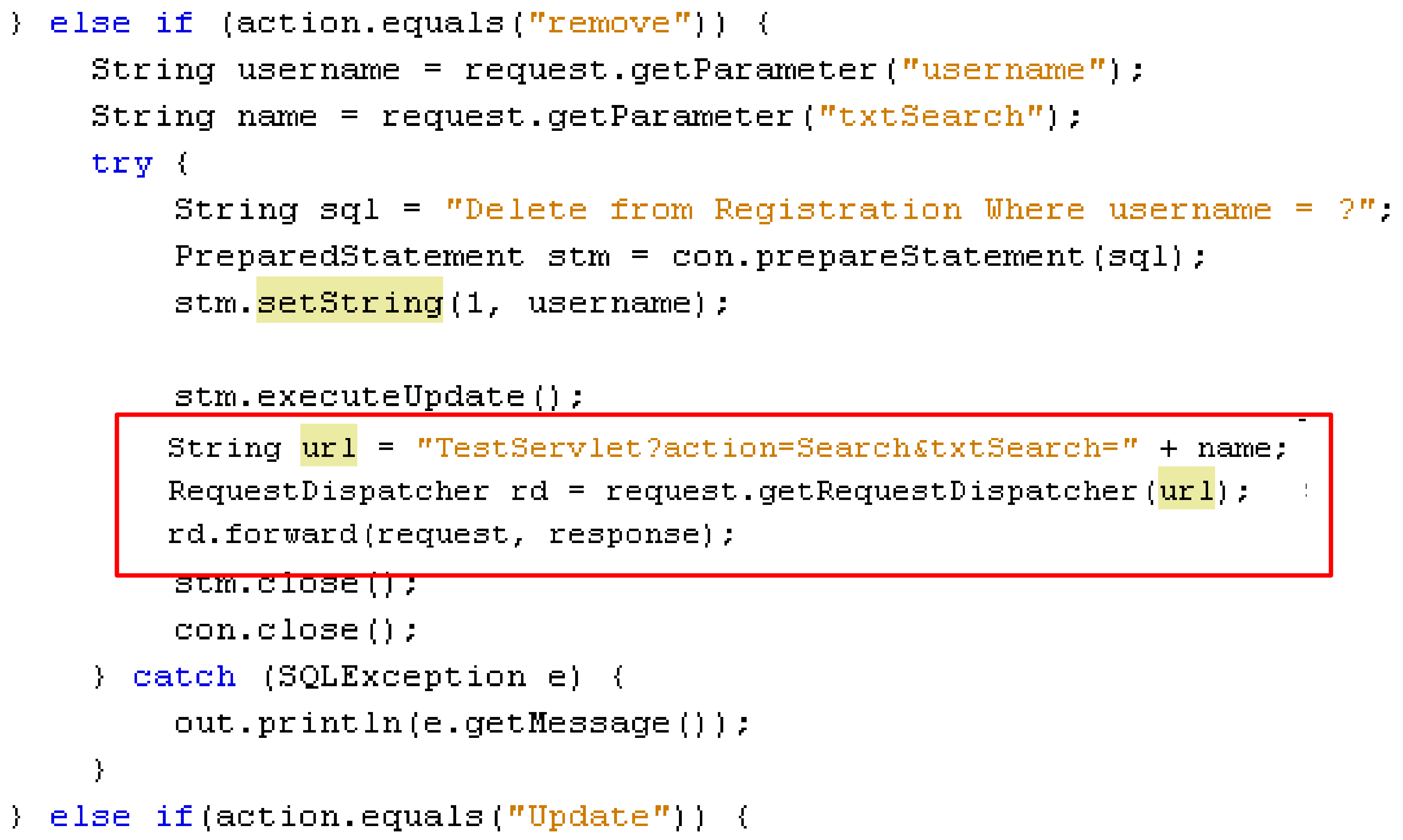
URL Rewriting – Example



```
} else if (action.equals("remove")) {
    String username = request.getParameter("username");
    String name = request.getParameter("txtSearch");
    try {
        String sql = "Delete from Registration Where username = ?";
        PreparedStatement stm = con.prepareStatement(sql);
        stm.setString(1, username);

        stm.executeUpdate();
        String encodeURL = response.encodeRedirectURL(
                "TestServlet?action=Search&txtSearch=" + name);
        response.sendRedirect(encodeURL);
        stm.close();
        con.close();
    } catch (SQLException e) {
        out.println(e.getMessage());
    }
} else if(action.equals("Update")) {
```

Sessions & Listeners

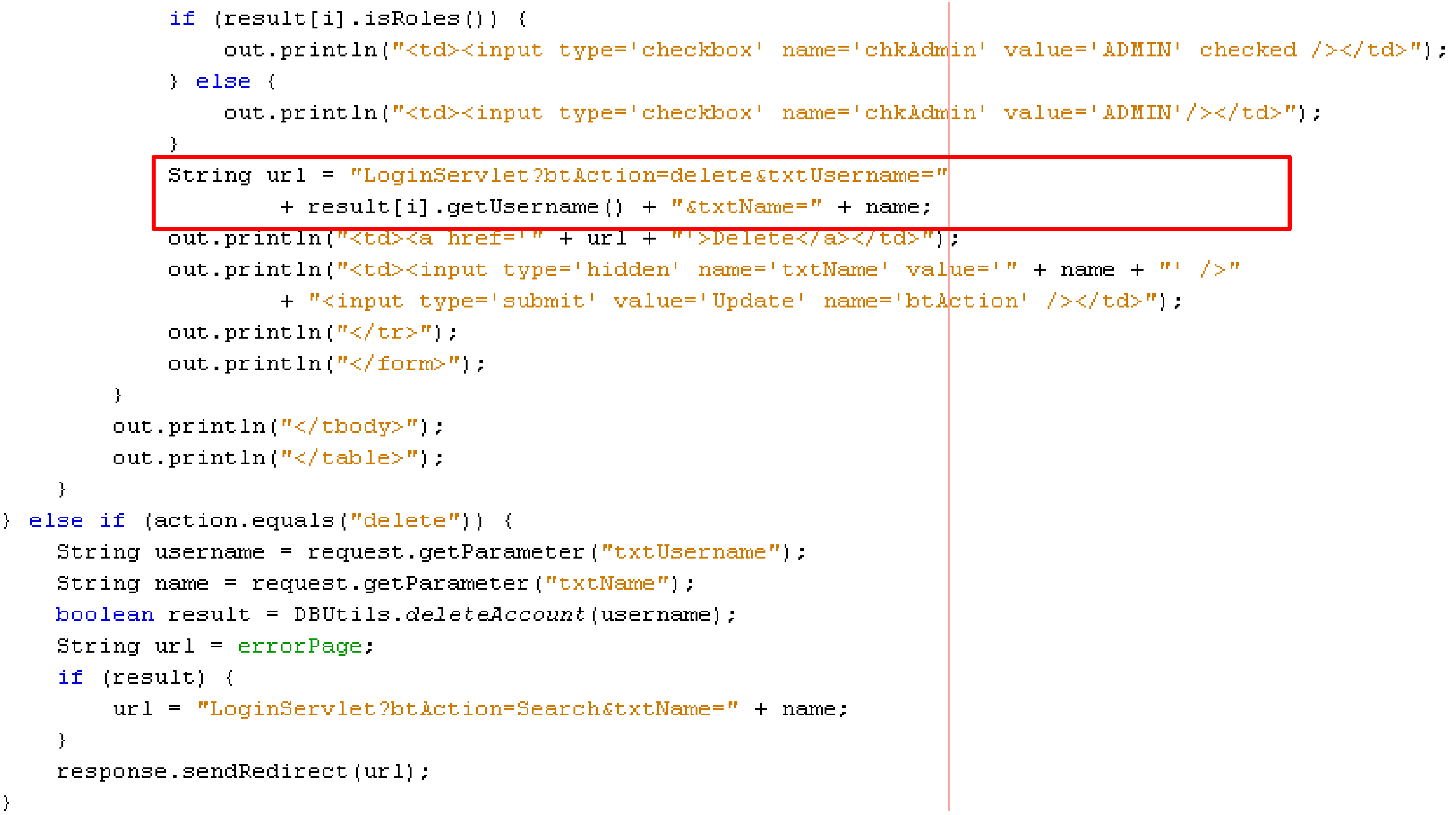
URL Rewriting – Example



```
} else if (action.equals("remove")) {
    String username = request.getParameter("username");
    String name = request.getParameter("txtSearch");
    try {
        String sql = "Delete from Registration Where username = ?";
        PreparedStatement stm = con.prepareStatement(sql);
        stm.setString(1, username);

        stm.executeUpdate();
        String url = "TestServlet?action=Search&txtSearch=" + name;
        RequestDispatcher rd = request.getRequestDispatcher(url);
        rd.forward(request, response);
        stm.close();
        con.close();
    } catch (SQLException e) {
        out.println(e.getMessage());
    }
} else if(action.equals("Update")) {
```

Sessions & Listeners

URL Rewriting – Example – Others



```
            if (result[i].isRoles()) {
                out.println("<td><input type='checkbox' name='chkAdmin' value='ADMIN' checked /></td>");
            } else {
                out.println("<td><input type='checkbox' name='chkAdmin' value='ADMIN'/></td>");
            }
            String url = "LoginServlet?btAction=delete&txtUsername="
                    + result[i].getUsername() + "&txtName=" + name;
            out.println("<td><a href='" + url + "'>Delete</a></td>");
            out.println("<td><input type='hidden' name='txtName' value='" + name + "' />"
                    + "<input type='submit' value='Update' name='btAction' /></td>");
            out.println("</tr>");
            out.println("</form>");
        }
        out.println("</tbody>");
        out.println("</table>");
    }
} else if (action.equals("delete")) {
    String username = request.getParameter("txtUsername");
    String name = request.getParameter("txtName");
    boolean result = DBUtils.deleteAccount(username);
    String url = errorPage;
    if (result) {
        url = "LoginServlet?btAction=Search&txtName=" + name;
    }
    response.sendRedirect(url);
}
```

Hidden Form Fields

- Simplest technique to **maintain the state of an end user**.
- **Insert** the session identifier into the **hidden form field** in the HTML of each page
- **Embedded the hidden form field in an HTML form** and **not visible** when you view an HTML file in a browser window.
- The session information can be **extracted** by the application by **searching** for these fields. The servlets or JSP pages read the field **using request.getParameter().**
- **Syntax**

  <input type=“hidden” name=“…” value=“…”>

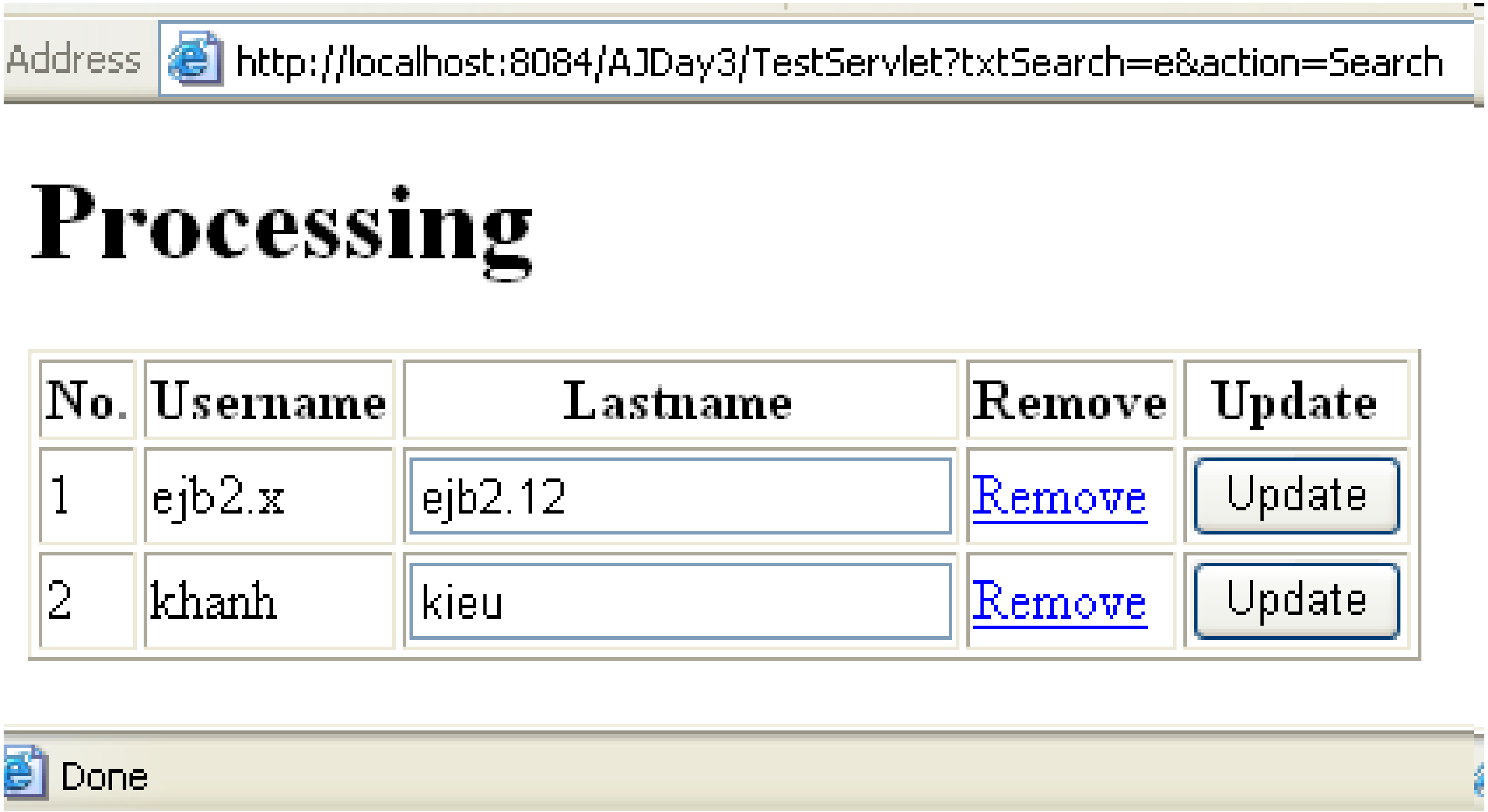
- **Ex**

  <input type=“hidden” name=“productId” value=“P01”>

- **Advantages**
  - **Simplest** way to implement session tracking
  - Displays **nothing** on the HTML page but can be used to hold any kind of data
  - Helps to maintain a **connection between two pages**
- **Disadvantages**:
  - **Work** on the **dynamic pages**.
  - This method of session tracking **displays sensitive information to the user.**

Hidden Form Fields – Examples

- Upgrade the Web Application in URLRewriting topic with update function on the selected row that allows the user update the last name on the data grid as following



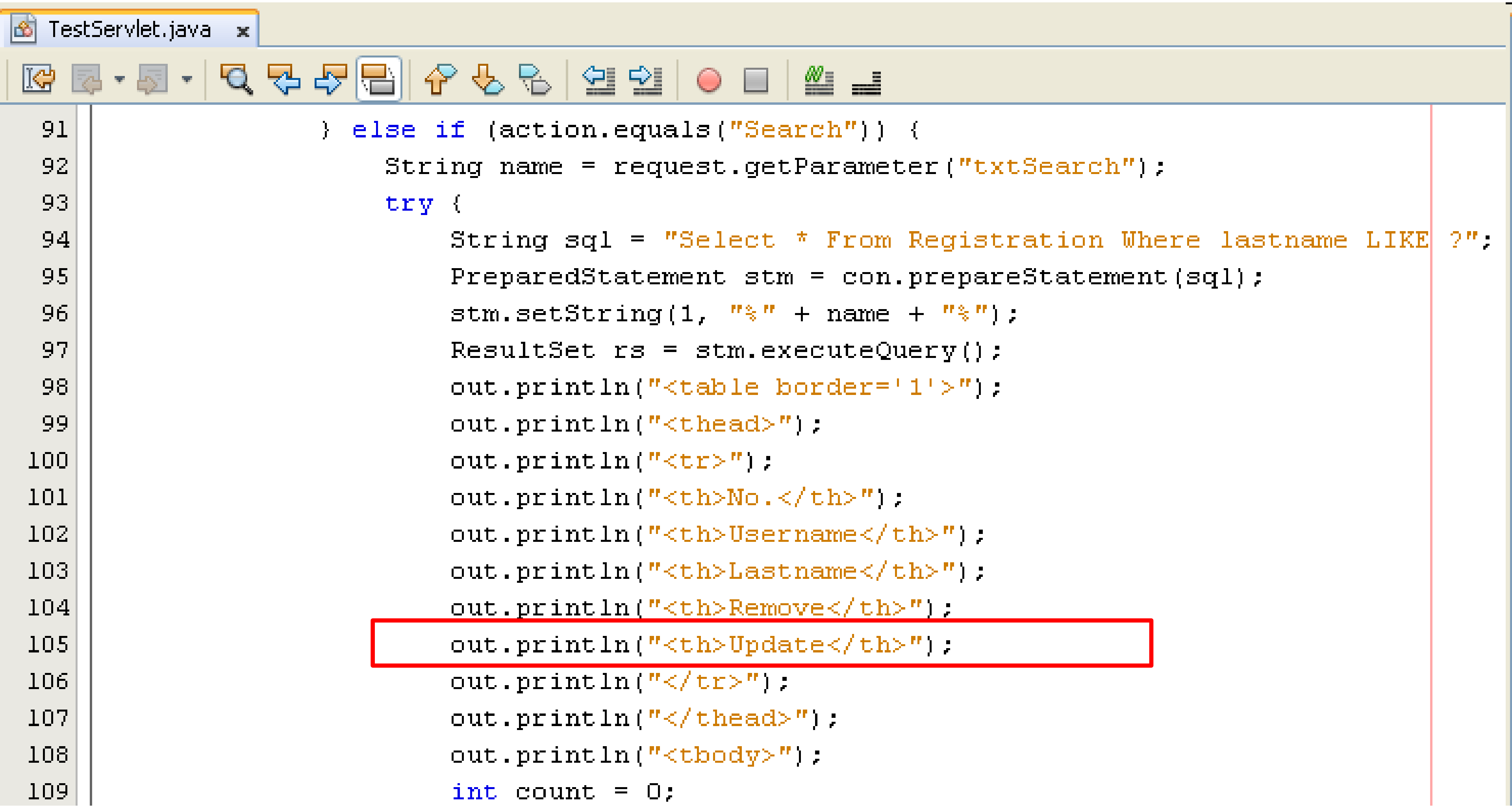
http://localhost:8084/AJDay3/TestServlet?txtSearch=e&action=Search

Processing

No.	Username	Lastname	Remove	Update
1	ejb2.x	ejb2.12	Remove	Update
2	khanh	kieu	Remove	Update

Sessions & Listeners

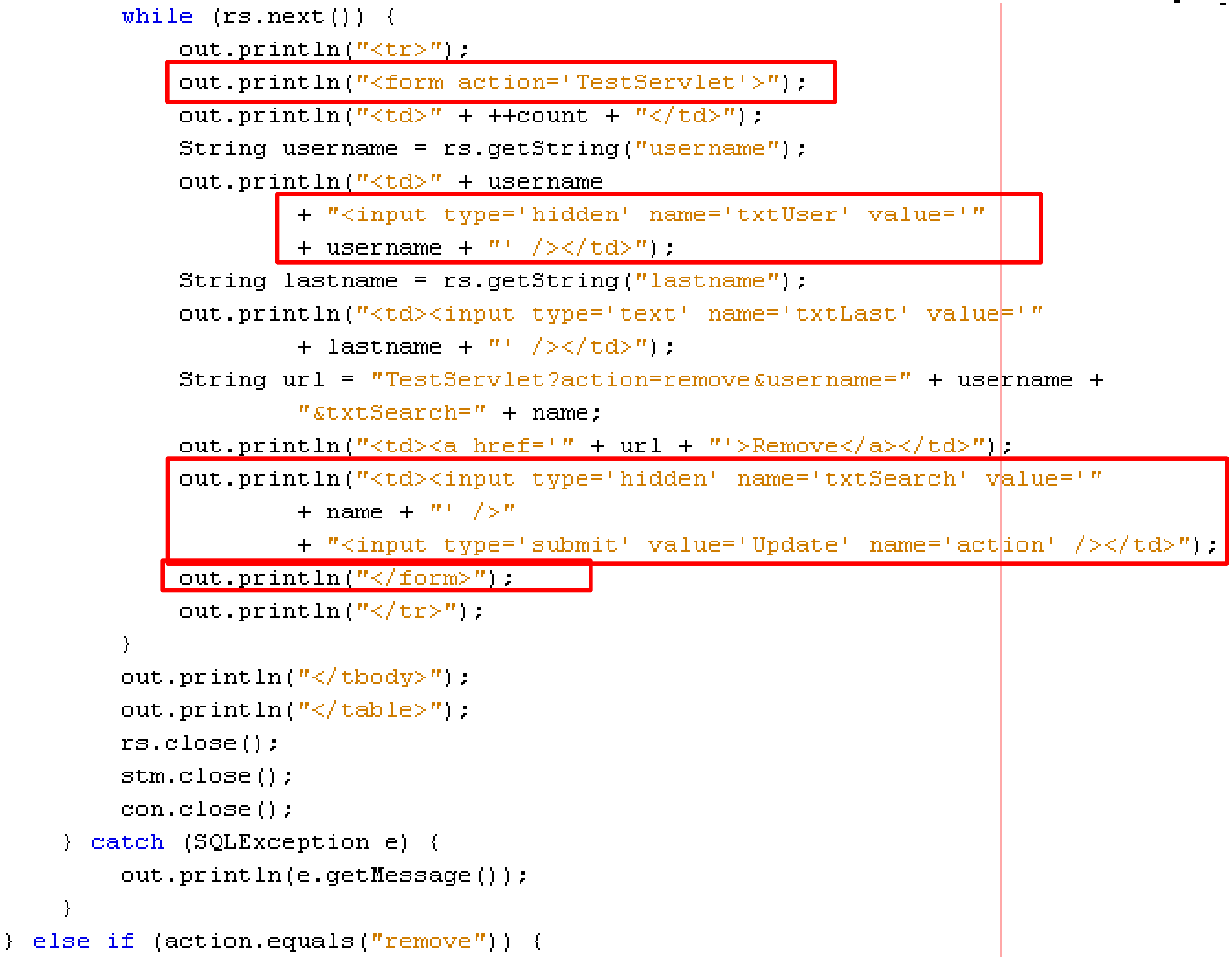
Hidden Form Fields – Examples



```
            } else if (action.equals("Search")) {
                String name = request.getParameter("txtSearch");
                try {
                    String sql = "Select * From Registration Where lastname LIKE ?";
                    PreparedStatement stm = con.prepareStatement(sql);
                    stm.setString(1, "%" + name + "%");
                    ResultSet rs = stm.executeQuery();
                    out.println("<table border='1'>");
                    out.println("<thead>");
                    out.println("<tr>");
                    out.println("<th>No.</th>");
                    out.println("<th>Username</th>");
                    out.println("<th>Lastname</th>");
                    out.println("<th>Remove</th>");
                    out.println("<th>Update</th>");
                    out.println("</tr>");
                    out.println("</thead>");
                    out.println("<tbody>");
                    int count = 0;
```

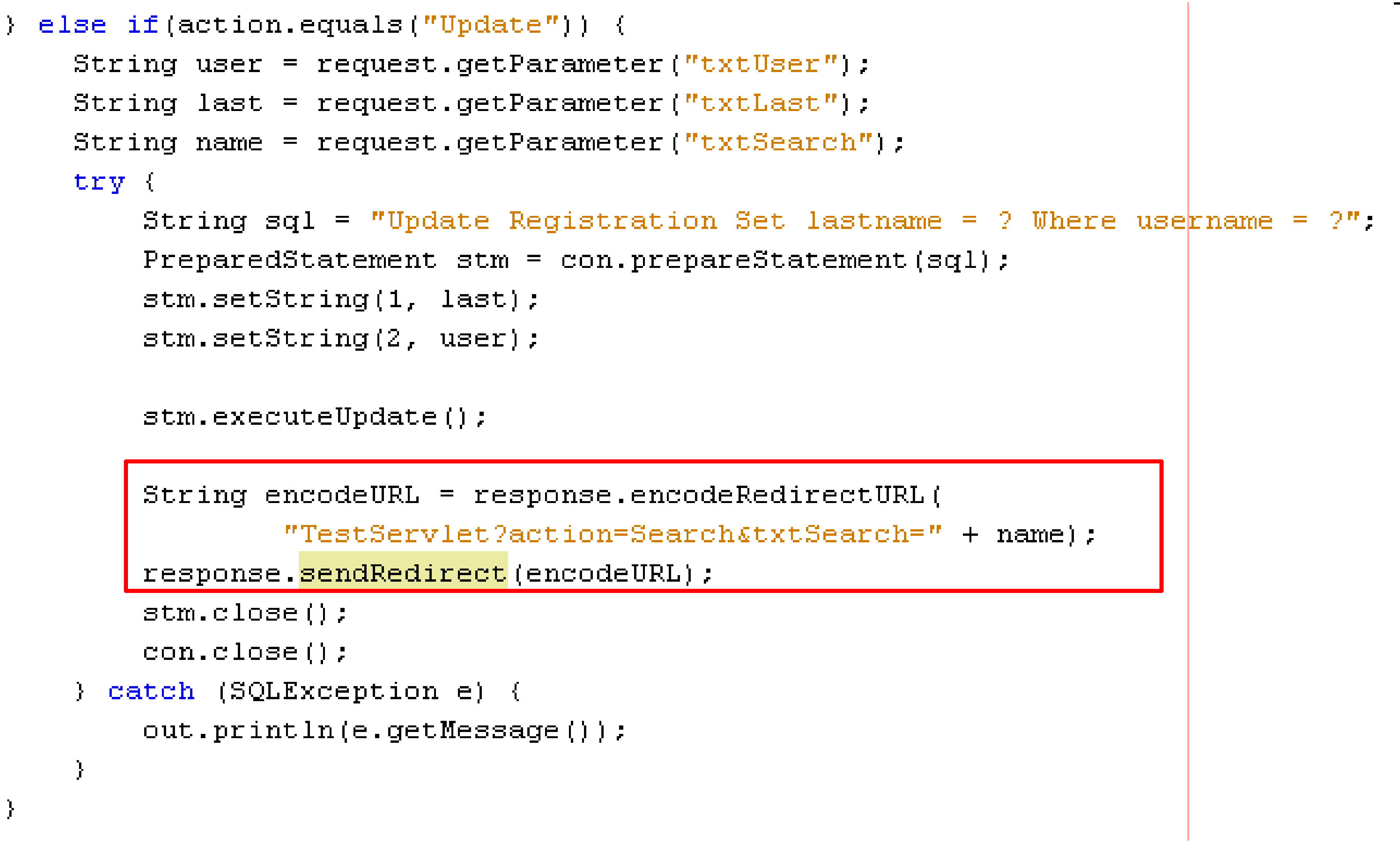
Sessions & Listeners

Hidden Form Fields – Examples



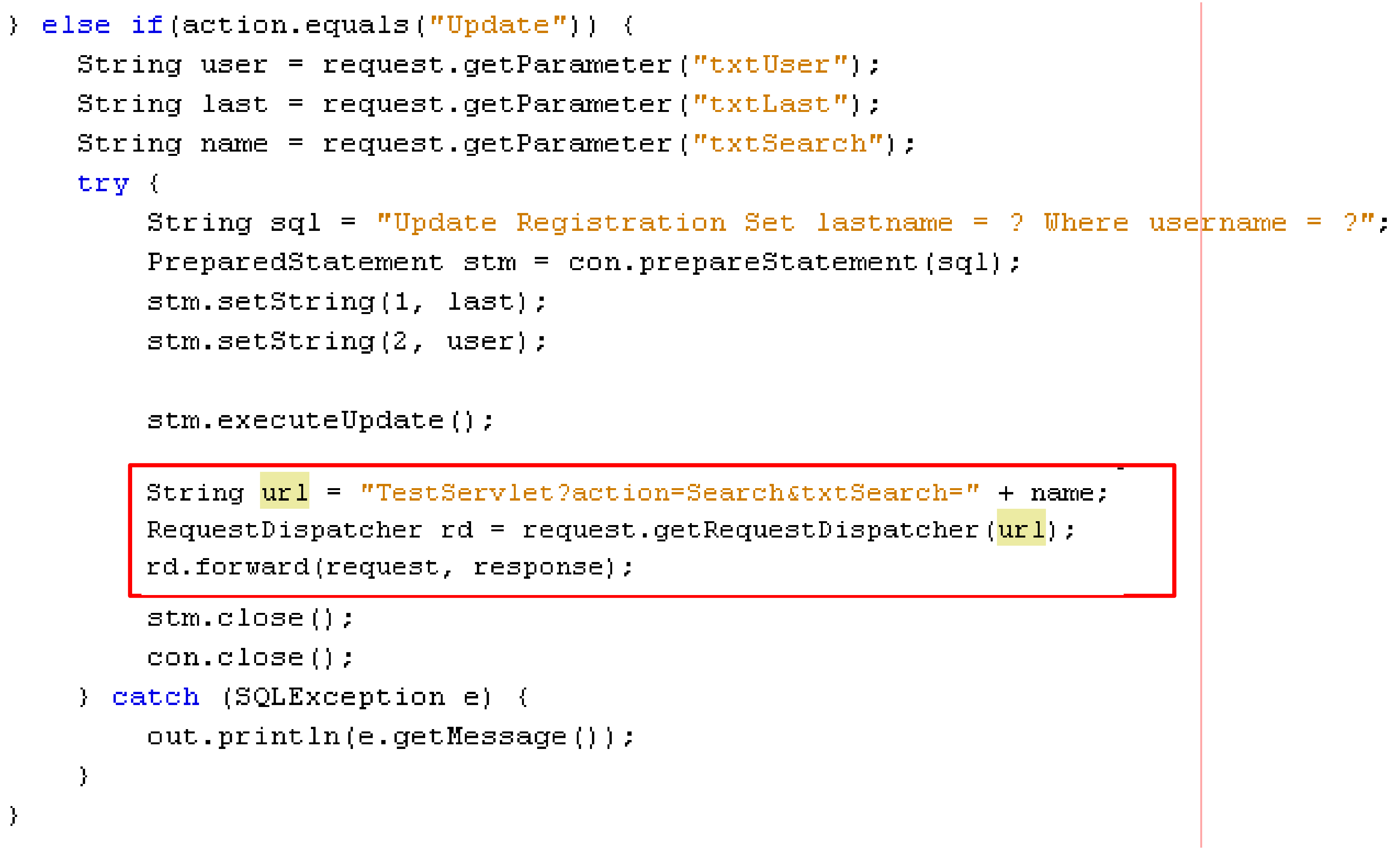
```
        while (rs.next()) {
            out.println("<tr>");
            out.println("<form action='TestServlet'>");
            out.println("<td>" + ++count + "</td>");
            String username = rs.getString("username");
            out.println("<td>" + username
                    + "<input type='hidden' name='txtUser' value='"
                    + username + "' /></td>");
            String lastname = rs.getString("lastname");
            out.println("<td><input type='text' name='txtLast' value='"
                    + lastname + "' /></td>");
            String url = "TestServlet?action=remove&username=" + username +
                    "&txtSearch=" + name;
            out.println("<td><a href='" + url + "'>Remove</a></td>");
            out.println("<td><input type='hidden' name='txtSearch' value='"
                    + name + "' />"
                    + "<input type='submit' value='Update' name='action' /></td>");
            out.println("</form>");
            out.println("</tr>");
        }
        out.println("</tbody>");
        out.println("</table>");
        rs.close();
        stm.close();
        con.close();
    } catch (SQLException e) {
        out.println(e.getMessage());
    }
} else if (action.equals("remove")) {
```

Hidden Form Fields – Examples



```
} else if(action.equals("Update")) {
    String user = request.getParameter("txtUser");
    String last = request.getParameter("txtLast");
    String name = request.getParameter("txtSearch");
    try {
        String sql = "Update Registration Set lastname = ? Where username = ?";
        PreparedStatement stm = con.prepareStatement(sql);
        stm.setString(1, last);
        stm.setString(2, user);

        stm.executeUpdate();

        String encodeURL = response.encodeRedirectURL(
                "TestServlet?action=Search&txtSearch=" + name);
        response.sendRedirect(encodeURL);
        stm.close();
        con.close();
    } catch (SQLException e) {
        out.println(e.getMessage());
    }
}
```

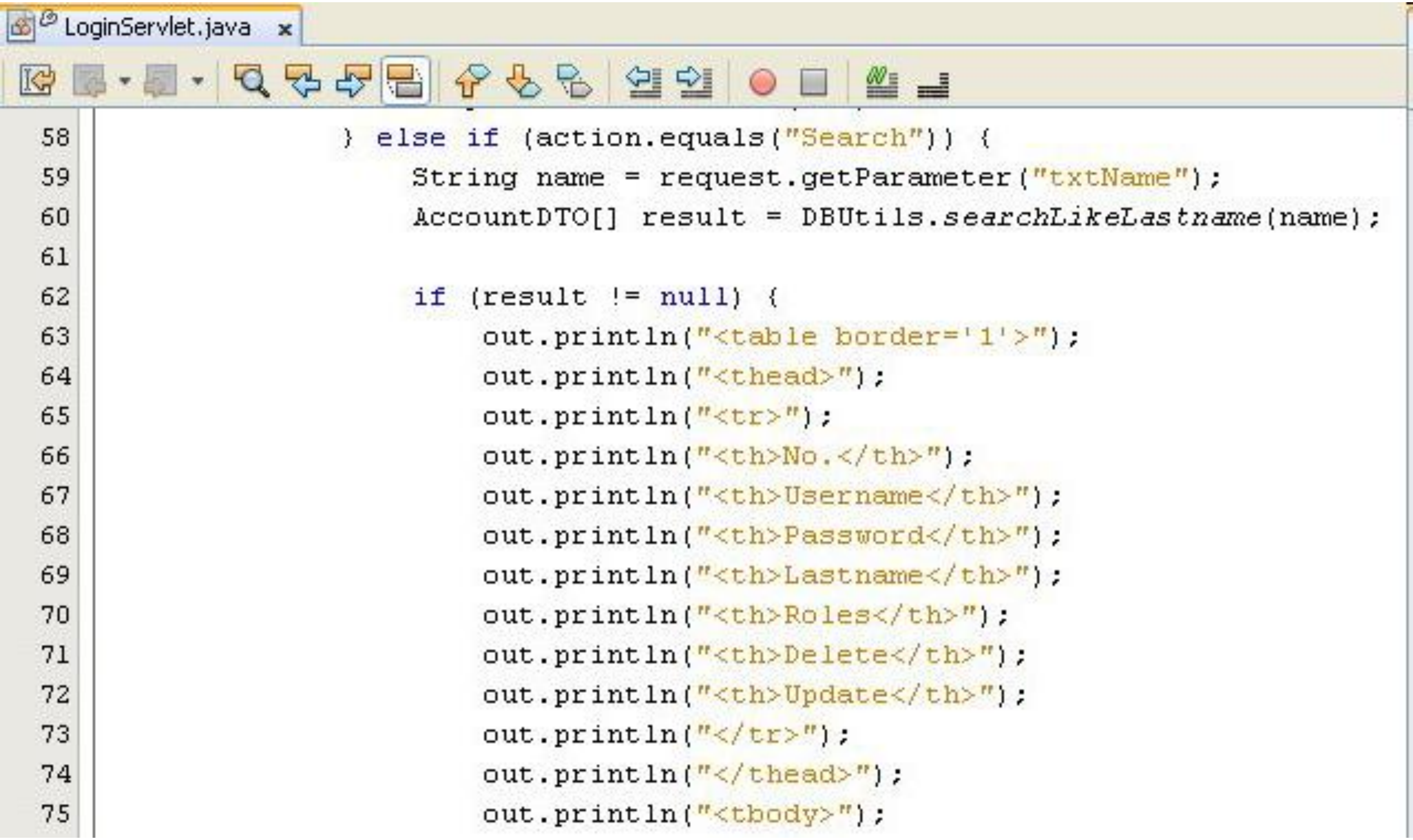
Hidden Form Fields – Examples



```
} else if(action.equals("Update")) {
    String user = request.getParameter("txtUser");
    String last = request.getParameter("txtLast");
    String name = request.getParameter("txtSearch");
    try {
        String sql = "Update Registration Set lastname = ? Where username = ?";
        PreparedStatement stm = con.prepareStatement(sql);
        stm.setString(1, last);
        stm.setString(2, user);

        stm.executeUpdate();

        String url = "TestServlet?action=Search&txtSearch=" + name;
        RequestDispatcher rd = request.getRequestDispatcher(url);
        rd.forward(request, response);
        stm.close();
        con.close();
    } catch (SQLException e) {
        out.println(e.getMessage());
    }
}
```

Sessions & Listeners

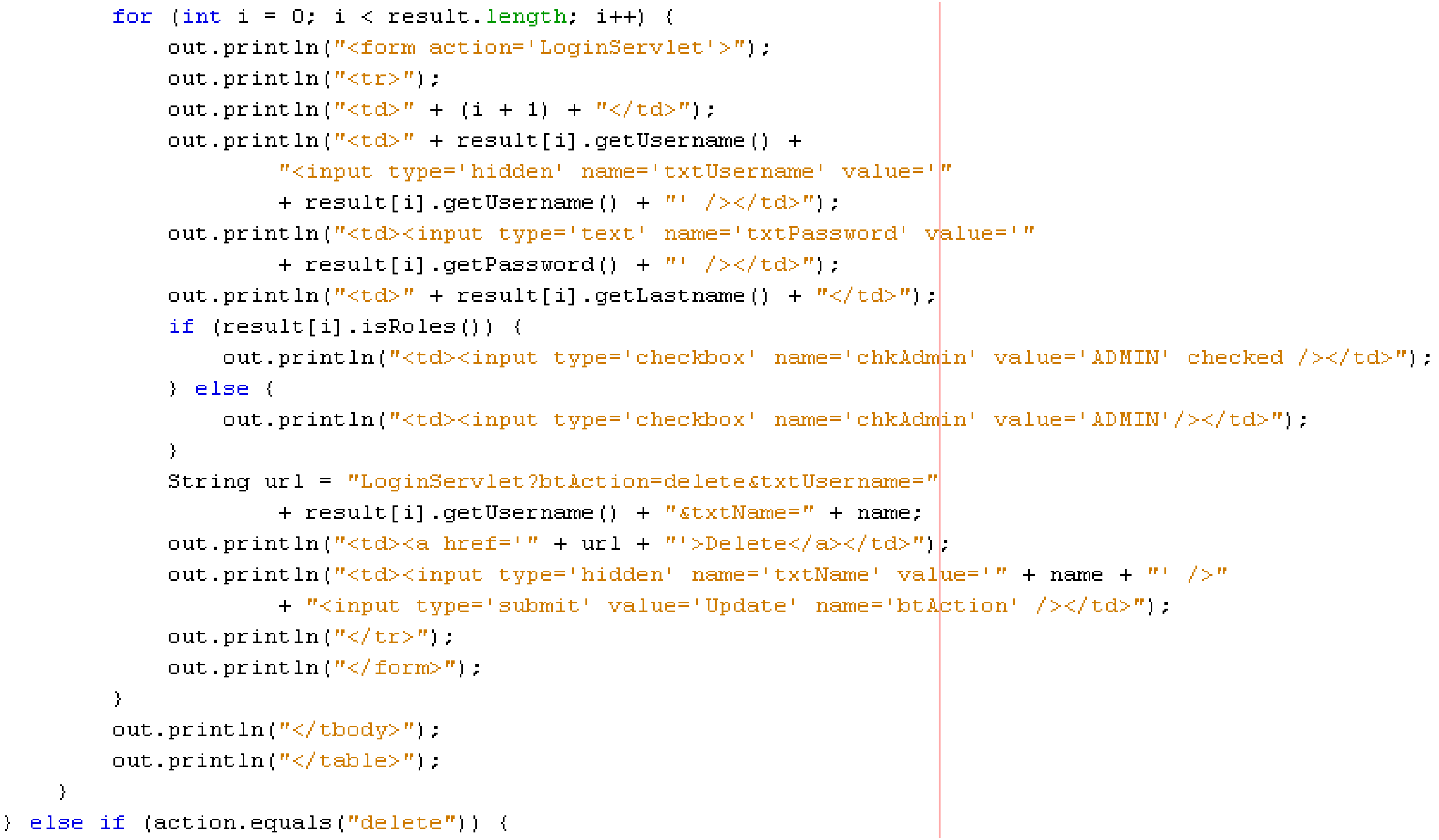
Hidden Form Fields – Examples – Others



```
            } else if (action.equals("Search")) {
                String name = request.getParameter("txtName");
                AccountDTO[] result = DBUtils.searchLikeLastname(name);

                if (result != null) {
                    out.println("<table border='1'>");
                    out.println("<thead>");
                    out.println("<tr>");
                    out.println("<th>No.</th>");
                    out.println("<th>Username</th>");
                    out.println("<th>Password</th>");
                    out.println("<th>Lastname</th>");
                    out.println("<th>Roles</th>");
                    out.println("<th>Delete</th>");
                    out.println("<th>Update</th>");
                    out.println("</tr>");
                    out.println("</thead>");
                    out.println("<tbody>");
```

Sessions & Listeners

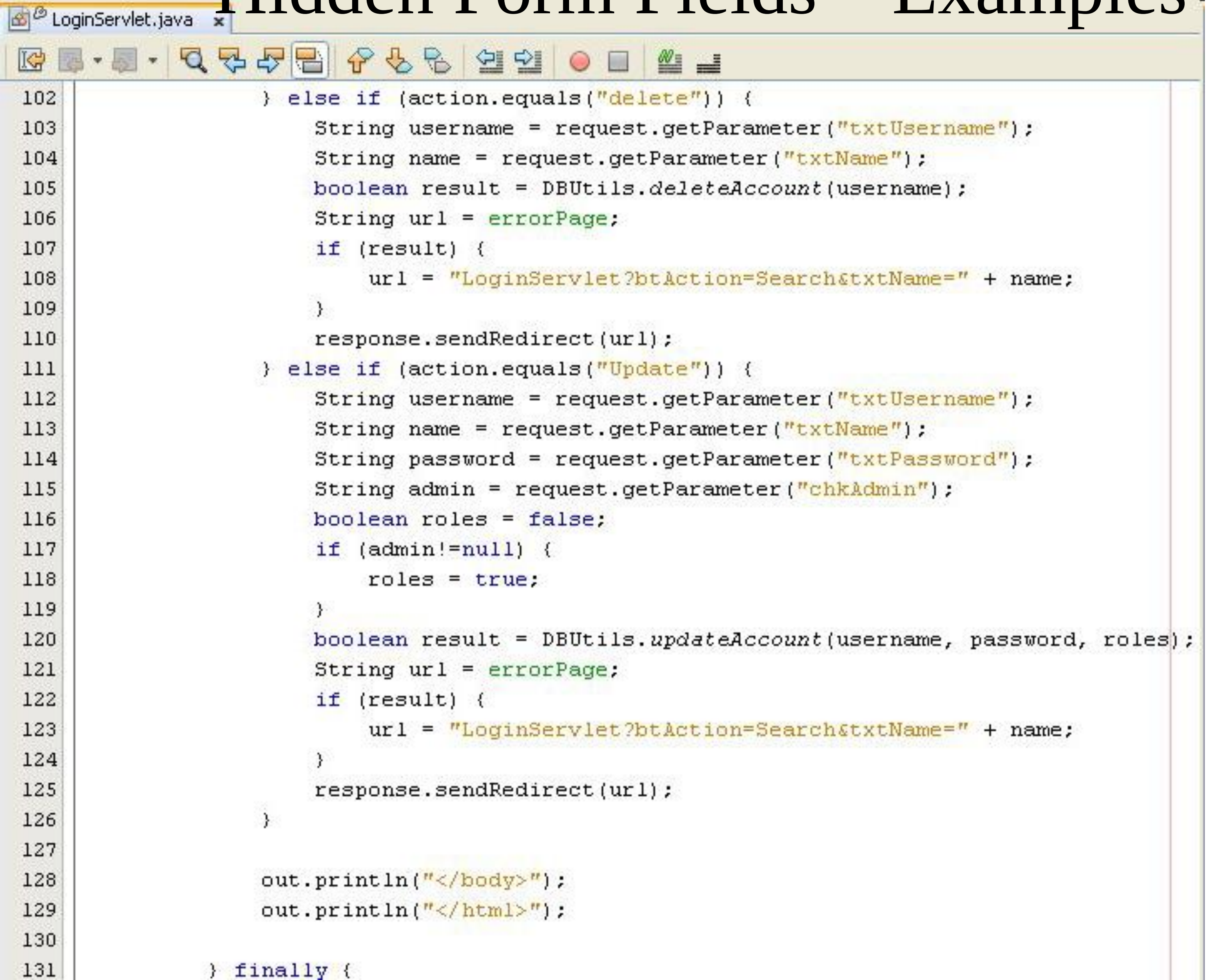
Hidden Form Fields – Examples – Others



```
        for (int i = 0; i < result.length; i++) {
            out.println("<form action='LoginServlet'>");
            out.println("<tr>");
            out.println("<td>" + (i + 1) + "</td>");
            out.println("<td>" + result[i].getUsername() +
                    "<input type='hidden' name='txtUsername' value='"
                    + result[i].getUsername() + "' /></td>");
            out.println("<td><input type='text' name='txtPassword' value='"
                    + result[i].getPassword() + "' /></td>");
            out.println("<td>" + result[i].getLastname() + "</td>");
            if (result[i].isRoles()) {
                out.println("<td><input type='checkbox' name='chkAdmin' value='ADMIN' checked /></td>");
            } else {
                out.println("<td><input type='checkbox' name='chkAdmin' value='ADMIN'/></td>");
            }
            String url = "LoginServlet?btAction=delete&txtUsername="
                    + result[i].getUsername() + "&txtName=" + name;
            out.println("<td><a href='" + url + "'>Delete</a></td>");
            out.println("<td><input type='hidden' name='txtName' value='" + name + "' />"
                    + "<input type='submit' value='Update' name='btAction' /></td>");
            out.println("</tr>");
            out.println("</form>");
        }
        out.println("</tbody>");
        out.println("</table>");
    }
} else if (action.equals("delete")) {
```

Sessions & Listeners

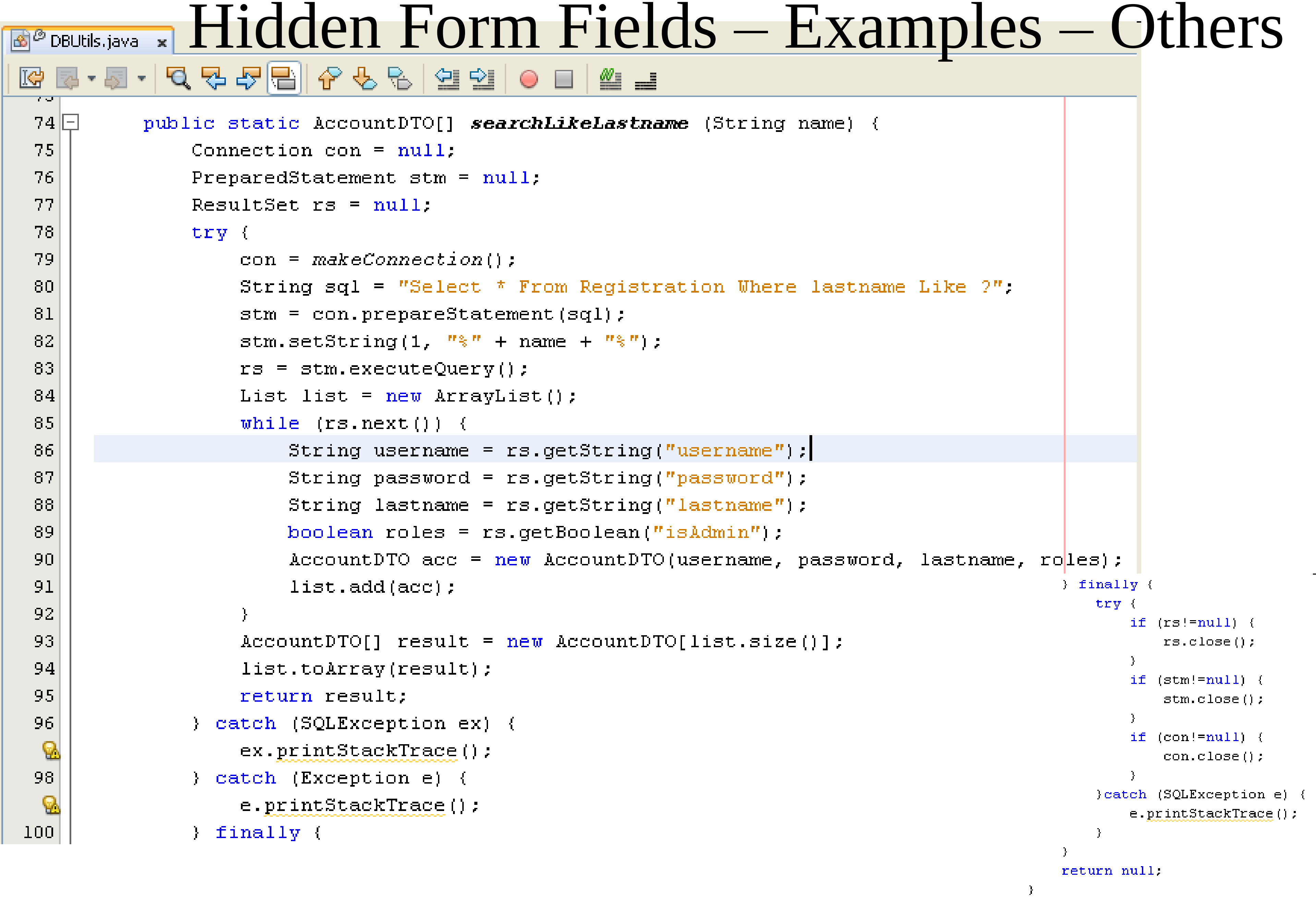
Hidden Form Fields – Examples – Others



```
            } else if (action.equals("delete")) {
                String username = request.getParameter("txtUsername");
                String name = request.getParameter("txtName");
                boolean result = DBUtils.deleteAccount(username);
                String url = errorPage;
                if (result) {
                    url = "LoginServlet?btAction=Search&txtName=" + name;
                }
                response.sendRedirect(url);
            } else if (action.equals("Update")) {
                String username = request.getParameter("txtUsername");
                String name = request.getParameter("txtName");
                String password = request.getParameter("txtPassword");
                String admin = request.getParameter("chkAdmin");
                boolean roles = false;
                if (admin!=null) {
                    roles = true;
                }
                boolean result = DBUtils.updateAccount(username, password, roles);
                String url = errorPage;
                if (result) {
                    url = "LoginServlet?btAction=Search&txtName=" + name;
                }
                response.sendRedirect(url);
            }

            out.println("</body>");
            out.println("</html>");

        } finally {
```

Sessions & Listeners

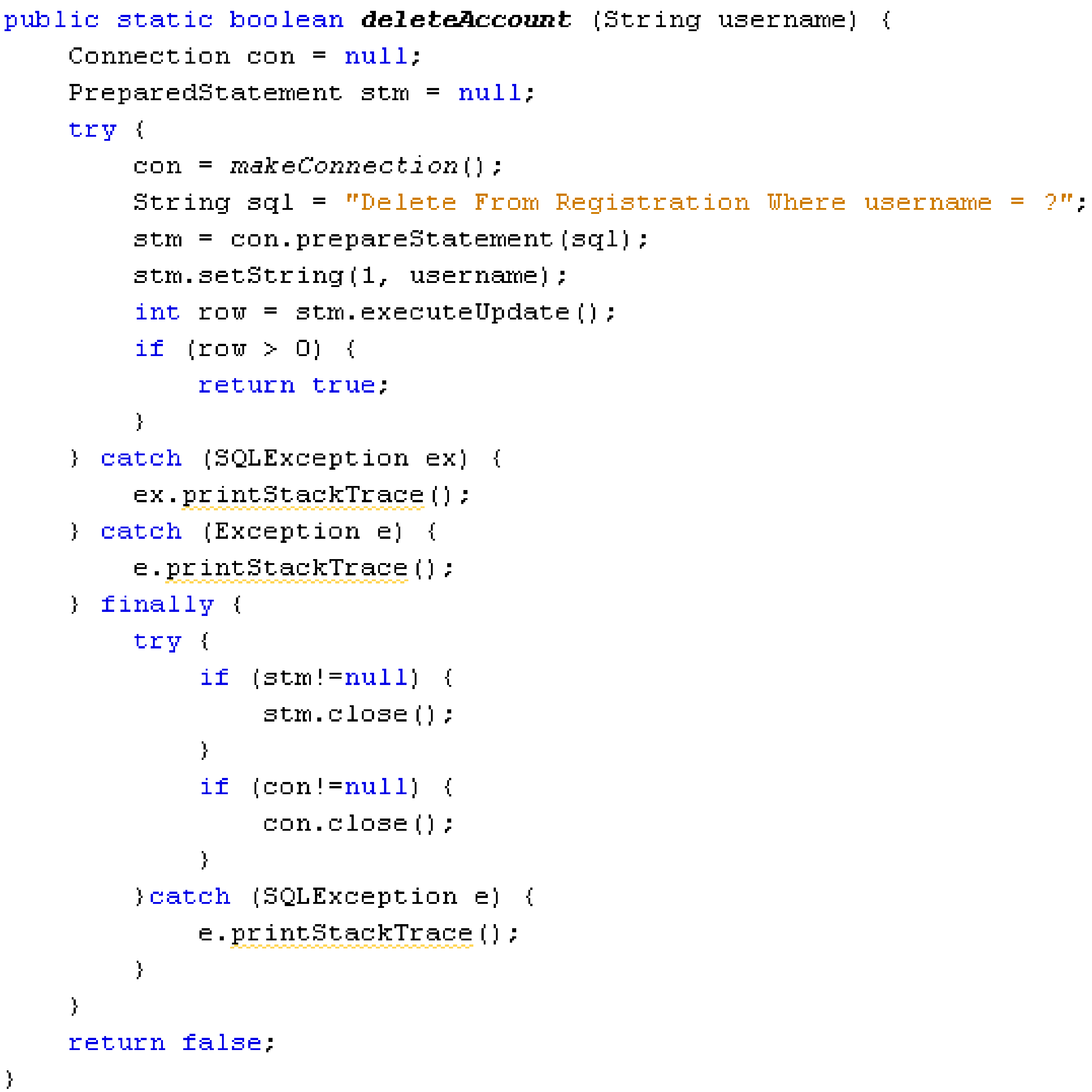
Hidden Form Fields – Examples – Others



```
    public static AccountDTO[] searchLikeLastname (String name) {
        Connection con = null;
        PreparedStatement stm = null;
        ResultSet rs = null;
        try {
            con = makeConnection();
            String sql = "Select * From Registration Where lastname Like ?";
            stm = con.prepareStatement(sql);
            stm.setString(1, "%" + name + "%");
            rs = stm.executeQuery();
            List list = new ArrayList();
            while (rs.next()) {
                String username = rs.getString("username");
                String password = rs.getString("password");
                String lastname = rs.getString("lastname");
                boolean roles = rs.getBoolean("isAdmin");
                AccountDTO acc = new AccountDTO(username, password, lastname, roles);
                list.add(acc);
            }
            AccountDTO[] result = new AccountDTO[list.size()];
            list.toArray(result);
            return result;
        } catch (SQLException ex) {
            ex.printStackTrace();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
```

```
        } finally {
            try {
                if (rs!=null) {
                    rs.close();
                }
                if (stm!=null) {
                    stm.close();
                }
                if (con!=null) {
                    con.close();
                }
            }catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return null;
    }
```

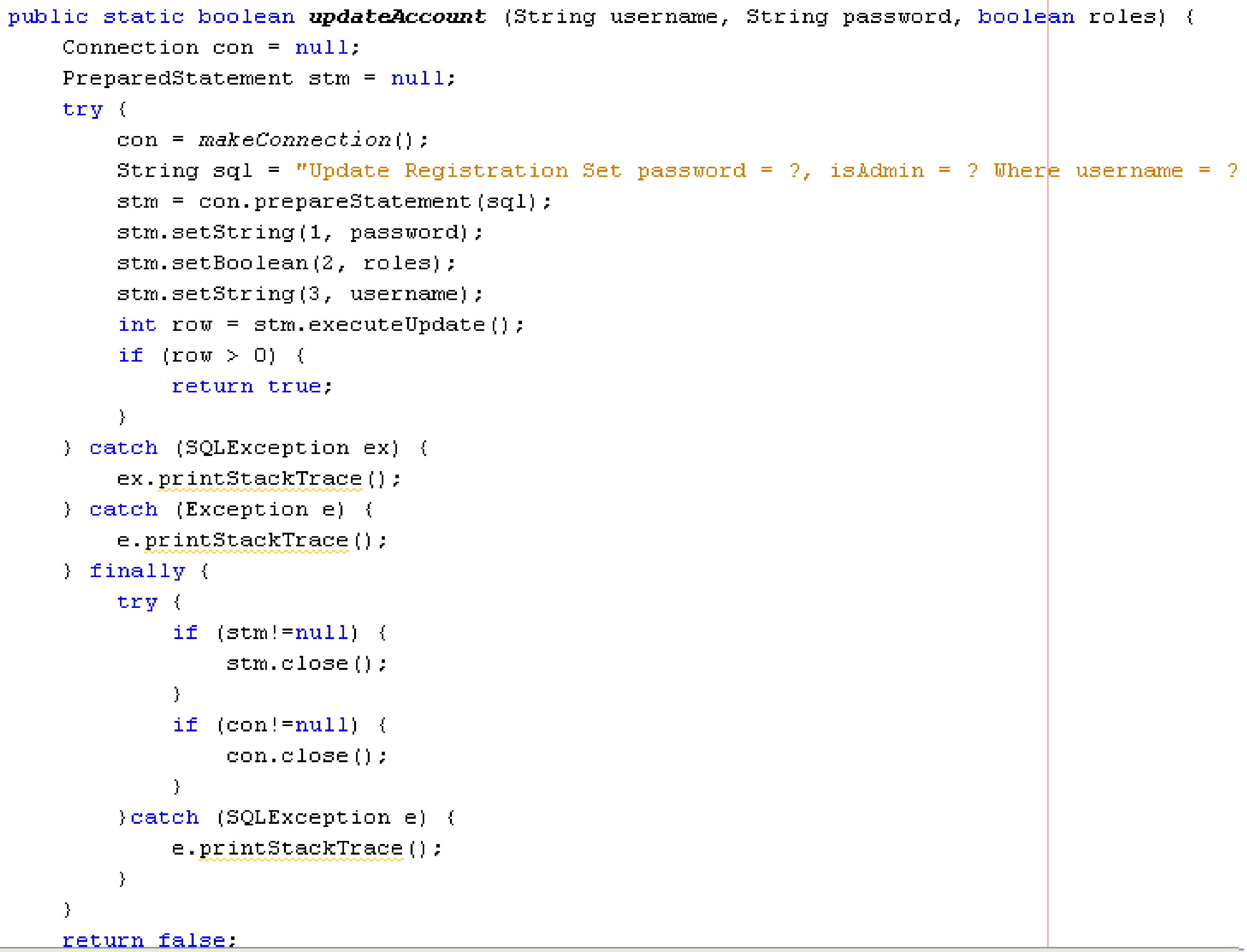
Hidden Form Fields – Examples – Others



```
public static boolean deleteAccount (String username) {
    Connection con = null;
    PreparedStatement stm = null;
    try {
        con = makeConnection();
        String sql = "Delete From Registration Where username = ?";
        stm = con.prepareStatement(sql);
        stm.setString(1, username);
        int row = stm.executeUpdate();
        if (row > 0) {
            return true;
        }
    } catch (SQLException ex) {
        ex.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            if (stm!=null) {
                stm.close();
            }
            if (con!=null) {
                con.close();
            }
        }catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return false;
}
```

Sessions & Listeners

Hidden Form Fields – Examples – Others

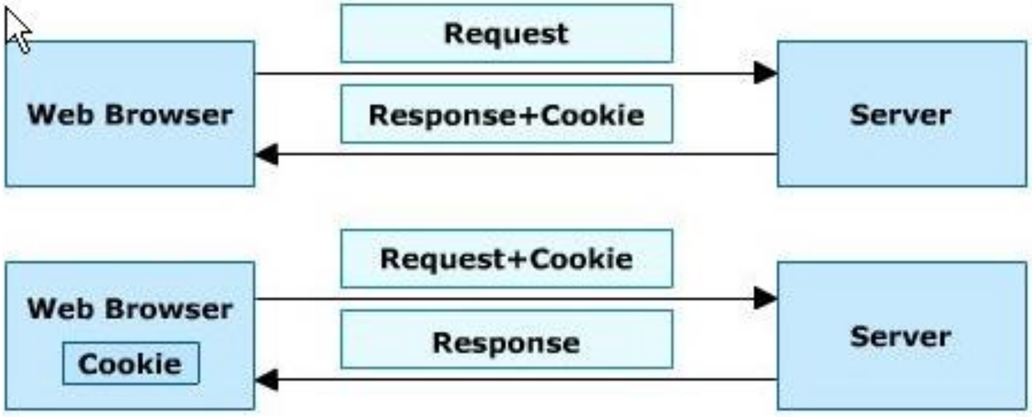


```
public static boolean updateAccount (String username, String password, boolean roles) {
    Connection con = null;
    PreparedStatement stm = null;
    try {
        con = makeConnection();
        String sql = "Update Registration Set password = ?, isAdmin = ? Where username = ?
        stm = con.prepareStatement(sql);
        stm.setString(1, password);
        stm.setBoolean(2, roles);
        stm.setString(3, username);
        int row = stm.executeUpdate();
        if (row > 0) {
            return true;
        }
    } catch (SQLException ex) {
        ex.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            if (stm!=null) {
                stm.close();
            }
            if (con!=null) {
                con.close();
            }
        }catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return false;
```

Sessions & Listeners

Cookies

- Is a **small piece** of information **sent by** the **web server to** the **client** to keep track of users.
- Size of each cookie can be a maximum of 4 KB.
- Cookie has values in the form of **key-value pairs**
- When the **server sends** a cookie, the **client receives** the cookie, **saves and sends** it **back** to the **server** each **time** the **client accesses** a page on that server
- Can uniquely identify a client (In the case of J2EE web applications, the cookie returned has a standard name **JSESSIONID** and store in memory)
- A web browser is expected to **support 20 Cookies** per host



Concept of Cookie

Sessions & Listeners

Cookies

- **Advantages**
  - **Remember** user IDs and password.(low security)
  - To **track** visitors on a Web site for better service and new features.
  - Cookies enable **efficient ad processing**.
  - Support **e-advertisement** on Internet.
  - Security (can not affect virus).
- **Disadvantages**
  - **Personal information** is **exposed** to the **other users**. (spam/ junk mail, pop up …)
  - Cookies fails to work if the security level is set too high in the Internet browser.
  - Most browsers **enable** the **user at** the **client machine to deactivate** (not to accept) cookies.
  - The size and number of cookies **stored are limited.**
- **Note**
  - **Browser is accepted cookies**
  - **Cookies are stored** at
    - **C:\Documents and Settings\LoggedUserName\Cookies\LoggedUserName@ContextPath[n].txt**
    - **C:\Users\LoggedUserName\AppData\Local\Microsoft\Windows\Temporary Internet Files \LoggedUserName@host[n].txt**
  - Cookies are existed following the **setMaxAge** and deleted automatically by OS

Cookies

- The servlet API provides **javax.servlet.http.Cookie** class **for creating and working** with cookies
- The **constructor** for the cookies class is: Cookie(java.lang.String name, java.lang.String value)
- **Sending Cookie**

Methods	Descriptions
addCookie	- **public void addCookie(cookie1);** - Adds field to the HTTP response headers to send cookies to the browser, one at a time - Adds specified cookie to the response - Can be called multiple times to set more than one cookies
setValue	- **public void setValue(String newValue);** - Assigns a new value to a cookie after the cookie is created. In case if binary value is used, base 64 can be used for encoding
setPath	- **public void setPath(String path);** - Sets the path for the cookie. The cookie is available to all the pages specified in the directory and its subdirectories. A cookie's path must have the servlet which sets the cookie
setMaxAge	- **public void setMaxAge(int expiry);** - The maximum age of the cookie in seconds. If the value is positive, then the cookie will expire after that many seconds which is specified by the expiry

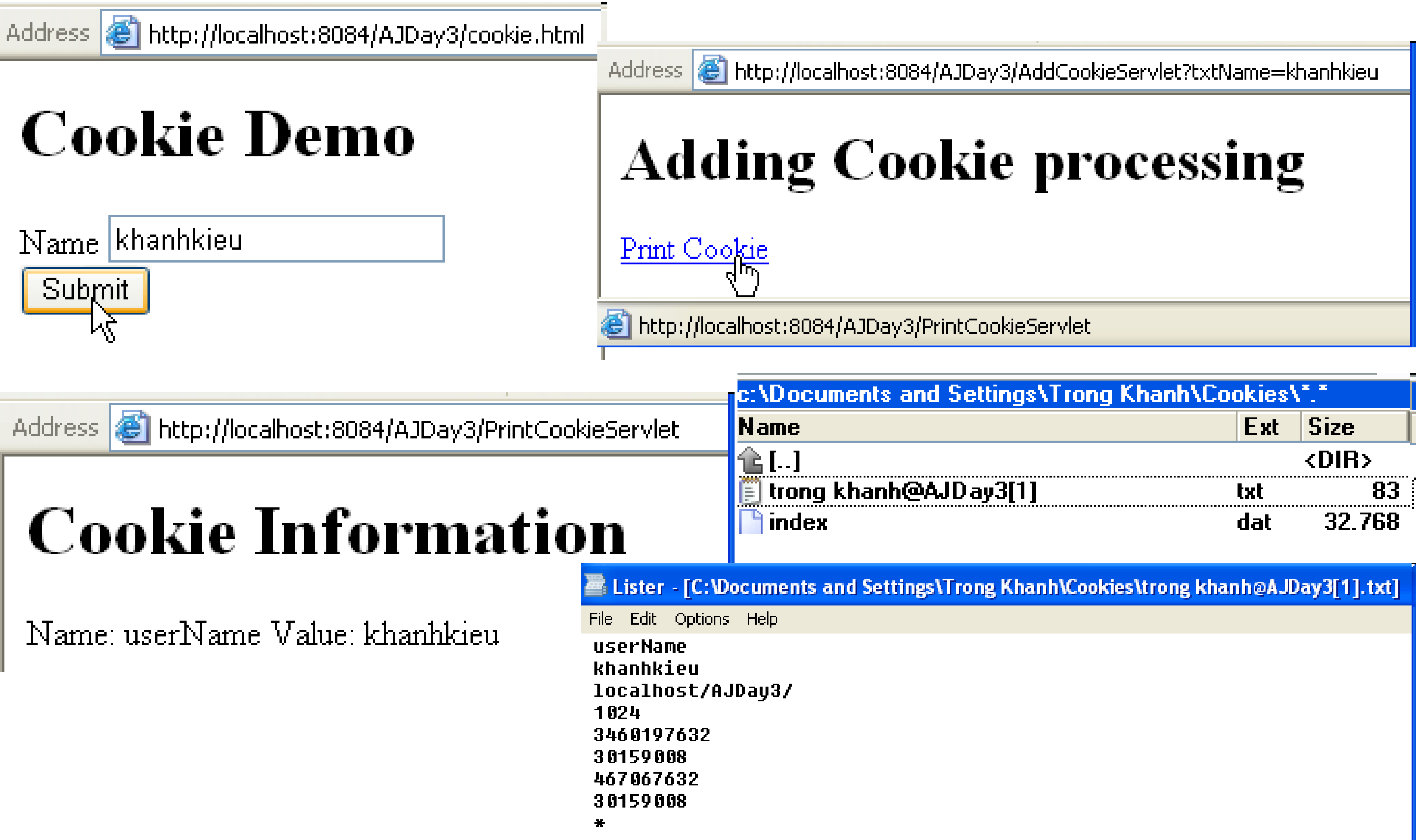
Sessions & Listeners
Cookies

- **Reading Cookie**

Methods	Descriptions
getCookies	- **Cookie [] cookies = request.getCookies();** - Returns an array containing all of the Cookie objects the client sends with the request
getMaxAge	- **public int getMaxAge();** - Returns the maximum age of the cookie. - Returns an integer which specify the maximum age of the cookies in seconds
getValue	- **public String getValue();** - Returns the value of the cookie
getName	- **public String getName()** - Returns the name of cookie. Once the cookie has been created its name cannot be changed
getPath	- **public void getPath()** - Returns the path on the server to which the client return the cookie. The cookie is available to all sub paths on the server

Sessions & Listeners

Cookies – Example



Address http://localhost:8084/AJDay3/cookie.html

Cookie Demo

Name khanhkieu

Submit

Address http://localhost:8084/AJDay3/AddCookieServlet?txtName=khanhkieu

Adding Cookie processing

Print Cookie

http://localhost:8084/AJDay3/PrintCookieServlet

Address http://localhost:8084/AJDay3/PrintCookieServlet

Cookie Information

Name: userName Value: khanhkieu

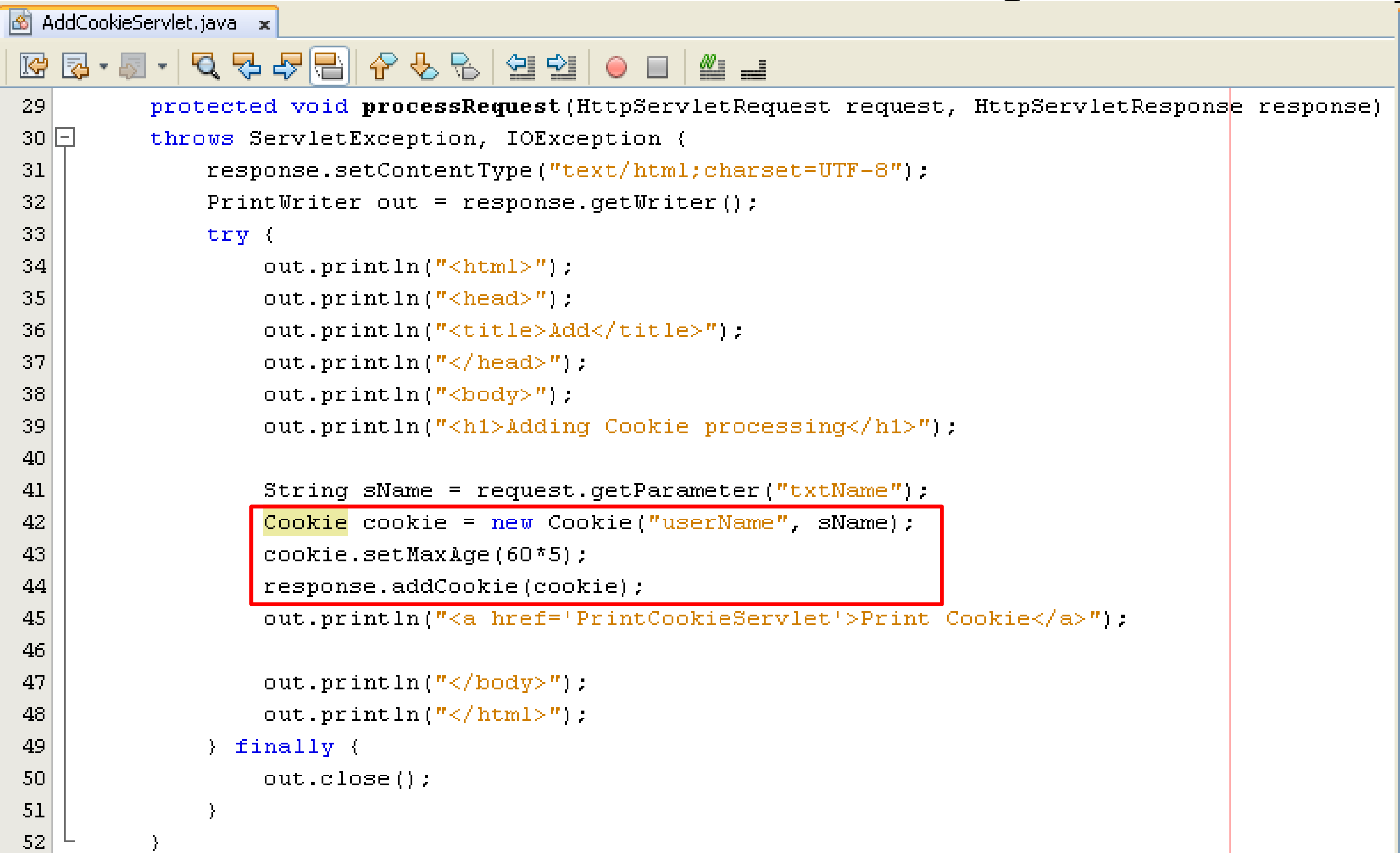
c:\Documents and Settings\Trong Khanh\Cookies\*.*

Name	Ext	Size
[..]		<DIR>
trong khanh@AJDay3[1]	txt	83
index	dat	32.768

Lister - [C:\Documents and Settings\Trong Khanh\Cookies\trong khanh@AJDay3[1].txt]

```
userName
khanhkieu
localhost/AJDay3/
1024
3460197632
30159008
467067632
30159008
*
```

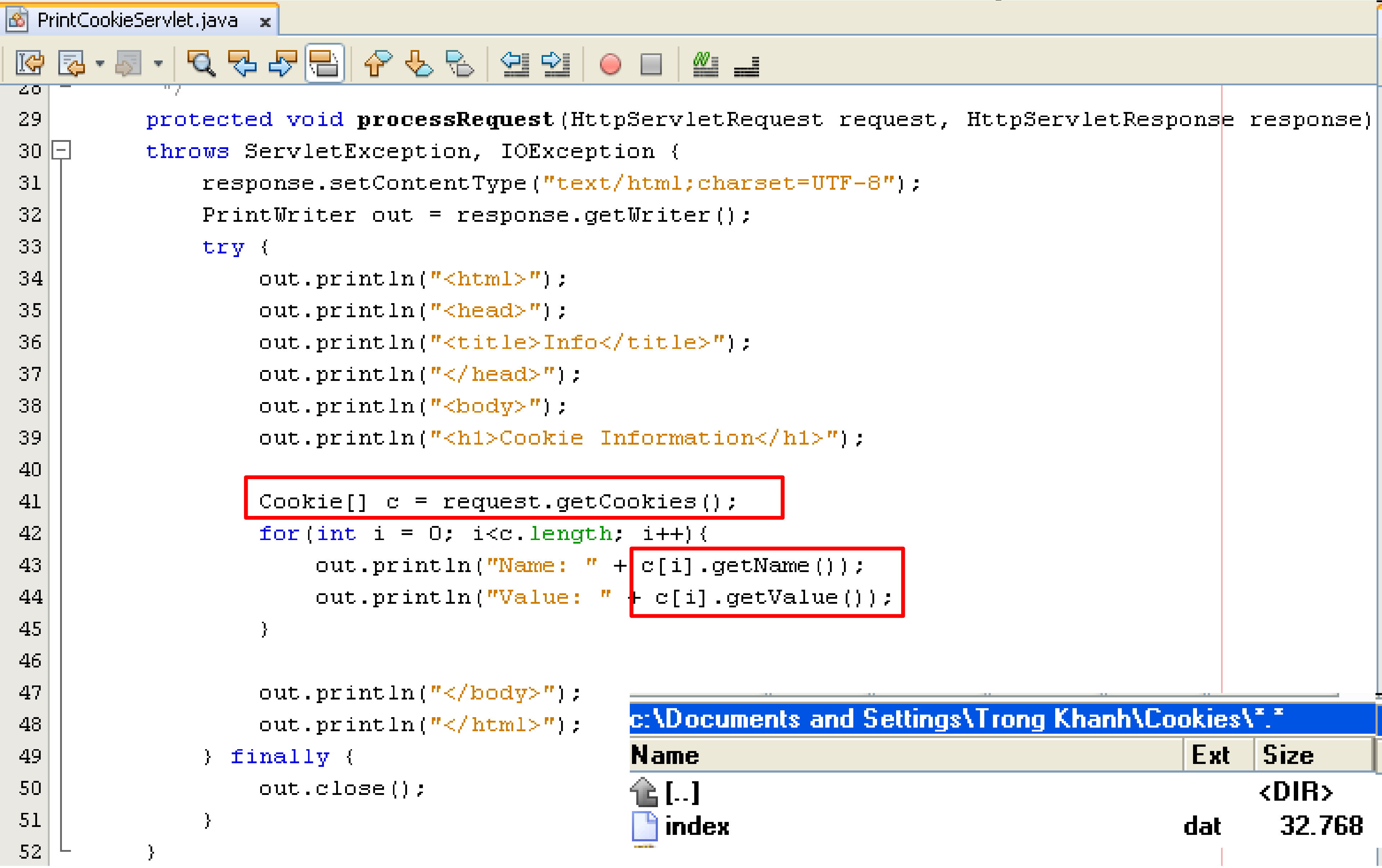
Cookies – Example



```java
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Add</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Adding Cookie processing</h1>");

            String sName = request.getParameter("txtName");
            Cookie cookie = new Cookie("userName", sName);
            cookie.setMaxAge(60*5);
            response.addCookie(cookie);
            out.println("<a href='PrintCookieServlet'>Print Cookie</a>");

            out.println("</body>");
            out.println("</html>");
        } finally {
            out.close();
        }
    }
```

Sessions & Listeners

Cookies – Example

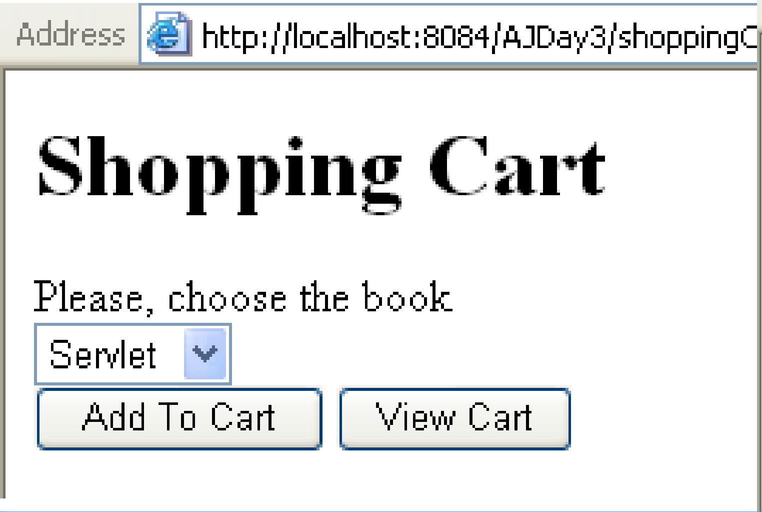


```java
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Info</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Cookie Information</h1>");

            Cookie[] c = request.getCookies();
            for(int i = 0; i<c.length; i++){
                out.println("Name: " + c[i].getName());
                out.println("Value: " + c[i].getValue());
            }

            out.println("</body>");
            out.println("</html>");
        } finally {
            out.close();
        }
    }
```

c:\Documents and Settings\Trong Khanh\Cookies\*.*

Name	Ext	Size
[..]		<DIR>
index	dat	32.768

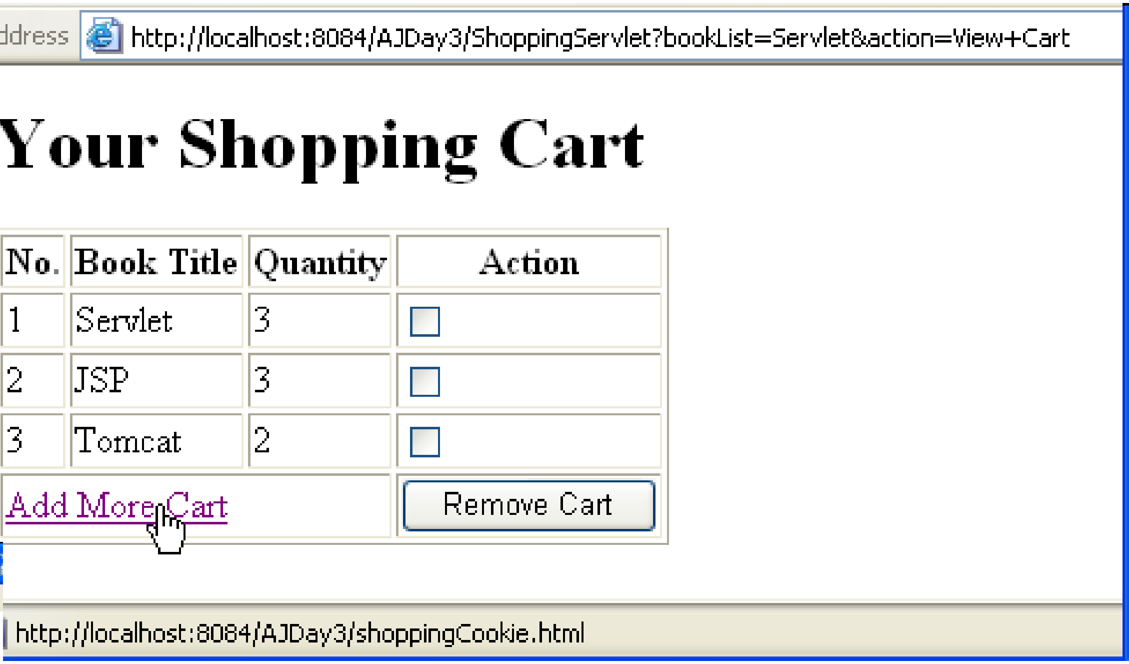
Shopping Cart using Cookies – Example



Address http://localhost:8084/AJDay3/shoppingC

Shopping Cart

Please, choose the book

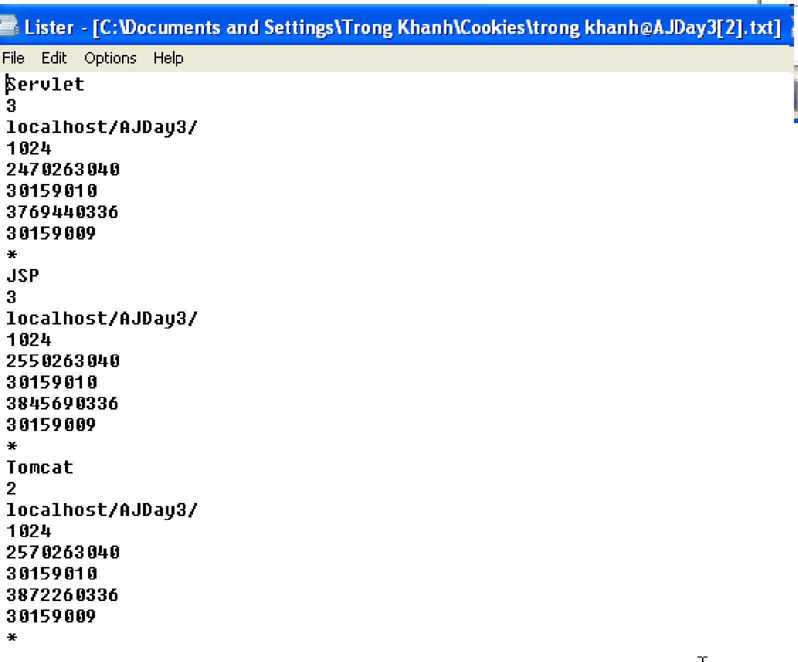


Address http://localhost:8084/AJDay3/ShoppingServlet?bookList=Servlet&action=View+Cart

Your Shopping Cart

No.	Book Title	Quantity	Action
1	Servlet	3	□
2	JSP	3	□
3	Tomcat	2	□
Add More Cart			Remove Cart

http://localhost:8084/AJDay3/shoppingCookie.html

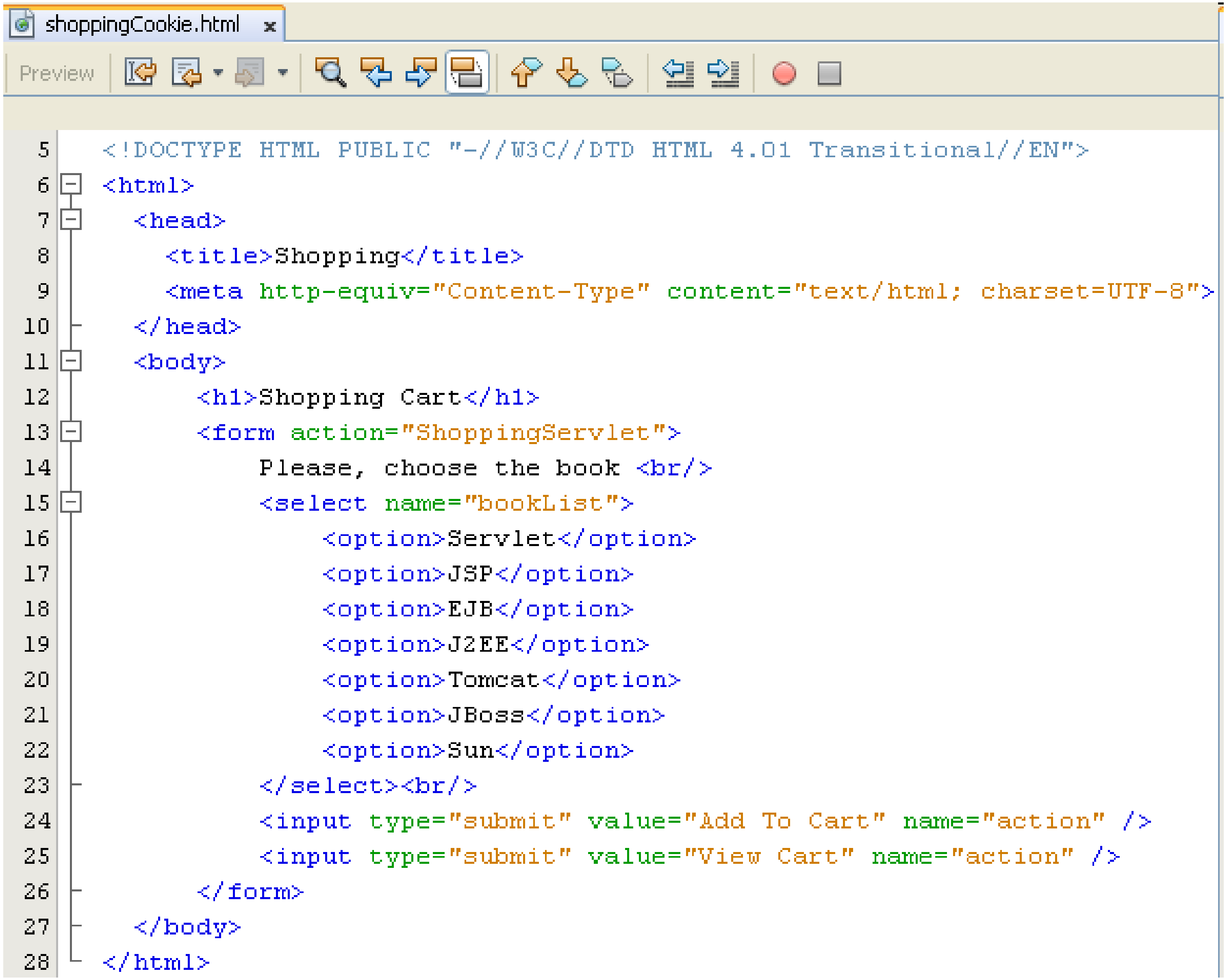


Lister - [C:\Documents and Settings\Trong Khanh\Cookies\trong khanh@AJDay3[2].txt]

```
Servlet
3
localhost/AJDay3/
1024
2470263040
30159010
3769440336
30159009
*
JSP
3
localhost/AJDay3/
1024
2550263040
30159010
3845690336
30159009
*
Tomcat
2
localhost/AJDay3/
1024
2570263040
30159010
3872260336
30159009
*
```

Sessions & Listeners

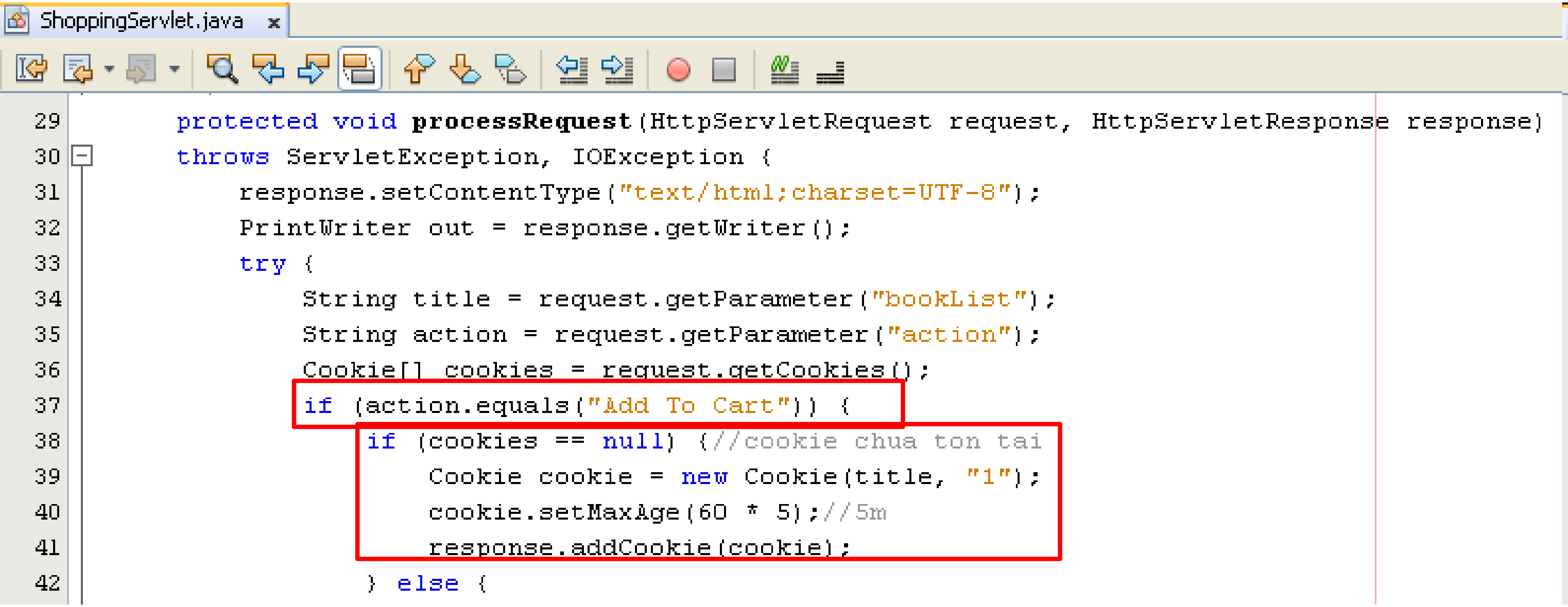
Shopping Cart using Cookies – Example



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Shopping</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
      <h1>Shopping Cart</h1>
      <form action="ShoppingServlet">
          Please, choose the book <br/>
          <select name="bookList">
              <option>Servlet</option>
              <option>JSP</option>
              <option>EJB</option>
              <option>J2EE</option>
              <option>Tomcat</option>
              <option>JBoss</option>
              <option>Sun</option>
          </select><br/>
          <input type="submit" value="Add To Cart" name="action" />
          <input type="submit" value="View Cart" name="action" />
      </form>
  </body>
</html>
```

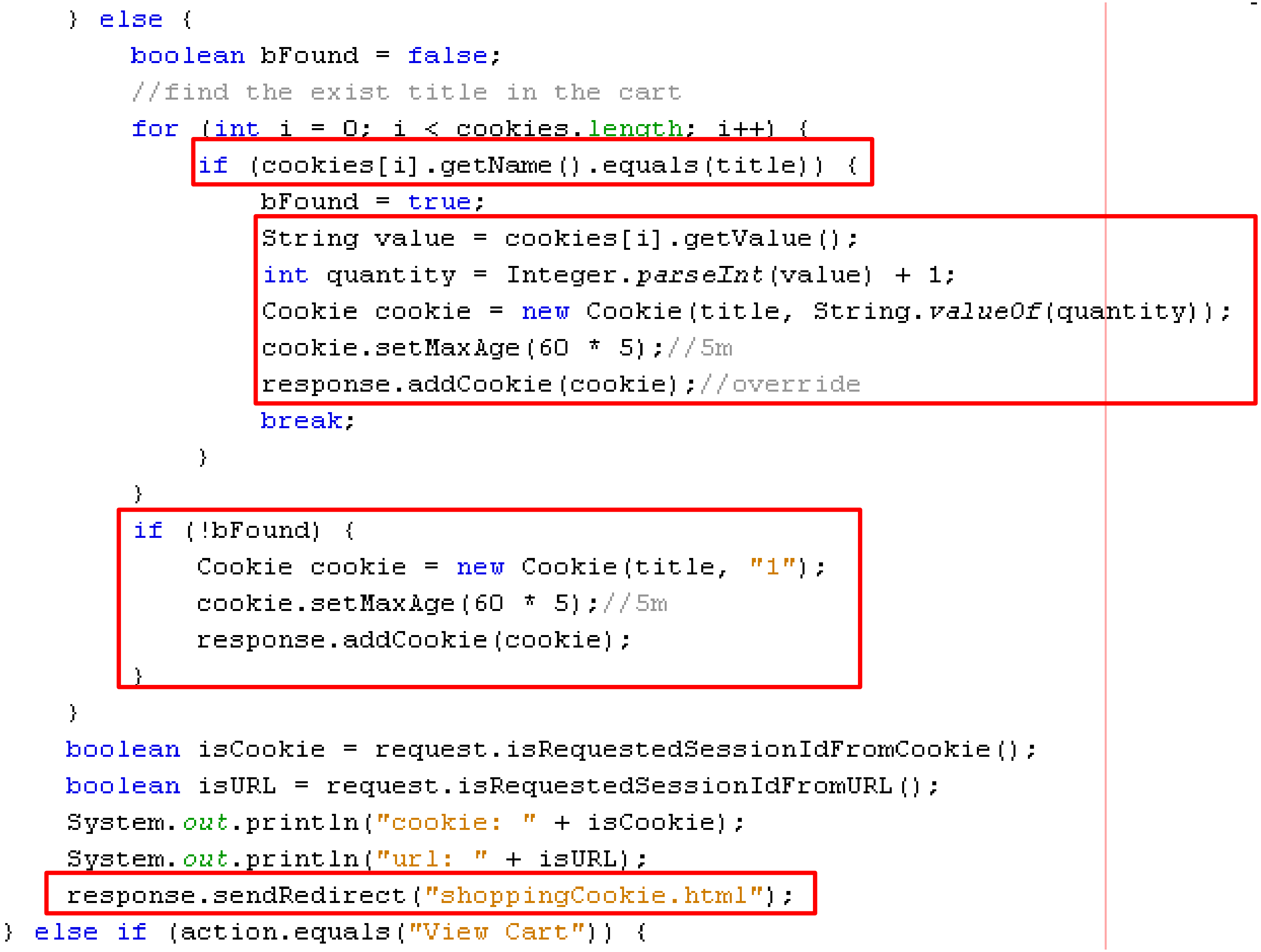
Sessions & Listeners

Shopping Cart using Cookies – Example



```
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            String title = request.getParameter("bookList");
            String action = request.getParameter("action");
            Cookie[] cookies = request.getCookies();
            if (action.equals("Add To Cart")) {
                if (cookies == null) {//cookie chua ton tai
                    Cookie cookie = new Cookie(title, "1");
                    cookie.setMaxAge(60 * 5);//5m
                    response.addCookie(cookie);
                } else {
```

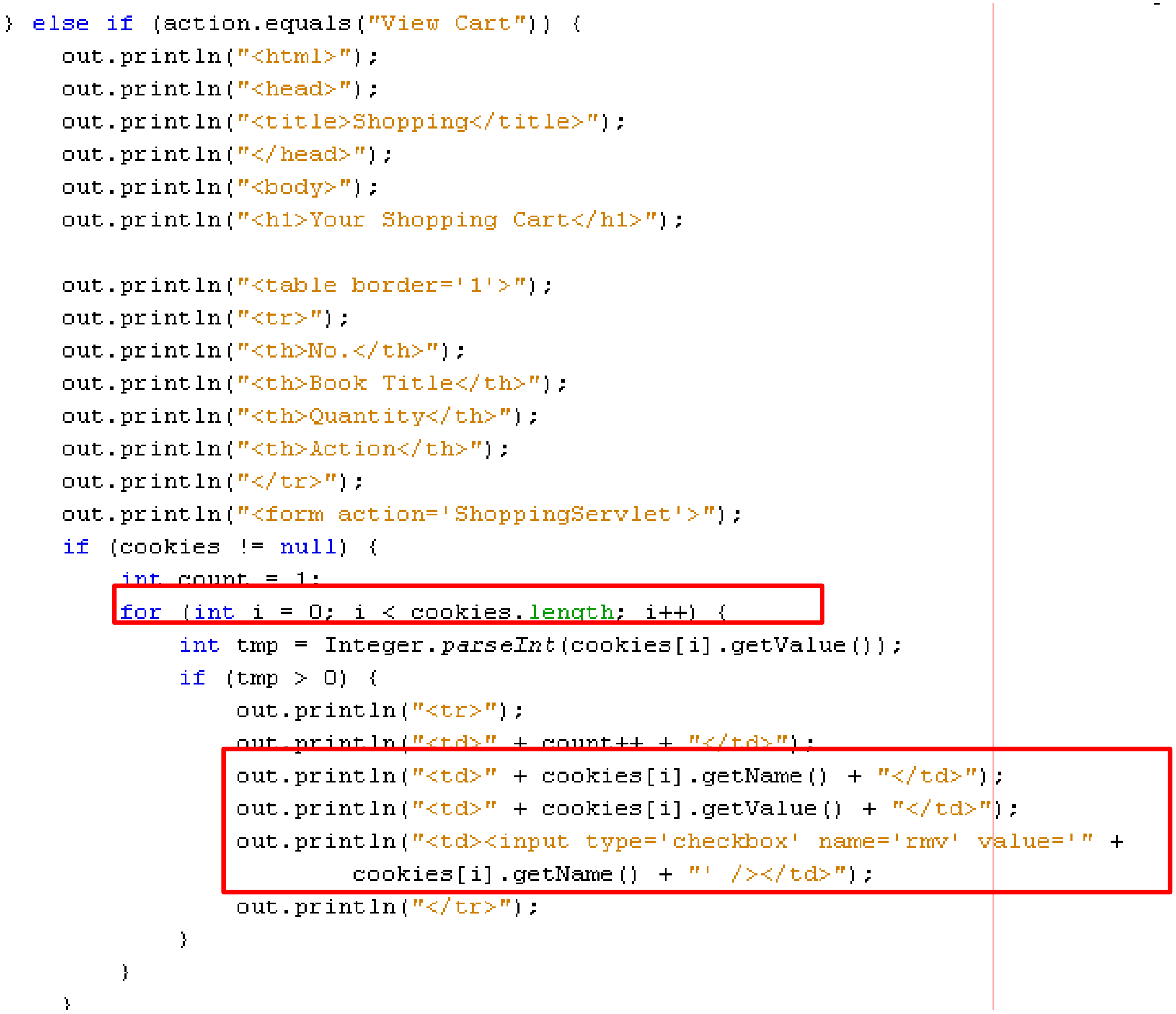
Shopping Cart using Cookies – Example



```
    } else {
        boolean bFound = false;
        //find the exist title in the cart
        for (int i = 0; i < cookies.length; i++) {
            if (cookies[i].getName().equals(title)) {
                bFound = true;
                String value = cookies[i].getValue();
                int quantity = Integer.parseInt(value) + 1;
                Cookie cookie = new Cookie(title, String.valueOf(quantity));
                cookie.setMaxAge(60 * 5);//5m
                response.addCookie(cookie);//override
                break;
            }
        }
        if (!bFound) {
            Cookie cookie = new Cookie(title, "1");
            cookie.setMaxAge(60 * 5);//5m
            response.addCookie(cookie);
        }
    }
    boolean isCookie = request.isRequestedSessionIdFromCookie();
    boolean isURL = request.isRequestedSessionIdFromURL();
    System.out.println("cookie: " + isCookie);
    System.out.println("url: " + isURL);
    response.sendRedirect("shoppingCookie.html");
} else if (action.equals("View Cart")) {
```

Sessions & Listeners

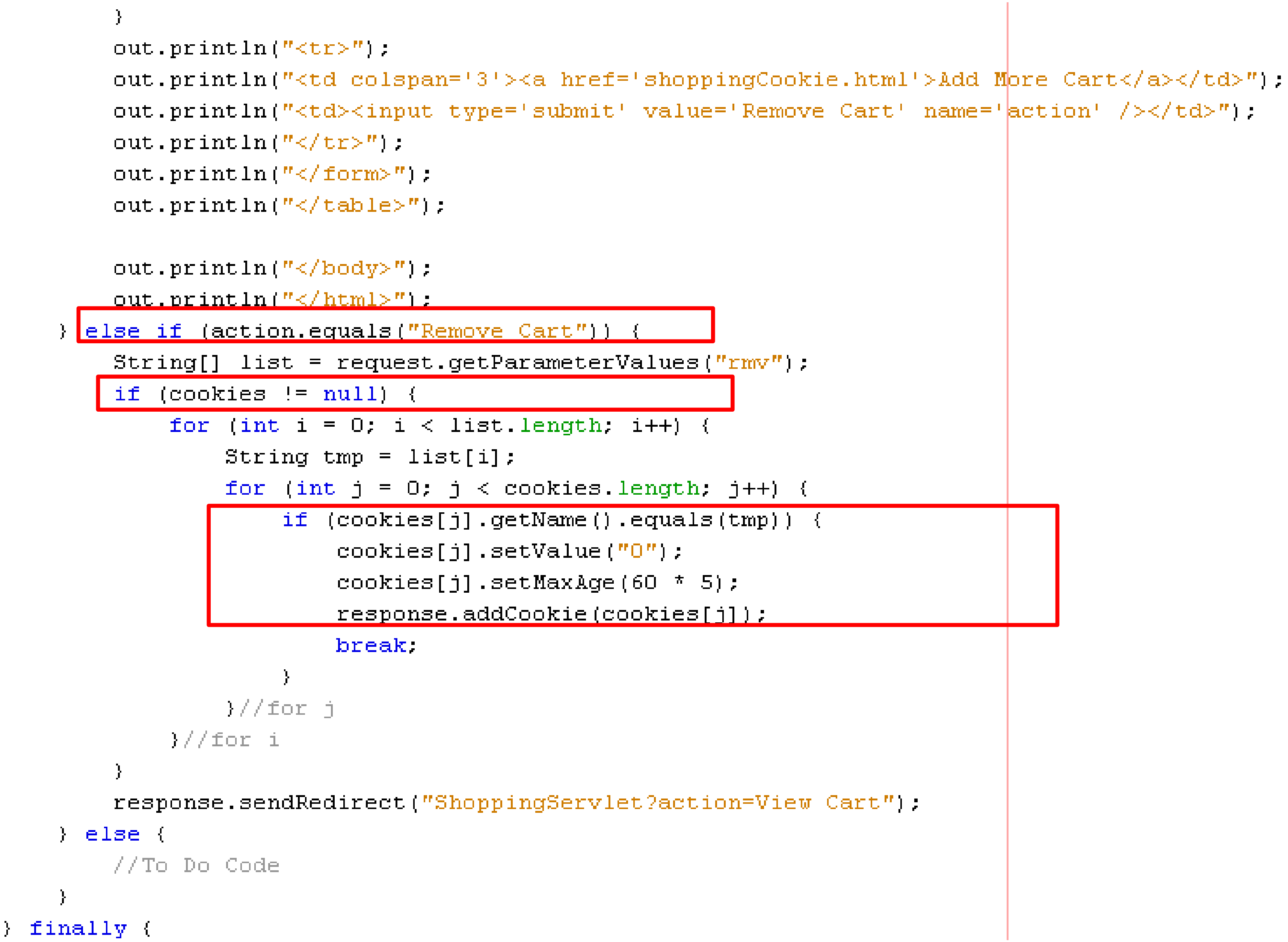
Shopping Cart using Cookies – Example



```
} else if (action.equals("View Cart")) {
    out.println("<html>");
    out.println("<head>");
    out.println("<title>Shopping</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<h1>Your Shopping Cart</h1>");

    out.println("<table border='1'>");
    out.println("<tr>");
    out.println("<th>No.</th>");
    out.println("<th>Book Title</th>");
    out.println("<th>Quantity</th>");
    out.println("<th>Action</th>");
    out.println("</tr>");
    out.println("<form action='ShoppingServlet'>");
    if (cookies != null) {
        int count = 1;
        for (int i = 0; i < cookies.length; i++) {
            int tmp = Integer.parseInt(cookies[i].getValue());
            if (tmp > 0) {
                out.println("<tr>");
                out.println("<td>" + count++ + "</td>");
                out.println("<td>" + cookies[i].getName() + "</td>");
                out.println("<td>" + cookies[i].getValue() + "</td>");
                out.println("<td><input type='checkbox' name='rmv' value='" +
                        cookies[i].getName() + "' /></td>");
                out.println("</tr>");
            }
        }
    }
```

Sessions & Listeners

Shopping Cart using Cookies – Example



```
        }
        out.println("<tr>");
        out.println("<td colspan='3'><a href='shoppingCookie.html'>Add More Cart</a></td>");
        out.println("<td><input type='submit' value='Remove Cart' name='action' /></td>");
        out.println("</tr>");
        out.println("</form>");
        out.println("</table>");

        out.println("</body>");
        out.println("</html>");
    } else if (action.equals("Remove Cart")) {
        String[] list = request.getParameterValues("rmv");
        if (cookies != null) {
            for (int i = 0; i < list.length; i++) {
                String tmp = list[i];
                for (int j = 0; j < cookies.length; j++) {
                    if (cookies[j].getName().equals(tmp)) {
                        cookies[j].setValue("0");
                        cookies[j].setMaxAge(60 * 5);
                        response.addCookie(cookies[j]);
                        break;
                    }
                }//for j
            }//for i
        }
        response.sendRedirect("ShoppingServlet?action=View Cart");
    } else {
        //To Do Code
    }
} finally {
```

Sessions & Listeners

HttpSession interface

- Identifying user in a multi-page request scenario and information about that user
- Is used to **created** a **session between** the **client and server** by servlet container
  - When **users make** a **request**, the **server signs** it a **session object** and a **unique session ID**
  - The session ID matches the user with the session object in subsequent requests
  - The **session ID and the session object** are **passed** along with the **request** to the **server**
- **Session Timeout**
  - Is necessary as session utilizes the memory locations
  - Prevent the number of session increasing infinitely.
  - Set either in the web.xml file or can be set by the method **setMaxInactiveInterval**()

Sessions & Listeners
HttpSession interface Methods

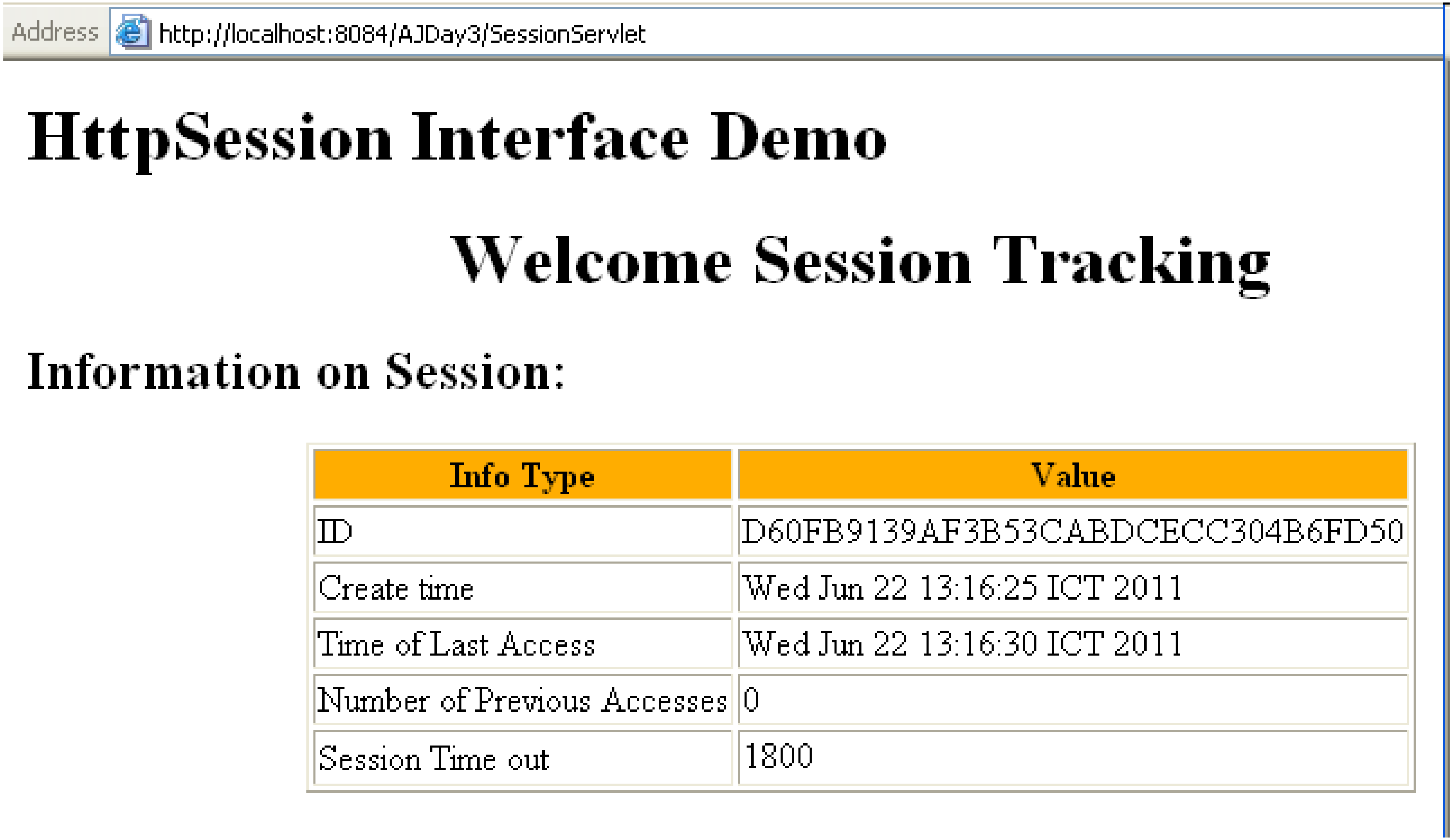
Methods	Descriptions
getSession	- **request.getSession(boolean create);** - Obtain a current session objects - The getSession() method with true parameter is used to create a new session (no current session)
getId	- **public String getId()** - Returns a string containing the unique identifier assigned to this session. The servlet container assigns the identifier and it is implementation independent
getAttribute	- **public Object getAttribute(String name)** - Returns the object which is bound with the specified name in the session. - Returns null in case there is no object bound under the name
setAttribute	- **public void setAttribute(String name, Object value)** - Binds the object to the current session by using the specified name. - In case of repetition of name of the object bound to the session, the object is replaced
removeAttribute	- **public void removeAttribute(String name)** - Removes the object bound with the specified name from the session. - In case of absence of object bound to specified name, this method does nothing.
getCreationTime	- **public long getCreationTime()** - Returns the creation time of session.

Sessions & Listeners

HttpSession interface Methods

Methods	Descriptions
getLastAccessedTime	- **public long getLastAccessedTime()** - Returns the last accessed Time of session
getMaxInactiveInterval	- **public int getMaxInactiveInterval()** - Returns the maximum time interval, in seconds, for which the servlet container will keep the session alive between the client accesses
setMaxInactiveInterval	- **public void setMaxInactiveInterval(int interval)** - Specifies the time, in seconds, between the client requests before the servlet container invalidates the current session
isNew	- **public boolean isNew()** - Returns true if the client is unaware about the session or choose not to be part of the session
invalidate	- **public void invalidate()** - Invalidates the session and the objects bound to the session are bounded. This method throws IllegalStateException if called on already invalidated session - To avoid the hacker from causing any harm - Destroys the data in a session that another servlet or JSP might require in future. Therefore, invalidating a session should be done cautiously as sessions are associated with client, not with individual servlets or JSP pages

HttpSession interface – Example



http://localhost:8084/AJDay3/SessionServlet

HttpSession Interface Demo

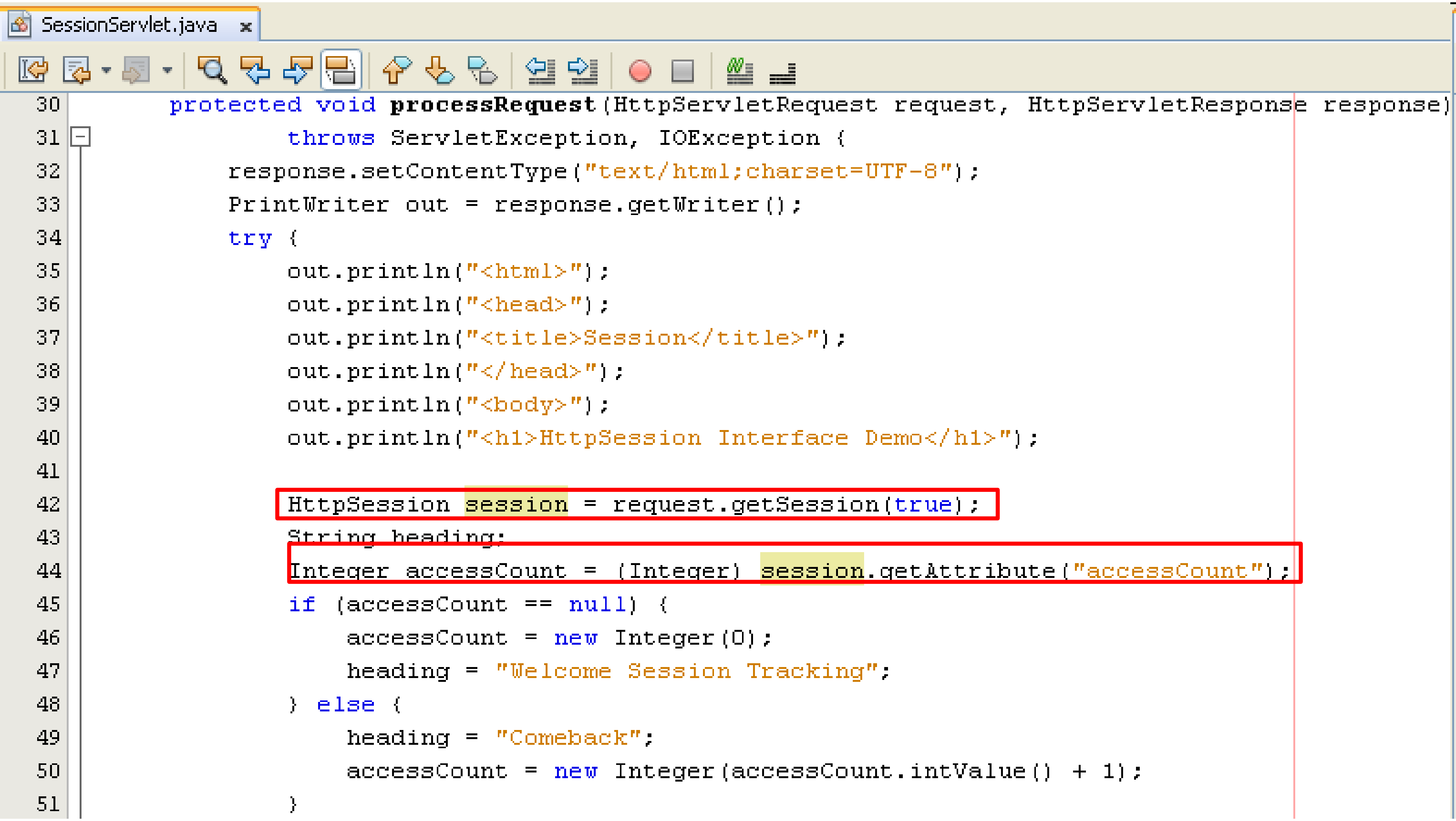
Welcome Session Tracking

Information on Session:

Info Type	Value
ID	D60FB9139AF3B53CABDCECC304B6FD50
Create time	Wed Jun 22 13:16:25 ICT 2011
Time of Last Access	Wed Jun 22 13:16:30 ICT 2011
Number of Previous Accesses	0
Session Time out	1800

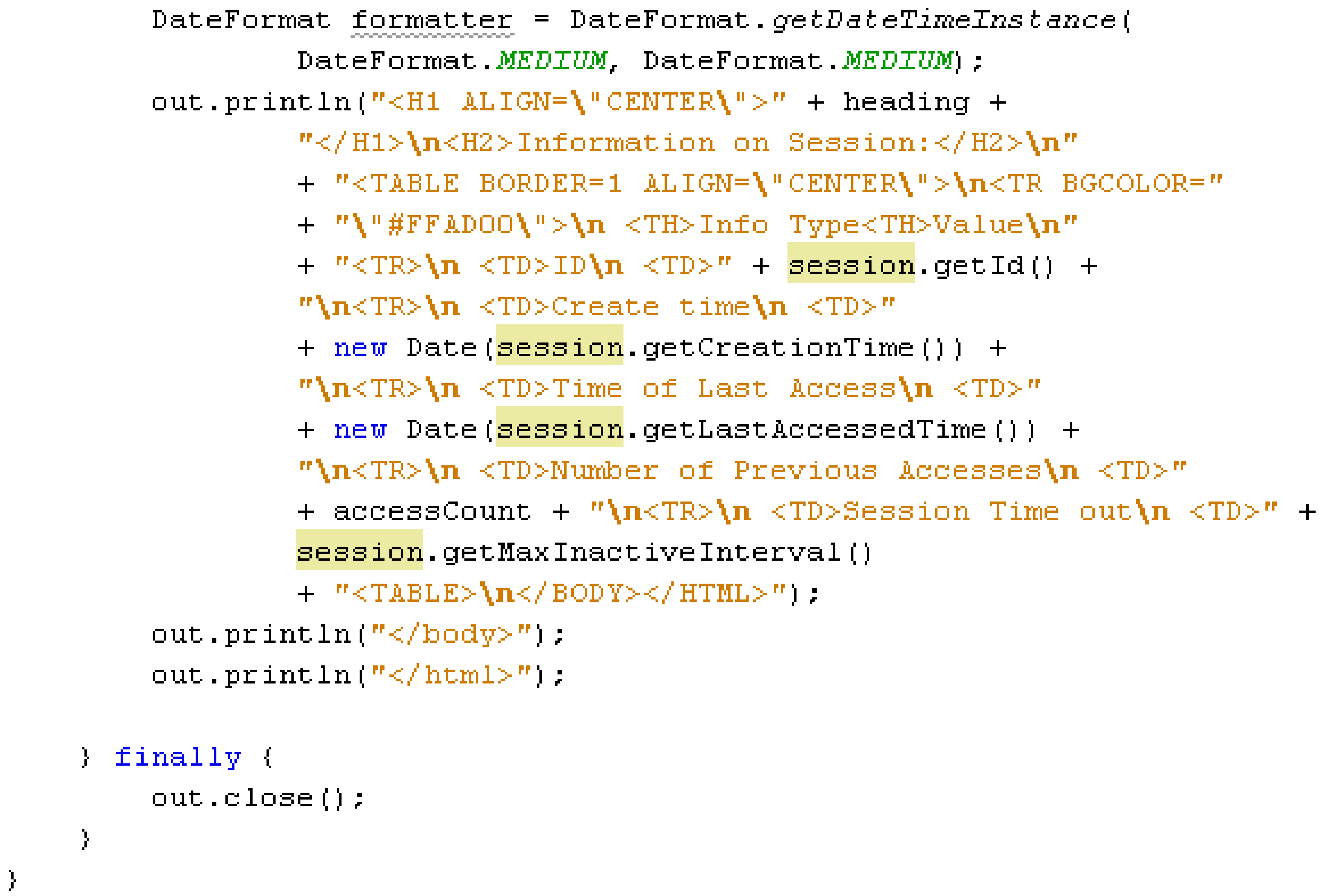
Sessions & Listeners

HttpSession interface – Example



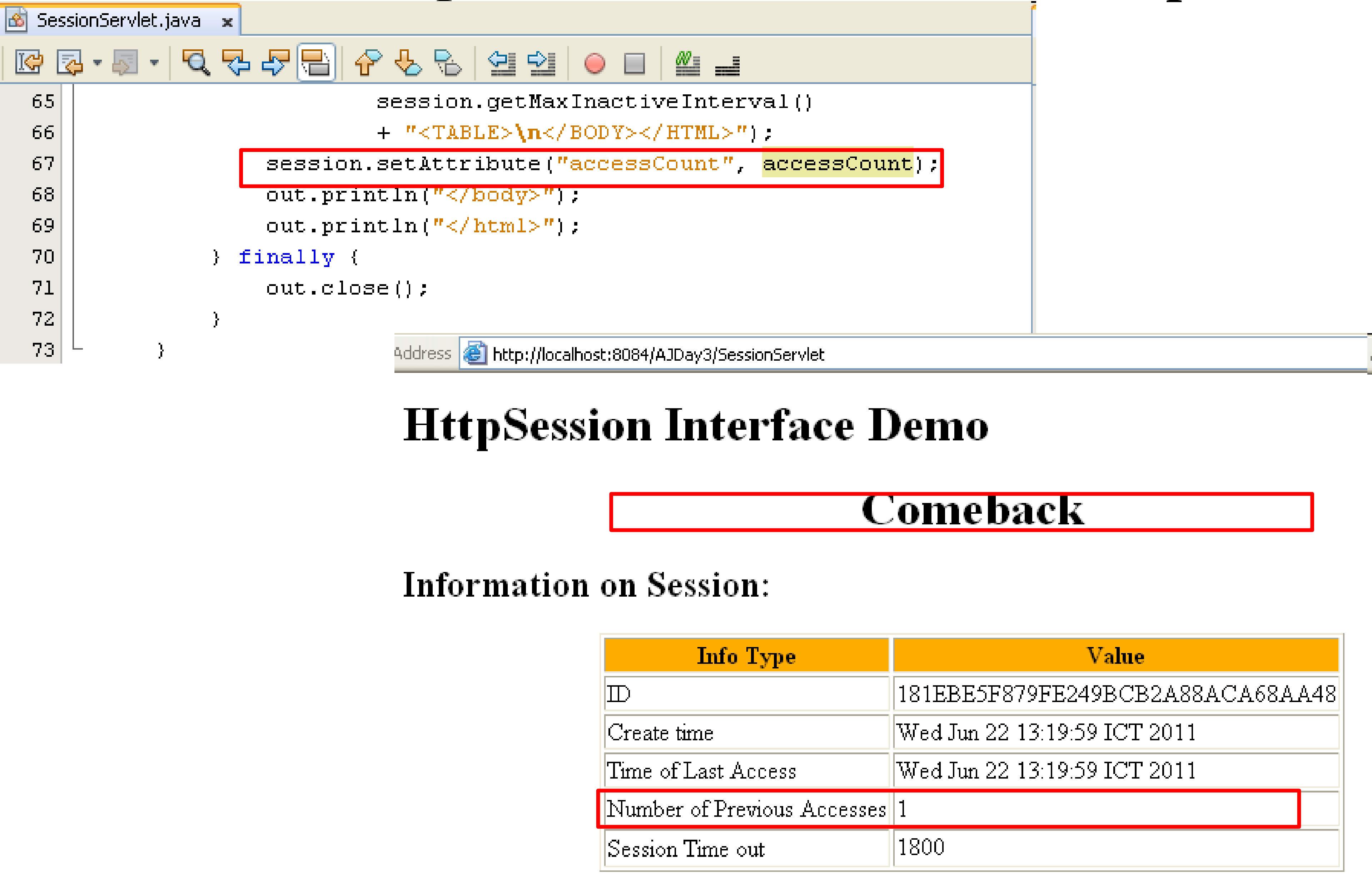
```
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Session</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>HttpSession Interface Demo</h1>");

            HttpSession session = request.getSession(true);
            String heading;
            Integer accessCount = (Integer) session.getAttribute("accessCount");
            if (accessCount == null) {
                accessCount = new Integer(0);
                heading = "Welcome Session Tracking";
            } else {
                heading = "Comeback";
                accessCount = new Integer(accessCount.intValue() + 1);
            }
```

Sessions & Listeners
HttpSession interface – Example



```
        DateFormat formatter = DateFormat.getDateTimeInstance(
                DateFormat.MEDIUM, DateFormat.MEDIUM);
        out.println("<H1 ALIGN=\"CENTER\">" + heading +
                "</H1>\n<H2>Information on Session:</H2>\n"
                + "<TABLE BORDER=1 ALIGN=\"CENTER\">\n<TR BGCOLOR="
                + "\"#FFAD00\">\n <TH>Info Type<TH>Value\n"
                + "<TR>\n <TD>ID\n <TD>" + session.getId() +
                "\n<TR>\n <TD>Create time\n <TD>"
                + new Date(session.getCreationTime()) +
                "\n<TR>\n <TD>Time of Last Access\n <TD>"
                + new Date(session.getLastAccessedTime()) +
                "\n<TR>\n <TD>Number of Previous Accesses\n <TD>"
                + accessCount + "\n<TR>\n <TD>Session Time out\n <TD>" +
                session.getMaxInactiveInterval()
                + "<TABLE>\n</BODY></HTML>");
        out.println("</body>");
        out.println("</html>");

    } finally {
        out.close();
    }
}
```

Sessions & Listeners
HttpSession interface – Example



SessionServlet.java

```
                      session.getMaxInactiveInterval()
                      + "<TABLE>\n</BODY></HTML>");
              session.setAttribute("accessCount", accessCount);
              out.println("</body>");
              out.println("</html>");
          } finally {
              out.close();
          }
      }
```

Address http://localhost:8084/AJDay3/SessionServlet

HttpSession Interface Demo

Comeback

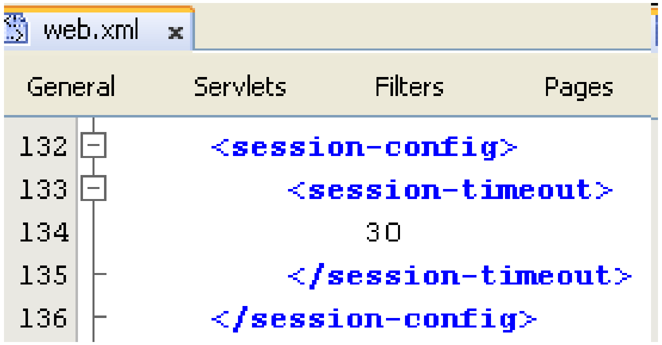
Information on Session:

Info Type	Value
ID	181EBE5F879FE249BCB2A88ACA68AA48
Create time	Wed Jun 22 13:19:59 ICT 2011
Time of Last Access	Wed Jun 22 13:19:59 ICT 2011
Number of Previous Accesses	1
Session Time out	1800

Sessions & Listeners

HttpSession interface

- **Distributed Session**
  - A **session** is **available** to be **shared between web resources** in a single web application (*e.g. a session **cannot cross web application boundaries***)
- Session **Death** is **controlled** in one of **3 ways**
  - **Application Server Global Default**
  - **Web Application Default (minutes)**
    - A negative value or zero value causes the session to never expire



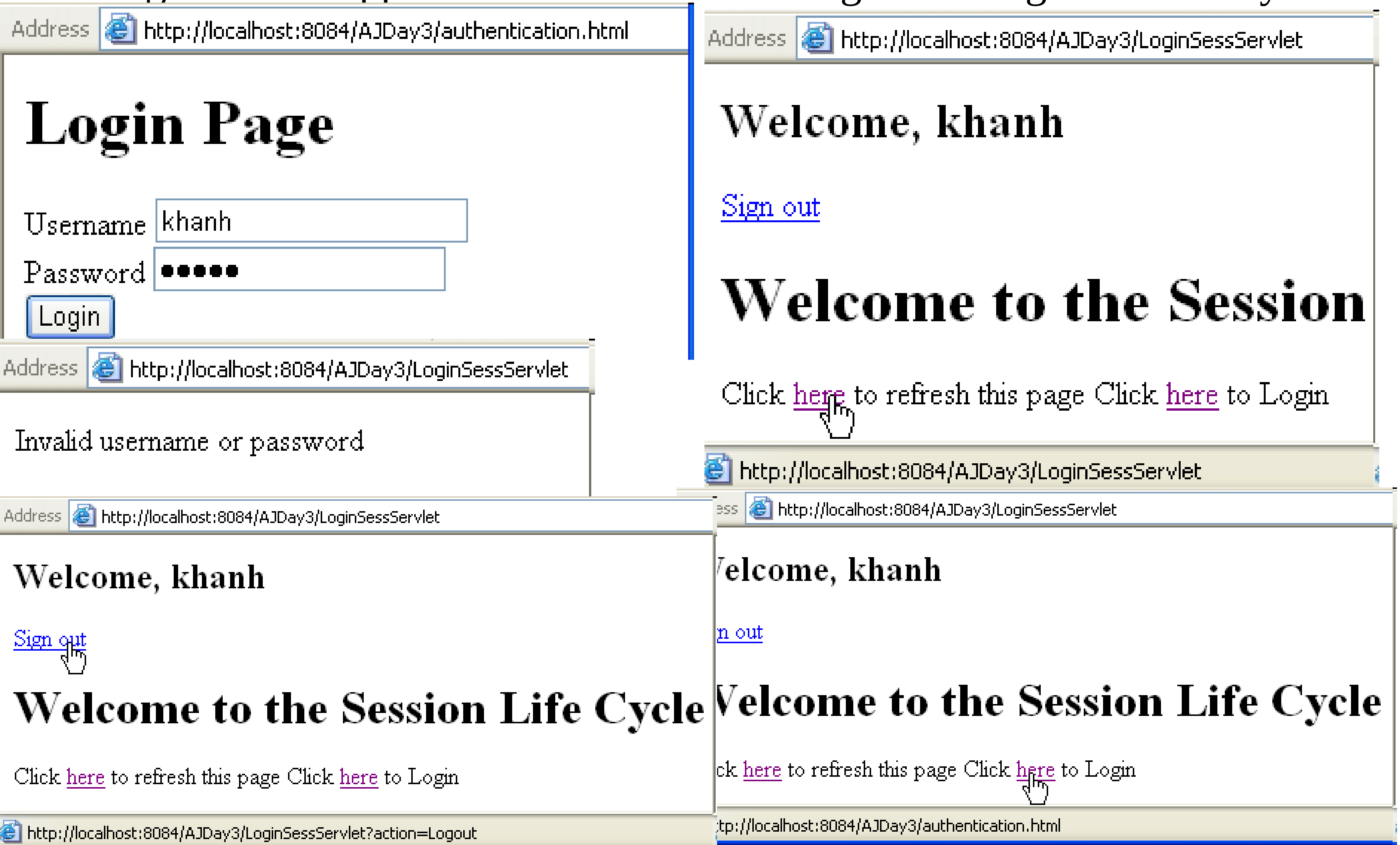
```
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
```

  - Individual Session Setting using **setMaxInactivateInterval**() method
    - A negative value supplied as an argument causes the session to never expire
- Other Session APIs
  - **HttpSession.getServletContext**() returns the SessionContext that the session is attached

Sessions & Listeners

HttpSession interface – Example

Building the Web Application can allow user log in and log out on the system



Address http://localhost:8084/AJDay3/authentication.html

Login Page

Username khanh

Password ●●●●●

Login

Address http://localhost:8084/AJDay3/LoginSessServlet

Invalid username or password

Address http://localhost:8084/AJDay3/LoginSessServlet

Welcome, khanh

Sign out

Welcome to the Session

Click here to refresh this page Click here to Login

http://localhost:8084/AJDay3/LoginSessServlet

Address http://localhost:8084/AJDay3/LoginSessServlet

Welcome, khanh

Sign out

Welcome to the Session Life Cycle

Click here to refresh this page Click here to Login

http://localhost:8084/AJDay3/LoginSessServlet?action=Logout

ess http://localhost:8084/AJDay3/LoginSessServlet

/elcome, khanh

n out

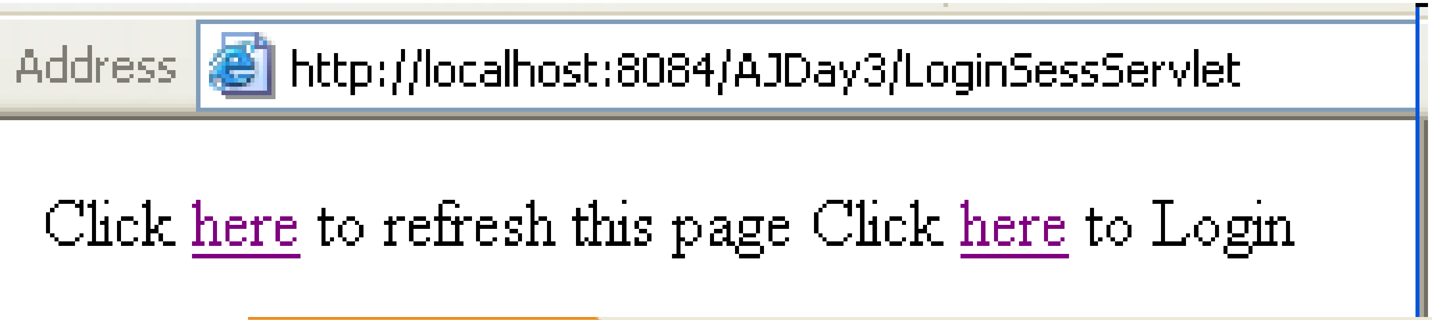
Velcome to the Session Life Cycle

ck here to refresh this page Click here to Login

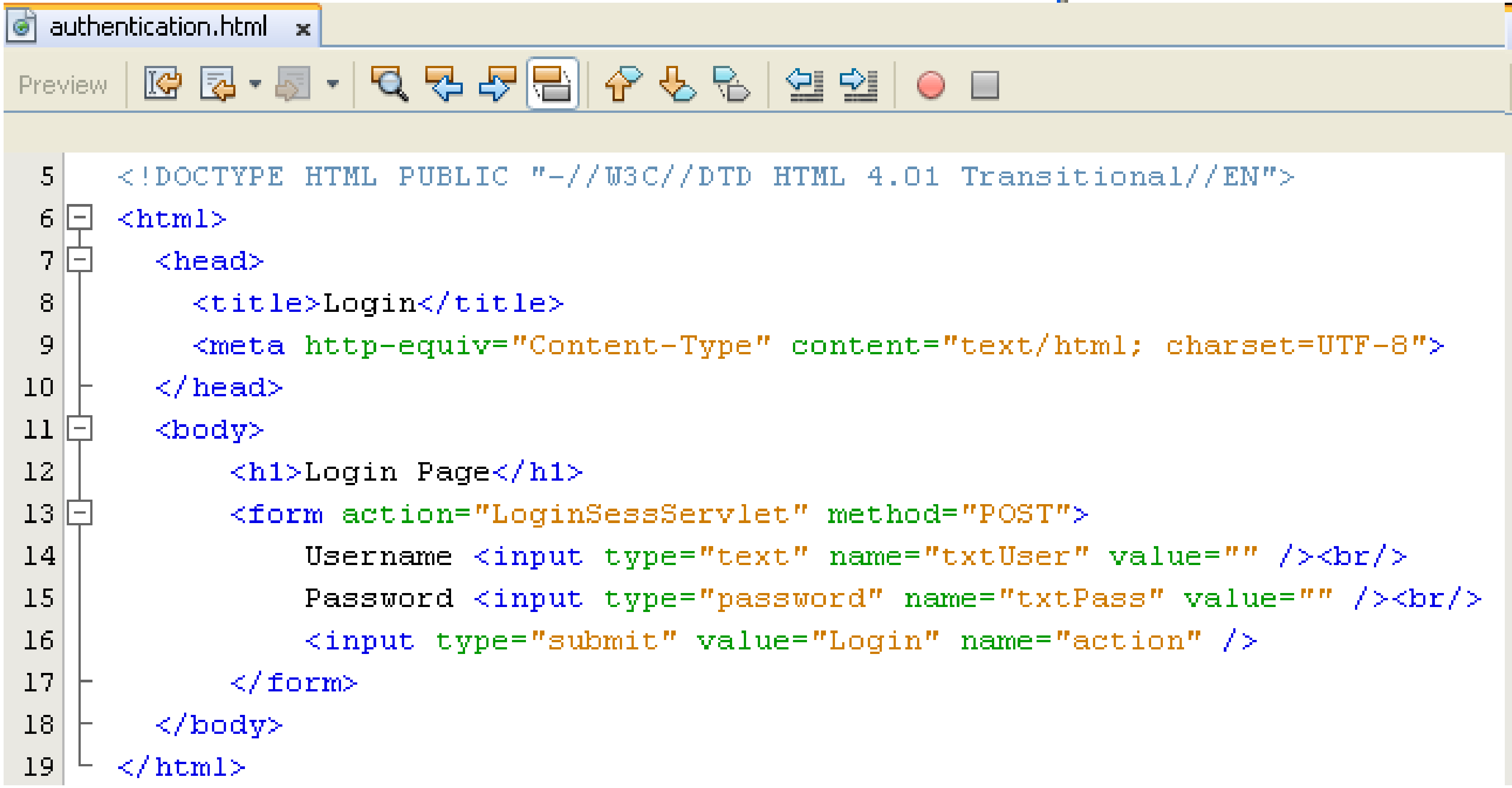
tp://localhost:8084/AJDay3/authentication.html

Sessions & Listeners

HttpSession interface – Example


Address http://localhost:8084/AJDay3/LoginSessServlet

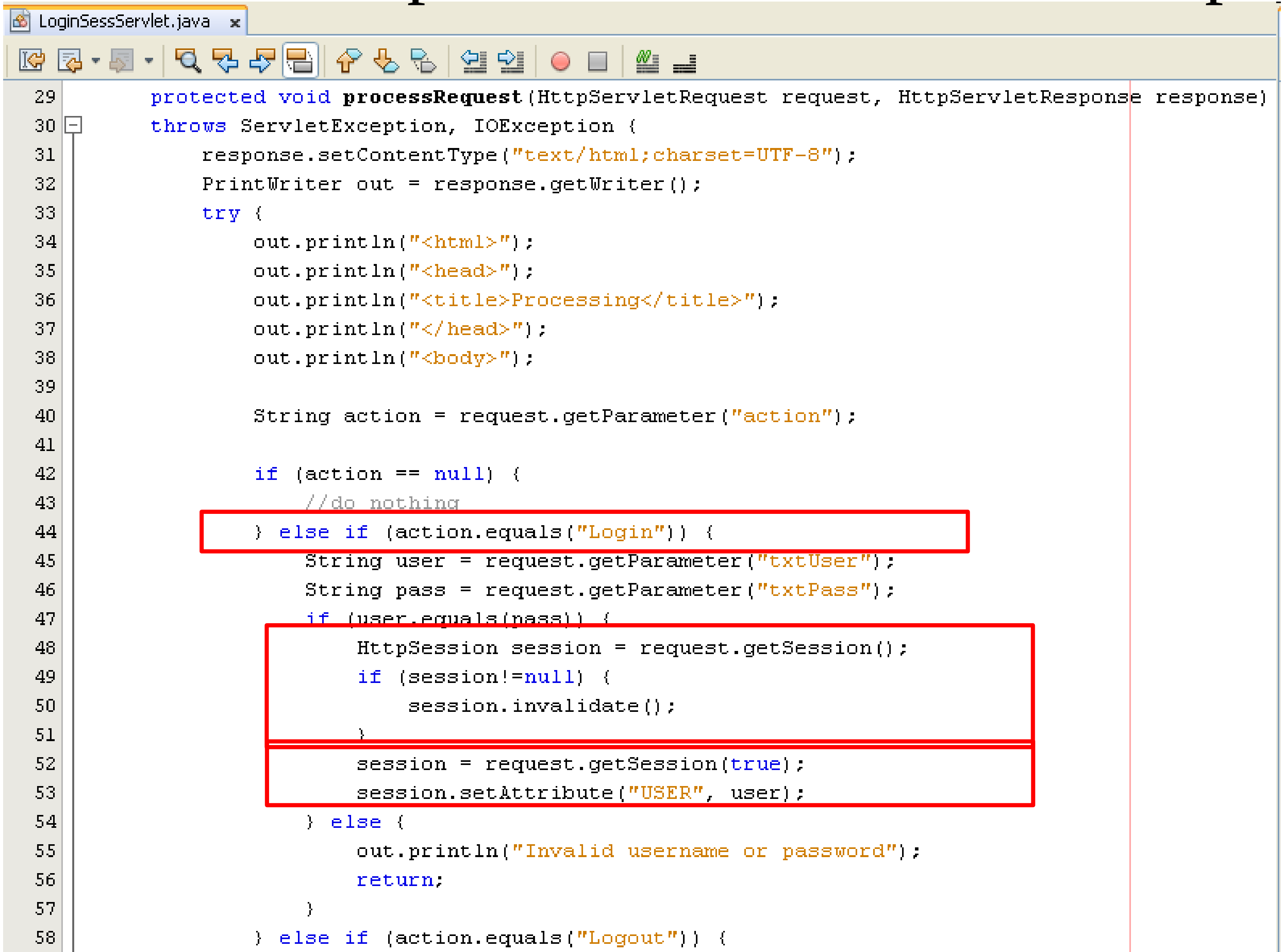
Click here to refresh this page Click here to Login



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Login</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
      <h1>Login Page</h1>
      <form action="LoginSessServlet" method="POST">
          Username <input type="text" name="txtUser" value="" /><br/>
          Password <input type="password" name="txtPass" value="" /><br/>
          <input type="submit" value="Login" name="action" />
      </form>
  </body>
</html>
```

Sessions & Listeners

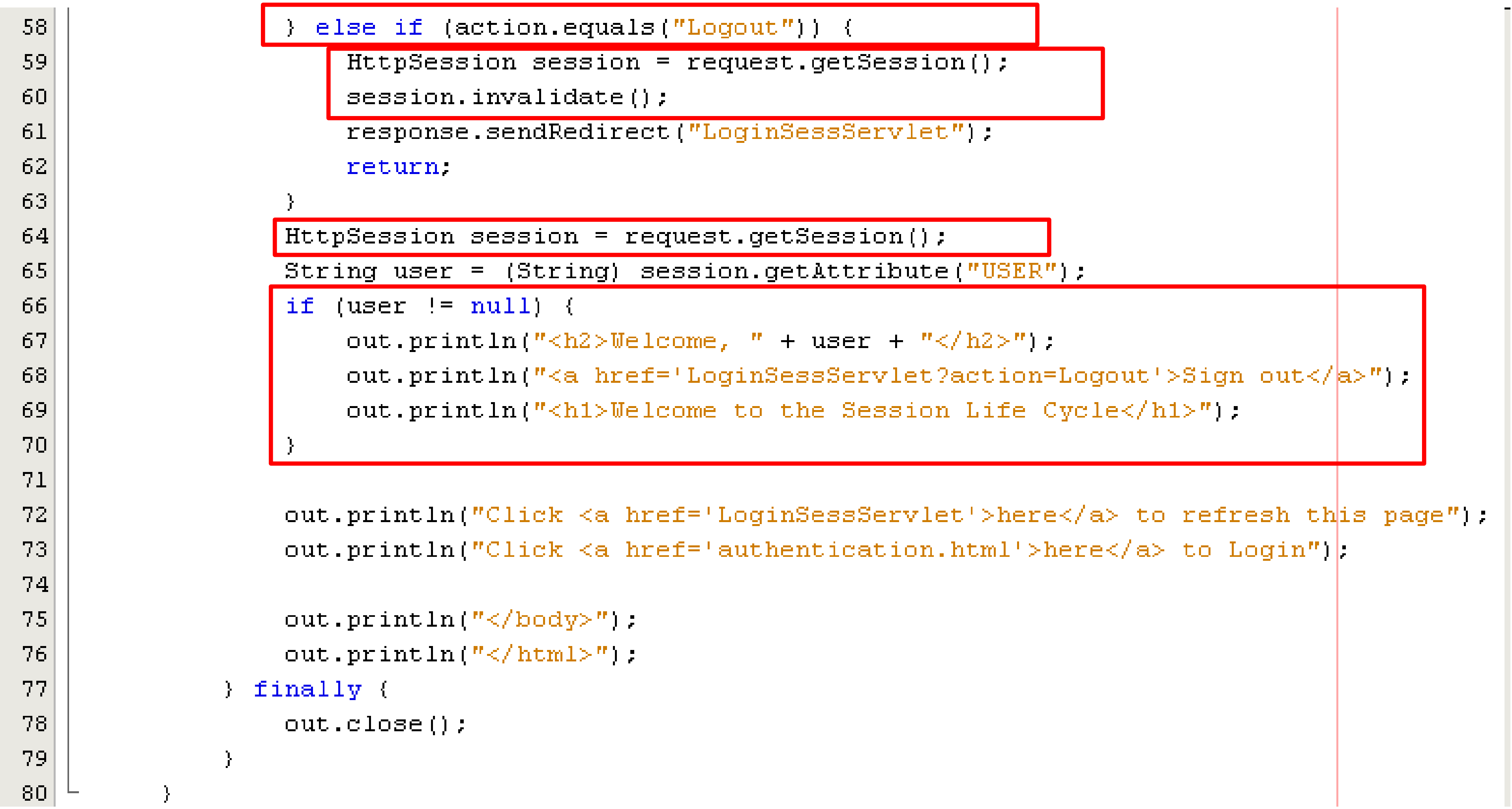
HttpSession interface – Example



```
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Processing</title>");
            out.println("</head>");
            out.println("<body>");

            String action = request.getParameter("action");

            if (action == null) {
                //do nothing
            } else if (action.equals("Login")) {
                String user = request.getParameter("txtUser");
                String pass = request.getParameter("txtPass");
                if (user.equals(pass)) {
                    HttpSession session = request.getSession();
                    if (session!=null) {
                        session.invalidate();
                    }
                    session = request.getSession(true);
                    session.setAttribute("USER", user);
                } else {
                    out.println("Invalid username or password");
                    return;
                }
            } else if (action.equals("Logout")) {
```

Sessions & Listeners

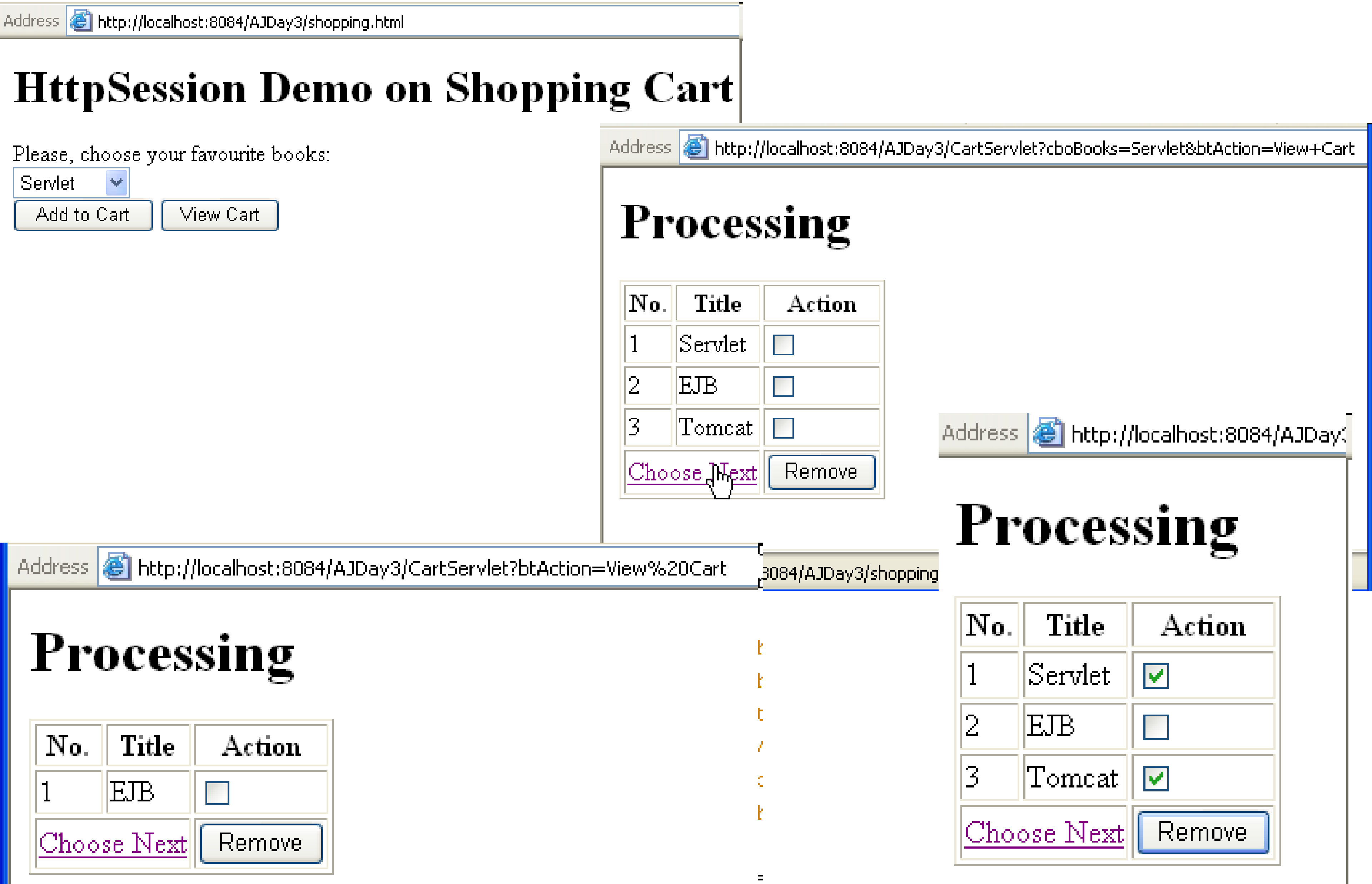
HttpSession interface – Example



```
            } else if (action.equals("Logout")) {
                HttpSession session = request.getSession();
                session.invalidate();
                response.sendRedirect("LoginSessServlet");
                return;
            }
            HttpSession session = request.getSession();
            String user = (String) session.getAttribute("USER");
            if (user != null) {
                out.println("<h2>Welcome, " + user + "</h2>");
                out.println("<a href='LoginSessServlet?action=Logout'>Sign out</a>");
                out.println("<h1>Welcome to the Session Life Cycle</h1>");
            }

            out.println("Click <a href='LoginSessServlet'>here</a> to refresh this page");
            out.println("Click <a href='authentication.html'>here</a> to Login");

            out.println("</body>");
            out.println("</html>");
        } finally {
            out.close();
        }
    }
```

Sessions & Listeners

HttpSession interface – Example



Address http://localhost:8084/AJDay3/shopping.html

HttpSession Demo on Shopping Cart

Please, choose your favourite books:

Servlet

Address http://localhost:8084/AJDay3/CartServlet?cboBooks=Servlet&btAction=View+Cart

Processing

No.	Title	Action
1	Servlet	□
2	EJB	□
3	Tomcat	□
Choose Next	Remove	

Address http://localhost:8084/AJDay3/CartServlet?btAction=View%20Cart

Processing

No.	Title	Action
1	EJB	□
Choose Next	Remove	

3084/AJDay3/shopping

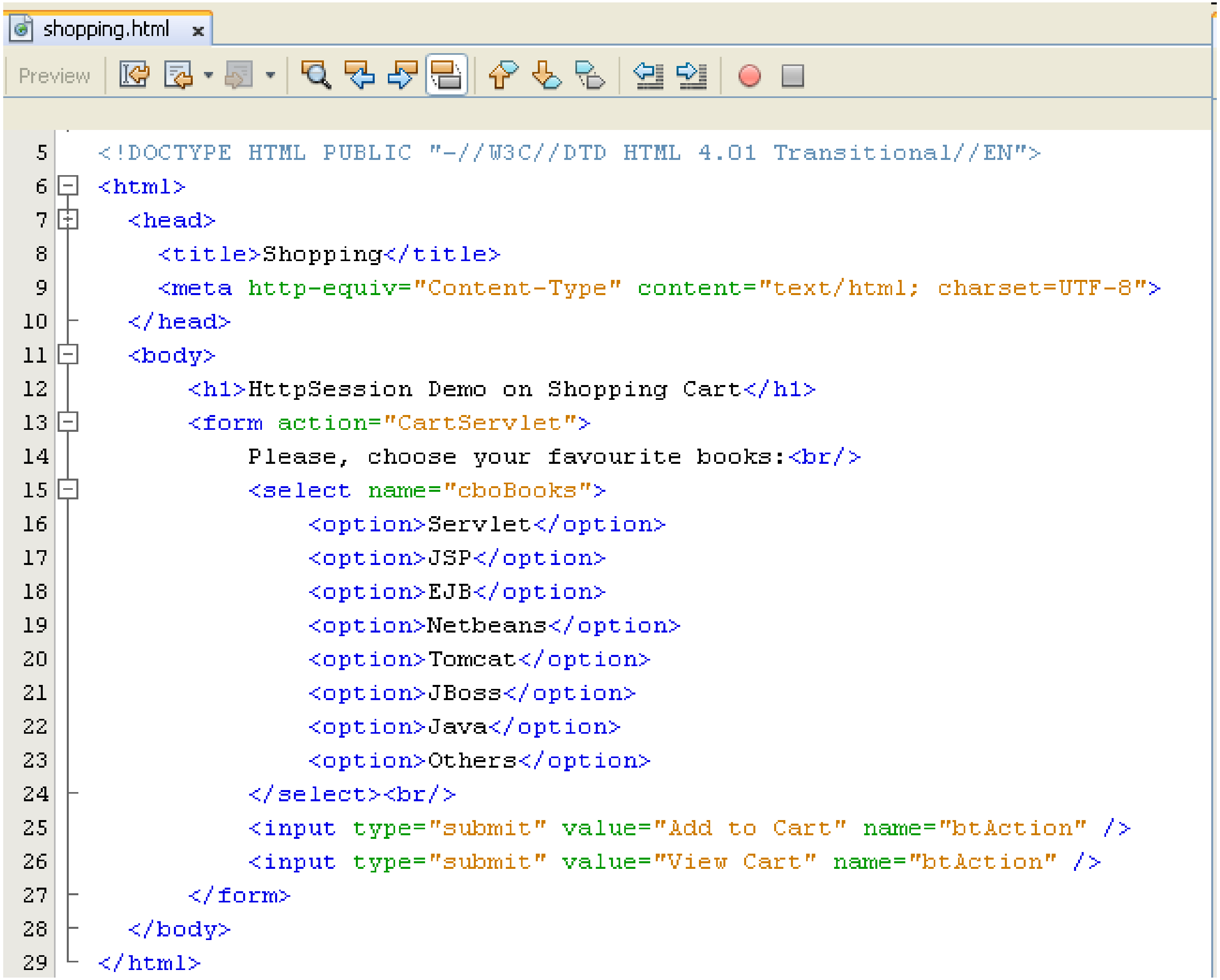
Address http://localhost:8084/AJDay3

Processing

No.	Title	Action
1	Servlet	☑
2	EJB	□
3	Tomcat	☑
Choose Next	Remove	

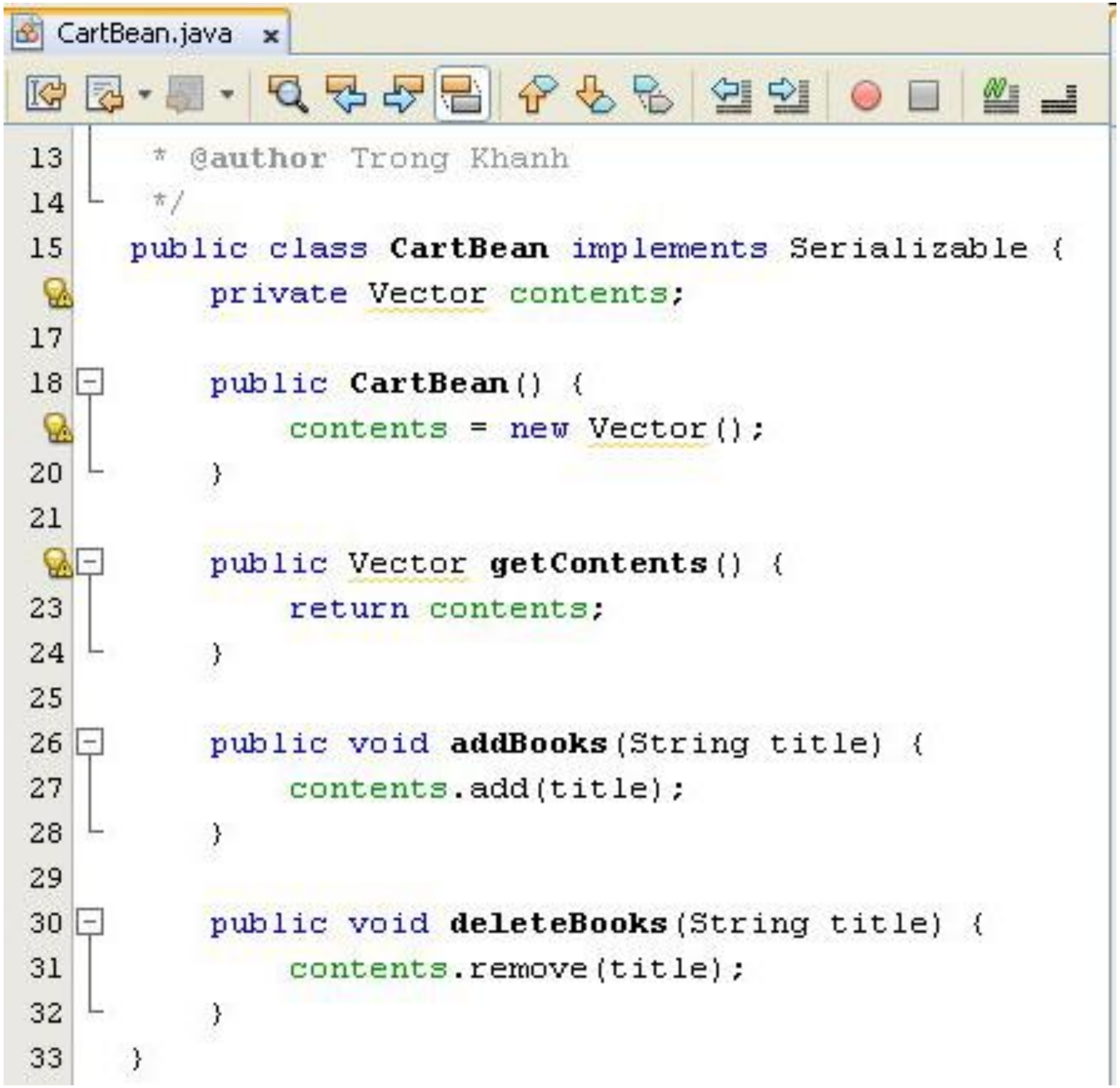
Sessions & Listeners

HttpSession interface – Example



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Shopping</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
      <h1>HttpSession Demo on Shopping Cart</h1>
      <form action="CartServlet">
          Please, choose your favourite books:<br/>
          <select name="cboBooks">
              <option>Servlet</option>
              <option>JSP</option>
              <option>EJB</option>
              <option>Netbeans</option>
              <option>Tomcat</option>
              <option>JBoss</option>
              <option>Java</option>
              <option>Others</option>
          </select><br/>
          <input type="submit" value="Add to Cart" name="btAction" />
          <input type="submit" value="View Cart" name="btAction" />
      </form>
  </body>
</html>
```

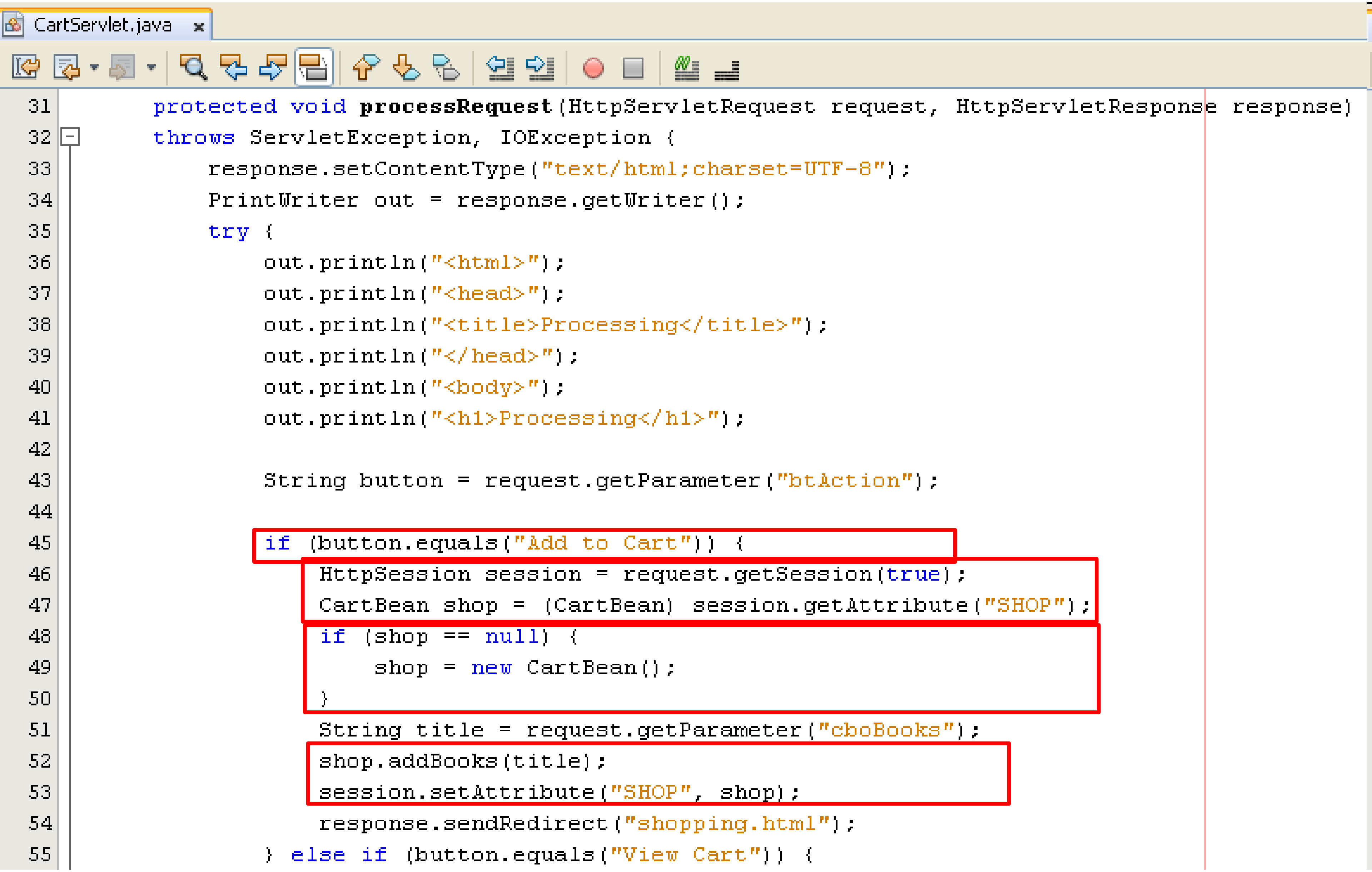
Sessions & Listeners
HttpSession interface – Example



```
 * @author Trong Khanh
 */
public class CartBean implements Serializable {
    private Vector contents;

    public CartBean() {
        contents = new Vector();
    }

    public Vector getContents() {
        return contents;
    }

    public void addBooks(String title) {
        contents.add(title);
    }

    public void deleteBooks(String title) {
        contents.remove(title);
    }
}
```

Sessions & Listeners

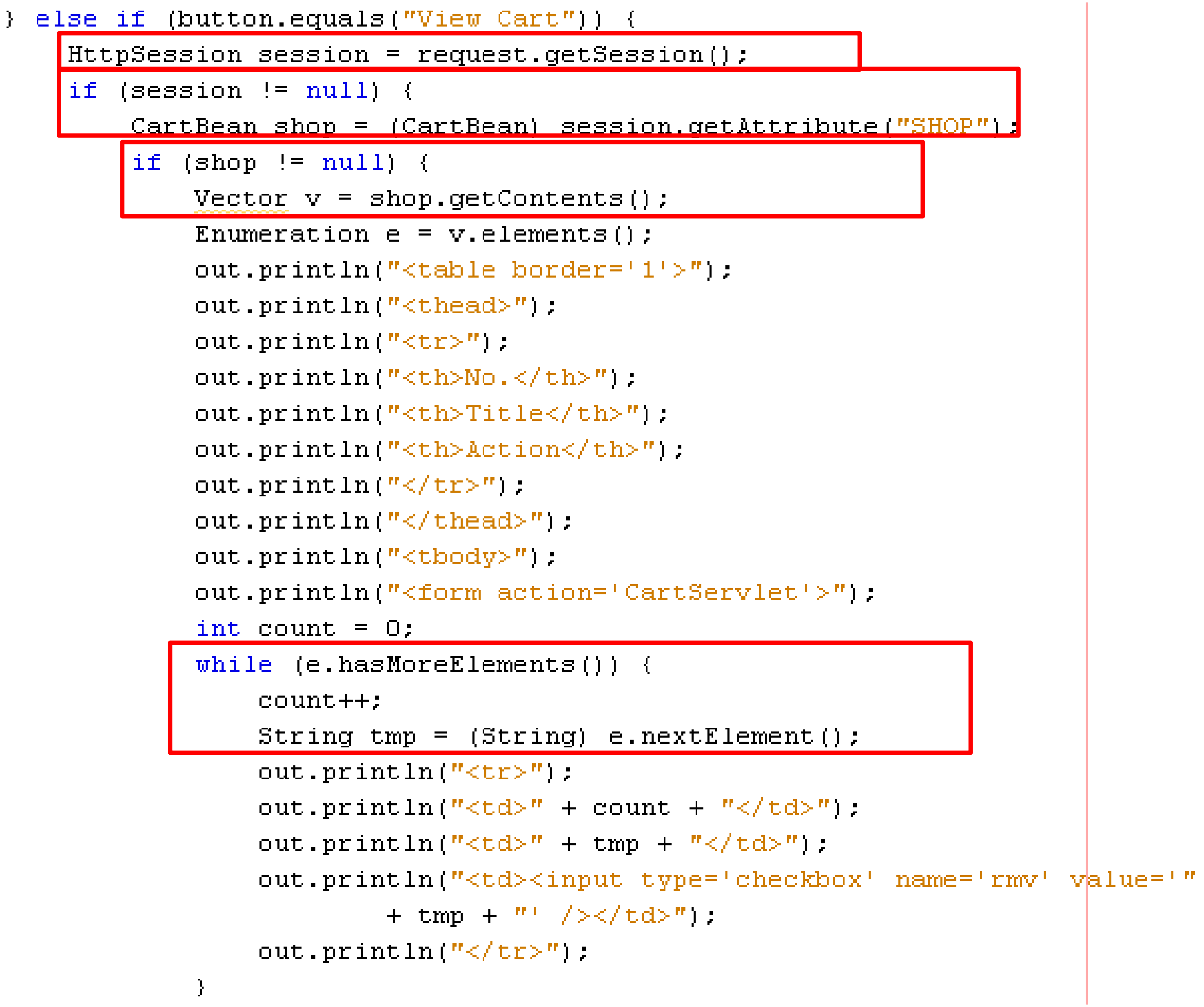
HttpSession interface – Example



```
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Processing</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Processing</h1>");

            String button = request.getParameter("btAction");

            if (button.equals("Add to Cart")) {
                HttpSession session = request.getSession(true);
                CartBean shop = (CartBean) session.getAttribute("SHOP");
                if (shop == null) {
                    shop = new CartBean();
                }
                String title = request.getParameter("cboBooks");
                shop.addBooks(title);
                session.setAttribute("SHOP", shop);
                response.sendRedirect("shopping.html");
            } else if (button.equals("View Cart")) {
```

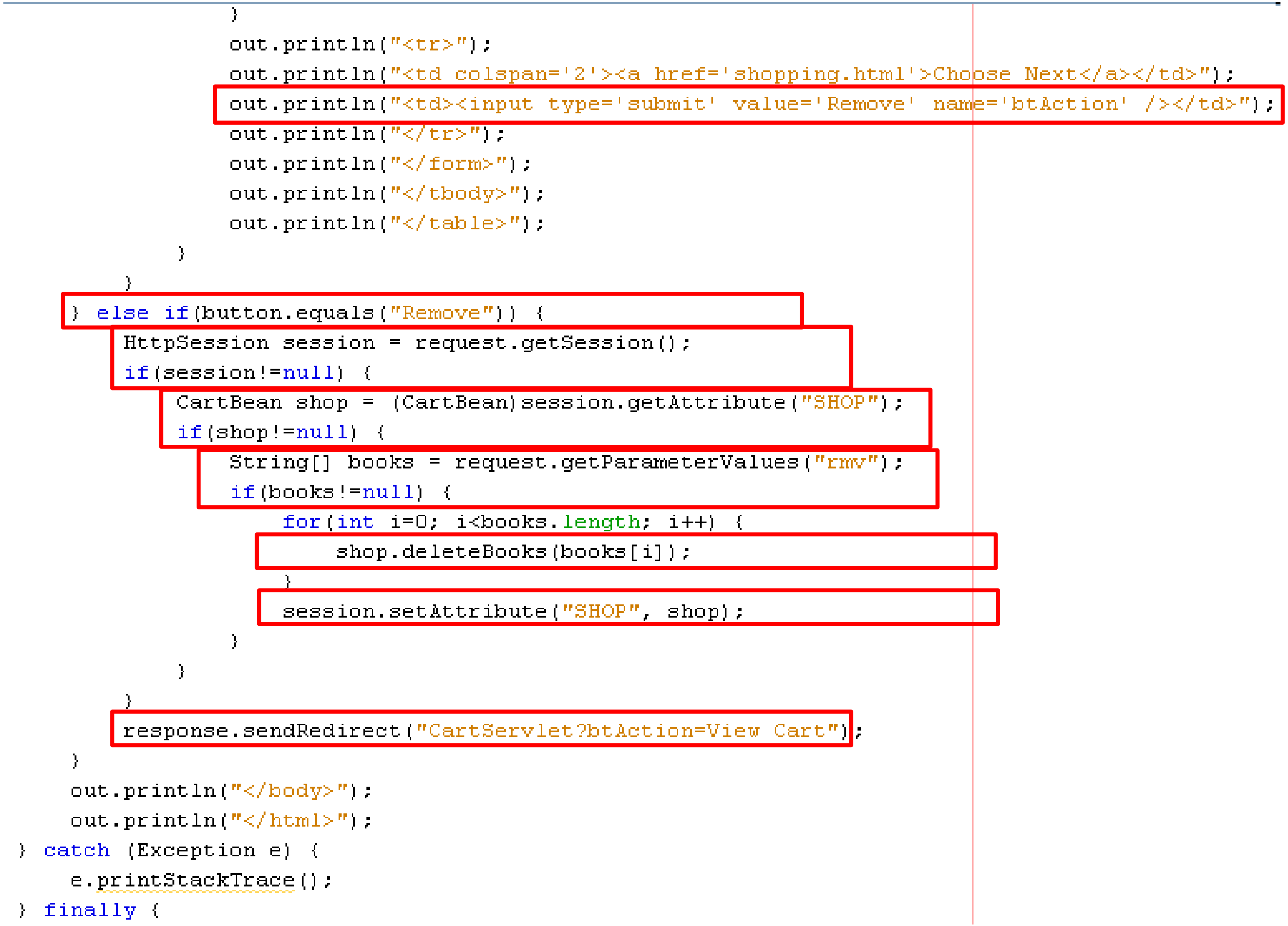
Sessions & Listeners

HttpSession interface – Example



```
} else if (button.equals("View Cart")) {
    HttpSession session = request.getSession();
    if (session != null) {
        CartBean shop = (CartBean) session.getAttribute("SHOP");
        if (shop != null) {
            Vector v = shop.getContents();
            Enumeration e = v.elements();
            out.println("<table border='1'>");
            out.println("<thead>");
            out.println("<tr>");
            out.println("<th>No.</th>");
            out.println("<th>Title</th>");
            out.println("<th>Action</th>");
            out.println("</tr>");
            out.println("</thead>");
            out.println("<tbody>");
            out.println("<form action='CartServlet'>");
            int count = 0;
            while (e.hasMoreElements()) {
                count++;
                String tmp = (String) e.nextElement();
                out.println("<tr>");
                out.println("<td>" + count + "</td>");
                out.println("<td>" + tmp + "</td>");
                out.println("<td><input type='checkbox' name='rmv' value='"
                        + tmp + "' /></td>");
                out.println("</tr>");
            }
```

Sessions & Listeners
HttpSession interface – Example

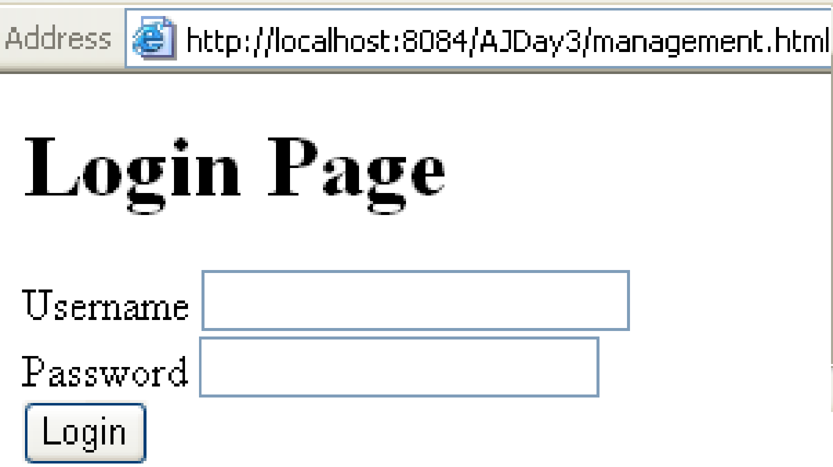


```
                }
                out.println("<tr>");
                out.println("<td colspan='2'><a href='shopping.html'>Choose Next</a></td>");
                out.println("<td><input type='submit' value='Remove' name='btAction' /></td>");
                out.println("</tr>");
                out.println("</form>");
                out.println("</tbody>");
                out.println("</table>");
            }
        }
    } else if(button.equals("Remove")) {
        HttpSession session = request.getSession();
        if(session!=null) {
            CartBean shop = (CartBean)session.getAttribute("SHOP");
            if(shop!=null) {
                String[] books = request.getParameterValues("rmv");
                if(books!=null) {
                    for(int i=0; i<books.length; i++) {
                        shop.deleteBooks(books[i]);
                    }
                    session.setAttribute("SHOP", shop);
                }
            }
        }
        response.sendRedirect("CartServlet?btAction=View Cart");
    }
    out.println("</body>");
    out.println("</html>");
} catch (Exception e) {
    e.printStackTrace();
} finally {
```

Session Management: General Principles

- Each of these requests **needs to carry a unique ID**, which identifies the session to which it belongs.
- The web application will **allocate this unique ID** on the first request from the client.
- The ID **must be passed back to the client** so that the client **can pass it back again with its next request**. In this way, the web application will know to which session the request belongs. This implies that the client **must need to store the unique ID somewhere—and** that's where session management mechanisms come in
- The **default mechanism for session management is cookie**

Session Management – Example



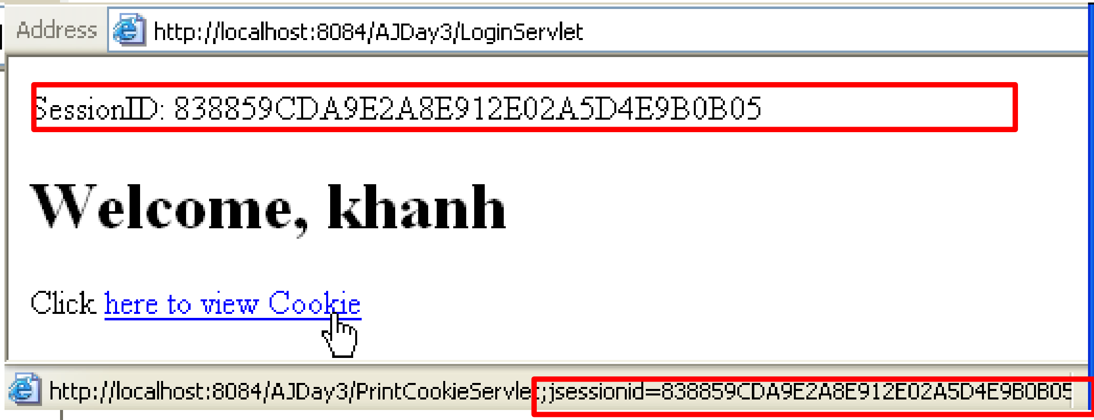
Address http://localhost:8084/AJDay3/management.html

Login Page

Username

Password

Login



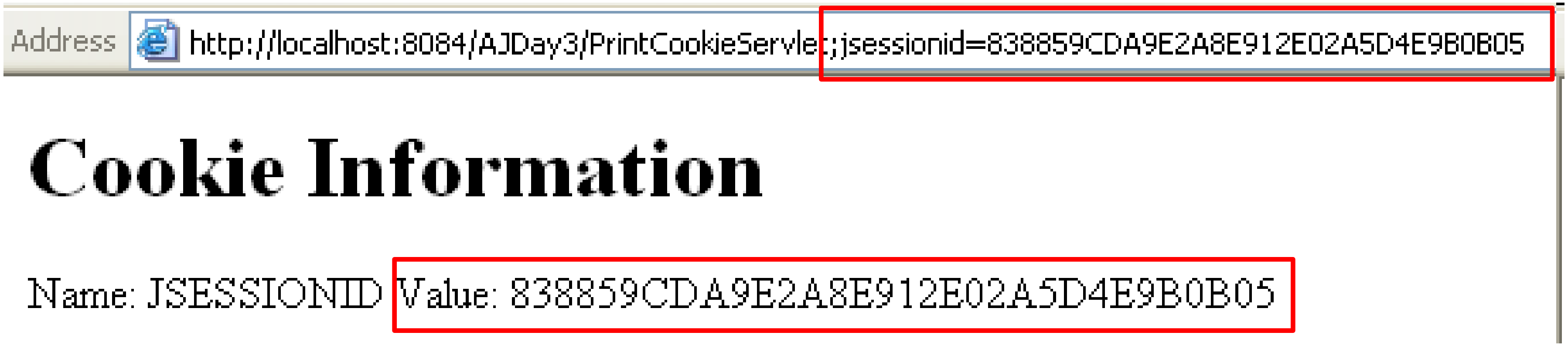
Address http://localhost:8084/AJDay3/LoginServlet

SessionID: 838859CDA9E2A8E912E02A5D4E9B0B05

Welcome, khanh

Click here to view Cookie

http://localhost:8084/AJDay3/PrintCookieServlet;jsessionid=838859CDA9E2A8E912E02A5D4E9B0B05

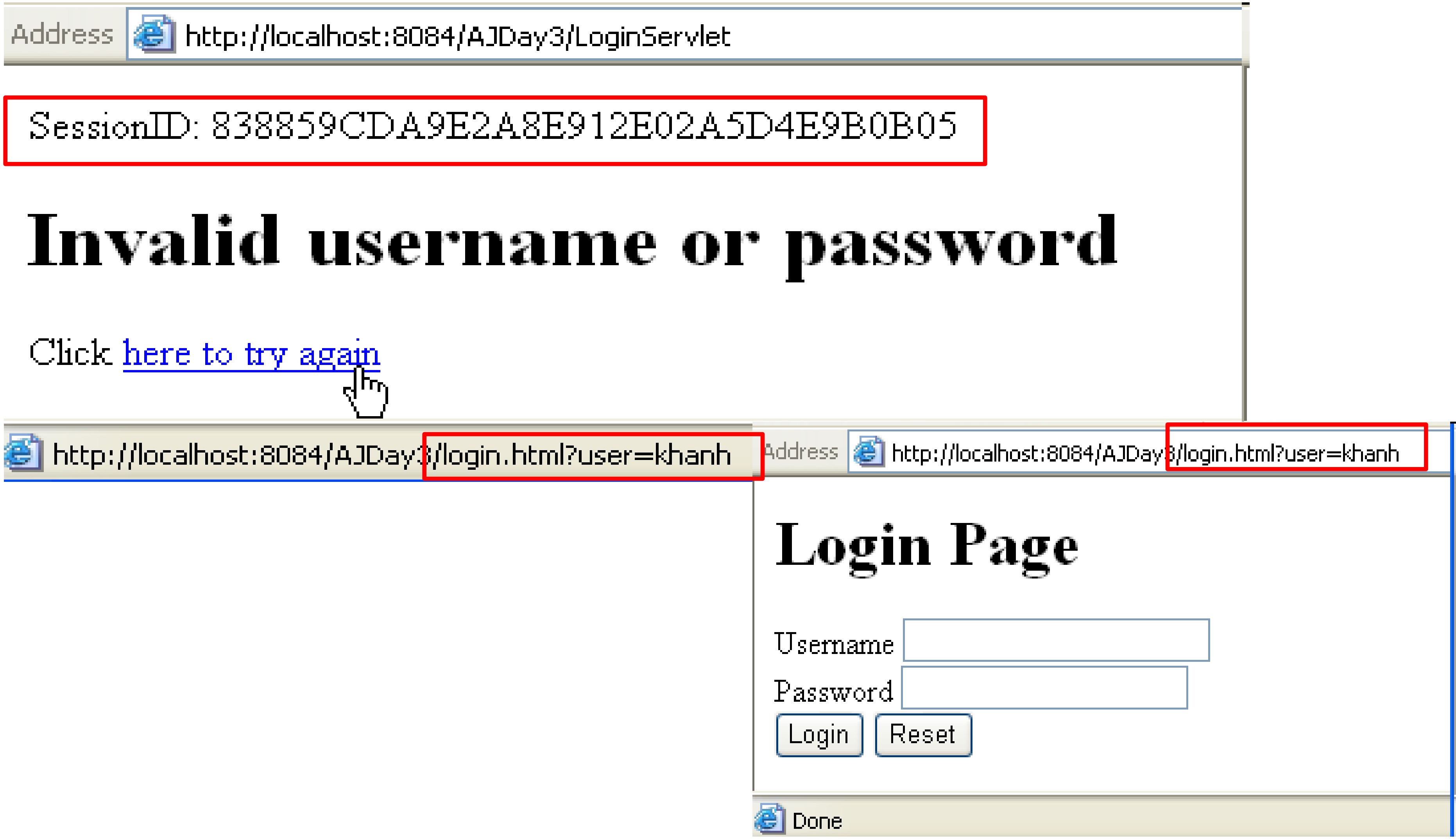


Address http://localhost:8084/AJDay3/PrintCookieServlet;jsessionid=838859CDA9E2A8E912E02A5D4E9B0B05

Cookie Information

Name: JSESSIONID Value: 838859CDA9E2A8E912E02A5D4E9B0B05

Session Management – Example



Address http://localhost:8084/AJDay3/LoginServlet

SessionID: 838859CDA9E2A8E912E02A5D4E9B0B05

Invalid username or password

Click here to try again

http://localhost:8084/AJDay3/login.html?user=khanh

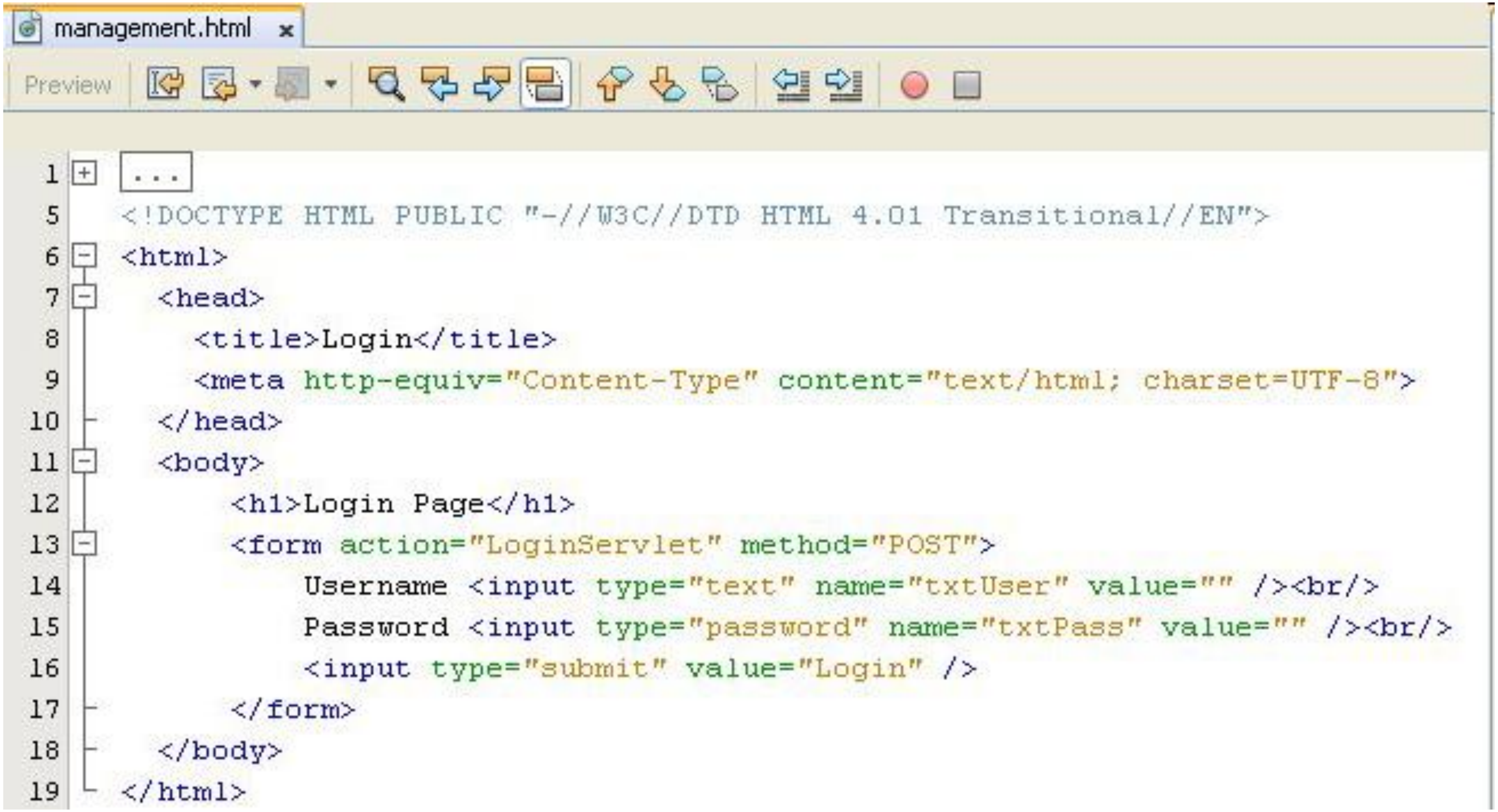
Address http://localhost:8084/AJDay3/login.html?user=khanh

Login Page

Username

Password

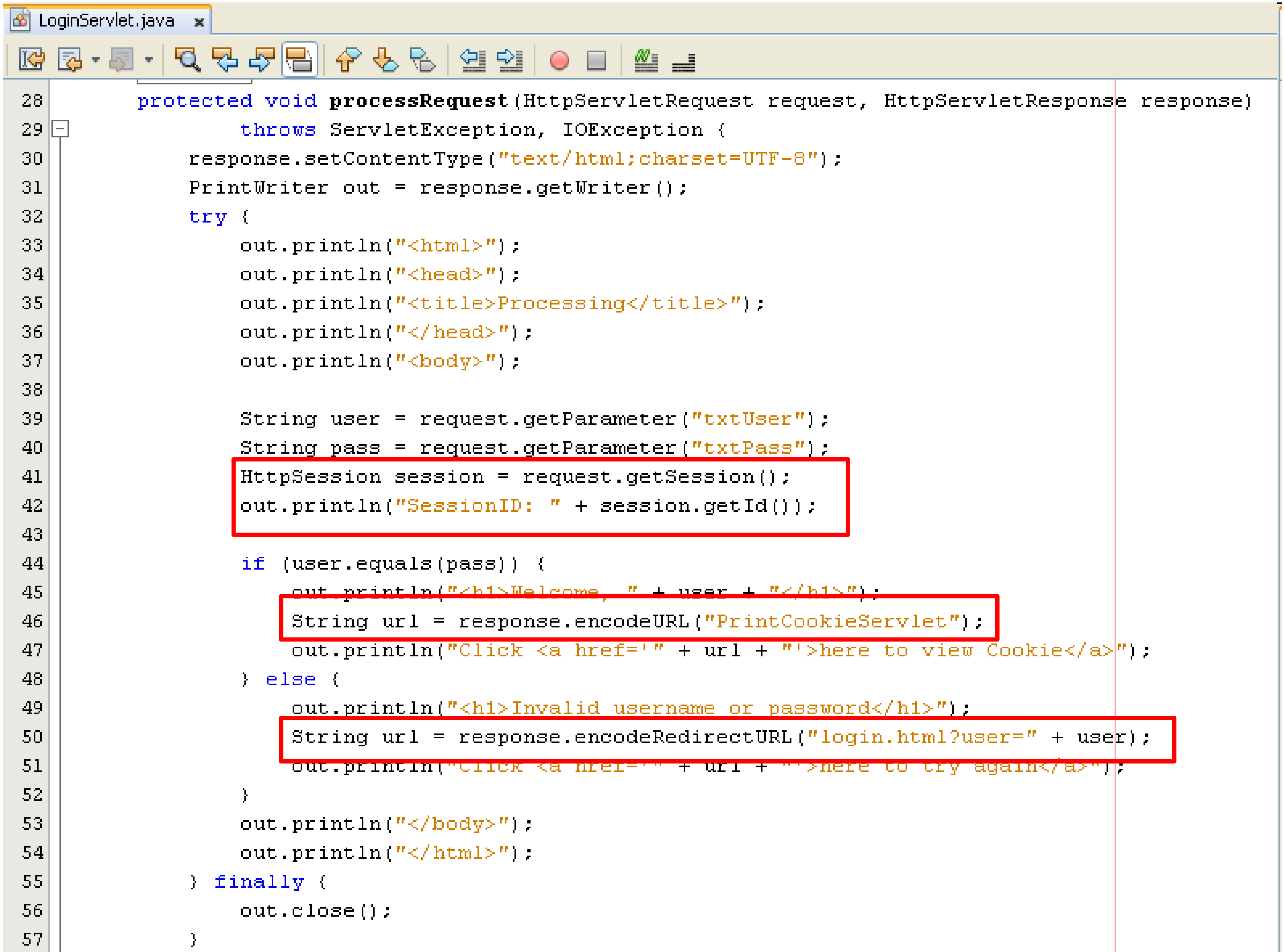
Session Management – Example



```
...
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Login</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
      <h1>Login Page</h1>
      <form action="LoginServlet" method="POST">
          Username <input type="text" name="txtUser" value="" /><br/>
          Password <input type="password" name="txtPass" value="" /><br/>
          <input type="submit" value="Login" />
      </form>
  </body>
</html>
```

Sessions & Listeners

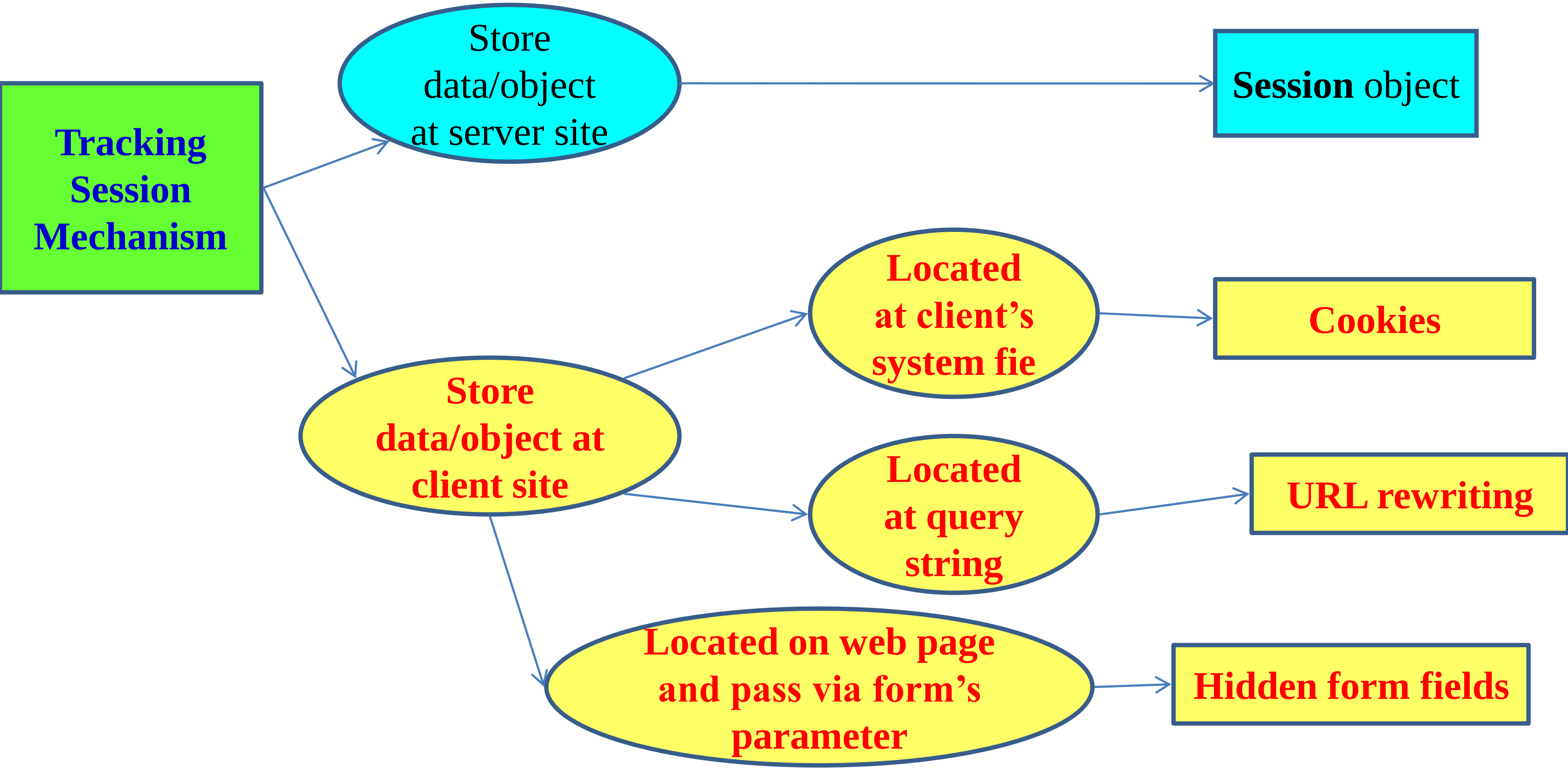
Session Management – Example



```
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
      response.setContentType("text/html;charset=UTF-8");
      PrintWriter out = response.getWriter();
      try {
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Processing</title>");
        out.println("</head>");
        out.println("<body>");

        String user = request.getParameter("txtUser");
        String pass = request.getParameter("txtPass");
        HttpSession session = request.getSession();
        out.println("SessionID: " + session.getId());

        if (user.equals(pass)) {
          out.println("<h1>Welcome, " + user + "</h1>");
          String url = response.encodeURL("PrintCookieServlet");
          out.println("Click <a href='" + url + "'>here to view Cookie</a>");
        } else {
          out.println("<h1>Invalid username or password</h1>");
          String url = response.encodeRedirectURL("login.html?user=" + user);
          out.println("Click <a href='" + url + "'>here to try again</a>");
        }
        out.println("</body>");
        out.println("</html>");
      } finally {
        out.close();
      }
```

Sessions & Listeners

Conclusion



Tracking Session Mechanism

Store data/object at server site

Session object

Store data/object at client site

Located at client's system fie

Cookies

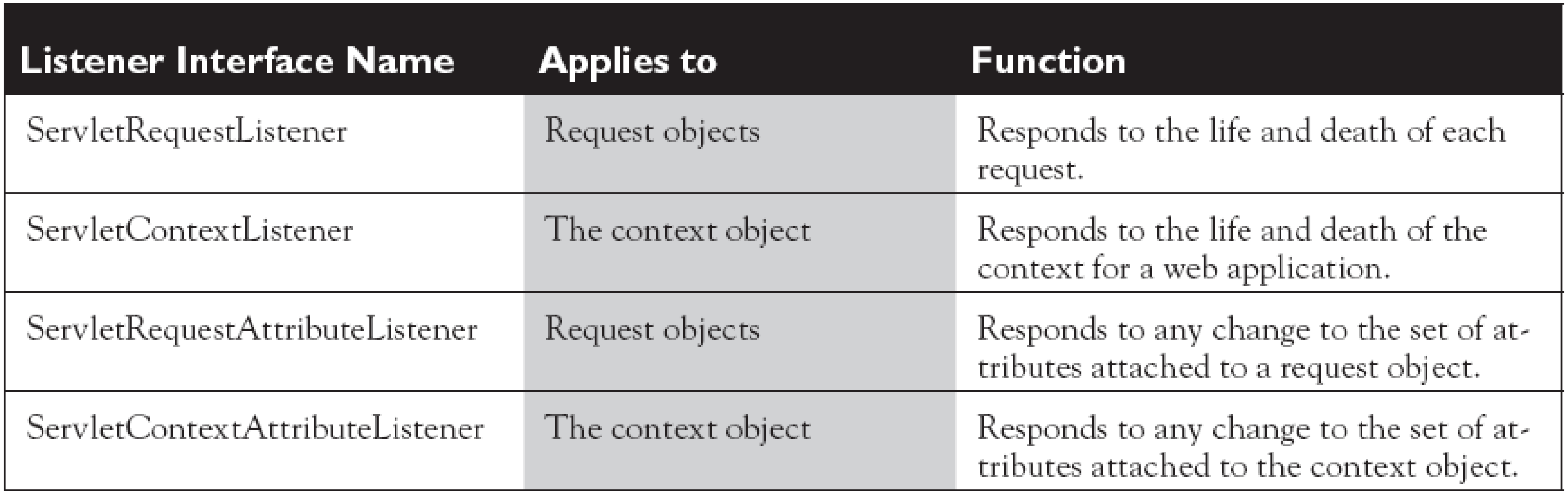
Located at query string

URL rewriting

Located on web page and pass via form's parameter

Hidden form fields

Request and Context Listeners



Listener Interface Name	Applies to	Function
ServletRequestListener	Request objects	Responds to the life and death of each request.
ServletContextListener	The context object	Responds to the life and death of the context for a web application.
ServletRequestAttributeListener	Request objects	Responds to any change to the set of attributes attached to a request object.
ServletContextAttributeListener	The context object	Responds to any change to the set of attributes attached to the context object.

- There are **two things** that need to do **to set up a listener** in a web application:
  - **Write a class** that **implements** the **appropriate listener interface**.
  - **Register** the **class name in** the **web** application **deployment descriptor**, web.xml.

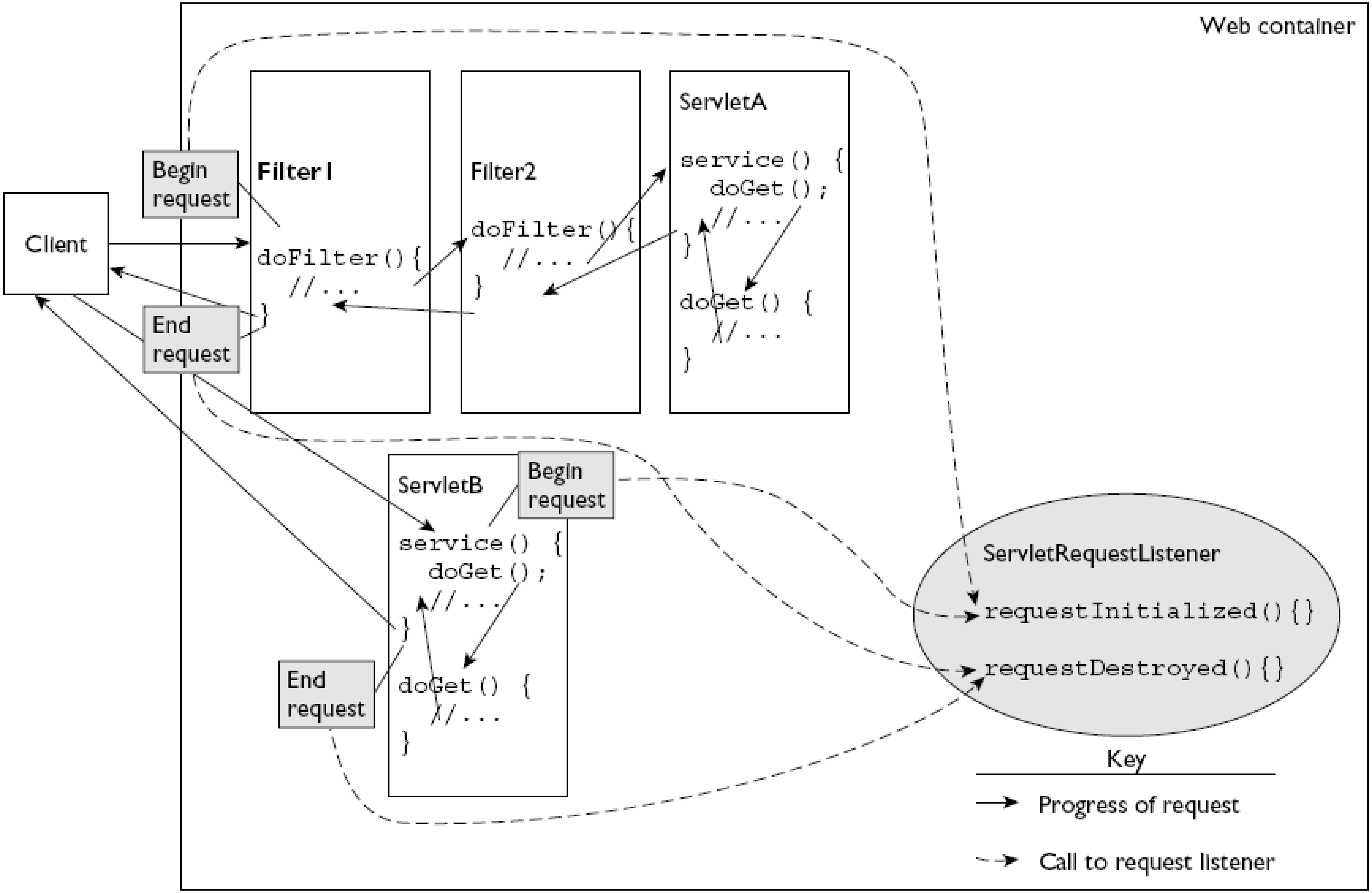
```
<listener>
    <listener-class>className</listener-class>
</listener>
```

Request Listeners

- ServletRequestListener **deals** with the **life cycle of each request object**
- A class **implementing** the **ServletRequestListener** interface has 2 methods
  - **requestInitialized**(): is **called** the **moment** that **any request** in the web container be **comes newly available** (or it is called at the **beginning of any request's scope)**
    - This is at the beginning of a servlet's service() method or earlier than that if filter chain is involved
  - **requestDestroyed**(): is called for each request that **comes to an end** – either at the **end of the servlet's service() method** or at the **end of the doFilter()** method for the first filter in a chain
- **Each** of these ServletRequestListener methods **accept** a **ServletRequestEvent** as a parameter. This event object has 2 methods
  - getServletContext()
  - getServletRequest()

Sessions & Listeners

Request Listeners



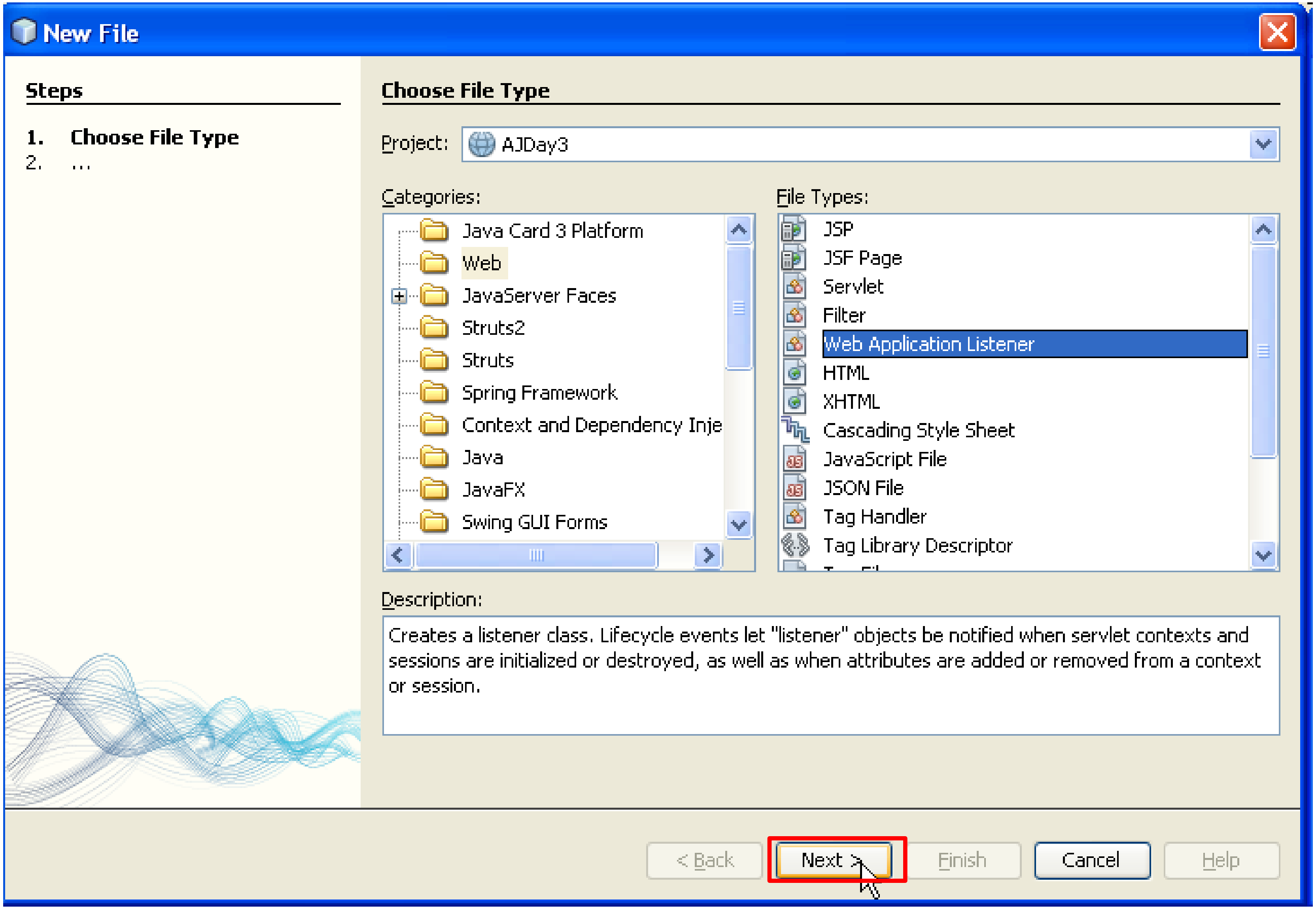
Sessions & Listeners

Request Attribute Listeners

- **ServletRequestAttributeListener deals** with the **life cycle of the attributes attached** to **request objects**
- A **class implementing** the **ServletRequestAttributeListener** interface **has 3 methods**
  - **attributeAdded**(): is called whenever a new attribute is added to any request
  - **attributeRemoved**(): is called whenever an attribute is removed from a request
  - **attributeReplaced**(): is called whenever an attribute is replaced
- Each of these ServletRequestAttributeListener methods **accept** a **ServletRequestAttributeEvent** as a parameter. This event object has **2 methods**
  - **getName**(): returns name of attribute
  - **getValue**(): returns old value of attribute
- The ServletRequestAttributeEvent inherits from ServetRequestEvent
- The "grandparent" of The ServletRequestEvent is java.util.EventObject
  - The getSource() method returns the object that is the source of the event

Sessions & Listeners

How to Add Listener to Web Project



New File

Steps

1. Choose File Type
2. ...

Choose File Type

Project: AJDay3

Categories:

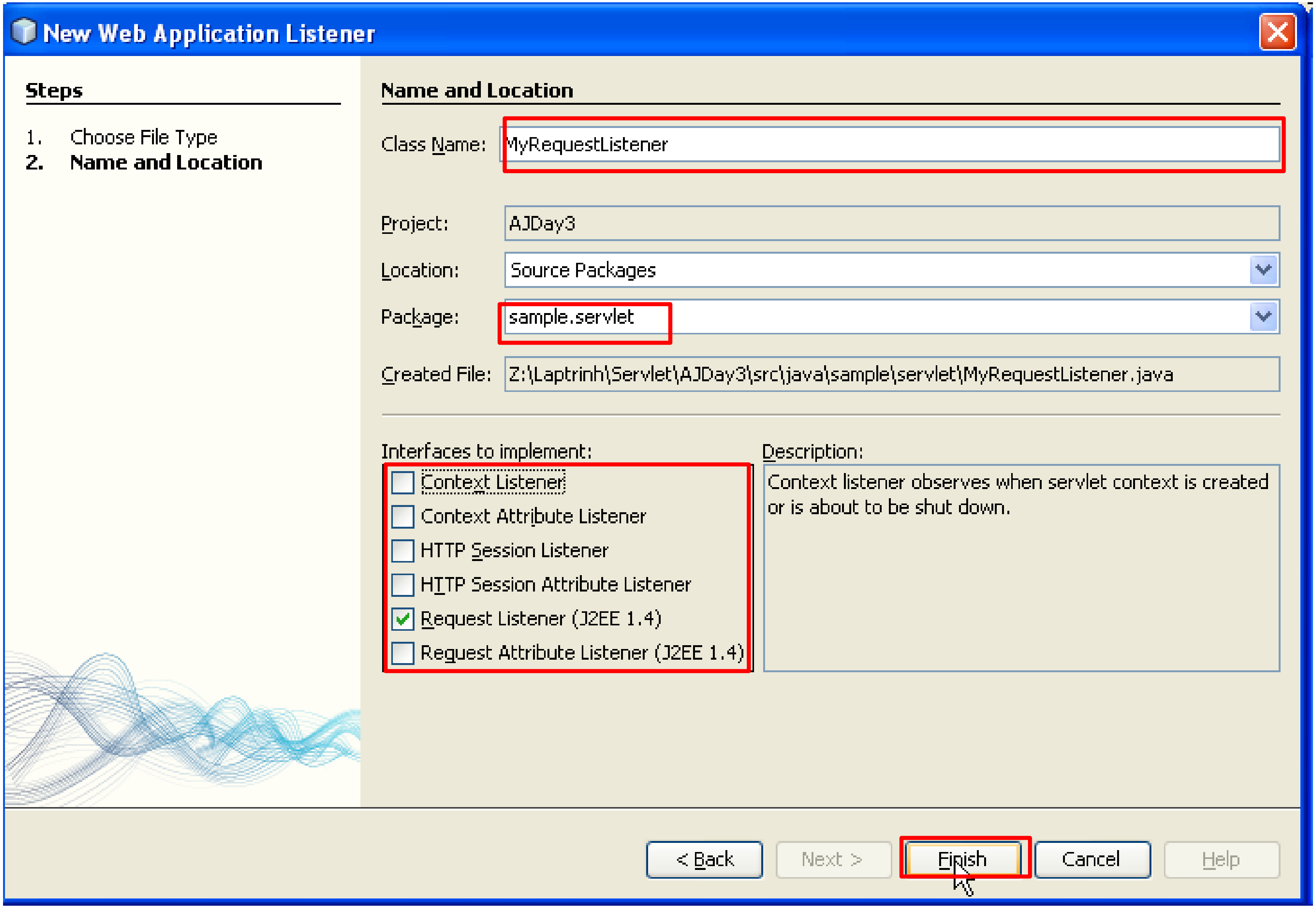
File Types:

Description:

Creates a listener class. Lifecycle events let "listener" objects be notified when servlet contexts and sessions are initialized or destroyed, as well as when attributes are added or removed from a context or session.

Sessions & Listeners

How to Add Listener to Web Project



New Web Application Listener

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name: MyRequestListener

Project: AJDay3

Location: Source Packages

Package: sample.servlet

Created File: Z:\Laptrinh\Servlet\AJDay3\src\java\sample\servlet\MyRequestListener.java

Interfaces to implement:

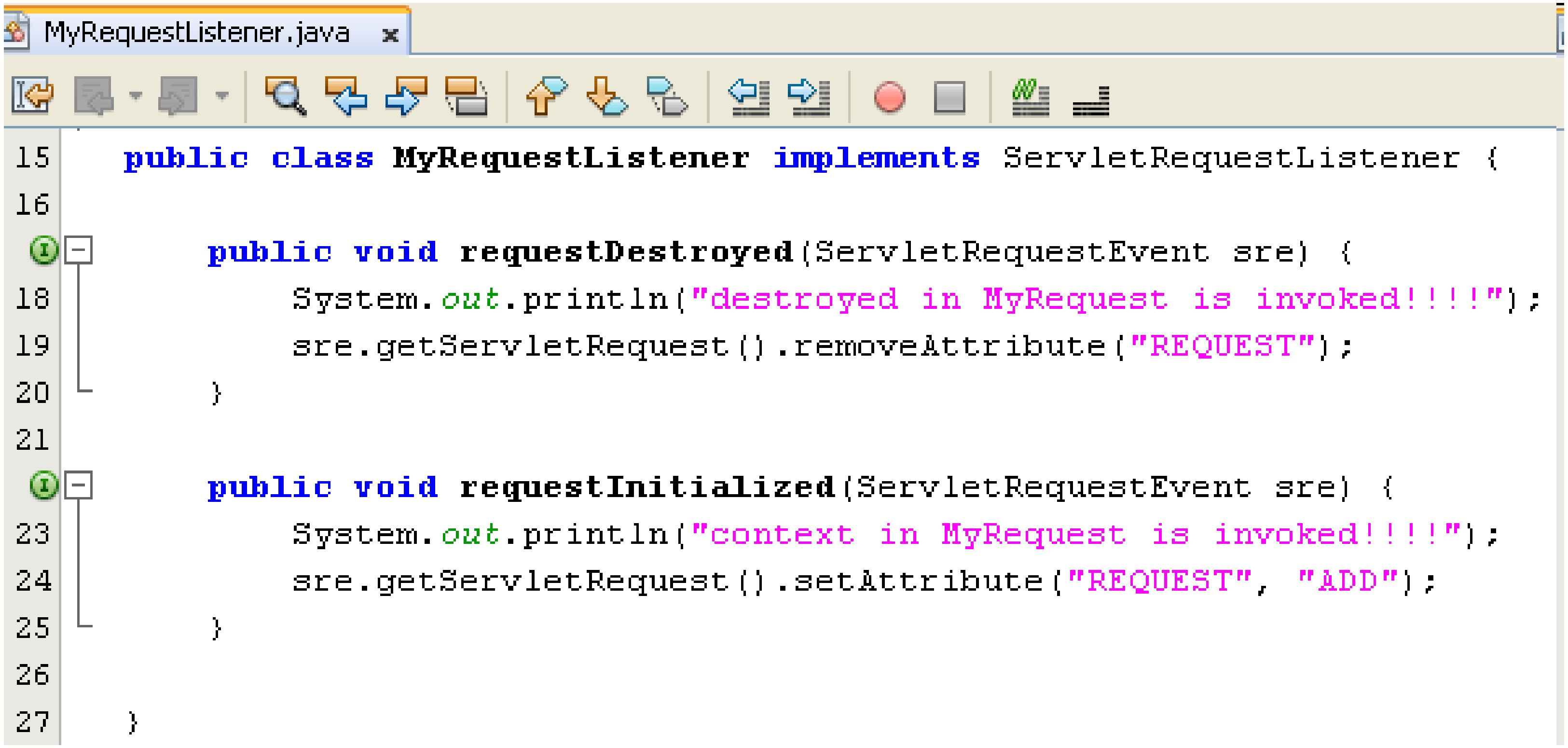
- □ Context Listener
- □ Context Attribute Listener
- □ HTTP Session Listener
- □ HTTP Session Attribute Listener
- ☑ Request Listener (J2EE 1.4)
- □ Request Attribute Listener (J2EE 1.4)

Description:

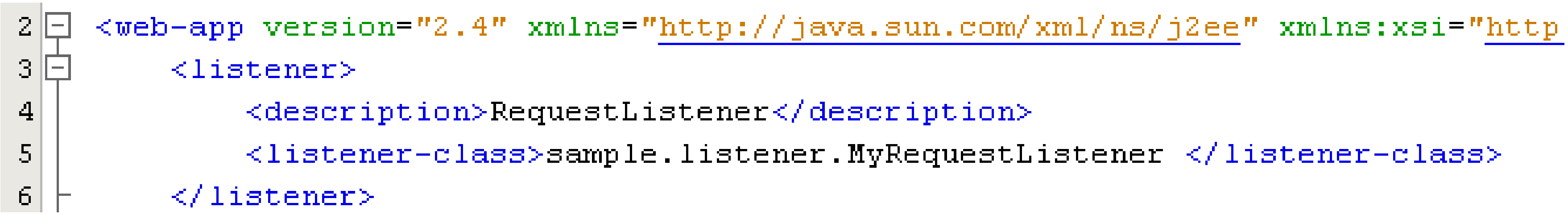
Context listener observes when servlet context is created or is about to be shut down.

Sessions & Listeners

Example



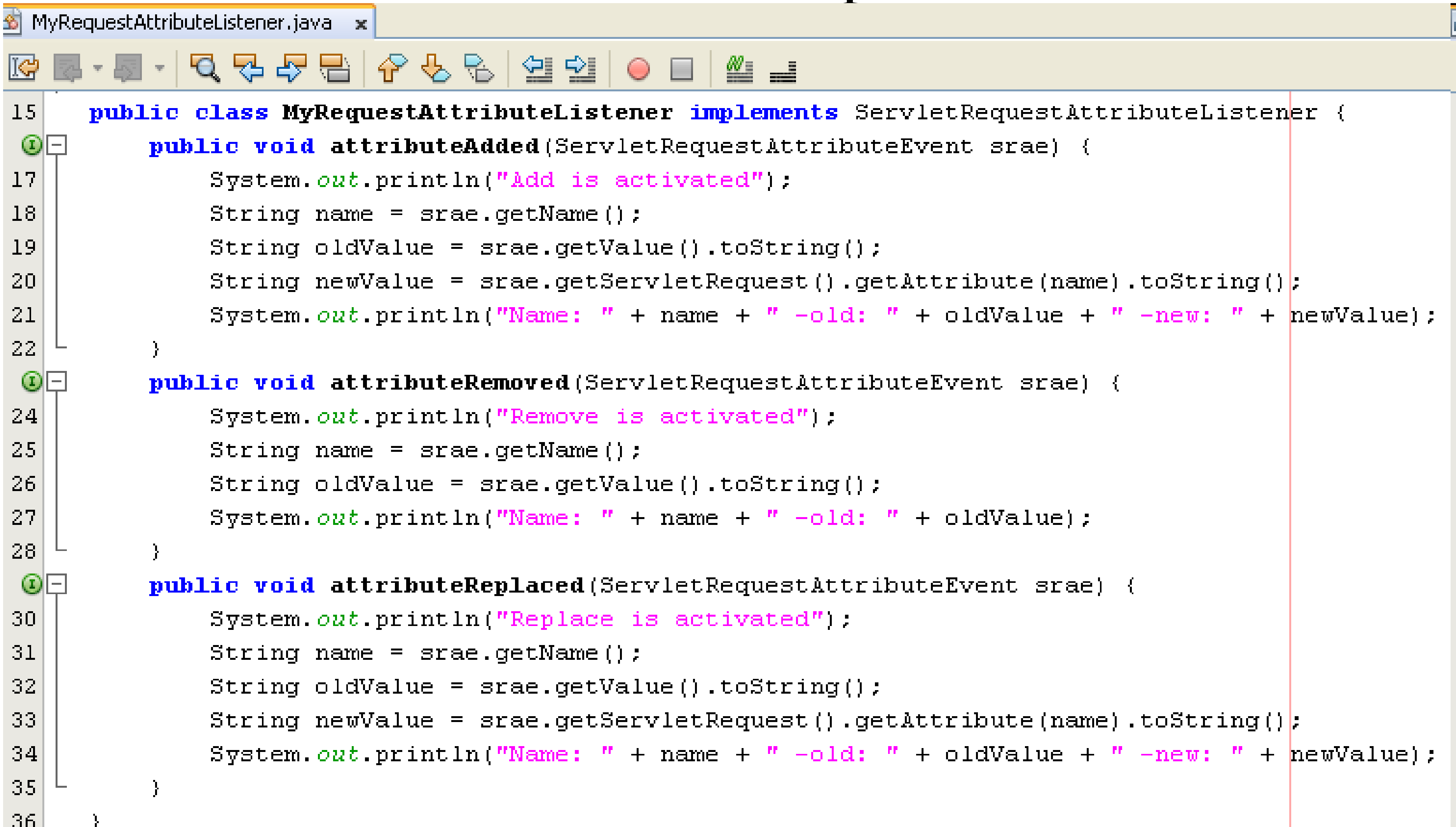
```java
public class MyRequestListener implements ServletRequestListener {

    public void requestDestroyed(ServletRequestEvent sre) {
        System.out.println("destroyed in MyRequest is invoked!!!!");
        sre.getServletRequest().removeAttribute("REQUEST");
    }

    public void requestInitialized(ServletRequestEvent sre) {
        System.out.println("context in MyRequest is invoked!!!!");
        sre.getServletRequest().setAttribute("REQUEST", "ADD");
    }

}
```



```xml
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http
    <listener>
        <description>RequestListener</description>
        <listener-class>sample.listener.MyRequestListener </listener-class>
    </listener>
```

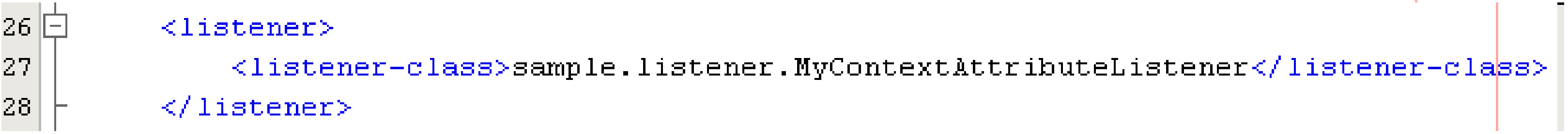
Sessions & Listeners

Example



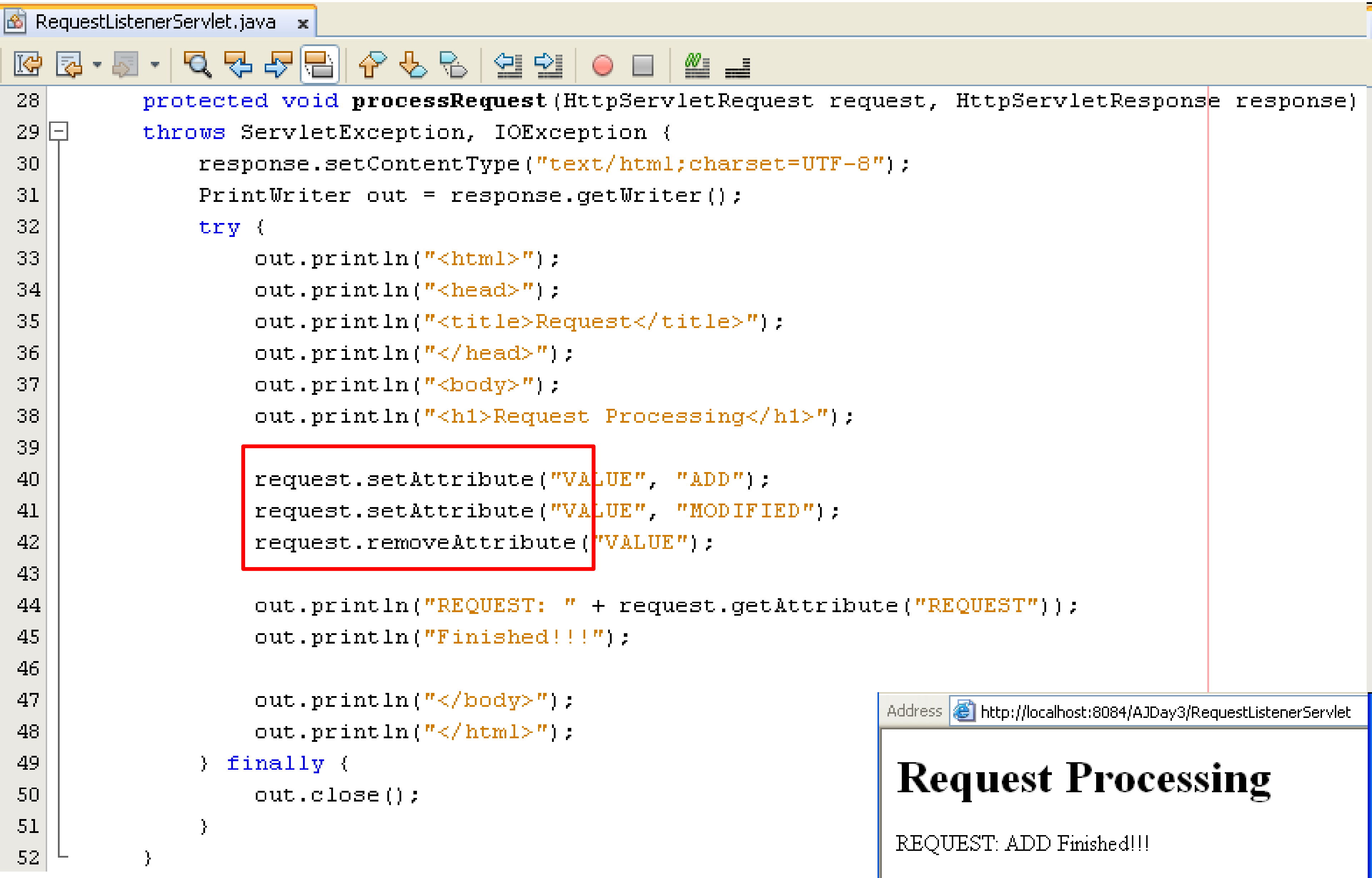
MyRequestAttributeListener.java

```java
public class MyRequestAttributeListener implements ServletRequestAttributeListener {
    public void attributeAdded(ServletRequestAttributeEvent srae) {
        System.out.println("Add is activated");
        String name = srae.getName();
        String oldValue = srae.getValue().toString();
        String newValue = srae.getServletRequest().getAttribute(name).toString();
        System.out.println("Name: " + name + " -old: " + oldValue + " -new: " + newValue);
    }
    public void attributeRemoved(ServletRequestAttributeEvent srae) {
        System.out.println("Remove is activated");
        String name = srae.getName();
        String oldValue = srae.getValue().toString();
        System.out.println("Name: " + name + " -old: " + oldValue);
    }
    public void attributeReplaced(ServletRequestAttributeEvent srae) {
        System.out.println("Replace is activated");
        String name = srae.getName();
        String oldValue = srae.getValue().toString();
        String newValue = srae.getServletRequest().getAttribute(name).toString();
        System.out.println("Name: " + name + " -old: " + oldValue + " -new: " + newValue);
    }
}
```



```xml
<listener>
    <listener-class>sample.listener.MyContextAttributeListener</listener-class>
</listener>
```

Example



RequestListenerServlet.java

```
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Request</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Request Processing</h1>");

            request.setAttribute("VALUE", "ADD");
            request.setAttribute("VALUE", "MODIFIED");
            request.removeAttribute("VALUE");

            out.println("REQUEST: " + request.getAttribute("REQUEST"));
            out.println("Finished!!!");

            out.println("</body>");
            out.println("</html>");
        } finally {
            out.close();
        }
    }
```

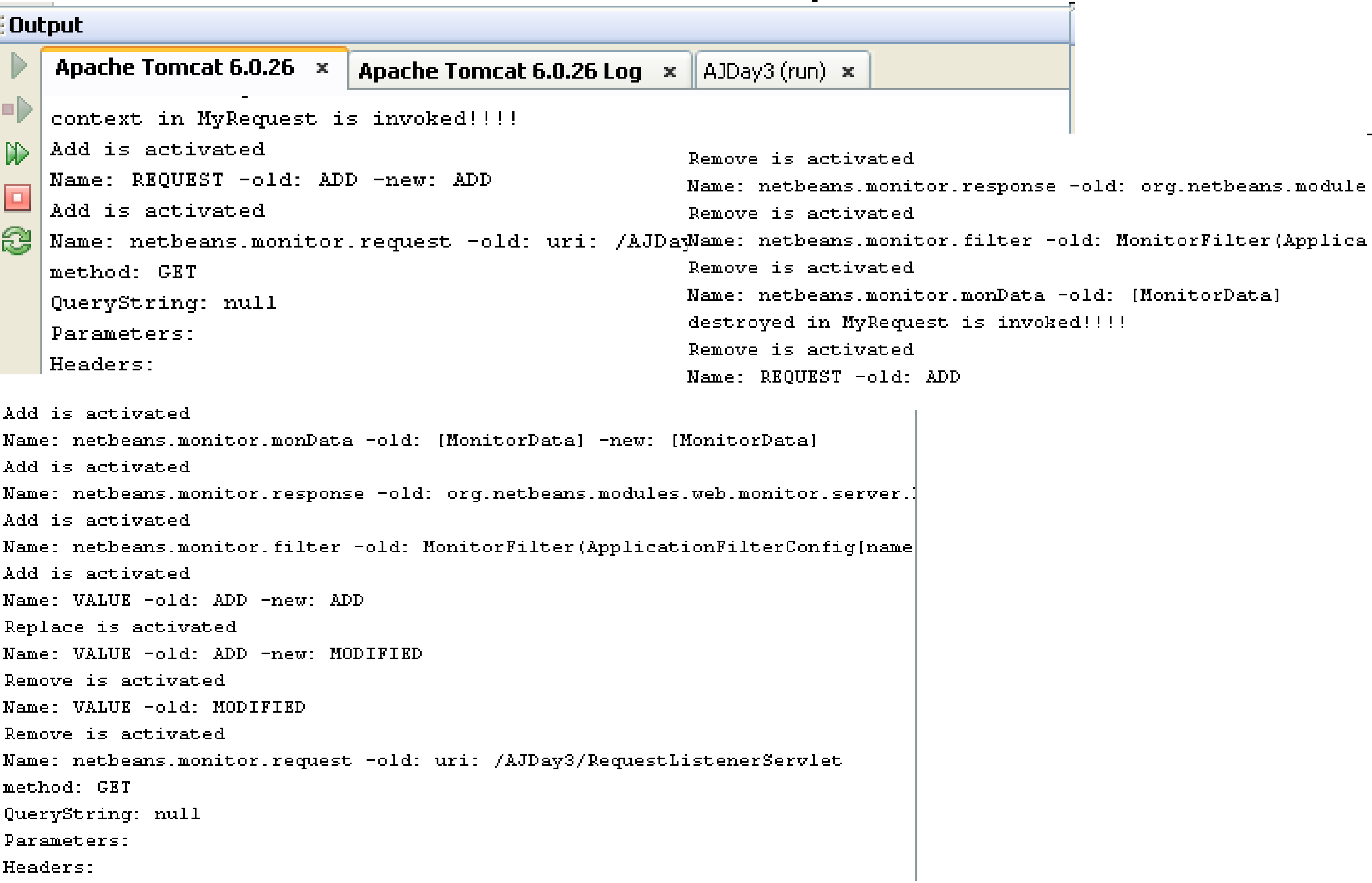
Address http://localhost:8084/AJDay3/RequestListenerServlet

Request Processing

REQUEST: ADD Finished!!!

Sessions & Listeners

Example



```
context in MyRequest is invoked!!!!
Add is activated
Name: REQUEST -old: ADD -new: ADD
Add is activated
Name: netbeans.monitor.request -old: uri: /AJDa
method: GET
QueryString: null
Parameters:
Headers:
```

```
Remove is activated
Name: netbeans.monitor.response -old: org.netbeans.module
Remove is activated
Name: netbeans.monitor.filter -old: MonitorFilter(Applica
Remove is activated
Name: netbeans.monitor.monData -old: [MonitorData]
destroyed in MyRequest is invoked!!!!
Remove is activated
Name: REQUEST -old: ADD
```

```
Add is activated
Name: netbeans.monitor.monData -old: [MonitorData] -new: [MonitorData]
Add is activated
Name: netbeans.monitor.response -old: org.netbeans.modules.web.monitor.server.
Add is activated
Name: netbeans.monitor.filter -old: MonitorFilter(ApplicationFilterConfig[name
Add is activated
Name: VALUE -old: ADD -new: ADD
Replace is activated
Name: VALUE -old: ADD -new: MODIFIED
Remove is activated
Name: VALUE -old: MODIFIED
Remove is activated
Name: netbeans.monitor.request -old: uri: /AJDay3/RequestListenerServlet
method: GET
QueryString: null
Parameters:
Headers:
```

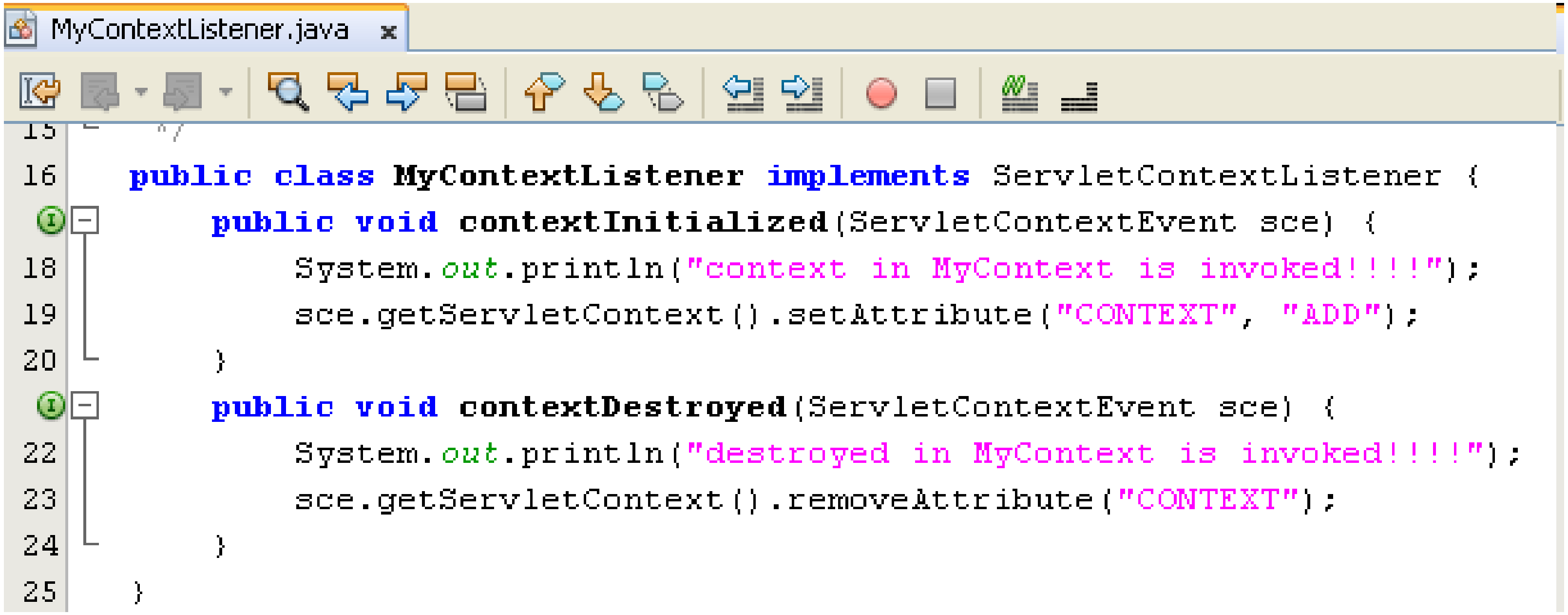
Sessions & Listeners

Context Listener

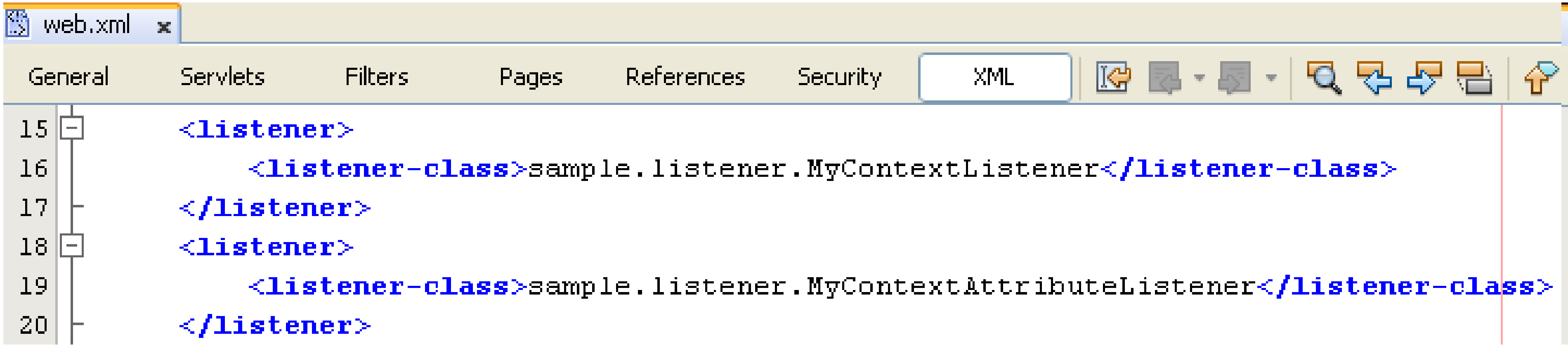
- Sessions have 02 listeners:
  - **ServletContextListener**
    - **Receive notifications** about **changes** to the **servlet context** of the Web application
      - **contextInitialized**(): gets **called before** any servlet's **init() method** or any filter's **doFilter() method**
      - **contextDestroyed**(): gets called **after** the **servlet's or filter's destroy()** method
      - Both of methods get passed a ServletContextEvent object that provides the getServletContext() method
  - **ServletContextAttributeListener**
    - **Recieves** a **notification** about any **modifications** made to the **attribute list** on the servlet context of a web application
    - Has the same trio of methods as **ServletRequestAttributeListener**

Sessions & Listeners

Example



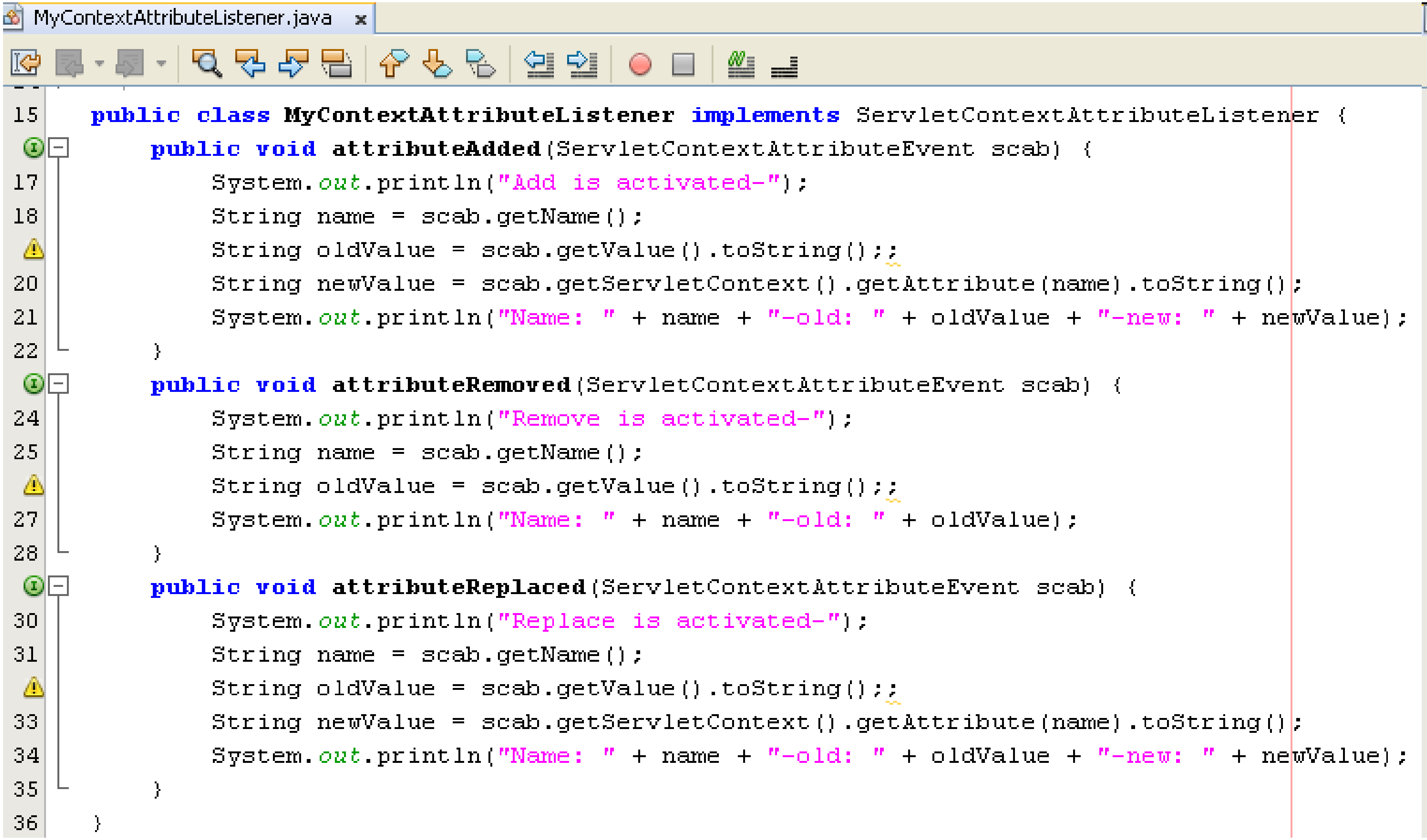
```
 */
public class MyContextListener implements ServletContextListener {
    public void contextInitialized(ServletContextEvent sce) {
        System.out.println("context in MyContext is invoked!!!!");
        sce.getServletContext().setAttribute("CONTEXT", "ADD");
    }
    public void contextDestroyed(ServletContextEvent sce) {
        System.out.println("destroyed in MyContext is invoked!!!!");
        sce.getServletContext().removeAttribute("CONTEXT");
    }
}
```



```
    <listener>
        <listener-class>sample.listener.MyContextListener</listener-class>
    </listener>
    <listener>
        <listener-class>sample.listener.MyContextAttributeListener</listener-class>
    </listener>
```

Sessions & Listeners

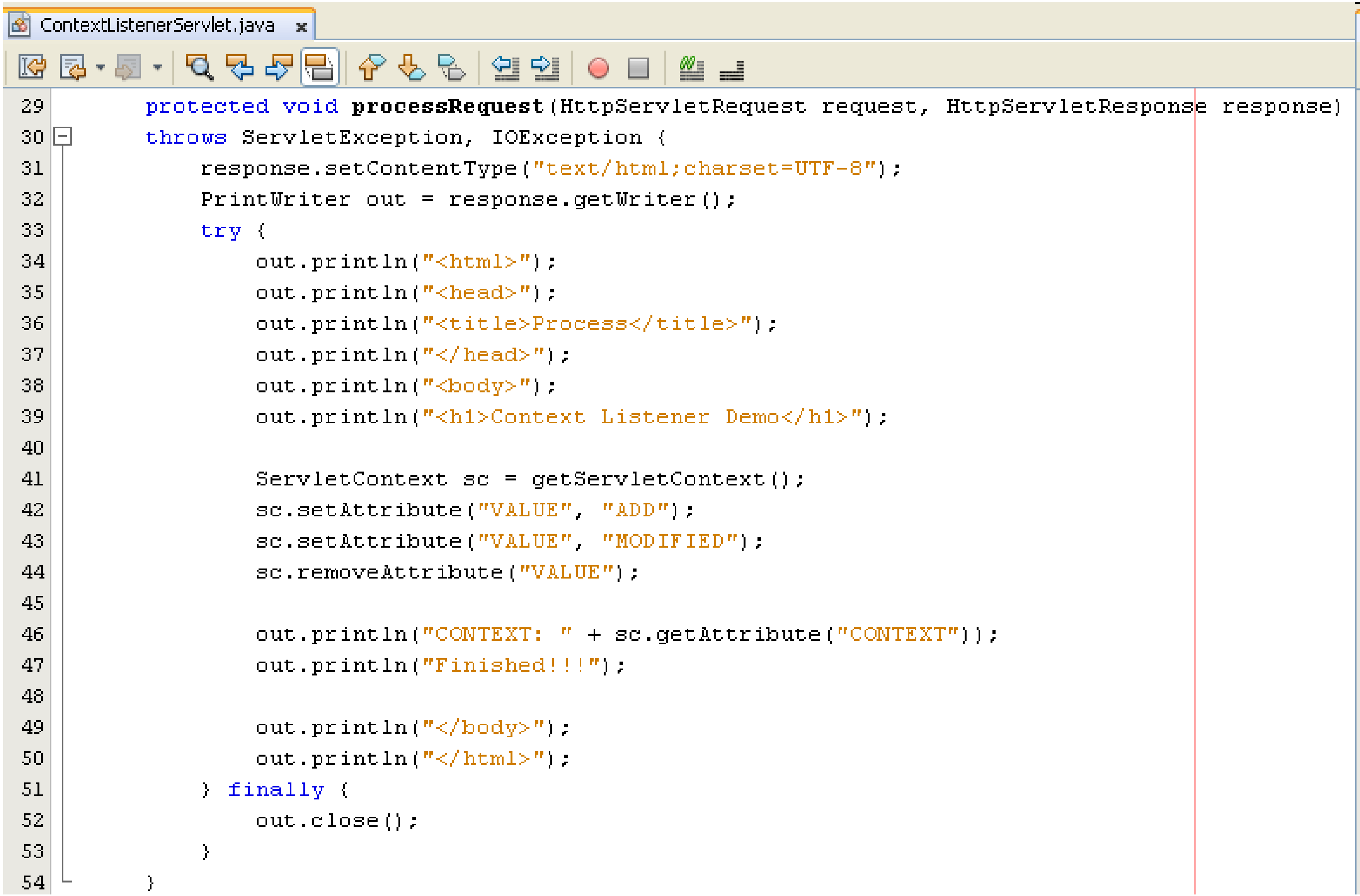
Example



```java
public class MyContextAttributeListener implements ServletContextAttributeListener {
    public void attributeAdded(ServletContextAttributeEvent scab) {
        System.out.println("Add is activated-");
        String name = scab.getName();
        String oldValue = scab.getValue().toString();;
        String newValue = scab.getServletContext().getAttribute(name).toString();
        System.out.println("Name: " + name + "-old: " + oldValue + "-new: " + newValue);
    }
    public void attributeRemoved(ServletContextAttributeEvent scab) {
        System.out.println("Remove is activated-");
        String name = scab.getName();
        String oldValue = scab.getValue().toString();;
        System.out.println("Name: " + name + "-old: " + oldValue);
    }
    public void attributeReplaced(ServletContextAttributeEvent scab) {
        System.out.println("Replace is activated-");
        String name = scab.getName();
        String oldValue = scab.getValue().toString();;
        String newValue = scab.getServletContext().getAttribute(name).toString();
        System.out.println("Name: " + name + "-old: " + oldValue + "-new: " + newValue);
    }
}
```

Sessions & Listeners

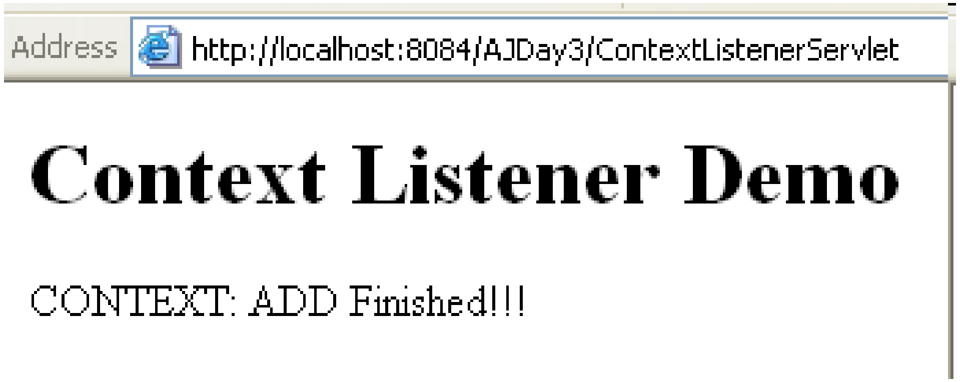
Example



```java
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Process</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Context Listener Demo</h1>");

            ServletContext sc = getServletContext();
            sc.setAttribute("VALUE", "ADD");
            sc.setAttribute("VALUE", "MODIFIED");
            sc.removeAttribute("VALUE");

            out.println("CONTEXT: " + sc.getAttribute("CONTEXT"));
            out.println("Finished!!!");

            out.println("</body>");
            out.println("</html>");
        } finally {
            out.close();
        }
    }
```

Sessions & Listeners

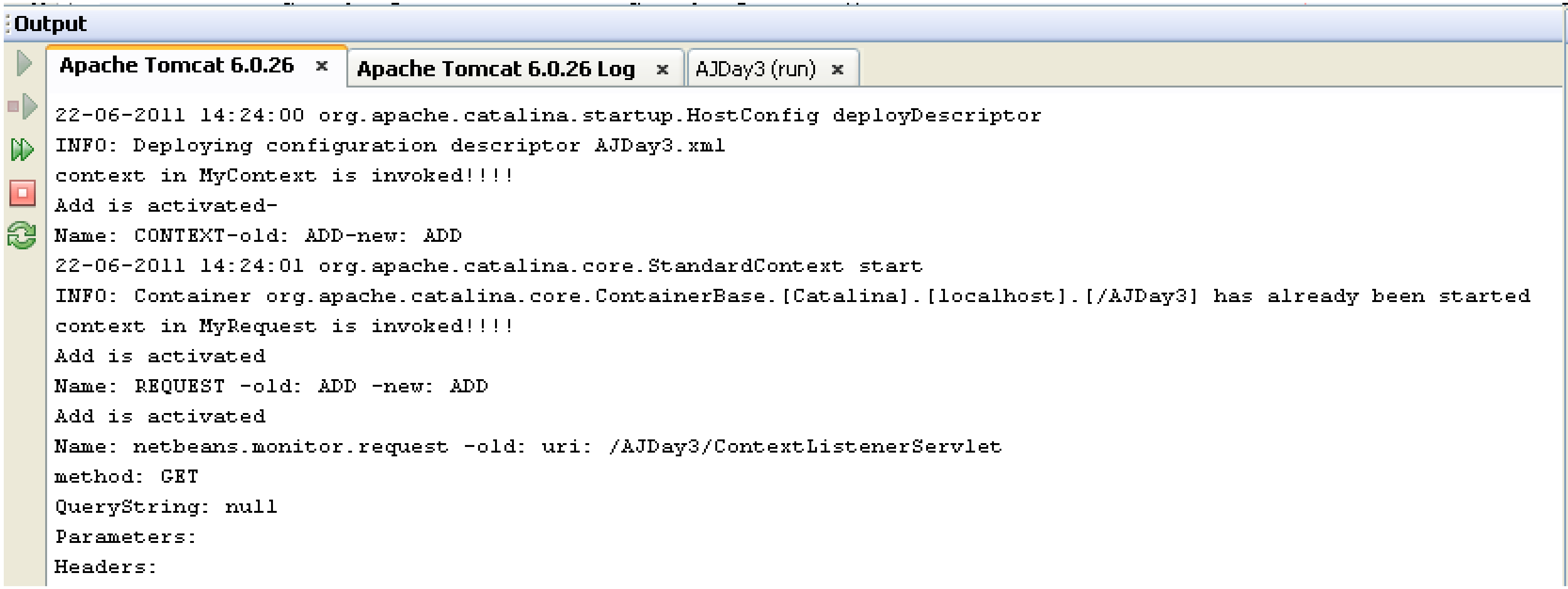
Example



http://localhost:8084/AJDay3/ContextListenerServlet

Context Listener Demo

CONTEXT: ADD Finished!!!



```
22-06-2011 14:24:00 org.apache.catalina.startup.HostConfig deployDescriptor
INFO: Deploying configuration descriptor AJDay3.xml
context in MyContext is invoked!!!!
Add is activated-
Name: CONTEXT-old: ADD-new: ADD
22-06-2011 14:24:01 org.apache.catalina.core.StandardContext start
INFO: Container org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/AJDay3] has already been started
context in MyRequest is invoked!!!!
Add is activated
Name: REQUEST -old: ADD -new: ADD
Add is activated
Name: netbeans.monitor.request -old: uri: /AJDay3/ContextListenerServlet
method: GET
QueryString: null
Parameters:
Headers:
```

Session Listeners Declared in DD

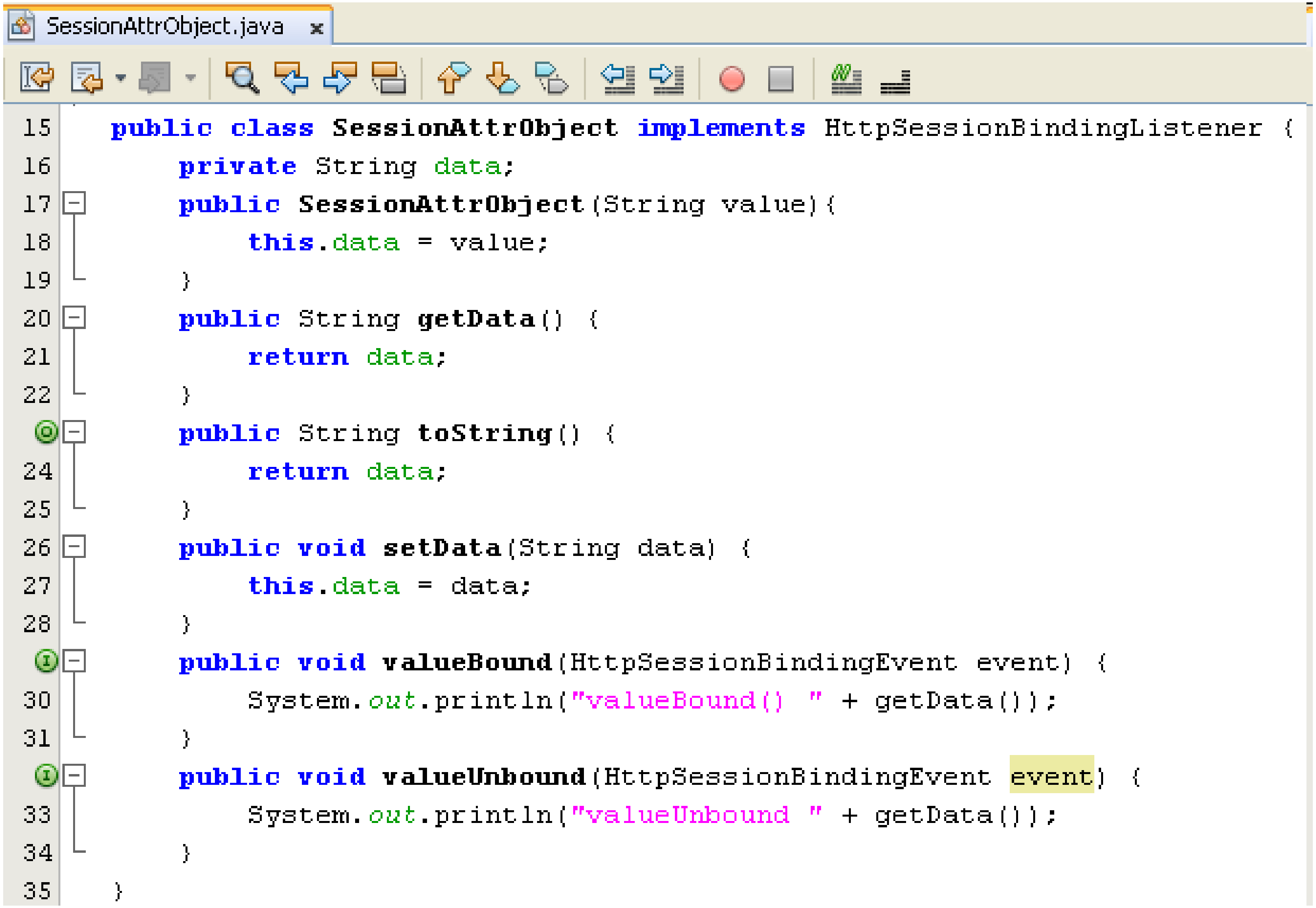
- Have **02 listeners:**
  - **HttpSessionListener**
    - Implements the changes to the list of active sessions in Web application
    - **sessionCreated**() method: is called whenever a **new session** is provided (*can say that **after** the **getSession()** method*)
    - **sessionDestroyed**(): is called at the **end of the sessions** (*within the call **invalidate()** or **session time out** but **before** the **session become invalid***)
    - Both of methods get passed a **HttpSessionEvent object** that provides the **getSession() method**
  - **HttpSessionAttributeListener**
    - Is **called** whenever **some changes** are made to the **attribute** list on the **servlet session** of a Web application
    - Is used to **notify when** an **attribute** has been **added**, **removed** or **replaced by another attribute**
    - Has the **same trio of** methods as **ServletRequestAttributeListener** that are passed the **HttpSessionBindingEvent** (is inherited from HttpSessionEvent)

Session Listeners Not Declared in DD

- Have **02 listeners:**
  - **HttpSessionBindingListener**
    - **Notifies** the **object** when it is being **bound** to **or unbound** from a **session**
    - This **notification** can be the **result** of a **forced unbinding of an attribute** from a **session by the programmer**, **invalidation of the session or due to timing out of session**
    - This **implementation** do **not require any configuration within** the deployment descriptor of the Web application
    - **Notes**: The object data types not implemented in **BindingListener don't fire any events!**

Methods	Descriptions
valueBound	- **public void valueBound(HttpSessionBindingEvent se);** - **Notifies** the object in being **bound** to a **session** and is responsible for identification of the session
valueUnbound	- **public void valueUnbound(HttpSessionBindingEvent se);** - **Notifies** the **object** on being **unbound** from a session and is responsible for identification of the session

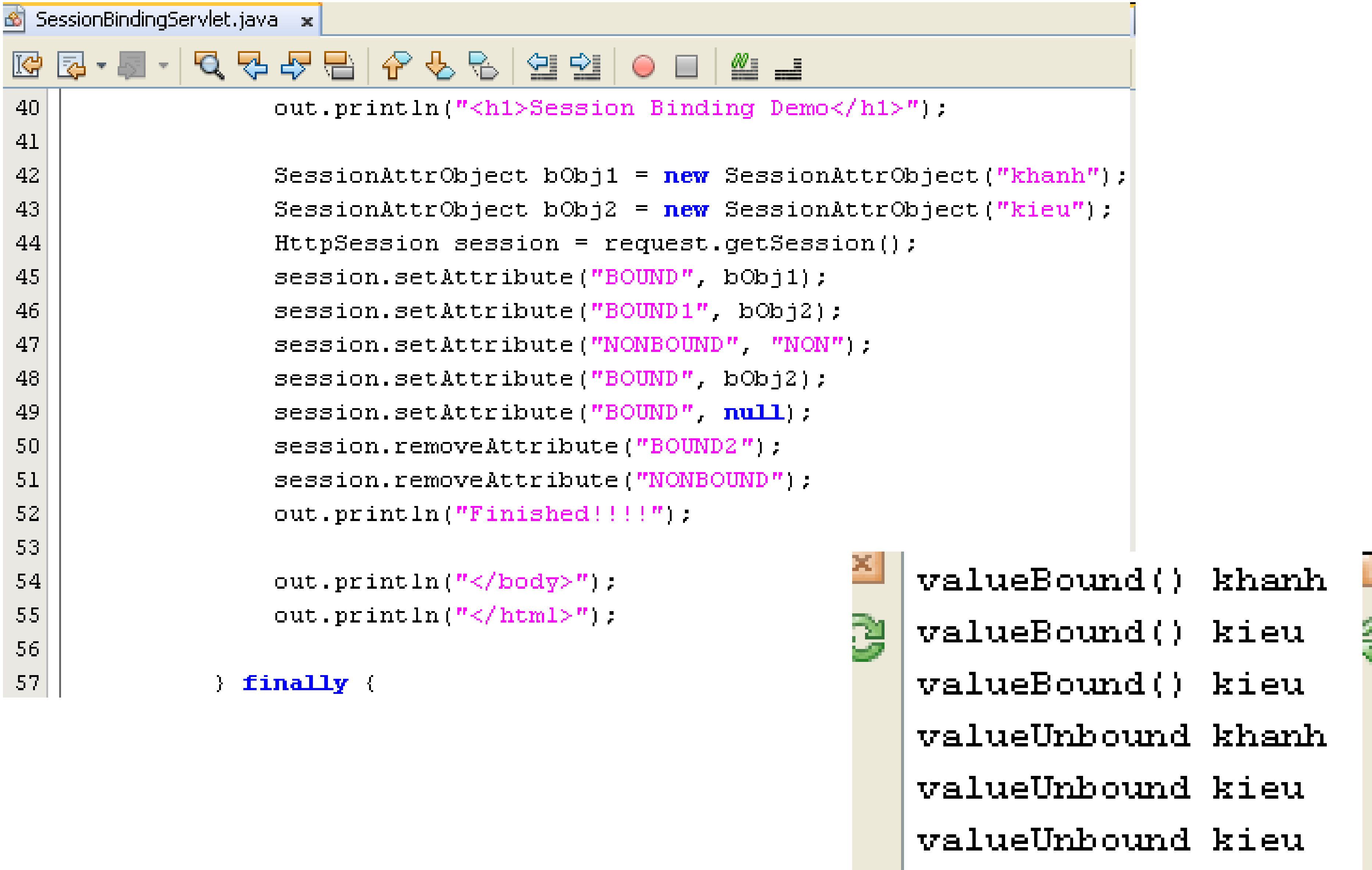
Session Listeners Not Declared in DD - Example



```
public class SessionAttrObject implements HttpSessionBindingListener {
    private String data;
    public SessionAttrObject(String value){
        this.data = value;
    }
    public String getData() {
        return data;
    }
    public String toString() {
        return data;
    }
    public void setData(String data) {
        this.data = data;
    }
    public void valueBound(HttpSessionBindingEvent event) {
        System.out.println("valueBound() " + getData());
    }
    public void valueUnbound(HttpSessionBindingEvent event) {
        System.out.println("valueUnbound " + getData());
    }
}
```

Sessions & Listeners

Session Listeners Not Declared in DD - Example



SessionBindingServlet.java

```
            out.println("<h1>Session Binding Demo</h1>");

            SessionAttrObject bObj1 = new SessionAttrObject("khanh");
            SessionAttrObject bObj2 = new SessionAttrObject("kieu");
            HttpSession session = request.getSession();
            session.setAttribute("BOUND", bObj1);
            session.setAttribute("BOUND1", bObj2);
            session.setAttribute("NONBOUND", "NON");
            session.setAttribute("BOUND", bObj2);
            session.setAttribute("BOUND", null);
            session.removeAttribute("BOUND2");
            session.removeAttribute("NONBOUND");
            out.println("Finished!!!!");

            out.println("</body>");
            out.println("</html>");

        } finally {
```

```
valueBound() khanh
valueBound() kieu
valueBound() kieu
valueUnbound khanh
valueUnbound kieu
valueUnbound kieu
```

Sessions & Listeners

Session Listeners Not Declared in DD

- Have **02 listeners** (cont)
  - **HttpSesssionActivationListener** (*receives events when a value object is transported across JVMs*).
    - **Stateful** session (activated and passivated)
    - Is **implemented** when a **container migrates** the **session between VM** or **persists sessions** and **is not required** any **configuration within** the **deployment descriptor**

Methods	Descriptions
sessionDidActivate	- **public void sessionDidActivate(HttpSessionEvent se);** - Provides notification that the session has just been activated.
sessionWillPassivate	- **public void sessionWillPassivate(HttpSessionEvent se);** - Provide notification that the session is about to be passivated.

Error Handling in Servlet

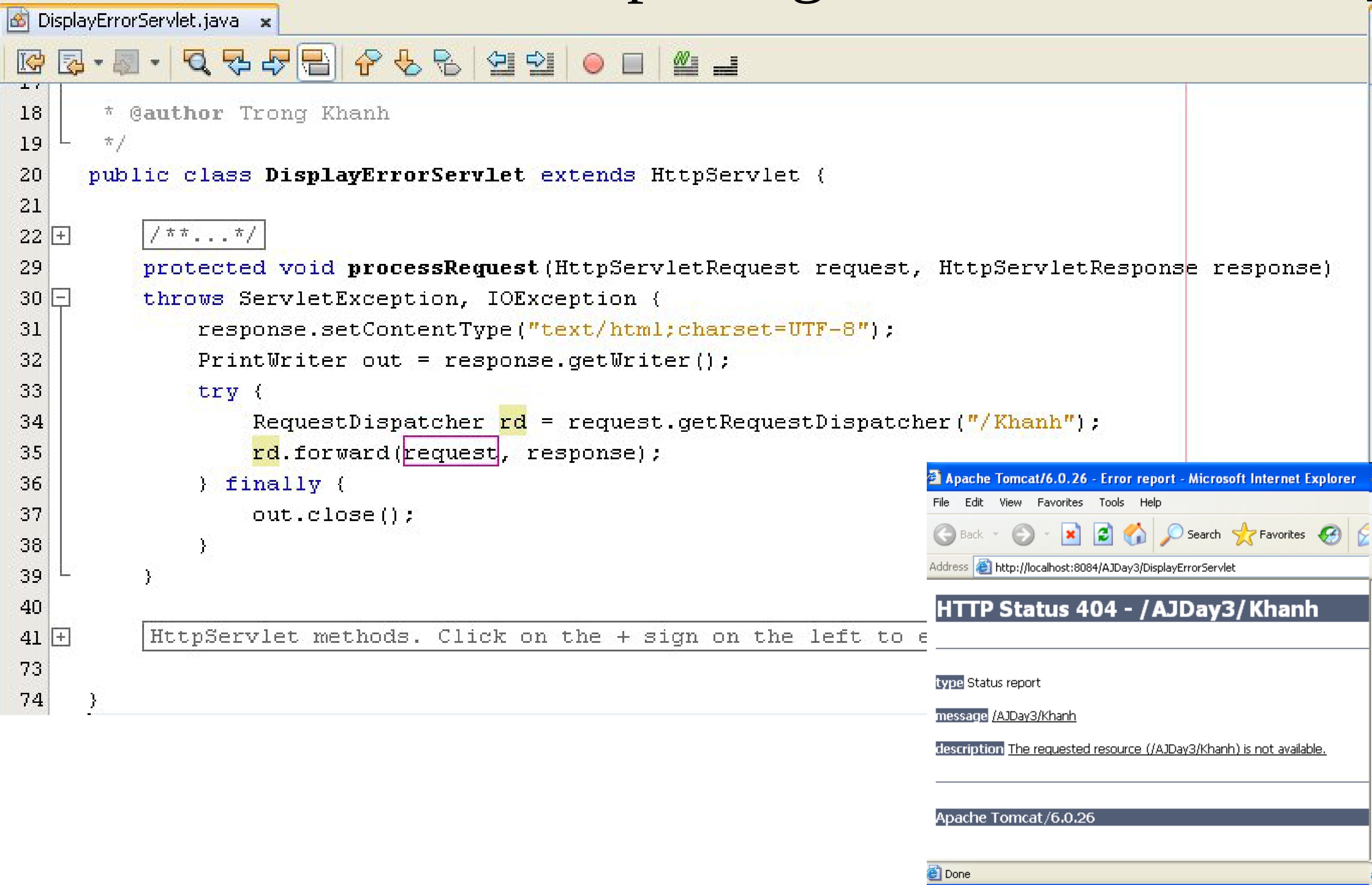
Reporting Error

- **There are many situations** occur an error
  - A requested page may be moved from one location to another.
  - The address may be wrongly typed.
  - The requested page may be forbidden, may be temporarily deleted or correct HTTP version might not have found.
  - There are other situations where an error may generated.
- Error during the execution of a web application are reported

Methods	Descriptions
sendError	- **public void sendError (int sc) throws IOException** - **Checks** for the **status code and sends** to the **user the specified** response message - **After sending** the error message the **buffer is cleared** - response.sendError(response.SC_NOT_FOUND);
setStatus	- **public void HttpServletResponse.setStatus (int sc)** - This **code** is **specified earlier** so that on **receiving the setStatus()** method, the error message is throw. Or **redirected to another default Web page** - response.setStatus(response.SC_NOT_MODIFIED);

Error Handling in Servlet

Reporting Error



```
 * @author Trong Khanh
 */
public class DisplayErrorServlet extends HttpServlet {

    /**...*/
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            RequestDispatcher rd = request.getRequestDispatcher("/Khanh");
            rd.forward(request, response);
        } finally {
            out.close();
        }
    }

    HttpServlet methods. Click on the + sign on the left to e

}
```

Address http://localhost:8084/AJDay3/DisplayErrorServlet

HTTP Status 404 - /AJDay3/Khanh

type Status report

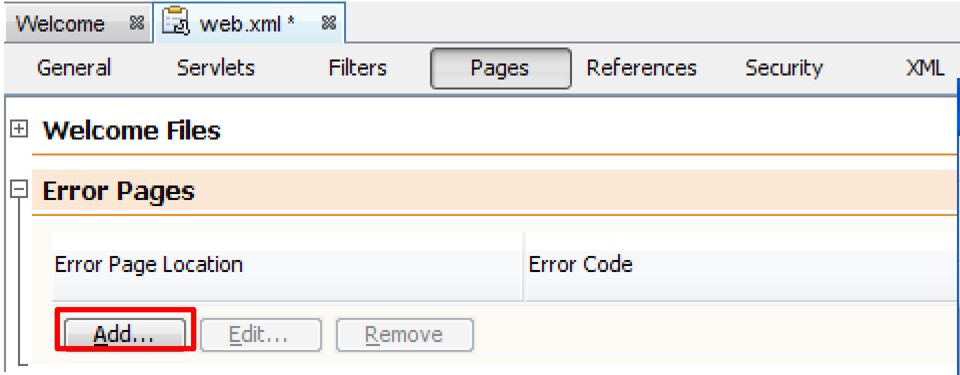
message /AJDay3/Khanh

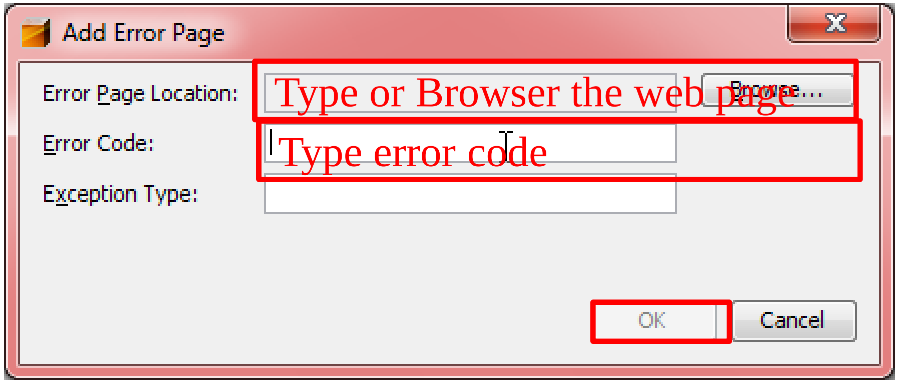
description The requested resource (/AJDay3/Khanh) is not available.

Apache Tomcat/6.0.26

Error Handling in Servlet
Reporting Error

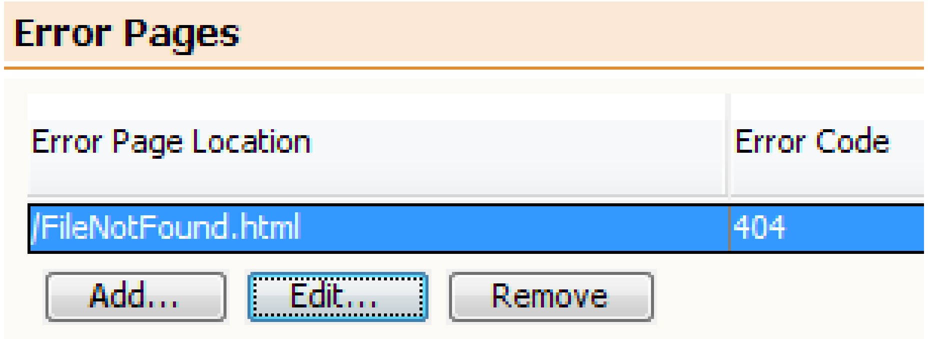
- **Addition the following contents to web.xml file**
  - **In web.xml, choose Page tab, choose Error Pages, click Add**



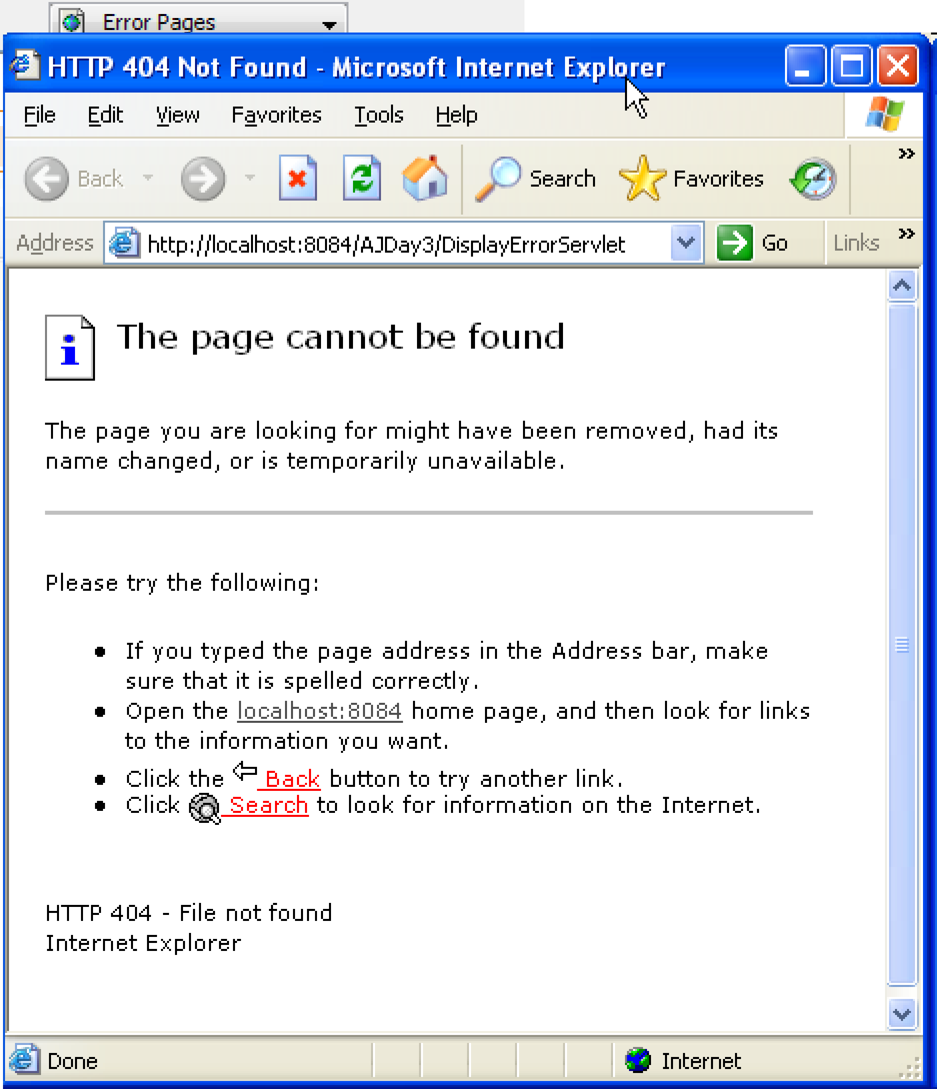


Type or Browser the web page

Type error code



Error Page Location	Error Code
/FileNotFound.html	404



Address: http://localhost:8084/AJDay3/DisplayErrorServlet

The page cannot be found

The page you are looking for might have been removed, had its name changed, or is temporarily unavailable.

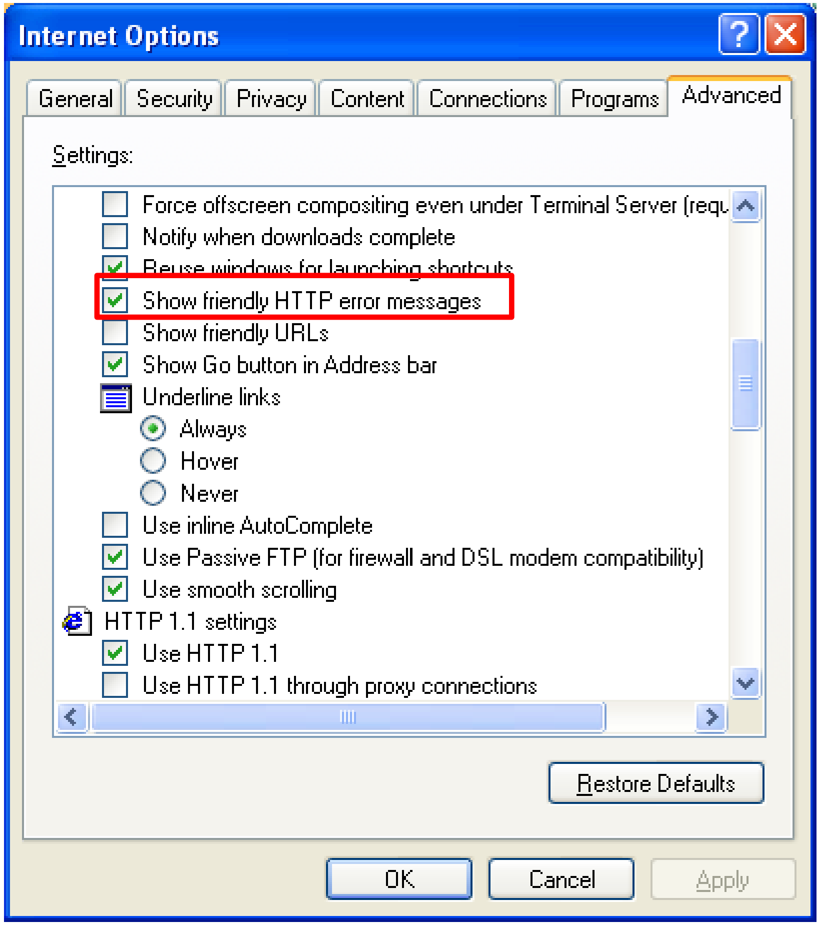
Please try the following:

- If you typed the page address in the Address bar, make sure that it is spelled correctly.
- Open the localhost:8084 home page, and then look for links to the information you want.
- Click the Back button to try another link.
- Click Search to look for information on the Internet.

HTTP 404 - File not found
Internet Explorer

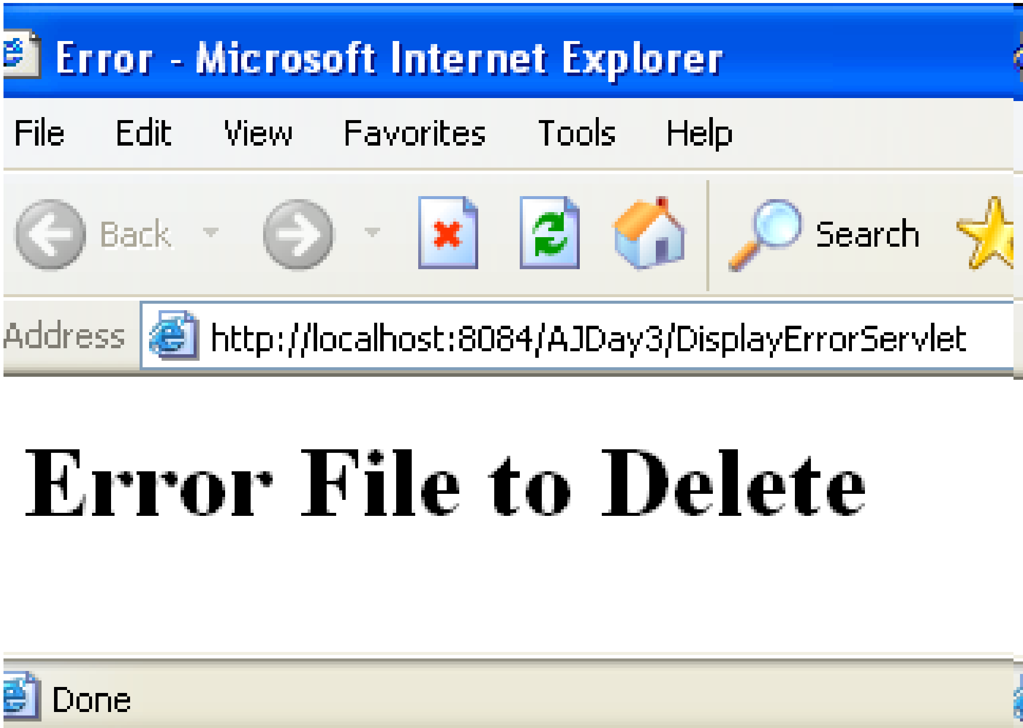
Error Handling in Servlet

Reporting Error



Settings:

- □ Force offscreen compositing even under Terminal Server (requ
- □ Notify when downloads complete
- ☑ Reuse windows for launching shortcuts
- ☑ Show friendly HTTP error messages
- □ Show friendly URLs
- ☑ Show Go button in Address bar
- Underline links
  - ● Always
  - ○ Hover
  - ○ Never
- □ Use inline AutoComplete
- ☑ Use Passive FTP (for firewall and DSL modem compatibility)
- ☑ Use smooth scrolling
- HTTP 1.1 settings
  - ☑ Use HTTP 1.1
  - □ Use HTTP 1.1 through proxy connections



Address: http://localhost:8084/AJDay3/DisplayErrorServlet

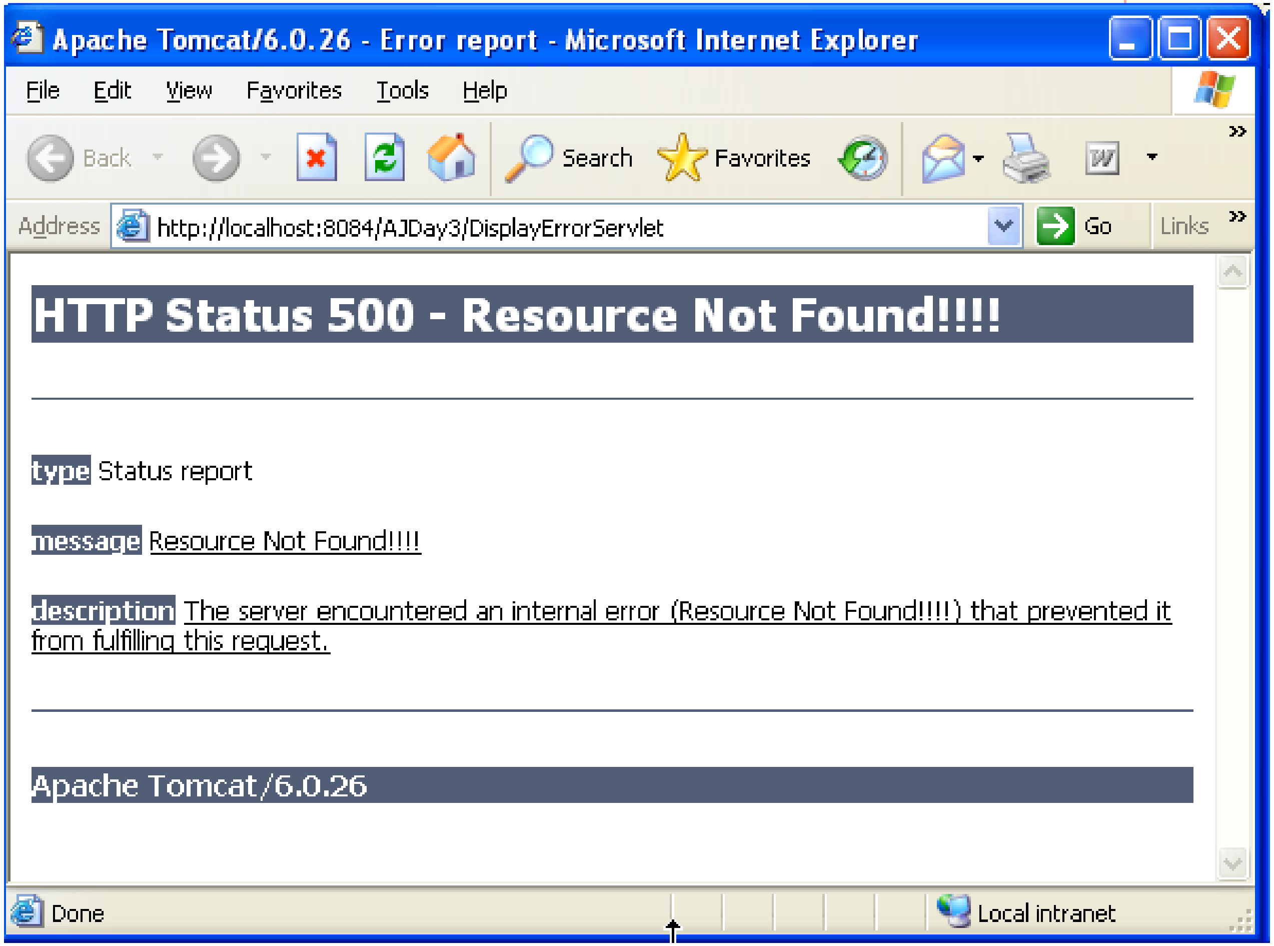
Error File to Delete

Uncheck the option "Show friendly HTTP error messages" from Tools/ "Internet Options" to set up the browser would be presented the user defined message

Error Handling in Servlet

Reporting Error – Example

```
String action = request.getParameter("action");
if(action==null){
    response.sendError(HttpServletResponse.SC_INTERNAL_SERVER_ERROR, "Resource not found");
}
```



Apache Tomcat/6.0.26 - Error report - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://localhost:8084/AJDay3/DisplayErrorServlet
Go
Links

HTTP Status 500 - Resource Not Found!!!!

type Status report

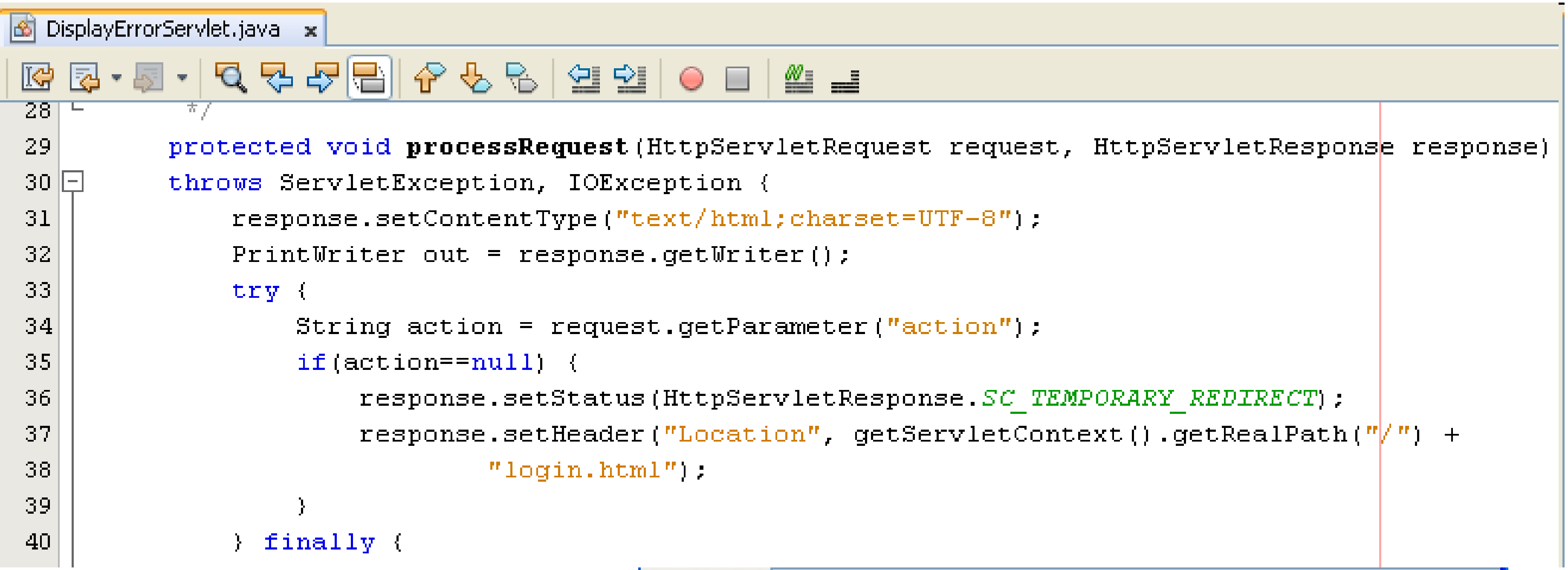
message Resource Not Found!!!!

description The server encountered an internal error (Resource Not Found!!!!) that prevented it from fulfilling this request.

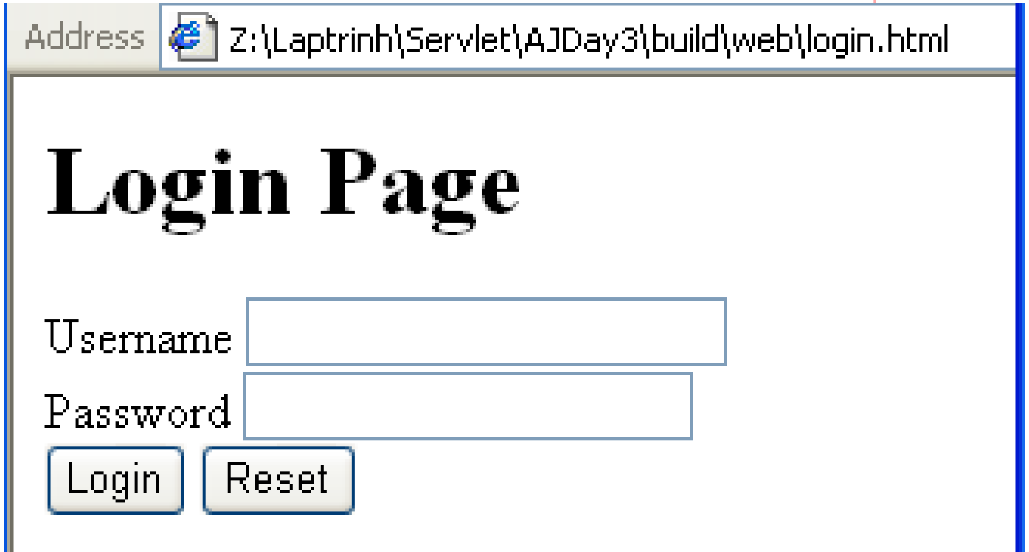
Apache Tomcat/6.0.26

Error Handling in Servlet

Reporting Error – Example



```
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            String action = request.getParameter("action");
            if(action==null) {
                response.setStatus(HttpServletResponse.SC_TEMPORARY_REDIRECT);
                response.setHeader("Location", getServletContext().getRealPath("/") +
                        "login.html");
            }
        } finally {
```



Z:\Laptrinh\Servlet\AJDay3\build\web\login.html

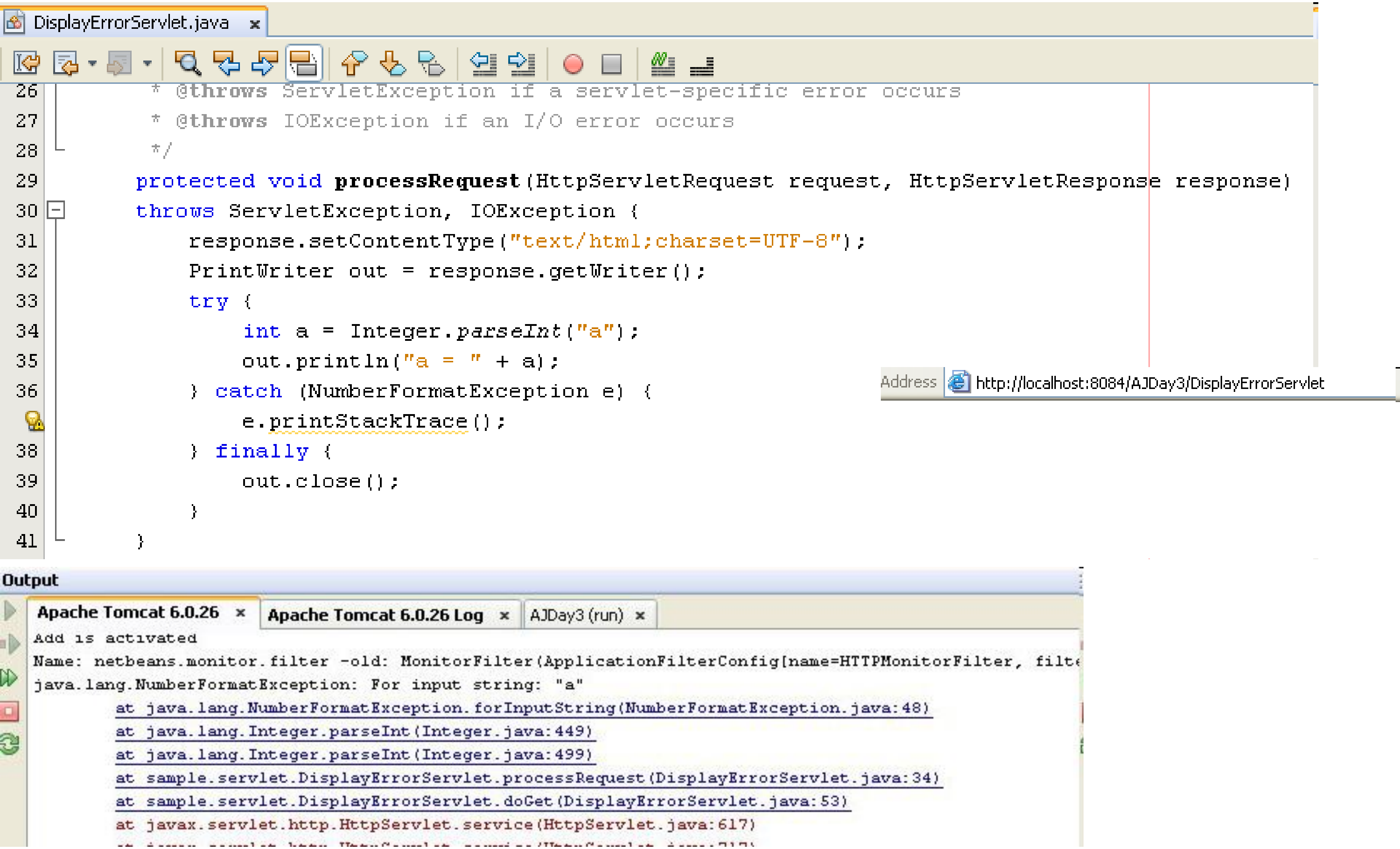
Login Page

Username

Password

Error Handling in Servlet

Logging Error



```
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            int a = Integer.parseInt("a");
            out.println("a = " + a);
        } catch (NumberFormatException e) {
            e.printStackTrace();
        } finally {
            out.close();
        }
    }
```

Address http://localhost:8084/AJDay3/DisplayErrorServlet

```
Add is activated
Name: netbeans.monitor.filter -old: MonitorFilter(ApplicationFilterConfig[name=HTTPMonitorFilter, filte
java.lang.NumberFormatException: For input string: "a"
        at java.lang.NumberFormatException.forInputString(NumberFormatException.java:48)
        at java.lang.Integer.parseInt(Integer.java:449)
        at java.lang.Integer.parseInt(Integer.java:499)
        at sample.servlet.DisplayErrorServlet.processRequest(DisplayErrorServlet.java:34)
        at sample.servlet.DisplayErrorServlet.doGet(DisplayErrorServlet.java:53)
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:617)
```

Error Handling in Servlet

Logging Error

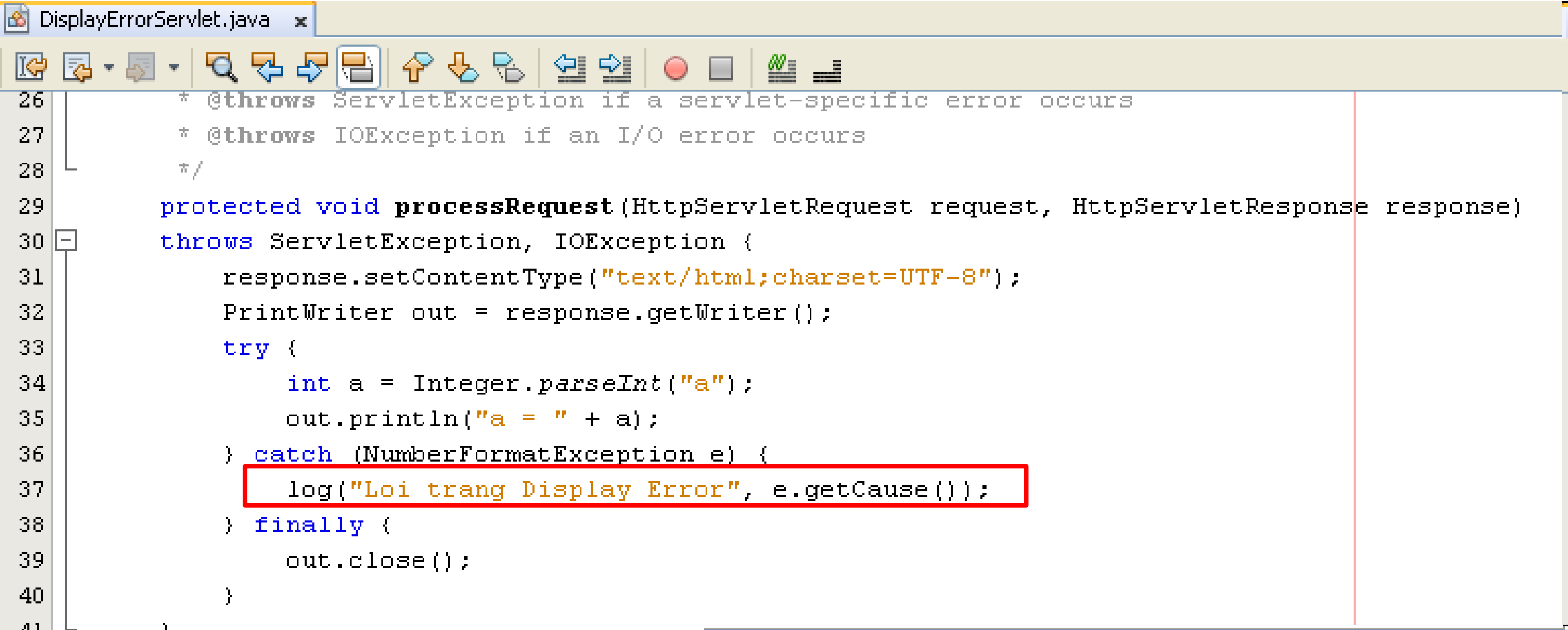
- Servlet can store **the actions and errors through the log() method** of the **GenericServlet class.**
- The log() method also **assists in debugging and can viewed record** in a server
- **Syntax**: **public void log (String msg [, Throwable t])**
- **Ex**:

```
…
log("Servlet is not found ");
response.sendError(response.SC_INTERNAL_SERVER_ERROR, "The requested page ["+ page + "] not found.");
…
```

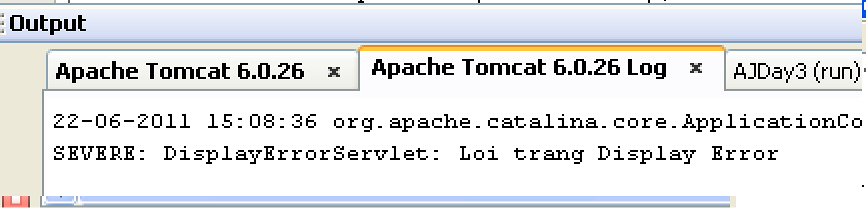
- **A log file locate at**
  - **C:\Documents and Settings\usernames\.netbeans\6.9\apache-tomcat-6.0.26_base\logs\localhost.yyyy-mm-dd.log**
  - **C:\Users\usernames\.netbeans\netbeans\6.9\apache-tomcat-6.0.26_base\logs\localhost.yyyy-mm-dd.log**

Error Handling in Servlet

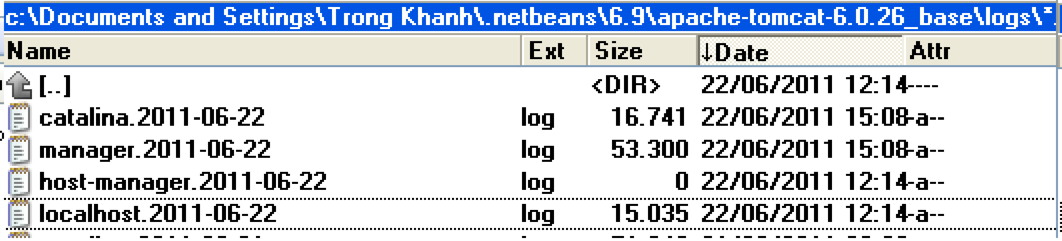
Logging Error



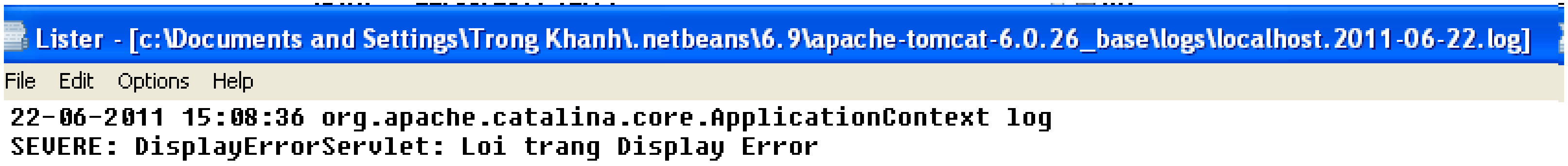
```
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            int a = Integer.parseInt("a");
            out.println("a = " + a);
        } catch (NumberFormatException e) {
            log("Loi trang Display Error", e.getCause());
        } finally {
            out.close();
        }
    }
```



```
22-06-2011 15:08:36 org.apache.catalina.core.ApplicationCo
SEVERE: DisplayErrorServlet: Loi trang Display Error
```


c:\Documents and Settings\Trong Khanh\.netbeans\6.9\apache-tomcat-6.0.26_base\logs\*

Name	Ext	Size	↓Date	Attr
[..]		<DIR>	22/06/2011 12:14	----
catalina.2011-06-22	log	16.741	22/06/2011 15:08	-a--
manager.2011-06-22	log	53.300	22/06/2011 15:08	-a--
host-manager.2011-06-22	log	0	22/06/2011 12:14	-a--
localhost.2011-06-22	log	15.035	22/06/2011 12:14	-a--


Lister - [c:\Documents and Settings\Trong Khanh\.netbeans\6.9\apache-tomcat-6.0.26_base\logs\localhost.2011-06-22.log]

```
22-06-2011 15:08:36 org.apache.catalina.core.ApplicationContext log
SEVERE: DisplayErrorServlet: Loi trang Display Error
```

Summary

- **Sessions and Listeners**
- **Techniques: Error Handling in Servlets**

Q&A

Next Lecture

- **Security Mechanisms**
  - Authentication
  - Authorization
  - Data Integrity
  - Confidentiality
- **Authentication of Security Mechanisms**
  - Basic
  - Form
  - Digest
  - CLIENT-CERT Authentication
  - HTTP Client Authentication
  - JDBC Realms
- **Describe the Deployment Descriptor Declarations**