

Handling Cookies

Lecture 5 – Chapter 8

Core Servlets & JSP book: www.coreservlets.com
 More Servlets & JSP book: www.moreservlets.com
 Servlet and JSP Training Courses: courses.coreservlets.com

Slides © Marty Hall, <http://www.coreservlets.com>, book © Sun Microsystems Press

Agenda

- The potential of cookies
- The problems with cookies
- Sending cookies to browser
- Reading cookies from browser
- Simple cookie-handling servlets
- Cookie utilities
- Methods in the Cookie API
- A customized search engine front end


www.coreservlets.com

The Potential of Cookies

- **Idea**
 - Servlet sends a simple name and value to client.
 - Client returns same name and value when it connects to same site (or same domain, depending on cookie settings).
- **Typical Uses of Cookies**
 - Identifying a user during an e-commerce session
 - Servlets have a higher-level API for this task
 - Avoiding username and password
 - Customizing a site
 - Focusing advertising


www.coreservlets.com

Cookies and Focused Advertising



Cookies

Cookies and Privacy



FoxTrot © 1998 Bill Amend. Reprinted with permission of Universal Press Syndicate. All rights reserved.

www.coreservlets.com

Some Problems with Cookies

- **The problem is privacy, not security.**
 - Servers can remember your previous actions
 - If you give out personal information, servers can link that information to your previous actions
 - Servers can share cookie information through use of a cooperating third party like doubleclick.net
 - Poorly designed sites store sensitive information like credit card numbers directly in cookie
 - JavaScript bugs let hostile sites steal cookies (old browsers)
- **Moral for servlet authors**
 - If cookies are not critical to your task, avoid servlets that totally fail when cookies are disabled
 - Don't put sensitive info in cookies

www.coreservlets.com

Sending Cookies to Browser

- **Standard approach:**

```
Cookie c = new Cookie("name", "value");
c.setMaxAge(...); // Means cookie persists on disk
// Set other attributes.
response.addCookie(c);
```

- **Simplified approach:**

- Use LongLivedCookie class:

```
public class LongLivedCookie extends Cookie {
    public static final int SECONDS_PER_YEAR =
        60*60*24*365;

    public LongLivedCookie(String name, String value) {
        super(name, value);
        setMaxAge(SECONDS_PER_YEAR);
    }
}
```

Cookies

www.coreservlets.com

Reading Cookies from Browser

- **Standard approach:**

```
Cookie[] cookies = request.getCookies();
if (cookies != null) {
    for(int i=0; i<cookies.length; i++) {
        Cookie c = cookies[i];
        if (c.getName().equals("someName")) {
            doSomethingWith(c);
            break;
        }
    }
}
```

- **Simplified approach:**

- Extract cookie or cookie value from cookie array by using ServletUtilities.getCookieValue or ServletUtilities.getCookie

Cookies

www.coreservlets.com

Simple Cookie-Setting Servlet

```
public class SetCookies extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        for(int i=0; i<3; i++) {
            Cookie cookie = new Cookie("Session-Cookie-" + i,
                "Cookie-Value-S" + i);
            response.addCookie(cookie);
            cookie = new Cookie("Persistent-Cookie-" + i,
                "Cookie-Value-P" + i);
            cookie.setMaxAge(3600);
            response.addCookie(cookie);
        }
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println(...);
    }
}
```

Cookies

www.coreservlets.com

Result of Cookie-Setting Servlet



Cookies

www.coreservlets.com

Simple Cookie-Viewing Servlet

```
public class ShowCookies extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String title = "Active Cookies";
        out.println(ServletUtilities.headWithTitle(title) +
            "<BODY BGCOLOR=\"#FDF5E6\">\n" +
            "<H1 ALIGN=\"CENTER\">" + title +
            "</H1>\n" +
            "<TABLE BORDER=1 ALIGN=\"CENTER\">\n" +
            "<TR BGCOLOR=\"#FFAD00\">\n" +
            " <TH>Cookie Name\n" +
            " <TH>Cookie Value");
    }
}
```

Cookies

www.coreservlets.com

Simple Cookie-Viewing Servlet (Continued)

```
Cookie[] cookies = request.getCookies();
if (cookies != null) {
    Cookie cookie;
    for(int i=0; i<cookies.length; i++) {
        cookie = cookies[i];
        out.println("<TR>\n" +
            " <TD>" + cookie.getName() + "\n" +
            " <TD>" + cookie.getValue());
    }
}
out.println("</TABLE></BODY></HTML>");
}
```

Cookies

www.coreservlets.com

Result of Cookie-Viewer (Before & After Restarting Browser)



Cookie Utilities

- **Problem**
 - get_cookies returns an array of cookies
 - You almost always only care about one particular cookie
- **Solution**
 - Static methods to
 - Extract a cookie value given a cookie name (default value if no match)
 - Extract a Cookie object given a cookie name (null if no match)

ServletUtilities.getCookieValue

```
public static String getCookieValue(Cookie[] cookies,
                                   String cookieName,
                                   String defaultVal) {
    if (cookies != null) {
        for(int i=0; i<cookies.length; i++) {
            Cookie cookie = cookies[i];
            if (cookieName.equals(cookie.getName()))
                return(cookie.getValue());
        }
    }
    return(defaultVal);
}
```

ServletUtilities.getCookie

```
public static Cookie getCookie(Cookie[] cookies,
                               String cookieName) {
    if (cookies != null) {
        for(int i=0; i<cookies.length; i++) {
            Cookie cookie = cookies[i];
            if (cookieName.equals(cookie.getName()))
                return(cookie);
        }
    }
    return(null);
}
```

Methods in the Cookie API

- **getDomain/setDomain**
 - Lets you specify domain to which cookie applies. Current host must be part of domain specified.
- **getMaxAge/setMaxAge**
 - Gets/sets the cookie expiration time (in seconds). If you fail to set this, cookie applies to current browsing session only. See LongLivedCookie helper class given earlier.
- **getName/setName**
 - Gets/sets the cookie name. For new cookies, you supply name to constructor, not to setName. For incoming cookie array, you use getName to find the cookie of interest.

Methods in the Cookie API (Continued)

- **getPath/setPath**
 - Gets/sets the path to which cookie applies. If unspecified, cookie applies to URLs that are within or below directory containing current page.
- **getSecure/setSecure**
 - Gets/sets flag indicating whether cookie should apply only to SSL connections or to all connections.
- **getValue/setValue**
 - Gets/sets value associated with cookie. For new cookies, you supply value to constructor, not to setValue. For incoming cookie array, you use getName to find the cookie of interest, then call getValue on the result.

A Customized Search Engine Interface

- Front end remembers settings for search engine, search string, and hits per page

- Front end uses cookies
- Back end sets cookies
- In real life, don't really show previous queries!



Front End to SearchEngines Servlet

```
public class SearchEnginesFrontEnd extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        Cookie[] cookies = request.getCookies();
        String searchString =
            ServletUtilities.getCookieValue(cookies,
                "searchString",
                "Java Programming");

        String numResults =
            ServletUtilities.getCookieValue(cookies,
                "numResults",
                "10");

        String searchEngine =
            ServletUtilities.getCookieValue(cookies,
                "searchEngine",
                "google");
    }
}
```

Front End to SearchEngines Servlet (Continued)

```
...
out.println
(
    "<FORM ACTION=\"/servlet/" +
    "coreservlets.CustomizedSearchEngines\">\n" +
    "<CENTER>\n" +
    "Search String:\n" +
    "<INPUT TYPE=\"TEXT\" NAME=\"searchString\" \n" +
    "    VALUE=\"\" + searchString + "\"><BR>\n" +
    "Results to Show Per Page:\n" +
    "<INPUT TYPE=\"TEXT\" NAME=\"numResults\" \n" +
    "    VALUE=\"\" + numResults + \" SIZE=3><BR>\n" +
    "<INPUT TYPE=\"RADIO\" NAME=\"searchEngine\" \n" +
    "    VALUE=\"google\" +
    checked("google", searchEngine) + ">\n" +
    ...);
```

Customized SearchEngines Servlet (Back End)

```
public class CustomizedSearchEngines extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        String searchString =
            request.getParameter("searchString");
        if ((searchString == null) ||
            (searchString.length() == 0)) {
            reportProblem(response, "Missing search string.");
            return;
        }
        Cookie searchStringCookie =
            new LongLivedCookie("searchString", searchString);
        response.addCookie(searchStringCookie);
        ...
    }
}
```

Summary

- Cookies involve name/value pairs sent from server to browser and returned when the same page, site, or domain is visited later
- Let you
 - Track sessions (use higher-level API)
 - Permit users to avoid logging in at low-security sites
 - Customize sites for different users
 - Focus content or advertising
- Setting cookies
 - Call Cookie constructor, set age, call response.addCookie
- Reading cookies
 - Call request.getCookies, check for null, look through array for matching name, use associated value



Questions?

Core Servlets & JSP book: www.coreservlets.com
More Servlets & JSP book: www.moreservlets.com
Servlet and JSP Training Courses: courses.coreservlets.com

Slides © Marty Hall, <http://www.coreservlets.com>, book © Sun Microsystems Press