


© 2003 Marty Hall



Session Tracking

JSP and Servlet Training Courses: <http://courses.coreservlets.com>
 JSP and Servlet Books from Sun Press: <http://www.coreservlets.com>


Agenda

- Implementing session tracking from scratch
- Using basic session tracking
- Understanding the session-tracking API
- Differentiating between server and browser sessions
- Encoding URLs
- Storing immutable objects vs. storing mutable objects
- Tracking user access counts
- Accumulating user purchases
- Implementing a shopping cart
- Building an online store

JSP/Servlet Training: <http://www.coreservlets.com>

Session Tracking and E-Commerce

- **Why session tracking?**
 - When clients at on-line store add item to their shopping cart, how does server know what's already in cart?
 - When clients decide to proceed to checkout, how can server determine which previously created cart is theirs?



Dilbert used with permission of United Syndicates Inc.

JSP/Servlet Training: <http://www.coreservlets.com>

Rolling Your Own Session Tracking: Cookies

- **Idea: associate cookie with data on server**

```
String sessionID = makeUniqueString();
HashMap sessionInfo = new HashMap();
HashMap globalTable = findTableStoringSessions();
globalTable.put(sessionID, sessionInfo);
Cookie sessionCookie =
    new Cookie("JSESSIONID", sessionID);
sessionCookie.setPath("/");
response.addCookie(sessionCookie);
```
- **Still to be done:**
 - Extracting cookie that stores session identifier
 - Setting appropriate expiration time for cookie
 - Associating the hash tables with each request
 - Generating the unique session identifiers

JSP/Servlet Training: <http://www.coreservlets.com>

Rolling Your Own Session Tracking: URL-Rewriting

- **Idea**
 - Client appends some extra data on the end of each URL that identifies the session
 - Server associates that identifier with data it has stored about that session
 - E.g., <http://host/path/file.html;jsessionid=1234>
- **Advantage**
 - Works even if cookies are disabled or unsupported
- **Disadvantages**
 - Must encode all URLs that refer to your own site
 - All pages must be dynamically generated
 - Fails for bookmarks and links from other sites

JSP/Servlet Training: <http://www.coreservlets.com>

Rolling Your Own Session Tracking: Hidden Form Fields

- **Idea:**

```
<INPUT TYPE="HIDDEN" NAME="session" VALUE="...">
```
- **Advantage**
 - Works even if cookies are disabled or unsupported
- **Disadvantages**
 - Lots of tedious processing
 - All pages must be the result of form submissions

JSP/Servlet Training: <http://www.coreservlets.com>

Session Tracking in Java

- **Session objects live on the server**
- **Sessions automatically associated with client via cookies or URL-rewriting**
 - Use `request.getSession` to get session
 - Behind the scenes, the system looks at cookie or URL extra info and sees if it matches the key to some previously stored session object. If so, it returns that object. If not, it creates a new one, assigns a cookie or URL info as its key, and returns that new session object.
- **Hashtable-like mechanism lets you store arbitrary objects inside session**
 - `setAttribute` stores values
 - `getAttribute` retrieves values

ISP/Servlet training: <http://www.coreservlets.com>

Session Tracking Basics

- **Access the session object**
 - Call `request.getSession` to get `HttpSession` object
 - This is a hashtable associated with the user
- **Look up information associated with a session.**
 - Call `getAttribute` on the `HttpSession` object, cast the return value to the appropriate type, and check whether the result is null.
- **Store information in a session.**
 - Use `setAttribute` with a key and a value.
- **Discard session data.**
 - Call `removeAttribute` discards a specific value.
 - Call `invalidate` to discard an entire session.

ISP/Servlet training: <http://www.coreservlets.com>

Session Tracking Basics: Sample Code

```
HttpSession session = request.getSession();
SomeClass value =
    (SomeClass)session.getAttribute("someID");
if (value == null) {
    value = new SomeClass(...);
    session.setAttribute("someID", value);
}
doSomethingWith(value);
```

- Do not need to call `setAttribute` again (after modifying value) if the modified value is the same object. But, if value is immutable, modified value will be a new object reference, and you must call `setAttribute` again.

ISP/Servlet training: <http://www.coreservlets.com>

What Changes if Server Uses URL Rewriting?

- **Session tracking code:**
 - No change
- **Code that generates hypertext links back to same site:**
 - Pass URL through `response.encodeURL`.
 - If server is using cookies, this returns URL unchanged
 - If server is using URL rewriting, this appends the session info to the URL
 - E.g.:
`String url = "order-page.html";`
`url = response.encodeURL(url);`
- **Code that does `sendRedirect` to own site:**
 - Pass URL through `response.encodeRedirectURL`

ISP/Servlet training: <http://www.coreservlets.com>

HttpSession Methods

- **getAttribute**
 - Extracts a previously stored value from a session object. Returns null if no value is associated with given name.
- **setAttribute**
 - Associates a value with a name. Monitor changes: values implement `HttpSessionBindingListener`.
- **removeAttribute**
 - Removes values associated with name.
- **getAttributeNames**
 - Returns names of all attributes in the session.
- **getId**
 - Returns the unique identifier.

11

ISP/Servlet training: <http://www.coreservlets.com>

HttpSession Methods (Continued)

- **isNew**
 - Determines if session is new to *client* (not to *page*)
- **getCreationTime**
 - Returns time at which session was first created
- **getLastAccessedTime**
 - Returns time at which session was last sent from client
- **getMaxInactiveInterval, setMaxInactiveInterval**
 - Gets or sets the amount of time session should go without access before being invalidated
- **invalidate**
 - Invalidates current session

12

ISP/Servlet training: <http://www.coreservlets.com>

A Servlet that Shows Per-Client Access Counts

```
public class ShowSession extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        HttpSession session = request.getSession();
        String heading;
        Integer accessCount =
            (Integer)session.getAttribute("accessCount");
        if (accessCount == null) {
            accessCount = new Integer(0);
            heading = "Welcome, Newcomer";
        } else {
            heading = "Welcome Back";
            accessCount =
                new Integer(accessCount.intValue() + 1);
        }
        session.setAttribute("accessCount", accessCount);
    }
}
```

15

ISP/Servlet training: <http://www.coreservlets.com>

A Servlet that Shows Per-Client Access Counts (Continued)

```
PrintWriter out = response.getWriter();
```

14

ISP/Servlet training: <http://www.coreservlets.com>

A Servlet that Shows Per-Client Access Counts: Result 1

Info Type	Value
ID	E4DED48A02D66B44A9EC00D3722558C6
Creation Time	Wed Apr 16 11:39:45 EDT 2003
Time of Last Access	Wed Apr 16 11:40:05 EDT 2003
Number of Previous Accesses	0

16

ISP/Servlet training: <http://www.coreservlets.com>

A Servlet that Shows Per-Client Access Counts: Result 2

Info Type	Value
ID	E4DED48A02D66B44A9EC00D3722558C6
Creation Time	Wed Apr 16 11:39:45 EDT 2003
Time of Last Access	Wed Apr 16 11:42:07 EDT 2003
Number of Previous Accesses	1

16

ISP/Servlet training: <http://www.coreservlets.com>

Accumulating a List of User Tracking Data

```
public class ShowItems extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        HttpSession session = request.getSession();
        ArrayList previousItems =
            (ArrayList)session.getAttribute("previousItems");
        if (previousItems == null) {
            previousItems = new ArrayList();
            session.setAttribute("previousItems",
                previousItems);
        }
    }
}
```

17

ISP/Servlet training: <http://www.coreservlets.com>

Accumulating a List of User Data (Continued)

```
String newItem = request.getParameter("newItem");
PrintWriter out = response.getWriter();
...
synchronized(previousItems) {
    if (newItem != null) {
        previousItems.add(newItem);
    }
    if (previousItems.size() == 0) {
        out.println("<I>No items</I>");
    } else {
        out.println("<UL>");
        for(int i=0; i<previousItems.size(); i++) {
            out.println("<LI>" +
                (String)previousItems.get(i));
        }
        out.println("</UL>");
    }
}
out.println("</BODY></HTML>");
}
```

18

ISP/Servlet training: <http://www.coreservlets.com>

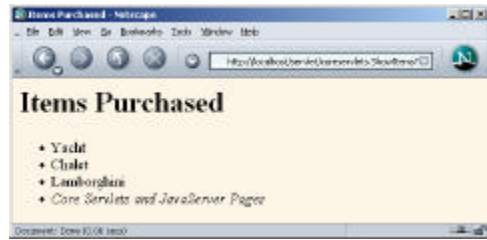
Accumulating a List of User Data: Front End



19

ISP/Servlet training: <http://www.coreservlets.com>

Accumulating a List of User Data: Result



20

ISP/Servlet training: <http://www.coreservlets.com>

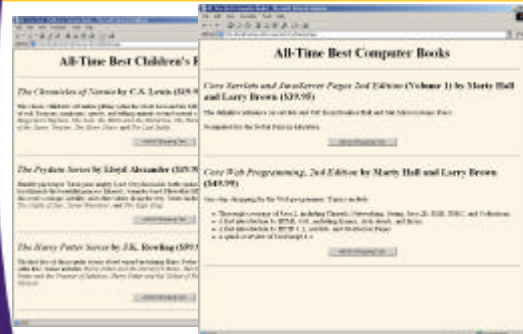
An On-Line Bookstore

- **Session tracking code stays the same as in simple examples**
- **Shopping cart class is relatively complex**
 - Identifies items by a unique catalog ID
 - Does not repeat items in the cart
 - Instead, each entry has a count associated with it
 - If count reaches zero, item is deleted from cart
- **Pages built automatically from objects that have descriptions of books**

21

ISP/Servlet training: <http://www.coreservlets.com>

An On-Line Bookstore



22

An On-Line Bookstore



23

<http://www.coreservlets.com>

Summary

- **Sessions do not travel across network**
 - Only unique identifier does
- **Get the session**
 - request.getSession
- **Extract data from session**
 - session.getAttribute
 - Do typecast and check for null
- **Put data in session**
 - session.setAttribute

24

ISP/Servlet training: <http://www.coreservlets.com>



Questions?

JSP and Servlet Training Courses: <http://courses.coreservlets.com>
JSP and Servlet Books from Sun Press: <http://www.coreservlets.com>