SESSION HANDLING

Session simply means a particular interval of time.Session Tracking is a way to maintain state (data) of a user. It is also known as **session management** in servlet.Http is a stateless protocol that means each request is considered as the new request. So the states are maintained using various session tracking techniques.Each time user requests to the server, server treats the request as the new request. So we need to maintain the state of a user to recognize to particular user.

Session Tracking Techniques

There are four techniques used in Session tracking:Cookies, Hidden Form Field, URL Rewriting and HttpSession Interface

Cookies

A cookie is a small piece of information that persist between the multiple client requests.In HTTP each request is considered as a new request even if two requests are issued by the same user.A cookie is added with response from the servlet which serves as an identifier to the user. This is done when the first request is made.So cookie is stored in the cache of the browser. After that if any request is sent by the user, the stored cookie is added with request by default. Thus, the server recognizes the user as the old user.

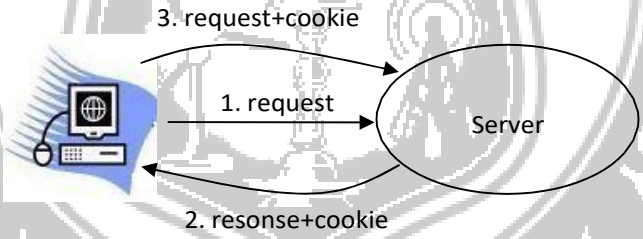

Figure 3.6 Cookies

Differences between Cookie and Session

Session	Cookie
Sessions are **server-side** files that contain user information	Cookies are **client-side** files that contain user information
Any amount of data can be stored within sessions.	Official MAX Cookie size is 4KB
Session ends when user close his browser.	Cookie ends depends on the life time you setfor it.

Hidden Form field

Here, a hidden (invisible) textfield is used for maintaining the state of end user. In such case, the information is stored in the hidden field. This approach is better if we have to submit form in all the pages and we don't want to depend on the browser.

<input type="hidden" name=”Field name" value="value">

Example: <INPUT TYPE="HIDDEN" NAME="session" VALUE="a1234">

This entry means that, when the form is submitted, the specified name and value are automatically included in the GET or POST data.

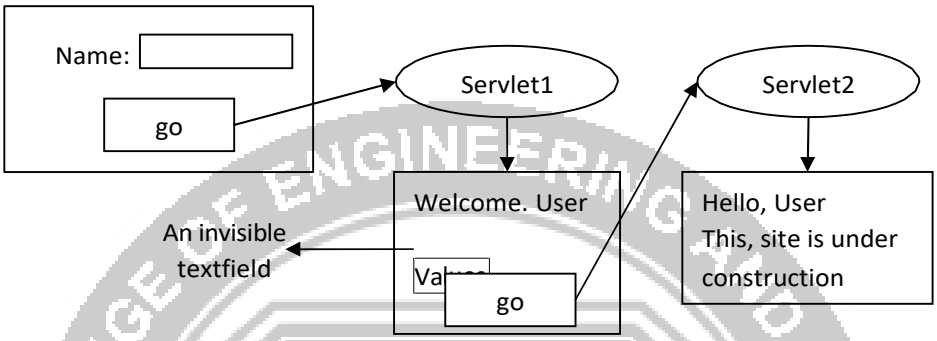

Figure 3.7 Hidden Form field

Advantage of Hidden Form Field

1. It will always work whether cookie is disabled or not.

Disadvantages of Hidden Form Field:

1. It is maintained at server side.
2. Extra form submission is required on each pages.
3. Only textual information can be used.

URL Rewriting

Appending the name of the user in the query string and getting the value from the query string in another page is called URL rewriting. The parameter **name-value pairs** are passed using the following format:

url?name1=value1&name2=value2&??

A name and a value is separated using an equal = sign, a parameter name-value pair is separated from another parameter using the ampersand(&). With URL rewriting, every local URL the user might click on is dynamically modified, or rewritten, to include extra information. The extra information can be in the form of extra path information, added parameters, or some custom, server-specific URL change.

Due to the limited space available in rewriting a URL, the extra information is usually limited to a unique session ID. When the user clicks the hyperlink, the parameter name/value pairs will be passed to the server. From a, the getParameter() method to obtain a parameter value.

http://server:port/servlet/Rewritten	Original URL
http://server:port/servlet/Rewritten/123	extra path information
http://server:port/servlet/Rewritten?sessionid=123	added parameter
http://server:port/servlet/Rewritten;$sessionid$123	custom change

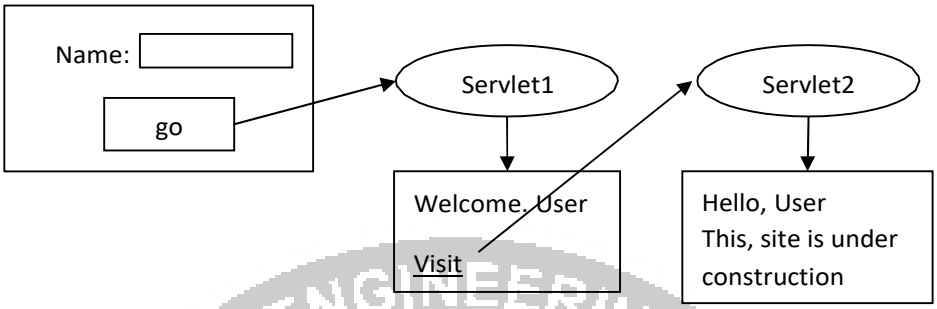

Figure 3.7(a) URL rewriting

Advantage of URL Rewriting

1. It will always work whether cookie is disabled or not (browser independent).
2. Extra form submission is not required on each pages.

Disadvantage of URL Rewriting

1. It will work only with links.
2. It can send only textual information.

HTTP Session Interface

The web container creates a session id for each user.The container uses this id to identify the particular user.An object of HttpSession can be used to perform two tasks:bind objects and view and manipulate information about a session, such as the session identifier, creation time, and last accessed time.

The HttpServletRequest interface provides two methods to get the object of HttpSession created by the container:

1. public HttpSession getSession()-Returns the current session associated with this request, or if the request does not have a session, creates one.
2. public HttpSession getSession(boolean create):-Returns the current HttpSession associated with this request or, if there is no current session and create is true, returns a new session.

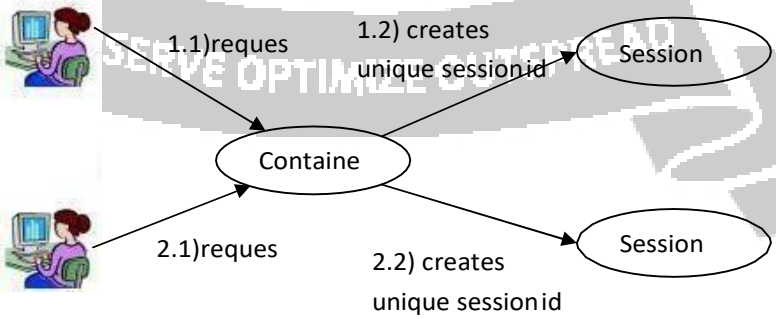

Figure 3.8 Creating session

Methods of HttpSession interface

Method	Description
public Object getAttribute(String name)	returns the object bound with the specified name in this session, or null if no object is bound under the name.
public Enumeration getAttributeNames()	returns an Enumeration of String objects containing the names of all the objects bound to this session.
public long getCreationTime()	returns the time when this session was created, measured in milliseconds since midnight January 1, 1970 GMT.
public String getId()	returns a string containing the unique identifier assigned to this session.
public long getLastAccessedTime()	returns the last time the client sent a request associated with this session, as the number of milliseconds since midnight January 1, 1970 GMT.
public int getMaxInactiveInterval()	returns the maximum time interval, in seconds, that the servlet container will keep this session open between client accesses.
public void invalidate()	invalidates this session and unbinds any objects bound to it.
public boolean isNew()	returns true if the client does not yet know about the session or if the client chooses not to join the session.
public void removeAttribute(String name)	removes the object bound with the specified name from this session.
public void setAttribute(String name, Object value)	binds an object to this session, using the name specified.
public void setMaxInactiveInterval(int interval)	specifies the time, in seconds, between client requests before the servlet container will invalidate this session.

Session tracking

```
import java.io.*;import javax.servlet.*;import javax.servlet.http.*;
import java.util.*;
public class SessionTrack extends HttpServlet {
```

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
{     HttpSession session = request.getSession(true); //session id is created
    Date createTime = new Date(session.getCreationTime()); // Get session creation time.
    Date lastAccessTime = new Date(session.getLastAccessedTime());
    // Get last access time of this web page.
    String title = "Welcome Back to my website";
    Integer visitCount = new Integer(0);
    String visitCountKey = new String("visitCount");
    String userIDKey = new String("userID");
    String userID = new String("ABCD");
    // Check if this is new comer on your web page.
    if (session.isNew()){       title = "Welcome to my website";
      session.setAttribute(userIDKey, userID);
    } else {
      visitCount = (Integer)session.getAttribute(visitCountKey);
      visitCount = visitCount + 1;
      userID = (String)session.getAttribute(userIDKey);
    }
    session.setAttribute(visitCountKey, visitCount);
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html>\n" + "<head><title>" + title + "</title></head>\n" +
          "<body bgcolor=\"#f0f0f0\">\n" +      "<h1 align=\"center\">" + title +
"</h1>\n" +
          "<h2 align=\"center\">Session Infomation</h2>\n" +
          "<table border=\"1\" align=\"center\">\n" + "<tr bgcolor=\"#949494\">\n" +
          " <th>Session info</th><th>value</th></tr>\n" + "<tr>\n"+
          " <td>id</td>\n" + " <td>" + session.getId() + "</td></tr>\n" +
          "<tr>\n" + " <td>Creation Time</td>\n" + " <td>" + createTime +
          " </td></tr>\n" + "<tr>\n" +" <td>Time of Last Access</td>\n" +
          "  <td>" + lastAccessTime +  " </td></tr>\n" +   "<tr>\n" + " <td>User
```

```
ID</td>\n" +
        " <td>" + userID + " </td></tr>\n" + "<tr>\n" + " <td>Number of
visits</td>\n" +
        "  <td>"  +  visitCount  +  "</td></tr>\n"  +"</table>\n"  +
"</body></html>");
 }}
```

Welcome to my website

Session Information

Session info	Value
id	0AE3EC93FF44E3C525B4351B77ABB2D5
Creation Time	Tue Jun 08 17:26:40 GMT+04:00 2010
Time of Last Access	Tue Jun 08 17:26:40 GMT+04:00 2010
User ID	ABCD
Number of visits	0

Now try to run the same servlet for second time, it would display following result.

Welcome Back to my website

Session Information

info type	Value
Id	0AE3EC93FF44E3C525B4351B77ABB2D5
Creation Time	Tue Jun 08 17:26:40 GMT+04:00 2010
Time of Last Access	Tue Jun 08 17:26:40 GMT+04:00 2010
User ID	ABCD
Number of visits	1