Session Tracking Methods

Last modified on September 7th, 2014 by Joe.

Following answer is applicable irrespective of the language and platform used. Before we enter into session tracking, following things should be understood.

**What is a session?**

A session is a conversation between the server and a client. A conversation consists series of continuous request and response.

**Why should a session be maintained?**

When there is a series of continuous request and response from a same client to a server, the server cannot identify from which client it is getting requests. Because HTTP is a stateless protocol.

When there is a need to maintain the conversational state, session tracking is needed. For example, in a shopping cart application a client keeps on adding items into his cart using multiple requests. When every request is made, the server should identify in which client’s cart the item is to be added. So in this scenario, there is a certain need for session tracking.

Solution is, when a client makes a request it should introduce itself by providing unique identifier every time. There are five different methods to achieve this.

**Session tracking methods:**

1. User authorization
2. Hidden fields
3. URL rewriting
4. Cookies
5. Session tracking API

The first four methods are traditionally used for session tracking in all the server-side technologies. The session tracking API method is provided by the underlying technology (java servlet or PHP or likewise). Session tracking API is built on top of the first four methods.

**1. User Authorization**

Users can be authorized to use the web application in different ways. Basic concept is that the user will provide username and password to login to the application. Based on that the user can be identified and the session can be maintained.

**2. Hidden Fields**

<INPUT TYPE=”hidden” NAME=”technology” VALUE=”servlet”>  
Hidden fields like the above can be inserted in the webpages and information can be sent to the server for session tracking. These fields are not visible directly to the user, but can be viewed using view source option from the browsers. This type doesn’t need any special configuration from the browser of server and by default available to use for session tracking. This cannot be used for session tracking when the conversation included static resources lik html pages.

**3. URL Rewriting**

Original URL: <http://server:port/servlet/ServletName>  
Rewritten URL: <http://server:port/servlet/ServletName?sessionid=7456>  
When a request is made, additional parameter is appended with the url. In general added additional parameter will be sessionid or sometimes the userid. It will suffice to track the session. This type of session tracking doesn’t need any special support from the browser. Disadvantage is, implementing this type of session tracking is tedious. We need to keep track of the parameter as a chain link until the conversation completes and also should make sure that, the parameter doesn’t clash with other application parameters.

**4. Cookies**

Cookies are the mostly used technology for session tracking. Cookie is a key value pair of information, sent by the server to the browser. This should be saved by the browser in its space in the client computer. Whenever the browser sends a request to that server it sends the cookie along with it. Then the server can identify the client using the cookie.  
In java, following is the source code snippet to create a cookie:

Cookie cookie = new Cookie(“userID”, “7456”);  
res.addCookie(cookie);

Session tracking is easy to implement and maintain using the cookies. Disadvantage is that, the users can opt to disable cookies using their browser preferences. In such case, the browser will not save the cookie at client computer and session tracking fails.

**5. Session tracking API**

Session tracking API is built on top of the first four methods. This is inorder to help the developer to minimize the overhead of session tracking. This type of session tracking is provided by the underlying technology. Lets take the java servlet example. Then, the servlet container manages the session tracking task and the user need not do it explicitly using the java servlets. This is the best of all methods, because all the management and errors related to session tracking will be taken care of by the container itself.

Every client of the server will be mapped with a javax.servlet.http.HttpSession object. Java servlets can use the session object to store and retrieve java objects across the session. Session tracking is at the best when it is implemented using session tracking api.