

Servlet Session Tracking

Introduction to Session Tracking

- To help the server distinguish between clients, each client must identify itself to the server.
- There are a number of popular techniques for distinguishing between clients.
- Two common techniques are **cookies** and **sessions**
- Two other techniques are hidden forms and URL-rewriting.

Cookies in Session Tracking

- Cookies are a popular technique for customizing web pages.
- Browsers can store cookies on the user's computer for retrieval later in the same browsing session or in future browsing sessions.
- For example, cookies are used in on-line shopping applications to store unique identifiers for the users.
- When users add items to their on-line shopping carts or perform other tasks resulting in a request to the web server, the server receives cookies containing unique identifiers for each user.
- The server then uses the unique identifier to locate the shopping carts and perform any necessary processing.
- Cookies are also used to indicate the client's shopping preferences.
- When the servlet receives the client's next communication, the servlet can examine the cookie(s) it sent to the client in a previous communication, identify the client's preferences and immediately display products of interest to that particular client.

How to create, add a Cookie and retrieve cookie

- **Creation of Cookie**

- `Cookie()`
 - `Cookie ob=new Cookie();`
 - `ob.setValue(String value);`
- `Cookie(String name, String value)`
 - `Cookie ob=new Cookie(String name, String value)`

- **Adding Cookie**

- Add the created cookie object to the response object
- `response.addCookie(cookie object);`//adding cookie in the response
- Ex: `response.addCookie(ob);`

- **Retrieving Cookie** (while retrieving, retrieve all the cookies created. So we get an array of cookies)

- `Cookie ob[]=request.getCookies();`

- **Read the data from the Cookie**

- Use `String getName()` and `String get Value()` to get the name of the cookie and value stored in the cookie

```
<form >  
Name: <input type=text name=userName></br>  
<input type=submit value=Go>  
</form>
```



Html Page

```
import java.io.*;  
import javax.servlet.*;  
import javax.servlet.http.*;  
public class A extends HttpServlet {  
    public void doPost(HttpServletRequest request, HttpServletResponse response)throws  
ServletException,IOException{  
    try{  
        response.setContentType("text/html");  
        PrintWriter out = response.getWriter();  
        String n=request.getParameter("userName");  
        Cookie ck=new Cookie("uname",n);//creating cookie object  
        response.addCookie(ck);//adding cookie in the response  
  
        Cookie ob[]=request.getCookies(); //retrieve all cookies  
        out.print("Hello "+ob[0].getValue()); //reading data stored in cookie  
  
    }catch(Exception e){System.out.println(e);}  
    }  
}
```

Setting time to cookie
ck. setMaxAge(5000);
Cookie exists for 5000 seconds



Corresponding Servlet class

Output : Hello VRSEC

Session Tracking using Session (HttpSession)

- The container creates a session id for each user.
- The container uses this id to identify the particular user
- An object of HttpSession is used to
 1. bind objects
 2. view and manipulate information about a session, such as the session identifier, creation time, and last accessed time.
- Constructors
 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.
- Ex: `HttpSession session = request.getSession();`
- Methods :
 - `void setAttribute(String name, String value)`
 - `Object getAttribute()`

Some of the methods in Session

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

```
<form >
Name: <input type=text name=userName></br>
<input type=submit value=Go>
</form>
```



Html Page

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class A extends HttpServlet {
    public void doPost(HttpServletRequest request, HttpServletResponse response)throws ServletException,IOException{
        try{
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
            String n=request.getParameter("userName");
            HttpSession session=request.getSession(); //creating session
            session.setAttribute("uname",n); //setting value to session

            String n=(String)session.getAttribute("uname"); //retrive the value from the session
            out.print("Hello "+n);
        }catch(Exception e){System.out.println(e);}
    }
}
```



Corresponding Servlet class

Output : Hello VRSEC