COOKIES

The cookie is stored on the user's machine, but it is not an executable program and cannot do anything to the stored machine.

A cookie is a short piece of data, not code, which is sent from a web server to a web browser when that browser visits the server's site.

Cookies are embedded in the HTML information flowing back and forth between the user's computer and the servers.Whenever a web browser requests a file from the web server it sends the request along with a cookie, the browser sends a copy of that cookie back to the server along with the request. Thus a server sends a cookie and the browser sends it back whenever it request another file from the same server. In this way, the server knows that the user have visited before and can coordinate the access to different pages on its web site.

Example: An Internet shopping site uses a cookie to keep track of which shopping basket belongs to a user.

Components of cookies

Cookies are a plain text data record of 5 variable-length fields:

- **Expiry time**: The date the cookie will expire. If this is blank, the cookie will expire when the visitor quits the browser.
- **Domain:** The domain name of the site.
- **Path:** The path to the directory or web page that set the cookie.
- **Secure:** If this field contains the word "secure" then the cookie may only be retrieved with a secure server. If this field is blank, no such restriction exists.
- **Name=Value:** Cookies are set and retrieved in the form of key and value pairs.

Uses of cookies

Identifying a user during an e-commerce session, Avoiding username and password, Customizing a site and Focusing advertising

Types of cookies

There are two types of cookies: Session Cookies and persistent cookies

Session Cookies

Session cookies are stored in memory during the applet or application session.Session cookies expire when the applet or application exits and are automatically deleted. These cookies usually store a session ID that is not personally identifiable to users, allowing the user to move from page to page without having to log-in repeatedly. They are widely used by commercial web sites.

Permanent or Persistent Cookies

Permanent cookies are stored in persistent storage and are not deleted when the application exits. They are deleted when they expire. They can retain user preferences for a particular web site, allowing those preferences to be used in future sessions. Permanent cookies can be used to identify individual users, so they may be used by web sites to analyze user's surfing behavior within the web site. These cookies can also be used to provide information about the numbers of visitors, the average time spent on a particular page, and the general performance of the web site.They are usually configured to keep track of users for a prolonged period of time, in some cases many years into the future.

Methods in Servlet Cookies:

Method	Description
public void setDomain(String pattern)	sets the domain to which cookie applies.
public String getDomain()	gets the domain to which cookie applies.
public void setMaxAge(int expiry)	sets how much time (in seconds) should elapse before the cookie expires. By default, the cookie will last only for the current session.
public int getMaxAge()	returns the maximum age of the cookie, specified in seconds, By default,
public String getName()	returns the name of the cookie. The name cannot be changed after creation.
public void setValue(String newValue)	sets the value associated with the cookie.
public String getValue()	gets the value associated with the cookie.
public void setPath(String uri)	sets the path to which this cookie applies. By default, the cookie is returned for all URLs in the same directory as the current page as well as all subdirectories.
public String getPath()	gets the path to which this cookie applies.
public void setSecure(boolean flag)	sets the boolean value indicating whether the cookie should only be sent over encrypted (i.e. SSL) connections.
public void setComment(String purpose)	specifies a comment that describes a cookie's purpose. The comment is useful if the browser presents the cookie to the user.

public String getComment()	returns the comment describing the purpose of this cookie, or null if the cookie has no comment.
public void setDomain(String pattern)	sets the domain to which cookie applies.

Programming cookies

Setting Cookies with Servlet

Setting cookies with servlet involves three steps:

(1) **Creating a Cookie object:** Call the Cookie constructor with a cookie name and a cookie value, both of which are strings.

```
Cookie cookie = new Cookie("key","value");
```

(2) **Setting the maximum age**:Use setMaxAge() to specify how long (in seconds) the cookie should be valid. Following would set up a cookie for 20 hours.

```
cookie.setMaxAge(60*60*20);
```

(3) **Sending the Cookie into the HTTP response headers:** Use response.addCookie() to add cookies in the HTTP response header as follows:

```
response.addCookie(cookie);
```

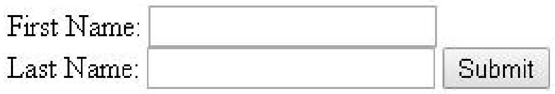
Setting cookies

```
import java.io.*;import javax.servlet.*;import javax.servlet.http.*;
public class HelloForm extends HttpServlet {
  public void doGet(HttpServletRequest request, HttpServletResponse response)
          throws ServletException, IOException
  { Cookie firstName = new Cookie("first_name",
request.getParameter("first_name"));
      Cookie lastName = new Cookie("last_name",
request.getParameter("last_name"));
      firstName.setMaxAge(60*60*24);
      lastName.setMaxAge(60*60*24);
      response.addCookie( firstName );
      response.addCookie( lastName );
      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" +
        "<ul>\n" + " <li><b>First Name</b>: " +
request.getParameter("first_name") + "\n" + " <li><b>Last
Name</b>: " + request.getParameter("last_name") + "\n" +
        "</ul>\n" + "</body></html>");
  } }
```

Form.html

```
<html><body>
<form action="HelloForm" method="GET">
First Name: <input type="text" name="first_name">
<br />
Last Name: <input type="text" name="last_name" />
<input type="submit" value="Submit" />
</form></body></html>
```



Reading Cookies with Servlet:

To read cookies, create an array of javax.servlet.http.Cookie objects by calling the getCookies( ) method of HttpServletRequest. Then cycle through the array, and use getName() and getValue() methods to access each cookie and associated value.

Reading Cookies

```
import java.io.*;import javax.servlet.*;import javax.servlet.http.*;
public class ReadCookies extends HttpServlet {
  publicvoiddoGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
  { Cookie cookie = null;
     Cookie[] cookies = null;
```

```
      cookies = request.getCookies();
        response.setContentType("text/html");
        PrintWriter  out  =  response.getWriter();
           out.println( "<html>\n" + "<head><title>" + title + "</title></head>\n" +
              "<body bgcolor=\"#f0f0f0\">\n" );
       if( cookies != null )
     { out.println("<h2> Found Cookies Name and Value</h2>");
           for (int i = 0; i < cookies.length; i++)
          {                cookie = cookies[i];
                 out.print("Name : " + cookie.getName( ) + ", ");
                   out.print("Value: " + cookie.getValue( )+" <br/>");        }     }
else{        out.println("<h2>No cookies founds</h2>");     }
      out.println("</body>");      out.println("</html>");  }}
```

```
Found Cookies Name and Value
Name : first_name, Value: Priya
Name : last_name, Value: Dharshini
```

Delete Cookies with Servlet:

The following three steps will delete a cookie:

1. Read an already exsiting cookie and store it in Cookie object.
2. Set cookie age as zero using setMaxAge() method to delete an existing cookie.
3. Add this cookie back into response header.

```
import java.io.*;import javax.servlet.*;import javax.servlet.http.*;
public class DeleteCookies extends HttpServlet {
  public void doGet(HttpServletRequest request, HttpServletResponse response)
           throws ServletException, IOException
  {          Cookie cookie = null;
           Cookie[]  cookies  =null;
                  cookies = request.getCookies();
                 response.setContentType("text/html");
```

```
              PrintWriter out = response.getWriter();
            out.println("<html>\n" + "<head><title>" + title + "</title></head>\n" +
             "<body bgcolor=\"#f0f0f0\">\n" );
      if( cookies != null )       {
      out.println("<h2> Cookies Name and Value</h2>");
      for (int i = 0; i < cookies.length; i++)
      {          cookie = cookies[i];
        if((cookie.getName( )).compareTo("first_name") == 0 )
        {   cookie.setMaxAge(0);
              response.addCookie(cookie);
              out.print("Deleted cookie : " + cookie.getName( ) + "<br/>");
        }
        out.print("Name : " + cookie.getName( ) + ", ");
        out.print("Value: " + cookie.getValue( )+" <br/>");         }     }
else
{  out.println("<h2>No  cookies founds</h2>");     }
     out.println("</body>");     out.println("</html>");  }}
```

```
Cookies Name and  Value
Deleted cookie : first_name
Name : first_name,  Value: Priya
Name : last_name,  Value: Dharshini
```

To delete the cookies in Internet Explorer manually. Start at the Tools menu and select Internet Options. To delete all cookies, press Delete Cookies.

Advantages of cookies:

- Cookies are simple to use and implement.
- Occupies less memory as they do not require any server resources and are stored on the user's computer so no extra burden on server.
- Cookies can be configured to expire when the browser session ends (session cookies) or they can exist for a specified length of time on the client’s computer (persistent cookies).
- Cookies persist a much longer period of time than Session state.

Disadvantages of cookies:

- Cookies are not secure as they are stored in clear text they may pose a possible security risk as anyone can open and tamper with cookies.
- Several limitations exist on the size of the cookie text (4kb in general), number of cookies(20 per site in general), etc.
- User has the option of disabling cookies on his computer from browser's settings.
- Cookies will not work if the security level is set to high in the browser.
- Users can delete cookies.
- Users browser can refuse cookies.
- Complex type of data not allowed (e.g. dataset etc). It allows only plain text (i.e. cookie allows only string content.