UNIT : 5

Handling Cookies

Reference :-Core Servlets and Java Server Pages Volume -1
Chapter :- 8 : Handling Cookies

(UNIT :5) CH -8 HANDLING COOKIES

➤ **Benefits of Cookies**
➤ 1) Identifying a user during an e-commerce session
➤ 2) Remembering usernames and passwords.
➤ 3) Customizing sites.
➤ 4) Focusing advertising.

DELETING COOKIES

➤ **To delete your cookies in Internet Explorer:**
➤ start at the Tools menu and select Internet Options. To delete all cookies, press Delete Cookies.
➤ **To delete your cookies in Firefox:**
➤ start at the Tools menu and select Options and click on privacy tab and in that click on remove individual cookies link and click on Remove cookies button.
➤ **To delete your cookies in chrome :**
➤ setting-> Privacy->clear browsing data->delete cookies and other site and plug-in data

---

SENDING AND RECEIVING COOKIES

➤ To send cookies to the client, it involves three steps
➤ 1) **Creating a Cookie object.**
➤ 2) **Setting the maximum age**
 – If you want the browser to store the cookie on disk instead of just keeping it in memory, you use setMaxAge to specify how long (in seconds) the cookie should be valid.
3) **Placing the Cookie into the HTTP response headers**
 • You use response.addCookie to accomplish this. If you forget this step, no cookie is sent to the browser!

➤ **Creating a Cookie Object**
➤ You create a cookie by calling the Cookie constructor, which takes two strings: the cookie name and the cookie value.
➤ Neither the name nor the value should contain white space or any of the following characters:

   [ ] ( ) = , " / ? @ : ;

➤ For Example :
➤ **Cookie c = new Cookie("key", "value");**

➤ **Setting the Maximum Age**
➤ If you create a cookie and send it to the browser, by default it is a **session-level** cookie: a cookie that is stored in the browser's memory and deleted when the user quits the browser.
➤ If you want the browser to store the cookie on disk, use setMaxAge with a time in seconds, as below.
➤ c.setMaxAge(15); //15 second
 Setting the maximum age instructs the browser to delete the cookie.

> **Placing the Cookie in the Response Headers**

> Cookie c1= new Cookie("unm", "mca");
  c1.setMaxAge(60);
  response.addCookie(c1);

> **OR**

> **response.addCookie(new Cookie("unm","mca"));**

> **Reading Cookies from the Client**
> 1) Call request.getCookies().
> This will return an array of Cookie objects.
> 2) Loop down the array, calling getName() on each one until you find the cookie of interest.

READING COOKIES FROM THE CLIENT

> String cookieName = "user";
> Cookie[] c1= request.getCookies();
> if (c1 != null)
> {
      for(int i=0; i<c1.length; i++)
      {
              Cookie c = c1[i];
              if (cookieName.equals(c.getName()))
              {
                      out.println(c.getValue());
              }
      }
}
Example : RepeatVisitor

TYPES OF COOKIES

> **Session cookies** – Session cookies are stored in memory and are accessible as long as the user is using the web application.
> Session cookies are lost when the user exits the web application. Such cookies are identified by a session ID and are most commonly used to store details of a shopping cart.
> **Permanent cookies** – Permanent cookies are used to store long-term information such as user preferences and user identification information.
> Permanent cookies are stored in persistent storage and are not lost when the user exits the application. Permanent cookies are lost when they expire.

Differentiating session cookies from persistent cookies

> Here, program performs two tasks.
> 1) The servlet sets four outgoing cookies.
> 2) The servlet uses request.getCookies to find all the incoming cookies and display their names and values.
> Example : CookieTest.java

USING COOKIE ATTRIBUTES

> Before adding the cookie to the outgoing headers, you can set various characteristics of the cookie by using the following setXxx methods, where Xxx is the name of the attribute you want to specify.
> Here are the Attributes of Cookies :

ATTRIBUTES OF COOKIES

Parameter	Description
Name	Name of cookie
Value	A value of the cookie
Comment	A comment explaining the purpose of the cookie
Max-Age	Maximum age of the cookie (a time in seconds after which the client should not send the cookie back to the server) .Default value is -1.
Domain	Domain to which the cookie should be sent
Path	The path to which the cookie should be sent
Secure	Specifies if the cookie should be sent securely via HTTPS
Version	Version of the cookie protocol. Version 0 is meant for the original Netscape version of cookies. Version 1 is meant for cookies standardized via RFC 2109. (Request for comment). Default value is 0.

METHODS.....

Some of the methods of Cookie class are given below:

Set Methods	Get Methods
	getName ( )
setValue( )	getValue ( )
setComment( )	getComment ( )
setMaxAge( )	getMaxAge ( )
setDomain ( )	getDomain ( )
setPath ( )	getPath ( )
setSecure ( )	getSecure ( )
setVersion ( )	getVersion ( )

> **public void setComment(String comment)**
> **public String getComment()**
> **public void setDomain(String domainPattern)**
> **public String getDomain()**
> – These methods set or retrieve the domain to which the cookie applies.
> **public void setMaxAge(int lifetimeinsecond)**
> **public int getMaxAge()**
> **public void setPath(String path)**
> **public String getPath()**
> – These methods set or get the path to which the cookie applies.
> – To specify that a cookie apply to all URLs on your site, use cookie.setPath("/").

> **public void setSecure(boolean secureFlag)**
> **public boolean getSecure()**
> – These method indicating whether the cookie should only be sent over encrypted (i.e., SSL) connections.
> **public void setValue(String cookieValue)**
> **public String getValue()**
> – The setValue method specifies the value associated with the cookie; getValue looks it up.
> **public void setVersion(int version)**
> **public int getVersion()**
> – These methods set and get the cookie protocol version with which the cookie complies. Version 0, the default, follows the original Netscape specification

---

USING COOKIES TO REMEMBER USER PREFERENCES

> One of the most common applications of cookies is to use them to "remember" user preferences.
> For Example **Registration.java, regservlet.java**

> **Using CookieUtility.**
> CookieUtility.java, CUUse.java

USE COOKIE S TO COUNT NO OF VISIT

> **Use can use CookieUtility class generated previously.**
> String ct;
> ct=CookieUtility.getCookieValue(req,"ac","0");
> int c=Integer.parseInt(ct) +1 ;
> CookieUtility.createCookie(res,"ac",String.value Of(c),0);
> out.println("No of Visit : " + c);

THANK YOU