

Servlet Cookie Example Tutorial

Cookies are used a lot in web client-server communication, it's not something specific to java.

Some of the common usage of cookies are:

1. Session authentication using Cookies, we learned in [Servlet Session Tutorial](#) that HttpSession uses "JSESSIONID" cookie to keep track of the user session.
2. Personalized response to the client based on their preference, for example we can set background color as cookie in client browser and then use it to customize response background color, image etc.

Cookie

Cookies are text data sent by server to the client and it gets saved at the client local machine. When client send request to server, it passes the cookies stored by the server in request header like below:

```
1 | Cookie Test="Test Cookie5"
```

Client can send multiple cookies to server and we can disable cookies to get stored at client side from browser preferences. Apart from the key-value pairs, server sends some other data to client in response header and it looks something like below.

```
1 | Set-Cookie Counter=7;  
2 | Version=1;  
3 | Comment="SetCookie Counter";  
4 | Domain="localhost";  
5 | Max-Age=86400;  
6 | Expires=Thu, 15-Aug-2013 20:19:19 GMT;  
7 | Path=/cookie/SetCookie  
8 |  
9 | Set-Cookie Test="Test Cookie7";  
10 | Version=1;  
11 | Comment="Test Cookie"
```

Note that server sends some additional information for cookie, such as comment, domain, maximum time before cookie expires and Path where browser should send the cookie back in request. But when client sends cookie to browser, it only sends the name and value of the cookie.

Servlet API provides cookies support through `javax.servlet.http.Cookie` class that implements Serializable and Cloneable interfaces.

HttpServletRequest `getCookies()` method is provided to get the array of Cookies from request, since there is no point of adding Cookie to request, there are no methods to set or add cookie to

request.

Similarly `HttpServletResponse addCookie(Cookie c)` method is provided to attach cookie in response header, there are no getter methods for cookie.

Cookie class has a single constructor that takes name and value because they are mandatory parameters for a cookie, all other parameters are optional.

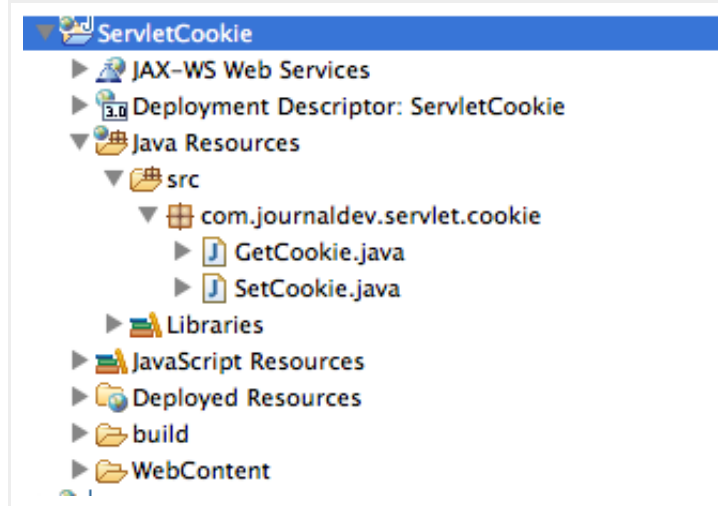
Some important methods of Cookie class are:

1. **getComment()** – Returns the comment describing the purpose of this cookie, used at client side. Note that server doesn't receive this information when client sends cookie in request header. We can use **setComment()** method to set cookie description at server side.
2. **getDomain()** – returns the domain name for the cookie. We can use **setDomain()** method to set the domain name for cookie, if domain name is set then the cookie will be sent only to that particular domain requests.
3. **getMaxAge()** – returns the maximum age in seconds. We can use **setMaxAge()** to set the expiration time of cookie.
4. **getName()** – returns the name of the cookie, can be used at both browser and server side. There is no setter for name, we can set name once through constructor only.
5. **getPath()** – Returns the path on the server to which the browser returns this cookie. We will see it's example where the cookie will be sent to specific resource only. We can use **setPath()** to instruct browser to send cookie to a particular resource only.
6. **getSecure()** – Returns true if the browser is sending cookies only over a secure protocol, or false if the browser can send cookies using any protocol. We can use **setSecure()** method to instruct browser to send cookie only over secured protocol.
7. **getValue()** – returns the value of the cookie as String. There is also **setValue()** method to change the value of cookie.
8. **getVersion()** – Returns the version of the protocol this cookie complies with. There is also a setter method for version.
9. **isHttpOnly()** – Checks whether this Cookie has been marked as HttpOnly. There is also a setter method that we can use to instruct client to use it for HTTP only.

Servlet Cookie Example

We will create two simple servlets to print cookies from client, in one of the servlet we will set a cookie for every domain and a cookie with Path settings so that other servlet won't receive this from client.

The project structure will look like below image.



SetCookie.java: This servlet will set some cookies and send it to browser. It will also print cookie information and send it as HTML response.

SetCookie.java

```
1  package com.journaldev.servlet.cookie;
2
3  import java.io.IOException;
4  import java.io.PrintWriter;
5
6  import javax.servlet.ServletException;
7  import javax.servlet.annotation.WebServlet;
8  import javax.servlet.http.Cookie;
9  import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12
13 @WebServlet("/cookie/SetCookie")
14 public class SetCookie extends HttpServlet {
15     private static final long serialVersionUID = 1L;
16     private static int count = 0;
17
18     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
19         PrintWriter out = response.getWriter();
20         Cookie[] requestCookies = request.getCookies();
21
22         out.write("<html><head></head><body>");
23         out.write("<h3>Hello Browser!!</h3>");
24         if(requestCookies != null){
25             out.write("<h3>Request Cookies:</h3>");
26             for(Cookie c : requestCookies){
27                 out.write("Name="+c.getName()+", Value="+c.getValue()+", Comment='");
28                 out.write(c.getComment()+", Domain="+c.getDomain()+", MaxAge="+c.getMaxAge()+", Path="+c.getPath()+", Version="+c.getVersion());
29                 out.write("<br>");
30             }
31         }
32         //Set cookies for counter, accessible to only this servlet
33         count++;
34         Cookie counterCookie = new Cookie("Counter", String.valueOf(count));
35         //add some description to be viewed in browser cookie viewer
36         counterCookie.setComment("SetCookie Counter");
37         //setting max age to be 1 day
38         counterCookie.setMaxAge(24*60*60);
39         //set path to make it accessible to only this servlet
40         counterCookie.setPath("/ServletCookie/cookie/SetCookie");
41     }
42 }
```

```

43 //adding cookie to the response
44 response.addCookie(counterCookie);
45
46 //set a domain specific cookie
47 Cookie domainCookie = new Cookie("Test", "Test Cookie"+String.valueOf(
48 domainCookie.setComment("Test Cookie");
49 response.addCookie(domainCookie);
50
51 out.write("</body></html>");
52 }
53
54 }

```

GetCookie.java: A simple servlet that will demonstrate that the cookie set in SetCookie with specific Path will not be send by browser to this servlet.

GetCookie.java

```

1  package com.journaldev.servlet.cookie;
2
3  import java.io.IOException;
4  import java.io.PrintWriter;
5
6  import javax.servlet.ServletException;
7  import javax.servlet.annotation.WebServlet;
8  import javax.servlet.http.Cookie;
9  import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12
13 @WebServlet("/cookie/GetCookie")
14 public class GetCookie extends HttpServlet {
15     private static final long serialVersionUID = 1L;
16
17     protected void doGet(HttpServletRequest request, HttpServletResponse response) {
18         PrintWriter out = response.getWriter();
19         Cookie[] requestCookies = request.getCookies();
20
21         out.write("<html><head></head><body>");
22         out.write("<h3>Hello Browser!!</h3>");
23         if(requestCookies != null){
24             out.write("<h3>Request Cookies:</h3>");
25             for(Cookie c : requestCookies){
26                 out.write("Name="+c.getName()+", Value="+c.getValue()+", Comment='
27                     +", Domain="+c.getDomain()+", MaxAge="+c.getMaxAge()+", P
28                     +", Version="+c.getVersion());
29                 out.write("<br>");
30                 //delete cookie
31                 if(c.getName().equals("Test")){
32                     c.setMaxAge(0);
33                     response.addCookie(c);
34                 }
35             }
36         }
37         out.write("</body></html>");
38     }
39
40 }

```

When you will run the program, you will notice few things:

- Cookie “Counter” is sent over to the SetCookie only, GetCookie will never receive this cookie.
- Except name and value all other variables are printing default values. MaxAge default value is -1 and version default value is 0.
- GetCookie is setting max age of “Test” cookie to 0, so that it will be expired and deleted by client browser.

That’s all for cookies and it’s usage in Servlet API, you might want to check out other servlet tutorials too.

1. [Java Web Application](#)
2. [Java Servlets](#)
3. [Servlet Session Management](#)
4. [Servlet Filter](#)
5. [Servlet Listener](#)



Download Servlet Cookie Example Project

602 downloads

Check out next articles in series about [Servlet Exception Handling](#) and [File Upload/Download Example](#).
