**16.1 Overview of JSP Cookies Handling**

Cookies are small textual that is sent from server to client and then client sends back this information to same server with all subsequent requests.

There are two types of cookies:

**· Session cookies** - are temporary cookies and are deleted as soon as user closes the browser. The next time user visits the same website, server will treat it as a new client as cookies are already deleted.

**· Persistent cookies** - remains on hard drive until we delete them or they expire.

There are several benefits and usage of Cookies.

**a) Remember Username and Password –** Several websites provides an auto login feature (Remember password) on private systems. This feature is implemented with the help of cookies.

**b) Remember Preferences-** Several sites uses cookies for user preferences.

**c) Advertising** – Several sites utilizes cookies to store the information of interested topics of user and later uses the information to display advertise.

Browsers provides options which user can use to delete or disable the cookies.

In this chapter we will discuss the how to send and read the cookies.

**16.2 JSP Cookies API**

JSP Specification provides a class Cookie in javax.servlet.http package .

Cookie class provides a two argument constructor (name and value of cookie) like below [**Cookie**](http://www.wideskills.com/jsp/jsp-cookies-handling#Cookie%28java.lang.String,%20java.lang.String%29)(String name , String value)

Following are the commonly used methods available in Cookie class.

**· public void setMaxAge(int expiry)**- This method sets maximum age ( in seconds )of cookie .If you don't set this, the cookie will last only for the current session.

**· public int getMaxAge()**- This method returns the maximum age of the cookie, specified in seconds, By default, -1 indicating the cookie will persist until browser shutdown.

**· public String getName()**- This method returns the name of the cookie.

**· public String getValue()**- This method gets the value of the cookie.

**· public void setValue(String value)**- This method sets the value of the cookie.

**· public void setComment(String comment)**-This method specifies a comment that describes a cookie's purpose.

**· public String getComment()**- This method returns the comment describing the purpose of this cookie, or null if the cookie has no comment.

HttpServletResponse class provides a method addCookie() to add the cookie in response and it is sent to the browser. Similarly HttpServletRequest class provides a method getCookies() to read the cookie sent by client along with request

**16.3 Restricted characters in cookies**

There are certain characters which cannot be used in name and value of a cookie. These characters are –

· , (Comma)

· = (equals)

· (

· )

· "

· ; (semi colon)

· [

· ]

· /

· ?

· : (colon)

· @

**16.4 Sending Cookies to Client**

Sending cookie to client is a three step process.

a) Create a cookie object using two argument constructor

b) You can call setMaxAge() method on cookie object created in #a **if you want to make the cookie persistent.**

c) Add the cookie using addCookie() method provided by HttpServletResponse object

Lets write an sendCookie.jsp which will send two cookies to client

[?](http://www.wideskills.com/jsp/jsp-cookies-handling)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | <!DOCTYPE html>  <html>    <head>      <title>Send Cookie Example</title>    </head>    <body>      <H1>JSP sending two cookies to client </H1>      <%        Cookie userIdCookie= new Cookie("userID", "guest");        userIdCookie.setMaxAge(24\*60\*60);        Cookie createdBy= new Cookie("createdBy","sendCookieJSP" );        createdBy.setMaxAge(24\*60\*60);        response.addCookie(userIdCookie);        response.addCookie(createdBy);      %>    </body>  </html> |

Now access sendCookie.jsp using [**http://localhost:8080/jsp-tutorial/sendCookie.jsp**](http://localhost:8080/jsp-tutorial/sendCookie.jsp) in Chrome browser

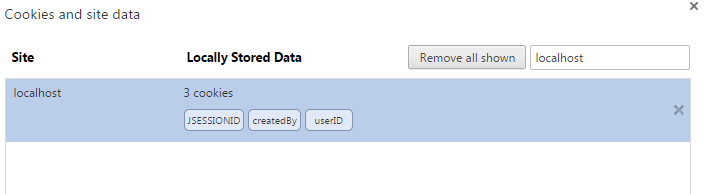
Reason I am saying Chrome because we can see the cookies sent by server to client using Google Chrome easily.



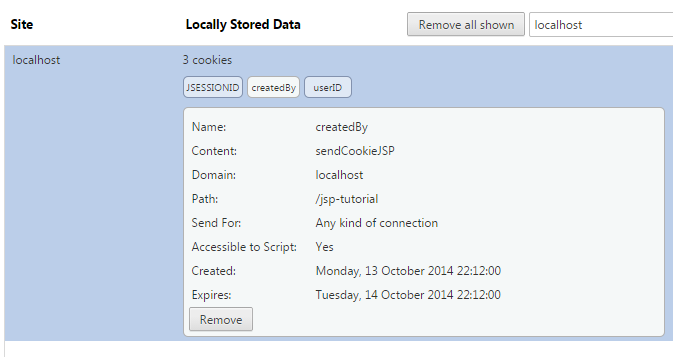
To see the cookies sent by browser, Go to

Settings à Show Advanced Settings à Privacy (Content Settings ) à All cookies and site data ...

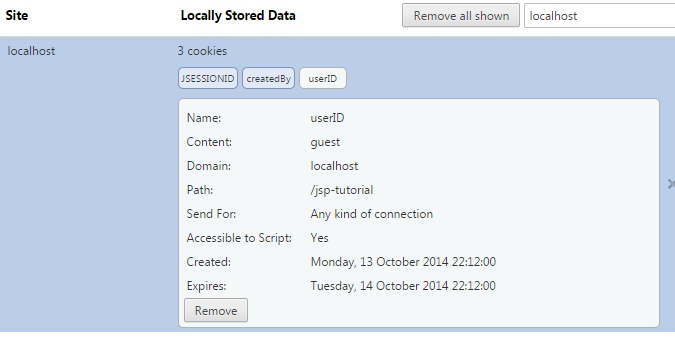
Search for localhost and you will something like below .Here we can see the cookies sent by sendCookie.jsp . JSESSIONID is special cookie which is send by server to identify the user



To see the details , click on createdBy cookie to see the details



We can see the value sent in cookie against Content label. Similarly we can see for userID cookie



**16.5 Reading Cookies**

Reading Cookies sent by client in request is

a) Get an array of Cookies using getCookies() method of HttpServletRequest

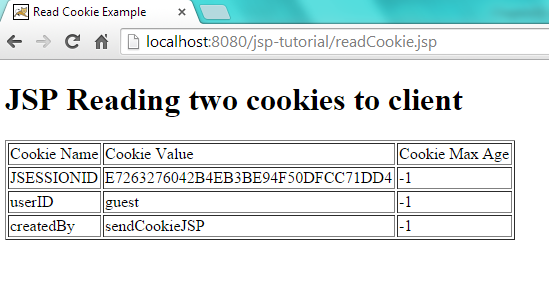
b) Run a loop and call getName() , getValue() and getMaxAge() to see the details of cookie is sent by client.

Lets write a readCookie.jsp which will read the cookie sent by server in section 16.4

[?](http://www.wideskills.com/jsp/jsp-cookies-handling)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | <html>    <head>      <title>Read Cookie Example</title>    </head>    <body>      <H1>JSP Reading two cookies to client </H1>      <table border=”1”>       <tr>        <td>Cookie Name </td>        <td>Cookie Value </td>        <td>Cookie Max Age  </td>       </tr>       <%         Cookie[] cookies= request.getCookies();         // check null because there are chances that there are no cookies         if(cookies !=null)         {           for(int i=0 ;i<cookies.length;i++ )           {             Cookie cookie = cookies[i];             out.println(" <tr> ");             out.println("<td>" + cookie.getName() + "</td>" );             out.println("<td>" + cookie.getValue() + "</td>" );             out.println("<td>" + cookie.getMaxAge() + "</td>" );             out.println(" </tr> ");           }         }       %>      </table>    </body>  </html> |

Now access readCookie.jsp using [**http://localhost:8080/jsp-tutorial/readCookie.jsp**](http://localhost:8080/jsp-tutorial/readCookie.jsp)



You can see the cookies sent by sendCookie.jsp. This is what we discussed earlier that browser send the cookies sent by server back to server with each subsequent request.

**16.6 Delete Cookies**

There is no direct API which can be used to delete any cookie. In case we need to delete existing cookie , we can delete it indirectly by setting the maximum age to 0 and add it back to response.

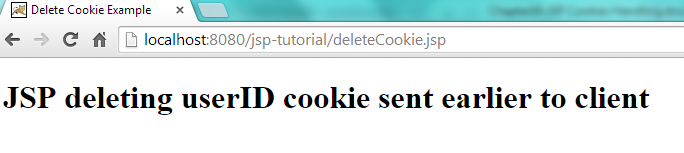
Lets delete the one of the cookie that we sent in section 16.4 and verify if the cookie is deleted or not using readCookie.jsp

To do so let’s write a**deleteCookie.jsp**

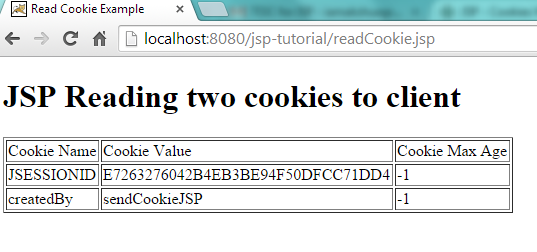
[?](http://www.wideskills.com/jsp/jsp-cookies-handling)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | <!DOCTYPE html>  <html>    <head>      <title>Delete Cookie Example</title>    </head>    <body>      <H1>JSP deleting  userID cookie sent earlier  to client </H1>      <%       Cookie[] cookies= request.getCookies();       //check null because there are chances that there are no cookies       if(cookies !=null)       {         for(int i=0 ;i<cookies.length;i++ )         {           Cookie cookie = cookies[i];           if(cookie.getName().equals("userID"))           {           cookie.setMaxAge(0);              response.addCookie(cookie);           }         }        }      %>    </body>  </html> |

Now access deleteCookie.jsp using [**http://localhost:8080/jsp-tutorial/deleteCookie.jsp**](http://localhost:8080/jsp-tutorial/deleteCookie.jsp)



To verify the cookie is deleted or not , again access readCookie.jsp using [**http://localhost:8080/jsp-tutorial/readCookie.jsp**](http://localhost:8080/jsp-tutorial/readCookie.jsp) and we can see only two cookies this time.



**16.7 Remember Username and Password Functionality**

As mentioned earlier , cookies can be used to achieve remember username and password features. You must have seen this feature on several websites.

Lets create a small example

a) Create RememberMe.jsp which will have username and password fields with a Remember Me check box .This jsp will check for a cookies and if found , it will set the value of fields with the cookie values.

[?](http://www.wideskills.com/jsp/jsp-cookies-handling)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | <html>    <head>     <title>Login Form</title>    </head>    <body>    <%      Cookie[] cookies = request.getCookies();      String username="";      String password = "";      if(cookies!=null)      {        for(int i=0;i<cookies.length;i++){          Cookie cookie = cookies[i];          if(cookie.getName().equals("username-cookie"))          {              username= cookie.getValue();          }          else if(cookie.getName().equals("password-cookie"))          {              password= cookie.getValue();          }        }      }     %>     <form name="logonform" action="displayHomePage.jsp" method="POST">        Username: <input type="text" name="username" value ="<%= username %>"/>        <br/>        Password:<input type="password" name="password" value="<%= password %>"/>        <br/>        Remember Me<input type="checkbox" name="rememberMe" value ="true"/>  <input type="submit" value="Submit"/>  </form>  </body>  </html> |

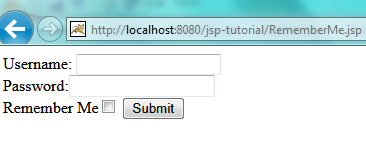
b) Create displayHomePage.jsp which will display username and password .Also if the user has checked the “Remember Me” check box , this jsp will add username and password as cookie.

[?](http://www.wideskills.com/jsp/jsp-cookies-handling)

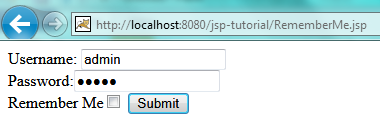
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | <html>  <head>  <title>Display Details</title>  </head>  <body>      <%      String username=request.getParameter("username");      String password=request.getParameter("password");  String message="Username is : "+ username + "<br/> Password is :" + password ;      String rememberMe=  request.getParameter("rememberMe");      if(rememberMe!=null)      {          Cookie usernameCookie = new Cookie("username-cookie", username);          Cookie passwordCookie = new Cookie("password-cookie", username);          usernameCookie.setMaxAge(24\*60\*60);          passwordCookie.setMaxAge(24\*60\*60);          response.addCookie(usernameCookie);          response.addCookie(passwordCookie);          }      %>      <strong>      <%= message %>      </strong>  </body>  </html> |

**Testing**

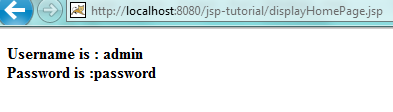
a) Access RememberMe.jsp using [**http://localhost:8080/jsp-tutorial/RememberMe.jsp**](http://localhost:8080/jsp-tutorial/RememberMe.jsp)



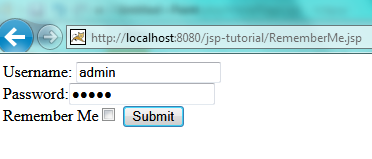
Enter username and password and do not check the “Remember Me” check box and click submit



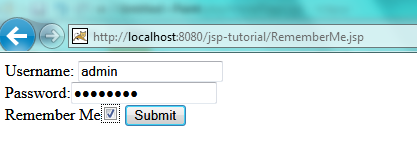
b) On submit below screen will be displayed . As “ Remember Me “ check box is not selected , cookies will not be added.



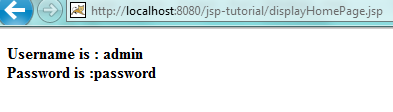
c) Again Access RememberMe.jsp using [**http://localhost:8080/jsp-tutorial/RememberMe.jsp**](http://localhost:8080/jsp-tutorial/RememberMe.jsp) .**This time username and password field will not be auto populated because we did not check the “Remember Me ” check box in #a**



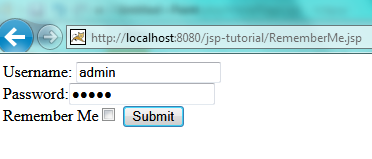
Enter username and password and check the “Remember Me” check box and click submit



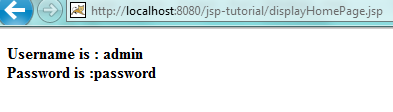
d) This time displayHomePage.jsp will add the cookies because “Remember Me ” is checked. On submit you will see below screen



e) Again Access RememberMe.jsp using [**http://localhost:8080/jsp-tutorial/RememberMe.jsp**](http://localhost:8080/jsp-tutorial/RememberMe.jsp) . This time username and password field will be auto populated because in #d , values were added in cookie



f) To verify if the values of username and password are correct or not , click submit



**To Load Results From Database On Page Scroll it takes only Three steps:-**

1. Make a PHP file and define markup and script to Load Results
2. Make a PHP file to get and send results from database
3. Make a CSS file and define styling for Load Results System

[**Extra Theme**](http://www.elegantthemes.com/affiliates/idevaffiliate.php?id=39665&url=31957)

[[](http://www.elegantthemes.com/affiliates/idevaffiliate.php?id=39665&url=31957)Extra takes the Divi Builder framework and extends its advanced Drag & Drop builder power to work on your homepage and categories with a brand new set of post-based modules.. Click Here For More Details](http://www.elegantthemes.com/affiliates/idevaffiliate.php?id=39665&url=31957)

**Step 1.Make a PHP file and define markup and script to Load Results**

We make a PHP file and save it with a name **load.php**

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" type="text/css" href="load\_style.css">

<script type="text/javascript" src="jquery.js"></script>

<script type="text/javascript">

$(window).scroll(function ()

{

if($(document).height() <= $(window).scrollTop() + $(window).height())

{

loadmore();

}

});

function loadmore()

{

var val = document.getElementById("row\_no").value;

$.ajax({

type: 'post',

url: 'get\_results.php',

data: {

getresult:val

},

success: function (response) {

var content = document.getElementById("all\_rows");

content.innerHTML = content.innerHTML+response;

// We increase the value by 10 because we limit the results by 10

document.getElementById("row\_no").value = Number(val)+10;

}

});

}

</script>

</head>

<body>

<h1>Load Results From Database On Page Scroll Using jQuery,Ajax And PHP</h1>

<div id="all\_rows">

<?php

mysql\_connect('localhost','root','');

mysql\_select\_db('demo');

$select=mysql\_query("select comment from sample\_comment limit 0,10");

while($row=mysql\_fetch\_array($select))

{

echo "<p class='rows'>".$row['comment']."</p>";

}

?>

</div>

<input type="hidden" id="row\_no" value="10">

</body>

</html>

In this step we fetch the results from our sample comments table in demo database you can use any data to display we want only 10 results at a time.And when the user scroll down the page and reached at the bottom then function is called which send the ajax request to get\_results.php and fetch the next 10 comments and then display on user screen with previous one.You may also like [load data from database without page refresh using ajax and jQuery](http://talkerscode.com/webtricks/load%20data%20from%20database%20without%20page%20refresh%20using%20ajax%20and%20jquery.php).

**Step 2.Make a PHP file to get and send results from database**

We make a PHP file named get\_results.php

<?php

if(isset($\_POST['getresult']))

{

mysql\_connect('localhost','root','');

mysql\_select\_db('demo');

$no = $\_POST['getresult'];

$select = mysql\_query("select comment from sample\_comment limit $no,10");

while($row = mysql\_fetch\_array($select))

{

echo "<p class='rows'>".$row['comment']."</p>";

}

exit();

}

?>

In this step we get the intial limit send by the ajax request which is used in query to get the further 10 results from database and then send back the results as a response to ajax request.You can also view our [move elements on page scroll using jQuery](http://talkerscode.com/webtricks/move-elements-on-page-scroll-using-jquery-and-css.php) tutorial.

**Step 3.Make a CSS file and define styling for Load Results System**

We make a CSS file and save it with name **load\_style.css**.

body

{

font-family:helvetica;

background-color:#58ACFA;

width:100%;

}

h1

{

text-align:center;

font-size:35px;

margin-top:50px;

color:#0B173B;

}

.rows

{

background-color:#0B3861;

color:white;

padding:20px;

margin-top:40px;

font-size:20px;

}

Thats all, this is how to load results from database on page scroll using jQuery,Ajax and PHP.You can customize this code further as per your requirement. And please feel free to give comments on this tutorial.

/Save it as jquery.cookie.js

/\*!

\* jQuery Cookie Plugin v1.3.1

\* https://github.com/carhartl/jquery-cookie

\*

\* Copyright 2013 Klaus Hartl

\* Released under the MIT license

\*/

(function (factory) {

if (typeof define === 'function' && define.amd) {

// AMD. Register as anonymous module.

define(['jquery'], factory);

} else {

// Browser globals.

factory(jQuery);

}

}(function ($) {

var pluses = /\+/g;

function raw(s) {

return s;

}

function decoded(s) {

return decodeURIComponent(s.replace(pluses, ' '));

}

function converted(s) {

if (s.indexOf('"') === 0) {

// This is a quoted cookie as according to RFC2068, unescape

s = s.slice(1, -1).replace(/\\"/g, '"').replace(/\\\\/g, '\\');

}

try {

return config.json ? JSON.parse(s) : s;

} catch(er) {}

}

var config = $.cookie = function (key, value, options) {

// write

if (value !== undefined) {

options = $.extend({}, config.defaults, options);

if (typeof options.expires === 'number') {

var days = options.expires, t = options.expires = new Date();

t.setDate(t.getDate() + days);

}

value = config.json ? JSON.stringify(value) : String(value);

return (document.cookie = [

config.raw ? key : encodeURIComponent(key),

'=',

config.raw ? value : encodeURIComponent(value),

options.expires ? '; expires=' + options.expires.toUTCString() : '', // use expires attribute, max-age is not supported by IE

options.path ? '; path=' + options.path : '',

options.domain ? '; domain=' + options.domain : '',

options.secure ? '; secure' : ''

].join(''));

}

// read

var decode = config.raw ? raw : decoded;

var cookies = document.cookie.split('; ');

var result = key ? undefined : {};

for (var i = 0, l = cookies.length; i < l; i++) {

var parts = cookies[i].split('=');

var name = decode(parts.shift());

var cookie = decode(parts.join('='));

if (key && key === name) {

result = converted(cookie);

break;

}

if (!key) {

result[name] = converted(cookie);

}

}

return result;

};

config.defaults = {};

$.removeCookie = function (key, options) {

if ($.cookie(key) !== undefined) {

// Must not alter options, thus extending a fresh object...

$.cookie(key, '', $.extend({}, options, { expires: -1 }));

return true;

}

return false;

};

}));

2. Add "Remember Me" check box on Login page

Add "Remember Me" check box in login form, then include "jquery.cookie.js" and your JS file (here: login.js).

<input type="text" id="username" /><br />

<input type="password" id="password" /><br />

<input type="checkbox" id="c1" />Remember Me<br />

<input type="submit" id="loginsubmit" />

.

.

.

<script src="js/jquery.cookie.js"></script>

<script src="js/login.js"></script>

3. Set and Get Cookie using JQuery

Now, set entered username on cookie, or display username if already avail on cookie. To do this,

include following line in your JS file (here: login.js)

$(document).ready(function()

{

// Please read these set of code after you read next set. So that you can understand simply.

var username=$.cookie("username"); // Get username from cookie on form load.

if(username!=undefined)

{

$('#loginsubmit').val(username); // Display username on input box if avail on cookie.

$('#c1').prop('checked', true); // Check check box manually by us. So that duration for current email will reset.

}

// Read this set first.

$('#loginsubmit').click(function()

{

if($('#c1').is(':checked')) // If user checked remember me check box

{

var email=$('#loginsubmit').val(); // Get entered username or email

// Set username on cookie when login form submit.

$.cookie("username", email, { expires: 365 }); // Remember username for 1 year.

}

// Your other codes...

}

});

Note:

1. Assign username or email to cookie after you done form validation like email validation, empty checking, so that you can sure for stored value are original.

2. And here we set username or email on cookie for one year. You don't need to worry about like "will expire after one year?". Because it will reset to one year whenever user try to login. And only go empty once if the user not login even one time for a year. You can change this duration as you need.

$(function() {

$projectKey = $('#projectKey');

$projectKey.change (

function() {

$.ajax({

type: "GET",

url: "getVersionsByProjectKey",

data: {"projectKey": $projectKey.val() },

dataType: 'json',

success: function(data){

$.each(data, function(index, value) {

$('#jiraVersion').append($('<option>').text(value).val(index));

});

}

}

});

}

);

});