9. Session Handling with Cookies

Contents

[1. Introduction 1](#_Toc12632)

[2. JSP cookies API 2](#_Toc6265)

[3. How to handle Cookies? 3](#_Toc349)

[4. Demo: Session Handling with Cookies 4](#_Toc4361)

# Introduction

=>slides: Pg. 1-2

##### ****When do we need to use Cookies?****

A cookie is a small object; it can be used to represent a single name-value pair and which will be maintained permanently at the client machine. In HttpSession Session Tracking Mechanism, all the client’s conversations will be maintained at the server machine in the form of HttpSession objects. In HttpSession Session Tracking Mechanism, if we increase the number of users then automatically a number of HttpSession objects will be created at the server. So that HttpSession Tracking Mechanism will increase the burden to the server machine. To overcome the above problem, we have to use an alternative mechanism, where we have to manage all the client’s conversations at the respective client machines only. To achieve the above requirement, we have to use Cookies Session Tracking Mechanism.

##### ****What are Cookies?****

Cookies are used to transfer the data between multiple form-based applications. Simply a cookie is a small piece of information sent by the server to any client. Every cookie will store the data in the form of key-value pair. The server will add the cookie to the response header by using the SetCookie header and send it to the client. Cookies will be stored in the browser by using the domain name of the website. The browser can contain cookies from multiple domains. The client needs to use HeaderCookie to get all the cookies available in the browser.

##### ****Types of Cookies in JSP****

1. ****Persistent Cookie:**** Persistent cookies means the cookie data will be available even though we close the browser up to a specified time. If we want to get persistent cookies we have to use the setMaxAge() method.
2. ****Non-Persistent Cookie:**** Non-persistent cookies means the cookie data will be deleted when we close the browser immediately. By default, we get non-persistent cookies.

##### ****How do the Cookies Work in JSP?****

Every cookie contains a name and allows one string value. Every cookie contains a cookie ID and also remembers the web application/domain name to which it belongs to. Server-side web resource programs of web applications create cookies and add them to the response to send to the client. So, cookies allocate memory as String information at the client-side. Cookies go back to the web application along with the request when the browser window gives a request back to the web application for which these cookies belong to. Cookies come to the client from the server/web application along with the response, as the values of the special response header called “set-cookie”. Cookies go back to the web application along with the request given by the browser window, as the values of the special request header called “cookie”. At the client-side or browser window, we can see multiple cookies belonging to different domains.

##### ****Advantages of Cookies:****

1. Compared to hidden variables cookies are better to transfer the data because once we store the cookie, we can get the data on any page of the website directly.
2. Cookies work with all server-side technologies and all servers.
3. Persistent cookies can remember client data even after the session having expiry time.

##### ****Disadvantages of Cookies:****

1. If the client disables the cookies in the browser we can work with cookies.
2. For security reasons, it is not recommended to use cookies.
3. If we want to a huge amount of data between server and client it will be complicated.

# JSP cookies API

=>slides: Pg. 3

##### ****Following are the cookies methods:****

##### **Public void setDomain(String domain)**

##### **This JSP set cookie is used to set the domain to which the cookie applies**

##### **Public String getDomain()**

##### **This JSP get cookie is used to get the domain to which cookie applies**

##### **Public void setMaxAge(int expiry)**

##### **It sets the maximum time which should apply till the cookie expires**

##### **Public intgetMaxAge()**

##### **It returns the maximum age of cookie in JSP**

##### **Public String getName()**

##### **It returns the name of the cookie**

##### **Public void setValue(String value)**

##### **Sets the value associated with the cookie**

##### **Public String getValue()**

##### **Get the value associated with the cookie**

##### **Public void setPath(String path)**

##### **This set cookie in JSP sets the path to which cookie applies**

##### **Public String getPath()**

##### **It gets the path to which the cookie applies**

##### **Public void setSecure(Boolean flag)**

##### **It should be sent over encrypted connections or not.**

##### **Public void setComment(String cmt)**

##### **It describes the cookie purpose**

##### **Public String getComment()**

##### **It the returns the cookie comments.**

# How to handle Cookies?

Handling cookies involves below three steps:

1. Creating a cookie object
2. Setting the maximum age of cookie
3. Sending cookie into the HTTP response headers

# Demo: Session Handling with Cookies

=>slides: Pg. 3

We will create login logout session code using stateless client based session cookie.

Creates a cookie, a small amount of information sent by a servlet to a Web browser, saved by the browser, and later sent back to the server. A cookie’s value can uniquely identify a client, so cookies are commonly used for session login logout management.

****Step 1:****

****login.html****

Step 2:

Create a Login Servlet

This servlet will

Authenticate the user (username: candidjava, password: candidjava)

create a new cookie object

redirect to the home page

If the username password is invalid it will redirect to login.html with an error message.

**Login.java**

**Step 3:**

**This page will be loaded after successful login, and it will automatically be logged out if the session time exceeds 10min or if the user deleted all cookie from the browser**

**home.jsp**

****Step 4:****

****logout.java****

****Step 5:****

****logout.html****

****Step 6:****

****sessionExpired.html****

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*