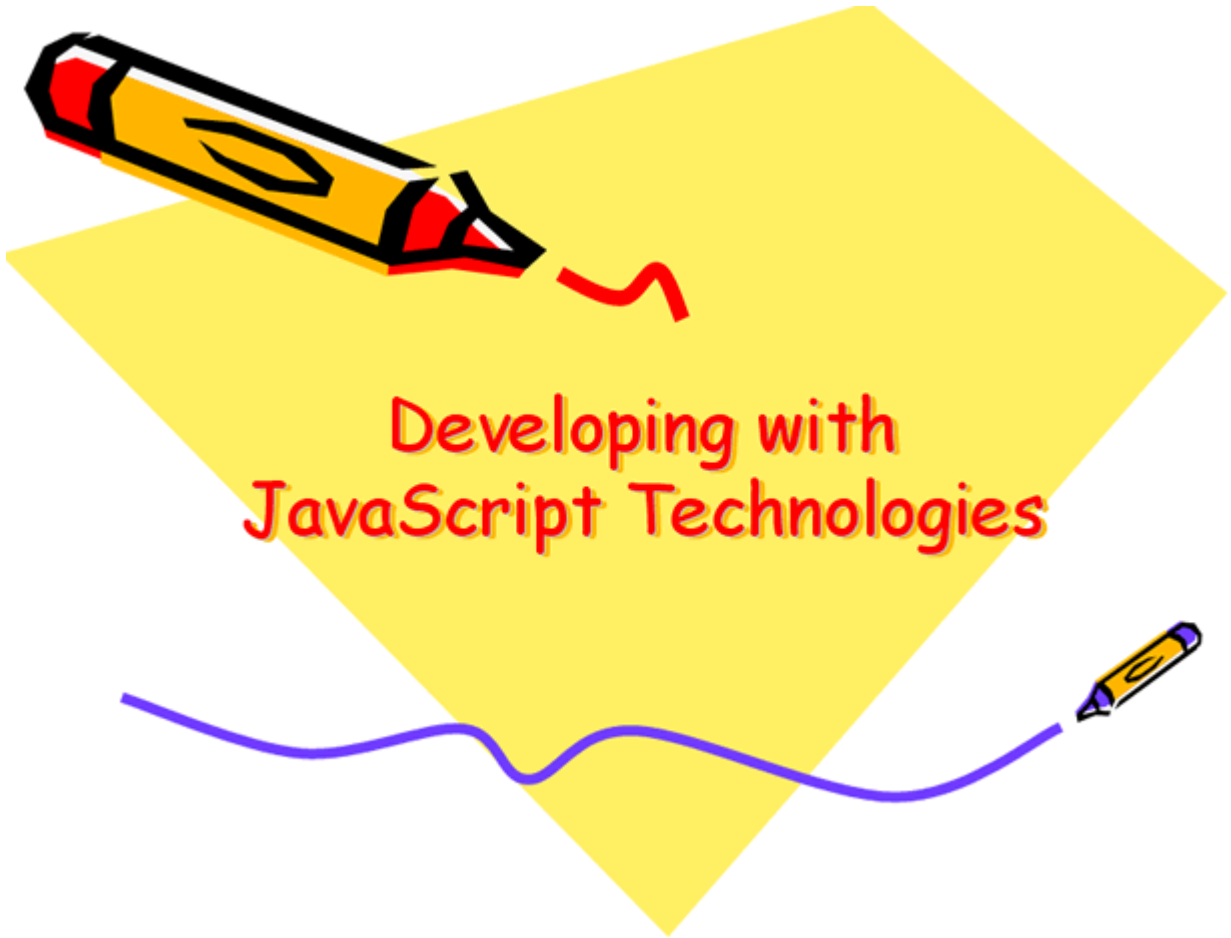
transentia

Exercise 7



Working with Cookies

Introduction

In this exercise you will learn how to write a small web application that utilises cookies to transmit state information between a server-side Java Server Page and a client-side JavaScript application.

This exercise contains a couple of 'pre-canned' scripts and files…you should take a bit of time to examine each one before you run it so that you can get a better idea of what is happening "behind the scenes."

Setting Up

This exercise involves both server-side and client-side software.

Installing the Software

There are a number of prerequisite pieces of software that need to be installed to establish a development framework for this exercise.

You will require the following:

- Sun JDK 1.3.1
- Tomcat 3.2.1

You have been given these on your CD-ROM.

Configuration

A bit of housekeeping is needed before development can begin.

Install the JDK

If you have not already done so, you should install the Sun Java Development Kit (this should be provided to you on your course CD-ROM). Note where it is installed.

Create and Populate the CookieStuff Home Directory

A (mostly) pre-installed and configured version of Tomcat has been provided for you on your CD-ROM. There remains a small bit of work, however.

Open a new Command Prompt window and issue the following command to copy the directory Z:\Exercises\7 CookieStuff\CookieStuff to C:\CookieStuff:

```
C:\> xcopy /ieq "Z:\Exercises\7 CookieStuff" C:\CookieStuff
```

Notes:

- This assumes that your CD-ROM device is Z:, you should use your real device letter instead

- Do not simply drag&drop from the CD-ROM; if you do, you will end up with a read-only directory structure and this will complicate things later on

You should end up with the following directory structure on your workstation's hard disk:

C:\CookieStuff
    bin
    CookieStuff
    jakarta-tomcat-3.2.1

Edit Utility Script

You have been provided with a useful utility script within C:\CookieStuff\bin. Using a text editor such as notepad, edit the supplied **setpath.bat**[1] file as follows: (for *path-to-jdk* use the place where you earlier installed the Java Development Kit):

```
@echo off
set JAVA_HOME=path-to-jdk
set PATH=%PATH%;%JAVA_HOME%\bin
set TOMCAT_HOME=C:\CookieStuff\jakarta-tomcat-3.2.1
```

This will make your life a lot easier later on.

A Simple Guessing Game

This example will show how simple cookies are to use.

There are two pieces to this game. First is a Java Server Page application that is written in Java and is used to generate a cookie containing a random number between 0 and 10 that is then passed off to the second piece: a simple HTML form containing JavaScript code. This code interacts with the user to pull data from a TextArea field in the HTML page and compare whatever was typed in with the value of the cookie that was set by the original JSP.

The game finishes if the user's guess matches the value of the cookie.

Create a Java Server Pages Application

A JSP is a plain HTML page with embedded Java code.

Edit the file **C:\CookieStuff\CookieStuff\CookieMaker.jsp** as shown below:

[1] If you are using notepad, be aware that notepad will append a '.txt' extension to anything it saves. To stop this behaviour you should enclose the full desired filename in double quote marks when you type it into notepad's "Save As…" dialog box.

```
<%@ page import="javax.servlet.http.*"
    contentType="text/html;charset=windows-1252" %>
<%!
final long max = 10;
final int age = 300;
%>
<html>
<head>
<title>Get your Cookies Here!</title>
<%
long random = Math.round(Math.random() * max);
Cookie thisCookie = new Cookie("CookieValue", random + "");
thisCookie.setMaxAge(-age);
response.addCookie(thisCookie);
%>
</head>
<body>
<h2>A Cookie has been set...</h2>
<p>A cookie with value between 0 and
  <%=max%>has been placed on your browser.</p>
<p>Follow this <a href="CookieGuesser.html">link</a>
to start guessing what it's value is.</p>
<p>Good luck!</p>
</body>
</html>
```

Points of interest:

- The cookie setting mechanism; the way in which the cookie is associated with the response to the client
- The various JSP code directives and sections (those delineated by <% and %>)

Create an HTML Form Page

Edit the file **C:\CookieStuff\CookieStuff\CookieGuesser.html** as shown below:

```
<html>
<head>
<title>Guess the Cookie Value...</title>
<script type="text/javascript"
        language="JavaScript">
function getCookie(name)
  {
  var start = document.cookie.indexOf(name+"=");
  var len = start+name.length+1;
  if ((!start) && (name != document.cookie.substring(0,name.length)))
    return null;
  if (start == -1)
    return null;
  var end = document.cookie.indexOf(";",len);
```

```
  if (end == -1)
    end = document.cookie.length;
  return unescape(document.cookie.substring(len,end));
  }

function guess()
  {
  var cookie = getCookie("CookieValue");
  var text = document.getElementById("TEXT").value;
  var res = document.getElementById("RESULT");
  res.innerText = text;
  if (cookie == text)
    {
    alert("Success! You guessed the cookie value: " + cookie);
    document.getElementById("BUTTON").disabled = true;
    document.getElementById("AGAIN").innerHTML =
      "Click here to <a href='CookieMaker.jsp'>Try again</a>.";
    }
  }
</script>
</head>
<body>
<h1>Guess the cookie value...</h1>
  <form>
    <input id="TEXT" type="textfield"><br/>
    <input id="BUTTON" type="submit" onClick="guess(); return false;">
  </form>
  <p>Last guess was:<span id="RESULT">NOTHING...</span>.</p>
  <div id="AGAIN"> </div>
</body>
</html>
```

Points of interest:

- The getCookie() function
- The use of the document.getElementById() function to manipulate the HTML document's DOM
- The installation of an onClick event handler; the way this handler returns 'false' to cancel any other possible page activities (reload, etc.)

Deployment

You will deploy this project to Tomcat in the form of a Web Archive. To do this, execute the following commands:

```
C:\CookieStuff\CookieStuff> ..\bin\setpath.bat
C:\CookieStuff\CookieStuff> jar cvf %TOMCAT_HOME%\webapps\Cookie.war .
```

Point of interest:

- Examine the web.xml file contained in the pre-supplied WEB-INF directory. It is this file that configures the archive and supplies information to Tomcat.

Execution Time

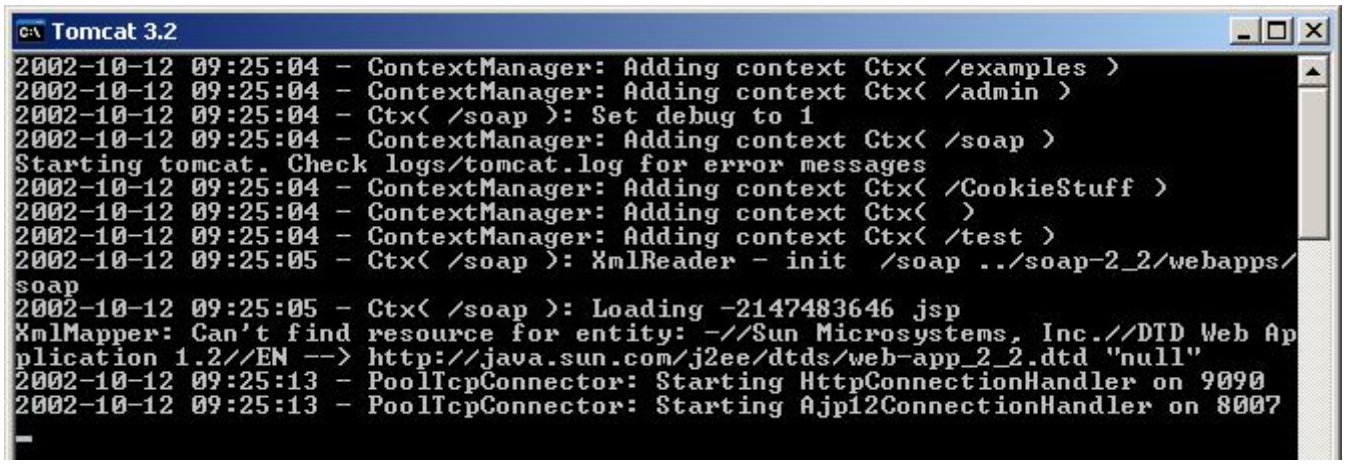
Time for all that hard work to pay off!

Start Tomcat

Ensure that you have Tomcat running:

```
C:\CookieStuff\CookieStuff> %TOMCAT_HOME%\bin\startup.bat
```

You should see a new Command Prompt window open for the Tomcat server process. Ensure that the window contains lines similar to what is shown in the following screendump. In particular, you should ensure that the CookieStuff context is deployed correctly:



Play the Game

It's a very simple game!

Point your browser at http://localhost:9090/Cookie/CookieMaker.jsp and play!

(Note the delay you experience prior to the JSP page's output being displayed in the browser window: this is caused by the required one-time 'on-the-fly' compilation of the JSP before it can be executed.)