## 实验五 会话管理

一、实验目的

1. 了解Web服务器对客户跟踪的各种方法；

2. 重点掌握使用HttpSession对象跟踪会话的方法；

3. 掌握使用Cookie技术跟踪会话的方法；

4. 了解URL重写和隐藏表单域的方法。

二、实验原理

HTTP协议是无状态的协议。在很多情况下，Web服务器必须能够跟踪客户的状态。比如，对于一个购物网站，在一个时刻可能有多个客户购物，Web服务器必须能够区分不同的客户。一般情况下，Web服务器为每个客户配置了虚拟的购物车（ShoppingCart）。当某个客户请求将一个商品放入购物车时，Web服务器必须根据发出请求的客户的身份，找到该客户的购物车，然后把商品放入其中。

Web服务器跟踪客户的状态通常有4种方法：

（1）使用HttpSession对象管理会话；（2）使用持久的Cookie对象；（3）使用URL重写机制；（4）使用隐藏的表单域。

三、实验内容与步骤

（一） 使用HttpSession对象管理会话。

package com.demo;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.util.Date;

public class ShowSession extends HttpServlet{

public void doGet(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

HttpSession session = request.getSession();

String heading =null;

Integer accessCount = (Integer)session.getAttribute("accessCount");

if(accessCount==null){

accessCount = new Integer(0);

heading = "Welcome, Newcomer";

}else{

heading = "Welcome Back.";

accessCount = new Integer(accessCount.intValue()+1);

}

session.setAttribute("accessCount",accessCount);

PrintWriter out = response.getWriter();

out.println("<html><head>");

out.println("<title>Session Tracking Example</title></head>");

out.println("<body><center>");

out.println("<h1>"+heading+"</h1>");

out.println("<h2>Information about your session.</h2>");

out.println("<table border='1'>");

out.println("<tr bgcolor=\"ffad00\"><th>Info Type<th>Value\n");

out.println("<tr><td>ID<td>"+session.getId()+"\n");

out.println("<tr><td>Creation Time:<td>");

out.println(""+new Date(session.getCreationTime())+"\n");

out.println("<tr><td>Time of last access:<td>");

out.println(""+new Date(session.getLastAccessedTime())+"\n");

out.println("<tr><td>Access number:<td>"+accessCount+"\n");

out.println("</table></center></body></html>");

}

}

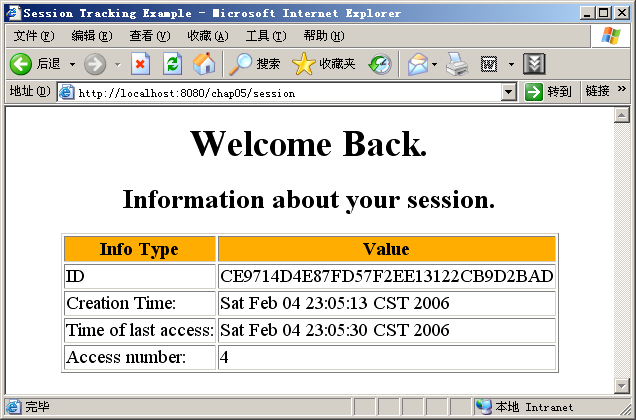


图5.1 ShowSession程序的运行结果

（二）使用Cookie跟踪会话

使用Cookie跟踪会话涉及到向客户端写入Cookie和从客户端读取Cookie。

【步骤1】向客户端写入Cookie对象

package test;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class SendCookie extends HttpServlet{

public void doGet(HttpServletRequest request,

HttpServletResponse response)

throws IOException,ServletException{

Cookie userCookie = new Cookie("user", "uid1234");

userCookie. setMaxAge(60\*60\*24\*7);

response.addCookie(userCookie);

response.setContentType("text/html");

PrintWriter out=response.getWriter();

out.println("<html><title>add cookies</title>");

out.println("<body><h2>"+

"A cookie has been sent to brower</h2></body>");

out.println("</html>");

}

}

访问上述Servlet将向客户端写入一个Cookie，执行后可查看客户机对应目录看是否写了一个文件。



图5.2 SendCookie程序的运行结果

【步骤2】从客户端读取Cookie对象

package test;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class GetCookie extends HttpServlet{

public void doGet(HttpServletRequest

request,HttpServletResponse response)

throws IOException,ServletException{

String cookieName = "user";

String cookieValue = null;

Cookie[] cookies = request.getCookies();

if (cookies!=null){

for(int i = 0;i<cookies.length;i++){

Cookie cookie = cookies[i];

if(cookie.getName().equals(cookieName))

cookieValue = cookie.getValue();

} }

response.setContentType("text/html");

PrintWriter out=response.getWriter();

out.println("<html><title>get cookies</title>");

out.println("<body><h2>A cookie has been got from brower</h2>");

out.println("CookieName:"+cookieName+"<br>");

out.println("CookieValue:"+cookieValue+"<br>");

out.println("</body></html>");

} }



图5.3 GetCookie程序的运行结果

四、思考题

1. 简述会话管理的常用方法有哪些？

2. 如何使用HttpSession对象共享数据？