



# cookie管理

JSP, Servlet, & Struts Training Courses: <http://courses.coreservlets.com>  
*Available in US, China, Taiwan, HK, and Worldwide*

JSP and Servlet Books from Sun Press: <http://www.coreservlets.com>  
*Available in English, Chinese (simplified and traditional script),  
and 12 other languages*

2

## 议程

- cookie的优点和缺点
- 输出cookie的发送
- 输入cookie的接收
- 跟踪重复用户
- cookie属性的指定
- 会话cookie和持续性cookie之间的差异
- 用实用工具类简化cookie的使用
- cookie值的修改
- 用户偏好的记录

3

# cookie的潜能

- 思想

- servlet发送简单的名称和值到客户程序。
- 客户程序在访问同一站点（或同一域，依cookie设置的不同而定）时将名称和值原样返回。

- cookie的典型应用

- 在电子商务会话中标识用户
  - 针对这项任务servlet拥有专门的高层API
- 避免存储用户名和密码
- 对站点进行定制
- 定向广告

4

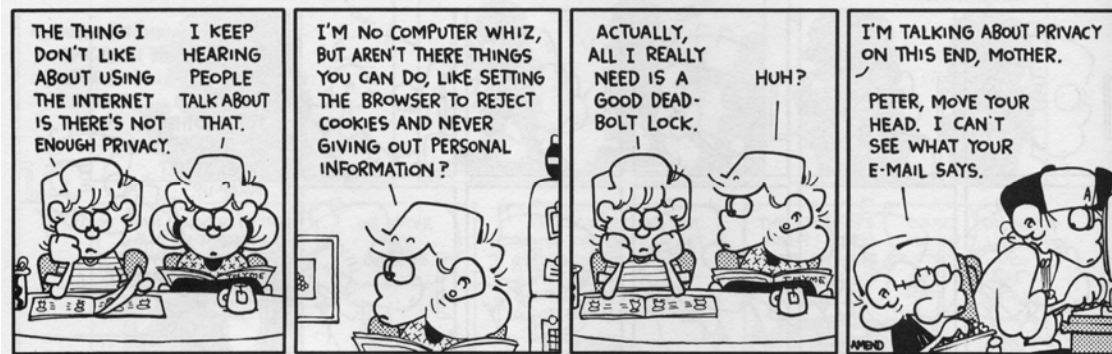
.JSP/servlet/Struts/JSF training: <http://www.coreservlets.com>

# cookie和定向广告

The image shows two overlapping browser windows from Microsoft Internet Explorer. The top window displays the AltaVista homepage with a search bar and navigation links. The bottom window shows search results for 'servlets jsp book' with a yellow advertisement for 'JavaServer Pages & Java Servlets HASSLE-FREE' highlighted by a red circle. Another red circle highlights the search bar in the bottom window containing the text 'servlets jsp book'.

5

## cookie和隐私



FoxTrot © 1998 Bill Amend. Reprinted with permission of Universal Press Syndicate. All rights reserved.

6

JSP/servlet/Struts/JSF training: <http://www.coreservlets.com>

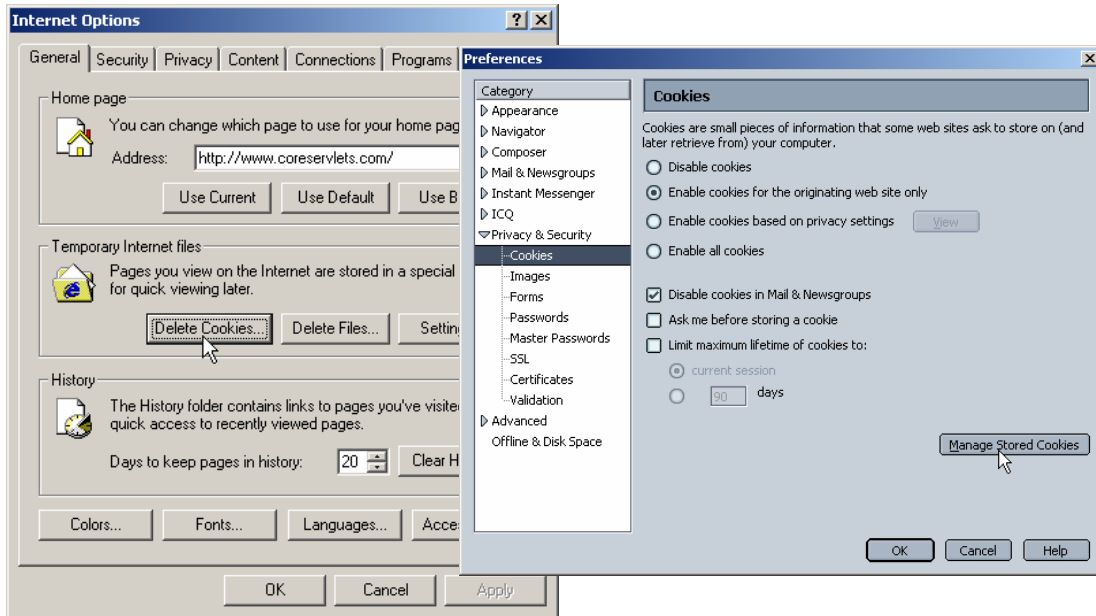
## 与cookie相关的一些问题

- 问题在于隐私，不是安全。
  - 服务器能够记住您之前的动作。
  - 如果您给出自己的个人信息，服务器能够将这些信息与您之前的动作关联起来。
  - 多个服务器能够通过提供协作服务的第三方，如 doubleclick.net，共享cookie信息。
  - 设计欠考虑的网站直接将敏感的信息，如信用卡号码，存储在cookie中。
  - JavaScript的bug能够使敌对网站窃取cookie (老浏览器)
- **servlet作者应该遵守下面这些法则**
  - 如果cookie对于您的任务并非至关重要，则要避免在cookie被禁用时servlet完全不能工作。
  - 不要将敏感的信息存储在cookie中。

7

JSP/servlet/Struts/JSF training: <http://www.coreservlets.com>

# 手动删除cookie (为使测试工作简化)



8

JSP/servlet/Struts/JSF training: <http://www.coreservlets.com>

## 向客户程序发送cookie

- **创建Cookie对象。**
  - 调用Cookie的构造函数，给出cookie的名称和cookie的值，二者都是字符串。

```
Cookie c = new Cookie("userID", "a1234");
```
- **设置最大时效。**
  - 如果要告诉浏览器将cookie存储到磁盘上，而非仅仅保存在内存中，使用setMaxAge (参数为秒数)。

```
c.setMaxAge(60*60*24*7); // One week
```
- **将Cookie放入到HTTP响应中**
  - 使用response.addCookie。
  - 如果忘记这一步，那么不会有任何cookie被发送到浏览器!

9

response.addCookie(qs) JSP/servlet/Struts/JSF training: <http://www.coreservlets.com>

## 从客户端读取cookie

- 调用`request.getCookies`
  - 这会得到Cookie对象组成的数组。
- 在这个数组中循环，调用每个对象的`getName`，直到找到想要的cookie为止。
  - 根据应用程序的具体情况使用这个值(`getValue`)。

```
String cookieName = "userID";
Cookie[] cookies = request.getCookies();
if (cookies != null) {
    for(int i=0; i<cookies.length; i++) {
        Cookie cookie = cookies[i];
        if (cookieName.equals(cookie.getName())) {
            doSomethingWith(cookie.getValue());
        }
    }
}
```

10

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## 使用cookie检测初访者

```
public class RepeatVisitor extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        boolean newbie = true;
        Cookie[] cookies = request.getCookies();
        if (cookies != null) {
            for(int i=0; i<cookies.length; i++) {
                Cookie c = cookies[i];
                if ((c.getName().equals("repeatVisitor")) &&
                    (c.getValue().equals("yes"))) {
                    newbie = false;
                    break;
                }
            }
        }
    }
}
```

11

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

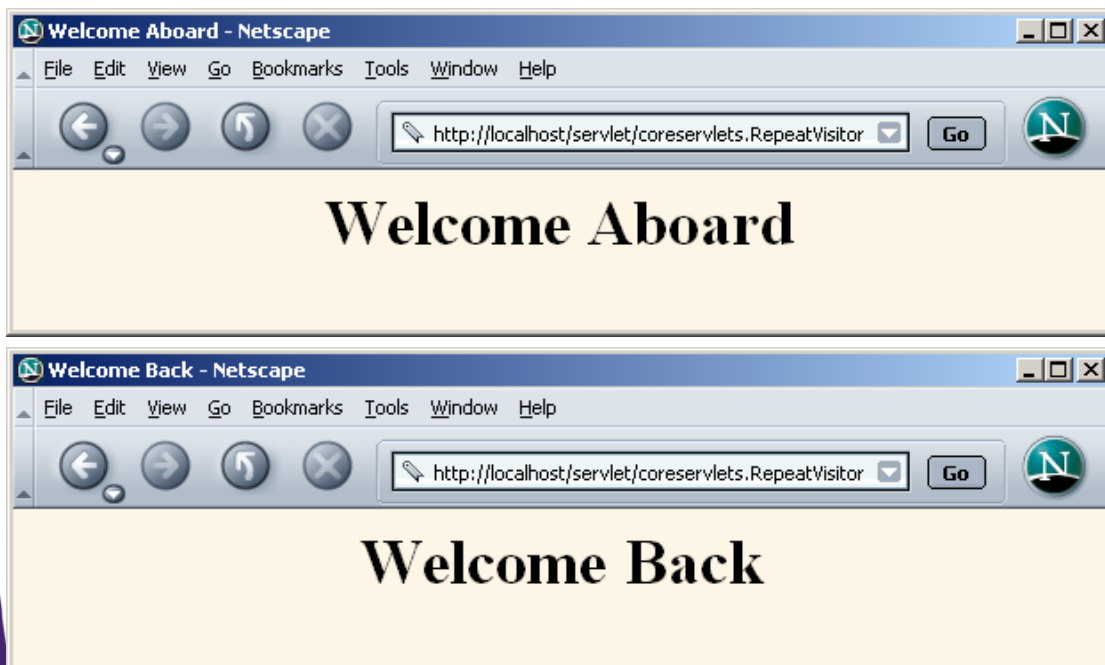
## 使用cookie检测初访者 ( 续 )

```
String title;
if (newbie) {
    Cookie returnVisitorCookie =
        new Cookie("repeatVisitor", "yes");
    returnVisitorCookie.setMaxAge(60*60*24*365);
    response.addCookie(returnVisitorCookie);
    title = "Welcome Aboard";
} else {
    title = "Welcome Back";
}
response.setContentType("text/html");
PrintWriter out = response.getWriter();
... // (Output page with above title)
```

12

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## 使用cookie检测初访者 ( 结果 )



13

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## 使用cookie的属性

- **getDomain/setDomain**
  - 允许我们指定cookie适用的域。当前的主机必须是所指定的域的一部分。
- **getMaxAge/setMaxAge**
  - 读取/设置cookie的到期时间(秒)。如果没有设置，则cookie只适用于当前的浏览会话。参见前面给出的LongLivedCookie辅助类。
- **getName**
  - 读取cookie的名称。不存在setName方法；因为我们在构造函数中提供了名称。对于输入cookie构成的数组来说，我们使用getName来找到感兴趣的cookie。

14

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## 使用cookie的属性

- **getPath/setPath**
  - 读取/设置cookie适用的路径。如果未指定，则cookie适用于含有当前页面的目录中的URL，以及该目录之下的URL。
- **getSecure/setSecure**
  - 读取/设置标志，标示cookie是否只应适用于SSL连接，或者适用于所有连接。
- **getValue/setValue**
  - 读取/设置与cookie关联的值。对于新的cookie，我们将值提供给构造函数，而非setValue。对于输入cookie组成的数组，我们使用getName找到感兴趣的cookie，然后调用所获得对象的getValue方法。如果设置了某个输入cookie的值，我们依旧需要用response.addCookie将它发送回去。

15

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## 区分会话cookie和持续性cookie

```
public class CookieTest extends HttpServlet {
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {
        for(int i=0; i<3; i++) {
            Cookie cookie =
                new Cookie("Session-Cookie-" + i,
                           "Cookie-Value-S" + i);
            // No maxAge (ie maxAge = -1)
            response.addCookie(cookie);
            cookie = new Cookie("Persistent-Cookie-" + i,
                                "Cookie-Value-P" + i);
            cookie.setMaxAge(3600);
            response.addCookie(cookie);
        }
    }
}
```

16

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## 区分会话cookie和持续性 cookie(Cont)

```
... // Start an HTML table
Cookie[] cookies = request.getCookies();
if (cookies == null) {
    out.println("<TR><TH COLSPAN=2>No cookies");
} else {
    Cookie cookie;
    for(int i=0; i<cookies.length; i++) {
        cookie = cookies[i];
        out.println
            ("<TR>\n" +
             " <TD>" + cookie.getName() + "\n" +
             " <TD>" + cookie.getValue());
    }
}
```

17

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## 区分会话cookie和持续性cookie

- 首次访问CookieTest servlet的结果
  - 访问该servlet，退出浏览器，等待一小时，然后再次访问这个servlet得到相同的结果。

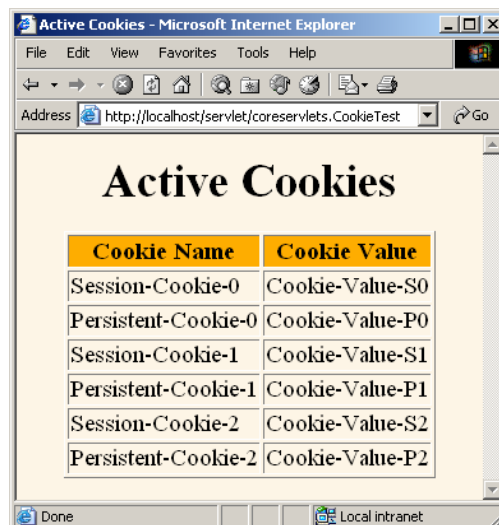


18

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## 区分会话cookie和持续性cookie

- 首次访问后一小时之内再次访问CookieTest的结果（使用同一浏览器会话）
  - 也就是说，初次访问和此处访问之间浏览器保持打开



19

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## 区分会话cookie和持续性cookie

- 首次访问后一小时之内再次访问CookieTest的结果（使用不同的浏览器会话）
  - 也就是说，浏览器在初次访问和此次访问之间重新启动过。



JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## 实用程序：查找指定名称的cookie

```
public class CookieUtilities {
    public static String getCookieValue
        (HttpServletRequest request,
         String cookieName,
         String defaultValue) {
        Cookie[] cookies = request.getCookies();
        if (cookies != null) {
            for(int i=0; i<cookies.length; i++) {
                Cookie cookie = cookies[i];
                if (cookieName.equals(cookie.getName())) {
                    return(cookie.getValue());
                }
            }
        }
        return(defaultValue);
    }
}
```

## 实用程序：创建长生存期的cookie

```
public class LongLivedCookie extends Cookie {
    public static final int SECONDS_PER_YEAR =
        60*60*24*365;

    public LongLivedCookie(String name, String value) {
        super(name, value);
        setMaxAge(SECONDS_PER_YEAR);
    }
}
```

22

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## 实用程序的使用: RepeatVisitor2

```
public class RepeatVisitor2 extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        boolean newbie = true;
        String value =
            CookieUtilities.getCookieValue(request,
                "repeatVisitor2",
                "no");

        if (value.equals("yes")) {
            newbie = false;
        }
        String title;
        if (newbie) {
            LongLivedCookie returnVisitorCookie =
                new LongLivedCookie("repeatVisitor2", "yes");
            response.addCookie(returnVisitorCookie);
            title = "Welcome Aboard";
        } else {
            title = "Welcome Back";
        }
    }
}
```

23

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## 修改cookie的值

- **替换cookie的值**
  - 使用不同的值发送相同的cookie名称。
  - 重用输入的cookie对象。
    - 需要调用response.addCookie；只是调用setValue是没用的。
    - 不需要通过调用setMaxAge，setPath等重新设置所有相关的cookie属性—输入的cookie中并不含有cookie的属性。
    - 一般不值得这样做，因而常使用新的Cookie对象。
- **指示浏览器删除某个cookie**
  - 使用setMaxAge将最大时效指定为0。

24

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## 记录用户的访问计数

```
public class ClientAccessCounts extends HttpServlet {
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {
        String countString =
            CookieUtilities.getCookieValue(request,
                                           "accessCount",
                                           "1");

        int count = 1;
        try {
            count = Integer.parseInt(countString);
        } catch (NumberFormatException nfe) { }
        LongLivedCookie c =
            new LongLivedCookie("accessCount",
                               String.valueOf(count+1));
        response.addCookie(c);
    }
}
```

25

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## 记录用户的访问计数 ( 续 )

```
...
out.println(docType +
    "<HTML>\n" +
    "<HEAD><TITLE>" + title +
    "</TITLE></HEAD>\n" +
    "<BODY BGCOLOR=\"#FDF5E6\">\n" +
    "<CENTER>\n" +
    "<H1>" + title + "</H1>\n" +
    "<H2>This is visit number " +
    count + " by this browser.</H2>\n"+
    "</CENTER></BODY></HTML>");
}
```

26

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## 记录用户的访问计数 ( 结果 )



27

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## 使用cookie记录用户的偏好

- **RegistrationForm servlet**
  - 使用cookie的值预先填写表单的字段
  - 如果未找到相应的cookie则使用默认值
  - 课程的后面将会在JSP中重做这项工作
- **Registration servlet**
  - 基于接收到的请求参数创建cookie
  - 如果得到所有的参数则显示这些值
  - 如果任何参数缺失，则重定向到表单

28

JSP/servlet/Struts/JSF training: <http://www.coreservlets.com>

## RegistrationForm Servlet

```
public class RegistrationForm extends HttpServlet {
    public void doGet(HttpServletRequest request,
                     HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String actionURL =
            "/servlet/coreservlets.RegistrationServlet";
        String firstName =
            CookieUtilities.getCookieValue(request,
                                           "firstName", "");
        String lastName =
            CookieUtilities.getCookieValue(request,
                                           "lastName", "");
        String emailAddress =
            CookieUtilities.getCookieValue(request,
                                           "emailAddress",
                                           "");
    }
}
```

29

JSP/servlet/Struts/JSF training: <http://www.coreservlets.com>

## RegistrationForm Servlet (续)

```
out.println
(docType +
 "<HTML>\n" +
 "<HEAD><TITLE>" + title + "</TITLE></HEAD>\n" +
 "<BODY BGCOLOR=\"#FDF5E6\"\>\n" +
 "<CENTER>\n" +
 "<H1>" + title + "</H1>\n" +
 "<FORM ACTION=\"" + actionURL + "\">\n" +
 "First Name:\n" +
 "  <INPUT TYPE=\"TEXT\" NAME=\"firstName\" " +
    "VALUE=\"" + firstName + "\"><BR>\n" +
 "Last Name:\n" +
 "  <INPUT TYPE=\"TEXT\" NAME=\"lastName\" " +
    "VALUE=\"" + lastName + "\"><BR>\n"+
 "Email Address: \n" +
 "  <INPUT TYPE=\"TEXT\" NAME=\"emailAddress\" " +
    "VALUE=\"" + emailAddress + "\"><P>\n" +
 "<INPUT TYPE=\"SUBMIT\" VALUE=\"Register\">\n" +
 "</FORM></CENTER></BODY></HTML>");
}
```

30

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## Registration Servlet

```
public class RegistrationServlet extends HttpServlet {
  public void doGet(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html");
    boolean isMissingValue = false;
    String firstName =
      request.getParameter("firstName");
    if (isMissing(firstName)) {
      firstName = "Missing first name";
      isMissingValue = true;
    }
    String lastName =
      request.getParameter("lastName");
    if (isMissing(lastName)) {
      lastName = "Missing last name";
      isMissingValue = true;
    }
  }
  ...
}
```

31

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## Registration Servlet ( 续 )

```
Cookie c1 =
    new LongLivedCookie("firstName", firstName);
response.addCookie(c1);
Cookie c2 =
    new LongLivedCookie("lastName", lastName);
response.addCookie(c2);
Cookie c3 = new LongLivedCookie("emailAddress",
                                emailAddress);

response.addCookie(c3);
String formAddress =
    "/servlet/coreservlets.RegistrationForm";
if (isMissingValue) {
    response.sendRedirect(formAddress);
} else { ... }
```

32

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

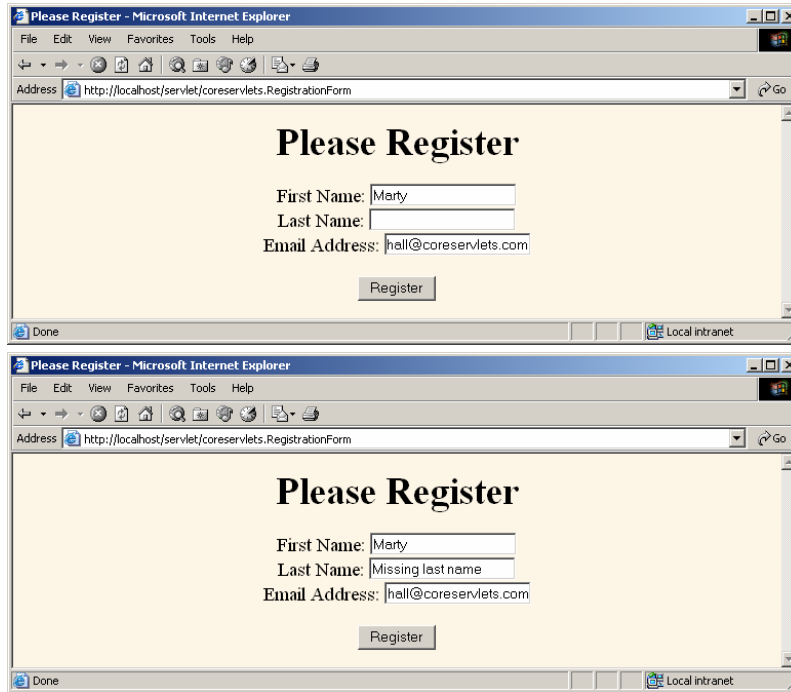
## RegistrationForm ( 最初结果 )



33

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

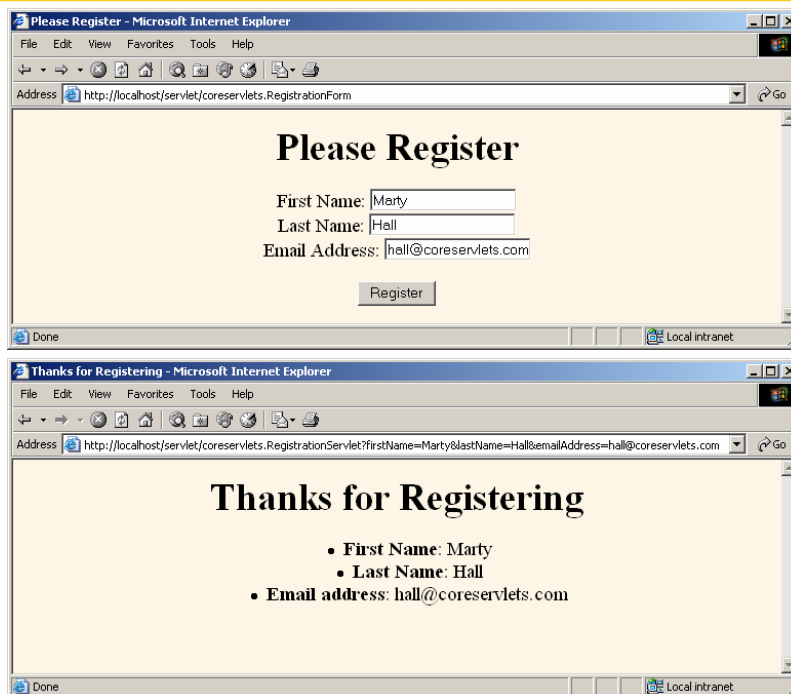
# RegistrationForm ( 提交不完整的表单 )



34

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

# RegistrationForm ( 提交完整的表单 )



35

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

# RegistrationForm ( 稍后访问得到的初始结果 )



36

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

## 小结

- **cookie**涉及将名称/值对从服务器发送到浏览器，并在之后访问相同的页面、站点或域时返回。
- 我们可以
  - 跟踪会话 ( 使用高层API )
  - 对安全性要求较低的网站，使用cookie可以避免用户每次都需要登录
  - 根据用户的不同定制网站
  - 有选择地投放内容或广告
- **设置cookie**
  - 调用Cookie的构造函数，设置时效，调用 `response.addCookie`
- **读取 cookie**
  - 调用 `request.getCookie`，检查得到的结果是否为null，在数组中查找匹配的名称，使用相关的值

37

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>



# 问题?

**JSP, Servlet, & Struts Training Courses: <http://courses.coreservlets.com>**  
*Available in US, China, Taiwan, HK, and Worldwide*

**JSP and Servlet Books from Sun Press: <http://www.coreservlets.com>**  
*Available in English, Chinese (simplified and traditional script),  
and 12 other languages*