# 尚马教育 JAVA WEB课程

# Jsp相关技术

文档编号：B14

创建日期： 2017-06-13

最后修改日期：2019-09-23

版 本 号：V2.0

电子版文件名：尚马教育-第二阶段-4.jsp相关技术.docx

**文档修改记录：**

|  |  |  |  |
| --- | --- | --- | --- |
| 更新日期 | 更新作者 | 更新说明 | 版本号 |
| 2018-08-01 | 王绍成 | 初始版本 | V1.0 |
| 2019-09-13 | 郭雪岩 | 内容更新 | V2.0 |

**主讲人：**

**郭雪岩**

目录

[1. 会话跟踪](#_Toc16836_WPSOffice_Level1) [3](#_Toc16836_WPSOffice_Level1)

[1.1. 什么叫会话跟踪](#_Toc14308_WPSOffice_Level2) [3](#_Toc14308_WPSOffice_Level2)

[1.2. 会话跟踪实现方式](#_Toc27358_WPSOffice_Level2) [3](#_Toc27358_WPSOffice_Level2)

[2. Cookie](#_Toc14308_WPSOffice_Level1) [4](#_Toc14308_WPSOffice_Level1)

[2.1. Cookie介绍](#_Toc22739_WPSOffice_Level2) [4](#_Toc22739_WPSOffice_Level2)

[2.2. Cookie特点](#_Toc7291_WPSOffice_Level2) [4](#_Toc7291_WPSOffice_Level2)

[2.3. Cookie主要属性](#_Toc604_WPSOffice_Level2) [5](#_Toc604_WPSOffice_Level2)

[2.4. Cookie操作（java部分）](#_Toc29150_WPSOffice_Level2) [5](#_Toc29150_WPSOffice_Level2)

[2.5. Cookie操作（js部分）](#_Toc25817_WPSOffice_Level2) [6](#_Toc25817_WPSOffice_Level2)

[3. Web storage使用](#_Toc27358_WPSOffice_Level1) [7](#_Toc27358_WPSOffice_Level1)

## 会话跟踪

### 什么叫会话跟踪

* 在无状态（一次请求响应就结束）的HTTP协议下,想要跟踪用户的路径（知道是谁，在干什么，要去哪）需要通过其他技术来辅助实现。

### 会话跟踪实现方式

* 常用的会话跟踪技术有四种
* URL方式：需要保存的信息直接追加到URL后，

例如：http://127.0.0.1:8080/chapter03/viewList?pageNo=12

* 隐藏域方式：可以使用表单中的隐藏域保存相关信息， 例如：

<input type="hidden" name=“status" value=“true">

* Session方式：将状态信息保存到服务器的会话对象中，通过唯一标记的ID值

与客户端进行绑定使用；例如访问控制功能就可以使用Session实现

* Cookie方式：将状态信息保存到客户端，服务器能够获得相关信息进行分析，

从而生成对客户端的响应；例如简化登录功能就可以使用Cookie实现；

## Cookie

### Cookie介绍

* Cookie译为小型文本文件或小甜饼，Web应用程序利用Cookie在客户端缓存服务器端文件。Cookie是以键值对形式存储在客户端主机硬盘中，由服务器端发送给客户端，客户端再下一次访问服务器端时，服务器端可以获取到客户端Cookie缓存文件。
* Cookie是由服务器端创建的，然后由服务器端发送给客户端，客户端以键值对形式存储Cookie，并标注Cookie的来源。客户端再次访问服务器端时，存储的Cookie会保存在请求协议中，服务器端可以获取上次存储的缓存文件内容。

### Cookie特点

* Cookie的用途：

电子商城中购物车功能（每买一样商品，保存一个Cookie）

用户自动登录功能（第一次登录时，将用户名和密码存储在Cookie）

* Cookie的缺点：

多人共用一台计算机（例如导致用户名和密码不安全等问题）。

Cookie被删除时，利用Cookie统计用户数量出现偏差。

一人使用多台计算机（网站会将看成多个用户等问题）

Cookie会被附加在每次Http请求协议中，增加流量。

Cookie使用明文（未加密）传递的，安全性低。

Cookie的大小限制在4KB左右，无法存储复杂需求。

* Cookie规范

Http协议提供了有关Cookie的规范，现今市场上出现大量浏览器，一些浏览器对该Cookie规范进行了一些“扩展”，但Cookie缓存文件不会占满硬盘空间。

Cookie存储的大小上限为4KB。

一个服务器最多在客户端浏览器中可以保存20个Cookie。

一个浏览器最多可以保存300个Cookie。

### Cookie主要属性

|  |
| --- |
| name：cookie的名字，每个cookie都有一个名字；  content：cookie的值，与名字一起作为键值对形式存在；  domain：域，该cookie的域名，例如csdn.net，说明当前cookie来自csdn.net;  path：路径，访问csdn.net下该路径时，当前cookie将被发送；  created：cookie被创建的时间；  Expired：cookie失效的时间；  最大生命时间：失效时间和创建时间的时间差，就是cookie的最大生命时间，超过该时间，cookie将失效，不再被发送到相应的域地址； |

### Cookie操作（java部分）

|  |
| --- |
| * 写cookie   //创建/修改cookie对象，键值对形式  Cookie cookie = new Cookie("username", "admin");  //设置过期时间 单位秒，删除Cookie只需要修改过期时间为负数  cookie.setMaxAge(60\*60);  //cookie还可以设置path等信息  //cookie对象存入response，响应给浏览器，浏览器接受到响应数据时才会把cookie存储在本地  response.addCookie(cookie);   * 读cookie   //从request对象中取值  Cookie[] cookies = request.getCookies();  //循环输出值  for(Cookie cookie : cookies){  //键: cookie.getName()  //值： cookie.getValue()  } |

### Cookie操作（js部分）

|  |
| --- |
| * Cookie写   document.cookie="jscook=jsval1;path=/;expires="+new Date("2021-11-11 11:11:11");   * Cookie读   var cookie = document.cookie //读到键值对格式的字符串 如果需要找到特定key的value需要手动解析  var mycks = document.cookie.split(";");  $.each(mycks,function(i,d){  var kvarr = d.split("= ");  console.log("key是："+ kvarr[0]+" value是："+kvarr[1]);  })   * 删除cookie   document.cookie="jskey=jsval;expires="+new Date(now.getTime()-1000).toGMTString(); |

注意：cookie在使用时，相同domain，不同path下可以存储同名cookie，为了让cookie设置统一，和解决cookie读取权限问题，经常会把cookie的path直接设置到根/

## Web storage使用

* HTML5 提供了两种在客户端存储数据的新方法 统称Web storage：

localStorage - 没有时间限制的数据存储

sessionStorage - 针对一个 session 的数据存储，数据在浏览器关闭后自动删除

* Web storage是一种设计由前端存储数据的技术，跟cookie有相似之处，但是因为设计给前端使用，所以操作上比cookie方便很多

|  |
| --- |
| https://images2015.cnblogs.com/blog/1120414/201707/1120414-20170702112005930-917310833.jpg |

* Web storage操作

|  |
| --- |
| localStorage.username = "admin";//设置值  console.log(localStorage.username)//取值  localStorage.setItem("user", "user\_name");//设置值  console.log(localStorage.getItem("user"));//取值  //以上两种方式存值/取值都可以  localStorage.removeItem("user");//移除指定的key  localStorage.clear();//清空所有的数据 |

注意：sessionStorage用法同localStorage

Web storage是前端的技术，跟服务端的session对象无直接关系

Web storage没有默认随请求发送到服务器的机制