

一、会话概述

1、什么是会话？如同打电话

2、会话过程要解决的问题是什么？

保持各个客户端自己的数据。

二、Cookie

由于Cookie数据是由客户端来保存和携带的，所以称之为客户端技术。

创建一个 cookie，cookie 是 servlet 发送到 Web 浏览器的少量信息，这些信息由浏览器保存，然后发回服务器。cookie 的值可以唯一地标识客户端，因此 cookie 常用于会话管理。

一个 cookie 拥有一个名称、一个值和一些可选属性，比如注释、路径、域限定符、最大生存时间和版本号。一些 Web 浏览器在处理可选属性方面存在 bug，因此有节制地使用这些属性可提高 servlet 的互操作性。

servlet 通过使用 `HttpServletResponse.addCookie` 方法将 cookie 发送到浏览器，该方法将字符串添加到 HTTP 响应头，以便一次一个地将 cookie 发送到浏览器。浏览器应该支持每个 Web 服务器有 20 个 cookie，总共有 300 个 cookie，并且可能将每个 cookie 的大小限定为 4 KB。

浏览器通过向 HTTP 请求头添加字符串 cookie 返回给 servlet，可使用 `HttpServletRequest.getCookies` 方法从请求中获取 cookie。一些 cookie 可能有相同的名称，但有不同的属性。

cookie 影响使用它们的 Web 页面的缓存。HTTP 1.0 不会缓存那些使用通过此类创建的 cookie 的页面。此类不支持 HTTP 1.1 中定义的缓存控件。

此类支持版本 0（遵守 Netscape 协议）和版本 1（遵守 RFC 2109 协议）cookie 规范。默认情况下，cookie 是使用版本 0 创建的，以确保最佳互操作性。

英文文档：

1、属性：

- l name：名称不能唯一确定一个Cookie。路径可能不同。
- l value：不能存中文。
- l path：默认值是写Cookie的那个程序的访问路径
- n 比如：http://localhost:8080/day10_00_cookie/servlet/ck1写的Cookie
- u path就是：/day10_00_cookie/servlet 看当前创建cookie的资源（servlet）文件路径
- u 客户端在访问服务器另外资源时，根据访问的路径来决定是否带着Cookie到服务器
- u 当前访问的路径如果是以cookie的path开头的路径，浏览器就带。否则不带。
- l maxAge：cookie的缓存时间。默认是-1（默认存在浏览器的内存中）。单位是秒。
- n 负数：cookie的数据存在浏览器缓存中
- n 0：删除。路径要保持一致，否则可能删错人。
- n 正数：缓存（持久化到磁盘上）的时间

三、HttpSession

1、为什么要学HttpSession?

- › 它也是一个域对象: session servletContext request
- › 同一个会话下, 可以使一个应用的多个资源共享数据
- › cookie客户端技术, 只能存字符串。HttpSession服务器端的技术, 它可以存对象。

2、常用方法

把数据保存在HttpSession对象中, 该对象也是一个域对象。

```
void setAttribute(String name, Object value);  
Object getAttribute(String name);  
void removeAttribute(String name);  
HttpSession.getId();
```

```
setMaxInactiveInterval(int interval)    设置session的存活时间  
invalidate() 使此会话无效
```

3、getSession():内部执行原理

HttpSession request.getSession():内部执行原理

- 1、获取名称为JSESSIONID的cookie的值。
- 2、没有这样的cookie, 创建一个新的HttpSession对象, 分配一个唯一的SessionID, 并且向客户端写了一个名字为JSESSIONID=sessionID的cookie
- 3、有这样的Cookie, 获取cookie的值(即HttpSession对象的值), 从服务器的内存中根据ID找那个HttpSession对象:
 - 找到了: 取出继续为你服务。
 - 找不到: 从2开始。

HttpSession request.getSession(boolean create):

参数:

true: 和getSession()功能一样。

false: 根据客户端JSESSIONID的cookie的值, 找对应的HttpSession对象, 找不到返回null(不会创建新的, 只是查询)。

3、客户端禁用Cookie后的会话数据保存问题

客户端禁用cookie：浏览器永远不会向服务器发送cookie的请求消息头

解决方案：

方案一：在主页上给出提示：请不要禁用您的cookie

方案二：URL重写。必须对网站的所有地址都重写。

`http://url--->http://url;JSESSIONID=111`

`response.encodeURL(String url);`

看浏览器有没有发送cookie请求消息头，没有就重写URL，有就不重写。

`request.getSession();`必须写