**首先在WEB应用中,我们把同一个浏览器与WEB服务器的一次一系列的各种交互活动称之为"会话".**

**在很多WEB应用中,我们都需要对用户的会话进行跟踪,需要记录用户的状态,比如商城中用户将商品添加到购物车,商城必须能识别哪些会话是属于同一个用户,既是哪些客户端添加了哪些商品,不能出现,A用户的客户端添加的商品跑到B用户客户端的购物车里头!**

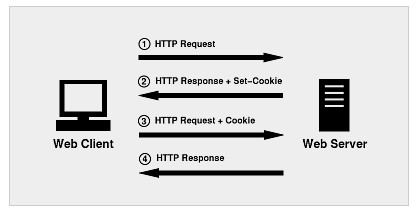
**在Servlet技术中,有两种机制可以实现这种对会话的跟踪:**

**①Cookie机制:放在客户端**

**②Session机制:放在服务器端**

**1.Cookie**

**Cookie技术是客户端的解决方案，当用户使用浏览器访问一个支持Cookie的网站的时候，一个cookie的运行过程分为以下四步：**



**当客户端浏览器接收到来自服务器的响应之后，浏览器会将这些信息存放在一个统一的位置，对于Windows操作系统而言，IE浏览器的Cookie 放在:[系统盘]:\Users\用户名\Cookies目录中；**

**通常，我们可以从很多网站的登录界面中看到“请记住我”这样的选项，如果你勾选了它之后再登录，那么在下一次访问该网站的时候就不需要进行重复而繁琐的登录动作了，而这个功能就是通过Cookie实现的。**

**其实本质上cookies就是http的一个扩展。有两个http头部是专门负责设置以及发送cookie的,它们分别是Set-Cookie以及Cookie。其中如果包含Set-Cookie这个头部时，意思就是指示客户端建立一个cookie，并且在后续的http请求中自动发送这个cookie到服务器端，直到这个cookie过期。如果cookie的生存时间是整个会话期间的话，那么浏览器会将cookie保存在内存中，浏览器关闭时就会自动清除这个cookie。另外一种情况就是保存在客户端的硬盘中，浏览器关闭的话，该cookie也不会被清除，下次打开浏览器访问对应网站时，这个cookie就会自动再次发送到服务器端。**

**Java中把Cookie封装成了javax.servlet.http.Cookie类。每个Cookie都是该Cookie类的对象。服务器通过操作Cookie类对象对客户端Cookie进行操作。通过response.addCookie(Cookie cookie)向客户端设置Cookie，通过request.getCookie()获取客户端提交的所有Cookie（以Cookie[]数组形式返回）。**



**Cookie对象里的属性(每个属性都有set和get方法):**

**①String name：该Cookie的名称。Cookie一旦创建，名称便不可更改。**

**② Object value：该Cookie的值。**

**③int maxAge：该Cookie失效的时间，单位秒。**

**(如果为正数，则该Cookie在>maxAge秒之后失效。如果为负数，该Cookie为临时Cookie，关闭浏览器即失效，浏览器也不会以任何形式保存该Cookie。如果为0，表示删除该Cookie。默认为–1。)**

**④boolean secure：该Cookie是否仅被使用安全协议传输。默认为false。**

**⑤String path：该Cookie的使用路径。**

**如果设置为“/sessionWeb/”，则只有contextPath为“/sessionWeb”的程序可以访问该Cookie。如果设置为“/”，则本域名下contextPath都可以访问该Cookie。注意最后一个字符必须为“/”。**

**⑥String domain：可以访问该Cookie的域名。**

**如果设置为“.google.com”，则所有以“google.com”结尾的域名都可以访问该Cookie。注意第一个字符必须为“.”。**

**⑦String comment：该Cookie的用处说明。**

**⑧ int version：该Cookie使用的版本号。0表示遵循Netscape的Cookie规范，1表示遵循W3C的。**





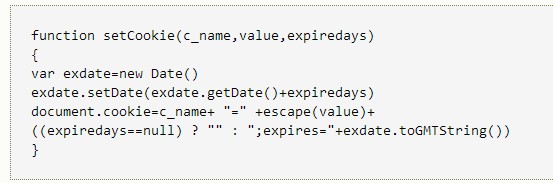


**JavaScript操作Cookie**

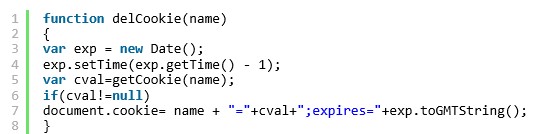
**<script>**

**document.write(document.cookie);//获取cookie的键值对的字符串**

**</script>**







**案例：永久登录**

**只在登录时查询一次数据库，以后访问验证登录信息时不再查询数据库。实现方式是把账号按照一定的规则加密后，连同账号一块保存到Cookie中。下次访问时只需要判断账号的加密规则是否正确即可。本例把账号保存到名为account的Cookie中，把账号连同密钥用MD5算法加密后保存到名为ssid的Cookie中。验证时验证Cookie中的账号与密钥加密后的ssid是否程序中计算的结果一样。相关代码如下： login.jsp：**



