**Cookie and session**

**1. 概述**

简单来说，他们只是http中的一个配置项，在servlet规范中也只对应一个类。Session和cookie的作用都是为了保持访问用户和后端服务器的交互状态。

限制：http对cookie的数量和大小限制，cookie被盗、cookie伪造问题；如何解决session在多台服务器之间的共享问题。

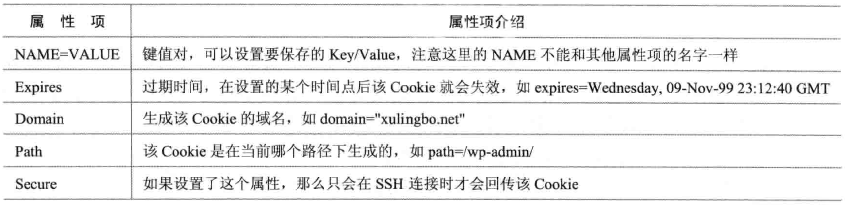
**2. cookie**

Cookie作用：通俗讲，当一个用户通过http访问一个服务器时。这个服务器会将一些key/value键值对方会给客户端浏览器，并给这些数据加上一些限制条件，在条件符合时，这个用户下次访问这个服务器时，数据又被完整的带回给服务器。

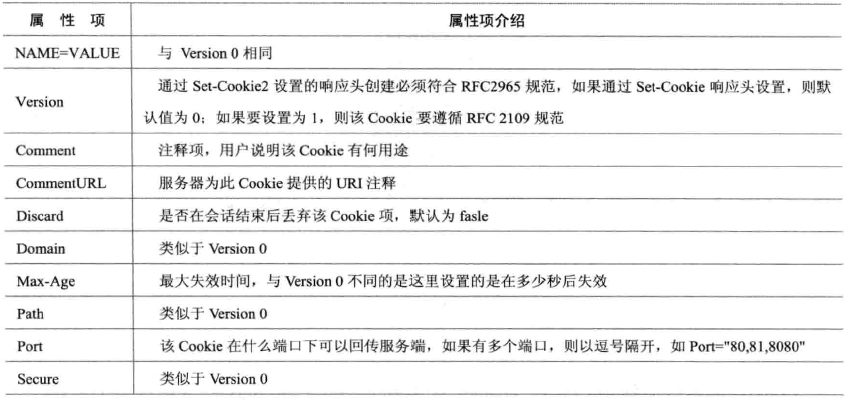
设计之初考虑的是，为了记录用户在一段时间内访问web应用的行为路径。由于http是一种无状态协议，当用户的一次访问请求结束后。后端服务器就无法知道下一次访问的还是不是上次访问的用户。当有了cookie之后，服务端就可以根据cookie值来划分访问的用户了。

**2.1 cookie的属性项**

Version 0：



Version 1：



**2.2 cookie是如何工作**

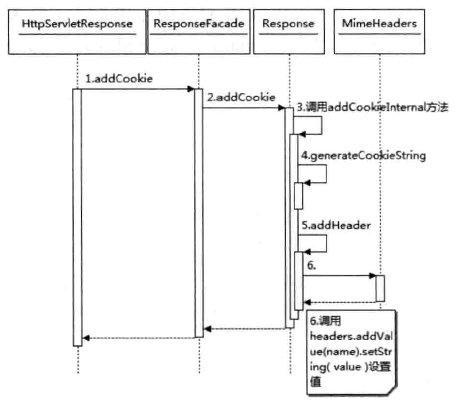


图1 Tomcat创建set-cookie响应头的时序图

由图1可知，真正构建cookie实在org.apache.catalina.connector.Response类中完成的，调用generateCookieString方法将Cookie对象构造成一个字符串，构造的字符串的格式如下：userName(cookie-key)=“yunfei”(cookie-value); Version=”1”; Domain=”baidu.com”; Max-Age=1000。然后将这个字符串命名为Set-Cookie添加到MimeHeaders中。

当我们通过response.addCookie()创建多个Cookie时，最终在请求返回时构造的http响应头是否将相同的Header标识的Set-Cookie值进行合并？在构建http返回字节流时是将Header中所有的项顺序的写出，而没有进行任何修改。所以可以想象浏览器在接受http返回的数据时是分别解析每一个Header项的。

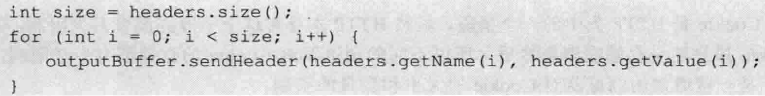


图2 tomcat最终构造http响应头的代码

当我们请求某个URL路径时，浏览器会根据这个URL路径将符合条件的Cookie放在Request请求头中传回给服务器，服务器通过reques.getCookie()来获取所有的cookie。

**2.3 cookie使用限制**

Cookie是http头中的一个字段，虽然http本身对这个字段并没有多少限制，但是cookie最终还是存储在浏览器里，所以不同浏览器对cookie的存储（数量和大小）都有一些限制。

**3 session**

前文讲述中，cookie可以让服务器追踪每个客户端的访问，但是每次客户端的访问都必须传回这些cookie，如果cookie很多，则增加了cs交互的数据传输量，而session的出现正是解决这个问题。

同一客户端每次和服务器交互时，不需要每次传回所有的cookie值，只需要传回一个id（客户端第一次访问服务器时生成的，每个客户端是唯一的），这样每个客户端就有了唯一的id，这个id通常是NAME为JESIONID的一个cookie。