**前几天面试问了一个问题，当时记不太清了，上网查了下发现这个问题还真的很有讲究而且很重要，自己总结下做下记录。**

**一、Session与Cookie介绍**

**这些都是基础知识，不过有必要做深入了解。先简单介绍一下。**

**二者的定义：**

**当你在浏览网站的时候，WEB 服务器会先送一小小资料放在你的计算机上，Cookie 会帮你在网站上所打的文字或是一些选择，**

**都纪录下来。当下次你再光临同一个网站，WEB 服务器会先看看有没有它上次留下的 Cookie 资料，有的话，就会依据 Cookie**

**里的内容来判断使用者，送出特定的网页内容给你。 Cookie 的使用很普遍，许多有提供个人化服务的网站，都是利用 Cookie**

**来辨认使用者，以方便送出使用者量身定做的内容，像是 Web 接口的免费 email 网站，都要用到 Cookie。**

**具体来说cookie机制采用的是在客户端保持状态的方案，而session机制采用的是在服务器端保持状态的方案。**

**同时我们也看到，由于采用**[服务器端](http://cpro.baidu.com/cpro/ui/uijs.php?adclass=0&app_id=0&c=news&cf=1001&ch=0&di=128&fv=11&is_app=0&jk=f27d78b457762e6d&k=%B7%FE%CE%F1%C6%F7%B6%CB&k0=%B7%FE%CE%F1%C6%F7%B6%CB&kdi0=0&luki=10&n=10&p=baidu&q=65035100_cpr&rb=0&rs=1&seller_id=1&sid=6d2e7657b4787df2&ssp2=1&stid=0&t=tpclicked3_hc&td=1836545&tu=u1836545&u=http%3A%2F%2Fwww%2Ebubuko%2Ecom%2Finfodetail%2D388671%2Ehtml&urlid=0)**保持状态的方案在客户端也需要保存一个**[标识](http://cpro.baidu.com/cpro/ui/uijs.php?adclass=0&app_id=0&c=news&cf=1001&ch=0&di=128&fv=11&is_app=0&jk=f27d78b457762e6d&k=%B1%EA%CA%B6&k0=%B1%EA%CA%B6&kdi0=0&luki=8&n=10&p=baidu&q=65035100_cpr&rb=0&rs=1&seller_id=1&sid=6d2e7657b4787df2&ssp2=1&stid=0&t=tpclicked3_hc&td=1836545&tu=u1836545&u=http%3A%2F%2Fwww%2Ebubuko%2Ecom%2Finfodetail%2D388671%2Ehtml&urlid=0)**，所以session机制可能需要借助于cookie机制**

**来达到保存标识的目的，但实际上它还有其他选择。**

**cookie机制。正统的cookie分发是通过扩展HTTP协议来实现的，服务器通过在HTTP的响应头中加上一行特殊的指示以提示**

**浏览器按照指示生成相应的cookie。然而纯粹的客户端脚本如JavaScript或者VBScript也可以生成cookie。而cookie的使用**

**是由浏览器按照一定的原则在后台自动发送给服务器的。浏览器检查所有存储的cookie，如果某个cookie所声明的作用范围**

**大于等于将要请求的资源所在的位置，则把该cookie附在请求资源的HTTP请求头上发送给服务器。  
   
cookie的内容主要包括：名字，值，过期时间，路径和域。路径与域一起构成cookie的作用范围。若不设置过期时间，则表示这**

**个cookie的生命期为浏览器会话期间，关闭浏览器窗口，cookie就消失。这种生命期为浏览器会话期的cookie被称为会话cookie。**

**会话cookie一般不存储在硬盘上而是保存在内存里，当然这种行为并不是规范规定的。若设置了过期时间，浏览器就会把cookie**

**保存到硬盘上，关闭后再次打开浏览器，这些cookie仍然有效直到超过设定的过期时间。**[存储](http://cpro.baidu.com/cpro/ui/uijs.php?adclass=0&app_id=0&c=news&cf=1001&ch=0&di=128&fv=11&is_app=0&jk=f27d78b457762e6d&k=%B4%E6%B4%A2&k0=%B4%E6%B4%A2&kdi0=0&luki=6&n=10&p=baidu&q=65035100_cpr&rb=0&rs=1&seller_id=1&sid=6d2e7657b4787df2&ssp2=1&stid=0&t=tpclicked3_hc&td=1836545&tu=u1836545&u=http%3A%2F%2Fwww%2Ebubuko%2Ecom%2Finfodetail%2D388671%2Ehtml&urlid=0)**在硬盘上的cookie可以在不同的浏**

**览器进程间共享，比如两个IE窗口。而对于保存在**[内存](http://cpro.baidu.com/cpro/ui/uijs.php?adclass=0&app_id=0&c=news&cf=1001&ch=0&di=128&fv=11&is_app=0&jk=f27d78b457762e6d&k=%C4%DA%B4%E6&k0=%C4%DA%B4%E6&kdi0=0&luki=5&n=10&p=baidu&q=65035100_cpr&rb=0&rs=1&seller_id=1&sid=6d2e7657b4787df2&ssp2=1&stid=0&t=tpclicked3_hc&td=1836545&tu=u1836545&u=http%3A%2F%2Fwww%2Ebubuko%2Ecom%2Finfodetail%2D388671%2Ehtml&urlid=0)**里的cookie，不同的浏览器有不同的处理方式**

**session机制。session机制是一种**[服务器端](http://cpro.baidu.com/cpro/ui/uijs.php?adclass=0&app_id=0&c=news&cf=1001&ch=0&di=128&fv=11&is_app=0&jk=f27d78b457762e6d&k=%B7%FE%CE%F1%C6%F7%B6%CB&k0=%B7%FE%CE%F1%C6%F7%B6%CB&kdi0=0&luki=10&n=10&p=baidu&q=65035100_cpr&rb=0&rs=1&seller_id=1&sid=6d2e7657b4787df2&ssp2=1&stid=0&t=tpclicked3_hc&td=1836545&tu=u1836545&u=http%3A%2F%2Fwww%2Ebubuko%2Ecom%2Finfodetail%2D388671%2Ehtml&urlid=0)**的机制，服务器使用一种类似于散列表的结构（也可能就是使用散列表）来保存信息。**

**当程序需要为某个客户端的请求创建一个session时，服务器首先检查这个客户端的请求里是否已包含了一个session标识**

**（称为session id），如果已包含则说明以前已经为此客户端创建过session，服务器就按照session id把这个session检索出来**

**使用（检索不到，会新建一个），如果客户端请求不包含session id，则为此客户端创建一个session并且生成一个与此session相**

**关联的session id，session id的值应该是一个既不会重复，又不容易被找到规律以仿造的字符串，这个session id将被在本次响应**

**中返回给客户端保存。保存这个session id的方式可以采用cookie，这样在交互过程中浏览器可以自动的按照规则把这个**[标识](http://cpro.baidu.com/cpro/ui/uijs.php?adclass=0&app_id=0&c=news&cf=1001&ch=0&di=128&fv=11&is_app=0&jk=f27d78b457762e6d&k=%B1%EA%CA%B6&k0=%B1%EA%CA%B6&kdi0=0&luki=8&n=10&p=baidu&q=65035100_cpr&rb=0&rs=1&seller_id=1&sid=6d2e7657b4787df2&ssp2=1&stid=0&t=tpclicked3_hc&td=1836545&tu=u1836545&u=http%3A%2F%2Fwww%2Ebubuko%2Ecom%2Finfodetail%2D388671%2Ehtml&urlid=0)**发送给**

**服务器。一般这个cookie的名字都是类似于SEEESIONID。但cookie可以被人为的禁止，则必须有其他机制以便在cookie被禁止时**

**仍然能够把session id传递回服务器。**

**经常被使用的一种技术叫做URL重写，就是把session id直接附加在URL路径的后面。还有一种技术叫做表单隐藏字段。就是服务器**

**会自动修改表单，添加一个隐藏字段，以便在表单提交时能够把session id传递回服务器。比如：   
<form name="testform" action="/xxx">   
<input type="hidden" name="jsessionid" value="ByOK3vjFD75aPnrF7C2HmdnV6QZcEbzWoWiBYEnLerjQ99zWpBng!-145788764">   
<input type="text">   
</form>**

实际上这种技术可以简单的用对action应用URL重写来代替。

**二、二者之间的区别**

cookie和session的比较

一、对于cookie：

①cookie是创建于服务器端

②cookie保存在浏览器端

③cookie的生命周期可以通过cookie.setMaxAge(2000);来设置，如果没有设置setMaxAge,

则cookie的生命周期当浏览器关闭的时候，就消亡了

④cookie可以被多个同类型的浏览器共享  可以把cookie想象成一张表

比较：

①存在的位置：

cookie 存在于客户端，临时文件夹中

session：存在于服务器的内存中，一个session域对象为一个用户浏览器服务

②安全性

cookie是以明文的方式存放在客户端的，安全性低，可以通过一个加密算法进行加密后存放

session存放于服务器的内存中，所以安全性好

③网络传输量

cookie会传递消息给服务器

session本身存放于服务器，不会有传送流量

④生命周期(以20分钟为例)

(1)cookie的生命周期是累计的，从创建时，就开始计时，20分钟后，cookie生命周期结束，

(2)session的生命周期是间隔的，从创建时，开始计时如在20分钟，没有访问session，那么session生命周期被销毁

但是，如果在20分钟内（如在第19分钟时）访问过session，那么，将重新计算session的生命周期

(3)关机会造成session生命周期的结束，但是对cookie没有影响

⑤访问范围

session为一个用户浏览器独享

cookie为多个用户浏览器共享

使session失效的方法：

1.关闭tomcat

2.reload web应用

3.session时间到

4invalidate  session