**[session超时和cookie过期](http://www.cnblogs.com/hnsongbiao/p/4245833.html)**

**一、Cookie的过期和Session的超时有什么区别？**  
   
会话的超时由服务器来维护，它不同于Cookie的失效日期。首先，会话一般基于驻留内存的cookie，不是持续性的cookie，因而也就没有截至日期。即使截取到JSESSIONID cookie，并为它设定一个失效日期发送出去。浏览器会话和服务器会话也会截然不同。  
   
**二、session cookie和session对象的生命周期是一样的吗？**  
   
当用户关闭了浏览器虽然session cookie已经消失，但session对象仍然保存在服务器端。  
   
**三、是否只要关闭浏览器，session就消失了？**  
   
程序一般都是在用户做log off的时候发个指令去删除session，然而浏览器从来不会主动在关闭之前通知服务器它将要被关闭，因此服务器根本不会有机会知道浏览器已经关闭。  
   
服务器会一直保留这个会话对象直到它处于非活动状态超过设定的间隔为止。  
之所以会有这种错误的认识，是因为大部分session机制都使用会话cookie来保存session id，而关闭浏览器后这个session id就消失了，再次连接到服务器时也就无法找到原来的session。  
   
如果服务器设置的cookie被保存到硬盘上，或者使用某种手段改写浏览器发出的HTTP请求报头，把原来的session id发送到服务器，则再次打开浏览器仍然能够找到原来的session。  
   
恰恰是由于关闭浏览器不会导致session被删除，迫使服务器为session设置了一个失效时间，当距离客户上一次使用session的时间超过了这个失效时间时，服务器就可以认为客户端已经停止了活动，才会把session删除以节省存储空间。

<system.web>

<sessionState mode="InProc" cookieless="false" timeout="50"/>

</system.web>

由此我们可以得出如下结论：  
关闭浏览器，只会是浏览器端内存里的session cookie消失，但不会使保存在服务器端的session对象消失，同样也不会使已经保存到硬盘上的持久化cookie消失。  
   
**四、如何使用会话（session）显示每个客户的访问次数？**  
   
由于客户的访问次数是一个整型的变量，但session的属性类型中不能使用int，double，boolean等基本类型的变量，所以我们要用到这些基本类型的封装类型对象作为session对象中属性的值。  
但像Integer是一种不可修改(Immutable)的数据结构：构建后就不能更改。这意味着每个请求都必须创建新的Integer对象，之后使用setAttribute来代替之前存在的老的属性的值。  
例如：

[复制代码](javascript:void(0);)

HttpSession session = request.getSession();

SomeImmutalbeClass value = (SomeImmutableClass)session.getAttribute(“SomeIdentifier”);

if (value= =null){

value = new SomeImmutableClass(…);　// 新创建一个不可更改对象

}else{

value = new SomeImmutableClass(calculatedFrom(value)); // 对value重新计算后创建新的对象

}

session.setAttribute(“someIdentifier”,value); // 使用新创建的对象覆盖原来的老的对象

[复制代码](javascript:void(0);)

使用可变的数据结构，比如数组、List、Map或含有可写字段的应用程序专有的数据结构。通过这种方式，除非首次分配对象，否则不需要调用setAttribute。