[**Session的生命周期**](http://www.cnblogs.com/binger/archive/2013/03/19/2970171.html)

　　以前在学习的时候没怎么注意，今天又回过头来仔细研究研究了一下Session的生命周期。

　　Session存储在服务器端，一般为了防止在服务器的内存中（为了高速存取），Sessinon在用户访问第一次访问服务器时创建，需要注意只有访问JSP、Servlet等程序时才会创建Session，只访问HTML、IMAGE等静态资源并不会创建Session，可调用request.getSession(true)强制生成Session。

**Session什么时候失效？**

　　1. 服务器会把长时间没有活动的Session从服务器内存中清除，此时Session便失效。Tomcat中Session的默认失效时间为20分钟。

　　2. 调用Session的invalidate方法。

**Session对浏览器的要求：**

　虽然Session保存在服务器，对客户端是透明的，它的正常运行仍然需要客户端浏览器的支持。这是因为Session需要使用Cookie作为识别标志。HTTP协议是无状态的，Session不能依据HTTP连接来判断是否为同一客户，因此服务器向客户端浏览器发送一个名为JSESSIONID的Cookie，它的值为该Session的id（也就是HttpSession.getId()的返回值）。Session依据该Cookie来识别是否为同一用户。

　　该Cookie为服务器自动生成的，它的maxAge属性一般为-1，表示仅当前浏览器内有效，并且各浏览器窗口间不共享，关闭浏览器就会失效。因此同一机器的两个浏览器窗口访问服务器时，会生成两个不同的Session。但是由浏览器窗口内的链接、脚本等打开的新窗口（也就是说不是双击桌面浏览器图标等打开的窗口）除外。这类子窗口会共享父窗口的Cookie，因此会共享一个Session。

　　注意：新开的浏览器窗口会生成新的Session，但子窗口除外。子窗口会共用父窗口的Session。例如，在链接上右击，在弹出的快捷菜单中选择"在新窗口中打开"时，子窗口便可以访问父窗口的Session。

如果客户端浏览器将Cookie功能禁用，或者不支持Cookie怎么办？例如，绝大多数的手机浏览器都不支持Cookie。Java Web提供了另一种解决方案：URL地址重写。

　　URL地址重写是对客户端不支持Cookie的解决方案。URL地址重写的原理是将该用户Session的id信息重写到URL地址中。服务器能够解析重写后的URL获取Session的id。这样即使客户端不支持Cookie，也可以使用Session来记录用户状态。HttpServletResponse类提供了encodeURL(String url)实现URL地址重写，该方法会自动判断客户端是否支持Cookie。如果客户端支持Cookie，会将URL原封不动地输出来。如果客户端不支持Cookie，则会将用户Session的id重写到URL中。

　　注意：TOMCAT判断客户端浏览器是否支持Cookie的依据是请求中是否含有Cookie。尽管客户端可能会支持Cookie，但是由于第一次请求时不会携带任何Cookie（因为并无任何Cookie可以携带），URL地址重写后的地址中仍然会带有jsessionid。当第二次访问时服务器已经在浏览器中写入Cookie了，因此URL地址重写后的地址中就不会带有jsessionid了。