[cookie、session、sessionid 与jsessionid](http://www.cnblogs.com/fnng/archive/2012/08/14/2637279.html)

2012-08-14 00:04 by 虫师, 37507 阅读, 12 评论, [收藏](http://www.cnblogs.com/fnng/archive/2012/08/14/2637279.html), [编辑](https://i.cnblogs.com/EditPosts.aspx?postid=2637279)

　　cookie、session、sessionid 与jsessionid，要想明白他们之间的关系，下面来看个有趣的场景来帮你理解。

　　我们都知道银行，银行的收柜台每天要接待客户存款/取款业务，可以有几种方案：

　　凭借柜台职员的记忆，由收柜台职员来为每位顾客办理存款/取款业务，单凭职员的记忆力，要记到每位顾客的相貌，并迅速这个顾客当前的存款以及存取的次数，每次存取的金额是多少。-----------这种方式表示协议本身支持状态。



     使用存折的方式，然后柜台职员就把每个顾客的存款/取款的信息保存在这张折子，然后交给顾客保管，当顾客来存款/取款时，只要拿出存折，职员查看存折就对当前这位顾客的存款/取款信息一目了然。当然，你马上会想到，顾客修改这个信息怎么办？我们也有措施对每次存款/取款记录后面盖章。无盖章的就是假冒信息。但如果顾客是真的要伪造，当然印章也是可以伪造的。-------------这种方式就是在客户端端保持状态。



     使用银行卡的方式，发给每位银行用户一张银行卡，银行卡上有一个唯一的卡号，没有其它任何信息，当顾客来存款/取款时，拿出银行卡，银行把卡号输入的电脑，很快就显示当前用户的存/取款记录。这种方式的安全性就会有很大的提高。用户想要手脚只有攻破银行的服务器来修改自己的存/取款信息，这样做难度会很大。---------这种方式就是服务器端保持状态。

**Cookie 与session的产生过程**

　　我们都知道HTTP协议本身是无状态的，客户只需要简单的向服务器来发送请求下载某些文件，客户端向服务器端发送的每次请求都是独立的。对于当前的web应用，HTTP的“无状态”，导致许多应用都不得不花费大量的精力来记录用户的操作步骤。就像我们上面介绍的第一种情况，银行职员要花费大量的精力来记忆每一位用户的存/取款记录。

　　程序员很快发现，如果能够提供一些按需生成的动态信息，会使web的交互能力大大增强。程序员一方面在HTML中添加表单、脚本、DOM等客户端行为，来增加web应用与客户端的交互性。另一方面在服务器端测出现了CGI规范以响应客户端的动态请求，作为传输载体的HTTP协议添加了文件上载、cookie 等特性。那cookie的原理与我们上面介绍的使用存折记录用户应为的方式是一样一样的。

　　通过前面的例子我们已经发现，通过cookie的方式存储信息，可能会存在一点点的安全性，因为所有的信息都是写在客户端的，客户可能会对这些信息进行修改或清除。然后就又出现session的方式用于保存用户行为，这种方式的原理与前面介绍银行卡的方式是一样的。

　　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，由于采用服务器端保持状态的方案在客户端也需要保存一个标识，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与session的机制与原理**

　　 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发挥给服务器。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。

**Jsessionid?**

　　Jsessionid只是tomcat的对sessionid的叫法，其实就是sessionid；在其它的容器也许就不叫jsessionid了。

-------------------最新更新--------

**Terry\_Huang**  回复：

　　给你举个更生动的例子，以前大学的时候，经常去大卡司去喝奶茶，每喝一杯，都可以得到一个印花（第一次喝的时候他会给你个积分卡片)，集齐6个印花之后，就可以免费获得一杯奶茶。这样子，印花的信息是保存在客户的积分卡上，你如果不怀好意的话，就自己搞几个神似的印花去骗奶茶喝吧。哈哈，这样子是不是更符合在客户端端保持状态。而拿银行卡去银行取钱，我们的卡只需要保存一个卡号，更多的信息是保存在服务器中，这样也比较符合服务器端保持状态。希望能帮到你。

    可能是我的描述不够贴切，积分卡和存折是类似的，用户的存取信息是保存在存折上的，存折又是在客户手里（客户端）可能现在很少有人用存折了，或存折已经被银行取消掉了。