**八幅漫画理解使用JSON Web Token设计单点登录系统**

9月 07, 2015 in [Engineering](http://blog.leapoahead.com/categories/Engineering/)

上次在[《JSON Web Token - 在Web应用间安全地传递信息》](http://blog.leapoahead.com/2015/09/06/understanding-jwt/)中我提到了JSON Web Token可以用来设计单点登录系统。我尝试用八幅漫画先让大家理解如何设计正常的用户认证系统，然后再延伸到单点登录系统。

如果还没有阅读[《JSON Web Token - 在Web应用间安全地传递信息》](http://blog.leapoahead.com/2015/09/06/understanding-jwt/)，我强烈建议你花十分钟阅读它，理解JWT的生成过程和原理。

**用户认证八步走**

所谓用户认证（Authentication），就是让用户登录，并且在接下来的一段时间内让用户访问网站时可以使用其账户，而不需要再次登录的机制。

小知识：可别把用户认证和用户授权（Authorization）搞混了。用户授权指的是规定并允许用户使用自己的权限，例如发布帖子、管理站点等。

首先，服务器应用（下面简称“应用”）让用户通过Web表单将自己的用户名和密码发送到服务器的接口。这一过程一般是一个HTTP POST请求。建议的方式是通过SSL加密的传输（https协议），从而避免敏感信息被嗅探。



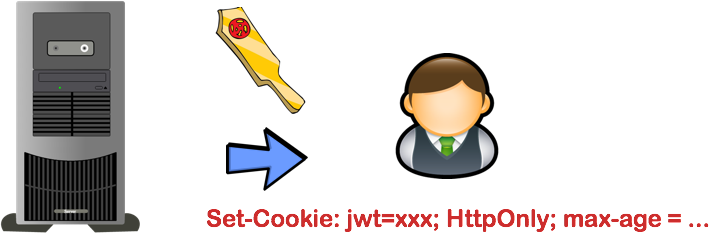
接下来，应用和数据库核对用户名和密码。



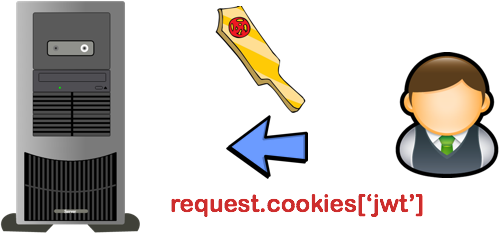
核对用户名和密码成功后，应用将用户的id（图中的user\_id）作为JWT Payload的一个属性，将其与头部分别进行Base64编码拼接后签名，形成一个JWT。这里的JWT就是一个形同lll.zzz.xxx的字符串。



应用将JWT字符串作为该请求Cookie的一部分返回给用户。注意，在这里必须使用HttpOnly属性来防止Cookie被JavaScript读取，从而避免[跨站脚本攻击（XSS攻击）](http://www.cnblogs.com/bangerlee/archive/2013/04/06/3002142.html" \t "_blank)。



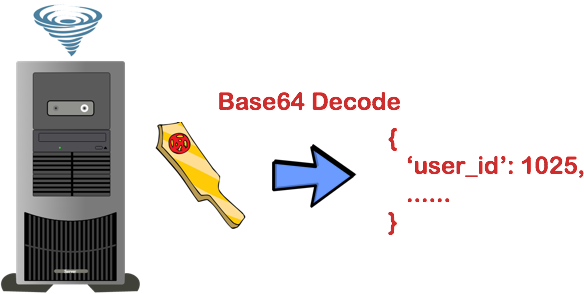
在Cookie失效或者被删除前，用户每次访问应用，应用都会接受到含有jwt的Cookie。从而应用就可以将JWT从请求中提取出来。



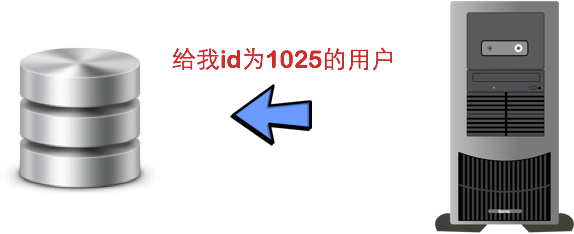
应用通过一系列任务检查JWT的有效性。例如，检查签名是否正确；检查Token是否过期；检查Token的接收方是否是自己（可选）。



应用在确认JWT有效之后，JWT进行Base64解码（可能在上一步中已经完成），然后在Payload中读取用户的id值，也就是user\_id属性。这里用户的id为1025。



应用从数据库取到id为1025的用户的信息，加载到内存中，进行ORM之类的一系列底层逻辑初始化。



应用根据用户请求进行响应。



**和Session方式存储id的差异**

Session方式存储用户id的最大弊病在于要占用大量服务器内存，对于较大型应用而言可能还要保存许多的状态。一般而言，大型应用还需要借助一些KV数据库（me:redis）和一系列缓存机制(me:memcache)来实现Session的存储。

而JWT方式将用户状态分散到了客户端中，可以明显减轻服务端的内存压力。除了用户id之外，还可以存储其他的和用户相关的信息，例如该用户是否是管理员、用户所在的分桶（见[《你所应该知道的A/B测试基础》一文](/2015/08/27/introduction-to-ab-testing/）等。

虽说JWT方式让服务器有一些计算压力（例如加密、编码和解码），但是这些压力相比磁盘I/O而言或许是半斤八两。具体是否采用，需要在不同场景下用数据说话。

**单点登录**

Session方式来存储用户id，一开始用户的Session只会存储在一台服务器上。对于有多个子域名的站点，每个子域名至少会对应一台不同的服务器(me：负载均衡也导致了存在多个服务器)，例如：

* www.taobao.com
* nv.taobao.com
* nz.taobao.com
* login.taobao.com

所以如果要实现在login.taobao.com登录后，在其他的子域名下依然可以取到Session，这要求我们在多台服务器上同步Session（me：也可以多个服务器使用一个用户认证系统，然后返回用户端token）。

使用JWT的方式则没有这个问题的存在，因为用户的状态已经被传送到了客户端(me：由此可见和单一认证系统是一样的，都是将用户信息放到了客户端，客户端请求服务器对客户端信息进行验证)。因此，我们只需要将含有JWT的Cookie的domain设置为顶级域名即可，例如

|  |  |
| --- | --- |
| 1 | Set-Cookie: jwt=lll.zzz.xxx; HttpOnly; max-age=980000; domain=.taobao.com |

注意domain必须设置为一个点加顶级域名，即.taobao.com。这样，taobao.com和\*.taobao.com就都可以接受到这个Cookie，并获取JWT了。

对于JWT的两篇文章有相关问题的同学请直接在下面的评论区与我讨论（请勿邮件讨论）。如果你感兴趣，你可以在下方订阅我的半月刊，我将给你推送更多精彩的内容；）