[TKeyword，专注技术干货聚合推荐](http://www.itkeyword.com/)

* [**首页**](http://www.itkeyword.com/)
* [Spring](http://www.itkeyword.com/tag/spring)
* [Mysql](http://www.itkeyword.com/tag/mysql)
* [Linux](http://www.itkeyword.com/tag/linux)
* [分布式](http://www.itkeyword.com/tag/%E5%88%86%E5%B8%83%E5%BC%8F)
* [设计模式](http://www.itkeyword.com/tag/%E8%AE%BE%E8%AE%A1%E6%A8%A1%E5%BC%8F)

[注册](javascript:void(0);) | [登录](javascript:void(0);)

单点登录（一）：原理分析

u014077165 分享于 2015-01-10

推荐：[CAS实现SSO单点登录原理](http://www.itkeyword.com/doc/5513842875378545424/cas" \t "_blank)

up vote 1 down vote favorite I find myself running the same terminal commands when I turn on the laptop. I press ctrl+alt+t to open a new terminal win

[2019阿里云全部产品优惠券(新购或升级都可以使用，强烈推荐)](https://promotion.aliyun.com/ntms/yunparter/invite.html?userCode=3uju1ok9" \o "阿里云优惠券" \t "_blank)  
**领取地址**：[https://promotion.aliyun.com/ntms/yunparter/invite.html](https://promotion.aliyun.com/ntms/yunparter/invite.html?userCode=3uju1ok9" \o "阿里云优惠券" \t "_blank)

**一.单点登录概念**

**1.1 概念**

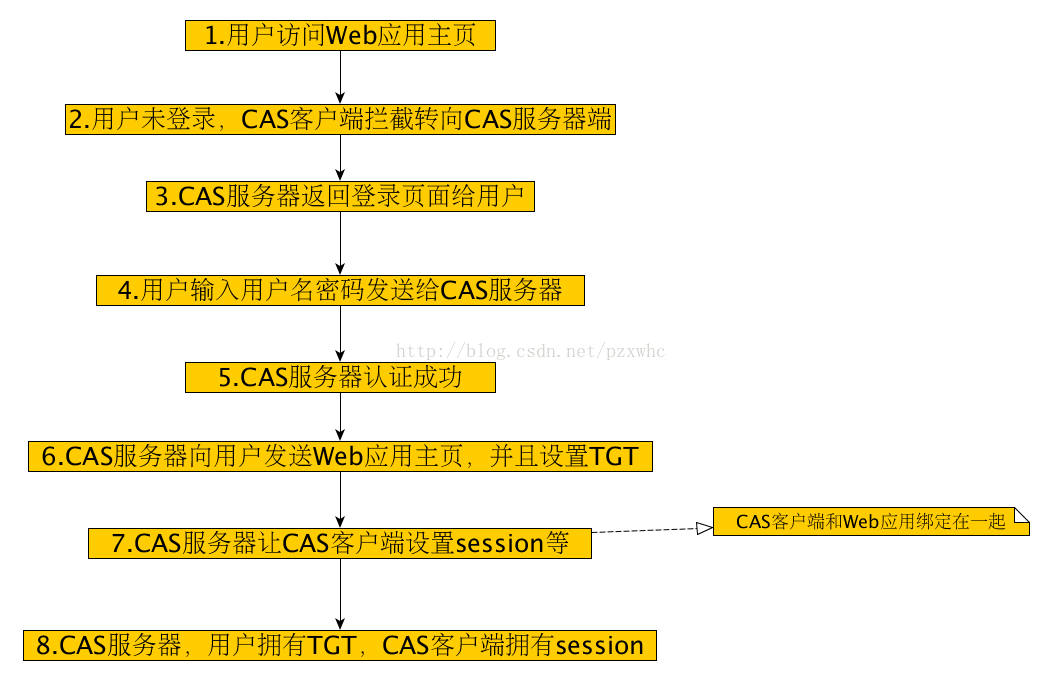
**1.2 SSO组成**

**1.3 技术实现思路**

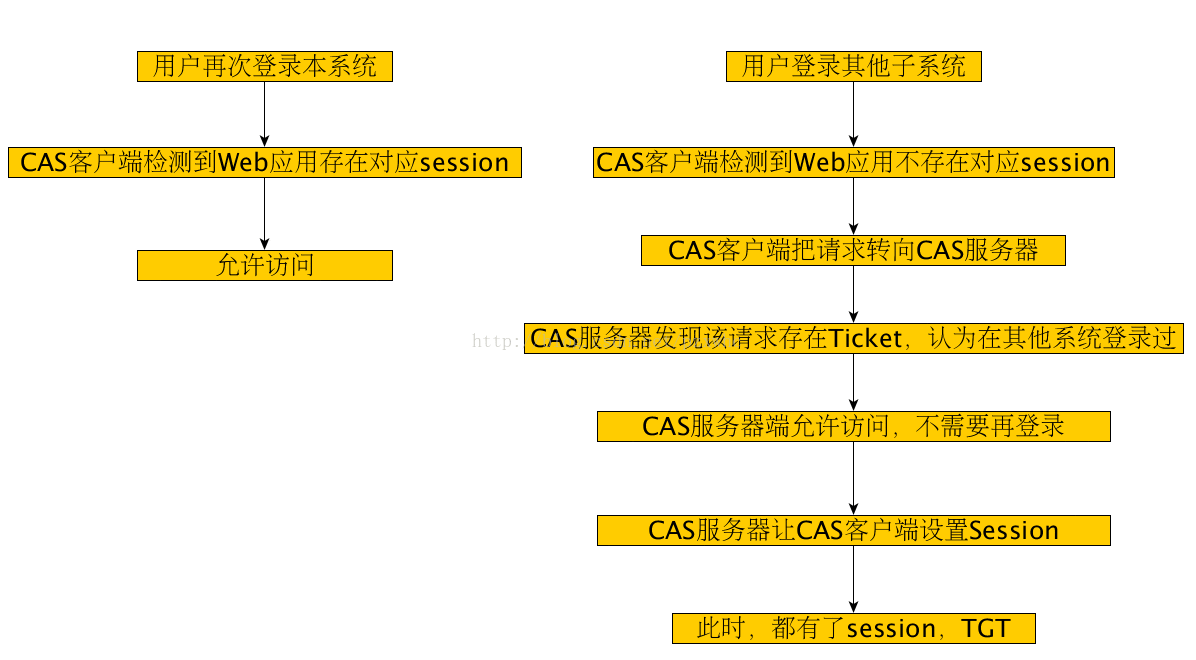
**1.4单点登录过程** 

**一.单点登录概念  
1.1 概念**  
单点登录，简称sso，single sign on主要用于多系统的集成，即在多个系统当中，用户只要用**到一个中央服务器登录一次**即可访问这些系统中的任何一个，无需多次登录。是目前比较流行的企业业务整合的解决方案之一。   
**1.2 SSO组成**   
一般SSO体系主要有3个 **(1)多个用户 (2)多个Web应用 (3)SSO认证中心 1个（也就是我们的中央服务器）**。其中多个用户访问不同的Web应用，是否需要登录由SSO认证中心来控制。     
**1.3 技术实现思路：**   
1.当用户第一次访问系统A的时候，因为还没有登录，会被引导到认证系统中进行登录；   
2.根据用户提供的登录信息，认证系统进行身份校验，如果通过校验，应该返回给用户一个认证的凭据－－ticket；   
3.用户再访问别的应用的时候，就会将这个ticket带上，作为自己认证的凭据。   
4.应用系统接受到请求之后会把ticket送到认证系统进行校验，检查ticket的合法性。   
5.如果通过校验，用户就可以在不用再次登录的情况下访问应用系统2和应用系统3了。**1.4单点登录过程**   
CAS（Central Authentication Service）按照字面的意思就是中央认证服务。主要是为Web应用系统提供一种可靠的单点登录解决方案。   
CAS主要分为两个部分： **CAS server和CAS client 也就是CAS的服务器端和CAS的客户端**。同时CAS服务器端需要独立部署，CAS服务器会处理用户名/密码等凭证。   
SSO访问流程主要有以下的形式：   
1.访问服务：CAS客户端发送请求访问应用系统的服务资源。   
2.定向认证：CAS客户端会重定向用户请求到CAS服务器 。     
3.用户认证：用户身份验证。   
4.发放票据：CAS服务器会产生一个随机的Service Ticket，并且允许客户端访问。   
6.传输用户信息：CAS服务器传输用户认证结果信息给客户端。 

例如一个用户还没有登录过任何的系统，如图：



此时用户再登录系统（可能是本系统，可能是其他系统）：



**文字再解释这个过程：**

CAS Client与受保护的客户端应用部署在一起，以Filter方式保护Web应用的受保护资源，**过滤从客户端发来的每一个请求**。   
同时，CAS Client会分析HTTP请求中是否包含请求Service Ticket，如果没有，则说明这个用户是没有经过认证的；于是，CAS Client会重定向用户请求到CAS Server，并且传递Service（要访问的目的资源地址）。   
用户认证过程，如果用户提供了正确的Credenties（输入了正确的帐号密码），CAS Server随机生成一个相当长度，唯一，不可伪造的Server Ticket，并且 **缓存以待将来认证**。并且重定向用户到Service所在地址（附带刚才产生的Service Ticket），**并且为客户端浏览器设置一个Ticket Granted Cookie（TGC）**。   
**另外的一种解释：**   
Step1: 用户第一次访问Biz系统主页http://bizserver/index.jsp ;部署在Biz系统上的CASFilter发现用户尚未登录，将用户重定向的CAS登录界面 https://casserver/cas/servlet/login?service=http://bizserver/index.jsp ，同时在重定向的URL上用service参数将用户的目标地址传给CAS服务器。   
  
Step2:用户在CAS的登录页上输入用户名密码登录，CAS服务器认证通过后，生成一个ticket，并带在目标地址的尾部返回客户端的浏览器redirect：http://bizserver/index.jsp?ticket=casticket.   
  
Step3:客户端浏览器获得CAS服务器的认证应答，取得凭证ticket后，使用重定向的链接http://bizserver/index.jsp?ticket=casticket访问Biz服务   
  
Step4: BizServer上的CASFilter再次过滤访问请求，并获得ticket凭证。Filter将使用该凭证通过URL https://casserver/cas/servlet/validate?service= http://bizserver/index.jsp &ticket=casticket 向CAS认证中心确认对应的服务请求和凭证是否有效。根据CAS服务器返回的结果，如果凭证有效，则CASFilter允许用户进入http://bizserver/index.jsp 所指向的页面；否则，再次重定向到https://casserver/cas/servlet/login?service=http://bizserver/index.jsp 上要求用户进行认证。   
  
**补充：**   
**CAS** **的SSO实现方式** **可以理解为：1个Cookie和N个Session。**CAS Server创建cookie，在所有应用认证时使用，各应用通过创建各自的Session来标识用户是否已经登录。   
用户在一个应用验证通过后，以后用户在同一个浏览器中访问此应用的时候，客户端应用中的过滤器会在Session中读取用户的信息，所以就不会去CAS Server认证。如果在此浏览器访问别的Web应用的时候，客户端应用中的过滤器在session中读取不到用户信息，就会去CAS Server的login接口认证，但是这个时候CAS Server会读取到浏览器传来的cookie（TGC），所以CAS Server不会要求用户去登录页面登录。   
  
**术语解释：**   
1.TGT（Ticket Grangting Ticket）：TGT是CAS为用户签发的登录票据，拥有了TGT，用户就可以证明自己在CAS成功登录过。TGT封装了Cookie值以及此Cookie值对应的用户信息。用户在CAS认证成功后， **CAS生成cookie（叫TGC），写入浏览器，同时生成一个TGT对象，放入自己的缓存**，TGT对象的ID就是cookie的值。当HTTP再次请求到来时，如果传过来的有CAS生成的cookie，则CAS以此cookie值为key查询缓存中有无TGT ，如果有的话，则说明用户之前登录过，如果没有，则用户需要重新登录。 2.TGC（Ticket-granting cookie）：**存放用户身份认证凭证的cookie**，在浏览器和CAS Server间通讯时使用，并且只能基于安全通道传输（Https），是CAS Server用来明确用户身份的凭证。   
3.ST（Service Ticket）：ST是CAS为用户签发的访问某一service的票据。用户访问service时，service发现用户没有ST，则要求用户去CAS获取ST。用户向CAS发出获取ST的请求，如果用户的请求中包含cookie，则CAS会以此cookie值为key查询缓存中有无TGT，如果存在TGT，则用此TGT签发一个ST，返回给用户。用户凭借ST去访问service，service拿ST去CAS验证，验证通过后，允许用户访问资源。   
4.PGT：绑定一个用户的特定服务，使其拥有向CAS Server申请，获得PT的能力。4.PGT（Proxy Granting Ticket）：Proxy Service的代理凭据。用户通过CAS成功登录某一Proxy Service后，CAS生成一个PGT对象，缓存在CAS本地，同时将PGT的值（一个UUID字符串）回传给Proxy Service，并保存在Proxy Service里。Proxy Service拿到PGT后，就可以为Target Service（back-end service）做代理，为其申请PT。   
5.PGTIOU(Proxy Granting Ticket I Owe You)；PGTIOU是CAS协议中定义的一种附加票据，它增强了传输、获取PGT的安全性。   
PGT的传输与获取的过程：Proxy Service调用CAS的serviceValidate接口验证ST成功后，CAS首先会访问pgtUrl指向的https url，将生成的 PGT及PGTIOU传输给proxy service，proxy service会以PGTIOU为key，PGT为value，将其存储在Map中；然后CAS会生成验证ST成功的xml消息，返回给Proxy Service，xml消息中含有PGTIOU，proxy service收到Xml消息后，会从中解析出PGTIOU的值，然后以其为key，在map中找出PGT的值，赋值给代表用户信息的Assertion对象的pgtId，同时在map中将其删除。   
6.PT(Proxy Ticket)： PT是用户访问Target Service（back-end service）的票据。如果用户访问的是一个Web应用，则Web应用会要求浏览器提供ST，浏览器就会用cookie去CAS获取一个ST，然后就可以访问这个Web应用了。如果用户访问的不是一个Web应用，而是一个C/S结构的应用，因为C/S结构的应用得不到cookie，所以用户不能自己去CAS获取ST，而是通过访问proxy service的接口，凭借proxy service的PGT去获取一个PT，然后才能访问到此应用。   
  
TGT、ST、PGT、PT之间关系：   
1）ST是TGT签发的。用户在CAS上认证成功后，CAS生成TGT，用TGT签发一个ST，ST的ticketGrantingTicket属性值是TGT对象，然后把ST的值redirect到客户应用。   
2）PGT是ST签发的。用户凭借ST去访问Proxy service，Proxy service去CAS验证ST（同时传递PgtUrl参数给CAS），如果ST验证成功，则CAS用ST签发一个PGT，PGT对象里的ticketGrantingTicket是签发ST的TGT对象。   
3）PT是PGT签发的。Proxy service代理back-end service去CAS获取PT的时候，CAS根据传来的pgt参数，获取到PGT对象，然后调用其grantServiceTicket方法，生成一个PT对象。 

推荐：[CAS单点登录登出原理](http://www.itkeyword.com/doc/6673631645686805877/cas" \t "_blank)

http://ttaale.iteye.com/blog/1166674 CAS单点登录原理 利用之前的角色及上面的用例，交互可以表示为如下步骤 1.用户访问application1,application1查看session

一.单点登录概念         1.1 概念     1.2 SSO组成     1.3 技术实现思路     1.4单点登录过程 一.单点登录概念 1.1 概念 单点登录，简称sso，single sign on主要用于多系统的集成，即在多个系

[](https://promotion.aliyun.com/ntms/act/qwbk.html?userCode=3uju1ok9)

相关阅读排行

* 1[基于Sip的P2P设计和原理分析](http://www.itkeyword.com/doc/2534365381696511605/sap-traversal-p2p" \o "基于Sip的P2P设计和原理分析" \t "_blank)
* 2[NAT路由器“打洞”技术，即P2P通信实现原理（非常详细）](http://www.itkeyword.com/doc/0350227234472688540" \o "NAT路由器\“打洞\”技术，即P2P通信实现原理（非常详细）" \t "_blank)
* 3[AudioQueue(AudioToolBox)的原理与使用](http://www.itkeyword.com/doc/6919081426359148761/audioqueue-ios" \o "AudioQueue(AudioToolBox)的原理与使用" \t "_blank)
* 4[编译原理——语言处理程序](http://www.itkeyword.com/doc/4482224087533879x141" \o "编译原理——语言处理程序" \t "_blank)
* 5[C指针原理（78)-递归(2)](http://www.itkeyword.com/doc/8844595873615710x735" \o "C指针原理（78)-递归(2)" \t "_blank)

用户评论

http://res.itkeyword.com/images/itkeyword/no-avatar.png

游客



[发表](javascript:;)

[](https://www.aliyun.com/acts/hi-group-buying?userCode=3uju1ok9)

相关内容推荐

* 1[微信公众号文章采集,并发布到WordPress](http://www.keydatas.com/doc/fuyYRzrY7vy2/weixin-gongzhonghao-caiji-wordpress" \o "微信公众号文章采集,并发布到WordPress" \t "_blank)
* 2[CAS实现SSO单点登录原理](http://www.itkeyword.com/doc/5513842875378545424/cas" \o "CAS实现SSO单点登录原理" \t "_blank)
* 3[CAS单点登录登出原理](http://www.itkeyword.com/doc/6673631645686805877/cas" \o "CAS单点登录登出原理" \t "_blank)
* 4[单点登录SSO的实现原理](http://www.itkeyword.com/doc/6360800729952232x561/sso" \o "单点登录SSO的实现原理" \t "_blank)
* 5[CAS 实现SSO单点登录过程及原理](http://www.itkeyword.com/doc/2518112878782597x659/sso-cas" \o "CAS 实现SSO单点登录过程及原理" \t "_blank)
* 6[采用CAS原理构建单点登录](http://www.itkeyword.com/doc/3115404666955390x234/cas" \o "采用CAS原理构建单点登录" \t "_blank)

最新文章

* 1[深入剖析基于并发AQS的重入锁(ReetrantLock)及其Condition实现原理](http://www.itkeyword.com/doc/3504492243724924135/lock-condition-aqs" \o "深入剖析基于并发AQS的重入锁(ReetrantLock)及其Condition实现原理" \t "_blank)
* 2[[置顶] 滴滴开源Android插件化框架VirtualAPK原理分析](http://www.itkeyword.com/doc/4074282075277667x723/android-virtualapk" \o "[置顶] 滴滴开源Android插件化框架VirtualAPK原理分析" \t "_blank)
* 3[Android深入理解JNI（一）JNI原理与静态、动态注册](http://www.itkeyword.com/doc/1590951185929542x862/jni-android" \o "Android深入理解JNI（一）JNI原理与静态、动态注册" \t "_blank)
* 4[Kafka学习笔记——Kafka原理与使用详解](http://www.itkeyword.com/doc/2281542918972358275/kafka" \o "Kafka学习笔记——Kafka原理与使用详解" \t "_blank)
* 5[HTTP响应报文与工作原理详解](http://www.itkeyword.com/doc/6335849286613797027/http" \o "HTTP响应报文与工作原理详解" \t "_blank)
* 6[安卓自定义View进阶-Matrix原理](http://www.itkeyword.com/doc/5689296178522024293/android-matrix-gcssloop" \o "安卓自定义View进阶-Matrix原理" \t "_blank)

ITKeyword

[关于我们](http://www.itkeyword.com/aboutUs.html" \o "关于我们" \t "_blank) |[联系我们](http://www.itkeyword.com/contactUs.html" \o "联系我们" \t "_blank)

友情链接

[阿里云服务器购买](http://www.itkeyword.com/html/aliyun-fuwuqi-goumai.html" \o "阿里云服务器购买" \t "_blank) [阿里云代金券](http://www.itkeyword.com/aliyun" \o "阿里云代金券" \t "_blank) [阿里云代金券](http://www.aliyun-youhui.com/" \o "阿里云代金券" \t "_blank)[博客园](https://www.cnblogs.com/" \o "博客园" \t "_blank) [Iteye](http://www.iteye.com/" \o "iteye" \t "_blank) [CSDN](http://www.csdn.com/" \o "csdn" \t "_blank) [cnblogs](http://www.cnblogs.com/" \o "cnblogs" \t "_blank)[Stackoverflow](http://www.stackoverflow.com/" \o "stackoverflow" \t "_blank) [Springsource](http://www.springsource.org/" \o "Springsource" \t "_blank)

本网站部分文章来自互联网，对于此类文章本站仅提供相关推荐和交流平台，不为其版权承担责任。如果您发现本网站上有侵犯您的知识产权的信息(文字或图片)，请发送邮件至:itkeyword@163.com通知我们，予以删除。

Copyright © 2013-2016,[ITKeyword.com](http://www.itkeyword.com/) All Rights Reserved 备案号：粤ICP备16069336号



