自从CAS 3.4就很好的支持了单点注销功能，配置也很简单。

之前版本因为在CAS服务器通过HttpClient发送消息时并未指定为POST方式，所以在CAS客户端的注销Filter中没有收到POST请求（要知道Filter只对Post请求起作用），也就没有做session销毁处理。

两个业务系统APP1和APP2

在没有配置单点退出时，效果是这样子的

1：登录APP1，然后经过CAS认证后进入APP1，再访问APP2无需要认证

2：在APP1中连接到cas的logout地址，现象注销成功界面，然后再访问APP1，还是可以进去的，因为APP1将用户的登录票据存入了session。

那么实现了单点退出后的效果应该是这样子的：

1：登录APP1，然后经过CAS认证后进入APP1，再访问APP2无需要认证

2：用户在APP1或者APP2点击注销，显示CAS的注销成功页面，然后再访问APP1或者APP2都需要再次认证。

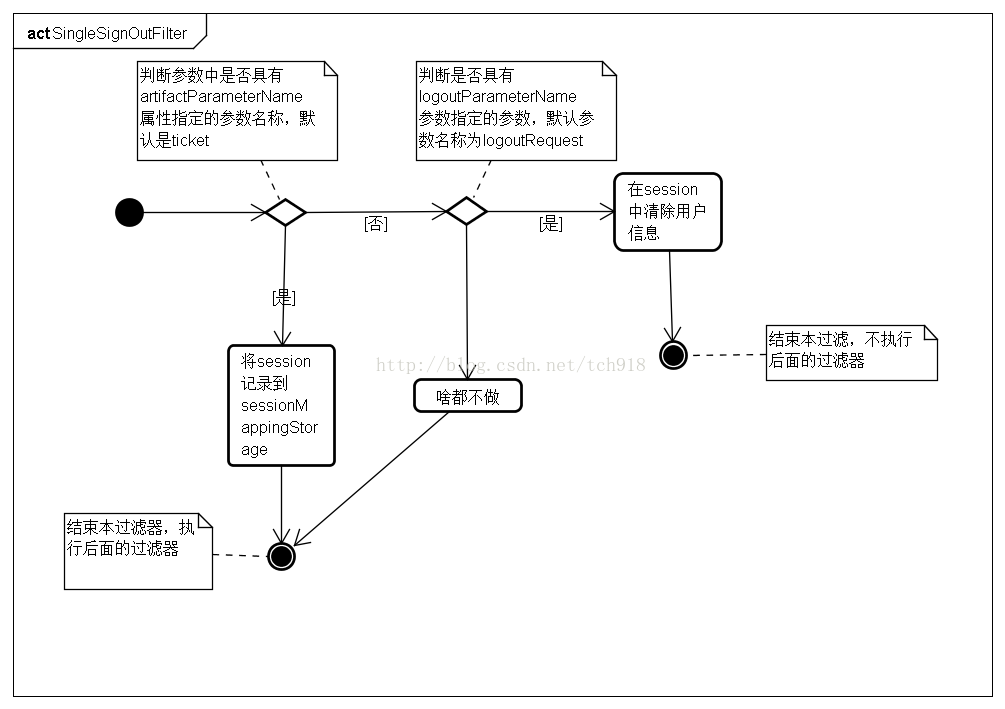
具体配置为，在APP1和APP2的web.xml文件中增加：

1. **<listener>**
2. **<listener-class>**org.jasig.cas.client.session.SingleSignOutHttpSessionListener**</listener-class>**
3. **</listener>**
5. **<filter>**
6. **<filter-name>**CAS Single Sign Out Filter**</filter-name>**
7. **<filter-class>**org.jasig.cas.client.session.SingleSignOutFilter**</filter-class>**
8. **</filter>**
9. **<filter-mapping>**
10. **<filter-name>**CAS Single Sign Out Filter**</filter-name>**
11. **<url-pattern>**/\***</url-pattern>**
12. **</filter-mapping>**

注销的Filter要在其它Filter之前，界面的注销连接到CAS的logout地址，如[http://localhost:8080/cas/logout](http://localhost:8080/cas/logout" \t "_blank)

SingleSignOutFilter，主要是在有ticket参数的时候，将session放到sessionMappingStorage，如果参数中存在logoutRequest，则注销session，那什么时候去注销sessionMappingStorage的东西呢？这是靠SingleSignOutHttpSessionListener来实现的，当有session被销毁的时候，触发将sessionMappingStorage中对应sessionid中的数据删除。

所以在配置单点登出的时候，一定要配置这个监听器，否则客户端很容易导致内存溢出的。让我们先来看看SingleSignOutFilter的整体逻辑。



那么这个是在什么时候会触发呢，这个是在你登陆的任意客户端，调用<https://localhost:8080/logout>，这个取得cookie里面的TGT数据，找到TGT中关联的所有ST对应的地址（即多个cas client），向每个地址方式一个http请求，并传递logoutRequest参数。

来看看源代码是怎么实现的：

**[java]** [view plain](http://blog.csdn.net/tch918/article/details/22276627) [copy](http://blog.csdn.net/tch918/article/details/22276627)

[在CODE上查看代码片](https://code.csdn.net/snippets/260719)

1. **public** **void** doFilter(**final** ServletRequest servletRequest, **final** ServletResponse servletResponse, **final** FilterChain filterChain) **throws** IOException, ServletException {
2. // 转换参数
3. **final** HttpServletRequest request = (HttpServletRequest) servletRequest;
4. //判断参数中是否具有artifactParameterName属性指定的参数名称，默认是ticket
5. **if** (handler.isTokenRequest(request)) {
6. // 如果存在，在本地sessionMappingStorage中记录session。
7. handler.recordSession(request);
8. } **else** **if** (handler.isLogoutRequest(request)) {//判断是否具有logoutParameterName参数指定的参数，默认参数名称为logoutRequest
9. // 如果存在，则在sessionMappingStorage中删除记录，并注销session。
10. handler.destroySession(request);
11. // 注销session后，立刻停止执行后面的过滤器
12. **return**;
13. } **else** {
14. log.trace("Ignoring URI " + request.getRequestURI());
15. }
16. //条件都不满足，继续执行下面的过滤器
17. filterChain.doFilter(servletRequest, servletResponse);
18. }

如果直接访问CAS的logout话，会出现注销成功页面，其实大部分情况下这个页面是没有必要的，更多的需求可能是退出后显示登录页面，并且登录成功后还是会进入到之前的业务系统，那么可以修改cas-servlet.xml文件，在"logoutController"的bean配置中增加属性“followServiceRedirects”，设置为“true”，然后在业务系统的注销连接中加入"service参数",值为业务系统的绝对URL，这样就OK了，如你的业务系统URL为：<http://localhost:8080/casClient>，那么注销URL就为：<http://localhost:8080/cas/logout?service=http://localhost:8080/casClient>

如果出现这种现象：访问过<http://localhost:8080/cas/logout>之后不关浏览器，还是能访问我的应用

可能因为：

1：你的CAS服务器将cookie设置成了浏览器有效，那么表示如果浏览器不关闭，则一直有效。

在WEB-INF/[**spring**](http://lib.csdn.net/base/17)-configuration/ticketGrantingTicketCookieGenerator.xml中设置cookie有效期，默认配置cookie有效期为-1

2：你的应用中注销的filter-mapping没有放在所有mapping之前

原文出处：http://blog.csdn.net/tch918/article/details/22276627