

一、CAS 服务端的处理逻辑

CAS 服务端总共对外暴露了 7 个接口，客户端通过访问这 7 个接口与服务端交互，这 7 个接口为：/login、/logout、/validate、/serviceValidate、/proxy、/proxyValidate、/CentralAuthenticationService /login 是认证接口，/logout 是退出接口，负责销毁认证 cookie,/validate、/serviceValidate 是验证 ticket 用的接口，其中/validate 是 CAS1.0 定义的，/serviceValidate 是 CAS2.0 定义的，其中/serviceValidate返回 xml 格式的数据，/proxy、/proxyValidate 是支持代理认证功能的接口，/CentralAuthenticationService接口用于和远程的 webservices 交互。对于一般web 应用的单点登录来讲,/login 、/logout 、/serviceValidate 这 3 个接口已经可以满足要求 下面是我对 CAS 各个接口实现的的详细说明。

/login:

登录流程这部分要考虑到不同种类用户凭证的获取方案，以及客户应用传来的 service、gateway、renew 参数的不同取值组合，CAS 为了实现流程的高度可配置性，采用了 SpringWebFlow 技术。通过阅读 CAS 发布包里的 login-webflow.xml cas-servlet.xml applicationContext.xml这 3 个文件，找到登录有关的所有组件如下图：

如果成功创建 **ServiceTicket**, 则转发到 **WarnAction**, 如果创建失败, 且 **gateway** 参数为 **true**, 则直接 **redirect** 到客户应用, 否则则需要重新认证。

3: **viewLoginForm** 这是登录页面, **CAS** 在此收集用户凭证。**CAS** 提供的默认实现是 **/WEB-INF/view/jsp/simple/ui/casLoginView.jsp**

4: **bindAndValidate** 对应 **AuthenticationViaFormAction** 的 **doBind** 方法, 该方法负责搜集登录页面上用户录入的凭证信息 (用户名、密码等), 然后把这些信息封装到 **CAS** 内部的 **Credentials** 对象中。用户在 **casLoginView.jsp** 页面上点击提交后, 会触发此方法。

5: **submit** 对应 **AuthenticationViaFormAction** 的 **submit** 方法, 如果 **doBind** 方法成功执行完, 则触发 **submit** 方法, 此方法负责调用 **centralAuthenticationService** 的 **grantServiceTicket** 方法, 完成认证工作, 如果认证成功, 则生成 **TicketGrantingTicket** 对象, 放在缓存里, **TicketGrantingTicket** 的 ID 就是 **TGCCookie** 的 **value** 值。

6: **warnCAS** 提供了一个功能: 用户在一个 **web** 应用中跳到另一个 **web** 应用时, **CAS** 可以跳转到一个提示页面, 该页面提示用户要离开一个应用进入另一个应用, 可以让用户自己选择。用户在登录页面 **viewLoginForm** 上选中了 **id="warn"** 的复选框, 才能开启这个功能。

WarnAction 就检查用户有没有开启这个功能, 如果开启了, 则转发到 **showWarnView**, 如果没开启, 则直接 **redirect** 到客户应用。

7: **SendTicketGrantingTicketAction** 此 **Action** 负责为 **response** 生成 **TGCCookie**, **cookie** 的值就是 **AuthenticationViaFormAction** 的 **submit** 方法生成的 **TicketGrantingTicket** 对象的 ID。

8: **viewGenerateLoginSuccess** 这是 **CAS** 的认证成功页面。

/logout: (对应实现类 **org.jasig.cas.web.LogoutController**)

处理逻辑:

1) **removeCookie**

2) 在服务端删除 **TicketGrantingTicket** 对象 (此对象封装了 **cookie** 的 **value** 值)

3) **redirect** 到退出页面, 有 2 种选择:

```
if(LogoutController的 followServiceRedirects属性为 true 值, 且 url 里的 service
参数非空){
redirect 到 sevice 参数标识的 url
}
else{
redirect 到内置的 casLogoutView
    (cas/WEB-INF/view/jsp/default/ui/casLogoutView.jsp), 如果 url 里有 url 参数,
    则此 url 参数标识的链接会显示在 casLogoutView页面上。
}
```

/serviceValidate: (对应实现类 org.jasig.cas.web.ServiceValidateController)

处理逻辑:

如果 service 参数为空或 ticket 参数为空, 则转发到 failureView
(/WEB-INF/view/jsp/default/protocol/2.0/casServiceValidationFailure.jsp
验证 ticket。以 ticket 为参数, 去缓存里找 ServiceTicketImpl 对象, 如果能找到,
且没有过期, 且 ServiceTicketImpl 对象对应的 service 属性和 service 参数对应,
则验证通过, 验证通过后, 请求转发至 casServiceSuccessView
(cas/WEB-INF/view/jsp/default/protocol/2.0/casServiceValidationSuccess.jsp

二 CAS 客户端 Filter 的处理逻辑

1 AuthenticationFilter

```
if(url 中无 ticket 参数 && session 中没有 TicketValidationFilter 置的 assertion 对
象){
response.sendRedirect(cas 服务器的 /login 接口); //生成 service 参数, 添加到
url 后面
}
else{
不做处理
}
```

2 TicketValidationFilter

```
if(url中有 ticket参数){
```

通过 httpclient工具访问 cas 服务器的/serviceValidate接口验证 ticket的有效性，验证失败，显示错误页面，验证成功，则生成标识用户身份的 assertion对象，放入 session。

```
}
```

```
else{
```

不做处理

```
}
```

注：

1AuthenticationFilter在前，TicketValidationFilter在后。

2AuthenticationFilter:

1) url 中无 ticket 参数，且 session中没有 TicketValidationFilter置的 assertion对象，这种情况说明用户还没有认证，AuthenticationFilter会去做认证处理；

2) url 中无 ticket 参数，且 session中有 TicketValidationFilter置的 assertion对象，这种情况说明用户已经认证成功，AuthenticationFilter不做处理；

3) url 中有 ticket 参数，这种情况说明用户已经认证成功，但还需要经TicketValidationFilter去验证 ticket,AuthenticationFilter不做处理。

3TicketValidationFilter: 只有客户端调用 cas 服务器的/login 接口,并成功认证，redirect回客户端时，url 里才带有 ticket 参数，在这种情况下，TicketValidationFilter才做处理。

三、配置php 客户端，下载php 客户端：

<http://downloads.jasig.org/cas-clients/php/> ，目前最新版本为：

CAS-1.2.0RC2

新建 php 工程：Phpcasclient 1将 CAS 文件夹和 CAS.php 复制到工程中，修改 CAS/client.php, 将其中的 https改为 http 将 docs/examples/example_simple.php

复制到工程中，修改如下：

- 1.
2. //
3. //phpCASsimpleclient

```

4. //
5. //importphpCASlib
6. include_once ( 'CAS.php' );
7. phpCAS::setDebug();
8. //initializ ephpCAS
9. phpCAS::client(CAS_VERSION_2_0, '192.168.18.8' ,8080, 'cas' );
10. //noSSLvalidationfortheCASserver
11. phpCAS::setNoCasServerValidation();
12. //forceCAsauthentication
13. phpCAS::forceAuthentication();
14. //atthisstep,theuserhasbeenauthenticatedbytheCASserver
15. //andtheuser'sloginnamecanbereadwithphpCAS::getUser().
16. //logoutifdesired
17. if (isset( $_REQUEST['logout' ])){
18.
19. $param=array ( "service" =>"http://localhost/Phpcasclient1/example_simple.php" );// 退出
    登录后返回
20. phpCAS::logout( $param);
21.
22. }
23. //forthis test,simplyprintthattheauthenticationwassuccessfull
24. ?>

```

```

1.
2.
3.
4.
5.
6.      SuccessfullAuthentication!   这是客户端
7. theuser'sloginis      echophpCAS::getUser(); ?> .
8. phpCASversionis      echophpCAS::getVersion(); ?> .
9.      href ="http://192.168.18.8:8989/Casclient1/index.jsp"      >去 java 客户端 1
10.      href ="?logout=" >退出
11.
12.

```

php 配置需要开启php_curl,可以复制Phpcasclient1为 Phpcasclient2

访问:http://localhost/Phpcasclient1/example_simple.php,并转到登录页面,登录成功后访问Phpcasclient2
不需要登录,

php单点登录成功，这时再访问java客户端发现也不需要登录，php和java应用之间单点登录成功。

注：php的phpCAS::client(CAS_VERSION_2_0,'192.168.18.8',8080,'cas地址需要和java的web.xml中的cas服务器地址一致，我开始一个写的ip: 192.168.18.8，一个写的localhost

php和java总是不能同步登录，郁闷了好久

-----到这里java和php的客户端已经配置完成，现在你会发现php和java之间不能单点登出，php端退出java客户端也退出，反之java退出但是php却没有同步退出

这里需要做一个配置，在

```
phpCAS::setNoCasServerValidation();
```

```
//forceCAsAuthentication
```

```
phpCAS::forceAuthentication();
```

这里加上

```
phpCAS::setNoCasServerValidation();
```

```
//forceCAsAuthentication
```

phpCAS::handleLogoutRequests()这里会检测服务器端java退出的通知，就能实现php和java间同步登出了。

```
phpCAS::forceAuthentication();
```

至于discuz+supesite的单点登录，再了解了php单点登录的原理后就需要改造discuz+supesite的登录代码了，discuz的为logging.php

supersite的为batch.login.php，俺是做java开发的，对php不是很熟悉，所以改造的觉得不是很靠谱，大致是先让discuz单点登录，获取用户名，根据用户

获取数据库中的密码再交给discuz系统自己的登录系统登录。discuz是采用cookie验证的，所以在java端退出后，discuz不会退出。

您的评论 *感谢支持，给文档评个星吧！

写点评论支持下文档

240

[发布评论](#)

[暂无评论](#)

评价文档：

分享到：

[QQ空间](#)[新浪微博](#) [微信](#)

扫二维码，快速分享到微信朋友圈

文档可以转存到百度网盘啦！

转为pdf格式

转为其他格式 >

VIP专享文档格式自由转换

下载券

立即下载

加入VIP

免券下载