## 环境准备

1. win8.1 64 bit

2. JDK1.7

3. Tomcat-8.0.15

4. cas-server-4.0.0 、cas-client-3.3.3

tomcat服务器需要部署三个，我分别命名为 apache-tomcat-8.0.15-app1、

apache-tomcat-8.0.15-app2、apache-tomcat-8.0.15-cas

分别的用途如下：

序号 服务器名称 用途

1 tomcat-app1 客户端服务器1：用户部署应用app1

2 tomcat-app2 客户端服务器2：用户部署应用app2

3 tomcat-cas cas服务器：用来部署CAS server

CAS 默认认证方式使用的是HTTPS协议，一般对安全性不高的话建议取消改成HTTP方式。因为，开启的话会经常提示证书过期、需要用户确认等，对客户的感知不好，当前有需要的可以开启。

## 第一步、Tomcat修改

1. 解压下载的 Tomcat-8.0.15.zip，并按照第二点的约定复制三个并命名

修改tomcat的相关启动等端口，使机器可以运行多个tomcat。我的访问端口对应如下：

序号 服务器名称 访问端口

1 tomcat-app1 8081

2 tomcat-app2 8082

3 tomcat-cas 18080

端口的修改方法：打开 x:\tomcat-app1\conf\server.xml 文件， 找到

第一个：修改Shutdown端口（默认为8005端口）

<Server port="8005" shutdown="SHUTDOWN">

第二个： 修改http访问端口（默认为8080端口）

<Connector port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

第三个：修改8009端口

<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />

修改成自己需要的端口即可。

## 第二步、CAS-SERVER 服务端的部署

1. 解压下载的 cas-server-4.0.0-release.zip 压缩包

2. 找到 X:\cas-server-4.0.0\modules\cas-server-webapp-4.0.0.war 文件

3. 解压到 tomcat-cas\webapps\ 下

4. 取消HTTPS协议：

1）打开 cas-server\WEB-INF\deployerConfigContext.xml 文件 ，找到如下配置：

<!-- Required for proxy ticket mechanism. -->

<bean id="proxyAuthenticationHandler"

class="org.jasig.cas.authentication.handler.support.HttpBasedServiceCredentialsAuthenticationHandler" p:httpClient-ref="httpClient" />

增加参数p:requireSecure="false"，是否需要安全验证，即HTTPS，false为不采用。修改后为：

<bean id="proxyAuthenticationHandler"

class="org.jasig.cas.authentication.handler.support.HttpBasedServiceCredentialsAuthenticationHandler" p:httpClient-ref="httpClient" p:requireSecure="false" />

2）打开 cas-server\WEB-INF\spring-configuration\ticketGrantingTicketCookieGenerator.xml ，找到如下配置：

<bean id="ticketGrantingTicketCookieGenerator"

class="org.jasig.cas.web.support.CookieRetrievingCookieGenerator"

p:cookieSecure="true"

p:cookieMaxAge="-1"

p:cookieName="CASTGC"

p:cookiePath="/cas" />

修改 p:cookieSecure="true" 为 p:cookieSecure="false"

即不开启https验证

3） 打开 cas-server\WEB-INF\spring-configuration\warnCookieGenerator.xml ，找到如下配置：

<bean id="warnCookieGenerator"

class="org.jasig.cas.web.support.CookieRetrievingCookieGenerator"

p:cookieSecure="true"

p:cookieMaxAge="-1"

p:cookieName="CASPRIVACY"

p:cookiePath="/cas" />

修改 p:cookieSecure="true" 为 p:cookieSecure="false"

即不开启https验证

5.启动 tomcat-cas ，访问 http://localhost:18080/cas-server ，可以看到如下界面

注意：cas-server4.0之前的默认验证规则：只要用户名和密码相同就认证通过

4.0 之后规则改了，默认是配置在 deployerConfigContext.xml 配置文件中，可以看到用户名密码为 casuser/Mellon。

<bean id="primaryAuthenticationHandler"

class="org.jasig.cas.authentication.AcceptUsersAuthenticationHandler">

<property name="users">

<map>

<entry key="casuser" value="Mellon"/>

</map>

</property>

</bean>

## 第三步、客户端（cas-client）的配置

注：我们直接用tomcat自带的examples工程作为客户端例子

1. 解压我们下载的cas-client-3.3.3-release.zip 包，复制

cas-client-3.3.3\modules\cas-client-core-3.3.3.jar 包

1. 放到 tomcat-app1\webapps\examples\WEB-INF\lib 下（两个client tomcat都需要放，这

边只列出其中一个）

3. 修改examples\WEB-INF\web.xml 文件，增加下面的内容：

<!-- ========================单点登录开始 ======================== -->

<!--用于单点退出，该过滤器用于实现单点登出功能，可选配置 -->

<listener> <listener-class>org.jasig.cas.client.session.SingleSignOutHttpSessionListener</listener-class>

</listener>

<!--该过滤器用于实现单点登出功能，可选配置。 -->

<filter>

<filter-name>CASSingle Sign OutFilter</filter-name>

<filter-class>org.jasig.cas.client.session.SingleSignOutFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>CASSingle Sign OutFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<filter>

<filter-name>CASFilter</filter-name>

<filter-class>org.jasig.cas.client.authentication.AuthenticationFilter</filter-class>

<init-param>

<param-name>casServerLoginUrl</param-name>

<param-value>http://localhost:18080/cas-server/login</param-value>

</init-param>

<init-param>

<param-name>serverName</param-name>

<param-value>http://localhost:8081</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>CASFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!--该过滤器负责对Ticket的校验工作，必须启用它 -->

<filter>

<filter-name>CASValidationFilter</filter-name>

<filter-class>

org.jasig.cas.client.validation.Cas20ProxyReceivingTicketValidationFilter

</filter-class>

<init-param>

<param-name>casServerUrlPrefix</param-name>

<param-value>http://localhost:18080/cas-server</param-value>

</init-param>

<init-param>

<param-name>serverName</param-name>

<param-value>http://localhost:8081</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>CASValidationFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!-- 该过滤器负责实现HttpServletRequest请求的包裹， 比如允许开发者通过HttpServletRequest的getRemoteUser()方法获得SSO登录用户的登录名，可选配置。 -->

<filter>

<filter-name>CASHttpServletRequest WrapperFilter</filter-name>

<filter-class>

org.jasig.cas.client.util.HttpServletRequestWrapperFilter

</filter-class>

</filter>

<filter-mapping>

<filter-name>CASHttpServletRequest WrapperFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!-- 该过滤器使得开发者可以通过org.jasig.cas.client.util.AssertionHolder来获取用户的登录名。 比如AssertionHolder.getAssertion().getPrincipal().getName()。 -->

<filter>

<filter-name>CASAssertion Thread LocalFilter</filter-name>

<filter-class>org.jasig.cas.client.util.AssertionThreadLocalFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>CASAssertion Thread LocalFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!-- ========================单点登录结束 ======================== -->

4.启动 tomcat-app1 ，然后访问 http://localhost:8081/examples ，页面会跳到

http://localhost:18080/cas-server/login?service=http%3A%2F%2Flocalhost%3A8081%2Fexamples%2F

说明单点成功

同样的 tomcat-app2 我这边就不演示了，

我把修改完的两个web.xml 上传上来

tomcat-app1 : web.xml

tomcat-app2 : web.xml

## 第四步、单点流程演示

我们首先依次把三个tomcat都启动后 ，我们先单独访问两个客户端看看效果

1. 访问 http://localhost:8081/examples ==》 跳到

http://localhost:18080/cas-server/login?service=http%3A%2F%2Flocalhost%3A8081%2Fexamples%2F

1. 访问 http://localhost:8082/examples ==》 跳到

http://localhost:18080/cas-server/login?service=http%3A%2F%2Flocalhost%3A8082%2Fexamples%2F

说明两个客户端第一次访问时都需要跳转到cas-server进行认证

接下来：我们登录其中一个客户端 http://localhost:8081/examples ，账号密码

casuser/Mellon

登录成功后,我们在打开一个新选项卡 ，直接访问 http://localhost:8082/examples

可以看到不会跳转到cas-server登录界面

说明 两个客户端单点登录成功，登录了其中一个，另一个不需要登陆即可进行访问。