**单点登录配置说明**

# 1 配置证书

将证书导入到tomcat所使用的jre证书文件中，可以使用命令导入或替换本地jre证书文件。

1. 命令导入

使用命令keytool将证书导入本地jre的证书库：

keytool –import -keystore "C:\ProgramFiles\Java\jdk1.6.0\_34\jre\lib\security\cacerts" -file D:\keys\server.crt -alias slimsmart

输入keystore密码：changeit

注意：导入证书时要确认证书的路径，否则会报错：

PKIX：unable to find valid certification path to requested target

slimsmart：为证书别名

关于证书的生成参考：<http://blog.csdn.net/zhu_tianwei/article/details/19160337>

1. 证书替换

将已经导入好的jre证书库替换本地jre下的证书库文件cacerts（替换前请做好备份）。

# 2 客户端配置

客户端可以使用统一认证平台提供的统一登录页面，也可以自己定义登录页面进行登录认证。具体配置如下：

## 2.1 统一登陆页配置

1. 引入客户端jar包

<dependency>

<groupId>org.jasig.cas.client</groupId>

<artifactId>cas-client-core</artifactId>

<version>3.2.1</version>

</dependency>

1. 配置web.xml

<!-- 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>CAS Single Sign Out Filter</filter-name>

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

</filter>

<filter-mapping>

<filter-name>CAS Single Sign Out Filter</filter-name>

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

</filter-mapping>

<!-- 该过滤器负责用户的认证工作，必须启用它 -->

<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>https://sso.slimsmart.cn:8443/cas/login</param-value>

</init-param>

<init-param>

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

<param-value>http://www.slimsmart.cn:8888</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>CAS Validation Filter</filter-name>

<filter-class>

org.jasig.cas.client.validation.Cas20ProxyReceivingTicketValidationFilter

</filter-class>

<init-param>

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

<param-value>https://sso.slimsmart.cn:8443/cas</param-value>

</init-param>

<init-param>

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

<param-value>http://www.slimsmart.cn:8888</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>CAS Validation Filter</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>CAS HttpServletRequest Wrapper Filter</filter-name>

<filter-class>

org.jasig.cas.client.util.HttpServletRequestWrapperFilter

</filter-class>

</filter>

<filter-mapping>

<filter-name>CAS HttpServletRequest Wrapper Filter</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>CAS Assertion Thread Local Filter</filter-name>

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

</filter>

<filter-mapping>

<filter-name>CAS Assertion Thread Local Filter</filter-name>

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

</filter-mapping>

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

1. 退出URL链接

<a href="https://sso.slimsmart.cn:8443/cas/logout?service= http://www.slimsmart.cn:8888/demo">单点登出</a>

## 2.2 自定义登陆页配置

1. 引入客户端jar包

<dependency>

<groupId>cn.slimsmart.cas.client</groupId>

<artifactId>slim-cas-client-support</artifactId>

<version>1.0.0</version>

</dependency>

1. 配置web.xml

<!-- 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>cn.slimsmart.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>CAS Single Sign Out Filter</filter-name>

<filter-class> cn.slimsmart.cas.client.session.SingleSignOutFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>CAS Single Sign Out Filter</filter-name>

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

</filter-mapping>

<!-- 该过滤器负责用户的认证工作，必须启用它 -->

<filter>

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

<filter-class> cn.slimsmart.cas.client.authentication.AuthenticationFilter</filter-class>

<init-param>

<param-name>localLoginUrl</param-name>

<param-value>http://www.slimsmart.cn:8888/demo/login.jsp</param-value>

</init-param>

<init-param>

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

<param-value>https://sso.slimsmart.cn:8443/cas/remoteLogin</param-value>

</init-param>

<init-param>

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

<param-value>http://www.slimsmart.cn:8888</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>CAS Validation Filter</filter-name>

<filter-class>

cn.slimsmart.cas.client.validation.Cas20ProxyReceivingTicketValidationFilter

</filter-class>

<init-param>

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

<param-value>https://sso.slimsmart.cn:8443/cas</param-value>

</init-param>

<init-param>

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

<param-value>http://www.slimsmart.cn:8888</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>CAS Validation Filter</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>CAS HttpServletRequest Wrapper Filter</filter-name>

<filter-class>

org.jasig.cas.client.util.HttpServletRequestWrapperFilter

</filter-class>

</filter>

<filter-mapping>

<filter-name>CAS HttpServletRequest Wrapper Filter</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>CAS Assertion Thread Local Filter</filter-name>

<filter-class>

org.jasig.cas.client.util.AssertionThreadLocalFilter

</filter-class>

</filter>

<filter-mapping>

<filter-name>CAS Assertion Thread Local Filter</filter-name>

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

</filter-mapping>

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

1. 配置登陆页面

登录form表单:

<form id="myLoginForm" action="https://sso.slimsmart.cn:8443/cas/remoteLogin" method="POST">

<input type="hidden" name="service" value="http://www.slimsmart.cn:8888/demo" />

<input type="hidden" name="loginUrl" value="http://www.slimsmart.cn:8888/demo /login.jsp" />

<input type="hidden" name="submit" value="true" />

<table>

<tr>

<td>用户名:</td>

<td><input type="text" value="" name="username"></td>

</tr>

<tr>

<td>密码:</td>

<td><input type="text" value="" name="password"></td>

</tr>

<tr>

<td>验证码:</td>

<td><input type="text" value="" name="authcode"><img

src="https://sso.slimsmart.cn:8443/cas/captcha.jpg" alt="" /></td>

</tr>

<tr>

<td align="right" colspan="2"><input type="submit" /></td>

</tr>

<tr>

<td align="right" colspan="2">

<div id="errorMsg" style="color: red; font-weight: bold;"></div>

</td>

</tr>

</table>

</form>

登录失败返回提示：

<script type="text/javascript">

**var** errorCode="<%=request.getParameter("errorCode")%>";

**if**(errorCode == 501){

document.getElementById("errorMsg").innerHTML="验证码不能为空";

}**else** **if**(errorCode == 502){

document.getElementById("errorMsg").innerHTML="验证码错误";

}**else** **if**(errorCode == 503){

document.getElementById("errorMsg").innerHTML="用户名或密码不能为空";

}**else** **if**(errorCode == 504){

document.getElementById("errorMsg").innerHTML="用户名或密码错误";

}

</script>

1. 退出URL链接

<a href="https://sso.slimsmart.cn:8443/cas/logout?service= http://www.slimsmart.cn:8888/demo ">单点登出</a>

以上红色字体的链接需要对应修改

https://sso.slimsmart.cn:8443/cas ---》单点登录服务器地址

http://www.slimsmart.cn:8888/demo ---》本地服务器地址

http://www.slimsmart.cn:8888/demo/login.jsp ---》本地服务器自定义登录页面

**注**：登录成功，获取用户名方法：

（1）request直接获取

String username = request.getRemoteUser()

（2）通过接口AttributePrincipal获取

AttributePrincipal principal=(AttributePrincipal) request.getUserPrincipal();

String username = principal.getName();

（3）通过类AssertionHolder静态方法获取

String username =AssertionHolder.getAssertion().getPrincipal().getName()