# CAS-Client

## 1下载cas-client客户端文件

客户端：<http://downloads.jasig.org/cas-clients/>

当前最新客户端版本 cas-client-3.2.1-release.zip

## 2 客户端web项目配置

2.1创建一个web工程。

2.2把解压后的cas-client-3.2.1-release.zip下modules文件夹下的cas-client-core-3.2.1.jar和 commons-logging-1.1.jar文件复制到web工程WEB-INF\lib中。

2.3 修改web.xml文件：

<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.greatsoft.net:8443/cas/login</param-value>

<!--这里的server是服务端的IP -->

</init-param>

<init-param>

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

<param-value>http:// locahost:8080/</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.greatsoft.net:8443/cas</param-value>

</init-param>

<init-param>

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

<param-value>http://locahost:8080/</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>

<!-- 自动根据单点登录的结果设置本系统的用户信息 -->

<filter>

<display-name>AutoSetUserAdapterFilter</display-name>

<filter-name>AutoSetUserAdapterFilter</filter-name>

<filter-class>net.greatsoft.filter.AutoSetUserAdapterFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>AutoSetUserAdapterFilter</filter-name>

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

</filter-mapping>

2.3 编写AutoSetUserAdapterFilter.java文件

package net.greatsoft.filter;

import java.io.IOException;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import javax.servlet.http.HttpServletRequest;

import org.jasig.cas.client.validation.Assertion;

/\*\*

\* 自动根据单点登录系统的信息设置本系统的用户信息

\*

\*/

public class AutoSetUserAdapterFilter implements Filter {

/\*\*

\* Default constructor.

\*/

public AutoSetUserAdapterFilter() {

}

/\*\*

\* @see Filter#destroy()

\*/

public void destroy() {

}

/\*\*

\* 过滤逻辑：首先判断单点登录的账户是否已经存在本系统中，

\* 如果不存在使用用户查询接口查询出用户对象并设置在Session中

\* @see Filter#doFilter(ServletRequest, ServletResponse, FilterChain)

\*/

public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException,

ServletException {

HttpServletRequest httpRequest = (HttpServletRequest) request;

// \_const\_cas\_assertion\_是CAS中存放登录用户名的session标志

Object object = httpRequest.getSession().getAttribute("\_const\_cas\_assertion\_");

if (object != null) {

Assertion assertion = (Assertion) object;

String loginName = assertion.getPrincipal().getName();

System.out.println("当前登录用户：" + loginName);

}

chain.doFilter(request, response);

}

/\*\*

\* @see Filter#init(FilterConfig)

\*/

public void init(FilterConfig fConfig) throws ServletException {

}

}

## 3 配置客户端DNS映射文件：

修改C:\Windows\System32\drivers\etc\hosts文件：

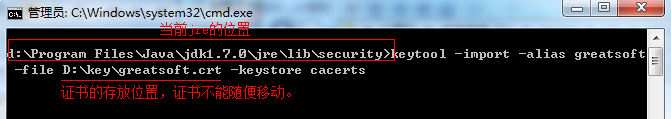
添加一条映射：

192.168.1.179 sso.greatsoft.net

说明：前半部分是CAS-Server的IP地址，后半部分是我们在创建证书时候的名字与姓氏。

## 4 导入证书文件到客户端

把服务器端创建完成的D:\key\greatsoft.crt证书导入到客户端的jre当中。



## 5 启动客户端服务

访问客户端后能够转向<https://sso.greatsoft.net:8443/cas/login>页面，输入配置好的用户名和密码后登陆到Cas-Server端并转向到客户端的首页。