**大兴区数字校园**

**CAS统一认证中心集成技术文档**

**一、适用范围**

大兴区数字校园CAS统一认证中心适用于C/S和B/S架构的应用，针对B/S的Web应用基于Cookie实现认证机制，需要浏览器确保Cookie未禁用。

**二、B／S集成说明**

下述配置以myinfo.daxingedu.cn为例，其他业务系统集成需要修改成自身的访问地址。

**2.1.引入依赖jar包（必须）**

配置单点登录，首先要引入依赖jar包，拷贝cas-client-core-3.2.1.jar到项目的WEB-INF/lib目录下。

**2.2.单点登录配置（必须）**

2.2.1 配置业务系统认证过滤器（必须）

在web.xml文件中加入如下配置：

<filter>

<filter-name>CAS Authentication Filter</filter-name>

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

<init-param>

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

<param-value>http://cas.daxingedu.cn</param-value>

</init-param>

<init-param>

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

<param-value>**http://myinfo.daxingedu.cn**</param-value>

</init-param>

<init-param>

<param-name>renew</param-name>

<param-value>false</param-value>

</init-param>

<init-param>

<param-name>gateway</param-name>

<param-value>false</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>CAS Authentication Filter</filter-name>

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

</filter-mapping>

2.2.2 配置业务系统票据验证过滤器（必须）

在web.xml文件中加入如下配置：

<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>http://cas.daxingedu.cn</param-value>

</init-param>

<init-param>

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

<param-value>**http://myinfo.daxingedu.cn**</param-value>

</init-param>

</filter>

<filter-mapping>

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

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

</filter-mapping>

2.2.3 配置CAS HttpServletRequest过滤器（可选）

在web.xml文件中加入如下配置：

<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>

2.2.4 配置CAS Assertion过滤器（可选）

在web.xml文件中加入如下配置：

<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>

2.2.5 业务系统获取用户信息

单点登录和单点退出配置完成后，在CAS认证通过后会重定向到业务系统中，此时业务系统可以通过request.getRemoteUser()获取登录用户名。

至此，B／S结构的业务系统与CAS集成完毕。

注意：使用request.getRemoteUser()方法获取登录用户名需要配置CAS HttpServletRequest过滤器和CAS Assertion过滤器。

**2.3.单点退出配置（可选）**

2.3.1 配置CAS Server退出地址

在web.xml文件中加入如下配置：

<context-param>

<param-name>casServerLogoutUrl</param-name>

<param-value>**http://cas.daxingedu.cn/logout**</param-value>

</context-param>

2.3.2 配置业务系统访问地址

在web.xml文件中加入如下配置：

<context-param>

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

<param-value>**http://myinfo.daxingedu.cn**</param-value>

</context-param>

2.3.3 配置业务系统退出监听器

在web.xml文件中加入如下配置：

<listener>

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

</listener>

2.3.4 配置业务系统退出过滤器

在web.xml文件中加入如下配置：

<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>

2.3.5 创建退出中间页

在业务系统中退出时访问该界面，例如：cas\_logout\_view.jsp，内容如下：

<%@ page language="java" contentType="text/html; charset=GBK"

    pageEncoding="GBK"%>

<%

Thread.sleep(1000);

session.invalidate();

response.sendRedirect(application.getInitParameter("casServerLogoutUrl") + "?service=" +application.getInitParameter("serverName"));

%>

注意：需要在web.xml配置到其他CAS过滤器之前

**三、C／S集成说明**

以java程序为例，使用apache的httpclient工具模拟客户端网络请求。

**3.1 单点登录示例**

3.1.1 提交用户名和密码获取TGT

// 构造httpclient对象

HttpClient client = new HttpClient();

// 构造postmethod对象，设置TGT获取地址

PostMethod post = new PostMethod("http://cas.daxingedu.cn/v1/tickets");

// 设置请求参数，用户名和密码，假设用户名zhangsan，密码123456

post.setRequestBody(new NameValuePair[] {

new NameValuePair("username", "zhangsan"),

new NameValuePair("password", "123456") });

try {

// 发送请求

client.executeMethod(post);

// 获取请求返回

String response = post.getResponseBodyAsString();

// 处理请求返回

switch (post.getStatusCode()) {

case 201: {

final Matcher matcher =

Pattern.*compile*(".\*action=\".\*/(.\*?)\".\*").matcher

(response);

// 获取CAS Server返回的TGT，红色标注的是TGT。

if (matcher.matches()) {

System.out.println(**matcher.group(1)**);

} else {

*System.out.println(*"Successful ticket granting

request, but no ticket found!");

*System.out.println(*"Response (1k): "

+ response.substring(0, Math.*min*(1024,

response.length())));

break;

}

}

default:

*System.out.println(*"Invalid response code (" +

post.getStatusCode() + ") from CAS server!");

*System.out.println(*"Response (1k): " +

response.substring(0, Math.*min*(1024,

response.length())));

break;

}

} catch (final IOException e) {

*System.out.println(*e.getMessage());

} finally {

// 释放链接

post.releaseConnection();

}

3.1.2 提交TGT获取ST

// 构造httpclient对象

HttpClient client = new HttpClient();

//构造postmethod对象，设置ST获取地址，其中红色标注的为上一步生成的TGT

PostMethod post =

new PostMethod("http://cas.daxingedu.cn/v1/tickets/" +

**TGT**);

// 设置请求参数，业务系统地址，假设用户名http://myinfo.daxingedu.cn/

post.setRequestBody(new NameValuePair[] {

new NameValuePair("service", "http://myinfo.daxingedu.cn/ ") });

try {

// 发送请求

client.executeMethod(post);

// 获取请求返回

String response = post.getResponseBodyAsString();

// 处理请求返回

switch (post.getStatusCode()) {

case 200:

// 获取CAS Server返回的ST，红色标注的是ST。

System.out.println(response);

default:

*System.out.println(*"Invalid response code (" +

post.getStatusCode() + ") from CAS server!");

*System.out.println(*"Response (1k): "

+ response.substring(0, Math.*min*(1024,

response.length())));

break;

}

} catch (final IOException e) {

*System.out.println(*e.getMessage());

} finally {

// 释放链接

post.releaseConnection();

}

3.1.3 传递ST单点登录业务系统

例如：登录myinfo.daxingedu.cn，打开浏览器，输入地址http://myinfo.daxingedu.cn?ticket=ST即可单点登录myinfo系统，其中红色标注的是上一步生成的ST，如下代码模拟登录业务系统：

// 构造httpclient对象

HttpClient client = new HttpClient();

//构造getmethod对象，设置业务系统访问地址

GetMethod method =

new GetMethod("http://myinfo.daxingedu.cn/");

//构造请求参数，设置ticket参数，其中红色标注的为上一步生成的ST

NameValuePair[] valuePair =

new NameValuePair[]{new NameValuePair("ticket", ST)};

method.setQueryString(valuePair);

try {

// 发送请求

client.executeMethod(method);

// 获取请求返回

String response = method.getResponseBodyAsString();

// 处理请求返回

switch (method.getStatusCode()){

case 200:

// 获取业务系统登录成功返回HTML代码。

*System.out.println(*"Response: " + response);

break;

default:

*System.out.println(*"Invalid response code (" +

method.getStatusCode() + ") from CAS server!");

*System.out.println(*"Response: " + response);

break;

}

}catch (final IOException e){

*System.out.println(*e.getMessage());

}finally{

// 释放链接

method.releaseConnection();

}

**3.2 单点退出示例**

// 构造httpclient对象

HttpClient client = new HttpClient();

//构造deletemethod对象，设置TGT注销地址，其中红色标注的为单点登录时提交用户名和密码获取的TGT

DeleteMethod method =

new DeleteMethod("http://cas.daxingedu.cn/" + **TGT**);

try {

// 发送请求

client.executeMethod(method);

// 获取请求返回

String response = method.getResponseBodyAsString();

// 处理请求返回

switch (method.getStatusCode()) {

case 200:

*System.out.println(*"Logged out");

break;

default:

*System.out.println(*"Invalid response code (" +

method.getStatusCode() + ") from CAS server!");

*System.out.println(*"Response: " + response);

break;

}

} catch (final IOException e) {

*System.out.println(*e.getMessage());

} finally {

// 释放链接

method.releaseConnection();

}

至此，C／S结构的业务系统与CAS集成完毕。

**四、技术支持**

技术支持：北京星立方科技股份有限公司

联系方式：13716788957，杨工

CAS参考网站：