**CAS集成手册(java版)**

**东软集团股份有限公司**

**数字媒体与教育事业部**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **修改记录** | | | | | |
|  | | | | | |
| 版本号 | 变更控制报告  编号 | 更改条款及内容 | 更改人 | 审批人 | 更改日期 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[一、设计思想 4](#_Toc283276337)

[二、实现逻辑 4](#_Toc283276338)

[三、编码实现 4](#_Toc283276339)

[1、 JDK版本要求 4](#_Toc283276340)

[2、 客户端JAR包 5](#_Toc283276341)

[3、 增加并配置casFilterConfig.xml 5](#_Toc283276342)

[4、 修改登录入口 6](#_Toc283276343)

# 一、设计思想

CAS（Central Authentication Service – 中心认证服务）的目的就是使分布在内部各个不同异构系统的认证工作集中在一起，通过一个公用的认证系统统一验证用户身份的合法性。通过CAS认证的用户将获得CAS颁发的一个证书，使用这个证书，用户可以在承认CAS证书的各个系统上自由穿梭访问，不需要再次的登录认证。

# 二、实现逻辑

CAS客户端在认证过程中的实现逻辑：

1）以filter的形式，对B/S结构应用系统资源进行过滤保护，可以保护所有资源，也可以保护单一资源。

2）获得CAS Server颁发的ServiceTicket，并凭此ST从CAS Server上取得登录用户信息。

3）为第三方应用提供开发接口，使得受保护的应用能够根据CAS认证信息进行正确的登录。

# 三、编码实现

通过CAS与第三方应用系统（以下统一简称为【**集成系统】**）集成时，应用系统开发商需要完成以下四步工作：

## JDK版本要求

程序使用的JDK必须是1.5及以上版本。

## 客户端JAR包

将我公司提供的CAS Client Jar包（neusoft-cas-4.0.0.jar），以及另外两个 Jar 包 cas-client-core-3.2.1.jar ，commons-logging-1.1.jar放在集成系统的WEB-INF/lib目录下。

## 增加并配置casFilterConfig.xml

新增配置文件casFilterConfig.xml，该文件放在集成系统的WEB-INF/classes目录下，内容示例如下：

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <config>  <loginServer>http://cas.\*\*\*.edu.cn/cas</loginServer>  <validateServer>http://cas.\*\*\*.edu.cn/cas</validateServer>  <serverName>localhost:8080</serverName >  <notForceAuthUrls>  <url-pattern>\.js$</url-pattern>  <url-pattern>\.xsl$</url-pattern>  <url-pattern>\.xml$</url-pattern>  <url-pattern>\.gif$</url-pattern>  <url-pattern>\.jpg$</url-pattern>  <url-pattern>\.bmp$</url-pattern>  <url-pattern>\.css$</url-pattern>  <url-pattern>\.swf$</url-pattern>  <url-pattern>\.htm$</url-pattern>  <url-pattern>\.html$</url-pattern>  </notForceAuthUrls>  </config> |

各节点代表含义说明：

<loginServer>指的是CAS Server的登陆URL；

<validateServer>指的是CAS Server的验证URL（需要这个配置来告诉CAS Client获取Service Ticket后发送给这个地址进行验证从而取得用户信息）。对于目前我公司的CAS Server版本来说与loginServer是相同的。

<serverName>指的是当前集成系统所在的服务器和端口号，服务器可以是机器名、域名或ip，建议使用域名。端口不指定的话默认是80**（此处不能附加http://和应用名称）**。

<notForceAuthUrls>下的<url-pattern>指定了不需要CAS Filter进行过滤的资源，当过滤所有资源的时候使用。

## 修改登录入口

统一使用CAS登录入口，在应用系统WEB-INF目录下的web.xml文件中，加入CAS Filter相关的过滤器配置，增加内容如下：

|  |
| --- |
| <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>  <filter>  <filter-name>CASAuthenticationFilter</filter-name>  <filter-class>com.neusoft.cas.AuthenticationFilter</filter-class>  </filter>  <filter-mapping>  <filter-name>CASAuthenticationFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>  <filter>  <filter-name>CASValidationFilter</filter-name>  <filter-class>com.neusoft.cas.Cas20ProxyReceivingTicketValidationFilter</filter-class>  </filter>  <filter-mapping>  <filter-name>CASValidationFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>  <filter>  <filter-name>CASHttpServletRequestWrapperFilter</filter-name>  <filter-class>org.jasig.cas.client.util.HttpServletRequestWrapperFilter</filter-class>  </filter>  <filter-mapping>  <filter-name>CASHttpServletRequestWrapperFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping> |

集成系统登录入口代码修改：

* 引入cas认证类

引入下面的类：

|  |
| --- |
| org.jasig.cas.client.authentication.AttributePrincipal |

* 修改原登录入口的获取用户账号方法，使用下面的方法获取登录用户

|  |
| --- |
| AttributePrincipal principal = (AttributePrincipal)request.getUserPrincipal();  String userID = principal.getName(); |

* 修改登录验证

根据上面的方法获取用户账号后，集成系统需要修改验证方法，不再验证密码信息，只判断该用户是否存在：

如果存在，则实现集成系统的登录（例如往session中写入身份信息、权限信息等）；

如果不存在，按集成系统逻辑处理。

上述配置封闭了原有登录入口，统一使用CAS登录入口。如果希望保留原有登录入口，新增加应用访问入口，新增入口与cas认证集成。在集成系统WEB-INF目录下的web.xml文件中的配置内容如下所示：

|  |
| --- |
| <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>  <filter>  <filter-name>CASAuthenticationFilter</filter-name>  <filter-class>com.neusoft.cas.AuthenticationFilter</filter-class>  </filter>  <filter-mapping>  <filter-name>CASAuthenticationFilter</filter-name>  <url-pattern>/index.jsp(用于登录的JSP文件)</url-pattern>  </filter-mapping>  <filter>  <filter-name>CASValidationFilter</filter-name>  <filter-class>com.neusoft.cas.Cas20ProxyReceivingTicketValidationFilter</filter-class>  </filter>  <filter-mapping>  <filter-name>CASValidationFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>  <filter>  <filter-name>CASHttpServletRequestWrapperFilter</filter-name>  <filter-class>org.jasig.cas.client.util.HttpServletRequestWrapperFilter</filter-class>  </filter>  <filter-mapping>  <filter-name>CASHttpServletRequestWrapperFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping> |

如果希望保留原始的登录页面，并且登录成功之后可以跳转到指定的URL地址，那么需要在用于登录的JSP文件中做一下简单的处理，接收一个 url 地址，然后重定向到这个接收的url地址上即可，如下：

|  |
| --- |
| <%  String url = request.getParameter("url");  if(url!=null && !"".equals(url)){  response.sendRedirect(java.net.URLDecoder.decode(url,"UTF-8"));  }  %> |

注：地址栏往登录页面跳转的时候url后面的参数需要用 encodeURIComponent 函数转换一下，如：

|  |
| --- |
| encodeURIComponent("http://www.neusoft.com?a=111&b=2222");  转后为：http%3A%2F%2Fwww.neusoft.com%3Fa%3D111%26b%3D2222 |