**1、添加依赖jar包**

1）maven工程，修改pom.xml文件，添加依赖：

|  |
| --- |
| <dependency>  <groupId>com.eebbk.webadmin</groupId>  <artifactId>webadmin-authority-client</artifactId>  <version>2.0.0</version>  </dependency> |

2）非maven工程

拷贝如下三个jar包到工程的中：

****

**,**

**2、配置统一登录认证**

1）修改web.xml文件，添加过滤器：

|  |
| --- |
| <!--该监听器用于处理退出登录，必须启用-->  <listener>  <listener-class>  org.jasig.cas.client.session.SingleSignOutHttpSessionListener  </listener-class>  </listener>  <filter>  <filter-name>cas-logout-filter</filter-name>  <filter-class>org.jasig.cas.client.session.SingleSignOutFilter</filter-class>  </filter>  <filter-mapping>  <filter-name>cas-logout-filter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>  <!--该过滤器负责登录校验工作，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>  http://admin.eebbk.com/webadmin-cas/login  </param-value>  </init-param>  <init-param>  <param-name>serverName</param-name>  <!--这里的value配置为客户端的ip和port-->  <param-value>http://[localhostIP]:[tomcatPort]</param-value>  </init-param>  </filter>  <!--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://admin.eebbk.com:10000/webadmin-cas  </param-value>  </init-param>  <init-param>  <param-name>serverName</param-name>  <!--这里的value配置为客户端的ip和port-->  <param-value> http://[localhostIP]:[tomcatPort]</param-value>  </init-param>  <init-param>  <param-name>proxyCallbackUrl</param-name>  <param-value>  <!--cas代理模式配置，黄色字体部分为当前项目的项目名-->  <!--这里的value配置为客户端的ip和port-->  http://[localhostIP]:[tomcatPort]/social\_admin/proxyCallback  </param-value>  </init-param>  <init-param>  <param-name>proxyReceptorUrl</param-name>  <param-value>/proxyCallback</param-value>  </init-param>  </filter>  <filter-mapping>  <filter-name>CASValidationFilter</filter-name>  <url-pattern>/proxyCallback</url-pattern>  </filter-mapping>  <filter-mapping>  <filter-name>CASValidationFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>  <filter-mapping>  <filter-name>CASFilter</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> |

（**上面拦截器配置位置尽量靠前，配置中红色字体部分为服务器地址，不要随改动。**）

2）获取用户ID（名），以及用户其它信息

|  |
| --- |
| AttributePrincipal principal = (AttributePrincipal)request.getUserPrincipal();  principal.getName()； //获取用户ID（登录用户名）  Map<String ,Object> userInfo = principal.getAttributes()); //获取用户其它信息  //用户其它信息主要包括以下字段：  // userId ,realName ,roleCh ,department ,tel ,email |

3）获取模块左边菜单树

|  |
| --- |
| URL：http://admin.eebbk.com/webadmin-authority/authority/authorization/getModuleMenuTree  TYPE：GET （必须用jsonp跨域方式访问）  参数：moduleId （模块ID）  返回结果：jquery插件zTree所需的节点树结构的json字符串  示例：  $.ajax({  type: "GET",  url: "/webadmin-authority/authority/authorization/getModuleMenuTree",  data: {'moduleId':'666'},  dataType: "jsonP",  jsonpCallback: "getMenuTreeJsonp",  success: ......  }); |

4）退出登录

|  |
| --- |
| URL：http://admin.eebbk.com/webadmin-cas/logout  注意：退出登录，会退出所有系统的登录态，慎用。推荐方式：  1、退出单个系统，直接关闭浏览器选项卡选项卡  2、退出整个平台，在导航页点击退出按钮；关闭浏览器。 |

**3、配置统一权限认证（以springMVC框架为例）**

1）创建一个拦截器，集成 BaseAuthorityFilter 类，实现 BaseAuthorityFilter 接口，处理方法示例如下：

|  |
| --- |
| public boolean preHandle(HttpServletRequest request  , HttpServletResponse response  , Object handler) throws AuthorityException  {  // BaseAuthorityFilter类中方法，功能：发送请求到远程服务器校验权限  boolean isAuthority = this.checkAuthority(request);  if(isAuthority)  {  return true;  }  //权限验证失败，抛出异常，让异常处理总类处理，并返回响应结果。  throw new AuthorityException("没有操作权限");  } |

2）配置spring拦截器：

|  |
| --- |
| <bean class="com.eebbk.demo.bussiness.interceptor.AuthorityInterceptor">  <!-- 以下2个属性为必配项，其中模块ID由权限认证中心分配 -->  <property name="serverAddress" value="http://admin.eebbk.com:10000/webadmin-authority/" />  <property name="moduleId" value="8888" />  </bean> |

3）数据级权限控制所需接口列表：

|  |
| --- |
| 描述：获取模块角色列表  type：get  URL：http://admin.eebbk.com/webadmin-authority/authority/authRole/queryRoleInfo  参数: moduleId（模块ID）、page（页码）、rows（每页记录数）  描述：获取模块资源列表  type：get  URL：http://admin.eebbk.com/webadmin-authority/authority/authResources/getAuthResources  参数: moduleId（模块ID） |

**4、前后端分离-首页地址跳转配置**

1）配置IndexRedirectFilter拦截器（需要配置在cas登录拦截器之后）

|  |
| --- |
| <filter>  <filter-name>indexRedirectFilter</filter-name>  <filter-class>  com.eebbk.webadmin.authority.client.IndexRedirectFilter  </filter-class>  <init-param>  <param-name>projectName</param-name>  <!-- 黄色标记为当前项目项目名 -->  <param-value>social\_admin</param-value>  </init-param>  </filter>  <filter-mapping>  <filter-name> indexRedirectFilter </filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping> |

2）业务系统nginx配置，需要添加rewrite配置

|  |
| --- |
| upstream front-html {  server uniplatform-front.eebbk.com;  }  location ~ /html/.+ {  proxy\_pass http://$ front-html;  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  } |

========至此配置完成，访问地址：http:// admin.eebbk.com/ ========

**5、注意事项**

1）页面中不要使用<base href="<%=basePath%>">，会影响https，造成跨域问题。