# 1.添加依赖

添加cas-user-sync项目依赖：

<dependency>

<groupId>com.credithc</groupId>

<artifactId>cas-user-sync</artifactId>

<version>1.0.0-SNAPSHOT</version>

</dependency>

# 2.修改web.xml文件

对红色部分的URL做修改：

casServerLoginUrl---------------cas系统登录URL

casServerUrlPrefix--------------cas系统前缀

serverName---------------------应用系统URL

<!-- ========================单点登录开始 ======================== -->

<!--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>CASSingle Sign OutFilter</filter-name> <filter-class>org.jasig.cas.client.session.SingleSignOutFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>CASSingle Sign OutFilter</filter-name>

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

</filter-mapping>

<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://localhost:8080/cas/login</param-value>

</init-param>

<init-param>

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

<param-value>http://localhost: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>CASValidationFilter</filter-name>

<filter-class>

org.jasig.cas.client.validation.Cas20ProxyReceivingTicketValidationFilter

</filter-class>

<init-param>

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

<param-value>http://localhost:8080/cas</param-value>

</init-param>

<init-param>

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

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

</init-param>

</filter>

<filter-mapping>

<filter-name>CASValidationFilter</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>

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

# 3.客户端获取用户信息方法

1. AssertionHolder

AttributePrincipal userinfo = AssertionHolder.*getAssertion*().getPrincipal();

String isAdmin = (String)userinfo.getAttributes().get("isAdmin");

String username = userinfo.getName();

2. HttpSession

AssertionImpl ast = (AssertionImpl)sesssion.getAttribute("\_const\_cas\_assertion\_");

AttributePrincipal principal = ast.getPrincipal();

3. HttpServletRequest

(1) AttributePrincipal userinfo = (AttributePrincipal)req.getUserPrincipal();

String isAdmin = (String)(userinfo.getAttributes().get("isAdmin"));

(2)String userId = request.getRemoteUser();

# 4.必要表：

DROP TABLE IF EXISTS `cas\_user`;

CREATE TABLE `cas\_user` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`user\_name` varchar(60) NOT NULL,

`enabled` tinyint(1) NOT NULL,

`cname` varchar(60) NOT NULL,

`company\_id` varchar(60) NOT NULL,

`ename` varchar(60) DEFAULT NULL

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8 ROW\_FORMAT=DYNAMIC;

# 5.注意事项

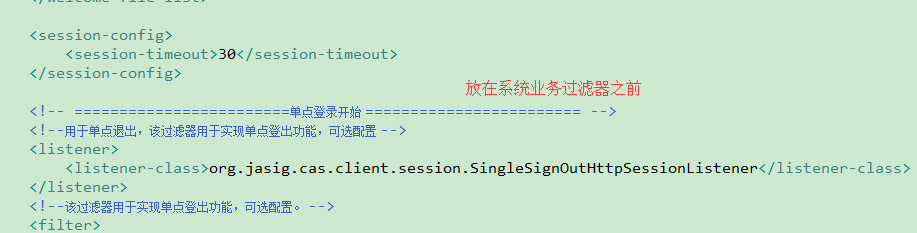
1. web.xml里面的地址配置最好用IP地址，配置域名需要在host文件中指明



2.dataSources数据源配置需要指明编码



3.cas的过滤器配置需要放在客户端系统自己业务过滤器的前面



4.有需要不被登录过滤的需要修改CASFilter

<filter>

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

<filter-class>com.credithc.cas.user.filter.LoginCheckFilter</filter-class>

<init-param>

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

<param-value>http://10.108.10.6:8080/cas/login</param-value>

</init-param>

<init-param>

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

<param-value>http://10.108.5.8:8080</param-value>

</init-param>

<init-param>

<param-name>excludedPages</param-name>

<param-value>/userSync/isLogin.do</param-value>

</init-param>

</filter>

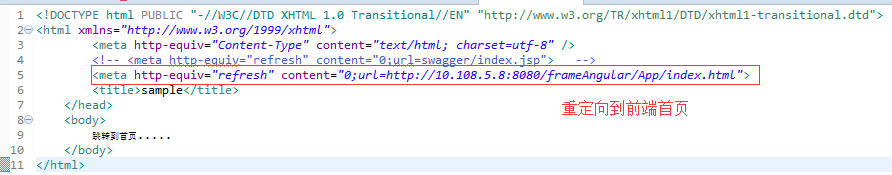
5.前后分离项目部署

前端项目增加判断逻辑：ajax请求后端/userSync/isLogin.do，返回false（即未登录）跳转到cas登录页（service参数为后端的一个中转页），登录成功后跳转到后端的中转页，中转页重定向到前端首页；返回true则正常跳转。

1.前端判断逻辑



2.后端中转页指向前端首页



# 6.客户端过滤器交给spring管理

1.增加spring security依赖包

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-web</artifactId>

<version>3.2.0.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-config</artifactId>

<version>3.2.0.RELEASE</version>

<scope>runtime</scope>

</dependency>

2.修改web.xml，把过滤器委托给spring



3.新建applicationContext-cas.xml文件，把cas过滤器放到该文件





4.把casLoginUrl,casServer,clientServer变量配置到xdiamond中

casLoginUrl：cas登录页面地址

casServer:cas服务器端域名端口

clientServer：应用自身域名端口