**业务系统集成统一认证平台说明-BS**

# 1. 系统要求

业务系统采用maven进行编译打包，业务系统采用shiro框架进行认证并通过URL进行权限控制。

# 2. 集成步骤

## 2.1 配置maven依赖

WEB模块POM文件或者根POM文件增加如下依赖：

<dependency>

<groupId>com.hisign</groupId>

<artifactId>sdk-shiro-filter</artifactId>

<version>1.0-SNAPSHOT</version>

</dependency>

## 2.2 Shiro的spring配置

### 2.2.1信用用户域的配置

业务系统需要增加信任Realm，com.hisign.sdk.web.shiro.realm.TrustUserRealm，用于单点登陆之后跳回业务系统的时候进行信任模式登录业务系统。

### 2.2.2过滤器的配置

业务系统所有需要授权访问的URL地址配置过滤器：com.hisign.sdk.web.shiro.filter.CustPermsAuthorizationFilter。

# 3. 电诈平台实例：

具体到新版电诈平台，在原有的spring-shiro-web.xml文件里修改。

## 3.1 信用用户域的配置

已经有id为trustUserRealm的Spring bean配置，class改成com.hisign.sdk.web.shiro.realm.TrustUserRealm，然后增加属性：<property name="userAuthorizer" ref="userAuthorizer"/>，完整trustUserRealm bean配置如下：

<bean id="trustUserRealm" class="com.hisign.sdk.web.shiro.realm.TrustUserRealm">

<property name="userAuthorizer" ref="userAuthorizer"/>

<property name="cachingEnabled" value="true"/>

<property name="authenticationCachingEnabled" value="true"/>

<property name="authenticationCacheName" value="authenticationCache"/>

<property name="authorizationCachingEnabled" value="true"/>

<property name="authorizationCacheName" value="authorizationCache"/>

</bean>

需要配置trustUserRealm引用到的userAuthorizer bean：

<bean id="userAuthorizer" class="com.hisign.dzpt.web.shiro.realm.UserAuthorizerImpl">

</bean>

同时增加代码类com.hisign.dzpt.web.shiro.realm.UserAuthorizerImpl，UserAuthorizerImpl需要实现com.hisign.sdk.web.shiro.realm.UserAuthorizer接口，然后继承业务系统已有的UserRealm，UserAuthorizer接口的findAuthorizationInfo方法的实现是调用UserRealm的doGetAuthorizationInfo方法，完整代码如下：

/\*\*

\*

\*/

package com.hisign.dzpt.web.shiro.realm;

import org.apache.shiro.authz.AuthorizationInfo;

import org.apache.shiro.subject.PrincipalCollection;

import com.hisign.sdk.web.shiro.realm.UserAuthorizer;

/\*\*

\* @author chailiangzhi

\* @date 2016-6-27

\*

\*/

public class UserAuthorizerImpl extends UserRealm implements UserAuthorizer {

/\* (non-Javadoc)

\* @see com.hisign.sdk.web.shiro.realm.UserAuthorizer#findAuthorizationInfo(java.lang.String)

\*/

@Override

public AuthorizationInfo findAuthorizationInfo(PrincipalCollection principals) {

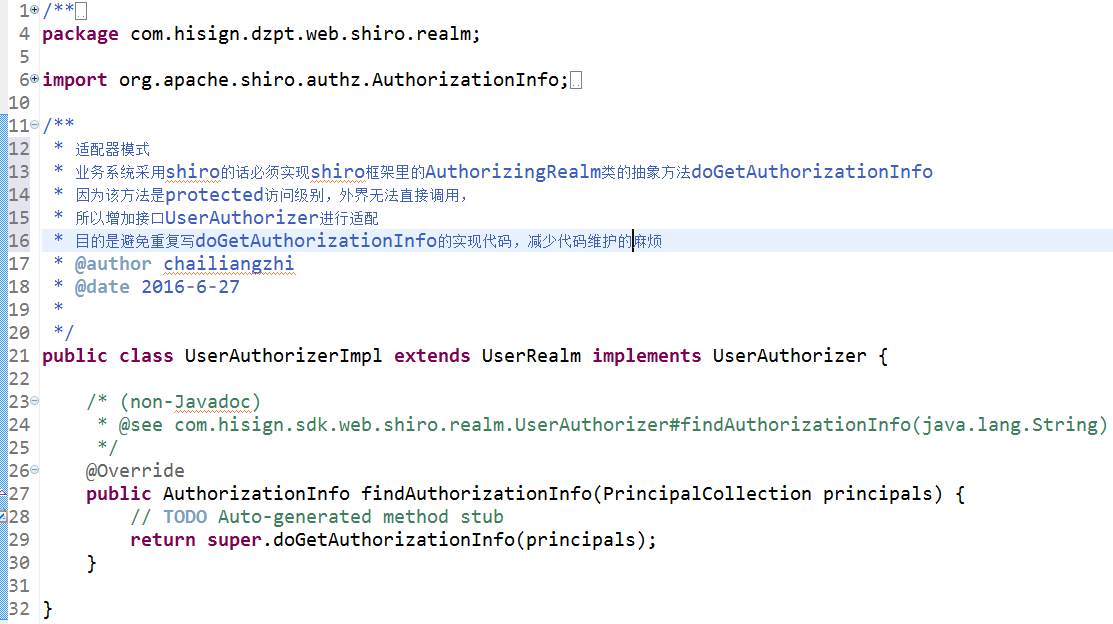
// TODO Auto-generated method stub

return super.doGetAuthorizationInfo(principals);

}

}

代码截图：



## 3.2 过滤器的配置

CustPermsAuthorizationFilter需要引用从数据库读取配置信息的接口，具体配置如下：

<bean id="sysParamService" class="com.hisign.sdk.config.service.impl.SysParamServiceImpl">

</bean>

<bean id="permsAuthorizationFilter" class="com.hisign.sdk.web.shiro.filter.CustPermsAuthorizationFilter">

<property name="sysParamService" ref="sysParamService"/>

</bean>

Shiro过滤器工厂类的filters属性增加CustPermsAuthorizationFilter实例的引用：

<property name="filters">

<util:map>

<entry key="authc" value-ref="passThruAuthenticationFilter"/><!-- formAuthenticationFilter -->

<entry key="user" value-ref="userFilter"/>

<entry key="perms" value-ref="permsAuthorizationFilter" />

</util:map>

</property>

filterChainDefinitions属性最后一行增加 /\*\* = perms。

完整如下图：

