系统的权限的最终解决方案

JOffice中的权限管理是基于角色的管理策略，采用Spring Security2的配置方式，同时能够结合EXT3来进行整个系统的权限管理，通过使用配置文件，进行整个系统的功能集中管理，包括系统左边的导航菜单，数据列表中的功能操作权限。我们知道，在传统的Ｗeb项目中，我们都是采用基于URL进行权限控制的。基于EXT的应用也不例外，只不过我们是需要结合前台的功能菜单一起，前台显示出来的功能菜单，其后台均代表可以访问。

使用方法：

一、先在menu.xml中配置系统的功能，如我们配置角色管理的权限，如下所示：

<?xml version="1.0" encoding="UTF-8"?>

<Menus>

<Items id="SystemSetting" text="系统设置" iconCls="menu-system">

...

<Item id="AppRoleView" iconCls="menu-role" text="角色设置">

<Function id="\_AppRoleList" text="查看角色" iconCls="menu-list">

<url>/system/listAppRole.do</url>

</Function>

<Function id="\_AppRoleAdd" text="添加角色" iconCls="menu-add">

<url>/system/listAppRole.do</url>

<url>/system/saveAppRole.do</url>

</Function>

<Function id="\_AppRoleEdit" text="编辑角色" iconCls="menu-add">

<url>/system/listAppRole.do</url>

<url>/system/saveAppRole.do</url>

</Function>

<Function id="\_AppRoleDel" text="删除角色" iconCls="menu-del">

<url>/system/listAppRole.do</url>

<url>/system/mulDelAppRole.do</url>

</Function>

<Function id="\_AppRoleGrant" text="授权角色">

<url>/system/listAppRole.do</url>

<url>/system/grantAppRole.do</url>

</Function>

</Item>

...

<Item id="ReportTemplateView" iconCls="menu-report" text="报表管理">

...

</Item>

</Items>

<!--其他模块 -->

<Items>

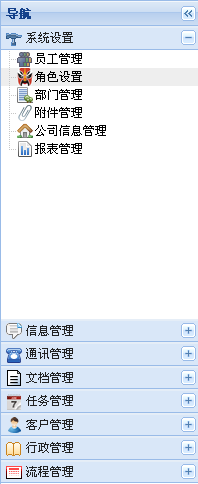
...

</Items>

</Menus>

说明：

1．Menu.xml为整个系统的功能列表，系统的左边菜单也是从该文件显示出来，如：



2．进入某项菜单中，会进入相应的模块的管理界面，如：



上面的功能“添加角色”，“删除角色”以及管理一栏中的“删除”，“编辑”，“授权”等功能，则对应为menu.xml中的Item下的Function标签的配置，如：

添加角色对应：

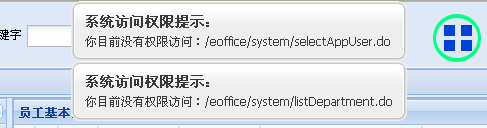
<Function id="\_AppRoleAdd" text="添加角色" iconCls="menu-add">

<url>/system/listAppRole.do</url>

<url>/system/saveAppRole.do</url>

</Function>

其下包括两个Url，一个是/system/listAppRole.do，另一Url /system/saveAppRole.do，表示当我们访问这个功能完成角色添加时，需要访问这两个URL，后台会根据当前用户的角色是否授权访问这个功能而决定是否开放这两个URL给当前用户访问，否则，前台浏览器会弹出一个类似如下的提示，告诉用户当前没有权限访问这个URL。



当然，这种还是要避免出现，因为系统在客户端出现的功能菜单，均是有权访问的。

３.前台的代码配置

前台的用户登录系统后，会有一个全局的变量存储用户的所有权限（其会把所有的角色的配置抽取出来，去掉重复的权限配置），可以在客户端直接检查用户是否有权限访问某一个菜单的功能，如我们检查用户是否直接有访问“添加角色”的功能，则可以调用系统提供的方法：

isGranted(‘\_AppRoleAdd’);

该函数返回为true则代表可以访问。

因此我们若要进行更高的权限粒度控制，我们需要把原来的代码进行改装一下。如：

AppRoleView.js需要进行更改。

修改一

**var** toolbar = **new** Ext.Toolbar({

id : 'AppRoleFootBar',

height : 30,

bodyStyle:'text-align:left',

items : []

});

**if**(isGranted('\_AppRoleAdd')){

toolbar.add(**new** Ext.Button({

iconCls : 'btn-add',

text : '添加角色',

handler : **function**() {

**new** AppRoleForm();

}

}));

}

**if** (isGranted('\_AppRoleDel')) {

toolbar.add(**new** Ext.Button({

iconCls : 'btn-del',

text : '删除角色',

handler : **function**() {

**var** grid = Ext.getCmp("AppRoleGrid");

**var** selectRecords = grid.getSelectionModel().getSelections();

**if** (selectRecords.length == 0) {

Ext.Msg.alert("信息", "请选择要删除的记录！");

**return**;

}

**var** ids = Array();

**for** (**var** i = 0; i < selectRecords.length; i++) {

ids.push(selectRecords[i].data.roleId);

}

AppRoleView.remove(ids);

}

}));

}

修改二

管理列中的栏目功能：

**if**(isGranted('\_AppRoleDel'))

str = '<button title="删除" value=" " class="btn-del" onclick="AppRoleView.remove('+ editId + ')"></button>';

**if**(isGranted('\_AppRoleEdit'))

str += '&nbsp;<button title="编辑" value=" " class="btn-edit" onclick="AppRoleView.edit('+ editId + ')"></button>';

**if**(isGranted('\_AppRoleGrant'))

str += '&nbsp;<button title="授权" value=" " class="btn-grant" onclick="AppRoleView.grant('+ editId + ',\'' + roleName + '\')">&nbsp;</button>';

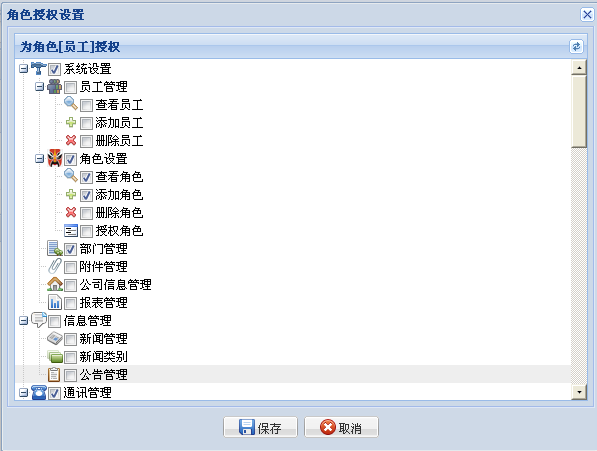
二、同步menu.xml的数据至数据库

打开配置common/config/下的config.properties的这个配置，如下：

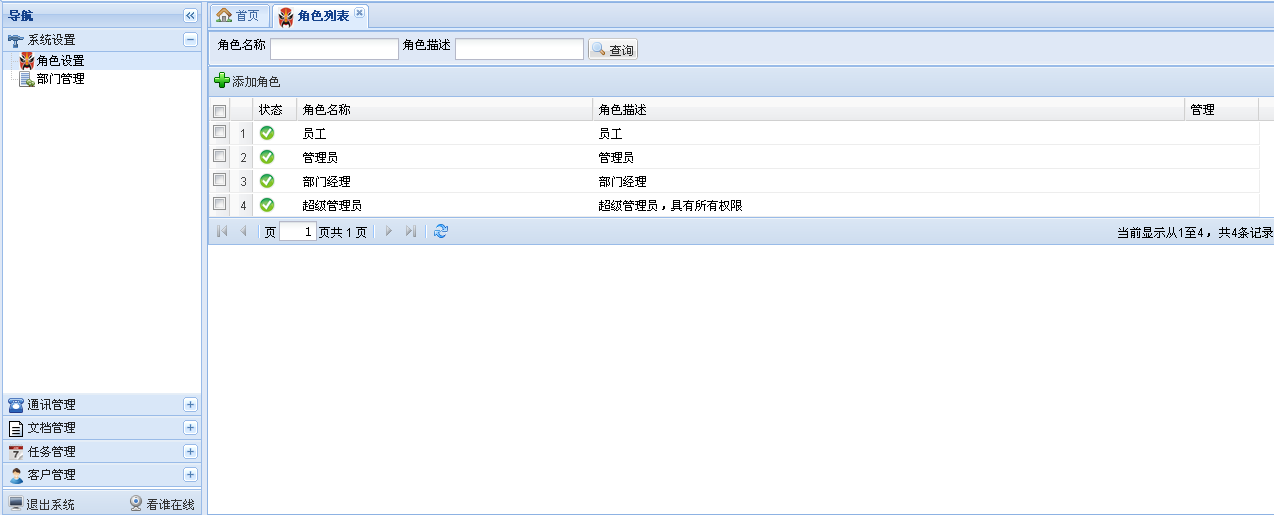
isSynMenu=true

打开此配置则表示会把menu.xml中的所有的Function中的url同步到数据库，转给Spring Security进行匹配，则此时进行角色的权限授予才会真正生效。

三、为角色进行相应的角色授权



在此，将看到该用户访问该模块时，只有“添加角色”这个功能。同样，以上的权限管理也控制着整个系统的左边菜单的权限，如下图所示:



四、拥有超级管理角色，将具有系统的所有访问权限。