市级医药采购系统项目实战



# 模块:权限管理

## 子功能:右键菜单对权限表的增删改查

### 功能描述

只有系统管理员可以实现对权限表的增删改查,当用户点击权限管理的时候,可以看到系统角色列表页面,通过这个页面,可以对”系统管理员”,””卫生室”,”卫生院”,”卫生局”,”供应商”等系统角色赋予不同角色,赋予角色是通过勾选树形菜单来操作的,点击更新权限,可以实现对不同用户可以操作的功能,进行控制,增加了系统的安全性,灵活性,因为系统左侧菜单是通过查询权限表来动态加载的,所以说,就可以实现对每个用户的功能的限制.

### 代码实现

#### Jsp页面（rolelist.jsp）

1.触发函数链接截图如下：

|  |
| --- |
| **function authorize\_role(id) {**  **//alert(id);**  **var sendUrl = "${baseurl}/perm/roleauth.action?sysRoleQueryCustom.id="+id;**  **createmodalwindow("角色权限授权", 800, 450, sendUrl);**  **};** |

2.实现代码

|  |
| --- |
| **function initGrid() {**  **$('#sysrolelist').datagrid({**  **title : '系统角色列表',**  **//nowrap : false,**  **striped : true,**  **//collapsible : true,**  **url : '${baseurl}/perm/rolelist\_result.action',**  **//sortName : 'code',**  **//sortOrder : 'desc',**  **//remoteSort : false,**  **idField : 'id',**  **//frozenColumns : frozenColumns,**  **columns : columns,**  **pagination : true,**  **rownumbers : true,**  **//toolbar : toolbar,**  **//loadMsg : "",**  **//pageList : [ 15, 30, 50, 100 ],**  **onClickRow : function(index, field, value) {**  **$('#sysrolelist').datagrid('unselectRow', index);**  **}**  **});**  **}** |

1. **PermAction**

实现代码：

|  |
| --- |
| /**/异步的获取sys\_permission表**  **public void permission\_result(){**  **PermVo permVo = this.getModel();**  **//1 获得所有权限**  **List<SysPermission> allPerm = sysPermissionService.findAll();**  **//2 获得指定角色的所有权限,也就是说通过判断是供应商还是卫生室,卫生局等等那个角色**  **List<SysPermission> roleAllPerm = sysPermissionService.findAllByRoleId(permVo.getSysRoleQueryCustom().getId());**  **//3 生成 List<TreeVo>**  **// \* 提供缓存 key=permId，value=TreeVo，方便孩子找到爹**  **Map<String, TreeVo> cacheMap = new HashMap<>();**  **// \* 拼凑tree需要的数据**  **List<TreeVo> treeList = new ArrayList<>();**  **for (SysPermission sysPerm : allPerm) {**  **TreeVo treeVo = new TreeVo();**  **treeVo.setId(sysPerm.getId());**  **treeVo.setText(sysPerm.getName());**  **if(roleAllPerm.contains(sysPerm)){**  **treeVo.setChecked("checked");**  **}**  **TreeVo parentTreeVo = cacheMap.get(sysPerm.getParentid());**  **if(parentTreeVo == null){**  **treeList.add(treeVo);**  **} else {**  **parentTreeVo.getChildren().add(treeVo);**  **//如果有子元素，将父目录选择状态取消，交予tree自己控制**  **parentTreeVo.setChecked(null);**  **}**  **cacheMap.put(sysPerm.getId(), treeVo);**  **}**  **// 响应json**  **this.write\_object(treeList);**  **}** |

2配置文件struts-plugin.xml

|  |
| --- |
| **<result name="list">/business/perm/list.jsp</result>** |

3.在rolelist.jsp页面绘制datagrid

|  |
| --- |
| <script type="text/javascript">  function authorize\_role(id) {  //alert(id);  var sendUrl = "${baseurl}/perm/roleauth.action?sysRoleQueryCustom.id="+id;  createmodalwindow("角色权限授权", 800, 450, sendUrl);  };  //工具栏  var toolbar = [ /\* {  id : 'authorize\_role',  text : '角色权限授权',  iconCls : 'icon-add',  handler : authorize\_role  } \*/];  var frozenColumns;  var columns = [ [ /\* {  field : 'id',  title : '',  checkbox : true  }, \*/ {  field : 'name',  title : '角色名称',  width : 180  }, {  field : 'opt1',  title : '角色授权',  width : 130,  formatter:function(value,row,index){  //return "<a href='${baseurl}/roleauth/roleauth.action?roleId="+row.id+"' target='\_blank'>角色授权</a>";  return "<a href=javascript:authorize\_role('"+row.id+"')>角色授权</a>";  }  } ] ];  function initGrid() {  $('#sysrolelist').datagrid({  title : '系统角色列表',  //nowrap : false,  striped : true,  //collapsible : true,  url : '${baseurl}/perm/rolelist\_result.action',  //sortName : 'code',  //sortOrder : 'desc',  //remoteSort : false,  idField : 'id',  //frozenColumns : frozenColumns,  columns : columns,  pagination : true,  rownumbers : true,  //toolbar : toolbar,  //loadMsg : "",  //pageList : [ 15, 30, 50, 100 ],  onClickRow : function(index, field, value) {  $('#sysrolelist').datagrid('unselectRow', index);  }  });  }  function sysrolelistquery() {  var formdata = $("#sysrolequeryForm").serializeJson();  //alert(formdata);  $('#sysrolelist').datagrid('unselectAll');  $('#sysrolelist').datagrid('load', formdata);  }  $(function() {  initGrid();  });  </script> |

表格如下：

|  |
| --- |
| <BODY>  <div id="sysrolequery\_div">  <TABLE border=0 cellSpacing=0 cellPadding=0 width="99%" align=center>  <TBODY>  <TR>  <TD>  <table id="sysrolelist"></table>  </TD>  </TR>  </TBODY>  </TABLE>  </div>  </BODY> |

#### Datagrid请求路径代码：

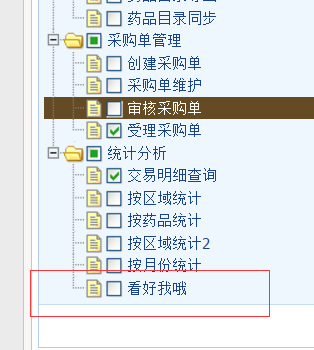
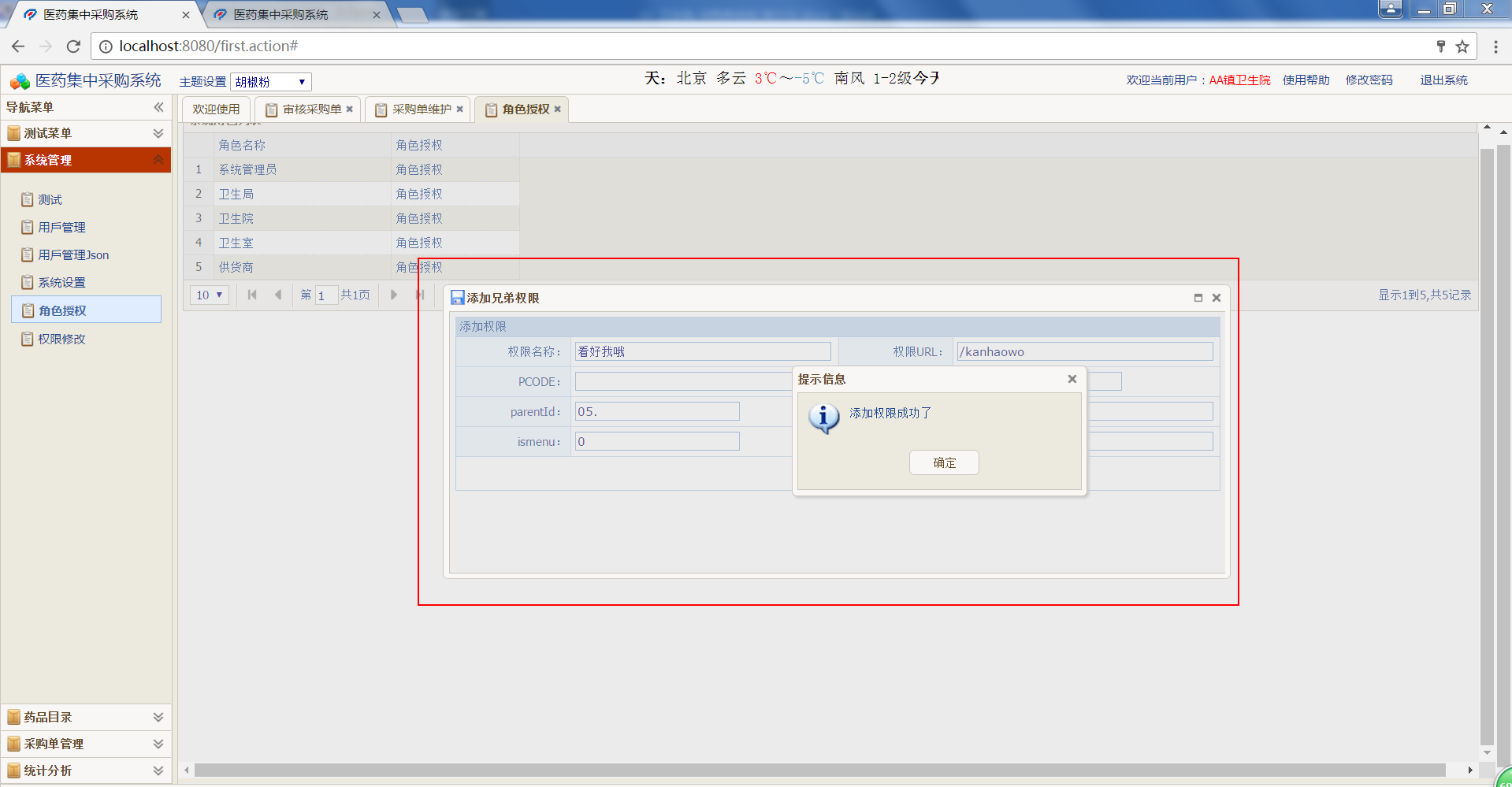
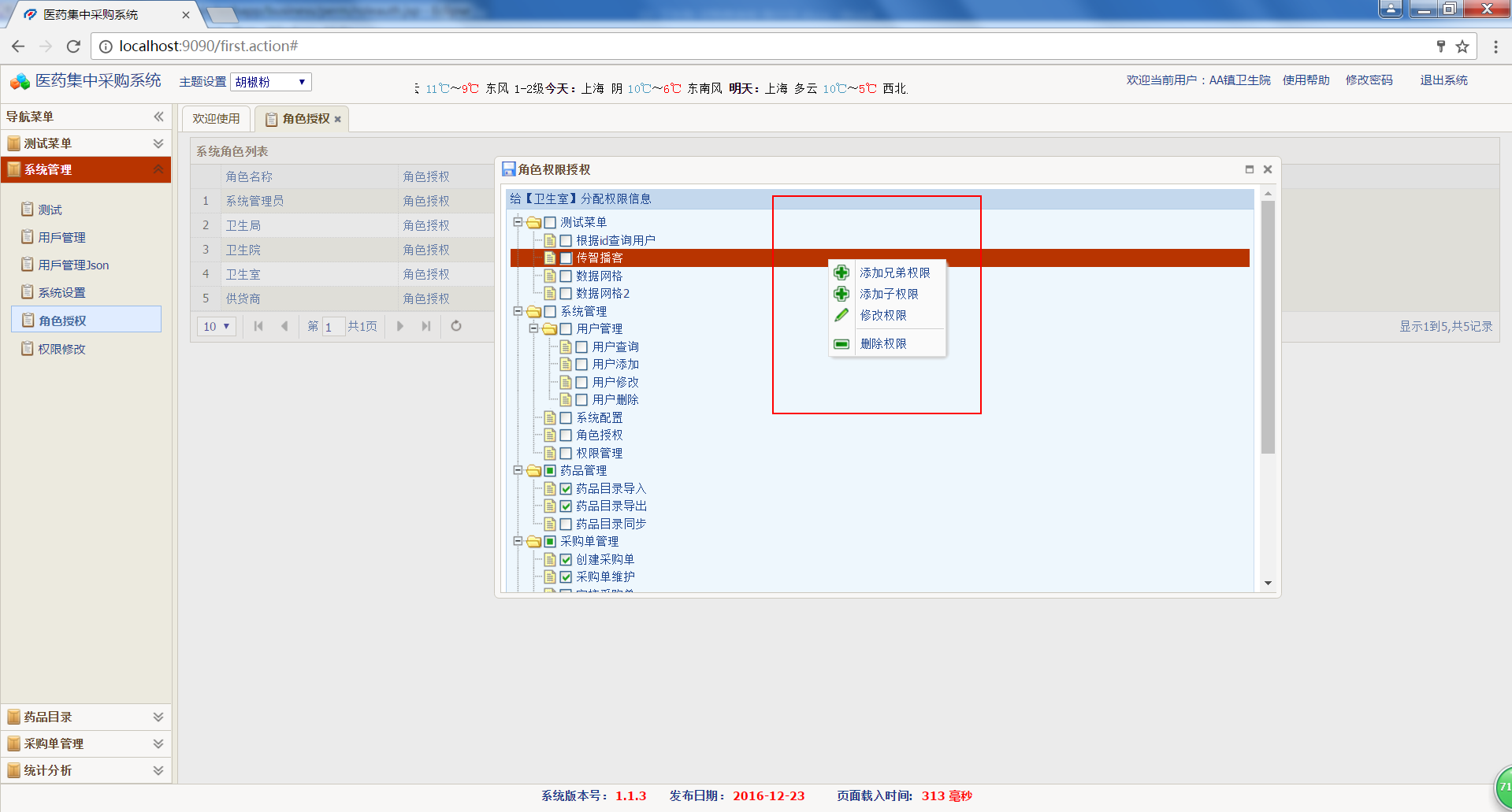
|  |
| --- |
| **//显示树形结构**  **public String roleauth(){**    **PermVo permVo = this.getModel();**    **SysRole sysRole = sysRoleService.findById(permVo.getSysRoleQueryCustom().getId());**    **permVo.setSysRole(sysRole);**    **return "roleauth";**  **}**  **//异步的获取sys\_permission表**  **public void permission\_result(){**    **PermVo permVo = this.getModel();**    **//1 获得所有权限**  **List<SysPermission> allPerm = sysPermissionService.findAll();**      **//2 获得指定角色的所有权限,也就是说通过判断是供应商还是卫生室,卫生局等等那个角色**  **List<SysPermission> roleAllPerm = sysPermissionService.findAllByRoleId(permVo.getSysRoleQueryCustom().getId());**    **//3 生成 List<TreeVo>**  **// \* 提供缓存 key=permId，value=TreeVo，方便孩子找到爹**  **Map<String, TreeVo> cacheMap = new HashMap<>();**      **// \* 拼凑tree需要的数据**  **List<TreeVo> treeList = new ArrayList<>();**  **for (SysPermission sysPerm : allPerm) {**  **TreeVo treeVo = new TreeVo();**  **treeVo.setId(sysPerm.getId());**  **treeVo.setText(sysPerm.getName());**  **if(roleAllPerm.contains(sysPerm)){**  **treeVo.setChecked("checked");**  **}**  **TreeVo parentTreeVo = cacheMap.get(sysPerm.getParentid());**  **if(parentTreeVo == null){**  **treeList.add(treeVo);**  **} else {**  **parentTreeVo.getChildren().add(treeVo);**  **//如果有子元素，将父目录选择状态取消，交予tree自己控制**  **parentTreeVo.setChecked(null);**  **}**  **cacheMap.put(sysPerm.getId(), treeVo);**  **}**      **// 响应json**  **this.write\_object(treeList);**      **}**    **public void roleauthsubmit(){**    **PermVo permVo = this.getModel();**      **sysPermissionService.updatePermission(permVo.getSysRoleCustom().getId(),permVo.getAddPermIds());**      **this.write\_object(ResultUtil.createSubmitResult("权限已更新"));**      **}** |

#### Service层实现类代码：

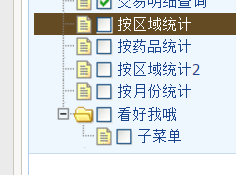
|  |
| --- |
| **@Override**  **public void savePermission(SysPermissionCustom sysPermissionCustom) {**  **SysPermission sysPermission=new SysPermission();**  **BeanUtils.copyProperties(sysPermissionCustom, sysPermission);**  **sysPermissionDao.save(sysPermission);**    **}**  **@Override**  **public void update(SysPermissionCustom sysPermissionCustom) {**  **String id=sysPermissionCustom.getId();**  **SysPermission sysPermission=sysPermissionDao.findById(id);**  **BeanUtils.copyProperties(sysPermissionCustom, sysPermission);**  **sysPermissionDao.update(sysPermission);**  **}**  **@Override**  **public void delete(SysPermission sysPermission) {**  **sysPermissionDao.delete(sysPermission);**  **}**  **@Override**  **public int findCount(SysPermQueryCustom sysPermQueryCustom) {**  **return sysPermissionDao.findCount(sysPermQueryCustom);**  **}**  **@Override**  **public List<SysPermission> findAllList(SysPermQueryCustom sysPermQueryCustom, int firstResult, int maxsResult) {**  **return sysPermissionDao.findAllList(sysPermQueryCustom,firstResult,maxsResult);**  **}**  **@Override**  **public void updateAll(SysPermission sysPermission) {**  **sysPermissionDao.update(sysPermission);**  **}**  **@Override**  **//删除用户权限**  **public void deleteById(String id) {**  **//先查询,看看是否为空**  **SysPermission sysPermission = sysPermissionDao.findById(id);**  **if(sysPermission==null){**  **ResultUtil.throwExcepionResult("删除失败!");**  **}**  **sysPermissionDao.delete(sysPermission);**  **}** |

### 效果展示

鼠标单击右键菜单,点击添加兄弟权限

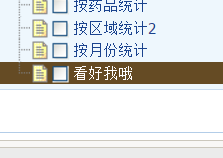
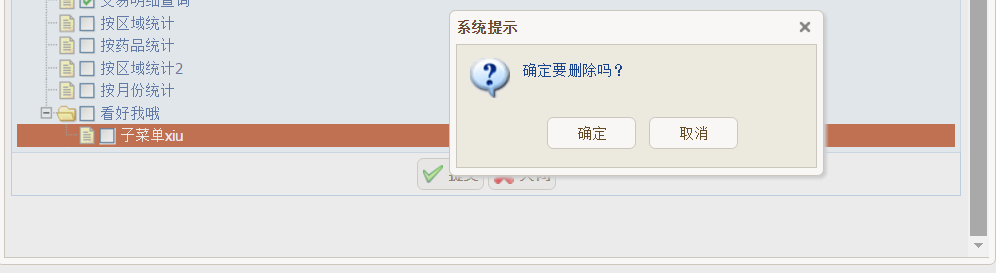


鼠标右键,点击添加子菜单,提示成功,并弹出对话框提示;

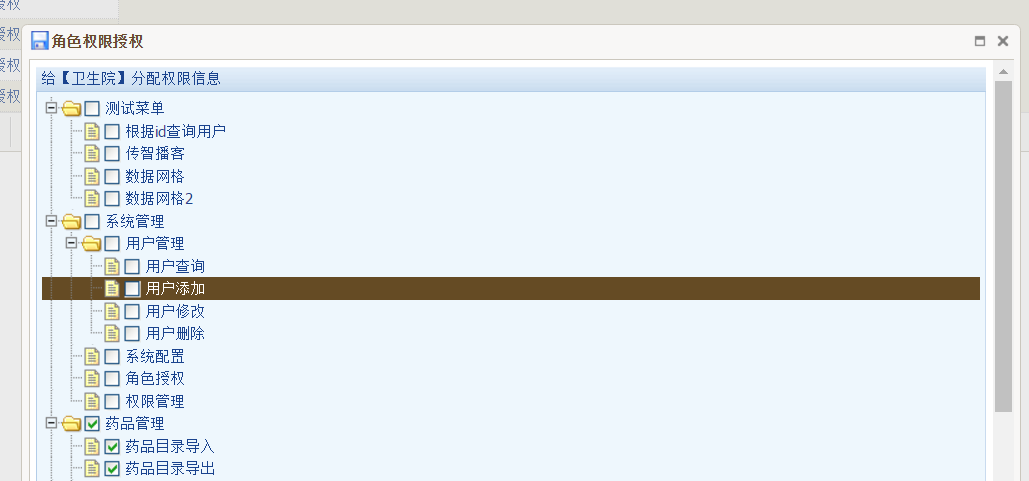
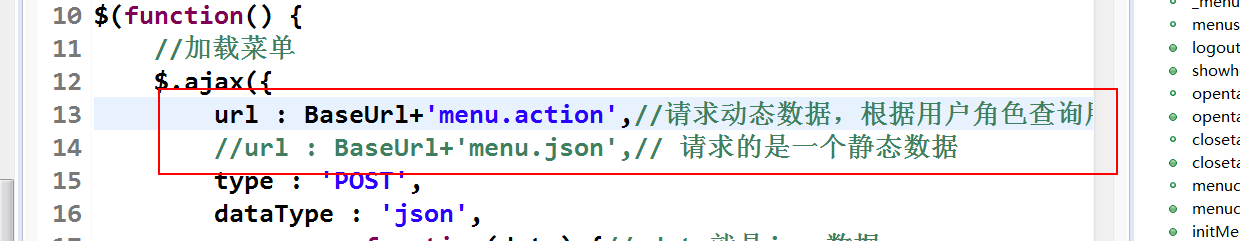


鼠标右键,点击修改权限,输入需要修改的内容后,点击更新,提示更新成功.

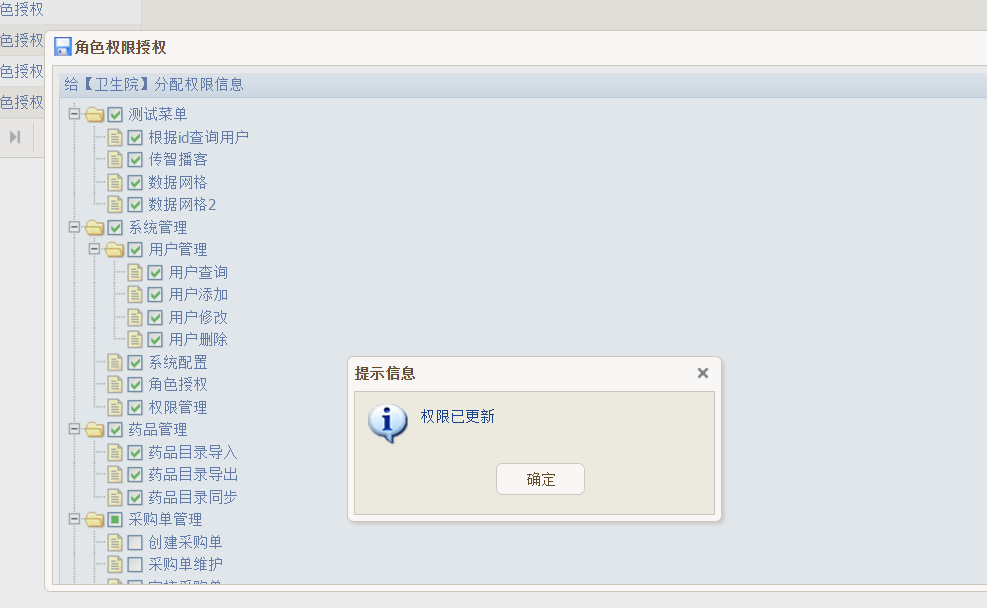
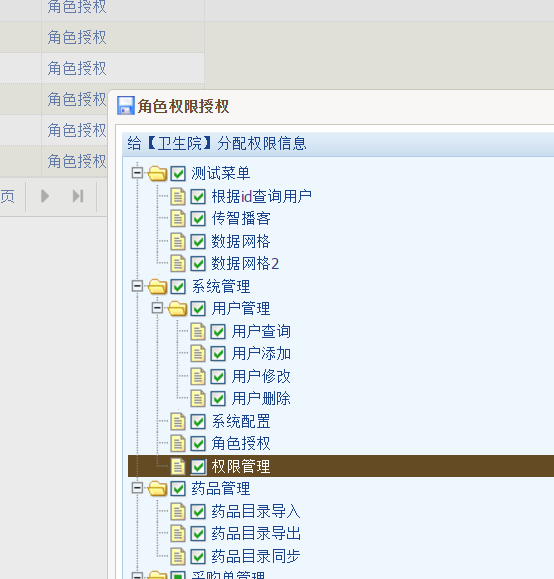
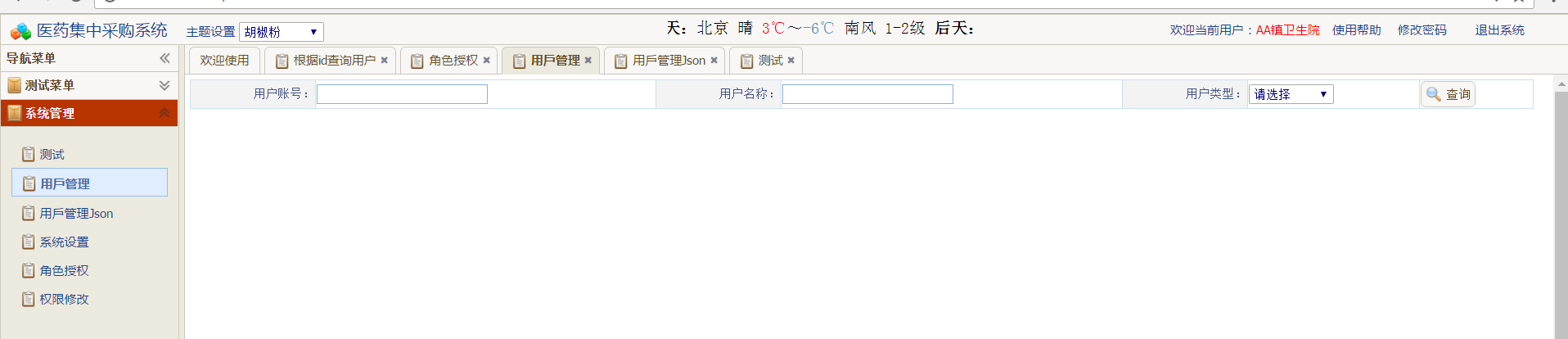
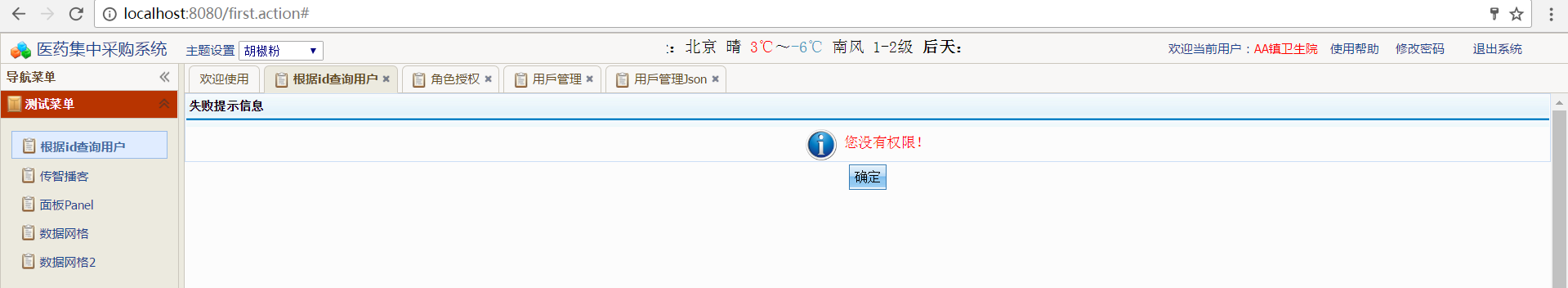




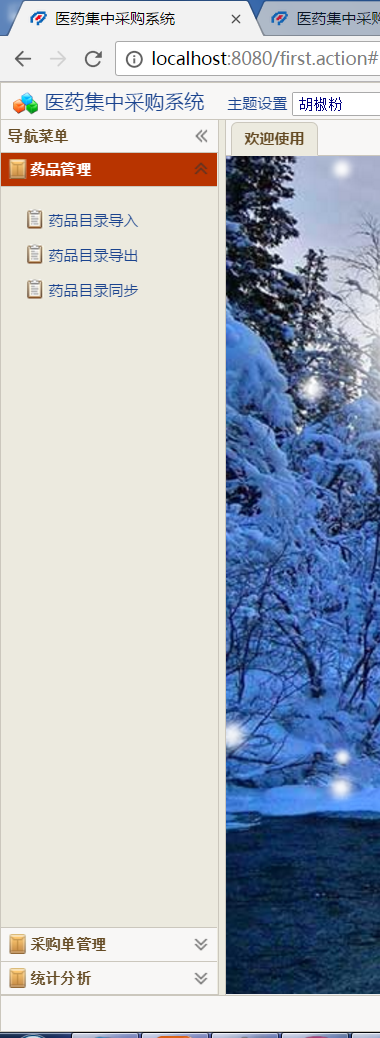
在first.js中修改系统菜单的加载方式,这里使用的是动态菜单



刷新页面,效果如下,没有勾选的功能就不可以使用了;







其他用户的使用功能和控制方法类似;