## 子系统管理

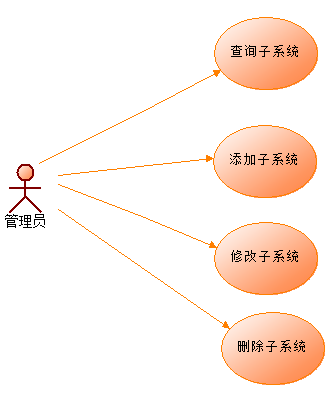
### 概述

子系统管理功能主要是用来管理所有的子系统，BPM系统作为基础平台需要管理所有子系统的用户、角色、资源信息,其中BPM系统也是属于其中一个子系统。

子系统和角色之间的关系为一对多关系，SYS\_ROLE系统角色表中通过字段系统ID与子系统表关联。

子系统和资源之间的关系为一对多关系，SYS\_RES资源表中通过字段系统ID与子系统表关联。

### 用例图



### 用例说明

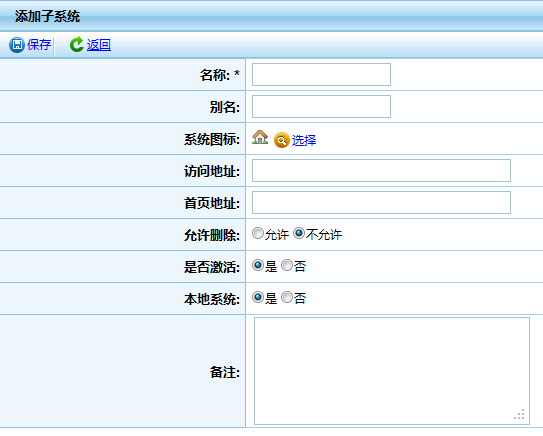
增删改查的功能不过多描述，子系统表中每个字段的说明如下：

* 名称：子系统的名称；
* 别名：子系统的别名，系统在通过cas-server进行单点登录的时候就是以系统别名为查询条件获取该子系统的URL资源，权限资源和角色等信息；
* 系统图标：左上角的系统logo；
* 首页地址：子系统的首页地址；
* 创建时间：子系统的创建时间；
* 允许删除：该子系统是否可以删除；
* 是否激活：该子系统是否已经激活；
* 本地系统：该子系统是否为本地系统，当该子系统和bpm同属于一个应用(也就是同一个ContextPath)，那么是本地子系统，否则不是。

### 用户操作界面



查询子系统界面



添加子系统界面

### 数据库表



### 涉及文件

|  |  |  |
| --- | --- | --- |
| 文件 | 类别 | 备注 |
| SubSystem.java | Model | 与表SYS\_SUBSYSTEM映射实体 |
| SubSystemDao.java | Dao | 实现对子系统的增删改查 |
| SubSystemService.java | Service | 处理业务逻辑 |
| SubSystemController.java | Controller | 控制业务与页面之间的调用 |
| SubSystemFormController.java | Controller | 主要处理添加数据的操作 |
| SubSystemList.jsp | View | 子系统管理列表展现页面 |
| SubSystemEdit.jsp | View | 添加/编辑子系统管理页面 |

## 资源管理

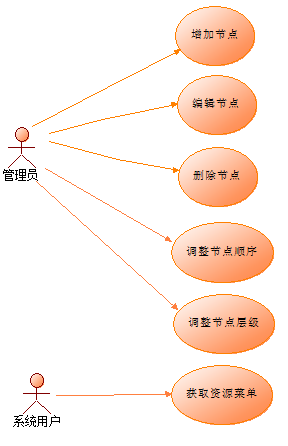
### 概述

资源管理功能主要是管理每个子系统的菜单、页面资源。这些资源由一个个节点组成，节点之间有层级关系。子系统下面的根节点构成该系统的菜单，根节点下面的节点构成每个菜单下面的左侧导航菜单。

资源和子系统之间为多对一关系，资源表中通过字段子系统ID与子系统关联。

资源和角色之间为多对多关系，通过SYS\_ROLE\_RES角色资源映射表进行关联。

### 用例图

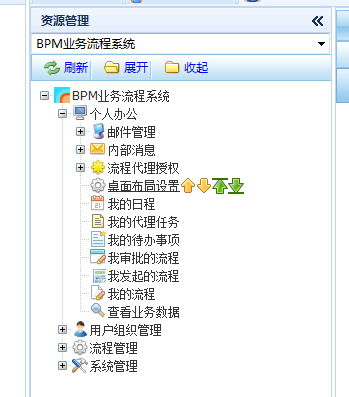


### 用例说明

增加、删除、编辑节点不做过多说明。

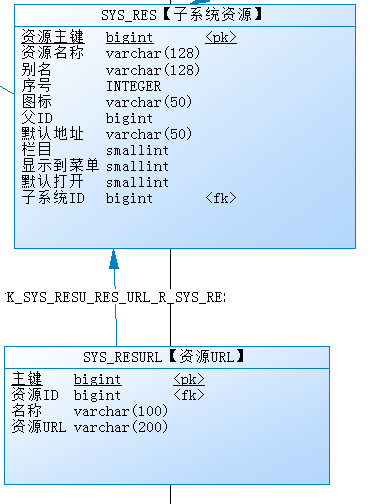
* 调整节点层级：如果需要调整某个节点层级关系，即想把某节点放在另一个节点内，或者从某节点内提取出来。调整步骤是，鼠标左键按住节点拖拽到想要放置的位置。
* 获取资源菜单：所有的系统用户登录的时候都需要获取资源列表，用来加载菜单。

### 用户操作界面



调整节点顺序

### 数据库表



### 外部接口

资源管理有四个接口，接口由com.hotent.platform.webservice.api

.SystemResourcesService提供，使用CXF框架发布为webservice。四个接口说明如下：

* 根据子系统别名获取子系统URL资源。
* 根据子系统别名获取子系统功能别名资源。
* 检查子系统URL资源，有更新则返回最新资源
* 检查子系统功能ID资源，有更新则返回最新资源

/\*\*

\* 取得子系统URL资源。

\* **@param** alias 子系统别名

\* **@return** List<ResourcesUrlExt>子系统URL对应角色的映的谢对像集合。

\*/

@WebMethod(operationName="SystemResourcesService\_loadSecurityUrl")

**public** List<ResourcesUrlExt> loadSecurityUrl(String alias);

/\*\*

\* 取得子系统功能资源。

\* **@param** alias 子系统别名

\* **@return** List<ResourcesUrlExt> 子系统功能ID对应角色的映的谢对像集合。

\*/

@WebMethod(operationName="SystemResourcesService\_loadSecurityFunction")

**public** List<ResourcesUrlExt> loadSecurityFunction(String alias);

/\*\*

\* 检查并决定是否返回子系统URL资源。如果系统资源有更新则返回最新资源，否即返回null。

\* **@param** alias 子系统别名

\* **@return** List<ResourcesUrlExt> 子系统URL对应角色的映的谢对像集合。

\*/

@WebMethod(operationName="SystemResourcesService\_checkSecurityUrl")

**public** List<ResourcesUrlExt> checkSecurityUrl(String alias);

/\*\*

\* 检查并决定是否返回子系统功能ID资源。如果系统资源有更新则返回最新资源，否即返回null。

\* **@param** alias 子系统别名

\* **@return** List<ResourcesUrlExt> 子系统功能ID对应角色的映的谢对像集合。

\*/

@WebMethod(operationName="SystemResourcesService\_checkSecurityFunction")

**public** List<ResourcesUrlExt> checkSecurityFunction(String alias);

### 涉及文件

其中SystemResourcesService.java 在/com/hotent/platform/webservice/api下，

SystemResourcesServiceImpl.java在/com/hotent/platform/webservice/impl下

|  |  |  |
| --- | --- | --- |
| 文件 | 类别 | 备注 |
| Resources.java | Model | 与表SYS\_RES映射实体 |
| ResourcesUrl.java | Model | 与表SYS\_RESURL映射实体 |
| ResourcesDao.java | Dao | 实现对数据表的增删改查 |
| ResourcesUrlDao.java | Dao | 实现对数据表的增删改查 |
| ResourcesService.java | Service | 处理业务逻辑 |
| ResourcesUrlService.java | Service | 处理业务逻辑 |
| SubSystemService.java | Service | 处理业务逻辑 |
| SystemResourcesService.java | 接口 | CXF框架接口 |
| SystemResourcesServiceImpl.java |  | 实现CXF框架接口 |
| ResourcesController.java | Controller | 控制业务与页面之间的调用 |
| ResourcesFormController.java | Controller | 主要处理添加数据的操作 |
| ResourcesTree.jsp | View | 资源树展现页面 |
| ResourcesEdit.jsp | View | 添加/编辑资源页面 |
| ResourcesGet.jsp | View | 子系统资源明细展现页面 |
| ResourcesIcons.jsp | View | 设置图标页面 |

## 角色管理

### 概述

角色管理功能主要为管理各子系统的角色。

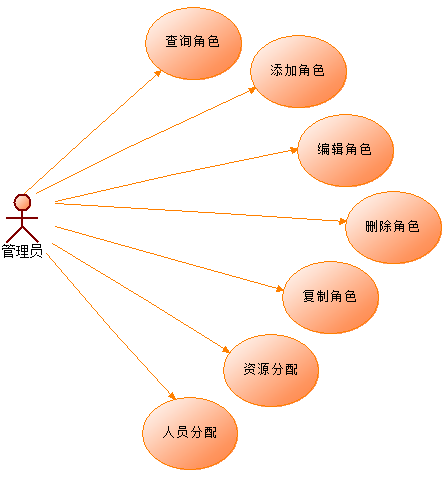
角色和用户之间的关系为多对多关系，使用SYS\_USER\_ROLE用户角色映射表进行关联。

角色和子系统之间的关系为一对多关系（其实还有一种特殊情况，即全局角色，这种角色可以看做属于所有子系统），在角色表中通过字段系统ID与子系统表进行关联。

角色和资源之间的关系为多对多关系，使用SYS\_ROLE\_RES角色资源映射表进行关联。

角色和岗位之间的关系为多对多关系，使用SYS\_ROLE\_POS岗位角色映射表进行关联。

### 用例图

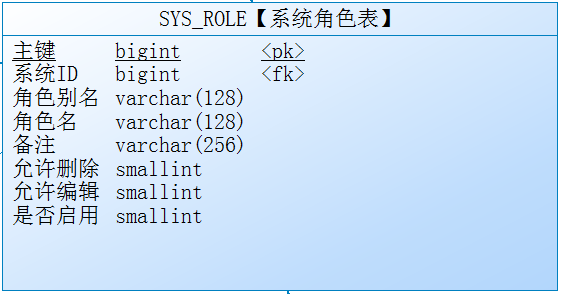


### 用例说明

增删改查的功能不做过多描述。

* 资源分配，为该角色分配资源，分配好资源以后该角色就拥有访问该资源的权限。
* 人员分配，为该角色分配人员，分配到该角色下的用户即拥有该角色，即能访问该角色所拥有的资源。
* 复制角色，如果该角色已经分配好资源和人员，复制角色的功能可以复制这些资源和人员分配到新的角色上。

### 数据库表



### 外部接口

角色管理有一个接口，接口由com.hotent.platform.webservice.api. UserDetailsService提供。使用CXF框架发布为webservice。该接口说明如下：

/\*\*

\* 通过用户名获取用户的角色信息列表

\* **@param** userName 用户名

\* **@return**

\*/

@WebMethod(operationName="UserDetailsService\_loadRoleByUsername")

**public** Collection<String> loadRoleByUsername(String userName);

### 涉及文件

|  |  |  |
| --- | --- | --- |
| 文件 | 类别 | 备注 |
| SysRole.java | Model | 与表SYS\_ROLE映射实体 |
| SysRoleDao.java | Dao | 实现对数据表的增删改查 |
| SysRoleService.java | Service | 处理业务逻辑 |
| SubSystemService.java | Service | 处理业务逻辑 |
| SysRoleController.java | Controller | 控制业务与页面之间的调用 |
| SysRoleFormController.java | Controller | 主要处理添加数据的操作 |
| SysRoleList.jsp | View | 系统角色列表展现页面 |
| SysRoleEdit.jsp | View | 添加/编辑资源页面 |
| SysRoleGet.jsp | View | 系统角色明细展现页面 |
| roleResourcesEdit.jsp | View | 资源分配页面 |
| userRoleEdit.jsp | View | 人员分配页面 |