权限系统核心API

目录

[**1.** **模块说明：** 1](#_Toc325458958)

[**2.** **接口、方法及参数说明** 1](#_Toc325458959)

[2.1. 功能授权模块 1](#_Toc325458962)

[2.1.1. 方法及参数说明 1](#_Toc325458963)

[2.2. 角色操作模块 2](#_Toc325458964)

[2.2.1. 方法及参数说明 2](#_Toc325458965)

[2.3. 用户组操作模块 3](#_Toc325458966)

[2.3.1. 方法及参数说明 4](#_Toc325458967)

[2.4. 权限配置模块 5](#_Toc325458968)

[2.4.1. 功能权限配置 5](#_Toc325458969)

[2.4.2. 数据权限配置: 8](#_Toc325458970)

# **模块说明：**

权限核心部分为四大模块，根据权限操作循序分为：

1、功能授权模块： 对机构进行功能范围的授权管理。

2、角色管理模块： 权限管理操作中的角色管理部分，包括角色的增删改查、角色指派等管理操作。角色指派，即对角色的使用权进行管理,即角色的归属机构可将角色的使用权赋予它的下级机构的管理操作。

3、用户组管理模块：权限管理操作中的用户组部分，包括用户组的增删改查等管理操作。

4、人员配置模块：各机构人员的权限管理部分，包括用户功能权限管理、用户数据权限管理等。

# **接口、方法及参数说明**



## 功能模块

实现对机构功能的相关操作。包括功能管理及功能授权。

接口： com.sinosoft.one.rms.service.facade.TaskService;

实现类: com.sinosoft.one.rms.service.facade.TaskServiceSpringImpl;

### 方法及参数说明

##### public void taskAuth(String userCode,String comCode,List<String> taskIDs);

说明：功能授权，上级机构将功能授予下级机构，无返回值。使用对比旧数据方式，即传入的TaskIds有新值才进行新增、相同则不变、没有则删除，删除时同时删除其机构的所有下级机构的该taskID。

参数：userCode: String，操作授权人的用户代码。

comCode: String，操作授权人的机构代码。

taskIDs: List<String>，所操作的功能ID集合。

##### Public Set<Task> findTaskAuthByComCode(String comCode);

说明：根据机构查询功能授权，获得机构当前功能集合。

参数： comCode: String，操作授权人的机构代码。

返回：Set<Task>,无重复值的Set<Task>集合。

##### public Set<Task> findTaskByRole(List<String> RoleIDs,String comCode);

说明：根据角色ID查询角色下的功能。

参数：RoleIDs :List<String>,角色ID集合。

comCode：String,所查询的机构代码。

##### 新public Set<Task> findTaskAuthByComCodeAndsysFlag (String comCode,String sysFlag);

说明：根据机构代码及系统标志查询功能授权，获取机构下的功能集合。

参数：sysFlag: String,系统标志。

comCode：String,所查询的机构代码。

返回：Set<Task>功能集合

##### 新public Set<Task> findTaskByRoleAndsysFlag (List<String> RoleIDs,String comCode,String sysFlag);

说明：根据角色ID、机构代码及系统标志查询角色关联的功能。

参数：RoleIDs :List<String>,角色ID集合。

comCode：String,所查询的机构代码。

sysFlag: String,系统标志。

返回：Set<Task>功能集合

##### 新public void addTaskHasSysFlag (String taskId,String name ,String menuURL,String menuName ,String des,String isAsMenu,String parentId,String loginUserCode,String loginComCode,String SysFlag);

说明：添加功能数据。功能创建后将直接执行授权操作，授权的机构为登陆机构，如果设定功能为默认时则将功能直接授权为登陆机构，否则授权为所有机构。

参数：taskId：功能ID

Name：功能名称

manuUrl：菜单链接

manuName：菜单名称

des：功能描述

parentId：父功能id

isAsMenu：是否为菜单

loginUserCode:登陆用户代码

loginComCode：登陆机构代码

SysFlag：功能系统标志

##### 新public void updateTaskHasSysFlag (String taskId,String name ,String menuURL,String isValidate,String menuName ,String des,String isAsMenu,String SysFlag,String loginComCode);

说明：更新功能数据。

参数：taskId：功能ID

Name：功能名称

manuUrl：菜单链接

isValidate：是否有效

manuName：菜单名称

des：功能描述

isAsMenu：是否为菜单

loginComCode：登陆机构代码

SysFlag：功能系统标志

##### 新public Task findTaskById (String taskId);

说明：根据id查询功能。

参数：taskId：功能ID

返回：Task功能对象。

##### 新public List<Task> findAllTask (String taskId);

说明：不含任何逻辑查询所有功能。

## 角色操作模块

实现对角色相关的操作。主要操作包含角色相关的增删改查、与角色指派等。

接口： com.sinosoft.ebusiness.rms.service.facade.RoleService;

实现类: com.sinosoft.ebusiness.rms.service.spirng.RoleServiceSpringImpl;

### 方法及参数说明

##### public void addRoleByType addRoleByType(String comCode, String userCode, List<String> TaskIDs,String roleName, String des,String type);

说明：新增角色。

参数：comCode：String,当前操作人的机构代码。

userCode：String, 当前操作人的用户代码。

TaskIDs：List<String>，新建角色所关联的功能iD集合。

roleName：String，新建角色角色名。

des：String，新建角色角色描述。

type：String，角色类型。

##### public void deleteRole(String roleId, String comCode);

说明：删除角色，在该操作中根据传入的机构代码判断，角色是否是本机构所创建，如果是则删除角色记录，如果不是则删除角色指派记录。

参数：roleId：String，所删除角色的ID。

comCode：String，当前操作人机构代码。用于当做删除时对比条件。

##### Public void updataRoleByIDAndType (String RoleID,List<String> comCodes,String logingComCode, String userCode,List<String> TaskIDs,String roleName,String des，String type);

说明：更新角色，同时根据List<String> comCodes更新角色在当前操作机构的下级机构指派。

参数：RoleID：String，所更新角色的ID。

logingComCode：String，当前操作的机构代码。

userCode：String,当前操作的用户代码。

TaskIDs: List<String>,角色更新的关联功能ID集合。

comCodes：String ,角色更新时操作的机构代码集合。

roleName：String，新建角色角色名。

des：String，新建角色角色描述。

type：String，角色类型。

##### public Role findRoleById( String roleId);

说明：根据角色ID查询单个角色对象。

参数：RoleID：String，所查询角色的ID。

返回：Role：角色对象。

##### public Page findRole(String comCode,String roleName, int pageNo, int pageSize);

说明：查询当前机构下可用的所有角色。

参数：comCode: String,所查询的机构代码。

roleName: String,可选查询条件角色名称。

PageNo: int,分页属性，页号。

pageSize: int ,分页属性，页大小。

返回：page：翻页显示对象。

##### public List<RoleDesignate> findRoleByCom(String comCode );

说明：根据机构代码查询角色指派。可根据返回指派对象RoleDesignate 获取角色相关的信息。

参数:comCode: String,所查询指派的机构代码。

返回:List<RoleDesignate>:RoleDesignate类型的List角色指派对象集合。可通过对象RoleDesignate 获取角色相关的信息。

##### public Set<Role> findRoleByGroup(List<String> groupIDs,String comCode);

说明：根据传入的用户组ID集合用户组关联的角色，查询时需要传入机构代码查询角色指派信息后过滤角色集合。

参数: groupIDs：List<String>，用户ID集合。

comCode：String，机构代码。

返回：Set<Role>：实体类Role类型的Set集合。

##### public void roleDesignate(List<String> roleIDs, String comCode,String userCode);

说明：角色指派操作，既将角色的使用权赋予下级机构的操作。同时指派到新的机构时为该机构创建一个默认用户组并将默认用户组关联上该角色。无返回值。

参数：roleIDs：List<string>，被指派的角色。

comCode：string,被指派的机构代码。

userCode：String,当前操作人用户代码。

## 用户组操作模块

实现对用户组相关操作的模块。主要操作包含用户组关的增删改查。

接口: com.sinosoft.ebusiness.rms.service.facade.GroupService;

实现类: com.sinosoft.ebusiness.rms.service.spirng.GroupServiceSpringImpl;

### 方法及参数说明

##### public void saveGroup(String userCode,String comCode,String groupName,String des, List<String> roleIDs);

说明：用户组增加操作，无返回值。

参数：userCode：String,操作的用户代码。

comCode：String,操作的机构代码。

groupName：String,新增用户组的用户组名。

des：String，新增用户组的用户组描述。

roleIDs: String，新增用户组所关联的角色ID集合。

##### public void updataGroup(String groupID,String userCode,String comCode,String groupName,String Des,List<String> RoleIDs);

说明：更新用户组操作。

参数：groupID:string，所更新的用户组ID。

userCode：String，当前操作的用户代码。

comCode：String，当前操作的机构代码。

groupName: String, 更新用户组的用户组名称。

Des：String，更新用户组的用户组描述。

##### public Group findGroupByID(String groupId);

说明：查询用户组获得单个用户组对象。

参数：groupId：String，用户组ID。

返回：Group：实体类用户组对象。

##### public Page findGroup(String name,String comCode, int pageNo, int pageSize);

说明：查询用户组获得按条件查询的用户组集合。

参数：name：String，用户组名，查询条件。

comCode：String,操作人机构代码，查询条件。

PageNo: int,分页属性，页号。

pageSize: int ,分页属性，页大小。

返回：page：翻页显示对象。

##### public List<Group> findGroupByCom(String comCode);

说明：根据机构查询机构下所有用户组。

参数：comCode：string,机构代码。

返回：List<Group>:实体类Group类型的List集合。

##### public Page findEmployeByGroup(String groupId ,String comCode,int pageSize,int pageNo);

说明：根据用户组查询该用户组下的组成员。

参数：groupId：String，用户组ID。

comCode：String，机构代码。

PageNo: int,分页属性，页号。

pageSize: int,分页属性，页大小。

返回：page：翻页显示page对象。

##### public List<Group> findGroupByUser(String userCode);

说明：根据人员用户组代码，获得该人员所在的用户组集合。

参数：userCode：用户代码。

返回：List<Group>:实体类Group类型的List集合。

##### public void deleteGroup(String GroupId);

说明：根据ID删除用户组

参数：GroupId：用户组代码。

##### public void addGroupUser (String userCode,String comCode, String groupId);

说明：添加用户组成员

参数：userCode：String,用户代码。

comCode：String,用户的权限机构代码。

groupId: String,用户组ID。

## 权限配置模块

实现对人员功能权限、数据规则相关配置操作的模块。

### 功能权限配置

实现人员功能权限的管理操作

接口: com.sinosoft.one.rms.service.facade.RmsService;

实现类:com.sinosoft.one.rms.service.spirng.RmsServiceSpringImpl;

这3个类中的方法统一由RmsService向外发布。

#### **方法及参数说明**

###### Public Void addUserPower (String userCode , String comCode, List<String> groupIDs ,List <String> roleIDS,List <String> ExcTaskIDs,String sysFlag);

说明：新增一个人员权限记录，通过excTaskIDs除外权限ID集合控制权有效限功能范围。更新操作也使用该方法。

参数：userCode：String,所添加权限的用户代码。

comCode：String,所添加权限的用户机构代码。

groupIDs：List<String>,所添加权限的用户归属到的用户组。

roleIDs: List<String>,用户组关联下的角色ID集合。

excTaskIDs：List<String>,所添加权限的用户配置的除外功能。

sysFlag: String,系统标志。

###### public List<Company> deleteUserPower (String userCode,String comCode)

说明：删除用户权限。

参数：userCode：String,用户代码。、

comCode：String,用户的权限机构代码。

返回：List<Company>:实体类机构Company类型的List集合。

###### public Page findEmployees(String comCode,String userCode,String userName,int pageNo,int pageSize)

说明：根据机构代码获得满足userCode,userName条件的，在该机构下有有效权限的人员。

参数：comCode：String,所查询的机构代码。

userCode：String,用户代码，查询条件。

userName：String, 用户名，查询条件。

PageNo: int,分页属性，页号。

pageSize: int ,分页属性，页大小。

返回：page：翻页显示对象。

###### public Page findNEmployees(String comCode,String userCode,String userName,int pageNo,int pageSize)

说明：根据机构代码获得满足userCode,userName条件的，在该机构下有有效权限之外的人员。

参数：comCode：String,所查询的机构代码。

userCode：String,用户代码，查询条件。

userName：String, 用户名，查询条件。

PageNo: int,分页属性，页号。

pageSize: int ,分页属性，页大小。

返回：page：翻页显示对象。

###### public Set<Task> findExcPower(String userCode, String comCode)

说明：查询用户的除外功能权限。

参数：userCode：String，所查询的用户代码。

comCode：String,所查询的机构代码。

返回：Set<Task>：实体类功能Task类型的无重复Set集合。

###### public Set<Task> findTaskMultSysByUserCode (String userCode, String comCode，String sysFlag)

说明：获得用户的完整有效功能权限。即出去除外权限后获得的功能权限集。

参数：userCode：String，所查询的用户代码。

comCode：String,所查询的机构代码。

sysFlag：String,系统标志

返回：Set<Task>：实体类功能Task类型的无重复Set集合。

###### public UserPower findUserPowerbyComUser(String userCode,String comCode)

说明：根据用户代码机构代查询用户权限记录。

参数：userCode:String,用户代码。

userCode:String,机构代码。

返回：UserPower：实体类UserPower对象。

###### public UserPower findUserPowersByUserCode (String userCode,String comCode)

说明：根据用户代码机构代查询用户权限记录。

参数：userCode:String,用户代码。

userCode:String,机构代码。

返回：UserPower：实体类UserPower对象。

## 机构部分

实现机构数据相关的操作包含一个实体接口

CompanyModelInterface

使用方式：public class **Company** implements **CompanyModelInterface**

### CompanyServiceInterface:

请必要实现该接口以下方法

###### public List<? extends CompanyModelInterface>findComAndNextSubCom(String comCode);

说明：根据机构代码获得本机构及下级子机构集合。

参数：comCode：string,机构代码。

返回：List<? extends CompanyModelInterface >： 实现了机构实体接口的实体集合。

###### public List<? extends CompanyModelInterface>findNextSubCompany (String comCode)

说明：根据机构代码获下级子机构集合。

参数：comCode：string,机构代码。

返回：List<? extends CompanyModelInterface >： 实现了实体接口CompanyModelInterface的实体集合。

###### public List <String> findNextSubComCodes (String comCode)

说明：查询下级子机构代码。

参数：comCode：string,机构代码。

返回：List <String>：机构代码集合.

###### public boolean isExtSubCom (String comCode)

说明：是否有下级机构。

参数：comCode：string,机构代码。

返回：boolean：true为有下级/false没有下级.

###### List<Company> findLevFourCom(String comCode)

###### public <T extends CompanyModelInterface>T findCompanyByComCode (String comCode)

说明：根据机构代码查询机构实。

参数：comCode：string,机构代码。

返回：实现了实体接口CompanyModelInterface的实体.

###### public List<String> findAllNextSubComCodesByComCode (String comCode)

说明：根据机构代码查询所有下级机构代码。

参数：comCode：string,机构代码。

返回：List <String>：机构代码集合.

###### public List <T extends CompanyModelInterface> findAllNextLevelCompanybyComCode (String comCode)

说明：根据机构代码查询所有下级机构集合。

参数：comCode：string,机构代码。

返回：List<? extends CompanyModelInterface >： 实现了实体接口CompanyModelInterface的实体集合。

###### public List<String> findComCodebySuperComCode (String comCode)

说明：根据父机构机构代码查询机构代码。

参数：comCode：string,父机构代码。

返回：List <String>：机构代码集合.

###### public List <T extends CompanyModelInterface> findAllNextLevelCompanybyComCode (String SuppercomCode,String comType)

说明：根据父机构机构代码、机构类型查询机构代码。

参数：comCode：string,父机构代码。

comType：string,机构类别/级别

返回：List<? extends CompanyModelInterface >： 实现了实体接口CompanyModelInterface的实体集合。

###### Public List <T extends CompanyModelInterface> findCompanysByComcodes (List<String> comCodes)

说明：根据机构代码集合查询机构集合。

参数：comCodes：List<String>,机构代码集合。

返回：List<? extends CompanyModelInterface >： 实现了实体接口CompanyModelInterface的实体集合。

## 人员部分部分

实现人员数据相关的操作包含一个实体接口

EmployeModelInterface

使用方式：public class **Employe** implements **EmployeModelInterface**

### EmployeServiceInterface：

请必要实现该接口以下方法

###### public <T extends EmployeModelInterface>T findUserByCode(String userCode)

说明：根据用户代码查询用户信息。

参数：userCode：String，用户代码。

返回：实现了实体接口EmployeModelInterface的实体.

###### Public void updatePassword(String userCode,String userPassword)

说明：修改用户密码。

参数：userCode：String，用户代码。

userPassword: ：String，用户新密码。

###### public List <T extends EmployeModelInterface> findUserByComCode(String comCode)

说明：根据机构代码查询机构下的所有员工。

参数：comCode：String，机构代码。

返回：List<? extends EmployeModelInterface >： 实现了实体接口EmployeModelInterface的实体集合。

###### public Page findUserNoInUserCodes(List<String> userCodes,String comCode ,String userName,String userCode,int pageNo, int pageSize);

说明：根据机构代码，用户名，用户代码等条件查询不属于用户代码集合下的用户。

参数：

userCodes：List<String>，用户代码集合。

userName：String，查询条件的用户名。Like匹配

userName：String，查询条件的用户代码。Like匹配

comCode：String，查询条件的机构代码。

返回：page：翻页显示对象。

###### public Page findUserInUserCodes(List<String> userCodes,String comCode ,String userName,String userCode,int pageNo, int pageSize);

说明：根据机构代码，用户名，用户代码等条件查询属于用户代码集合下的用户。

参数：

userCodes：List<String>，用户代码集合。

userName：String，查询条件的用户名。Like匹配

userName：String，查询条件的用户代码。Like匹配

comCode：String，查询条件的机构代码。

pageNo：int，页号。

pageSize：int，每页大小。

返回：page：翻页显示对象。