目录

[1 用户 3](#_Toc497248931)

[1.1 机构管理 3](#_Toc497248932)

[1.1.1 查找机构\部门列表接口 3](#_Toc497248933)

[1.1.2 查找机构\部门及其所有子孙部门列表信息接口 5](#_Toc497248934)

[1.1.3 查找部门列表接口 7](#_Toc497248935)

[1.1.4 查找机构/部门信息接口 8](#_Toc497248936)

[1.1.5 增加/修改机构/部门信息接口 9](#_Toc497248937)

[1.1.6 修改机构/部门顺序接口 10](#_Toc497248938)

[1.1.7 删除机构/部门接口 11](#_Toc497248939)

[1.1.8 机构/部门增加/修改员工 12](#_Toc497248940)

[1.1.9 查找机构/部门下员工列表接口 14](#_Toc497248941)

[1.1.10 部门配置员工接口 15](#_Toc497248942)

[1.1.11 查找前端员工信息接口 16](#_Toc497248943)

[1.1.12 删除员工接口 17](#_Toc497248944)

[1.1.13 批量导入机构/部门员工接口 18](#_Toc497248945)

[1.1.14 批量导出机构/部门员工接口 20](#_Toc497248946)

[1.1.15 查找未配置在本部门人员列表 有什么用？ 21](#_Toc497248947)

[1.1.16 查看人员登录记录接口 22](#_Toc497248948)

[1.1.17 查找机构/部门下管理员列表接口 23](#_Toc497248949)

[1.1.18 增加/修改管理员信息接口 24](#_Toc497248950)

[1.1.19 管理员匹配员工接口 25](#_Toc497248951)

[1.1.20 员工匹配管理员接口 26](#_Toc497248952)

[1.1.21 删除管理员接口 28](#_Toc497248953)

[1.1.22 机构/部门信息转化接口 29](#_Toc497248954)

[1.2 数据词典管理 29](#_Toc497248955)

[1.2.1 查找数据词典列表接口 29](#_Toc497248956)

[1.2.2 查找数据词典信息接口 31](#_Toc497248957)

[1.2.3 增加/修改数据词典接口 32](#_Toc497248958)

[1.2.4 删除数据词典接口 33](#_Toc497248959)

[1.2.5 修改数据词典顺序接口 35](#_Toc497248960)

[1.2.6 导入数据词典 36](#_Toc497248961)

[1.2.7 更新数据词典接口 暂时没有用 37](#_Toc497248962)

[1.2.8 导出数据词典到excel接口 37](#_Toc497248963)

[1.2.9 导入机构数据词典到excel接口 38](#_Toc497248964)

[1.2.10 导出机构数据词典到excel接口 39](#_Toc497248965)

[1.2.11 配置机构数据词典 40](#_Toc497248966)

[1.2.12 删除机构数据词典 41](#_Toc497248967)

[1.2.13 更新机构数据词典，这个是做什么？没有用了 42](#_Toc497248968)

[1.2.14 查询机构已配置的数据词典列表 42](#_Toc497248969)

[1.2.15 生成pid获取机构中数据词典Json接口 43](#_Toc497248970)

[1.3 角色管理 45](#_Toc497248971)

[1.3.1 查找角色列表接口 45](#_Toc497248972)

[1.3.2 查找角色信息接口 47](#_Toc497248973)

[1.3.3 删除角色接口 48](#_Toc497248974)

[1.3.4 修改角色顺序接口 49](#_Toc497248975)

[1.3.5 增加/修改角色接口 50](#_Toc497248976)

[1.3.6 导入角色接口 51](#_Toc497248977)

[1.3.7 导出角色接口 53](#_Toc497248978)

[1.3.8 导入机构角色接口 54](#_Toc497248979)

[1.3.9 导出机构角色接口 56](#_Toc497248980)

[1.3.10 配置机构\部门前端角色接口 57](#_Toc497248981)

[1.3.11 配置机构\部门后台角色接口 58](#_Toc497248982)

[1.3.12 删除机构\部门前端角色接口 59](#_Toc497248983)

[1.3.13 删除机构\部门后台角色接口 60](#_Toc497248984)

[1.3.14 查找机构或部门已配置的前端角色列表 61](#_Toc497248985)

[1.3.15 查找机构或部门已配置的后台角色列表 62](#_Toc497248986)

[1.3.16 配置员工前端角色接口 63](#_Toc497248987)

[1.3.17 配置管理后台角色接口 64](#_Toc497248988)

[1.3.18 删除员工前端角色接口 65](#_Toc497248989)

[1.3.19 删除管理员后台角色接口 66](#_Toc497248990)

[1.3.20 查看机构/部门前端人员的角色列表 67](#_Toc497248991)

[1.3.21 查看机构/部门管理员的角色列表 68](#_Toc497248992)

[1.3.22 查看机构/部门前端角色下人员列表 70](#_Toc497248993)

[1.3.23 查看机构/部门后台角色下管理员列表 71](#_Toc497248994)

[1.4 菜单管理 72](#_Toc497248995)

[1.4.1 新增/修改菜单 72](#_Toc497248996)

[1.4.2 查找菜单列表接口 74](#_Toc497248997)

[1.4.3 查找菜单信息接口 75](#_Toc497248998)

[1.4.4 修改菜单顺序 76](#_Toc497248999)

[1.4.5 删除菜单接口 78](#_Toc497249000)

[1.4.6 新增/修改功能 79](#_Toc497249001)

[1.4.7 查找功能列表接口 80](#_Toc497249002)

[1.4.8 查找功能信息接口 81](#_Toc497249003)

[1.4.9 删除功能接口 82](#_Toc497249004)

[1.4.10 查找机构前端公共菜单功能、按钮列表 83](#_Toc497249005)

[1.4.11 查找机构后台公共菜单功能、按钮列表 85](#_Toc497249006)

[1.4.12 配置机构前端/后台公共菜单功能、按钮列表 86](#_Toc497249007)

[1.4.13 查找机构后台菜单功能列表（全部带层级关系） 88](#_Toc497249008)

[1.4.14 查找部门前/后端菜单功能列表 89](#_Toc497249009)

[1.4.15 新增部门前/后端菜单 90](#_Toc497249010)

[1.4.16 删除部门前/后端菜单 92](#_Toc497249011)

[1.4.17 查找机构或部门前端角色菜单及按钮 93](#_Toc497249012)

[1.4.18 查找机构或部门后端角色菜单及按钮 94](#_Toc497249013)

[1.4.19 新增机构或部门前/后端角色菜单及按钮 95](#_Toc497249014)

[1.4.20 删除机构或部门前端角色菜单及按钮 97](#_Toc497249015)

[1.4.21 查询人员前/后端私有菜单及按钮 99](#_Toc497249016)

[1.4.22 新增人员前/后端私有菜单及按钮 100](#_Toc497249017)

[1.4.23 删除人员前/后端私有菜单及按钮 102](#_Toc497249018)

[1.4.24 查找人员前/后端菜单 104](#_Toc497249019)

[1.4.25 配置机构后台按钮接口 105](#_Toc497249020)

[1.4.26 查询机构后台按钮列表接口 106](#_Toc497249021)

[1.4.27 删除机构后台按钮接口 107](#_Toc497249022)

[1.5 按钮管理 109](#_Toc497249023)

[1.5.1 获取按钮列表 109](#_Toc497249024)

[1.5.2 获取按钮详情 110](#_Toc497249025)

[1.5.3 修改和增加按钮 111](#_Toc497249026)

[1.5.4 删除按钮 112](#_Toc497249027)

**管理员单部门，用户可以多部门**

1. 用户
   1. 机构管理
      1. 查找机构\部门列表接口

**a.接口url:**

/pub/admin/user/organization/findOrgList.do

**b.接口说明:**

该接口用于根据参数查询机构\部门列表

相关信息：

1. 超级管理员没有orgid，参数orgid值为空或者为一个固定数-9个9，查询平台所有机构。
2. 机构\部门管理员，参数orgid值为本机构id，只能看到本机构

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值 |
| findOrgList | | Info | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| pageSize | int | | 每页显示信息数 | |
| nowPage | Int | | 当前页 | |
| title | String | | 输入的机构名称 | |
| parentOrg | String | | 父机构ID | |
| orgid | String | | 当前登录人的orgid | |
| peid | String | | 用户id | |

**d.返回值示例与说明**

{

"infos": [

{

"description": "渠道商",

"ftitle": "",

"id": "e3f10943-a6dc-4690-b489-d4184c9f180e",

"isdelete": 0,

"pid": "",

"pids": "",

"title": "渠道商",

"zorder": 4

},

{

"description": "云医机构",

"ftitle": "",

"id": "14b8c8f3-7338-4d4c-a924-37c3db7a7fb5",

"isdelete": 0,

"pid": "",

"pids": "",

"title": "云医机构",

"zorder": 10

}

],

"page": {

"lines": 6,

"page": 1,

"pageSize": 1000,

"pageStart": 0,

"pages": 1

},

"result": {

"flag": 1,

"msg": "查询成功",

"orgid": "",

"rid": "",

"token": "",

"ttime": "",

"userId": ""

},

"total": 6

}

**e.功能实现说明**

1. 根据参数orgid, title,查找机构：

① 判断orgid是否为空，如果为空：

1） 如果orgid为空 ，拼接查找条件，查找zjxx\_orgnization （机构｛部门｝基础表）中pid字段值为空值的数据;

2）此段逻辑有问题，无法判断是否是机构管理员，先查orgid=orgid的数据 ~~如果orgid不为空，根据peid和orgid拼接查找条件，查询zjxx\_adminorg （机构｛部门｝员工表-后台）数据（有且只有一条），判断数据中orgids（管理员所属部门id集合）中~~

~~包含orgid：就是机构管理员，查找机构表中id字段值为orgid值的数据;然后再执行②③④⑤⑥~~

~~不包含orgid：就是部门管理员，获取orgids查找机构表中id字段值为orgids值的数据;然后再执行②③④⑤~~

② 当title不为空时拼接查找条件，查找机构表中title字段的值like参数title值的数据;

③ 拼接查找条件，查找机构中isdelete字段值为0的数据;

④根据参数nowpage，pageSize计算出查找条件，追加查找条件limit;

⑥ 最后根据拼接的sql语句执行查询操作，返回查询列表，并根据返回的数据列表，分页实体类Page,返回信息类Info整合数据。

* + 1. 查找机构\部门及其所有子孙部门列表信息接口

**a.接口url:**

/pub/admin/user/organization/findOrgAllListInfo.do

**b.接口说明:**

该接口用于查询机构以及该机构的子孙部门

相关信息：

1. 递归查询所有子孙部门。
2. 返回信息带层级关系

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值 |
| findOrgAllListInfo | | Info | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构ID | |

**d.返回值示例与说明**

{

"infos": [

{

"description": "",

"ftitle": "",

"id": "253ce0f1-fbb0-46cb-b738-37dcef6d9aac",

"isdelete": 0,

"pid": "a628d0c4-3b6a-4efa-a2ff-b1090b8fc593",

"pids": "14b8c8f3-7338-4d4c-a924-37c3db7a7fb5,a628d0c4-3b6a-4efa-a2ff-b1090b8fc593,",

"title": "部门儿子",

"zorder": 0

},

{

"description": "",

"ftitle": "",

"id": "4da59b9b-5123-4e37-b719-d726d11cc036",

"isdelete": 0,

"pid": "253ce0f1-fbb0-46cb-b738-37dcef6d9aac",

"pids": "14b8c8f3-7338-4d4c-a924-37c3db7a7fb5,a628d0c4-3b6a-4efa-a2ff-b1090b8fc593,253ce0f1-fbb0-46cb-b738-37dcef6d9aac,",

"title": "部门孙子",

"zorder": 0

}

],

"page": null,

"result": {

"flag": 1,

"msg": "查询成功",

"orgid": "",

"rid": "",

"token": "",

"ttime": "",

"userId": ""

},

"total": null

}

**e.功能实现说明**

(1)根据参数orgid查找机构（部门）：

① 查找机构表中pid字段值为orgid字段值的数据,然后再根据查找到的数据id递归查询机构表中pid字段值为数据id的数据;

分别递归查询 并且该数据的isdelete字段值为0的数据。

* + 1. 查找部门列表接口

**a.接口url:**

/pub/admin/user/organization/findOrgChildListInfo.do

**b.接口说明:**

该接口用于查询机构下部门列表的信息

相关信息：

1. 通过机构或部门id，只查询儿子。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值 |
| findOrgChildListInfo | | Info | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| pid | String | | 父机构ID | |
| pageSize | int | | 每页显示信息数 | |
| nowPage | Int | | 当前页 | |

**d.返回值示例与说明**

{

"infos": [

{

"description": "",

"ftitle": "",

"id": "a628d0c4-3b6a-4efa-a2ff-b1090b8fc593",

"isdelete": 0,

"pid": "14b8c8f3-7338-4d4c-a924-37c3db7a7fb5",

"pids": "",

"title": "部门父亲",

"zorder": 11

}

],

"page": null,

"result": {

"flag": 1,

"msg": "查询成功",

"orgid": "",

"rid": "",

"token": "",

"ttime": "",

"userId": ""

},

"total": null

} **e.功能实现说明**

(1)根据参数pid查找机构（部门）：

查找机构表中pid字段值为pid字段值并且该数据的isdelete字段值为0的数据，返回到页面

* + 1. 查找机构/部门信息接口

**a.接口url:**

/pub/admin/user/organization/findOrgById.do

**b.接口说明:**

该接口用于根据机构ID查找机构（部门）详细信息

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值 |
| findOrgById | | Info | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构\部门ID | |

**d.返回值示例与说明**

{

"infos": [

{

"description": "云医机构",

"ftitle": "",

"id": "14b8c8f3-7338-4d4c-a924-37c3db7a7fb5",

"isdelete": 0,

"pid": "",

"pids": "",

"title": "云医机构",

"zorder": 10

}

],

"page": null,

"result": {

"flag": 1,

"msg": "查询成功",

"orgid": "",

"rid": "",

"token": "",

"ttime": "",

"userId": "".

},

"total": null

} **e.功能实现说明**

1. 机构有三张表，需要将三张表的数据都拿到
2. 根据参数orgid查找机构表中id字段值为参数orgid并且该数据的isdelete字段值为0的数据。
   * 1. 增加/修改机构/部门信息接口

**a.接口url:**

/pub/admin/user/organization/addOrUpdateOrg.do

**b.接口说明:**

该接口用于增加/修改机构/部门信息

相关信息：

1. 增加部门注意pid（父级机构id），pids（父机构或部门id集合）字段。
2. 修改部门信息，部门层级不能修改即pid（父级机构id），pids（父机构或部门id集合）字段不能变。前台页面也要做限制。
3. 系统设置-网路防火墙：提交内容屏蔽敏感词汇

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| addOrUpdateOrg | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgnization | ZjxxOrgnization （对象） | | 机构｛部门｝基础表表类对象 | |
| functioninfo | zjxx\_orgnization\_functioninfo（对象） | | 机构详情表-机构表功能分表对象 | |
| otherinfo | zjxx\_orgnization\_info（对象） | | 机构详情表-机构表分表对象 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "增加机构（部门）成功" //返回的信息；

}

}

**e.功能实现说明**

提交ip函数（禁止和允许）、屏蔽敏感词函数

机构有三张表，前端传三个对象对应三张表，需要将对应数据放入对应的表中

（1）根据对象参数orgnization 判断对象中id字段值是否为空：

① id值为空时，生成UUID赋值给三个对象中id字段，然后再判断该对象中pid 字段值是否为空：

1. 若为空则直接将该三个对象插入对应的机构表中，pids就是orgnization 对象的id的值;
2. 若不为空，则在机构表中查出id为对象中pid字段值的那条数据，获取这条数据中的pids然后加上orgnization 对象的id的值，然后把生成的pids赋值给数据orgnization 对象的pids。

② id值不为空时：

1. 根据三个对象参数id查找对应机构表中id字段值为参数id值的数据，根据该数据中对应的参数修改信息。目前只要前台传了，后台就会修改。后台未做字段修改限制。比如：pid和pids要做限制，不能修改。
   * 1. 修改机构/部门顺序接口

**a.接口url:**

/pub/admin/user/organization/updateOrgOrder.do

**b.接口说明:**

该接口用于修改部门

相关信息：

1. 同级排序。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| updateOrgOrder | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| idStr | String[] | | 遍历页面数据的id的字符串数组 | |
| orderStr | String[] | | 遍历页面数据顺序的得到的字符串数组 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "修改顺序成功" //返回的信息；

}

}

**e.功能实现说明**

1.根据提交ip调用禁止和允许ip提交函数函数。

2.两个数组长度不一致或者为空直接返回错误信息。

3.根据参数idStr ，orderStr ，分别遍历数组，修改机构表中对应的数据的顺序字段值。

* + 1. 删除机构/部门接口

**a.接口url:**

/pub/admin/user/organization/deleteOrg.do

**b.接口说明:**

该接口用于删除或批量删除机构或部门用

相关信息：

1. 伪删除
2. 暂不考虑机构/部门人员等相关信息的处理。

**c.接口定义:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteOrg | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| idStr | String[] | | 机构/部门id的字符串数组 | |

**d.返回值示例与说明**

{

"flag": 1,

"msg": "删除成功",

"orgid": "",

"rid": "",

"token": "",

"ttime": "",

"userId": ""

}

**e.功能实现说明**

根据提交ip调用禁止和允许ip提交函数。

1. 根据参数idStr删除机构：

① 解析并遍历idStr，每次遍历得到一个id参数，根据参数id查找机构表，

然后修改该数据的isdelete字段，将该字段值改为“1”即可。

* + 1. 机构/部门增加/修改员工

**a.接口url:**

/pub/admin/user/organization/addStaff.do

**b.接口说明:**

该接口用于增加员工

相关信息：

1. 添加的是前端用户
2. 同步增加、修改人员基础表和机构人员表相关字段
3. 部门添加人员，机构人员表需添加机构人员信息
4. 同步更新机构人员表中权限字段
5. 修改人员信息不能修改人员所属机构/部门，页面做隐藏
6. 系统设置-网路防火墙：提交内容屏蔽敏感词汇
7. 系统设置-用户设置-前端用户设置：注册用户名限制、注册密码限制、注册用户邮箱唯一性检查、同一IP注册间隔限制、用户名保留关键
8. 根据提交ip和提交内容调用禁止和允许ip提交函数。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| addStaff | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| people | ZjxxPeople （对象） | | 人员表类对象 | |
| orgid | String | | 机构ID | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "修改/添加员工成功" //返回的信息；

}

}

**e.功能实现说明**

1. 调用屏蔽相关词汇函数和后台允许ip提交函数。
2. 调用查找系统设置参数的方法：查询系统参数表-判断系统设置-用户设置-前端用户设置：注册用户名限制、注册密码限制、注册用户邮箱唯一性检查、同一IP注册间隔限制、用户名保留关键。
3. 根据对象参数people判断该对象中的id字段是否为空：
4. 若id为空，生成一个UUID赋值给该对象的id字段：

判断帐号是否存在，存在返回“帐号已存在”

1) 查询orgid是否是部门id，根据orgid查询zjxx\_orgnization （机构｛部门｝基础表），获取对应信息orgnizationinfor，判断orgnizationinfor数据中pid字段是否为空。

若不为空为部门id，获取orgnizationinfor数据中的pids的值

1. pids值的第一个id就是机构id（可能会出现脏数据）

2、select \* from zjxx\_orgnization where id in “‘+pids+’”(需要再查一遍sql，查不到返回错误)

根据方法1获取机构id，然后将人员信息插入zjxx\_people （人员基础表）。机构id 添加的是机构人员信息，其中机构人员信息中的所属部门id集合就是orgnizationinfor数据中的pids（将,替换成| 例如：A|B|C），理由：所属部门id集合orgids 里面就包含了所有父级部门，父级部门查询所属人员的时候只要like一下这个字段就行）。

若为空，就是机构id。然后将人员信息插入zjxx\_people （人员基础表），zjxx\_frontorg（机构｛部门｝员工表-前端）需添加机构人员信息，其中所属部门id集合为空。

3) 调用人员权限函数，同步更新机构人员表中 机构人员信息的权限字段。

② 若id不为空:

1) 根据参数id查找员工表中id字段值为参数id值的数据，根据对应的 参数修改信息。同步修改zjxx\_frontorg（机构｛部门｝员工表-前端）人员id=id的数据的相应相关字段。

人员所属部门发生变化，不是在修改人员这里修改。是在部门管理人员，部门可以删除\配置部门人员，这样人员所属部门就改变了。

人员所属机构发生变化，如果是机构删除人员，人员基本信息也会删除。所以人员所属机构变化只能删除人员，重新在另一个机构新建人员。人员无法从一个机构转移到另一个机构。

* + 1. 查找机构/部门下员工列表接口

**a.接口url:**

/pub/admin/user/organization/findpeoplelist.do

**b.接口说明:**

该接口用于查找机构/部门下员工列表

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findpeoplelist | | Info | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构/部门ID | |
| pageSize | int | | 每页显示信息数 | |
| nowPage | Int | | 当前页 | |

**d.返回值示例与说明**

{

"infos": [

{

"accont": "",

"id": "11111111",

"menuids": "",

"orgid": "123456",

"orgids": "",

"peid": " 4054aec8-db28-42e8-a682-5f5b4db851e8",

"personnelbuttonids": "",

"personnelmenuids": "",

"rids": "",

"rtime": "235465641",

"title": "万磊"

}

],

"page": null,

"result": {

"flag": 1,

"msg": "",

"orgid": "",

"rid": "",

"token": "",

"ttime": "",

"userId": ""

},

"total": 1

}

**e.功能实现说明**

1. 根据参数orgid查询zjxx\_frontorg（机构｛部门｝员工表-前端）orgid=orgid or orgid in orgids 并根据人员id去重数据。
   * 1. 部门配置员工接口

**a.接口url:**

pub/admin/user/organization/setDevStaff.do

**b.接口说明:**

该接口用于给部门配置员工

**c.接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| setDevStaff | Result | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| peopleids[] | String | 人员ID集合，逗号隔开 |
| orgid | String | 部门ID |

**d.返回值示例与说明**

{

"flag": 1,

"msg": "配置成功",

"orgid": "",

"rid": "",

"token": "",

"ttime": "",

"userId": ""

}

**e.功能实现说明**

根据提交ip调用禁止和允许ip提交函数。

检测两个参数peopleids[]、orgid是否为空或是否长度为0，有一个为真则返回msg:来源错误。

参数都不为空时，遍历参数peopleids[]，每次遍历得到一个人员id,根据人员id和部门orgid，查zjxx\_frontorg（机构｛部门｝员工表-前端），数据是否已存在，orgids中是否包含部门orgid。

存在：处理下一个人员id数据。

不存在：根据人员id查询出人员信息，通过部门orgid找到该部门的机构id（通过部门id查找部门的机构id，请参考1.1.8接口）和pids，根据机构id和人员查询zjxx\_frontorg（机构｛部门｝员工表-前端），部门数据中的pids（将,替换成| 例如：A|B|C）加入到机构员工信息中orgids（员工所属部门id集合）。

* + 1. 查找员工信息接口

**a.接口url:**

/pub/admin/user/organization/findStaffById.do

**b.接口说明:**

该接口用于根据员工ID查询员工信息

**c.接口定义:**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| findStaffById | Info | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| peid | String | 员工ID |

**d.返回值示例与说明**

{

{

"infos": [

{

"account": "1234567",

"address": "",

"description": "员工2",

"email": "",

"ftitle": "",

"id": "88167148-9d93-4a66-8e31-3d1e567ac163",

"id\_number": "",

"linkruleid": "",

"password": "123456",

"phone": "",

"phonecode": "",

"phonecodeTime": "",

"r\_time": "",

"rorder": 0,

"t\_time": null,

"title": "王婷",

"token": ""

}

],

"page": null,

"result": {

"flag": 1,

"msg": "查询成功",

"orgid": "",

"rid": "",

"token": "",

"ttime": "",

"userId": ""

},

"total": null

}}

**e.功能实现说明**

(1) 根据参数peid查询员工表中id字段值为参数peid值的数据。返回的人员对象，包括密码也给前端。需要将密码设置成空。

* + 1. 删除员工接口

**a.接口url:**

/pub/admin/user/organization/deleteOrgStaff.do

**b.接口说明:**

该接口用于删除机构\或部门下员工

相关信息：

1. 伪删除

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteOrgStaff | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| pid | String | | 员工ID | |
| orgid | String | | 机构或部门id | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "删除员工成功" //返回的信息；

}

}

**e.功能实现说明**

根据提交ip调用禁止和允许ip提交函数。

查询orgid是否是部门id，根据orgid查询zjxx\_orgnization （机构｛部门｝基础表），获取对应信息orgnizationinfor，判断orgnizationinfor数据中pid字段是否为空。

1. 为空：机构id 根据机构orgid和人员id查询机构员工-前端表，获取orgids（员工所属部门id集合）字段，遍历参orgids ，每次遍历得到一个人员部门id,根据人员部门id和人员id删除 zjxx\_porg （机构｛部门｝员工角色表）、zjxx\_frontorg（机构｛部门｝员工表-前端）对应数据。然后根据机构orgid和人员id删除zjxx\_porg （机构｛部门｝员工角色表）、zjxx\_frontorg（机构｛部门｝员工表-前端）对应数据。最后根据人员id查询人员基础表删除该人员数据，人员基础表伪删除
2. 不为空：部门id 根据部门orgid和人员id删除 zjxx\_porg （机构｛部门｝员工角色表）、zjxx\_frontorg（机构｛部门｝员工表-前端）对应数据。根据orgid查询部门所属机构id（通过部门id查找部门的机构id，请参考1.1.8接口），根据机构id和人员id查询机构员工-前端表，获取相应信息。将orgids、rids字段中带orgid的数据删除。
   * 1. 批量导入机构/部门员工接口

**a.接口url:**

/pub/admin/user/organization/importStaff.do

**b.接口说明:**

该接口用于从excel批量导入员工到数据库

相关信息：

导入数据中没有机构/部门信息，默认导入的是当前页面的机构或部门，接口会上传。

导入的人员信息中带有机构/部门和角色信息时，相关信息在数据库中没有，提示相关信息没有的报错。例如：导入A部门。但机构表中没有A部门，报错提示：系统中无A部门

系统设置-网路防火墙：提交内容屏蔽敏感词汇

系统设置-用户设置-前端用户设置：注册用户名限制、注册密码限制、注册用户邮箱唯一性检查、同一IP注册间隔限制、用户名保留关键

系统设置-用户设置-统一设置：允许提交ip、禁止提交ip

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| importStaff | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
|  |  | |  | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "批量导入员工成功" //返回的信息；

}

}

**e.功能实现说明**

1.根据提交ip和提交内容调用禁止和允许ip提交函数、屏蔽敏感词函数。

查询系统参数表-判断系统设置-用户设置-前端用户设置：注册用户名限制、注册密码限制、注册用户邮箱唯一性检查、同一IP注册间隔限制、用户名保留关键。

系统参数表相关字段，先在缓存中查找，没有-去数据库系统参数表查找，并判断是否需要加入缓存，是-添加相应缓存

1. 解析上传的excel文件，然后批量添加信息：

① 添加员工信息到人员表中；

1. 在机构员工表中也要添加对应的信息。
   * 1. 批量导出机构/部门员工接口

**a.接口url:**

/pub/admin/user/organization/exportStaff.do

**b.接口说明:**

该接口用于批量导出员工到excel

相关信息：

只导出人员基础信息，不含密码。Id导出，但在excel中默认隐藏。可参考钉钉的考勤报表。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| exportStaff | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构ID | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "批量导出员工成功" //返回的信息；

}

}

**e.功能实现说明**

1. 根据参数orgid批量导出员工到excel：

① 根据orgid查找人员表中orgid字段值为参数orgid值的数据;

1. 将产找到的数据生成excel文件。
   * 1. 查找未配置在本部门人员列表 有什么用？

**a.接口url:**

/pub/admin/user/organization/ findOrgUnStaffList.do

**b.接口说明:**

该接口用于部门未配置的人员列表。

**c.接口定义:**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| findOrgUnStaffList | Info | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| orgid | String | 机构ID |
| pageSize | int | 每页显示信息数 |
| nowpage | Int | 当前页 |
| Keys[] | String | 键 |
| Values[] | String | 值 |

**d.返回值示例与说明**

{

“infos”:

{

"id": "8128", //员工ID;

"title": "小明", //员工姓名;

"orgid": "子杰信息", //所属机构;

"account": "221231 ", //账号;

"password": "12", //密码;

"idNumber": "123456", //身份证号;

"rTime": "20", //创建时间;

"rorder": "1" //显示顺序;

}

}

**e.功能实现说明**

(1) 根据参数pid查询员工表中id字段值为参数pid值的数据。

* + 1. 查看人员登录记录接口

**a.接口url:**

/pub/admin/user/organization/findFrontLoginRecord.do

**b.接口说明:**

该接口用于查找人员登录信息

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findFrontLoginRecord | | Info | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| pid | String | | 员工ID | |
| isadmin | Int | | 前端==0 后台==1 | |

**d.返回值示例与说明**

{

“infos”:

{

[ "id": "8128", //员工ID;

"title": "小明", //员工姓名;

"orgid": "子杰信息", //所属机构;

"account": "221231 ", //账号;

"password": "12", //密码;

"idNumber": "123456", //身份证号;

"rTime": "20", //创建时间;

"rorder": "1" //显示顺序;

]

}

}

**e.功能实现说明**

1. 根据参数pid，isadmin查询zjxx\_loginlog （机构员工表登录日志）数据：
   * 1. 查找机构/部门下管理员列表接口

**a.接口url:**

/pub/admin/user/organization/findAdminlist.do

**b.接口说明:**

该接口用于查找机构/部门下管理员列表

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findAdminlist | | Info | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构/部门ID | |
| pageSize | int | | 每页显示信息数 | |
| nowPage | Int | | 当前页 | |

**d.返回值示例与说明**

{

“infos”:

{

[ "id": "8128", //员工ID;

"title": "小明", //员工姓名;

"orgid": "子杰信息", //所属机构;

"account": "221231 ", //账号;

"password": "12", //密码;

"idNumber": "123456", //身份证号;

"rTime": "20", //创建时间;

"rorder": "1" //显示顺序;

]

}

}

**e.功能实现说明**

1. 根据参数orgid查询zjxx\_adminorg （机构｛部门｝员工表-后台）orgid=orgid or orgid in orgids 并根据人员id去重数据。
   * 1. 增加/修改管理员信息接口

**a.接口url:**

pub/admin/user/organization/addOrgAdmin.do

**b.接口说明:**

**增加机构\部门管理员。**

相关信息：

1. 添加的是后台管理员
2. 同步增加、修改管理员基础表和机构人员表相关字段
3. 部门添加管理员，机构人员表需添加机构人员信息
4. 同步更新机构人员表中权限字段
5. 修改管理员信息不能修改人员所属机构/部门，页面做隐藏
6. 系统设置-网路防火墙：提交内容屏蔽敏感词汇
7. 系统设置-用户设置-统一设置：允许提交ip、禁止提交ip

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| addOrgAdmin | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| zjxxadmin | Zjxxadmin | | 管理员对象 | | |
| orgid | String | | 机构id | | |

**d.返回值示例与说明**{

{

"result": { //结果

"msg": "添加成功！", //信息

"flag": "1" //标识

}

}

**e.功能实现说明**

1. 根据提交ip和提交内容调用禁止和允许ip提交函数、屏蔽敏感词函数。
2. 根据对象参数people判断该对象中的id字段是否为空：
3. 若id为空，生成一个UUID赋值给该对象的id字段：

判断帐号是否存在，存在返回“帐号已存在”，帐号、密码不能为空，任意为空返回错误

1). 查询orgid是否是部门id，根据orgid查询zjxx\_orgnization （机构｛部门｝基础表），获取对应信息orgnizationinfor，判断orgnizationinfor数据中pid字段是否为空。

若不为空为部门id，获取orgnizationinfor数据中的pids的值

1. pids值的第一个id就是机构id（可能会出现脏数据）

2、select \* from zjxx\_orgnization where id in “‘+pids+’”(需要再查一遍sql，查不到返回错误)

根据方法1获取机构id，然后将人员信息插入zjxx\_manager（管理员表）。机构id 添加的是机构人员信息，其中机构人员信息中的所属部门id集合就是orgnizationinfor数据中的pids（将,替换成| 例如：A|B|C），理由：所属部门id集合orgids 里面就包含了所有父级部门，父级部门查询所属人员的时候只要like一下这个字段就行）。

若为空，就是机构id。然后将人员信息插入zjxx\_manager（管理员表），zjxx\_adminorg （机构｛部门｝员工表-后台）需添加机构人员信息，其中所属部门id集合为空。

3) 调用人员权限函数，同步更新机构管理员表中 机构管理员信息的权限字段。

② 若id不为空:

1) 根据参数id查找管理员表中id字段值为参数id值的数据，根据对应的 参数修改信息。同步修改机构管理员表-后台相关字段 -参考1.1.8接口

* + 1. 管理员匹配员工接口

**a.接口url:**

/pub/admin/user/organization/adminMatchStaff.do

**b.接口说明:**

该接口用于把员工升级为管理员

相关信息：

1. Id同步，以员工id为准

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| adminMatchStaff | | Result | | Json |
| 参数说明： | |  | | |
| 参数名称 | 参数类型 | | 备注 | |
| pid | String | | 员工id | |
| orgid | String | | 机构id | |
| adminid | String | | 管理员id | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "成功" //返回的信息；

}

}

**e.功能实现说明**

根据提交ip调用禁止和允许ip提交函数。

1. 根据管理员adminid查zjxx\_manager（管理员表）和zjxx\_adminorg （机构｛部门｝员工表-后台），将查到数据的id修改成人员pid。
   * 1. 员工匹配管理员接口

**a.接口url:**

/pub/admin/user/organization/staffMatchAdmin.do

**b.接口说明:**

该接口用于把员工升级为管理员

相关信息：

1. 将人员相关信息带过来，id、帐号、密码
2. 同步刷新该管理员权限信息（相当于新建一个带id的管理员）

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| staffMatchAdmin | | Result | | Json |
| 参数说明： | |  | | |
| 参数名称 | 参数类型 | | 备注 | |
| Id | String | | 员工id | |
| orgid | String | | 机构id | |
| rid | String | | 管理员角色id | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "成功" //返回的信息；

}

}

**e.功能实现说明**

1.根据提交ip调用禁止和允许ip提交函数。

1. 根据人员id查zjxx\_manager（管理员表），如果已经升级过，就返回已经升级过。
2. 根据人员id查出人员表一条数据
3. 创建管理员对象，根据查出的数据给管理员对象赋值。
4. 参照增加管理员接口逻辑。
5. 创建Result返回对象，并赋值参数，返回对象。
   * 1. 删除管理员接口

**a.接口url:**

/pub/admin/user/organization/deleteAdmin.do

**b.接口说明:**

该接口用于删除管理员

相关信息：

1. 伪删除

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteAdmin | | Result | | Json |
| 参数说明： | |  | | |
| 参数名称 | 参数类型 | | 备注 | |
| id | String | | 管理员id | |
| orgid | String | | 机构或部门id | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "成功" //返回的信息；

}

}

**e.功能实现说明**

1. 根据提交ip调用禁止和允许ip提交函数。

查询orgid是否是部门id，根据orgid查询zjxx\_orgnization （机构｛部门｝基础表），获取对应信息orgnizationinfor，判断orgnizationinfor数据中pid字段是否为空。

1. 为空：机构id 根据机构orgid和管理员id查询机构管理员-后台表，获取orgids（管理员所属部门id集合）字段，遍历参orgids ，每次遍历得到一个管理员部门id,根据管理员部门id和管理员id删除 zjxx\_porg （机构｛部门｝员工角色表）、zjxx\_adminorg （机构｛部门｝员工表-后台）对应数据。然后根据机构orgid和人员id删除zjxx\_porg （机构｛部门｝员工角色表）、zjxx\_adminorg （机构｛部门｝员工表-后台）对应数据。最后根据管理员id查询管理员基础表删除该管理员数据，管理员基础表伪删除
2. 不为空：部门id 根据部门orgid和管理员id删除 zjxx\_porg （机构｛部门｝员工角色表）、zjxx\_adminorg （机构｛部门｝员工表-后台）对应数据。根据orgid查询部门所属机构id（通过部门id查找部门的机构id，请参考1.1.8接口），根据机构id和管理员id查询机构管理员-后台表，获取相应信息。将orgids、rids字段中带orgid的数据删除。

。

* + 1. 机构/部门关联人员接口

**a.接口url:**

pub/admin/user/organization/setlinkStaff.do

**b.接口说明:**

该接口用于给部门配置员工

**c.接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| setlinkStaff | Result | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| peopleids[] | String | 人员ID集合，逗号隔开 |
| orgid | String | 机构/部门ID |

**d.返回值示例与说明**

{

"flag": 1,

"msg": "配置成功",

"orgid": "",

"rid": "",

"token": "",

"ttime": "",

"userId": ""

}

**e.功能实现说明**

根据提交ip调用禁止和允许ip提交函数。

检测两个参数peopleids[]、orgid是否为空或是否长度为0，有一个为真则返回msg:来源错误。参数都不为空时，

判断orgid如果为机构id走机构增加人员接口逻辑（请参考1.1.8接口），如果orgid为部门id 遍历参数peopleids[]，每次遍历得到一个人员id,根据人员id和部门orgid，查zjxx\_frontorg（机构｛部门｝员工表-前端），数据是否已存在，orgids中是否包含部门orgid。

存在：处理下一个人员id数据。

不存在：根据人员id查询出人员信息，通过部门orgid找到该部门的机构id（通过部门id查找部门的机构id，请参考1.1.8接口）和pids，新增机构员工信息并将部门数据中的pids（将,替换成| 例如：A|B|C）加入到机构员工信息中orgids（员工所属部门id集合）。

* + 1. 查询所有人员列表接口

**a.接口url:**

/pub/admin/user/organization/findAllpeoplelist.do

**b.接口说明:**

该接口用于查找所有人员列表

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findAllpeoplelist | | Info | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| pageSize | int | | 每页显示信息数 | |
| nowPage | Int | | 当前页 | |
| searchkey |  | |  | |
| searchvalue |  | |  | |
| Type | Int | | 0查所有人员，1查没有机构配置过的人员 | |

**d.返回值示例与说明**

{

“infos”:

{

[ "id": "8128", //员工ID;

"title": "小明", //员工姓名;

"orgid": "子杰信息", //所属机构;

"account": "221231 ", //账号;

"password": "12", //密码;

"idNumber": "123456", //身份证号;

"rTime": "20", //创建时间;

"rorder": "1" //显示顺序;

]

}

}

**e.功能实现说明**

1. type=0时查询zjxx\_people （人员表）的所有数据。
2. type=1时，查询人员id在人员表但不在机构人员表的人员。
   1. 数据词典管理
      1. 查找数据词典列表接口

**a.接口url:**

/pub/admin/user/dataDic/findDatadicList.do

**b.接口说明:**

该接口用于根据参数条件查询数据词典列表

相关信息：

查询数据词典表

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findDatadicList | | Info | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| pageSize | int | | 每页显示信息数 | |
| nowpage | Int | | 当前页 | |
| dataname | String | | 搜索数据词典名称 | |
| pid | String | | 父级数据词典ID | |

**d.返回值示例与说明**

{

"page": { //分页信息;

"page": 2, //当前页;

"pages": 10, //总页数;

"lines": 100, //信息总条数;

"pageSize": 10, //每页显示信息数;

"pageStart": 1 //起始页;

},

"total": 8, //当前页信息条数;

"infos": [ //信息;

{

"id": "80", //数据词典ID;

"title": "数据词典", //数据词典名称;

"description ": "数据词典", //数据词典描述;

"pid": "", //父级数据词典ID;

"cid": "17db2a", //子级数据词典ID集合;

"order\_num": 0 //数据词典顺序;

},

{}

]

}

**e.功能实现说明**

1.根据提交ip调用禁止和允许ip提交函数。

1. 根据参数dataname,pid,查找数据词典列表：
2. 查找数据词典表中pid字段值为参数pid的数据;
3. 当dataname不为空时拼接查找条件，查找数据词典表中title字段值like参数dataname值的数据;
4. 根据参数nowpage，pageSize计算出查找条件，追加查找条件limit;
5. 如果nowpage，pageSize为空则设置默认页面大小
6. 返回查询列表，并根据返回的数据列表，分页实体类Page,返回信息类Info整合数据。
   * 1. 查找数据词典信息接口

**a.接口url:**

/pub/admin/user/dataDic/findDatadicById.do

**b.接口说明:**

该接口用于根据参数条件查询数据词典列表

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findDatadicById | | ZjxxDataDictionary | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| did | String | | 数据词典ID | |

**d.返回值示例与说明**

{infos：[ {

"id": "80", //菜单ID;

"title": "机构A菜单", //菜单名称;

"description ": "菜单", //菜单描述;

"isFinal": "1", //是否终极菜单;

"parentMenu": "", //父菜单ID;

"childMenu": "17db2a", //子菜单ID集合;

"pageUrl": "", //页面URL;

"menuOrder": 0 //菜单顺序;

"whitelist": 0 //白名单;

"blacklist": 0 //黑名单;

"seegroupid": 0 //可浏览用户;

}

]

}

**e.功能实现说明**

（1）根据参数did查询数据词典表中id字段值为参数did字段值的数据。

* + 1. 增加/修改数据词典接口

**a.接口url:**

/pub/admin/user/dataDic/addOrUpdateDataDic.do

**b.接口说明:**

该接口用于增加/修改数据词典

相关信息：

1. 同级数据词典title唯一
2. 修改数据词典时，同步更新相应机构数据词典数据（title字段），pid不能修改
3. 系统设置-网路防火墙：提交内容屏蔽敏感词汇
4. 系统设置-用户设置-统一设置：允许提交ip、禁止提交ip

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| addOrUpdateDataDic | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| dictionary | ZjxxDataDictionary (对象) | | 数据词典对象 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "增加数据词典成功" //返回的信息；

}

}

或

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "修改数据词典成功" //返回的信息；

}

}

**e.功能实现说明**

1. 根据提交ip和提交内容调用禁止和允许ip提交函数、屏蔽敏感词函数。
2. 取出dictionary对象中的id，pid、title字段值，查询数据词典表中pid、title等于对象中pid、title，且id不等的数据，若为空：(同级title 唯一性)
   1. ：判断id是否等于空，为空生成UUID赋值给对象中id字段,pids所有父级词典id+自己的id，将数据存入数据库。

②：不为空，将传入对象updata入库 linkruleid 字段还未涉及

（2）不为空 ：修改对应数据信息，pid不能修改（页面也要做对应限制）。

* + 1. 删除数据词典接口

**a.接口url:**

/pub/admin/user/dataDic/deleteDatadic.do

**b.接口说明:**

该接口用于删除数据词典

相关信息：

1. 暂时不管相关机构数据词典的问题
2. 暂不管子集数据词典的问题

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteDatadic | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| idStr | String[] | | 数据词典ID拼接的字符串数组 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "删除数据词典成功" //返回的信息；

}

}

**e.功能实现说明**

遍历传入的数据词典id集合，以id为条件删除数据词典表中的数据。

根据提交ip调用禁止和允许ip提交函数函数。

1. 根据参数idStr删除数据词典，遍历idStr数组，每次遍历得参数id，
2. 搜索数据词典表，删除pids中包含id的数据（删除子孙）；
3. 通过参数id删除查到的对应数据

~~① 根据id查找数据词典表中id字段值为参数id值的数据，根据该数据中的 linkruleid字段查找规则表中id字段值为linkruleid字段值的数据，再根据查 找规则表得到的数据的rule\_content字段值，得到转换的数据表以及参数id对应 的转换后的信息id，然后根据得到的转换的数据表名以及id删除该表中该数据.~~

~~② 判断该数据是否有父数据词典：~~

~~1） 若有父数据词典则修改父级数据词典的词典cid字段，然后删除该数据词 典的数据记录以及递归查找并删除子级数据词典;~~

~~2） 若无父数据词典直接删除该数据词典的数据记录以及递归查找并删除子级 数据词典;~~

~~3） 删除每个数据词典数据记录时都要删除机构数据词典表中did字段值为删 除的数据词典id值的数据;~~

* + 1. 修改数据词典顺序接口

**a.接口url:**

/pub/admin/user/dataDic/updateDataDicOrder.do

**b.接口说明:**

该接口用于修改数据词典顺序

相关信息：

1. 同级顺序

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| updateDataDicOrder | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| idStr | String[] | | 遍历页面数据词典ID拼接的字符串数组 | |
| orderStr | String[] | | 遍历页面数据词典顺序拼接的字符串数组 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "修改数据词典顺序成功" //返回的信息；

}

}

**e.功能实现说明**

（1）1.根据提交ip调用禁止和允许ip提交函数函数。

两个数组长度不一致或者为空直接返回错误信息。

根据idStr，orderStr修改数据词典顺序,遍历idStr，orderStr数组，每次遍历得到id,order,根据参数id，order修改数据词典表中id字段值为参数id值的数据的order\_num字段，修改其值为参数order的值。

* + 1. 导入数据词典

相关信息：

1. 参考增加数据词典接口
2. 导入的数据不带父数据词典信息，就是顶级数据词典
3. 导入的数据带父数据词典信息，若父数据词典不存在，提示报错
4. 系统设置-网路防火墙：提交内容屏蔽敏感词汇
5. 系统设置-用户设置-统一设置：允许提交ip、禁止提交ip

**a.接口url:**

/pub/admin/user/dataDic/importDatadic.do

**b.接口说明:**

该接口用于从excel中导入数据词典

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| importDatadic | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
|  |  | |  | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "导入数据词典成功" //返回的信息；

}

}

**e.功能实现说明**

根据提交ip和提交内容调用禁止和允许ip提交函数、屏蔽敏感词函数。

系统参数表相关字段，先在缓存中查找，没有-去数据库系统参数表查找，并判断是否需要加入缓存，是-添加相应缓存

（1）遍历excel表格，取出每行的值以及对应的键存入map，将map存入lsit。

（2）遍历list，判断pid是否为空？

① 若为空则将map转换为数据词典对象存入数据词典表。

②若不为空，则查询pid在数据词典中是否存在？

若存在，则就将map转换为数据词典对象存入数据词典表。

若不存在就返回插入失败，父数据词典不存在。

* + 1. 更新数据词典接口 暂时没有用

主要四更新根据pid更新pids，是吗？

* + 1. 导出数据词典到excel接口

**a.接口url:**

/pub/admin/user/dataDic/exportDatadic.do

**b.接口说明:**

该接口用于导出数据词典到excel

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| exportDatadic | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
|  |  | |  | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "导出数据词典成功" //返回的信息；

}

}

**e.功能实现说明**

(1)查询出数据词典表中的全部信息，然后依次写入excel表格中

* + 1. 导入机构数据词典到excel接口

相关信息：

1. 导入的数据不带父数据词典信息，就是顶级数据词典
2. 导入的数据带父数据词典信息，若父数据词典不存在，提示报错
3. 先判断数据词典表是否有该数据，没有就新增，新增是参考新增数据词典说明。然后配置机构数据词典数据
4. 系统设置-网路防火墙：提交内容屏蔽敏感词汇
5. 系统设置-用户设置-统一设置：允许提交ip、禁止提交ip
6. **a.接口url:**

/pub/admin/user/dataDic/importOrgDatadic.do

**b.接口说明:**

该接口用于从excel中导入数据词典

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| importOrgDatadic | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
|  |  | |  | |

1. **d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "导入数据词典成功" //返回的信息；

}

}

1. **e.功能实现说明**

根据提交ip和提交内容调用禁止和允许ip提交函数、屏蔽敏感词函数。

系统参数表相关字段，先在缓存中查找，没有-去数据库系统参数表查找，并判断是否需要加入缓存，是-添加相应缓存

1. 遍历excel表格，取出每行的值以及对应的键存入集合
2. 遍历集合，然后根据数据词典名查询数据词典表中是否存在该信息，若不存在，判断父数据词典是否存在，若不存在则则返回 数据词典不存在，父数据词典不存在。

若父数据词典存在，向数据词典表插入一条信息。

1. 判断父数据词典是否为空，若不为空则查询父数据词典id，记录父数据词典id。
2. 将数据存入数据词典表。

* + 1. 导出机构数据词典到excel接口

**a.接口url:**

**/pub/admin/user/dataDic/exportDatadic.do**

**b.接口说明:**

该接口用于导出机构数据词典到excel

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| exportOrgDatadic | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构id | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "导出数据词典成功" //返回的信息；

}

}

**e.功能实现说明**

根据参数orgid查询出机构数据词典表中的orgid字段相同信息，然后依次写入excel表格中

* + 1. 配置机构数据词典

**a.接口url:**

**/pub/admin/user/dataDic/**setOrgDataDic.do

**b.接口说明:**

该接口用于给机构配置数据词典用

相关信息：

1. 只配置本当前数据词典数据，其子孙数据不管。
2. 需要关联父级数据词典，同步配置该数据词典的所有父级数据词典。
3. 后台获取数据词典的pids字段配置所有的父级词典。
4. 要做判断该数据词典是否已经配置过。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| setOrgDataDic | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构ID | |
| didStr | String[] | | 数据词典ID拼接的字符串数组 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "配置机构数据词典成功" //返回的信息；

}

}

**e.功能实现说明**

（1）根据提交ip调用禁止和允许ip提交函数函数。

根据参数orgid，didStr为机构配置数据词典，遍历数组didStr，每次遍历得did1：

通过did1获取did1的pids，遍历数组pids，每次遍历得did2

根据参数orgid，did2,判断是否已存在，存在处理下一条，

不存在生成一个id（orgid + "|" + did）在机构数据词典表插入数据。

* + 1. 删除机构数据词典

**a.接口url:**

pub/admin/user/dataDic/deleteOrgDataDic.do

**b.接口说明:**

该接口用于删除机构配置的数据词典

相关信息：

1. 暂不管其子孙数据。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteOrgDataDic | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| ids | String[] | | 机构数据词典id数组 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "删除机构数据词典成功" //返回的信息；

}

}

**e.功能实现说明**

1. 根据提交ip调用禁止和允许ip提交函数函数。
2. 遍历didStr数组，每次遍历得did1

① 然后根据orgid，did1删除机构数据词典表中oid字段值为参数orgid值并且 did字段值为遍历得到的参数did1值的数据。

* + 1. 更新机构数据词典，这个是做什么？没有用了
    2. 查询机构已配置的数据词典列表

**a.接口url:**

/pub/admin/user/organization/ FindOrgDataDicList.do

**b.接口说明:**

该接口用于查询已配置的数据词典列表

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| FindOrgDataDicList | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构ID | |
| pid | String | | 父id | |
| pageSize | int | | 每页显示信息数 | |
| nowpage | Int | | 当前页 | |
| dataname | String | | 数据词典名 | |
| key | String | | 关键字 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "配置机构数据词典成功" //返回的信息；

}

}

**e.功能实现说明**

（1）判断机构id是否为空，若为空则调用获取平台机构id函数，若还是返回空直接返回获取信息失败。

（2）判断pageSize和nowPage是否为空，为空则设置默认值.

(3) 判断dataname是否为空，不为空则拼接 条件datanname sql

(4) 拼接pid sql。

（5）连接机构数据词典表和数据词典表，查出机构数据词典表中did等于数据词典表中的id的数据，再用条件机构id等于传入的机构id，拼接上dataname、pid、pageSize条件。

* + 1. 生成pid获取机构中数据词典Json接口

**a.接口url:**

/pub/admin/user/dataDic/dataDicToJson.do

**b.接口说明:**

该接口用于将数据词典生成Json

**c.接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| dataDicToJson | Info | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| id | String | 数据词典id |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //数据词典ID;

"title": "数据词典", //数据词典名称;

"description ": "数据词典", //数据词典描述;

"pid": "", //父级数据词典ID;

"cid": "17db2a", //子级数据词典ID集合;

"order\_num": 0 //数据词典顺序;

},

{

"id": "80", //数据词典ID;

"title": "数据词典", //数据词典名称;

"description ": "数据词典", //数据词典描述;

"pid": "", //父级数据词典ID;

"cid": "17db2a", //子级数据词典ID集合;

"order\_num": 0 //数据词典顺序;

},

{ }

]

}

**e.功能实现说明**

（1）根据参数id查找数据词典表中id字段值为参数id值的数据，以及其子孙数据，返回json对象数组

* 1. 角色管理
     1. 查找角色列表接口

**a.接口url:**

/pub/admin/user/role/findRoleList.do

**b.接口说明:**

该接口用于根据参数条件查找角色列表

相关信息：

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findRoleList | | Info | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| pageSize | int | | 每页显示信息数 | |
| nowpage | Int | | 当前页 | |
| rolename | String | | 输入的角色名称 | |
| pid | String | | 父角色ID | |
| isadmin | Int | | 0.否；1.是； | |
| orgid | String | | 机构ID | |
| peopleid | String | | 人员ID | |

**d.返回值示例与说明**

{

"page": { //分页信息;

"page": 2, //当前页;

"pages": 10, //总页数;

"lines": 100, //信息总条数;

"pageSize": 10, //每页显示信息数;

"pageStart": 1 //起始页;

},

"total": 8, //当前页信息条数;

"infos": [ //信息;

{

"id": "wedff887sds", //角色ID;

"title": "角色11", //角色名称;

"pid": "123", //父角色ID;

"rorder": 0, //角色顺序;

"description": "123", //角色描述;

"isdelete": "123", //是否删除;

"cid": "123", //子角色ID集合;

"orgid": "123" //

},

{}

]

}

**e.功能实现说明**

1. 根据参数rolename，pid，查询角色：

① 当rolename不为空时拼接查找条件，查找角色表中title字段值like参数 rolename值的数据;

④ 当pid不为空时拼接查找条件，查找角色表中pid字段值为参数pid值的数据;

⑤ 当pid为空时拼接查找条件，查找角色表中pid字段值为空的数据;

⑥ 拼接查找条件，查找角色表中isdelete值为“0”的数据;

⑦ 根据参数nowpage，pageSize计算出查找条件，追加查找条件limit;

⑧ 根据拼接的sql语句执行查询操作，返回查询列表，并根据返回的数据列表， 分页实体类Page,返回信息类Info整合数据。

* + 1. 查找角色信息接口

**a.接口url:**

/pub/admin/user/role/findRole.do

**b.接口说明:**

该接口用于查找角色信息

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findRole | | Info | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| rid | String | | 角色ID | |

**d.返回值示例与说明**

{

“infos”:

{

"id": "wedff887sds", //角色ID;

"title": "角色11", //角色名称;

"pid": "123", //父角色ID;

"rorder": 0, //角色顺序;

"description": "123", //角色描述;

"isdelete": "123", //是否删除;

"cid": "123", //子角色ID集合;

"orgid": "123" //机构ID

}

}

**e.功能实现说明**

（1）根据参数rid查找角色表中id字段值为rid值的数据。

* + 1. 删除角色接口

**a.接口url:**

/pub/admin/user/role/deleteRole.do

**b.接口说明:**

该接口用于删除角色

相关信息：

1. 伪删除。
2. 暂不管机构角色

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteRole | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| idStr | String | | 页面勾选的角色ID拼接的字符串，多个逗号隔开，例如：id1,id2,id3, | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "删除角色成功" //返回的信息；

}

}

**e.功能实现说明**

1. 根据提交ip调用禁止和允许ip提交函数函数。
2. 删除角色，遍历idStr数组，每次遍历得参数id，
3. 根据id查找角色表中id字段值为参数id值的数据，然后修改该角色的isdelete字段 修改为“1”;
   * 1. 修改角色顺序接口

**a.接口url:**

/pub/admin/user/role/updateRoleOrder.do

**b.接口说明:**

该接口用于修改角色顺序

相关信息：

1. 同级顺序。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| updateRoleOrder | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| idStr | String[] | | 遍历页面角色ID拼接的字符串数组 | |
| orderStr | String[] | | 遍历页面角色顺序拼接的字符串数组 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "修改角色顺序成功" //返回的信息；

}

}

**e.功能实现说明**

（1）根据提交ip调用禁止和允许ip提交函数函数。

两个数组长度不一致或者为空直接返回错误信息。

根据参数idStr，orderStr修改角色顺序：遍历idStr，orderStr，每次遍历得到id,order,根据参数id，order修改角色表中id字段值为参数id值的数据的rorder字段，修改其值为参数order的值。

* + 1. **增加/修改角色接口**

**a.接口url:**

/pub/admin/user/role/addRole.do

**b.接口说明:**

该接口用于增加/修改角色

相关信息：

同级角色title唯一

修改，同步更新机构角色表title字段

系统设置-网路防火墙：提交内容屏蔽敏感词汇

系统设置-用户设置-统一设置：允许提交ip、禁止提交ip

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| addRole | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| role | ZjxxRole （对象） | | 角色表类对象 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "增加角色成功" //返回的信息；

}

}

或

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "修改角色成功" //返回的信息；

}

}

**e.功能实现说明**

根据提交ip和提交内容调用禁止和允许ip提交函数、屏蔽敏感词函数。

（1）根据对象参数role判断对象中id字段值是否为空：

①id值为空时，生成UUID赋值给对象中id字段，然后再判断该对象中pid字段 值是否为空：

1）若为空，根据对象中title字段和值且pid为空、isdelete==0为条件，查询角色表中是否有数据据，没有则直接将该对象插入角色表中;若有返回错误信息‘该角色名称已存在’。对象中pid不为空：在角色表中查出id为对象中pid字段值的那条数据，获取这条数据中的pids然后加上对象的id的值，然后把生成的pids赋值给数据对象的pids。对象中pid为空：pids就是对象的id。

② id值不为空时：

1. 根据参数id查找角色表中id字段值为参数id值的数据，更新角色表中该数据;
2. 根据参数id查找机构角色（前端、后台）表中rid字段值为参数id值的数据，更新title字段;
   * 1. **导入角色接口**

相关信息：

1. 同级角色title唯一
2. 导入的角色不带父角色词典信息，就是顶级角色
3. 导入的数据带父角色信息，若父角色不存在，提示报错（或者若父角色不在，就创建一个顶级角色作为父角色）
4. 若父角色有多条，就在每个父角色下都新增一条对应子角色
5. 系统设置-网路防火墙：提交内容屏蔽敏感词汇
6. 系统设置-用户设置-统一设置：允许提交ip、禁止提交ip

**a.接口url:**

/pub/admin/user/role/roleImport.do

**b.接口说明:**

该接口用于批量导入角色。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| roleImport | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| File | file | | Excel文件 | |

**d.返回值示例与说明**

{

"result": { //结果;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "导入角色成功" //返回的信息；

}

}

**e.功能实现说明**

根据提交ip和提交内容调用禁止和允许ip提交函数、屏蔽敏感词函数。

系统参数表相关字段，先在缓存中查找，没有-去数据库系统参数表查找，并判断是否需要加入缓存，是-添加相应缓存

(1)遍历读取excel表格中的数据。

(2)取出每一行的数据中的父角色字段和角色名称，以父角色名称字段查询父角色id判断父角色id是否为空

①若不为空（且是多条则下面步骤执行多次）

以父角色id且角色名称为条件查询，角色表中是否存在满足条件的数据，

若存在则返回title具体名称存在重复

若不存在则则将当前条数据父角色名称替换为title存入表中

②若为空则插入一条角色信息。记录id

将当前条数据父角色名称替换为title存入表中

* + 1. 导出**角色接口**

**a.接口url:**

/pub/admin/user/role/exportRole.do

**b.接口说明:**

该接口用于批量导出角色。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| exportRole | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| role | ZjxxRole （对象） | | 角色表类对象 | |

**d.返回值示例与说明**

{

"result": { //结果;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "导出角色成功" //返回的信息；

}

}

**e.功能实现说明**

(1)查询出角色表中所有数据存入集合。

(2) 新建一个excel文件，写入数据

(3) 返回文件地址

* + 1. **导入机构角色接口**

相关信息：

1. 导入的数据不带父角色信息，就是顶级角色
2. 导入的数据带父角色信息，若父角色不存在，提示报错不存在（或者若父角色不在，就创建一个顶级角色作为父角色）
3. 若父角色有多条，就在每个父角色下都新增一条对应子角色
4. 先判断角色表是否有该数据，没有就新增，新增是参考新增角色说明。然后配置机构角色数据
5. 系统设置-网路防火墙：提交内容屏蔽敏感词汇
6. 系统设置-用户设置-统一设置：允许提交ip、禁止提交ip

**a.接口url:**

/pub/admin/user/role/OrgroleImport.do

**b.接口说明:**

该接口用于批量导入机构角色。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| OrgroleImport | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| Orgid | String | | 机构id | |
|  | File | | Excel 文件 | |

**d.返回值示例与说明**

{

"result": { //结果;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "导入角色成功" //返回的信息；

}

}

**e.功能实现说明**

根据提交ip和提交内容调用禁止和允许ip提交函数、屏蔽敏感词函数。

系统参数表相关字段，先在缓存中查找，没有-去数据库系统参数表查找，并判断是否需要加入缓存，是-添加相应缓存

(1)遍历读取excel表格中的数据。

(2)取出当前条数据角色名称，查询角色表中是否存在该角色

(3)若不存在。则拿出父角色名称，判断是否为空，若不为空则查询出父角色名称对应id，若不存在则返回角色不存在，父角色不存在。

若存在，先将数据存入角色表.

(5)取出父角色名称，判断是否为空，为空则将数据直接存入机构角色表，不为空查询父角色id，(当有多个父角色id，就是存入条数据)将数据存入机构角色表。

* + 1. **导出机构角色接口**

**a.接口url:**

/pub/admin/user/role/OrgexportRole.do

**b.接口说明:**

该接口用于批量导出机构角色。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| OrgexportRole | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| Orgid | String | | 机构id | |

**d.返回值示例与说明**

{

"result": { //结果;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "导出角色成功" //返回的信息；

}

}

(1)查询出机构角色表中orgid为传入参数的所有数据存入集合。

(2) 新建一个excel文件，写入数据

(3) 返回文件地址

* + 1. 配置机构\部门前端角色接口

**a.接口url:**

/pub/admin/user/organization/setOrgStaffRole.do

**b.接口说明:**

该接口用于给机构配置前端角色

相关信息：

1. 只配置本当前角色，其子孙角色不管。
2. 需要关联父级角色，同步配置该角色的所有父级角色，后台做
3. 将角色的pid、title字段带过来。
4. 要做判断该角色是否已经配置过。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| setOrgStaffRole | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构ID | |
| ridStr | String [] | | 遍历页面勾选的角色id拼接的字符串数组 | |
| type | int | | 0=前端角色，1=后台角色 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "配置机构角色成功" //返回的信息；

}

}

**e.功能实现说明**

1. 根据提交ip调用禁止和允许ip提交函数函数。
2. 根据参数orgid ，ridStr 配置机构角色：
   1. 参数orgid和每次遍历ridStr数组得到的rid，根据每次遍历ridStr数组得到的rid在角色表中查找获得角色的数据（title、pid、pids），拆分pids获取拆分的rid在机构前端角色表中判断数据是否已经存在，若不存在：，根据参数orgid在机构前端角色表中增加数据，注意ridpid（父角色ID）。
      1. 配置机构\部门后台角色接口

**a.接口url:**

/pub/admin/user/organization/setOrgAdminRole.do

**b.接口说明:**

该接口用于给机构配置后台角色

相关信息：

1. 只配置本当前角色，其子孙角色不管。
2. 需要关联父级角色，同步配置该角色的所有父级角色。后台做
3. 将角色的pid、title字段带过来。
4. 要做判断该角色是否已经配置过。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| setOrgAdminRole | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构ID | |
| ridStr | String [] | | 遍历页面勾选的角色id拼接的字符串数组 | |
| type | int | | 0=前端角色，1=后台角色 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "配置机构角色成功" //返回的信息；

}

}

**e.功能实现说明**

(1) 根据提交ip调用禁止和允许ip提交函数函数。

(2)根据参数orgid ，ridStr 配置机构角色：

* 1. 参数orgid和每次遍历ridStr数组得到的rid，根据每次遍历ridStr数组得到的rid在角色表中查找获得角色的数据（title、pid、pids），拆分pids获取拆分的rid在机构后台角色表中判断数据是否已经存在，若不存在：根据参数orgid在机构后台角色表中增加数据。
     1. 删除机构\部门前端角色接口

**a.接口url:**

/pub/admin/user/organization/deleteOrgStaffRole.do

**b.接口说明:**

该接口用于删除机构配置的角色用

暂不考虑人员以配置的角色—后期：机构\部门删除角色，则本机构\部门人员关联了这个角色，要在机构员工表和机构员工角色表删除关联关系。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteOrgStaffRole | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构ID | |
| ridStr | String [] | | 勾选的角色id的字符串数组 | |
| type | int | | 0=前端角色，1=后台角色 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "删除机构角色成功" //返回的信息；

}

}

**e.功能实现说明**

1. 根据提交ip调用禁止和允许ip提交函数函数。
2. 遍历ridStr中所有的角色rid;根据遍历出来的rid和参数orgid删除机构前端角色表中对应的数据;
   * 1. 删除机构\部门后台角色接口

**a.接口url:**

/pub/admin/user/organization/deleteOrgAdminRole.do

**b.接口说明:**

该接口用于删除机构配置的角色用

暂不考虑人员以配置的角色—后期：机构\部门删除角色，则本机构\部门人员关联了这个角色，要在要在机构员工表和机构员工角色表删除关联关系。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteOrgAdminRole | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构ID | |
| ridStr | String [] | | 勾选的角色id的字符串数组 | |
| type | int | | 0=前端角色，1=后台角色 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "删除机构角色成功" //返回的信息；

}

}

**e.功能实现说明**

1. 根据提交ip调用禁止和允许ip提交函数。
2. 遍历ridStr中所有的角色rid;根据遍历出来的rid和参数orgid删除机构后台角色表中对应的数据;
   * 1. 查找机构或部门已配置的前端角色列表

**a.接口url:**

/pub/admin/user/role/findOrgStaffRoleLIst.do

**b.接口说明:**

该接口用于查询机构\部门前端角色列表

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findOrgStaffRoleLIst | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| ridpid | String | | 角色父ID | |
| orgid | String | | 机构id | |
| title | String | | 角色名称（条件查询） | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "配置员工角色成功" //返回的信息；

}

}

**e.功能实现说明**

1. 根据参数ridpid,，orgid查询zjxx\_frontrorg机构前端角色表。
2. 判断是否是机构，如果不是找到其机构id。根据机构id去查。部门默认拥有机构的角色。
3. ridpid为空查询的是机构配置的顶级角色，ridpid不为空，查询ridpid值为ridpid的数据；一句sql where ridpid=ridpid
   * 1. 查找机构或部门已配置的后台角色列表

**a.接口url:**

/pub/admin/user/role/findOrgAdminRoleLIst.do

**b.接口说明:**

该接口用于查询机构\部门后台角色列表

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findOrgAdminRoleLIst | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| ridpid | String | | 角色父ID | |
| orgid | String | | 机构id | |
| title | String | | 角色名称（条件查询） | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "配置员工角色成功" //返回的信息；

}

}

**e.功能实现说明**

1. 根据参数ridpid,，orgid查询zjxx\_adminrorg机构前端角色表。
2. 判断是否是机构，如果不是找到其机构id。根据机构id去查。部门默认拥有机构的角色。
3. ridpid为空查询的是机构配置的顶级角色，ridpid不为空，查询ridpid值为ridpid的数据；一句sql where ridpid=ridpid
   * 1. 配置员工前端角色接口

**a.接口url:**

/pub/admin/user/organization/setStaffRole.do

**b.接口说明:**

该接口用于给员工配置角色用

相关信息：

调用函数同步更新人员权限信息

只配置当前角色，父级角色不管

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| setStaffRole | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| pid | String | | 员工ID | |
| rid | String | | 角色id | |
| orgid | String | | 机构id | |
| type | int | | 0=前端角色，1=后台角色 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "配置员工角色成功" //返回的信息；

}

}

**e.功能实现说明**

根据提交ip调用禁止和允许ip提交函数函数。

根据参数pid,rid，orgid查询机构员工角色表数据是否存在，存在返回已配置，若不存在新增数据 isadmin=0。判断orgid是否为机构id，若不是：找到该部门的机构id，查询机构人员表判断机构人员信息中rids（角色id集合）中是否包含rid|orgid，不包含就添加，包含跳过，~~再判断机构角色表是否存在 这条部门角色数据，不存在就添加~~。然后调用人员权限函数根据pid，orgid更新人员权限。若是：判断机构人员信息中rids（角色id集合）中是否包含rid|orgid，不包含就添加，包含跳过。然后调用人员权限函数根据pid，orgid更新人员权限

* + 1. 配置管理员后台角色接口

**a.接口url:**

/pub/admin/user/organization/setAdminRole.do

**b.接口说明:**

该接口用于给管理员配置角色用

相关信息：

调用函数同步更新人员权限信息

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| setAdminRole | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| pid | String | | 员工ID | |
| rid | String | | 角色id | |
| orgid | String | | 机构id | |
| type | int | | 0=前端角色，1=后台角色 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "配置员工角色成功" //返回的信息；

}

}

**e.功能实现说明**

根据提交ip调用禁止和允许ip提交函数函数。

根据参数pid,rid，orgid查询机构管人员角色表数据是否存在，存在返回已配置，若不存在新增数据 isadmin=1。判断orgid是否为机构id，若不是：找到该部门的机构id，查询机构管理员表判断机构管理员信息中rids（角色id集合）中是否包含rid|orgid，不包含就添加，包含跳过，~~再判断机构角色表是否存在 这条部门角色数据，不存在就添加~~。。然后调用人员权限函数根据pid，orgid更新人员权限。若是：判断机构管理员信息中rids（角色id集合）中是否包含rid|orgid，不包含就添加，包含跳过。然后调用人员权限函数根据pid，orgid更新人员权限

* + 1. 删除员工前端角色接口

**a.接口url:**

/pub/admin/user/organization/deleteStaffRole.do

**b.接口说明:**

该接口用于删除员工配置的角色

相关信息：

同步更新人员权限信息

**c.接口定义:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteStaffRole | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| rid | String | | 角色id | |
| pid | String | | 员工ID | |
| orgid | String | | 机构id | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "删除员工角色成功" //返回的信息；

}

}

**e.功能实现说明**

1. 根据提交ip调用禁止和允许ip提交函数函数。
2. 根据参数pid，orgid，rid查询机构人员角色表数据，删除对应数据
3. 根据参数orgid判断orgid是否为机构id
4. 是：根据参数pid，orgid查询机构人员表数据，修改角色表ID集合（rids）字段，删除字段中orgid|rid。根据pid，orgid调用人员权限函数更新人员权限。
5. 否：根据orgid查询到该部门的机构id，根据参数pid,，查询机构人员表数据，删除字段中orgid|rid。根据pid，orgid调用人员权限函数更新人员权限。
   * 1. 删除管理员后台角色接口

**a.接口url:**

/pub/admin/user/organization/deleteAdminRole.do

**b.接口说明:**

该接口用于删除员工配置的角色

相关信息：

同步更新人员权限信息

**c.接口定义:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteAdminRole | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| rid | String | | 角色id | |
| pid | String | | 员工ID | |
| orgid | String | | 机构id | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "删除员工角色成功" //返回的信息；

}

}

**e.功能实现说明**

(1) 根据提交ip调用禁止和允许ip提交函数函数。

(2)根据参数pid，orgid，rid查询机构人员角色表数据，删除对应数据

(3)根据参数orgid判断orgid是否为机构id

1. 是：根据参数pid，orgid查询机构管理员表数据，修改角色表ID集合（rids）字段，删除字段中orgid|rid。根据pid，orgid调用管理员权限函数更新管理员权限。
2. 否：根据orgid查询到该部门的机构id，根据参数pid,，机构i查询机构管理员表数据，，删除字段中orgid|rid。根据pid，orgid调用管理员权限函数更新管理员权限。
   * 1. 查看机构/部门前端人员的角色列表

**a.接口url:**

/pub/admin/user/organization/findRoleByStaff.do

**b.接口说明:**

查看该员工在机构/部门配置的角色列表

相关信息：

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findRoleByStaff | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| pid | String | | 角色id | |
| orgid | String | | 机构id | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "配置员工角色成功" //返回的信息；

}，

"infos": [ //信息;

{

"id": "80", //机构id;

"title": "部门名称", //部门名称;

},

{

"id": "80", //机构id;

"title": "部门名称", //部门名称;

}

}

**e.功能实现说明**

1. 根据参数orgid，pid，查询zjxx\_porg （机构｛部门｝员工角色表）获取数据，再根据角色id查询角色表获取角色名称。
   * 1. 查看机构/部门管理员的角色列表

**a.接口url:**

/pub/admin/user/organization/findRoleByAdmin.do

**b.接口说明:**

查看该管理员在机构/部门配置的角色列表

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findRoleByAdmin | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| pid | String | | 角色id | |
| orgid | String | | 机构id | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "配置员工角色成功" //返回的信息；

}，

"infos": [ //信息;

{

"id": "80", //机构id;

"title": "部门名称", //部门名称;

},

{

"id": "80", //机构id;

"title": "部门名称", //部门名称;

}

}

**e.功能实现说明**

根据参数orgid，pid，isadmin=1查查询zjxx\_porg （机构｛部门｝员工角色表）获取数据，再根据角色id查询角色表获取角色名称。

* + 1. 查看机构/部门前端角色下人员列表

**a.接口url:**

/pub/admin/user/organization/findStaffByrid.do

**b.接口说明:**

查看该角色在机构/部门配置的员工列表

相关信息：

1. 如果是机构，就查询该机构及其子孙部门下的所有人员
2. 如果是部门，就查询该部门~~及其子孙部门下的~~人员（后期在查子孙部门）

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findStaffByrid | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| rid | String | | 角色id | |
| orgid | String | | 机构id | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "配置员工角色成功" //返回的信息；

}，

"infos": [ //信息;

{

"id": "80", //机构id;

"title": "部门名称", //部门名称;

},

{

"id": "80", //机构id;

"title": "部门名称", //部门名称;

}

}

**e.功能实现说明**

1. 根据参数orgid判断orgid是否为机构id
2. 是：根据参数orgid，角色合集（rids）字段like rid，查询机构人员表数据，
3. 否：根据orgid查询到该部门的机构id，根据机构id，角色合集（rids）字段like orgid|rid，查询机构人员表数据
   * 1. 查看机构/部门后台角色下管理员列表

**a.接口url:**

/pub/admin/user/organization/ findAdminByrid.do

**b.接口说明:**

查看该角色在机构/部门配置的管理员列表

相关信息：

1. 如果是机构，就查询该机构及其子孙部门下的所有管理员
2. 如果是部门，就查询该部门~~及其子孙部门下的~~管理员（后期在查子孙部门）

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findAdminByrid | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| rid | String | | 角色id | |
| orgid | String | | 机构id | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "0", //标志，0为成功，1为操作失败

"msg": "配置员工角色成功" //返回的信息；

}，

"infos": [ //信息;

{

"id": "80", //机构id;

"title": "部门名称", //部门名称;

},

{

"id": "80", //机构id;

"title": "部门名称", //部门名称;

}

}

**e.功能实现说明**

1. 根据参数orgid判断orgid是否为机构id
2. 是：根据参数orgid，角色合集（rids）字段like rid，查询机构管理员表数据，
3. 否：根据orgid查询到该部门的机构id，根据机构id，角色合集（rids）字段like orgid|rid，查询机构管理员表数据
   1. 菜单管理
      1. 新增/修改菜单

相关信息：

先增加菜单表数据，然后配置机构表的菜单字段

暂不刷新机构人员权限信息，通过页面按钮点击后刷新

系统设置-网路防火墙：提交内容屏蔽敏感词汇

系统设置-用户设置-统一设置：允许提交ip、禁止提交ip

**a.接口url:**

/pub/admin/user/menu/ operateMenu.do

**b.接口说明:**

该接口用于新增或修改菜单信息

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| operateMenu | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| isadmin | String | | 0-前端,1-后端 | |
| menuid | String | | 菜单id(为空时调用新增,不为空则修改) | |
| title | String | | 菜单名称 | |
| pid | String | | 父菜单id | |
| pic | String | | 图片 | |
| isFinal | String | | 是否是终极菜单.0-不是,1-是 | |
| bewrite | String | | 描述 | |
| orgid | String | | 机构id | |
| url | String | | 终极菜单对应url | |

**d.返回值示例与说明**

{

“infos”:

{

}，

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 根据提交ip和提交内容调用禁止和允许ip提交函数、屏蔽敏感词函数。
2. 判断是否传递了菜单id,如果没传就生成一个uuid，
3. 把前台传递的参数整合成map,调用菜单的server里的新增或修改的方法，

新增：pid不为空：若不为空，则在菜单表中查出id为对象中pid字段值的那条数据，获取这条数据中的pids然后加上对象的uuid的值，然后把生成的pids赋值给数据对象的pids。

1. 当操作成功时,如果调用的是新增菜单,就把菜单的uuid存入zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）的adminmenuids(后台菜单功能id集合)或menuids(前台菜单功能id集合)字段中,where orgid = 传递的orgid
2. 返回结果集,告知操作成功或失败
   * 1. **查找平台后台基本菜单列表接口**

**a.接口url:**

/pub/admin/user/menu/findBasicMenuList.do

**b.接口说明:**

该接口用于查找菜单列表

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findMenuList | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构id-999999999 | |

**d.返回值示例与说明**

{

“infos”:

{

}，

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 机构id默认传999999999，获取在zjxx\_menu （菜单表）中orgid=999999999数据，按层级全部返回。
2. 返回符合条件的菜单信息列表
   * 1. **查找菜单列表接口**

**a.接口url:**

/pub/admin/user/menu/findMenuList.do

**b.接口说明:**

该接口用于查找菜单列表

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findMenuList | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| pid | String | | 父菜单id | |
| orgid | String | | 机构id | |
| isadmin | int | | 0：前台，1：后台 | |

**d.返回值示例与说明**

{

“infos”:

{

}，

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 判断orgid、isadmin任意一个为空，返回错误
2. 判断pid是否为空，
3. 为空：通过传递的pid、orgid、isadmin,在zjxx\_menu （菜单表）中查找对应数据与zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）中id= orgid的数据中adminmenuids（后台菜单）、menuids（前端菜单）做比较，返回id在adminmenuids（后台菜单）或menuids（前端菜单）中的数据。
4. 不为空：通过传递的pid、orgid、isadmin,在zjxx\_menu （菜单表）中查找对应数据与zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）中id= orgid的数据中adminmenuids（后台菜单）、menuids（前端菜单）做比较，返回id在adminmenuids（后台菜单）或menuids（前端菜单）中的数据。
5. 返回符合条件的菜单信息列表
   * 1. **查找菜单信息接口**

**a.接口url:**

/pub/admin/user/menu/findMenuInfo.do

**b.接口说明:**

接该接口用于查找菜单信息

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findMenuInfo | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| menuid | String | | 菜单ID | |

**d.返回值示例与说明**

{

“infos”:

{

"id": "80", //菜单ID;

"title": "机构A菜单", //菜单名称;

"description ": "菜单", //菜单描述;

"isFinal": "1", //是否终极菜单;

"pid": "", //父菜单ID;

"child": "17db2a", //子菜单ID集合;

"shortTitle": "17db2a", //菜单名称简称;

"englishTitle": "17db2a", //菜单名称英文;

"orgid": "kjflkfj", //机构id;

"type": "1", //类型;

}，

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 根据菜单id去菜单表(zjxx\_menu)找菜单数据。
2. 把找到的数据放到infor对象的info字段中。再给infor对象设置结果集result。
   * 1. 修改菜单顺序

**a.接口url:**

/pub/admin/user/menu/ reviseMenuOrder.do

**b.接口说明:**

接该接口用于修改菜单顺序

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| reviseMenuOrder | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| menuids | String[] | | 菜单ID | |
| rorders | int[] | | 菜单的顺序值,越小越靠前 | |

**d.返回值示例与说明**

{

“infos”:

{

"id": "80", //菜单ID;

"title": "机构A菜单", //菜单名称;

"description ": "菜单", //菜单描述;

"isFinal": "1", //是否终极菜单;

"pid": "", //父菜单ID;

"child": "17db2a", //子菜单ID集合;

"shortTitle": "17db2a", //菜单名称简称;

"englishTitle": "17db2a", //菜单名称英文;

"orgid": "kjflkfj", //机构id;

"type": "1", //类型;

}，

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 根据提交ip调用禁止和允许ip提交函数函数。
2. 两个数组长度不一致或者为空直接返回错误信息。
3. 根据遍历接收到的菜单id,在菜单表中更新rorder(顺序)字段。
4. 返回操作成功或失败
   * 1. **删除菜单接口**

**a.接口url:**

/pub/admin/user/menu/deleteMenu.do

**b.接口说明:**

该接口用于删除菜单功能

相关信息：

暂不刷新机构人员权限信息，通过页面按钮点击后刷新

菜单依附于机构，现在没有单独的菜单管理。只能通过机构去管理菜单。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteMenu | | Result | | Json |
| 参数说明： | |
| 参数名称 | 参数类型 | | 备注 | |
| ids | String[] | | 菜单id | |
| orgid | String | | 机构id | |
| isadmin | int | | 0-前端,1-后端 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "成功" //返回的信息；

}

}

**e.功能实现说明**

1. 根据提交ip调用禁止和允许ip提交函数函数。
2. 把菜单id去菜单表(zjxx\_menu)中移除
3. 把菜单id去zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）中的前端或后端菜单集合中移除,where orgid = 接收的orgid
   * 1. ~~新增/修改功能 （删除）~~

相关信息：

暂不刷新机构人员权限信息，通过页面按钮点击后刷新

系统设置-网路防火墙：提交内容屏蔽敏感词汇

系统设置-用户设置-统一设置：允许提交ip、禁止提交ip

**a.接口url:**

/pub/admin/user/menu/operationMenuFunction.do

**b.接口说明:**

该接口用于新增/修改菜单功能

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| operationMenuFunction | | Result | | Json |
| 参数说明： | |
| 参数名称 | 参数类型 | | 备注 | |
| id | String | | 功能id | |
| title | String | | 功能名称 | |
| menuid | String | | 菜单id | |
| pic | String | | 图片 | |
| word | String | | 文字 | |
| URL | String | | url地址 | |
| page | String | | 页面 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "成功" //返回的信息；

}

}

**e.功能实现说明**

1. 根据提交ip和提交内容调用禁止和允许ip提交函数、屏蔽敏感词函数。
2. 判断是否传递了功能id,如果没传就生成一个uuid
3. 把前台传递的参数整合成map,调用功能的server里的新增或修改的方法
4. 当操作成功时,如果调用的是新增功能,就把功能的uuid存入zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）的adminmenuids(后台菜单功能id集合)或menuids(前台菜单功能id集合)字段中,where orgid = 传递的orgid
5. 返回结果集,告知操作成功或失败
   * 1. **~~查找功能列表接口（删除）~~**

**a.接口url:**

/pub/admin/user/menu/findFunctionList.do

**b.接口说明:**

接该接口用于根据参数条件查询菜单

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findFunctionList | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| menuid | String | | 菜单id | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单功能ID;

"title": "机构A菜单", //功能名称;

"description ": "菜单", //功能描述;

"shortTitle": "17db2a", //功能名称简称;

"englishTitle": "17db2a", //功能名称英文;

"menuid": "kjflkfj", //菜单id;

"picture": "1", //图片;

"url": "1", //页面url;

"page": "1", //页面;

}，{…}],

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 通过菜单id在zjxx\_function （功能表）中查找所有menuid = 接收的menuid的信息
2. 将信息列表返回给前端
   * 1. ~~查找~~**~~功能信息接口（删除）~~**

**a.接口url:**

/pub/admin/user/menu/findFunctionInfo.do

**b.接口说明:**

接该接口用于查找菜单功能信息

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findFunctionInfo | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| id | String | | 功能ID | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单功能ID;

"title": "机构A菜单", //功能名称;

"description ": "菜单", //功能描述;

"shortTitle": "17db2a", //功能名称简称;

"englishTitle": "17db2a", //功能名称英文;

"menuid": "kjflkfj", //菜单id;

"picture": "1", //图片;

"url": "1", //页面url;

"page": "1", //页面;

}]，

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

根据菜单功能id在zjxx\_function （功能表）中查找功能信息

* + 1. **~~删除功能接口（删除）~~**

**a.接口url:**

/pub/admin/user/menu/ deleteFunction.do

**b.接口说明:**

接该接口用于查找菜单功能信息

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteFunction | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| id | String | | 功能ID | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单功能ID;

"title": "机构A菜单", //功能名称;

"description ": "菜单", //功能描述;

"shortTitle": "17db2a", //功能名称简称;

"englishTitle": "17db2a", //功能名称英文;

"menuid": "kjflkfj", //菜单id;

"picture": "1", //图片;

"url": "1", //页面url;

"page": "1", //页面;

}]，

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

根据提交ip调用禁止和允许ip提交函数函数。

根据菜单功能id在zjxx\_function （功能表）中查找功能信息删除

* + 1. 查找机构前端公共菜单、按钮列表

a.**接口url:**

/pub/admin/user/menu/findStaffPublicMenuButton.do

**b.接口说明:**

接该接口用于查找机构前端公共菜单、按钮列表

~~只需要返回公共菜单、按钮id，权限最终体现在人上，页面显示哪些菜单按钮是根据人来定的，这个接口主要是用于配置机构公共菜单页面，只需要返回相应id集合，页面就可以完成相应数据展示。~~

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findStaffPublicMenuButton | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构id | |
| menuid | String | | 菜单id | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单功能ID;

"title": "机构A菜单", //功能名称;

"description ": "菜单", //功能描述;

"shortTitle": "17db2a", //功能名称简称;

"englishTitle": "17db2a", //功能名称英文;

"menuid": "kjflkfj", //菜单id;

"picture": "1", //图片;

"url": "1", //页面url;

"page": "1", //页面;

}]，

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 通过orgid,在zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）中查找publicadminmenuids(前端公共菜单id集合)和publicadminpagebuttons(前端公共页面按钮隐藏id集合)
2. 通过orgid,在zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）中查找publicpagebuttons (前端公共页面按钮隐藏id集合)，然后去zjxx\_buttonfn获取按钮详情数据。
3. 判断menuid是否有值，为空则根据orgid通过orgid在zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）中查找publicmenuids (前端公共菜单功能id集合),然后去zjxx\_menu （菜单表）中查询id在publicmenuids中且menuid为空的数据。
4. 不为空：根据orgid通过orgid在zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）中查找publicmenuids (前端公共菜单功能id集合),然后去zjxx\_menu （菜单表）中查询id在publicmenuids中且menuid为menuid值的数据。
   * 1. 查找机构后台公共菜单、按钮列表

a.**接口url:**

/pub/admin/user/menu/findAdminPublicMenuButton.do

**b.接口说明:**

接该接口用于查找机构后台公共菜单、按钮列表

只需要返回公共菜单功能、按钮id，权限最终体现在人上，页面显示哪些菜单按钮是根据人来定的，这个接口主要是用于配置机构公共菜单页面，只需要返回相应id集合，页面就可以完成相应数据展示。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findAdminPublicMenuButton | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构id | |
| menuid | String | | 菜单id | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单功能ID;

"title": "机构A菜单", //功能名称;

"description ": "菜单", //功能描述;

"shortTitle": "17db2a", //功能名称简称;

"englishTitle": "17db2a", //功能名称英文;

"menuid": "kjflkfj", //菜单id;

"picture": "1", //图片;

"url": "1", //页面url;

"page": "1", //页面;

}]，

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 通过orgid,在zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）中查找publicadminmenuids(后台公共菜单id集合)和publicadminpagebuttons(后台公共页面按钮隐藏id集合)
2. 返回给数据
   * 1. **配置机构前端/后台公共菜单、按钮列表**

a.**接口url:**

/pub/admin/user/menu/setPublicMenuButton.do

**b.接口说明:**

接该接口用于配置机构前端/后台公共菜单、按钮列表

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| setPublicMenuButton | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构id | |
| isAdmin | int | | 0-前端,1-后端 | |
| meunids | String[] | | 菜单功能id数组 | |
| buttonids | String[] | | 按钮id数组 | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单功能ID;

"title": "机构A菜单", //功能名称;

"description ": "菜单", //功能描述;

"shortTitle": "17db2a", //功能名称简称;

"englishTitle": "17db2a", //功能名称英文;

"menuid": "kjflkfj", //菜单id;

"picture": "1", //图片;

"url": "1", //页面url;

"page": "1", //页面;

}]，

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

根据提交ip调用禁止和允许ip提交函数函数。

循环meunids，分别获取每个菜单的pids，然后组成一个新的allmeunids数组去重。

在zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）中根据orgid查询到publicmenuids(前端公共菜单id集合)和publicpagebuttons(前端公共页面按钮隐藏id集合)或者publicadminmenuids(后台公共菜单id集合)和publicadminpagebuttons(后台公共页面按钮隐藏id集合),然后在对应的集合中,将allmeunids与集合去重,再加入集合,然后再讲数据update进数据库

* + 1. **查找机构后台菜单列表（全部带层级关系）**

**a.接口url:**

/pub/admin/user/menu/findMenuFunctionList.do

**b.接口说明:**

接该接口用于查找后台菜单功能列表

目前先做三层

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findMenuFunctionList | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构ID | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单功能ID;

"title": "机构A菜单", //功能名称;

"description ": "菜单", //功能描述;

"shortTitle": "17db2a", //功能名称简称;

"englishTitle": "17db2a", //功能名称英文;

"menuid": "kjflkfj", //菜单id;

"picture": "1", //图片;

"url": "1", //页面url;

"page": "1", //页面;

}]，

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

根据机构id,在zjxx\_orgnizationinfo （机构详情表-机构表功能分表）中取出后台菜单集合字段,然后通过三次循环,先拼接sql,取到顶级菜单,再取子集,再取子集的子集

* + 1. **查找部门前/后端菜单列表**

**a.接口url:**

/pub/admin/user/menu/findDepartMenuFunctionList.do

**b.接口说明:**

接该接口用于查找部门前端菜单列表

目前没有部门后台菜单

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findDepartMenuFunctionList | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 部门id | |
| menuPid | String | | 菜单父id | |
| isAdmin | int | | 0-前端，1-后端 | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单ID;

"title": "机构A菜单", //菜单名称;

"description ": "菜单", //菜单描述;

"isFinal": "1", //是否终极菜单;

"pid": "", //父菜单ID;

"child": "17db2a", //子菜单ID集合;

"shortTitle": "17db2a", //菜单名称简称;

"englishTitle": "17db2a", //菜单名称英文;

"orgid": "kjflkfj", //机构id;

"type": "1", //类型;

}，{…}],

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 通过部门id,在zjxx\_orgnization\_functioninfo（机构详情表-机构表功能分表）取出前端功能菜单id集合 ,如果查菜单，就查zjxx\_menu （菜单表），
2. menuPid为空，查的就是部门的顶级菜单。就返回菜单表中menuPid为空的数据，与功能菜单id集合匹配。
3. 不为空，查询菜单表中pid为menuPid的数据和功能表menuid为menuPid的数据，与功能菜单id集合匹配。
4. 目前部门没有后台菜单，没写后台菜单逻辑。
   * 1. 新增部门前/后端菜单

关信息：

* + - 1. 机构前端菜单中选择配置
      2. 暂不刷新机构人员权限信息，通过页面按钮点击后刷新

**a.接口url:**

/pub/admin/user/menu/addDepartMenu.do

**b.接口说明:**

接该接口用于查找部门前端菜单功能列表

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| addDepartMenu | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 部门id | |
| menuids | String | | 菜单id | |
| isAdmin | int | | 0-前端,1-后端 | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单ID;

"title": "机构A菜单", //菜单名称;

"description ": "菜单", //菜单描述;

"isFinal": "1", //是否终极菜单;

"pid": "", //父菜单ID;

"child": "17db2a", //子菜单ID集合;

"shortTitle": "17db2a", //菜单名称简称;

"englishTitle": "17db2a", //菜单名称英文;

"orgid": "kjflkfj", //机构id;

"type": "1", //类型;

}，{…}],

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

根据提交ip和提交内容调用禁止和允许ip提交函数、屏蔽敏感词函数。

循环meunids，分别获取每个菜单的pids，然后组成一个新的allmeunids数组去重。

把allmeunids加入到zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）中的menuids(前台菜单id集合)或adminmenuids(后台菜单id集合)中，去重后重新update相应数据。

* + 1. 删除部门前/后端菜单

**a.接口url:**

/pub/admin/user/menu/deleteDepartMenu.do

**b.接口说明:**

该接口用于删除部门菜单

相关信息：

暂不刷新机构人员权限信息，通过页面按钮点击后刷新

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteDepartMenu | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| id | String | | 菜单Id | |
| orgid | string | | 部门id | |
| isAdmin | int | | 0-前端,1-后端 | |

**d.返回值示例与说明**

{

"result": { //结果集;

"flag": "1", //标志，1为成功，0为操作失败

"msg": "删除菜单成功" //返回的信息；

}

}

**e.功能实现说明**

根据提交ip调用禁止和允许ip提交函数函数。

根据orgid,在zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）中取出adminmenuids(后台菜单id集合)或menuids(前台菜单id集合),然后再取出的内容中删除对应的菜单id,再把内容update进数据库

* + 1. 查找机构或部门前端角色菜单及按钮

**a.接口url:**

/pub/admin/user/menu/findFrontRoleMenuButtonList.do

**b.接口说明:**

接该接口用于查找机构或部门前端角色菜单及按钮

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findFrontRoleMenuButtonList | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构id | |
| rid | String | | 角色id | |
| Menuid | String | | 菜单id | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单ID;

"title": "机构A菜单", //菜单名称;

"description ": "菜单", //菜单描述;

"isFinal": "1", //是否终极菜单;

"pid": "", //父菜单ID;

"child": "17db2a", //子菜单ID集合;

"shortTitle": "17db2a", //菜单名称简称;

"englishTitle": "17db2a", //菜单名称英文;

"orgid": "kjflkfj", //机构id;

"type": "1", //类型;

}，{…}],

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 通过orgid和rid,在zjxx\_adminrorg （机构｛部门｝角色表-前端）中查找menuids(菜单id集合)和pagebuttons(按钮隐藏设置)
2. Menuid为空，查的就是顶级菜单。就返回zjxx\_menu菜单表中menuPid为空的数据，与功能菜单id集合匹配。
3. 不为空，查询zjxx\_menu菜单表中pid为Menuid的数据，与菜单id集合匹配。
4. 通过查询到的pagebuttons,在zjxx\_buttonfn（页面按钮表）中查找按钮的详细信息
5. 将菜单,按钮的详细信息列表返回给前端
   * 1. 查找机构或部门后端角色菜单及按钮

**a.接口url:**

/pub/admin/user/menu/findAdminRoleMenuButtonList.do

**b.接口说明:**

接该接口用于查找机构或部门后端角色菜单及按钮

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findAdminRoleMenuButtonList | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构id | |
| rid | String | | 角色id | |
| Menuid | String | | 菜单id | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单ID;

"title": "机构A菜单", //菜单名称;

"description ": "菜单", //菜单描述;

"isFinal": "1", //是否终极菜单;

"pid": "", //父菜单ID;

"child": "17db2a", //子菜单ID集合;

"shortTitle": "17db2a", //菜单名称简称;

"englishTitle": "17db2a", //菜单名称英文;

"orgid": "kjflkfj", //机构id;

"type": "1", //类型;

}，{…}],

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 通过orgid和rid,在zjxx\_adminrorg（机构｛部门｝角色表-后台）中查找menuids(菜单功能id集合)和pagebuttons(按钮隐藏设置)
2. ~~Menuid为空，查的就是顶级菜单。就返回菜单表中menuPid为空的数据，与功能菜单id集合匹配。~~
3. ~~不为空，查询菜单表中pid为Menuid的数据，与菜单id集合匹配。~~
4. ~~通过查询到的pagebuttons,在zjxx\_buttonfn（页面按钮表）中查找按钮的详细信息~~
5. ~~将菜单,按钮的详细信息列表返回给前端~~
   * 1. 新增机构或部门前/后端角色菜单及按钮

相关信息：

根据系统设置判断是否调用函数同步更新该角色下人员权限信息

1. 只配置本当前菜单，其子孙菜单不管。
2. 将菜单的pid字段带过来。
3. 要做判断该菜单是否已经配置过。

**接口url:**

/pub/admin/user/menu/addRoleMenuButtonList.do

**b.接口说明:**

接该接口用于查找机构或部门前/后端角色菜单及按钮

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| addRoleMenuButtonList | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构id | |
| rid | String | | 角色id | |
| menuids | String[] | | 菜单id数组(包含父id) | |
| buttonids | String[] | | 按钮id数组 | |
| isAdmin | int | | 0-前端,1-后端 | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单ID;

"title": "机构A菜单", //菜单名称;

"description ": "菜单", //菜单描述;

"isFinal": "1", //是否终极菜单;

"pid": "", //父菜单ID;

"child": "17db2a", //子菜单ID集合;

"shortTitle": "17db2a", //菜单名称简称;

"englishTitle": "17db2a", //菜单名称英文;

"orgid": "kjflkfj", //机构id;

"type": "1", //类型;

}，{…}],

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 根据提交ip和提交内容调用禁止和允许ip提交函数、屏蔽敏感词函数。
2. 判断机构角色表是否存在 这条机构、部门角色数据，不存在就添加
3. 通过isAdmin来区分zjxx\_adminrorg （机构｛部门｝角色表-后台）或zjxx\_frontrorg （机构｛部门｝角色表-前端）
4. ~~通过机构/部门id和角色id来查询出~~根据上传的menuids(菜单id集合)字段和pagebuttons(按钮隐藏设置)字段
5. 循环meunids，分别获取每个菜单的pids，然后组成一个新的allmeunids数组去重。
6. allmeunids和buttonid在menuids(菜单id集合)和pagebuttons(按钮隐藏设置)中去重后加入
7. 将menuids(菜单功能id集合)和pagebuttons(按钮隐藏设置)重新update进数据库
   * 1. 删除机构或部门前端角色菜单及按钮

相关信息：

根据系统设置判断是否调用函数同步更新该角色下人员权限信息

**接口url:**

/pub/admin/user/menu/deleteRoleMenuButtonList.do

**b.接口说明:**

接该接口用于查找机构或部门前/后端角色菜单及按钮

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteRoleMenuButtonList | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构id | |
| rid | String | | 角色id | |
| menuids | String[] | | 菜单id数组(包含父id) | |
| buttonid | String | | 按钮id | |
| isAdmin | int | | 0-前端,1-后端 | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单ID;

"title": "机构A菜单", //菜单名称;

"description ": "菜单", //菜单描述;

"isFinal": "1", //是否终极菜单;

"pid": "", //父菜单ID;

"child": "17db2a", //子菜单ID集合;

"shortTitle": "17db2a", //菜单名称简称;

"englishTitle": "17db2a", //菜单名称英文;

"orgid": "kjflkfj", //机构id;

"type": "1", //类型;

}，{…}],

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 根据提交ip调用禁止和允许ip提交函数函数。
2. 通过isAdmin来区分zjxx\_adminrorg （机构｛部门｝角色表-后台）或zjxx\_frontrorg （机构｛部门｝角色表-前端）
3. 通过机构/部门id和角色id来查询出menuids(菜单功能id集合)字段和pagebuttons(按钮隐藏设置)字段
4. 将接收的menuids和buttonid在menuids(菜单功能id集合)和pagebuttons(按钮隐藏设置)中去除
5. 将menuids(菜单功能id集合)和pagebuttons(按钮隐藏设置)重新update进数据库
   * 1. 查询人员前/后端私有菜单及按钮

**接口url:**

/pub/admin/user/menu/findPeopleMenuButtonList.do

**b.接口说明:**

接该接口用于查询人员前/后端私有菜单及按钮

前端需要根据menuid判断是否是 顶级菜单，还是某个父菜单的子菜单。

后台只需要返回相应的id集合，后台展示页面会展示所有菜单，只需要根据id集合匹配打勾 就可以了。

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findPeopleMenuButtonList | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构id | |
| peoid | String | | 人员id | |
| menuid | String | | 父菜单id | |
| isAdmin | int | | 0-前端,1-后端 | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单ID;

"title": "机构A菜单", //菜单名称;

"description ": "菜单", //菜单描述;

"isFinal": "1", //是否终极菜单;

"pid": "", //父菜单ID;

"child": "17db2a", //子菜单ID集合;

"shortTitle": "17db2a", //菜单名称简称;

"englishTitle": "17db2a", //菜单名称英文;

"orgid": "kjflkfj", //机构id;

"type": "1", //类型;

}，{…}],

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 通过isAdmin来区分zjxx\_frontorg（机构｛部门｝员工表-前端）或zjxx\_adminorg （机构｛部门｝员工表-后台）
2. isAdmin==1，通过机构id和人员id来查询出personnelmenuids (人员私有的功能权限)字段和personnelbuttonids (人员按钮私有的功能权限)字段，返回这两个id集合就可以了。
3. isAdmin==0，通过机构id和人员id来查询出personnelmenuids (人员私有的功能权限)字段和personnelbuttonids (人员按钮私有的功能权限)字段。menuPid为空，查的就是顶级菜单。就返回zjxx\_menu菜单表中menuPid为空的数据，与功能菜单personnelmenuids集合匹配。不为空，查询zjxx\_menu菜单表中pid为menuPid的数据的数据，与功能菜单personnelmenuids集合匹配，获取匹配的菜单功能信息列表。在根据personnelbuttonids去 zjxx\_buttonfn（页面按钮表）查出按钮信息。
   * 1. 新增人员前/后端私有菜单及按钮

相关信息：

调用函数同步更新该人员权限信息

1. 只配置本当前菜单，其子孙菜单不管。
2. 需要关联父级菜单，同步配置该角色的所有父级菜单。
3. 将菜单的pid字段带过来。
4. 要做判断该菜单是否已经配置过。

**接口url:**

/pub/admin/user/menu/addPeopleMenuButtonList.do

**b.接口说明:**

接该接口用于新增人员前/后端私有菜单及按钮

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| addPeopleMenuButtonList | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构id | |
| peoid | String | | 人员id | |
| isAdmin | int | | 0-前端,1-后端 | |
| menuids | String[] | | 菜单id数组(包含父id) | |
| buttonids | String[] | | 按钮id数组 | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单ID;

"title": "机构A菜单", //菜单名称;

"description ": "菜单", //菜单描述;

"isFinal": "1", //是否终极菜单;

"pid": "", //父菜单ID;

"child": "17db2a", //子菜单ID集合;

"shortTitle": "17db2a", //菜单名称简称;

"englishTitle": "17db2a", //菜单名称英文;

"orgid": "kjflkfj", //机构id;

"type": "1", //类型;

}，{…}],

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 根据提交ip和提交内容调用禁止和允许ip提交函数、屏蔽敏感词函数。
2. 通过isAdmin来区分zjxx\_frontorg（机构｛部门｝员工表-前端）或zjxx\_adminorg （机构｛部门｝员工表-后台）
3. 循环meunids，分别获取每个菜单的pids，然后组成一个新的allmeunids数组去重。
4. 通过机构id和人员id来查询出personnelmenuids (人员私有的功能权限)字段和personnelbuttonids (人员按钮私有的功能权限)字段
5. 将allmeunids数组和按钮id数组与数据库中查到的数据去重后,重新update进数据库
   * 1. 删除人员前/后端私有菜单及按钮

相关信息：

调用函数同步更新该人员权限信息

**接口url:**

/pub/admin/user/menu/deletePeopleMenuButtonList.do

**b.接口说明:**

接该接口用于新增人员前/后端私有菜单及按钮

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deletePeopleMenuButtonList | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构id | |
| peoid | String | | 人员id | |
| isAdmin | int | | 0-前端,1-后端 | |
| menuids | String[] | | 菜单功能id数组(包含父id) | |
| buttonids | String[] | | 按钮id数组 | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单ID;

"title": "机构A菜单", //菜单名称;

"description ": "菜单", //菜单描述;

"isFinal": "1", //是否终极菜单;

"pid": "", //父菜单ID;

"child": "17db2a", //子菜单ID集合;

"shortTitle": "17db2a", //菜单名称简称;

"englishTitle": "17db2a", //菜单名称英文;

"orgid": "kjflkfj", //机构id;

"type": "1", //类型;

}，{…}],

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 通过isAdmin来区分zjxx\_frontorg（机构｛部门｝员工表-前端）或zjxx\_adminorg （机构｛部门｝员工表-后台）
2. 通过机构id和人员id来查询出personnelmenuids (人员私有的功能权限)字段和personnelbuttonids (人员按钮私有的功能权限)字段
3. 将前端传递的菜单id数组和按钮id数组在数据库中查到的数据中去除后,重新update进数据库
   * 1. 查找人员前端菜单

**接口url:**

/pub/admin/user/menu/findPeopleMenuList.do

**b.接口说明:**

接该接口用于查找人员前端菜单

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findPeopleMenuList | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构id | |
| peid | String | | 人员id | |
| menuid | text | | 查询子菜单时使用，查顶级菜单时为空 | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单ID;

"title": "机构A菜单", //菜单名称;

"description ": "菜单", //菜单描述;

"isFinal": "1", //是否终极菜单;

"pid": "", //父菜单ID;

"child": "17db2a", //子菜单ID集合;

"shortTitle": "17db2a", //菜单名称简称;

"englishTitle": "17db2a", //菜单名称英文;

"orgid": "kjflkfj", //机构id;

"type": "1", //类型;

}，{…}],

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 通过机构id和人员id在zjxx\_frontorg（机构｛部门｝员工表-前端）来查询出menuids (菜单功能id集合)字段
2. menuid为空：查询顶级菜单则用查询到的菜单id集合和父id为空作为条件在zjxx\_menu （菜单表）中查找顶级菜单详情
3. menuid不为空：如果查询子菜单则用菜单id集合和父id为上传的menuid为值,在zjxx\_menu （菜单表）查找详情信息
   * 1. 查找人员后台菜单

**接口url:**

/pub/admin/user/menu/getTopMenu.do

/pub/admin/user/menu/getLeftMenu.do

**b.接口说明:**

接该接口用于查找人员后端菜单

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findPeoplefrontMenuList | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构id | |
| peoid | String | | 人员id | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单ID;

"title": "机构A菜单", //菜单名称;

"description ": "菜单", //菜单描述;

"isFinal": "1", //是否终极菜单;

"pid": "", //父菜单ID;

"child": "17db2a", //子菜单ID集合;

"shortTitle": "17db2a", //菜单名称简称;

"englishTitle": "17db2a", //菜单名称英文;

"orgid": "kjflkfj", //机构id;

"type": "1", //类型;

}，{…}],

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

1. 查询本地顶部菜单html文件是否存在，如存在直接读取，若不存在则通过机构id和人员id在zjxx\_ adminorg（机构｛部门｝员工表-后台）来查询出menuids (菜单功能id集合)字段
2. 如果查询顶级菜单则用查询到的菜单id集合和父id为空作为条件在zjxx\_menu （菜单表）中查找顶级菜单详情构成html形式返回并生成html文件
3. 点击顶部菜单时分为两种情况：a、点击的的是信息，调用查找栏目接口，

查找到的如果是终极栏目直接放在页面，如果为非终极栏目放在页面并将栏目id放入数组，再把数组再次调用接口查找，并按照标签id的位置把查到的信息放入，如果有非终极栏目重复执行上一步。（查栏目时注意该用户是否有栏目的权限）

1. B、点击的不是信息，调用查找左侧菜单接口。先查询本地左侧菜单html文件是否存在，如存在直接读取，若不存在则通过机构id和人员id在zjxx\_ adminorg（机构｛部门｝员工表-后台）来查询出menuids (菜单功能id集合)字段，循环对比找到顶部菜单的子菜单，再对比找儿子。

，将结果以HTML形式返回并生成文件。

Topmenu：topmenu\_14b8c8f3-7338-4d4c-a924-37c3db7a7fb5\_9db7c86d-a6c2-4bd4-ad02-3dcc325f1b40.html

Leftmenu：leftmenu\_14b8c8f3-7338-4d4c-a924-37c3db7a7fb5\_9db7c86d-a6c2-4bd4-ad02-3dcc325f1b40273f4644-4ce4-4812-95bb-e236d03a1365

* + 1. 配置机构后台按钮接口(删除，不实现)

**接口url:**

/pub/admin/user/menu/setAdminButton.do

**b.接口说明:**

该接口用于查找人员前/后端菜单

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| setAdminButton | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| ZjxxPageButton | obj | | 按钮对象 | |
| orgid | string | | 机构id | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单ID;

"title": "机构A菜单", //菜单名称;

"description ": "菜单", //菜单描述;

"isFinal": "1", //是否终极菜单;

"pid": "", //父菜单ID;

"child": "17db2a", //子菜单ID集合;

"shortTitle": "17db2a", //菜单名称简称;

"englishTitle": "17db2a", //菜单名称英文;

"orgid": "kjflkfj", //机构id;

"type": "1", //类型;

}，{…}],

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

根据提交ip调用禁止和允许ip提交函数函数。

通过orgid在zjxx\_orgnizationinfo （机构详情表-机构表功能分表）中取出pagebuttons(页面按钮设置),然后把按钮对象加入后,update

* + 1. 查询机构后台按钮列表接口(删除，不实现)

**接口url:**

/pub/admin/user/menu/findAdminButton.do

**b.接口说明:**

接该接口用于查询机构后台按钮列表接口

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findAdminButton | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | string | | 机构id | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单ID;

"title": "机构A菜单", //菜单名称;

"description ": "菜单", //菜单描述;

"isFinal": "1", //是否终极菜单;

"pid": "", //父菜单ID;

"child": "17db2a", //子菜单ID集合;

"shortTitle": "17db2a", //菜单名称简称;

"englishTitle": "17db2a", //菜单名称英文;

"orgid": "kjflkfj", //机构id;

"type": "1", //类型;

}，{…}],

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

通过orgid在zjxx\_buttonfn （页面按钮表）中取出相应信息列表

* + 1. 删除机构后台按钮接口(删除，不实现)

**接口url:**

/pub/admin/user/menu/deleteAdminButton.do

**b.接口说明:**

接该接口用于查询机构后台按钮列表接口

**c.接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteAdminButton | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | string | | 机构id | |
| buttonid | string | | 按钮id | |

**d.返回值示例与说明**

{

“infos”:

[{

"id": "80", //菜单ID;

"title": "机构A菜单", //菜单名称;

"description ": "菜单", //菜单描述;

"isFinal": "1", //是否终极菜单;

"pid": "", //父菜单ID;

"child": "17db2a", //子菜单ID集合;

"shortTitle": "17db2a", //菜单名称简称;

"englishTitle": "17db2a", //菜单名称英文;

"orgid": "kjflkfj", //机构id;

"type": "1", //类型;

}，{…}],

“result”：{

“flag”：1， //成功标志

“msg”：“查找成功” //说明

}

}

**e.功能实现说明**

根据提交ip调用禁止和允许ip提交函数函数。

通过orgid在zjxx\_orgnizationinfo （机构详情表-机构表功能分表）中取出pagebuttons(页面按钮设置),然后把传递的id在其中删除,再update进数据库

* 1. 按钮管理
     1. 获取按钮列表

**a.接口url:**

/pub/admin/GetPageButtonList.do

**b.接口说明:**

**查询配置按钮的信息**

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| GetPageButtonList | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| pagesize | String | | 页数 | | |
| orgid | String | | 机构ID | | |
| nowpage | int | | 条数 | | |

**d.返回值示例与说明**{

{

"result": { //结果

"msg": "", //信息

"flag": "1" //标识

}，

"infos": [ //信息;

{

"buttonid": "wedff887sds", //按钮ID；

"buttonname": "按钮1", //按钮名称;

"htmlname": 0, //按妞页面;

},

{

"buttonid": "wedff887sds", //按钮ID；

"buttonname": "按钮1", //按钮名称;

"htmlname": 0, //按妞页面;

}

]

}

**e.功能实现说明**

（1）~~根据参数orgid，在zjxx\_buttonfn（页面按钮表）中获取按钮列表~~

根据orgid，查找zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）的pagebuttons字段，再去zjxx\_buttonfn（页面按钮表）中获取对应按钮详情列表

* + 1. 获取按钮详情

**a.接口url:**

/pub/admin/GetPageButtonById.do

**b.接口说明:**

**查询配置按钮的信息**

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| GetPageButtonById | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| id | String | | 按钮id | | |

**d.返回值示例与说明**{

{

"result": { //结果

"msg": "", //信息

"flag": "1" //标识

}，

"infos": [ //信息;

{

"buttonid": "wedff887sds", //按钮ID；

"buttonname": "按钮1", //按钮名称;

"htmlname": 0, //按妞页面;

},

]

}

**e.功能实现说明**

（1）根据参数id，在zjxx\_buttonfn（页面按钮表）获取按钮详情

* + 1. 修改和增加按钮

**a.接口url:**

/pub/admin/AddOrUpdataPageButton.do

**b.接口说明:**

**查询配置按钮的信息**

**相关信息：**

1. 系统设置-网路防火墙：提交内容屏蔽敏感词汇
2. 系统设置-用户设置-统一设置：允许提交ip、禁止提交ip

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| AddOrUpdataPageButton | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| ZjxxPageButton | obj | | 按钮对象 | | |
| orgid | String | | 机构id | | |

**d.返回值示例与说明**{

{

"result": { //结果

"msg": "增加或修改成功！", //信息

"flag": "1" //标识

}

}

**e.功能实现说明**

根据提交ip和提交内容调用禁止和允许ip提交函数、屏蔽敏感词函数。

* + 1. 判断按钮对象中id字段是否为空，为空就是新增，不为空就是修改。
    2. 新增时，将信息添加于zjxx\_buttonfn（页面按钮表）中。根据orgid获取机构功能信息，将按钮id放入机构功能表中pagebuttons（页面按钮设置）字段中。
    3. 修改时，将信息更新于zjxx\_buttonfn（页面按钮表）中
    4. 删除按钮

**a.接口url:**

/pub/admin/DeletePageButton.do

**b.接口说明:**

**查询配置按钮的信息**

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| DeletePageButton | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| id | String | | 按钮id | | |

**d.返回值示例与说明**{

{

"result": { //结果

"msg": "删除成功！", //信息

"flag": "1" //标识

}

}

**e.功能实现说明**

根据提交ip调用禁止和允许ip提交函数函数。

根据参数id，查找zjxx\_buttonfn（页面按钮表）中的数据，获取按钮orgid，通过orgid获取机构功能信息，删除中pagebuttons（页面按钮设置）字段中id的值。同时删除zjxx\_buttonfn（页面按钮表）中的数据。

* 1. 人员权限
     1. 更新前端人员权限（个人）

**a.接口url:**

/pub/admin/refreshUserPermission.do

**b.接口说明:**

**更新某个前端人员权限**

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| refreshUserPermission | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| userid | String | | 人员id | | |
| orgid | String | | 机构id | | |

**d.返回值示例与说明**{

{

"result": { //结果

"msg": "删除成功！", //信息

"flag": "1" //标识

}

}

**e.功能实现说明**

* 1. 根据提交ip调用禁止和允许ip提交函数函数。
  2. 判断orgid是否为机构id，如果不是获取部门的机构id（现在默认orgid就是机构id，没做判断。）
  3. 根据参数userid和机构id，查找zjxx\_frontorg（机构｛部门｝员工表-前端）中的数据。
  4. 根据机构id查询zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）获取机构前端公共菜单id集合。
  5. 根据zjxx\_frontorg（机构｛部门｝员工表-前端）中的数据获取orgids（员工所属部门id集合），分别获取部门公共菜单id合集。
  6. 根据zjxx\_frontorg（机构｛部门｝员工表-前端）中的数据获取rids（角色表ID集合），分别获取机构\部门角色菜单id合集。
  7. 根据zjxx\_frontorg（机构｛部门｝员工表-前端）中的数据获取personnelbuttonids（人员私有的按钮权限id合集）。
  8. 整合 d、e、f、g数据，去重获取人员功能权限id合集数据，更新zjxx\_frontorg（机构｛部门｝员工表-前端）中的对应数据
     1. 更新机构\部门人员权限

**a.接口url:**

/pub/admin/refreshUserOrgMenu.do

**b.接口说明:**

**更新机构\部门下所有前端人员权限**

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| refreshUserOrgMenu | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| userid | String | | 人员id | | |
| orgid | String | | 机构id | | |

**d.返回值示例与说明**{

{

"result": { //结果

"msg": "删除成功！", //信息

"flag": "1" //标识

}

}

**e.功能实现说明**

根据提交ip调用禁止和允许ip提交函数函数。

1. 第一次前端传 userid为空，根据orgid获取机构\部门第一员工的id（按时间升序），返回给前端。select peid from zjxx\_ frontorg where orgid = orgid order by peid desc limit 1。
2. 判断orgid是否为机构id，如果不是获取部门的机构id（现在默认orgid就是机构id，没做判断。）
3. 根据前端传的userid和机构id，调用刷新员工权限函数，更新员工权限。
4. 更新完人员权限后，根据userid获取下一个userid。select peid from zjxx\_frontorg where peid = userid and orgid = orgid order by peid desc limit 1;返回给前端下一个userid。
   * 1. 更新管理员权限（个人）

**a.接口url:**

/pub/admin/refreshAdminPermission.do

**b.接口说明:**

**更新某个管理员的权限**

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| refreshAdminPermission | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| userid | String | | 管理员id | | |
| orgid | String | | 机构id | | |

**d.返回值示例与说明**{

{

"result": { //结果

"msg": "删除成功！", //信息

"flag": "1" //标识

}

}

**e.功能实现说明**

1. 根据提交ip调用禁止和允许ip提交函数函数。
2. 判断orgid是否为机构id，如果不是获取部门的机构id（现在默认orgid就是机构id，没做判断。）
3. 根据参数userid和orgid，查找zjxx\_adminorg （机构｛部门｝员工表-后台）中的数据。
4. 根据机构id查询zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）获取机构后台公共菜单id集合。
5. 根据zjxx\_adminorg （机构｛部门｝员工表-后台）中的数据获取orgids（员工所属部门id集合），分别获取部门后台公共菜单id合集。
6. 根据zjxx\_adminorg （机构｛部门｝员工表-后台）中的数据获取rids（角色表ID集合），分别获取机构\部门角色菜单id合集。
7. 根据zjxx\_adminorg （机构｛部门｝员工表-后台）中的数据获取personnelbuttonids（管理员私有的按钮权限id合集）。
8. 整合 d、e、f、g数据，去重获取管理员功能权限id合集数据，更新zjxx\_adminorg （机构｛部门｝员工表-后台）中的对应数据
   * 1. 更新机构\部门管理员权限

**a.接口url:**

/pub/admin/.do

**b.接口说明:**

**更新机构\部门下所有管理员的权限**

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| refreshAdminOrgMenu | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| userid | String | | 管理员id | | |
| orgid | String | | 机构\部门id | | |

**d.返回值示例与说明**{

{

"result": { //结果

"msg": "删除成功！", //信息

"flag": "1" //标识

}

}

**e.功能实现说明**

根据提交ip调用禁止和允许ip提交函数函数。

1. 第一次前端传 userid为空，根据orgid获取机构\部门第一个管理员的id（按时间降序），返回给前端。select peid from zjxx\_ adminorg where orgid = orgid order by peid desc limit 1。
2. 判断orgid是否为机构id，如果不是获取部门的机构id（现在默认orgid就是机构id，没做判断。）
3. 根据前端传的userid和机构id，调用刷新管理员权限函数，更新管理员权限。
4. 更新完管理员权限后，根据userid获取下一个userid。select peid from zjxx\_ adminorg where peid = userid and orgid = orgid order by peid desc limit 1;返回给前端下一个userid。

。

1. 函数
   * 1. 更新前端人员权限函数

**函数定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 函数名 | 对象名 | 返回值类型 |
| getUserMenu |  |  |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| userid | String | 人员id |
| orgid | string | 机构id |

**功能说明：**

1. 默认orgid就是机构id（现在默认orgid就是机构id，没做判断。）
2. 根据参数userid和orgid，查找zjxx\_adminorg （机构｛部门｝员工表-后台）中的数据。
3. 根据orgid查询zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）获取机构后台公共菜单id集合。
4. 根据zjxx\_adminorg （机构｛部门｝员工表-后台）中的数据获取orgids（员工所属部门id集合），分别获取部门后台公共菜单id合集。
5. 根据zjxx\_adminorg （机构｛部门｝员工表-后台）中的数据获取rids（角色表ID集合）；拆分rids（部门|角色）在根据部门和角色分别查询pids （f分别查询机构表和角色表）；根据获取的orgpids（部门）和rolepids(角色)，组成sql(select menuids from zjxx\_adminrorg where orgid in ("+orgpids+") and rid in ("+rolepids+")")
6. 根据zjxx\_adminorg （机构｛部门｝员工表-后台）中的数据获取personnelbuttonids（管理员私有的按钮权限id合集）。
7. 整合 d、e、f、g数据，去重获取管理员功能权限id合集数据，更新zjxx\_adminorg （机构｛部门｝员工表-后台）中的对应数据
   * 1. 更新后台人员权限函数

**函数定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 函数名 | 对象名 | 返回值类型 |
| getAdminMenu |  |  |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| userid | String | 人员id |
| orgid | string | 机构id |

**功能说明：**

1. 默认orgid是机构id（现在默认orgid就是机构id，没做判断。）
2. 根据参数userid和orgid，查找zjxx\_adminorg （机构｛部门｝员工表-后台）中的数据。
3. 根据机构id查询zjxx\_orgnization\_functioninfo （机构详情表-机构表功能分表）获取机构后台公共菜单id集合。
4. 根据zjxx\_adminorg （机构｛部门｝员工表-后台）中的数据获取orgids（员工所属部门id集合），分别获取部门后台公共菜单id合集。
5. 根据zjxx\_adminorg （机构｛部门｝员工表-后台）中的数据获取rids（角色表ID集合），分别获取机构\部门角色菜单id合集。
6. 根据zjxx\_adminorg （机构｛部门｝员工表-后台）中的数据获取personnelbuttonids（管理员私有的按钮权限id合集）。
7. 整合 d、e、f、g数据，去重获取管理员功能权限id合集数据，更新zjxx\_adminorg （机构｛部门｝员工表-后台）中的对应数据
   * 1. 刷新pids函数

**函数定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 函数名 | 对象名 | 返回值类型 |
| refreshPids |  |  |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| id | String | 基础数据id |
| type | int | 基础数据类型 0机构 1角色2数据词典 3菜单 |

**功能说明：**

先刷新自己的pids，返回pid=id的数据，递归继续刷