目录

[说明：接口都要增加机构id和人员id作为参数 2](#_Toc494025778)

[1 栏目 2](#_Toc494025779)

[1.1 栏目管理 2](#_Toc494025780)

[1.1.1 查看栏目列表接口 2](#_Toc494025781)

[1.1.2 查看栏目详情接口 5](#_Toc494025782)

[1.1.3 删除栏目接口 5](#_Toc494025783)

[1.1.4 增加/修改栏目接口 6](#_Toc494025784)

[1.1.5 修改栏目选项设置接口 7](#_Toc494025785)

[1.1.6 查找栏目选项设置接口 7](#_Toc494025786)

[1.1.7 修改栏目生成选项接口 7](#_Toc494025787)

[1.1.8 查找栏目生成选项接口 7](#_Toc494025788)

[1.1.9 修改栏目顺序接口 7](#_Toc494025789)

[1.1.10 栏目转换信息接口（只转信息） 8](#_Toc494025790)

[1.1.11 刷新栏目SQL缓存 9](#_Toc494025791)

[1.1.12 刷新栏目列表信息接口 10](#_Toc494025792)

[1.1.13 清除栏目sql缓存接口 10](#_Toc494025793)

[1.1.14 清除栏目列表信息缓存接口 10](#_Toc494025794)

[1.1.15 栏目转换信息接口 10](#_Toc494025795)

[1.1.16 批量修改栏目接口 11](#_Toc494025796)

[1.1.17 批量设置栏目信息转换接口 11](#_Toc494025797)

[1.1.18 修改栏目顺序接口 11](#_Toc494025798)

[1.1.19 终极栏目转化接口 11](#_Toc494025799)

[1.2 暂不解决接口 11](#_Toc494025800)

[1.2.1 修改栏目模板接口 11](#_Toc494025801)

[1.2.2 查询栏目模板接口 11](#_Toc494025802)

[1.2.3 查找关联字段 14](#_Toc494025803)

[1.2.4 栏目归档接口 15](#_Toc494025804)

[1.2.5 设置栏目信息转换规则接口 15](#_Toc494025805)

[1.3 函数 15](#_Toc494025806)

[1.3.1 “刷新条件列表栏目sql”函数 15](#_Toc494025807)

[1.3.2 “刷新条件列表信息缓存”函数 15](#_Toc494025808)

[1.3.3 “清除条件列表信息缓存”函数 15](#_Toc494025809)

[1.3.4 “刷新栏目sql缓存”函数 15](#_Toc494025810)

[1.3.5 “刷新栏目列表信息缓存”函数 15](#_Toc494025811)

[1.3.6 “清除栏目sql缓存”函数 15](#_Toc494025812)

[1.3.7 “清除栏目列表信息缓存”函数 15](#_Toc494025813)

说明：接口都要增加机构id和人员id作为参数

1. 栏目
   1. 栏目管理
      1. **查看栏目列表接口**

**a.接口url:**

pub/admin/ column/manageColumn/findColumnList.do

**b.接口说明：（**pageSize和nowPage两个参数必须要传**）**

相关说明：

1. 根据管理员的机构id获取本机构的栏目

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findColumnList | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| orgid | String | | 机构Id | |
| userid | String | | 人员id | |
| pageSize | Integer | | 每页尺寸 | |
| nowPage | Integer | | 当前页 | |
| classname | String | | 栏目名称(搜索使用) | |
| pclassid | String | | 父栏目id | |

**d.返回值示例与说明**

{

"page": { //分页信息;

"page": , //当前页;

"pages": , //总页数;

"lines": , //信息总条数;

"pageSize": , //每页显示信息数;

"pageStart": //起始页;

},

"total": , //当前页信息条数;

"infos": [

{

},

{

}

]

}

**e.功能实现说明**

1. 先检查orgid，若上传参数orgid为空，则查询系统设置的项目基本信息配置中机构id（获取平台orgid函数）如果函数返回false，则直接返回false。
2. 检测nowPage是否为空或为0，如果是，设置nowPage=1；检测pageSize是否为空或为0，如果是，设置pageSize=30
3. 如果参数classname不为空，拼接查询条件，模糊查询栏目表中title字段值like参数classname值的数据
4. 拼接查找条件，根据pid，zorder字段拼接

limit (nowPage\*pageSize-pageSize),pagesize；查找isdelete为0的数据

1. 统计查询数据条数和行数
2. 最后将数据都整合到info对象中返回出来
   * 1. **查看栏目详情接口**

**a.接口url:**

pub/admin/column/manageColumn/findColumnDetails.do

**b.接口说明：**

1．根据classid查询该栏目详情

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findColumnDetails | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| classid | String | | 栏目的id | |
| orgid | String | | 机构Id | |
| userid | String | | 人员id | |
| type | Int | | 0-所有，1-主表，2-基本属性分表，3-选项设置分表 | |

**d.返回值示例与说明**

{

"infos": [

{

},

{

}

]

}

**e.功能实现说明**

1. 先检查orgid，若上传参数orgid为空，则查询系统设置的项目基本信息配置中机构id（获取平台orgid函数）如果函数返回false，则直接返回false。
2. 根据参数classid查询栏目表中id字段值为参数classid的数据

3）返回Result

* + 1. **删除栏目接口**

**a.接口url:**

pub/admin/column/manageColumn/deleteColumn.do

**b.接口说明：**

1．删除栏目只删除栏目的信息（即该栏目在栏目表对应的该条数据），子栏目和栏目内的数据暂时不管，不需要作处理

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteColumn | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| classid |  | | 栏目id | |
| orgid | String | | 机构Id | |
| userid | String | | 人员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": "删除栏目成功" //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
2. 如果是终极栏目，根据栏目的classid查找出栏目主表一条记录,从记录中的得到系统模型modid（所属系统模型ID）。根据modid来查找zjxx\_model（系统模型表）的记录，得到终极栏目ID集合sonclass。删除终极栏目ID集合sonclass里面的对应classid。
3. 将isdelete改为1，实现伪删除
4. 创建Resultd对象，删除成功则设置msg为删除栏目成功，flag为1，否则设置msg为“删除失败”，flag为0，返回这个Result对象。
   * 1. **增加/修改栏目接口**

**a.接口url:**

pub/admin/column/manageColumn/addOrUpdateColumn.do

**b.接口说明：**

增加或修改栏目的数据

相关说明：

1. 超级管理员没有机构id，建立的栏目没有机构id。
2. 增加栏目时，将系统模型内勾选的内容带过来（存在固定的字段内）
3. 增加栏目时，先调用该接口，再根据返回的classid去修改栏目的选项设置

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| addOrUpdateColumn | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| columninfor | ZjxxClass对象 | | 栏目信息对象 | |
| orgid | String | | 机构Id | |
| userid | String | | 人员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": "修改成功" //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. 根据对象参数columninfor，判断对象中id字段值是否为空：
   * 1. 假如id为空：（增加）
        1. 设置uuid给对象中columninfor,的id。
        2. 修改相应的栏目信息（如果islast为1，为终极栏目，根据提交的modid来查找zjxx\_model（系统模型表）的记录，得到终极栏目ID集合sonclass。在终极栏目ID集合sonclass里面加上的对应classid）
        3. 根据增加好的栏目的classid查找出栏目主表一条记录,从记录中的得到系统模型modid（所属系统模型ID）根据modid来查找zjxx\_model（系统模型表）的记录，得到listtempvar管理员列表项，frontlist前端列表项，frontinfor前端内容项，分别修改栏目表里对应的listtempvar列表接口管理员字段，frontlist列表接口用户字段，frontinfor内容接口用户字段，再更新zjxx\_cache\_classsql （栏目sql缓存表）里的adminfields管理员fields子句，listfields列表fields子句，inforfields内容fields字句
        4. 如果增加成功，设置msg为增加栏目成功，flag为0，否则设置msg为增加栏目失败，flag为1。
     2. 假如id为不为空：（修改）
        1. 如果islast为1，为终极栏目，判断所属系统模型是否改变（根据栏目的classid查找出栏目主表一条记录,从记录中的得到系统模型modid（所属系统模型ID）。

与要修改提交的modid对比，相同则属系统模型未改变。

不同则先根据得到的modid来查找zjxx\_model（系统模型表）的记录，得到终极栏目ID集合sonclass。删除终极栏目ID集合sonclass里面的对应classid。再根据要修改提交的modid来查找zjxx\_model（系统模型表）的记录，得到终极栏目ID集合sonclass。在终极栏目ID集合sonclass里面加上的对应classid。）。为0，则为非终极栏目 ，

* + - 1. 修改相应的栏目信息（修改栏目表里的listtempvar列表接口管理员字段，frontlist列表接口用户字段，frontinfor内容接口用户字段，并更新zjxx\_cache\_classsql （栏目sql缓存表）里的adminfields管理员fields子句，listfields列表fields子句，inforfields内容fields字句，同样修改栏目表里的listorder管理信息排序方式，reorder用户信息排序方式字段时并更新zjxx\_cache\_classsql （栏目sql缓存表）里的adminorder管理员order子句，order用户order子句）
      2. 如果修改成功，设置msg为添加栏目成功，flag为0，否则设置msg为修改栏目失败，flag为1.
    1. 调用刷新栏目sql缓存函数，设置或刷新栏目sql缓存。
    2. 返回Result对象
    3. **修改栏目顺序接口**

**a.接口url:**

pub/admin/column/manageColumn/updateColumnOrder.do

**b.接口说明：**

修改栏目顺序的接口

相关说明：值越小，顺序越靠前

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| updateColumnOrder | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| myorder | String[] | | 顺序值的数组 | |
| classid | String[] | | 栏目Id的数组（顺序值和栏目Id是一一对应） | |

**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": "修改成功" //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. 创建Resultd对象。
3. 遍历两个数组，每次遍历得到order，classid参数，根据每次遍历的这两个参数修改栏目表中id字段值为参数classid值的顺序字段，将顺序字段值修改为参数order的值。
4. 如果修改成功，设置msg为修改栏目顺序成功，flag为1，如果遍历中抛异常设置msg为修改栏目顺序失败，flag为0.
5. 返回这个Result对象。
   * 1. **栏目转换信息接口(栏目转信息)**

**a.接口url:**

pub/admin/column/manageColumn/infoTransferTocolumn.do

**b.接口说明:**

将栏目转换为信息数据

该栏目不变动，实质是在转化父栏目中复制一条信息（该信息的id、title为转化栏目的id、title）

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| infoTransferTocolumn | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| classid | String | | 栏目的id | | |
| classname | String | | 栏目的名字 | | |
| fclassid | String | | 目标栏目id | | |

**d.返回值示例与说明**

{

"flag": "1", //返回值标志

"msg": "转移成功", //返回值信息

}

**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
2. 接收参数classid、classnamee、fid，判断有任意一个空值则返回“来源错误”。
3. 根据classid获取来源数据，若找不到数据则同样返回“来源错误”。
4. 获取来源字段fclassid，获取栏目对应的系统模型。
5. 根据系统模型获得对应的信息表的表名tbname。
6. 根据参数classid、classname，将classid作为信息表id，将classname作为信息表title，将这条信息插入信息表tbname。
7. 返回result。
   * 1. **刷新栏目SQL缓存**
8. **接口url:**

pub/admin/column/manageColumn/refreshColumnSql.do

**b.接口说明:**

**c. 接口定义：**

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

d. **返回值示例与说明**

e. **功能实现说明**

判断栏目是否为条件列表-zjxx\_column （栏目主表）listselect（为空代表不是条件列表），如果是则调用刷新栏目条件列表sql函数，刷新栏目条件列表sql缓存，不是：刷新栏目sql函数，刷新栏目sql缓存。

* + 1. **刷新栏目列表信息缓存**

1. **接口url:**

pub/admin/column/manageColumn/refreshColumnListCache.do

**b.接口说明:**

**c. 接口定义：**

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

d. **返回值示例与说明**

e. **功能实现说明**

判断栏目是否为条件列表-zjxx\_column （栏目主表）listselect（为空代表不是条件列表），如果是调用刷新栏目条件列表信息缓存函数来刷新栏目条件列表信息函数，刷新栏目条件列表信息缓存，不是：刷新栏目列表信息函数和刷新栏目信息函数，刷新栏目信息缓存。

* + 1. 清除栏目列表信息缓存

1. **接口url:**

pub/admin/column/manageColumn/removeColumnListCache.do

**b.接口说明:**

**c. 接口定义：**

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

d. **返回值示例与说明**

e. **功能实现说明**

判断栏目是否为条件列表-zjxx\_column （栏目主表）listselect（为空代表不是条件列表），如果是调用清空栏目条件列表信息函数，刷新栏目条件列表信息缓存，不是：清空栏目信息函数，刷新栏目信息缓存。

* 1. 函数
     1. **“刷新条件列表栏目sql”函数**

**a.函数url:**

pub/admin/column/manageColumn/refreshConditionlistSqlFun

**b.接口说明：**

**刷新条件列表 sql的本质是在 缓存表中删除对应的数据,然后当前端获取时,可以重新生成数据**

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| refreshConditionlistSqlFun | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| classid |  | | 栏目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": "删除栏目成功" //返回的信息；

}

}

**e.功能实现说明**

1. 删除条件列表栏目 sql 缓存表中 classid=前端传递的 classid 的数据
   * 1. **“刷新条件列表信息缓存”函数**

**a.函数url:**

pub/admin/column/manageColumn/refreshConditionlistCacheFun

**b.接口说明：**

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| refreshConditionlistCache | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| classid |  | | 栏目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": "删除栏目成功" //返回的信息；

}

}

1. **功能实现说明** 
   1. **通过 classid, 在条件列表信息缓存表中读取相关数据,然后遍历每一条数据,将里面的 sql 取出,运行后得到数据,转成 json 字符串后,存入到这条信息的 json 字段中**
      1. **“清除栏目条件列表信息缓存”函数**

**a.函数url:**

pub/admin/column/manageColumn/removeConditionlistCacheFun

**b.接口说明：**

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| removeConditionlistCacheFun | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| classid |  | | 栏目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": "删除栏目成功" //返回的信息；

}

}

1. **功能实现说明** 
   1. **通过 classid,在条件列表信息缓存表中, 删除相关的数据**
      1. **“刷新栏目sql缓存”函数**

**a.函数url:**

pub/admin/column/manageColumn/refreshColumnSqlFun

**b.接口说明：**

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| refreshColumnSqlFun | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| classid |  | | 栏目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": "删除栏目成功" //返回的信息；

}

}

**e.功能实现说明**

通过 classid, 找到栏目对象,获取模型 id, 再获取模型对象,

从模型对象中获取前端列表项,后端列表项,前端内容项,排序项,

生成fields 子句和on 子句(以用户列表 fields 子句和用户列表 on 子句为例): 遍历前端列表项中的字段名,以表名.字段名的形式存入fields子句,遍历期间发现有多表关联则存入 on 子句(注意 on 子句的开头是本表表名)

所有的 where 子句设置为 1=1

遍历排序项,生成 order 子句

将 on,where,order 子句组合成 countsql, 并执行一遍,得到总条数

最后将这些数据存入数据库

* + 1. **“刷新栏目列表信息缓存”函数**

**a.函数url:**

pub/admin/column/manageColumn/refreshColumnListCacheFun

**b.接口说明：**

1具体逻辑 吴佳玮补充

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| refreshColumnListCacheFun | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| classid |  | | 栏目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": "删除栏目成功" //返回的信息；

}

}

**e.功能实现说明**

通过栏目 id, 在栏目列表信息缓存表中获取相关信息,遍历,取出 sql, 执行后将结果转为 json 字符串,存入该信息的 json 字段中

* + 1. **“清除栏目列表信息缓存”函数**

**a.函数url:**

pub/admin/column/manageColumn/removeColumnListCacheFun

**b.接口说明：**

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| removeColumnListCacheFun | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| classid |  | | 栏目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": "删除栏目成功" //返回的信息；

}

}

**e.功能实现说明**

**根据栏目 id, 在栏目列表信息缓存表中删除相关信息**

* 1. 暂不解决接口
     1. **“清除条件列表信息缓存”函数**
     2. **“清除栏目sql缓存”函数**
     3. **刷新栏目列表信息接口**

**a.接口url:** pub/admin/column/manageColumn/refreshColumnListCache.do

**b.接口说明:**

根据栏目定义判断是否是条件列表，如果是，调用“刷新条件列表信息缓存”，如果不是，调用“刷新栏目列表信息缓存”函数

例如：如果是新增或修改了栏目内的列表信息，则调用该接口将列表信息刷新

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| refreshColumnListCache | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| classid | String[] | | 栏目id | |

* + 1. **清除栏目sql缓存接口**

**a.接口url:**

pub/admin/column/manageColumn/deleteColumnSQLCache.do

**b.接口说明:**

清除该栏目SQL缓存

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteColumnSQLCache | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| classid | String[] | | 栏目id | |

d. **返回值示例与说明**

e. **功能实现说明**

* + 1. **清除栏目列表信息缓存接口**

**a.接口url:**

pub/admin/column/manageColumn/deleteColumnListCache.do

**b.接口说明:**

清除该栏目列表信息SQL缓存

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| deleteColumnListCache | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| classid | String[] | | 栏目id | |

d. **返回值示例与说明**

e. **功能实现说明**

* + 1. 栏目转换信息接口（信息转栏目、其他）

**a.接口url:**

pub/admin/column/manageColumn/transferToInfo.do

**b.接口说明:**

将栏目或其他数据表（八大类）转化为信息表数据。

相关说明：

1. 该栏目下新增的信息将自动转化为勾选的信息表数据（栏目、人员、角色、标签、数据词典、机构），即该信息具有多重性质

2. 转化为栏目时可以选择（父栏目、系统模型，父id）

3.新增的信息依旧在该栏目下，本质上是在栏目表内复制了一条具有id、title的信息

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| transferToInfo | Result | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| id | String | 信息id |
| formType | Int | 信息来源类别：   1. 栏目表 2. 机构表 3. 数据词典表 4. 角色表 5. 人员表 6. 标签表 |
| Infojson | Json | **1:转栏目**  { "classid": "abcd34fr",  "isshow": "1",  "mid": "1" }  **2:转机构**  { "orgid": "abcd34fr" }  **3:转数据词典**  { "did": "abcd34fr" }  **4:转角色**  { "rid": "abcd34fr" }  **5:转人员**  { "orgid": "abcd34fr"}  **6:转标签**  { } |

**d.返回值示例与说明**

{

"result": { //结果

"msg": "转换成功", //信息

"flag": "1" //标识

}

}

**e.功能实现说明**

1. 直接
2. 返回方法传出的返回值
   * 1. **批量修改栏目接口**
     2. **批量设置栏目信息转换接口**

**a.接口url:**

pub/admin/ column/manageColumn/。。。.do

**b.接口说明:**

将栏目批量转换为信息数据

该栏目不变动，实质是在转化父栏目中复制信息（该信息的id、title为转化栏目的id、title）

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| infoTransferTocolumn | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| classids[] | String | | 栏目的id字符串数组 | | |
| classnames[] | String | | 栏目的名字字符串数组 | | |
| fid | String | | 目标栏目id | | |

**d.返回值示例与说明**

{

"flag": "1", //返回值标志

"msg": "转移成功", //返回值信息

}

**e.功能实现说明**

① 接收参数classids[]、classnames[]、fid，判断有任意一个空值则返回“来源错误”。

② 遍历数组classids[]、classnames[]，每次遍历得到classid、classname

③ 根据classid获取来源数据，若找不到数据则同样返回“来源错误”。

④ 获取来源字段fid，获取栏目对应的系统模型。

⑤ 根据系统模型获得对应的信息表的表名tbname。

⑥ 根据参数classid、classname，将classid作为信息表id，将classname作为信息表title，将这条信息插入信息表tbname。

* + 1. **终极栏目转化接口**

**a.接口url:**

pub/admin/ column/manageColumn/。。。.do

**b.接口说明:**

终极栏目与非终极栏目相互转换，

终极栏目如何转为非终极栏目-转换前提：该终极栏目下面没有信息数据方可顺利进行转换；如该栏目下已建立信息，则须先将当前栏目的信息数据转移，再进行栏目属性转换，否则会出现冗余数据。

非终极栏目如何转换为终极栏目-转换前提：该非终极栏目下面不能有子栏目，方可顺利进行转换

**c. 接口定义：**

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

**d.返回值示例与说明**

{

"flag": "1", //返回值标志

"msg": "转化成功", //返回值信息

}

**e.功能实现说明**

1. 接收参数classid空值则返回“来源错误”。
2. 参数classid非空则根据该参数查询栏目表数据，根据查询出来的数据判断该栏目是否是终极栏目
3. 如果该栏目是终极栏目,查询该栏目下是否有信息数据：
4. 如果有则返回result，设置msg=“转化失败，请先将该栏目下信息转移或备份后清空！ ”， 设置 flag=0.
5. 如果没有则修改该栏目为非终极栏目，返回result，设置msg=“转化成功！ ”，设置 flag=1.
6. 如果该栏目是非终极栏目，根据查询出来的数据再判断该栏目有无子栏目：
7. 如果有，设置msg=“转化失败，转化的栏目不能含有子栏目，！ ”，设置 flag=0.
8. 如果没有则修改该栏目为终极栏目，返回result，设置msg=“转化成功！ ”，设置flag=1;
   * 1. **修改栏目选项设置接口**

**a.接口url:**

pub/admin/column/manageColumn/updateColumnOptSet.do

**b.接口说明：**

配置栏目权限

相关说明：

1. 超级管理员设置栏目权限是对于平台。(配置权限的用户组（即部门、角色）是整个平台的用户组)
2. 机构管理员设置栏目权限是对于本机构（配置权限的用户组是本机构的用户组）.
3. 全部可见：放开权限，所有机构均有权限访问
4. 用户组：调用查找本机构下的用户组接口获取

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| updateColumnOptSet | | Result | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| classsid |  | | 栏目id | |
| infoJson |  | | id:栏目ID,  showclass:是否显示到导航,  isshow:是否显示,  is\_zt:0, 预定义字段  qadddeps:可投稿部门用户,  qaddroles:可投稿角色用户,  qaddwhitelist:白名单（投稿）,  qaddblacklist:黑名单（投稿）,  seedeps:可浏览用部门户组,  seeroles:可浏览用角色户组,  seewhitelist:白名单（查看）,  seeblacklist:黑名单（查看）,  updatedeps:可修改部门用户,  updaterides:可修改角色用户,  updatewhitelist:白名单（修改）,  updateblacklist:黑名单（修改）,  cgtoinfo:栏目访问权限应用于信息,  checkqadd:前台投稿是否要审核,  qeditchecked:修改投稿是否需要审核, | |

d. **返回值示例与说明**

e. **功能实现说明**

通将上传的json数据放到选项设置表，根据参数classid修改栏目表中id字段值为参数classid值的数据，修改其值为对应infojson中的数据

注：Infojson是拼接的json字符串

* + 1. **修改栏目模板接口**
    2. **查询栏目模板接口**

**a.接口url:**

pub/admin/ column/manageColumn/findByTempid.do

**b.接口说明：**

传入模板的id，查询有没有栏目绑定该模板

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| FindByTempid | | Infor | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| listtempid | String | | 列表模板id | |
| covertempid | String | | 封面模板id | |
| contenttempid | String | | 内容模板Id | |

**d.返回值示例与说明**

{

"total": 8, //当前页信息条数;

"infos": [ //信息;

{

"classid": "sk", //栏目id

"classname": "sf", //栏目名

"onclick": "sfasf", //访问量

"myorder": "sfdf", //顺序

"onclick": "", //点击数

"islast": "", //是否为终极栏目

"bname": "", //栏目别名

"columntype": "", //栏目类型

"isgd":"", //是否归档

"isinfo": "", //是否自动转换为信息

"showclass": "", //是否显示到导航

"openadd": "", //是否开放投稿

"columnclass":"", //栏目页模式

"showdt":"", //内容页模式

"isflash":"", //内容条数增删时，列表页是否自动刷新

"maxnum":"", //显示总记录数

"lencord":"", //每页显示信息数

"linknum":"", //相关链接数

"classtype": "", //栏目文件扩展名

"filetype": "", //内容页文件扩展名

"author":"", //作者

"top": "", //是否置顶

"downnum": "", //每行下载地址数

"onlinenum": "", //每行在线观看地址数

"addinforfen":"", //前台投稿加点数

"listdt": "", //列表模式

"doctime": "", //信息归档时间

"stpnum": "", //生成静态页数

"addreinfor": "", //发布信息生成内容页

"modid": "", //为所属系统模型ID

"isshow": "", //是否显示

“pid”：“”， //父栏目id

"openpl":"", //是否开启评论

"plmodids":"", //评论系统模型id集合

"orgid":"", //所属机构id

"trainfo":"", //转换为信息

"columnid": "" //自动转换信息栏目

},

“result” : { //结果集;

"flag": "1", //标志，0为成功，1为操作失败

"msg": "有栏目绑定" //返回的信息；

}

]

}

**e.功能实现说明**

1. 创建Resultd对象。
2. 根据传入的参数查询绑定了模板的栏目id和传入的模板id相一致的栏目
3. 如果listtempid不为空时拼接查找条件，查询栏目表中列表模板id为传入的listtempid的模板
4. 如果covertempid不为空时拼接查找条件，查询栏目表中列表模板id为传入的covertempid的模板
5. 如果contenttempid不为空时拼接查找条件，查询栏目表中列表模板id为传入的contenttempid的模板

3）判断有没有找到栏目，根据查找情况设置infor信息病房result

* + 1. **查找关联字段**

**a.接口url:**

pub/admin/ column/manageColumn/ findRelatedFields.do

**b.接口说明：**

查找关联字段

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| findRelatedFields | | Info | | Json |
| 参数说明： | | | | |
| 参数名称 | 参数类型 | | 备注 | |
| modid | String | | 模型id | |

**d.返回值示例与说明**

{

“infos”：

[

{

"field": "zjxx\_a2" //关联字段

},

{

"field": "aa" //关联字段

}]

}

**e.功能实现说明**

1. 根据modid查询这个模型属于哪张表。
2. 搜索这张表每个字段的关联字段和这张表的分表中的关联字段。把这些关联字段放到list<map>中，一个关联字段一个map，（其中map的key值为field）。
3. 用递归的方法查找这张表的父表和父表的分表，直到这张表是父表为空。把这些父表和父表的分表中的关联字段写到list<map>中，一个关联字段一个map，（其中map的key值为field）。
4. 返回info。
   * 1. **栏目归档接口**
     2. **设置栏目信息转换规则接口**
     3. **模板选项**