目录

[说明：接口都要增加机构id和人员id作为参数 2](#_Toc494025735)

[1 信息模块接口 2](#_Toc494025736)

[1.1 接口 2](#_Toc494025737)

[1.1.1 查询前端列表接口 2](#_Toc494025738)

[1.1.2 查询后台列表接口 4](#_Toc494025739)

[1.1.3 查询条件列表接口 6](#_Toc494025740)

[1.1.4 查询条件列表条件集合接口 6](#_Toc494025741)

[1.1.5 查询前端内容信息数据接口 6](#_Toc494025742)

[1.1.6 查询后台内容信息数据接口 8](#_Toc494025743)

[1.1.7 前端增加/修改信息数据接口 8](#_Toc494025744)

[1.1.8 前端增加/修改信息栏目数据接口 8](#_Toc494025745)

[1.1.9 后台增加/修改信息数据接口 8](#_Toc494025746)

[1.1.10 删除信息接口 8](#_Toc494025747)

[1.1.11 威客列表查询接口 8](#_Toc494025748)

[1.1.12 威客信息查询接口 10](#_Toc494025749)

[1.1.13 信息转换栏目接口 11](#_Toc494025750)

[1.1.14 信息转换数据词典接口 12](#_Toc494025751)

[1.1.15 信息转换角色接口 12](#_Toc494025752)

[1.1.16 信息转换部门接口 12](#_Toc494025753)

[1.1.17 信息转换人员接口 13](#_Toc494025754)

[1.1.18 增加阅读记录接口 14](#_Toc494025755)

[1.1.19 查询阅读记录接口 15](#_Toc494025756)

[1.1.20 获取未读条数接口 17](#_Toc494025757)

[1.1.21 增加点击量接口 18](#_Toc494025758)

[1.1.22 获取信息点击量接口 19](#_Toc494025759)

[1.2 旧接口（或暂不解决） 20](#_Toc494025760)

[1.2.1 查询信息修改详情接口（旧版） 20](#_Toc494025761)

[1.2.2 查询信息详情接口（旧版） 21](#_Toc494025762)

[1.2.3 查询信息列表接口（旧版） 22](#_Toc494025763)

[1.2.4 删除信息接口（旧版） 24](#_Toc494025764)

[1.2.5 增加/修改信息接口（旧版） 25](#_Toc494025765)

[1.2.6 信息路径刷新接口 26](#_Toc494025766)

[1.2.7 增加威客信息接口（这个删除，在修改信息时会增加威客信息） 26](#_Toc494025767)

[1.2.8 信息转移/复制接口 26](#_Toc494025768)

[1.2.9 信息转栏目(其他)接口 27](#_Toc494025769)

[1.2.10 批量信息（根据给定id集）归档接口 28](#_Toc494025770)

[1.2.11 批量信息（根据给定时间范围）归档接口 28](#_Toc494025771)

[1.2.12 批量信息修改接口 29](#_Toc494025772)

[1.3 函数 30](#_Toc494025773)

[1.3.1 GetFieldItem 30](#_Toc494025774)

[1.3.2 “增加威客信息”函数 30](#_Toc494025775)

[1.3.3 “信息转栏目”函数 30](#_Toc494025776)

[1.4 对象 30](#_Toc494025777)

说明：接口都要增加机构id和人员id作为参数

**信息的权限设置主要针对的是前端用户。**

**后台管理员的信息权限—菜单+按钮+栏目设置+机构，能看到信息+有操作按钮=就能操作**

1. 信息模块接口
   1. 接口
      1. 查询前端列表接口

**a.接口url:**

pub/admin/information/Infohtml/findFrontInfoTempList.do

**b.接口说明:**

该接口用于查询对应栏目下的信息数据前端列表。

相关信息：

1. 是否是动态信息或静态信息从栏目配置信息中获取
2. 先查询信息列表所在栏目的栏目页模式设置，看是否是缓存信息

缓存信息：从缓存中获取数据

动态信息：

a.如果是动态信息，则再查询系统设置中动态内容页变量缓存 :是否有值，若无值，则每次都是拿最新信息

b.若系统设置中有值则查询栏目表 该栏目的updatetime:若updatetime+动态内容页变量缓存<当前时间，则拿最新信息，并更新updatetime的值，将本次信息存入缓存；若updatetime+动态内容页变量缓存>当前时间，则拿缓存信息

1. 权限从权限缓存中获取
2. 信息相关权限：信息选项设置中“可浏览用户组”、栏目选项设置中“可访问用户、信息机构开关”
3. 每次返回数据条数来源（每页信息总数）可能有：客户端上传>栏目设置>系统设置>默认值
4. 系统设置中“访问控制设置”有ip限制
5. 信息字段：系统模型的前端列表项+栏目列表接口用户字段
6. 信息排序：客户端上传>栏目排序设置>系统默认值
7. 搜索时间逻辑：

if(startTime!=0&&startTime!=""&&startTime!=null){

if(endtime!=0&&endtime!=""&&endtime!=null){

startTime<newstime<endtime

}else{

newstime>startTime

}

}else{

if(endtime!=0&&endtime!=""&&endtime!=null){

newstime<endtime

}else{

不用查询时间

}

}**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| findFrontInfoTempList | Info | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| userid | String | 用户id |
| orgid | String | 机构id |
| pid | String | 父信息id |
| page | String | 当前页码 |
| num | String | 每页显示信息条数 |
| orderType | String | 排序类型 |
| tablechange | String | gd表示查询归档表 |
| searchAttr | String | 搜索方法的属性，如：  1为置顶、2为推荐、3为头条、4为未审核、5为已审核、6为投稿、7为签发、8为我的信息 |
| searchValue | String[] | 搜索方法的关键字数组 |
| searchField | String[] | 搜索方法的字段数组，即该表包含的字段，如：  标题、机构、副标题等。 |
| startTime | Bigint | 搜索开始时间-newstime |
| endTime | Bigint | 搜索结束时间- newstime |

**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": 1, //当前页信息数

"infos": [ //信息

{

……

},

{

……

}

]

}

**e.功能实现说明**

1. 检查参数userid、orgid和classid:
2. 先检查orgid，若上传参数orgid为空，则调用获取平台机构id函数获取机构id，如果也为空，返回flag=0,msg=“来源错误”。再检查userid和classid，若任一为空则返回flag=0,msg=“来源错误”。
3. 通过classid查询栏目表，获取栏目信息机构开关设置，获取栏目信息列表的限制条件。
4. 一次检测searchAttr、searchValue、searchField、searchTime、orderType参数（非空值、则根据对应数值拼接sql查询条件），检查num和page的值, 每次返回数据条数来源（每页信息总数）可能有page值：客户端上传>栏目设置>系统设置>默认值，拼接limit内容。
5. 根据参数clasid从zjxx\_cache\_classsql栏目sql缓存表获取前端列表项相关sql语句，加上3和4拼接成完整的sql语句。
6. 先查询信息列表所在栏目的栏目页模式设置，看是否是缓存信息

缓存信息：从缓存中获取数据

动态信息：

a.如果是动态信息，则再查询系统设置中动态内容页变量缓存 :是否有值，若无值，则每次都是拿最新信息

b.若系统设置中有值则查询栏目表 该栏目的updatetime:若updatetime+动态内容页变量缓存<当前时间，则拿最新信息，并更新updatetime的值，将本次信息存入缓存；若updatetime+动态内容页变量缓存>当前时间，则拿缓存信息

1. 根据5中的缓存设置，并根据其设置采取相应方式访问数据
2. 如果是拿缓存，则根据其classid和组好的sql的MD5值去缓存表查询是否有缓存数据，如果缓存表没有数据，则调用拿表数据方式，获取数据并设置缓存表数据。
3. 如果是拿表数据，执行生成好的sql语句、查询并返回数据 infos:[], 最后更新栏目对象中的updatetime值，根据4中的类栏目缓存设置更新相应缓存。
   * 1. 查询后台列表接口

**a.接口url:**

pub/admin/information/Infohtml/findInfoTempList.do

**b.接口说明:**

该接口用于查询对应栏目下的信息数据列表。

注意：

1. 权限从权限缓存中获取
2. 信息相关权限：栏目选项设置中“信息机构开关”
3. 每次返回数据条数(每页显示数据总数)来源可能有：栏目设置>系统设置
4. 信息字段：系统模型的管理员列表项+栏目列表接口管理员字段
5. 信息排序：栏目排序设置>系统默认值
6. 搜索时间逻辑：

if(startTime!=0&&startTime!=""&&startTime!=null){

if(endtime!=0&&endtime!=""&&endtime!=null){

startTime<newstime<endtime

}else{

newstime>startTime

}

}else{

if(endtime!=0&&endtime!=""&&endtime!=null){

newstime<endtime

}else{

不用查询时间

}

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| findInfoTempList | Info | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| userid | String | 用户id |
| orgid | String | 机构id |
| pid | String | 父信息id |
| page | String | 当前页码 |
| num | String | 每页显示信息条数 |
| orderType | String | 排序类型 |
| tablechange | String | gd表示查询归档表 |
| searchAttr | String | 搜索方法的属性，如：  1为置顶、2为推荐、3为头条、4为未审核、5为已审核、6为投稿、7为签发、8为我的信息 |
| searchValue | String[] | 搜索方法的关键字数组 |
| searchField | String[] | 搜索方法的字段数组，即该表包含的字段，如：  标题、机构、副标题等。 |
| startTime | Bigint | 搜索开始时间 |
| endTime | Bigint | 搜索结束时间 |

**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": 1, //当前页信息数

"infos": [ //信息

{

……

},

{

……

}

]

}

**e.功能实现说明**

1. 检查参数userid、orgid和classid:
2. 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，如果也为空，返回flag=0,msg=“来源错误”。再检查userid和classid，若任一为空则返回flag=0,msg=“来源错误”。三者都非空则通过classid找到栏目对象，
3. 通过classid查询栏目表，获取栏目信息机构开关设置，获取栏目信息列表的限制条件。
4. 如果有权限，一次检测searchAttr、searchValue、searchField、searchTime参数（非空值、则根据对应数值拼接sql查询条件），检查num和page的值, 每次返回数据条数来源（每页信息总数）可能有page值：客户端上传>栏目设置>系统设置>默认值，拼接limit内容。
5. 根据参数clasid从zjxx\_cache\_classsql栏目sql缓存表获取后台列表项sql语句，加上3和4拼接成完整的sql语句。
6. 执行生成好的sql语句、查询并返回数据 infos:[],
   * 1. 查询条件列表条件集合接口

**a.接口url:**

pub/admin/information/Infohtml/findTempInfo.do

**b.接口说明:**

根据当前上传当前条件为参数，生成可能条件集合

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| findTempInfo | Info | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| id | String | 信息id |
| method | String | method为”modify”时为修改信息查找，method为“show”时为显示信息时查找信息 |

**d.返回值示例与说明**

//信息

{

“infos”：{……}

}

**e.功能实现说明**

* + 1. 查询前端内容信息数据接口

**a.接口url:**

pub/admin/information/Infohtml/findFrontTempInfo.do

**b.接口说明:**

该接口用于查询对应栏目下的信息前端内容详情。

注意：

1. 权限从权限缓存中获取
2. 信息相关权限：信息选项设置中“可浏览用户组”、栏目选项设置中“可访问用户、信息机构开关” 信息详情暂不考虑栏目设置
3. 系统设置中“访问控制设置”有ip限制
4. 系统模型的前端内容项+栏目内容接口用户字段

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| findFrontTempInfo | Info | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| id | String | 信息id |
| userid | String | 用户id |
| orgid | String | 机构id |
| method | String | method为”modify”时为修改信息查找，method为“show”时为显示信息时查找信息 |

**d.返回值示例与说明**

//信息

{

“infos”：{……}

}

**e.功能实现说明**

1. 检查参数userid、orgid、id和classid:
2. 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，如果也为空，返回flag=0,msg=“来源错误”。再检查userid和classid、id，若任一为空则返回flag=0,msg=“来源错误”。三者都非空则通过classid找到栏目对象，
3. 通过classid，userid、id查询栏目权限缓存表和信息权限表，判断权限。如果数据为空：通过classid，userid、id查询栏目对象中的权限配置和信息权限表；进行该用户进行权限判断，如果无权限，返回flag=0,msg=“无访问权限”，并在栏目权限缓存表创建相应的栏目权限缓存
4. 根据参数clasid从zjxx\_cache\_classsql栏目sql缓存表获取类容列表项sql语句
5. 执行生成好的sql语句、查询并返回数据 infos:[],
   * 1. 查询后台内容信息数据接口

**a.接口url:**

pub/admin/information/Infohtml/ findInfoTemp.do

**b.接口说明:**

该接口用于查询对应栏目下的信息后台内容详情。

注意：

1. 权限从权限缓存中获取

信息相关权限：栏目选项设置中“信息机构开关”信息详情暂不考虑栏目设置

1. 系统模型的管理员输入项

**c.接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| findInfoTemp | Info | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| id | String | 信息id |
| userid | String | 用户id |
| orgid | String | 机构id |

**d.返回值示例与说明**

//信息

{

“infos”：{……}

}

**e.功能实现说明**

1. 检查参数userid、orgid、id和classid:

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，如果也为空，返回flag=0,msg=“来源错误”。再检查userid和classid、id，若任一为空则返回flag=0,msg=“来源错误”。

1. 根据classid找到栏目信息中的系统模型mid
2. 根据通过mid找到系统模型信息中的表名（tbname）以及管理员输入项
3. 分解管理员输入项重新组合成字段拼接字符串：字段1,字段2,字段3
4. 执行拼接好的sql语句，返回数据
   * 1. 查询后台内容信息所有数据接口

**a.接口url:**

pub/admin/information/Infohtml/findInfoTempAllField.do

**b.接口说明:**

该接口用于查询对应栏目下的信息后台内容详情包括所有字段。

**c.接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| findInfoTempAllField | Info | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| id | String | 信息id |
| userid | String | 用户id |
| orgid | String | 机构id |

**d.返回值示例与说明**

//信息

{

“infos”：{……}

}

**e.功能实现说明**

1. 逻辑同1.1.5接口，查询数据的所有字段，不用获取管理员输入项。
2. Select \* from xxxx where id=xxxx
   * 1. 前端增加/修改信息数据接口

**a.接口url:**

pub/admin/information/Infohtml/updateFrontInfo.do

**b.接口说明:**

该接口用于前端增加修改对应栏目下的信息内容详情。

注意点：

1. 系统设置-用户设置-前端用户设置：禁止 IP提交
2. 系统设置-用户设置-前端用户设置：允许 IP提交
3. 系统设置-网络防火墙：屏蔽提交敏感字符
4. 系统模型的用户输入项+必填项
5. 通过栏目信息确定是否开启威客功能，如果开启，每次修改都要添加威客信息。（在栏目里设置）。
6. 是否有提交权限，由栏目选项设置中“可投稿用户、可修改用户”+信息选项设置中“可修改用户”。
7. 增加信息时：栏目转换设置中“信息转换栏目”是否勾选，判断信息是否需要转换。若需要转换，逻辑走转换分支。
8. 修改信息时，需判断信息设置中的修改权限
9. 关联表字段：field—模型名称|字段名，value：值

**c.接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| updateFrontInfo | Info | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| ~~id~~ | ~~String~~ | ~~信息id~~ |
| userid | String | 用户id |
| orgid | String | 机构id |
| InfoJson |  |  |
| modelname | String | 模型名称 |

**d.返回值示例与说明**

//信息

{

“infos”：{……}

}

**e.功能实现说明**

1. 先从系统设置中查询相关配置：用户设置-前端用户设置：、禁止 IP提交、允许 IP提交 、网络防火墙：屏蔽提交敏感字符 检查，判断是否与系统设置匹配，如果不匹配，设置flag为0，msg为不匹配信息。
2. 系统设置中相关字段参数，先在内存中查找，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
3. 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，如果也为空，返回flag=0,msg=“来源错误”。
4. 检查参数userid、classid；任何一个为空，返回错误。
5. 通过userid，classid查找栏目权限缓存表：数据存在-获取对应数据，数据不存在-通过classid查找栏目表，获取对应数据对象，验证是否有提交权限，由栏目选项设置中“可投稿用户、可修改用户”+信息选项设置中“可修改用户”（修改信息时才需要判断）。没有权限，则返回错误
6. 通过栏目对象找到mid，通过mid找到系统模型，通过系统模型找到系统模型的用户输入项+必填项以及tbname。
7. 验证infojson中的参数是否符合系统模型中的用户输入项+必填项，不满足必填项（必填项无值）则返回错误，不属于用户输入项则过滤
8. 检查infojson中的参数id，若id 为空，则调用增加信息函数，若id不为空，则调用修改信息函数
9. 增加信息：生成1个uuid,赋值给 infojson中的参数id并将处理过的infojson组成sql存入tbname对应的表，返回uuid（如果栏目设置中设置了威客表，则将生成的数据存入威客表）
10. 修改信息：需判断信息设置中的修改权限
11. 修改信息：通过tbname找到对应表，并根据infojson中的参数id找到相应的那条数据，将上传的infojson组成sql修改该条数据对应的字段（如果栏目设置中设置了威客表，则同时生成威客表数据）
    * 1. 选项设置接口

**.接口url:**

pub/admin/information/Infohtml/setInfor.do

**b.接口说明:**

该接口用于增加修改信息时的选项设置。

注意：

**c.接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| setInfor | Info | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| id | String | 信息id |
| userid | String | 用户id |
| orgid | String | 机构id |
| istop |  | 置顶级别 |
| updatedeps | text | 可修改部门组 |
| updateroles | text | 可修改角色组 |
| updateblacklist | text | 可修改黑名单 |
| updatewhitelist | text | 可修改白名单 |
| seedeps | text | 可浏览部门组 |
| seegroles | text | 可浏览角色组 |
| seeblacklist | text | 可浏览黑名单 |
| seewhitelist | text | 可浏览白名单 |

**d.返回值示例与说明**

//信息

{

“infos”：{……}

}

**e.功能实现说明**

* + 1. 后台增加/修改信息数据接口

**a.接口url:**pub/admin/information/Infohtml/ updateInfo.do

**b.接口说明:**

该接口用于后台增加修改对应栏目下的信息内容详情。

注意点：

1. 系统模型的管理员输入项+必填项
2. 通过栏目信息确定是否开启威客功能，如果开启，每次修改都要添加威客信息。
3. 系统设置-用户设置-前端用户设置：禁止 IP提交
4. 系统设置-用户设置-前端用户设置：允许 IP提交
5. 系统设置-网络防火墙：屏蔽提交敏感字符
6. 增加信息时：栏目转换设置中“信息转换栏目”是否勾选，判断信息是否需要转换。若需要转换，逻辑走转换分支。
7. 关联表字段：field—模型名称|字段名，value：值。

**c.接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| updateInfo | Info | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| ~~id~~ | ~~String~~ | ~~信息id~~ |
| userid | String | 用户id |
| orgid | String | 机构id |
| InfoJson |  |  |
| modelname | String | 模型名称 |

**d.返回值示例与说明**

//信息

{

“infos”：{……}

}

**e.功能实现说明**

1. 先从系统设置中查询相关配置：用户设置-管理员设置：、禁止 IP提交、允许 IP提交 、网络防火墙：屏蔽提交敏感字符 检查，判断是否与系统设置匹配，如果不匹配，设置flag为0，msg为不匹配信息。
2. 系统设置中相关字段参数，先在内存中查找，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
3. 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，如果也为空，返回flag=0,msg=“来源错误”。
4. 检查参数userid、classid；任何一个为空，返回错误。
5. 通过栏目对象找到mid，通过mid找到系统模型，通过系统模型找到系统模型的用户输入项+必填项以及tbname。
6. 验证infojson中的参数是否符合系统模型中的管理员输入项+必填项，不满足必填项（必填项无值）则返回错误，不属于管理员输入项则过滤
7. 检查infojson中的参数id，若id 为空，则调用增加信息函数，若id不为空，则调用修改信息函数
8. 增加信息：生成1个uuid,复制给infojson中的参数id，并将处理过的infojson组成sql存入tbname对应的表，返回uuid（如果栏目设置中设置了威客表，则将生成的数据存入威客表）
9. 修改信息：通过tbname找到对应表，并根据infojson中的参数id找到相应的那条数据，将上传的infojson组成sql修改该条数据对应的字段（如果栏目设置中设置了威客表，则同时生成威客表数据）

”

* + 1. 删除信息接口

**a.接口url:**pub/admin/information/Infohtml/ deleteInfo.do

**b.接口说明:**

该接口用于删除对应栏目下的信息。

注意点：

1. 栏目选项设置中“可删除用户组”。

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| deleteInfo | Result | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| ids | String[] | 信息id数组 |
| userid | String | 用户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传入参数classid、ids，通过classid找到栏目对象，通过栏目对象找到mid，通过mid找到系统模型，通过系统模型找到tbname。

②遍历ids数组，将表名于id传入**同步删除方法**

* + 1. 威客列表查询接口

**a.接口url:**

pub/admin/information/Infohtml/findModifyRecordList.do

**b.接口说明:**

查询威客信息列表

注意点：同“栏目”信息查询权限，从栏目信息中获取

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| findModifyRecordList |  | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| id | String | 信息id（可为空） |
| classid | String | 栏目id |
| editer | String | 修改人 |
| title | String | 标题 |
| orderType | String | 排序类型 |
| searchValue | String[] | 搜索方法的关键字数组 |
| searchField | String[] | 搜索方法的字段数组，即该表包含的字段，如：  标题、机构、副标题等。 |
| orgid | String | 机构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": 1, //当前页信息数

"infos": [ //信息

{

……

},

{

……

}

]

}

**e.功能实现说明**

1. Id为空：查询该栏目下所有威客信息
2. Id不为空：查询该栏目下这个信息的威客信息
3. 参考后台列表接口逻辑
   * 1. 威客信息查询接口
4. **接口url:**

pub/admin/information/Infohtml/findModifyRecordById.do

**b.接口说明:**

注意点：同栏目信息查询权限，从栏目信息中获取

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| findModifyRecordById | Result | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| id | String | 威客id |
| classid | String | 栏目id |
| userid | String | 用户id |
| orgid | String | 机构id |

**d.返回值示例与说明**

{

"infos": [ //信息

{

……

},

{

……

}

]

}

**e.功能实现说明**

1. 参考后台内容信息接口逻辑
   * 1. 信息转换栏目接口

**a.接口url:**

pub/admin/information/Infohtml/transferinfo.do

**b.接口说明:**

将信息数据转换为栏目

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| transferinfo | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| id | String | | 信息的id | | |
| title | String | | 信息的名字 | | |
| pid | String | | 选择父栏目id | | |
| Ishide | String | | 是否隐藏栏目（1隐藏，0显示） | | |
| modelname | String | | 信息转换栏目时的系统模型 | | |
| showclass | String | | 是否显示到导航 | | |
| classid | String | | 信息所属的栏目id | | |
| orgid | String | | 栏目所属机构 | | |

**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. 如果参数id、title为空，mid和classid同时为空则返回“来源错误”。
2. 如果mid为空，则根据classid获取mid
3. 根据参数调用新增栏目逻辑，默认为终极栏目，这里的同级栏目title可以相同
4. 返回result。
   * 1. 增加阅读记录接口

**a.接口url:**

pub/admin/information/Infohtml/addReadingLog.do

**b.接口说明:**

（1）必须传入classid，infoid，peopleid、orgid 参数。

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| addReadingLog | Result | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| infoid | String | 信息id |
| peopleid | String | 人员id |
| orgid | String | 机构id |

**d.返回值示例与说明**

{

"result": { //结果

"msg": "成功", //信息

"flag": "1" //标识

}

}

**e.功能实现说明**

（1）根据classid，infoid，peopleid、orgid查询有无记录。

（2）若有，则证明已该人员已阅读过此条信息，返回已读过这条信息。

（3）若无，则zjxx\_readlog插入一条记录，返回已阅读。

* + 1. 查询阅读记录接口

**a.接口url:**

pub/admin/information/Infohtml/findReadingLogList.do

**b.接口说明:**

（1）单独传id或者传classid、infoid、peopleid、orgid参数，查询指定人的指定记录。

（2）传classid，查询指定栏目所有人的所有记录。

（3）传classid和infoid，查询指定信息所有人的记录。

（4）传orgid、peopleid，查询指定人员所有记录。

（5）传orgid、peopleid和classid，查询指定人的指定栏目的所有记录。

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| findReadingLogList | Infor | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| id | String | 阅读记录id |
| classid | String | 栏目id |
| infoid | String | 信息id |
| peopleid | String | 人员id |
| orgid | String | 机构id |

**d.返回值示例与说明**

{  
    "infos":[  
        {  
            "classid":"be68990e-6985-4b7e-9620-6d9c31c9e683",  
            "infoid":"64585518-23c5-4e85-89c6-0e08fa4cbcbd",  
            "peopleid":"9db7c86d-a6c2-4bd4-ad02-3dcc325f1b40",  
            "id":"f90a8006-605b-488d-ab9c-e425c4805184",  
            "readtime":1495093555  
        }  
    ],  
    "page":null,  
    "result":{  
        "flag":1,  
        "msg":"success",  
        "orgid":"",  
        "rid":"",  
        "token":"",  
        "ttime":"",  
        "userId":""  
    },  
    "total":1  
}

**e.功能实现说明**

（1）单独传id或者传classid、infoid、peopleid、orgid参数，查询指定人的指定记录。

（2）传classid，查询指定栏目所有人的所有记录。

（3）传classid和infoid，查询指定信息所有人的记录。

（4）传orgid、peopleid，查询指定人员所有记录。

（5）传orgid、peopleid和classid，查询指定人的指定栏目的所有记录。

* + 1. 获取未读条数接口

**a.接口url:**

pub/admin/information/Infohtml/getNoReadCount.do

**b.接口说明:**

（1）传classid、orgid 、peopleid，查询该人员指定栏目下有多少条未读消息,以及未读消息的id集合

**c. 接口定义：**

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

**d.返回值示例与说明**

{  
    "infos":[  
        {  
            "infoid":"1e1546fe-e1d2-4b23-9462-49f819b90abe"  
        },  
        {  
            "infoid":"5a825b38-bf4d-42e1-95b2-38d6edcc23f3"  
        },  
        {  
            "infoid":"8885f1d0-cb92-4178-b0ca-9604ffd3bf2d"  
        },  
        {  
            "infoid":"dd4d9c6b-4ce5-4584-aab0-6d9a347c3e20"  
        }  
    ],  
    "page":null,  
    "result":{  
        "flag":1,  
        "msg":"success",  
        "orgid":"",  
        "rid":"",  
        "token":"",  
        "ttime":"",  
        "userId":""  
    },  
    "total":4  
}

**e.功能实现说明**

（1）根据classid查询该栏目下有多少信息，把所有的id集合存放在list中。

（2）根据classid和orgid 、peopleid查询该人员阅读过该栏目下的多少信息，把阅读过的信息id存放在另一list中

（3）根据2个list统计总共有多少条未读信息，并把未读信息的id返回。

* + 1. 增加点击量接口

**a.接口url:**

pub/admin/information/Infohtml/clickAddOne.do

**b.接口说明:**

（1）传classid和id，增加信息的点击量。

**c. 接口定义：**

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

**d.返回值示例与说明**

{  
    "flag":1,  
    "msg":"成功"  
}

**e.功能实现说明**

（1）根据classid和id查询该条信息。

通过classid查找moduleid，通过moduleid查找对应表名

Select id， onclick from 表明 where id=id

（2）将信息的onclick字段+1 ，更新信息。

Update onclick+1

* + 1. 获取信息点击量接口

**a.接口url:**

pub/admin/information/Infohtml/getClickCount.do

**b.接口说明:**

（1）获取信息的点击量。

**c. 接口定义：**

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

**d.返回值示例与说明**

{  
    "infos":[  
        {  
            "onclick":1  
        }  
    ],  
    "page":null,  
    "result":{  
        "flag":1,  
        "msg":"查询成功",  
          },  
    "total":null  
}

**e.功能实现说明**

（1）根据classid和id查询该条信息。

（2）将信息的onclick字段返回。

* + 1. 添加或修改收藏接口

**a.接口url:**

pub/admin/information/Infohtml/addorupdatecollect.do

**b.接口说明:**

（1）增加、修改人员信息收藏接口。

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| addorupdatecollect | Infor | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| id | String | 收藏i |
| classid | String | 栏目id |
| infoid | String | 信息id |
| peopleid | String | 人员id |
| orgid | String | 机构id |

**d.返回值示例与说明**

{  
    "infos":[  
        {  
            "onclick":1  
        }  
    ],  
    "page":null,  
    "result":{  
        "flag":1,  
        "msg":"查询成功",  
          },  
    "total":null  
}

**e.功能实现说明**

id为空就是添加，不为空就是修改

* + 1. 查询人员列表收藏接口

**a.接口url:**

pub/admin/information/Infohtml/collectlist.do

**b.接口说明:**

（1）查询人员信息收藏列表接口。

**c. 接口定义：**

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

**d.返回值示例与说明**

{  
    "infos":[  
        {  
            "onclick":1  
        }  
    ],  
    "page":null,  
    "result":{  
        "flag":1,  
        "msg":"查询成功",  
          },  
    "total":null  
}

**e.功能实现说明**

* + 1. 删除人员收藏接口

**a.接口url:**

pub/admin/information/Infohtml/addorupdatecollect.do

**b.接口说明:**

（1）删除人员信息收藏接口。

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| addorupdatecollect | Infor | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| infoid | String | 信息id |
| peopleid | String | 人员id |
| orgid | String | 机构id |

**d.返回值示例与说明**

{  
    "infos":[  
        {  
            "onclick":1  
        }  
    ],  
    "page":null,  
    "result":{  
        "flag":1,  
        "msg":"查询成功",  
          },  
    "total":null  
}

**e.功能实现说明**

* + 1. 批量查询信息接口

注意：批量查询同一个栏目的信息

**a.接口url:**

pub/admin/information/Infohtml/batchFrontInfoTempList.do

**b.接口说明:**

（1）删除人员信息收藏接口。

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| batchFrontInfoTempList | Infor | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| userid | String | 用户id |
| orgid | String | 机构id |
| ids | String | 信息id集合 |
| page | String | 当前页码 |
| num | String | 每页显示信息条数 |
| orderType | String | 排序类型 |
| tablechange | String | gd表示查询归档表 |
| searchAttr | String | 搜索方法的属性，如：  1为置顶、2为推荐、3为头条、4为未审核、5为已审核、6为投稿、7为签发、8为我的信息 |
| searchValue | String[] | 搜索方法的关键字数组 |
| searchField | String[] | 搜索方法的字段数组，即该表包含的字段，如：  标题、机构、副标题等。 |
| startTime | Bigint | 搜索开始时间-newstime |
| endTime | Bigint | 搜索结束时间- newstime |

**d.返回值示例与说明**

{  
    "infos":[  
        {  
            
        }  
    ],  
    "page":null,  
    "result":{  
        "flag":1,  
        "msg":"查询成功",  
          },  
    "total":null  
}

**e.功能实现说明**

* + 1. 通过classid查询表前端信息列表

**a.接口url:**

pub/admin/information/Infohtml/findFrontInfoTempListByClassid.do

1. **接口说明:**

同1.1.1接口

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| findFrontInfoTempListByClassid | Info | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| userid | String | 用户id |
| orgid | String | 机构id |
| pid | String | 父信息id |
| page | String | 当前页码 |
| num | String | 每页显示信息条数 |
| orderType | String | 排序类型 |
| tablechange | String | gd表示查询归档表 |
| searchAttr | String | 搜索方法的属性，如：  1为置顶、2为推荐、3为头条、4为未审核、5为已审核、6为投稿、7为签发、8为我的信息 |
| searchValue | String[] | 搜索方法的关键字数组 |
| searchField | String[] | 搜索方法的字段数组，即该表包含的字段，如：  标题、机构、副标题等。 |
| startTime | Bigint | 搜索开始时间-newstime |
| endTime | Bigint | 搜索结束时间- newstime |

**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": 1, //当前页信息数

"infos": [ //信息

{

……

},

{

……

}

]

}

**e.功能实现说明**

1. 检查参数userid、orgid和classid:
2. 先检查orgid，若上传参数orgid为空，则调用获取平台机构id函数获取机构id，如果也为空，返回flag=0,msg=“来源错误”。再检查userid和classid，若任一为空则返回flag=0,msg=“来源错误”。
3. 通过classid查询栏目表，获取栏目信息机构开关设置，获取栏目信息列表的限制条件。
4. 一次检测searchAttr、searchValue、searchField、searchTime、orderType参数（非空值、则根据对应数值拼接sql查询条件），检查num和page的值, 每次返回数据条数来源（每页信息总数）可能有page值：客户端上传>栏目设置>系统设置>默认值，拼接limit内容。
5. 根据参数clasid从zjxx\_cache\_classsql栏目sql缓存表获取前端列表项相关sql语句，加上3和4拼接成完整的sql语句。
6. 先查询信息列表所在栏目的栏目页模式设置，看是否是缓存信息

缓存信息：从缓存中获取数据

动态信息：

a.如果是动态信息，则再查询系统设置中动态内容页变量缓存 :是否有值，若无值，则每次都是拿最新信息

b.若系统设置中有值则查询栏目表 该栏目的updatetime:若updatetime+动态内容页变量缓存<当前时间，则拿最新信息，并更新updatetime的值，将本次信息存入缓存；若updatetime+动态内容页变量缓存>当前时间，则拿缓存信息

1. 根据5中的缓存设置，并根据其设置采取相应方式访问数据
2. 如果是拿缓存，则根据其classid和组好的sql的MD5值去缓存表查询是否有缓存数据，如果缓存表没有数据，则调用拿表数据方式，获取数据并设置缓存表数据。
3. 如果是拿表数据，执行生成好的sql语句、查询并返回数据 infos:[], 最后更新栏目对象中的updatetime值，根据4中的类栏目缓存设置更新相应缓存。
   * 1. 通过classid查询表后台信息列表

**a.接口url:**

pub/admin/information/Infohtml/findInfoTempListByClassid.do

1. **接口说明:**

同1.1.2接口

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| findInfoTempListByClassid | Info | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| userid | String | 用户id |
| orgid | String | 机构id |
| pid | String | 父信息id |
| page | String | 当前页码 |
| num | String | 每页显示信息条数 |
| orderType | String | 排序类型 |
| tablechange | String | gd表示查询归档表 |
| searchAttr | String | 搜索方法的属性，如：  1为置顶、2为推荐、3为头条、4为未审核、5为已审核、6为投稿、7为签发、8为我的信息 |
| searchValue | String[] | 搜索方法的关键字数组 |
| searchField | String[] | 搜索方法的字段数组，即该表包含的字段，如：  标题、机构、副标题等。 |
| startTime | Bigint | 搜索开始时间 |
| endTime | Bigint | 搜索结束时间 |

**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": 1, //当前页信息数

"infos": [ //信息

{

……

},

{

……

}

]

}

**e.功能实现说明**

1. 检查参数userid、orgid和classid:
2. 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，如果也为空，返回flag=0,msg=“来源错误”。再检查userid和classid，若任一为空则返回flag=0,msg=“来源错误”。三者都非空则通过classid找到栏目对象，
3. 通过classid查询栏目表，获取栏目信息机构开关设置，获取栏目信息列表的限制条件。
4. 如果有权限，一次检测searchAttr、searchValue、searchField、searchTime参数（非空值、则根据对应数值拼接sql查询条件），检查num和page的值, 每次返回数据条数来源（每页信息总数）可能有page值：客户端上传>栏目设置>系统设置>默认值，拼接limit内容。
5. 根据参数clasid从zjxx\_cache\_classsql栏目sql缓存表获取后台列表项sql语句，加上3和4拼接成完整的sql语句。
6. 执行生成好的sql语句、查询并返回数据 infos:[],
   * 1. 审核，推荐，头条，置顶接口

注意：调用增加修改信息接口修改对应字段

* + 1. 批改试题接口

**a.接口url:**

pub/admin/answer/ correctingtest.do

**b.接口说明:**

（1）提交试题答案，后台判断答案并给分接口。

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| correctingtest | Infor | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| peopleid | String | 答题人id |
| orgid | String | 机构id |
| examid | String | 考试Id |
| testid | String | 题目id |
| stuanswer | String | 学生答案 |
| score | Float | 试题的总分 |
| modelname | String | 信息转栏目时的模型名称 |
| typecode | Integer | 试题类型（0：单选题。1：多选题。2：填空题）； |

**d.返回值示例与说明**

{

    "infos":[  
        {  
           "haspg": 1, 是否已经批改 1已经批改，0没有批改

"point": 0, 得分

"statemark": 0, 正确与否1正确，0错误

        }  
    ],  
    "page":null,  
    "result":{  
        "flag":1,  
        "msg":"查询成功",  
          },  
    "total":null  
}

**e.功能实现说明**

提交试题答案，后台判断答案并给分

直接通过试题id和人员id搜索人员试题表，判断是否存在记录，如果存在则继续，不存在则先添加记录并根据前端传的模型名称将信息转栏目；

首先获取到试题id，查询出这道题的信息，可得到此题是否为立即批改；

不是立即批改（0），则直接添加一条人员考试、试题记录，得分为0分，批改状态为未批改（0），其他字段信息正常返回给前端；

是立即批改（1），则获取到题目的类型，选择题或者多选题，直接比对学生

答案和正确答案（试题信息中有），相同则将得分字段的值添加为试题的总分（前端传），状态标识为正确（1）；不同则0分，状态标识为错误（0）；批改状态为已批改（1），其他字段信息正常返回给前端；

题型如果为填空题，则获取学生答案，根据英文逗号进行分割，获取到各个

空学生的答案，同时分割正确答案，进行对比，全部匹配则满分，部分匹配则部分分数。比如学生答案：1,2,3,5；正确答案：1,2,3,4；满分100分,则得分为75分。将分数存入，满分则状态标识为正确（1），不是满分则状态标识为错误（0），批改状态为已批改（1），其他字段信息正常返回给前端；

* + 1. 统计分数接口

**a.接口url:**

pub/admin/answer/ countscore.do

**b.接口说明:**

（1）查询考试总分的接口。

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| countscore | Infor | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| peopleid | String | 答题人id |
| orgid | String | 机构id |
| examid | String | 考试Id |
| paperid | String | 试卷id |
| userid | String | 登录人id |
| type | Integer | type(0代表是学生调用统计分数接口，1代表老师调用) |

**d.返回值示例与说明**

{  
    "infos":[  
        {  
             "haspg": 1, 是否已经批改 1已经批改，0没有批改

"point": 0, 统计分数

        }  
    ],  
    "page":null,  
    "result":{  
        "flag":1,  
        "msg":"查询成功",  
          },  
    "total":null  
}

**e.功能实现说明**

根据试卷id获取到是否为立即批改；  
立即批改（1），根据上传的classid，人员id和试卷id调用 通过classid查询后台信息列表去查询人员试题记录表中的试题得分相加并返回给前端。

不是立即批改（0），则获取type的值，值为0，说明是学生调用的这个接口，则直接返回已批改字段（未批改：0）返回给前端。

如果type的值为1，则代表是老师调用这个接口，和立即批改的逻辑是一样的， 相加分数统计后，将总分和批改状态字段传（1）代表已经批改，返回给前端 。

* + 1. 考试结果判定

**a.接口url:**

pub/admin/answer/ countgrade.do

**b.接口说明:**

（1）判定考试结果的接口。

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| countgrade | Infor | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| userid | String | 登录人id |
| orgid | String | 机构id |
| examid | String | 考试Id |
| peopleid | String | 答题人id |

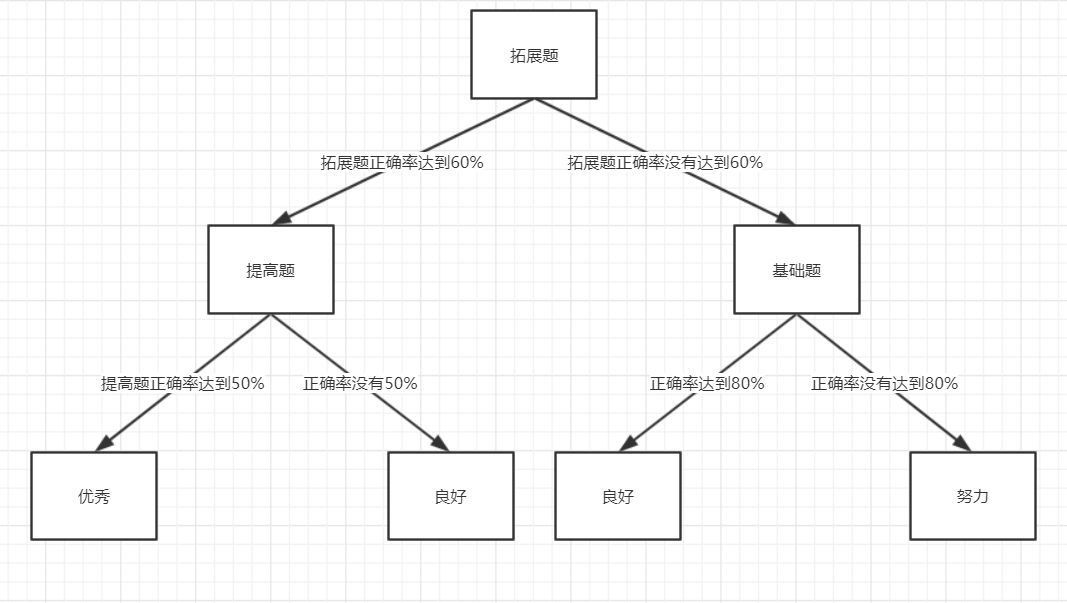
**d.返回值示例与说明**

{  
    "infos":[  
        {  
            " Haspg ":1（0是努力，1是良好，2是优秀）  
        }  
    ],  
    "page":null,  
    "result":{  
        "flag":1,  
        "msg":"查询成功",  
          },  
    "total":null  
}

**e.功能实现说明**

1. 布置作业时设置评分标准，示例如下：
2. 设置三个评分标准（百分比）分别给：基础题、拓展题、提高题；
3. 如：基础题（80%），拓展题（60%），提高题（50%）
4. 根据某个学生在本次作业中题目达标率进行评价，流程如下：
5. 每次考试完成批改的同时，根据下方流程进行评分计算。  
   以拓展题的计算为初始点，比如一张试卷中，共30道题，基础题10道，拓展题10道，提高题10道。每种类型的题目在布置作业时设置了评分基准（百分比制），如：基础题（80%（基础题做对8题）），拓展题（60%（拓展题做对6题）），提高题（50%（提高题做对5题））。
6. 先判断本次考试中当前学生的拓展题正确率，  
    本次作业拓展题正确率达到60%，则属于没有重大错误，进入提高题的判断，若提高题正确率达到50%，本次考试评价则为优秀，未到达则良好；

本次作业拓展题正确率未达到60%，则属于有重大错误，进入基础题的判断，若基础题正确率达到80%，则本次考试评价为良好，未达到则为努力。



* 1. 函数
     1. “增加威客信息”函数

**a.接口url:**

pub/admin/information/Infohtml/updateModifyRecord

1. **接口说明:**

威客添加逻辑：新增、修改信息时，将最新的信息数据同步添加的威客中。

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| updateModifyRecord | Result | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| inforobj |  | 信息对象 |
| userid | String | 修改用户id |
| tablename | String | 信息所在表名（可为空） |

**d.返回值示例与说明**

{

"infos": [ //信息

{

……

},

{

……

}

]

}

**e.功能实现说明**

判断tablename是否为空：

为空：根据inforobj（信息对象）中的classid找模型mid，在查到信息表名。

不为空：获取tablename为信息所在表名。

根据信息表表名，在对应的威客表中添加威客信息

* 1. 旧接口（或暂不解决）
     1. 信息转换数据词典接口

注意：targettable增加记录jp\_dictionary

* + 1. 信息转换人员接口

**a.接口url:**

infoTransfer.do

**b.接口说明:**

将信息数据转换为栏目

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| Transferinfo | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| id | String | | 信息的id | | |
| title | String | | 信息的名字 | | |
| tbname | String | | 转换目标的表名 | | |
| orgid | String | | 机构的id | | |
| btype | string | | 传入“人员” | | |
| origintable | String | | 来源表名字段 | | |
| Targettable | String | | 目标表名字段（多个表名以“，”隔开） | | |
| classid | String | | 信息所属的栏目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id、titl、orgid，判断有任意一个空值则返回“来源错误”。
2. 根据接收的参数拼接sql，向人员表中添加一条数据。
3. 再次拼接sql，向机构员工角色表中添加数据。例如;insert into 表名 (id,oid,rids,peid) values(‘’,’’,’’,’’),其中的oid就是参数orgid的值，peid即为参数id得值。
4. 根据id的值和通过classid获取的表名获取targettable的值，然后更新这个值（把原有的值附加新的表名用逗号隔开）。
5. 如果再次转换成其他机构下的人员，更新机构人员表的oid字段中间以，隔开。
6. 执行sql，往信息表里插入一条记录。
7. 返回result。
   * 1. 信息转换部门接口

**a.接口url:**

Transferinfo.do

**b.接口说明:**

注意：targettable增加记录role

**c. 接口定义：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口定义 | | | | | |
| 接口名 | | 对象名 | | | 返回值类型 |
| InfoTransfer | | Result | | | Json |
| 参数说明： | | | |  | |
| 参数名称 | 参数类型 | | 备注 | | |
| Id | String | | 信息id | | |
| title | String | | 信息的名字 | | |
| tbname | String | | 转换目标的表名(栏目的主表名称) | | |
| btype | String | | 转换的类型 | | |
| origintable | String | | 来源表名字段 | | |
| Targettable | String | | 目标表名字段（多个表名以“，”隔开） | | |
| classid | String | | 信息所属的栏目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**

**调用信息转换成其他的方法5.1.11**

* + 1. 信息转换角色接口

注意：targettable增加记录role

* + 1. 搜索信息接口

**a.接口url:**pub/admin/information/Infohtml/ searchInfo.do

**b.接口说明:**

该接口用于搜索对应栏目下的信息。

注意点：

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| searchInfo | Info | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| userid | String | 用户id |
| page | String | 当前页码 |
| num | String | 每页显示信息条数 |
| orderType | String | 排序类型 |
| tablechange | String | gd表示查询归档表 |
| searchAttr | String | 搜索方法的属性，如：  1为置顶、2为推荐、3为头条、4为未审核、5为已审核、6为投稿、7为签发、8为我的信息 |
| searchValue | String[] | 搜索方法的关键字数组 |
| searchField | String[] | 搜索方法的字段数组，即该表包含的字段，如：  标题、机构、副标题等。 |
| searchTime | String | 搜索方法的时间，如：  1天、2天、一周、1个月、3个月、6个月、1年。 |

**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": 1, //当前页信息数

"infos": [ //信息

{

……

},

{

……

}

]

}

**e.功能实现说明**

1. 检查参数classid、为空则返回信息“来源错误”。非空则通过classid找到栏目对象，通过栏目对象找到mid，通过mid找到系统模型，通过系统模型找到tbname。
2. 根据tbname找到相关分表
3. 如果tablechange非空，则根据tablechange修改各表名至对应归档表
4. 一次检测searchAttr、searchValue、searchField、searchTime参数，非空值、则根据对应数值拼接sql查询条件
5. 若orderType非空则拼接order by排序内容
6. 根据num和page拼接limit内容。若为空值则使用默认值
7. 执行生成好的sql语句、查询并返回数据
   * 1. 查询信息修改详情接口（旧版）

**a.接口url:**

pub/admin/information/Infohtml/findUpdateInfo.do

**b.接口说明:**

该接口用于后台修改信息数据前，查询对应栏目下的要修改的信息数据。

**c. 接口定义：**

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

**d.返回值示例与说明**

//信息

{

“infos”：{……}

}

**e.功能实现说明**

1. 接收参数id、classid，有任意一个空值则返回“来源错误”
2. 根据classid找到栏目信息中的系统模型mid
3. 根据通过mid找到系统模型信息中的表名（tbname）以及录入项（enter）
4. 分解enter重新组合成字段拼接字符串：字段1,字段2,字段3
5. 根据id和tbname查询对应表中的enter所示字段以及系统字段，返回数据
   * 1. 查询信息详情接口（旧版）

**a.接口url:**

pub/admin/information/Infohtml/findInfo.do

**b.接口说明:**

该接口用于查询对应栏目下的信息数据。

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| findInfo | Info | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| id | String | 信息id |
| method | String | method为”modify”时为修改信息查找，method为“show”时为显示信息时查找信息 |

**d.返回值示例与说明**

//信息

{

“infos”：{……}

}

**e.功能实现说明**

1. 接收参数id、method、classid，有任意一个空值则返回“来源错误”
2. 根据classid找到栏目信息中的系统模型mid
3. 判断method为“modify”还是为“show”

1.当method为“modify”时

1. 根据通过mid找到系统模型信息中的表名（tbname）以及录入项（enter）
2. 分解enter重新组合成字段拼接字符串：字段1,字段2,字段3
3. 根据id和tbname查询对应表中的enter所示字段以及系统字段，返回 数据

2.当method为“show”时

1）根据通过mid找到系统模型信息中的表名（tbname）以及内容项（tempvar）

2）分解tempvar重新组合成字段拼接字符串：字段1,字段2,字段3

3）根据id和tbname查询对应表中的tempvar所示字段，返回数据

* + 1. 查询信息列表接口（旧版）

**a.接口url:**

pub/admin/information/Infohtml/findInfoList.do

**b.接口说明:**

该接口用于查询对应栏目下的信息数据列表。

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| findInfoList | Info | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| userid | String | 用户id |
| page | String | 当前页码 |
| num | String | 每页显示信息条数 |
| orderType | String | 排序类型 |
| tablechange | String | gd表示查询归档表 |
| searchAttr | String | 搜索方法的属性，如：  1为置顶、2为推荐、3为头条、4为未审核、5为已审核、6为投稿、7为签发、8为我的信息 |
| searchValue | String[] | 搜索方法的关键字数组 |
| searchField | String[] | 搜索方法的字段数组，即该表包含的字段，如：  标题、机构、副标题等。 |
| searchTime | String | 搜索方法的时间，如：  1天、2天、一周、1个月、3个月、6个月、1年。 |

**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": 1, //当前页信息数

"infos": [ //信息

{

……

},

{

……

}

]

}

**e.功能实现说明**

1. 检查参数classid、为空则返回信息“来源错误”。非空则通过classid找到栏目对象，通过栏目对象找到mid，通过mid找到系统模型，通过系统模型找到tbname。
2. 根据tbname找到相关分表
3. 如果tablechange非空，则根据tablechange修改各表名至对应归档表
4. 一次检测searchAttr、searchValue、searchField、searchTime参数，非空值、则根据对应数值拼接sql查询条件
5. 若orderType非空则拼接order by排序内容
6. 根据num和page拼接limit内容。若为空值则使用默认值
7. 执行生成好的sql语句、查询并返回数据
   * 1. 删除信息接口（旧版）

**a.接口url:**

pub/admin/information/Infohtml/deleteInfo.do

**b.接口说明:**

该接口用于删除、批量删除某个栏目下的信息

**c. 接口定义：**

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

**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传入参数classid、ids，通过classid找到栏目对象，通过栏目对象找到mid，通过mid找到系统模型，通过系统模型找到tbname。

②遍历ids数组，将表名于id传入**同步删除方法**

③返回result

* + 1. 增加/修改信息接口（旧版）

**a.接口url:**

pub/admin/information/Infohtml/operateInfo.do

**b.接口说明:**

该接口用于添加/修改信息内容。

**c. 接口定义：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口定义 | | | | |
| 接口名 | | 对象名 | | 返回值类型 |
| operateInfo | |  | | Result |
|  | |  | |  |
| infoJson | String | | 信息内容组成的json，因为不同栏目信息字段不同，所以动态生成json，格式为  {  "id":"",//信息id  "classid":"",//栏目id  "recommendation":"0",//推荐  "headline":"0"， //头条  "top":"0"， //置顶  "time":"0"， //修改信息的发布时间  "check":"0" //是否审核  ... //其他系统模型字段    } | |
| userid | String | | 用户ID | |

**d.返回值示例与说明**

{

"result": { //结果

"msg": "修改成功", //信息

"flag": "1" //标识

}

}

**e.功能实现说明**

①直接调用[**信息操作的方法**](#_5.1.9__信息操作方法) **5.1.9**，传入infojson

②根据方法返回值，返回信息

* + 1. 信息路径刷新接口
    2. 增加威客信息接口（这个删除，在修改信息时会增加威客信息）
    3. 信息转移/复制接口

**a.接口url:**

pub/admin/information/Infohtml/batchCopyOrTransferInfo.do

**b.接口说明:**

信息批量转移/复制

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| batchCopyOrTransferInfo | Result | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| dotype | Int | 操作类型 0代表移动 1代表复制 |
| toClassid | String | 目标栏目id |
| classid | String | 栏目id |
| ids | String | 转移Id集合 |
| start | Int | 转移的起点下标 |
| size | Int | 转移多少条数据 |

**d.返回值示例与说明**

{

"result": { //结果

"msg": "200", //成功转移/复制数量

"flag": "1" //标识

}

}

**e.功能实现说明**

1. 检查参数、classid，toClassid，为空值则返回信息 “来源错误”
2. 检查参数start、size、若为空则设置默认值start=0、size=200
3. 根据classid，toclassid分别获取对应系统模型信息，再进一步获取到数据表以及分表信息。
4. 若2个classid的数据表不同，则返回提示“数据表不符”
5. 检查参数ids、若非空、则拆分后组成sql条件【where id in (‘id1’, ‘id2’, ‘id3’)】
6. 根据操作类型判断，若是转移操作，则由sql条件生成update语句，直接修改classid并返回
7. 若是赋值操作，则根据classid结合id条件 组成sql查询语句执行
8. 遍历查询结果、依次把数据内容复制进目标栏目（注意主表、分表信息）。并记录操作信息数
9. 最后将记录操作信息数返回。
   * 1. 信息转栏目(其他)接口

**a.接口url:**

pub/admin/ column/manageColumn/infoTransferToOthers.do

**b.接口说明:**

将信息数据转换为栏目&其他表数据

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| infoTransferToOthers | Result | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| fid | String | 父项id |
| Id | String | 信息id |
| Tbname | String | 信息表名 |
| ctype | Int | 转换目标类型：  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 |
| isshow | int | 是否隐藏（仅限栏目生效） 0隐藏 1显示 |
| Modid | String | 系统模型id（仅限栏目生效） |

**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. 直接将参数传入 **信息转栏目（其他）方法** [5.1.11](#_5.1.11__信息转其他方法)
2. 返回方法传出的返回值
   * 1. 批量信息（根据给定id集）归档接口
     2. 批量信息（根据给定时间范围）归档接口

**a.接口url:**

pub/admin/information/Infohtml/batchInfoArchive.do

**b.接口说明:**

栏目信息归档/批量选择信息归档

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| batchInfoArchive | Result | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| classid | String | 栏目id |
| ids | String | 转入数据表名 |
| start | Int | 转移的起点下标 |
| size | Int | 转移多少条数据 |

**d.返回值示例与说明**

{

"result": { //结果

"msg": "200", //成功归档数量

"flag": "1" //标识

}

}

**e.功能实现说明**

1. 检查参数、classid，为空值则返回信息 “来源错误”
2. 检查参数start、size、若为空则设置默认值start=0、size=200
3. 调用[归档方法](#_5.1.8__归档方法)传入classid、ids、start、size
4. 把归档方法的返回值存入result归档数量、返回数据。
   * 1. 批量信息修改接口

**a.接口url:**

pub/admin/information/Infohtml/batchUpdateInfo.do

**b.接口说明:**

批量修改信息单个字段（批量审核、批量置顶等操作）

**c. 接口定义：**

|  |  |  |
| --- | --- | --- |
| 接口定义 | | |
| 接口名 | 对象名 | 返回值类型 |
| batchUpdateInfo | Result | Json |
| 参数说明： | | |
| 参数名称 | 参数类型 | 备注 |
| ids | String | Id集合、使用逗号分隔：  Id1,id2,id3, |
| Classid | String | 栏目id |
| key | String | 需要修改的字段名称 |
| value | Int | 需要修改的字段值 |

**d.返回值示例与说明**

{

"result": { //结果

"msg": "批量修改成功", //成功转移数量

"flag": "1" //标识

}

}

**e.功能实现说明**

1. 检查参数、ids、key、任意值为空值则返回信息 “来源错误”
2. 拆分ids后组成sql条件【where id in (‘id1’, ‘id2’, ‘id3’)】
3. 根据key和value参数组成update语句、并接上id条件
4. 执行sql语句
5. 返回result
   1. 对象