

Open Api 接口说明

修订历史记录

日期	版本	说明	作者
2014-10-17	1.0.0.1	创建	wynn
2014-10-22	1.0.0.2	更新	wynn
2014-10-28	1.0.0.3	异常 Code 更新	wynn

1 接口分类

1. 登入

POST /openapi/1.0/login

2. 登出

GET /openapi/1.0/logout

3. 打开表单

GET /openapi/1.0/{模板名称}/{objectId} 获取指定的表单

4. 更新表单

POST /openapi/1.0/{模板名称}/{objectId} 提交指定表单

5. 打开总表

GET /openapi/1.0/{模板名称} 打开指定模板总表

6. 打开总表过滤

POST /openapi/1.0/{模板名称} 打开指定模板总表，传入过滤条件

7. 运行数据规范

GET /openapi/1.0/{模板名称}/{数据规范名称}/query 执行模板内部数据规范

GET /openapi/1.0/global/数据规范名称/query 执行全局数据规范

POST /openapi/1.0/{模板名称}/{数据规范名称}/query 打开指定模板数据接口，传入过滤条件

8. 打开用户总表

GET /openapi/1.0/users 打开用户表单列表

POST /openapi/1.0/users	打开用户表单列表(传入 Json 格式进行分页过滤等操作)
9. 打开用户表单	
GET /openapi/1.0/users/{objectId}	打开指定用户
10. 更新用户表单	
POST /openapi/1.0/users/{objectId}	更新指定用户
11. 打开角色总表	
GET /openapi/1.0/roles	打开角色总表
POST /openapi/1.0/roles	打开用户表单列表(传入 Json 格式进行分页过滤等操作)
12. 打开角色表单	
GET /openapi/1.0/roles/{objectId}	打开角色指定
13. 更新角色表单	
POST /openapi/1.0/roles/{objectId}	更新指定角色
14. 上传附件	
POST /openapi/1.0/{templateName}/file	将附件上传至服务器端
15. 下载附件	
GET /openapi/1.0/{templateName}/file/{objectId}	从服务器下载指定的附件到本地

2 鉴权方式

2.1 名词解释

AppKey: 云表平台分配给每个应用空间的 AppKey, 使用进行生成 Sign(签名)

AppName: 云表平台分配给每个应用空间的 AppName, 用户需要进行保密, 是需要传递给后台服务器进行验证签名是否正确合法.

AppId: 应用空间 ID, 每个应用空间将会有个唯一 ID, 云表平台负责分配.

Sign: 签名, 每次访问服务器端都需要进行签名验证.

Timestamp: 时间戳, 每次访问服务器, 必须将当前时间戳传递给服务器, 6 分钟内的访问是有效的.

2.2 签名生成

生成签名的方式为, 使用云表提供的 AppKey 与当前访问时间戳进行 MD5 32 位加密, $Sign = (AppKey + 时间戳) MD5$ 加密 转成大写.

例如:

AppKey = 058457b7-0e4a-4f6c-b7f4-10d56689f036

Timestamp=1389085779854

合成 Sign = ((1389085779854058457b7-0e4a-4f6c-b7f4-10d56689f036)MD5) 转大写

最后合成值为:F3DFAE8C1B952779ED068B9DB66BE258

3 服务器接口访问

3.1 请求格式

Open Api 链接服务器所有接口都必须使用 **Https** 协议, 并且在 head 中传递服务器所需的参数.

参数列表:

x-eversheet-request-sign: 访问签名与时间戳一起传递使用, 进行分割

例如:

x-eversheet-request-sign = F3DFAE8C1B952779ED068B9DB66BE258, 1389085779854。

x-eversheet-application-name: 传递给服务器进行匹配签名。

x-eversheet-session-token: 临时令牌, 当登入成功后服务器将会生成一个临时令牌供客户使用. 如果进行登入, 值为空.

3.2 登入请求

访问接口方式 POST

访问接口协议 Https

访问相对地址: /openapi/1.0/login

Head 参数列表参照文档 3.1

Post 参数列表:

类型	参数名称	格式
String	loginJson	json

Json 格式:

```
{  
  
  "password": "F3DFAE8C1B952779ED068B9DB66BF36",  
}
```

```
"account": "wynn"  
  
}
```

类型	名称	描述
String	password	用户登入密码(将明文进行 MD5 32 位(转成大写) 加密传递给服务器)
String	Account	用户登入账号

登入成功返回值:

```
{  
  
  "logout": "/0/openapi/1.0/logout",  
  
  "token": "bd4bd75b569e4f8c5e8785f2c9afb7cc4236e9f7",  
  
  "serverVersion": "3.3.2.79",  
  
  "redirect": "文件夹",  
  
  "user": "admin"  
  
}
```

类型	名称	描述
String	Token	服务器访问地址
String	serverVersion	服务器版本号
String	User	当前登入名

3.3 登出请求

访问接口方式 GET

访问接口协议 Https

访问相对地址: /openapi/1.0/logout

Head 参数列表参照文档 **3.1** 需要将登入成功后的 token 传入 head 中, 服务器验证临时令牌, 如果验证不通过将会返回 500 错误异常. 可以参照 **3.10** 失败错误返回值. 访问登出接口返回值

```
{  
  
  "msg": "登出成功"  
  
}
```

3.4 打开表单总表接口

访问接口方式 GET

访问接口协议 Https

访问相对地址: /openapi/1.0/{模板名称}

Head 参数列表参照文档 3.1 需要将登入成功后的 token 传入 head 中, 服务器验证临时令牌, 如果验证不通过将会返回 500 错误异常. 可以参照 3.10 失败错误返回值. 访问表单总表接口返回值

```
{  
  
  "results": [  
  
    {  
  
      "objectVersion": 13,  
  
      "objectLocked": false,  
  
      "createdBy": "admin",  
  
      "createdAt": "2014-10-20 10:30:39",  
  
      "updatedAt": "admin",  
  
      "updatedAt": "2014-10-22 14:10:57",  
  
      "objectId": 104,  
  
    }  
  
  ]  
  
}
```

```
    "字段1": "33333",

    "字段2": "33"

  },

  {

    "objectVersion": 10,

    "objectLocked": false,

    "createdBy": "admin",

    "createdAt": "2014-10-20 10:30:39",

    "updatedBy": "admin",

    "updatedAt": "2014-10-20 16:45:56",

    "objectId": 103,

    "字段1": "2222111fg2",

    "字段2": "11"

  }

], pageInfo{

  "isUseRowIndex": false,

  "pageCount": 1,

  "pageIndex": 0,

  "pageSize": 50,
```

```

    "rowIndex": 0,

    "total": 1

}

}

```

类型	名称	描述
String	主表名称	模板中的主表
Int	objectVersion	表单的版本号
Boolean	objectLocked	表单是否锁定，如果锁定了的表单，是不允许进行修改的.
String	createdBy	表单创建者
String	updatedBy	表单最后修改者
String	createdAt	表单创建时间
String	updatedAt	表单修改时间
Int	objectId	表单ID

Object	其他	模板中定义的主表字段
--------	----	------------

JSONObject	PageInfo	分页信息																					
<table border="1"> <thead> <tr> <th data-bbox="391 309 762 425">类型</th> <th data-bbox="762 309 1058 425">名称</th> <th data-bbox="1058 309 1437 425">描述</th> </tr> </thead> <tbody> <tr> <td data-bbox="391 425 762 548">Int</td> <td data-bbox="762 425 1058 548">pageIndex</td> <td data-bbox="1058 425 1437 548">当前页面索引</td> </tr> <tr> <td data-bbox="391 548 762 672">Int</td> <td data-bbox="762 548 1058 672">rowIndex</td> <td data-bbox="1058 548 1437 672">当前记录行索引</td> </tr> <tr> <td data-bbox="391 672 762 795">Int</td> <td data-bbox="762 672 1058 795">pageCount</td> <td data-bbox="1058 672 1437 795">页数</td> </tr> <tr> <td data-bbox="391 795 762 918">Int</td> <td data-bbox="762 795 1058 918">pageSize</td> <td data-bbox="1058 795 1437 918">每页最大显示记录数量</td> </tr> <tr> <td data-bbox="391 918 762 1041">Int</td> <td data-bbox="762 918 1058 1041">Total</td> <td data-bbox="1058 918 1437 1041">记录总数</td> </tr> <tr> <td data-bbox="391 1041 762 1227">Boolean</td> <td data-bbox="762 1041 1058 1227">isUseRowIndex</td> <td data-bbox="1058 1041 1437 1227">使用行数进行分页,类似微博一样的分页</td> </tr> </tbody> </table>			类型	名称	描述	Int	pageIndex	当前页面索引	Int	rowIndex	当前记录行索引	Int	pageCount	页数	Int	pageSize	每页最大显示记录数量	Int	Total	记录总数	Boolean	isUseRowIndex	使用行数进行分页,类似微博一样的分页
类型	名称	描述																					
Int	pageIndex	当前页面索引																					
Int	rowIndex	当前记录行索引																					
Int	pageCount	页数																					
Int	pageSize	每页最大显示记录数量																					
Int	Total	记录总数																					
Boolean	isUseRowIndex	使用行数进行分页,类似微博一样的分页																					
List<JSONObject>	paramList	参数列表																					
<table border="1"> <thead> <tr> <th data-bbox="391 1357 695 1473">类型</th> <th data-bbox="695 1357 1074 1473">名称</th> <th data-bbox="1074 1357 1452 1473">描述</th> </tr> </thead> <tbody> <tr> <td data-bbox="391 1473 695 1597">String</td> <td data-bbox="695 1473 1074 1597">Param</td> <td data-bbox="1074 1473 1452 1597">参数名称</td> </tr> <tr> <td data-bbox="391 1597 695 1720">Object</td> <td data-bbox="695 1597 1074 1720">Value</td> <td data-bbox="1074 1597 1452 1720">参数值</td> </tr> </tbody> </table>			类型	名称	描述	String	Param	参数名称	Object	Value	参数值												
类型	名称	描述																					
String	Param	参数名称																					
Object	Value	参数值																					
List<JSONObject>	sortList	排序列表																					
<table border="1"> <thead> <tr> <th data-bbox="391 1850 715 1966">类型</th> <th data-bbox="715 1850 1093 1966">名称</th> <th data-bbox="1093 1850 1471 1966">描述</th> </tr> </thead> <tbody> <tr> <td data-bbox="391 1966 715 2089">String</td> <td data-bbox="715 1966 1093 2089">fieldName</td> <td data-bbox="1093 1966 1471 2089">排序字段名称</td> </tr> <tr> <td data-bbox="391 2089 715 2210">Boolean</td> <td data-bbox="715 2089 1093 2210">isDesc</td> <td data-bbox="1093 2089 1471 2210">是否为降序</td> </tr> </tbody> </table>			类型	名称	描述	String	fieldName	排序字段名称	Boolean	isDesc	是否为降序												
类型	名称	描述																					
String	fieldName	排序字段名称																					
Boolean	isDesc	是否为降序																					

3.4.1 总表表单过滤

访问接口方式 POST

访问接口协议 Https

访问相对地址: /openapi/1.0/{模板名称}

Head 参数列表参照文档 3.1 需要将登入成功后的 token 传入 head 中, 服务器验证临时令牌, 如果验证不通过将会返回 500 错误异常. 可以参照 3.10 失败错误返回值. 访问表单总表接口返回值

提交保存表单请求 Post 参数列表:

类型	参数名称	格式
String	formJson	json

提交的 Json 格式:

```
{  
  
  "pageInfo": {  
  
    "isUseRowIndex": false,  
  
    "pageCount": 1,  
  
    "pageIndex": 0,  
  
    "pageSize": 50,  
  
    "RowIndex": 0,  
  
  }  
}
```

```

        "total": 5

    },

    "filter": [

        {

            "expressionList": [

                {

                    "operator": "in",

                    "isAnd": true,

                    "value": "匹配值"

                }

            ],

            "filterField": "匹配字段"

        }

    ]

}
    
```

类型	名称	描述
JSONObject	pageInfo	总表分页信息
Boolean	isUseRowIndex	使用行数进行分页，类似微博一样

			的分页
	Int	rowIndex	显示记录的行索引, 当 <code>isUseRowIndex=true</code> 时才有用, 配合, 表示要显示从 <code>rowIndex</code> 索引起, 显示的条数 <code>pageSize</code>
	Int	pageCount	总页数
	Int	pageIndex	当前分页索引(0开始)
	Int	pageSize	每页显示条数
	Int	total	记录总数
List<JSONObject>		Filter	总表过滤信息列表
	List<JSONObject>	expressionList	过滤列表(同一字段的过滤条件)
	String	Operator	参数匹配比较可以在文档 3.10.2 查看到
	Boolean	isAnd	是否使用并且连接, 选择 false则使用或者
	String	Value	与字段匹配的值
	String	filterField	字段名称

例:

1. 我们需要匹配 “客户信息”模板, ”姓名”字段值为”陈远光”的信息. 那么访问地址为:

/openapi/1.0/客户信息

过滤条件 JSON 为:

```
{  
  
  "pageInfo": {  
  
    "isUseRowIndex": false,  
  
    "pageCount": 1,  
  
    "pageIndex": 0,  
  
    "pageSize": 50,  
  
    "rowIndex": 0,  
  
    "total": 5  
  
  },  
  
  "filter": [  
  
    {  
  
      "expressionList": [  
  
        {  
  
          "operator": "$e",  
  
          "isAnd": true,  
  

```

```
      "value": "陈远光"
    }
  ],
  "filterField": "姓名"
}
]
```

2. 我们需要匹配 “客户信息” 模板, “姓名” 字段值为 “陈远光” 并且性别字段为 “男” 的信息, 那么访问地址为:

/openapi/1.0/客户信息

过滤条件 JSON 为:

```
{
  "pageInfo": {
    "isUseRowIndex": false,
    "pageCount": 1,
    "pageIndex": 0,
    "pageSize": 50,
    "RowIndex": 0,
```

```
    "total": 5

  },

  "filter": [

    {

      "expressionList": [

        {

          "operator": "$e",

          "isAnd": true,

          "value": "陈远光"

        }

      ],

      "filterField": "姓名"

    },

    {

      "expressionList": [

        {

          "operator": "$e",

          "isAnd": true,

          "value": "男"

        }

      ]

    }

  ]

}
```

```
    }  
  
    ],  
  
    "filterField": "性别"  
  
  }  
  
]  
  
}
```

3.4.2 过滤的匹配比较列表

Key	Operation
\$e	等于
\$lt	小于
\$lte	小于等于
\$gt	大于
\$gte	大于等于
\$ne	不等于

\$in	包含
\$nin	不包含

3.5 打开表单接口

访问接口方式 GET

访问接口协议 Https

访问相对地址: /openapi/1.0/{模板名称}/{objectId}

Head 参数列表参照文档 **3.1** 需要将登入成功后的 token 传入 head 中, 服务器验证临时令牌, 如果验证不通过将会返回 500 错误异常. 可以参照 **3.10** 失败错误返回值. 访问表单总表接口返回值

3.5.1 成功访问表单接口返回值

```
{  
  
  "字段1": "33333",  
  
  "字段2": "11",  
  
  "明细": [  
  
    {  
  
      "objectId": 110,  

```

```
"recOrder": 1,  
  
"字段 3": "231",  
  
"字段 4": "热 1",  
  
"字段 5": "v 个",  
  
"字段 6": "12"  
  
},  
  
{  
  
"objectId": 111,  
  
"recOrder": 2,  
  
"字段 3": "1234",  
  
"字段 4": "第三方",  
  
"字段 5": "23",  
  
"字段 6": "问问"  
  
},  
  
{  
  
"objectId": 112,  
  
"recOrder": 3,  
  
"字段 3": "5423",  
  
"字段 4": "的",
```

```

    "字段 5": "收到",

    "字段 6": "32"

}

],

"objectVersion": 12,

"objectLocked": false,

"createdAt": "2014-10-20 10:30:39",

"updatedAt": "2014-10-20 16:51:30",

"objectId": 104

}

```

类型	名称	描述
String	objectVersion	表单版本，每次保存，版本都会发生改变，保存时需要获取到最新的表单版本进行保存。不然会提示，错误信息，版本不匹配。
Int	objectId	表单ID
Boolean	objectLocked	表单是否锁定，如果锁定了的表单，是不允许进行修改的。

String	createdAt	表单创建时间
String	updatedAt	表单修改时间
List<JSONObject>	明细	明细表的值信息情况, 这是一个列表, 每个JSONObject都是一条记录值, 由明细的字段与值名称与值进行组合的Json

3.6 提交保存表单接口

访问接口方式 POST

访问接口协议 Https

访问相对地址: /openapi/1.0/{模板名称}/{表单 ID}

Head 参数列表参照文档 **3.1** 需要将登入成功后的 token 传入 head 中, 服务器验证临时令牌, 如果验证不通过将会返回 500 错误异常. 可以参照 3.10 失败错误返回值. 访问表单总表接口返回值

提交保存表单必须先获取到表单的信息, 进行修改值后再进行提交到服务器, 因为每个表单都有自己的版本号, 版本号必须要匹配才能进行保存.

1. 先根据 3.5 打开表单, 获取到需要修改的表单 Json.
2. 根据客户修改的值构建一个与打开表单一样的 Json 进行提交到服务器端. 服务器进行保存.
3. 服务器保存成功后会返回与传入的 Json 一样格式的 Json, 并且会修正表单版本号. 方便用户进行二次保存.

提交保存表单请求 Post 参数列表:

类型	参数名称	格式
String	formJson	json

提交的 Json 格式:

```
{  
  
  "字段2": "33333",  
  
  "字段1": "11",  
  
  "明细": [  
  
    {  
  
      "objectId": 110,  
  
      "recOrder": 1,  
  
      "字段3": "231",  
  
      "字段6": "热 1",
```

```
    "字段5": "v个",  
  
    "字段4": "12"  
  
  },  
  
  {  
  
    "objectId": 111,  
  
    "recOrder": 2,  
  
    "字段3": "1234",  
  
    "字段6": "第三方",  
  
    "字段5": "23",  
  
    "字段4": "问问"  
  
  },  
  
  {  
  
    "objectId": 112,  
  
    "recOrder": 3,  
  
    "字段3": "5423",  
  
    "字段6": "的",  
  
    "字段5": "收到",  
  
    "字段4": "32"  
  
  }  
}
```

FORMS USE TABLE COMMENTS COMMENTS

```

],

  "objectVersion": 12,

  "objectLocked": false,

  "createdAt": "2014-10-20 10:30:39",

  "updatedAt": "2014-10-20 16:51:30",

  "objectId": 104

}
    
```

类型	名称	描述
String	objectVersion	表单版本，每次保存，版本都会发生改变，保存时需要获取到最新的表单版本进行保存。不然会提示，错误信息，版本

		不匹配.
Int	objectId	表单ID, 如果objectId为0或者为空, 则会新创建一条新的记录.
Boolean	objectLocked	表单是否锁定, 如果锁定了的表单, 是不允许进行修改的.
String	createdAt	表单创建时间
String	updatedAt	表单修改时间
List<JSONObject>	明细	明细表的值信息情况, 这是一个列表, 每个JSONObject都是一条记录值, 由明细的字段与值名称与值进行组合的Json

如果出现异常, 服务器将会放回 500 错误

失败返回值:

服务器如果失败, 则会返回 500 错误, 错误 Json 格式:

```
{
  "Exception": {
    "code": 0,
    "msg": "\n登录失败, 请检查账号密码是否正确\n"
```

```

}

}
    
```

类型	名称	描述
JSON	Exception	Json对象
Int	Code	错误编码
String	Msg	异常内容(抛出的异常服务器经过了Unicode编码)

提交成功返回值:

提交成功后服务器返回值与客户提交的 Json 是一样的, 服务器将最新的表单版本从新赋值. 方便进行二次修改.

```

{

  "字段2": "33333",

  "字段1": "11",

  "明细": [

    {

      "objectId": 110,

      "recOrder": 1,
    }
  ]
}
    
```

```
"字段 3": "231",

"字段 6": "热 1",

"字段 5": "v 个",

"字段 4": "12"

},

{

"objectId": 111,

"recOrder": 2,

"字段 3": "1234",

"字段 6": "第三方",

"字段 5": "23",

"字段 4": "问问"

},

{

"objectId": 112,

"recOrder": 3,

"字段 3": "5423",

"字段 6": "的",

"字段 5": "收到",
```

```
    "字段4": "32"  
  
  }  
  
],  
  
  "objectVersion": 13,  
  
  "objectLocked": false,  
  
  "createdAt": "2014-10-20 10:30:39",  
  
  "updatedAt": "2014-10-20 16:51:30",  
  
  "objectId": 104  
  
}
```

3.7 删除表单

3.7.1 删除用户表单

访问接口方式 DELETE

访问接口协议 Https

访问相对地址: /openapi/1.0/users/{表单 ID}

3.7.2 删除角色表单

访问接口方式 DELETE

访问接口协议 Https

访问相对地址: /openapi/1.0/roles/{表单 ID}

3.7.3 删除普通表单

访问接口方式 DELETE

访问接口协议 Https

访问相对地址: /openapi/1.0/{模板名称}/{表单 ID}

3.8 用户对象操作

3.8.1 打开用户总表

访问接口方式 GET

访问接口协议 Https

访问相对地址: /openapi/1.0/users

Head 参数列表参照文档 [3.1](#) 需要将登入成功后的 token 传入 head 中, 服务器验证临时令牌, 如果验证不通过将会返回 500 错误异常. 可以参照 [3.10](#) 失败错误返回值. 访问表单总表接口返回值

3.8.1.1 成功访问用户总表接口返回值

```
{  
  
  "results": [  
  
    {  
  
      "m_org_name_path": null,  
  
      "m_role_name": "everyone, learner",  
  
    }  
  
  ]  
}
```

```
"m_post_name": "注册岗位",

"m_account": "a1",

"createdAt": null,

"updatedAt": "2014-10-21 16:19:45",

"objectId": 14,

"m_email": "",

"m_phone": "",

"m_name": "a12 额"

},

{

"m_org_name_path": "",

"m_role_name": "a2",

"m_post_name": "",

"m_account": "a2",

"createdAt": null,

"updatedAt": "2014-10-14 17:15:39",

"objectId": 15,

"m_email": "",

"m_phone": "",
```

```
    "m_name": "a2"

  }

], pageInfo{

  "isUseRowIndex": false,

  "pageCount": 1,

  "pageIndex": 0,

  "pageSize": 50,

  "RowIndex": 0,

  "total": 1

}

}
```

类型

名称

描述

JSONObject	PageInfo	分页信息																					
<table border="1"> <thead> <tr> <th>类型</th> <th>名称</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>Int</td> <td>pageIndex</td> <td>当前页面索引</td> </tr> <tr> <td>Int</td> <td>rowIndex</td> <td>当前记录行索引</td> </tr> <tr> <td>Int</td> <td>pageCount</td> <td>页数</td> </tr> <tr> <td>Int</td> <td>pageSize</td> <td>每页最大显示记录数量</td> </tr> <tr> <td>Int</td> <td>Total</td> <td>记录总数</td> </tr> <tr> <td>Boolean</td> <td>isUseRowIndex</td> <td>使用行数进行分页,类似微博一样的分页</td> </tr> </tbody> </table>			类型	名称	描述	Int	pageIndex	当前页面索引	Int	rowIndex	当前记录行索引	Int	pageCount	页数	Int	pageSize	每页最大显示记录数量	Int	Total	记录总数	Boolean	isUseRowIndex	使用行数进行分页,类似微博一样的分页
类型	名称	描述																					
Int	pageIndex	当前页面索引																					
Int	rowIndex	当前记录行索引																					
Int	pageCount	页数																					
Int	pageSize	每页最大显示记录数量																					
Int	Total	记录总数																					
Boolean	isUseRowIndex	使用行数进行分页,类似微博一样的分页																					
List<JSONObject>	paramList	参数列表																					
<table border="1"> <thead> <tr> <th>类型</th> <th>名称</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>String</td> <td>Param</td> <td>参数名称</td> </tr> <tr> <td>Object</td> <td>Value</td> <td>参数值</td> </tr> </tbody> </table>			类型	名称	描述	String	Param	参数名称	Object	Value	参数值												
类型	名称	描述																					
String	Param	参数名称																					
Object	Value	参数值																					
List<JSONObject>	sortList	排序列表																					
<table border="1"> <thead> <tr> <th>类型</th> <th>名称</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>String</td> <td>fieldName</td> <td>排序字段名称</td> </tr> <tr> <td>Boolean</td> <td>isDesc</td> <td>是否为降序</td> </tr> </tbody> </table>			类型	名称	描述	String	fieldName	排序字段名称	Boolean	isDesc	是否为降序												
类型	名称	描述																					
String	fieldName	排序字段名称																					
Boolean	isDesc	是否为降序																					

List<JSONObject>	用户	用户列表
String	createdAt	创建时间
String	updatedAt	修改时间
String	m_org_name_path	用户部门全称
Int	objectId	用户的ID, 唯一值, 可以根据ID获取到用户对象
String	m_email	用户电子邮箱
String	m_role_name	当前用户所拥有的角色列表
String	m_account	用户的账号
String	m_email	用户填写到电话号码
String	m_post_name	用户所属岗位
String	m_name	用户填写的姓名

3.8.2 打开用户表单

访问接口方式 GET

访问接口协议 Https

访问相对地址: /openapi/1.0/users/{objectId}

Head 参数列表参照文档 **3.1** 需要将登入成功后的 token 传入 head 中, 服务器验证临时令牌, 如果验证不通过将会返回 500 错误异常. 可以参照 3.10 失败错误返回值. 访问表单总表接口返回值

3.8.2.1 成功访问表单接口返回值

```
{  
  
  "createDate": "2014-10-22 11:33:50",  
  
  "mainPostId": 3,  
  
  "roleList": [  
  
    {  
  
      "createDate": "2014-09-05 11:33:52",  
  
      "objectId": 7,  
  
      "sysName": "learner",  
  
      "desc": "此角色能够查看, 学习模板, 不能编辑模板",  
  
      "name": "learner"  
    },  
  
    {  
  
      "createDate": "2014-09-05 11:33:52",  
  
      "objectId": 4,  
  

```

```
    "sysName": "everyone",

    "desc": "所有人。所有用户都属于这个角色",

    "name": "everyone"

  }

],

"postList": [

  {

    "objectId": 3,

    "parentId": 0,

    "desc": "用户选择组织机构",

    "name": "注册岗位"

  }

],

"account": "用户账号",

"formId": 14,

"email": "614**@qq.com",

"phone": "135***",

"name": "用户名称"

}
```

类型	名称	描述	
String	createDate	创建时间	
Int	mainPostId	主岗位ID	
String	Account	用户账号	
String	Email	用户邮箱	
String	Phone	用户电话	
String	Name	用户姓名	
List<JSONObject>	roleList	用户角色列表集合对象	
	String	createDate	角色创建时间
	Int	objectId	角色对象ID
	String	sysName	角色系统名称
	String	Desc	角色的描述
	String	Name	角色名称
List<JSONObject>	postList	当前用户岗位对象列表	

	Int	objectId	岗位对象的Id唯一值
	Int	parentId	岗位父对象ID
	String	Desc	岗位描述
	String	Name	岗位名称

3.8.3 更新用户对象

访问接口方式 POST

访问接口协议 Https

访问相对地址: /openapi/1.0/users/{objectId}

Head 参数列表参照文档 3.1 需要将登入成功后的 token 传入 head 中, 服务器验证临时令牌, 如果验证不通将会返回 500 错误异常. 可以参照 3.10 失败错误返回值. 访问表单总表接口返回值.

提交参数名: formJson **参数格式:** json

内容: (根据打开表单的方式获取到用户对象, 根据用户对象修改需要改变的值, 如: 需要修改姓名将 json 中 “name” 值进行修改后提交给服务器, 服务器将进行保存.)

```
{
  "createDate": "2014-10-22 11:33:50",
  "mainPostId": 3,
  "roleList": [
```

```
{

  "createDate": "2014-09-05 11:33:52",

  "objectId": 7,

  "sysName": "learner",

  "desc": "此角色能够查看，学习模板，不能编辑模板",

  "name": "learner"

},

{

  "createDate": "2014-09-05 11:33:52",

  "objectId": 4,

  "sysName": "everyone",

  "desc": "所有人。所有用户都属于这个角色",

  "name": "everyone"

}

],

"postList": [

  {

    "objectId": 3,

    "parentId": 0,
```

```
    "desc": "用户选择组织机构",

    "name": "注册岗位"

  }

],

"account": "用户账号",

"formId": 14,

"email": "614**@qq.com",

"phone": "135***",

"name": "用户名称",

"password": "密码(需要使用 MD5 进行加密 32 位进行加密. 传入服务器端, 服务器端构建正确的密码, 当用户没有传递密码, 则不保存密码, 使用旧密码, 如果传入密码, 则会使用新的密码)"

}
```

需要新建用户, 则将用户的"objectId"值赋值为 0, 服务器将会创建出一个新的用户对象. 密码必须是使用 MD5 加密后的结果.

3.9 角色对象操作

3.9.1 打开角色总表

访问接口方式 GET

访问接口协议 Https

访问相对地址: /openapi/1.0/roles

Head 参数列表参照文档 **3.1** 需要将登入成功后的 token 传入 head 中, 服务器验证临时令牌, 如果验证不通过将会返回 500 错误异常. 可以参照 3.10 失败错误返回值. 访问表单总表接口返回值

3.9.1.1 成功访问角色总表接口返回值

```
{  
  
  "results": [  
  
    {  
  
      "m_description": "",  
  
      "createdAt": "2014-10-22 11:11:45",  
  
      "updatedAt": "2014-10-22 11:11:45",  
  
      "objectId": 153,  
  
      "m_isSys": null,  
  
      "m_name": "a23"  
    },  
  
    {  
  
      "m_description": "",  
  
      "createdAt": "2014-10-22 11:14:26",  
  
      "updatedAt": "2014-10-22 11:14:26",  
  
      "objectId": 154,  
  
    }  
  ]  
}
```

```
    "m_isSys": null,  
  
    "m_name": "a44455"  
  
  }  
  
  ], pageInfo{  
  
    "isUseRowIndex": false,  
  
    "pageCount": 1,  
  
    "pageIndex": 0,  
  
    "pageSize": 50,  
  
    "RowIndex": 0,  
  
    "total": 1  
  
  }  
  
}
```

类型	名称	描述
----	----	----

JSONObject		PageInfo	分页信息
类型	名称	描述	
Int	pageIndex	当前页面索引	
Int	rowIndex	当前记录行索引	
Int	pageCount	页数	
Int	pageSize	每页最大显示记录数量	
Int	Total	记录总数	
Boolean	isUseRowIndex	使用行数进行分页,类似微博一样的分页	
List<JSONObject>	paramList	参数列表	
类型	名称	描述	
String	Param	参数名称	
Object	Value	参数值	
List<JSONObject>	sortList	排序列表	
类型	名称	描述	
String	fieldName	排序字段名称	
Boolean	isDesc	是否为降序	

List<JSONObject>	角色	角色列表
String	createdAt	创建时间
String	updatedAt	修改时间
Int	objectId	角色对象的ID, 唯一值, 可以根据ID获取到角色对象
String	m_name	角色名称
Boolean	M_isSys	是否是系统角色
String	m_description	角色描述

3.9.2 打开指定角色对象

访问接口方式 GET

访问接口协议 Https

访问相对地址: /openapi/1.0/roles/{objectId}

Head 参数列表参照文档 3.1 需要将登入成功后的 token 传入 head 中, 服务器验证临时令牌, 如果验证不通过将会返回 500 错误异常. 可以参照 3.10 失败错误返回值. 访问表单总表接口返回值

3.9.2.1 成功访问表单接口返回值

```
{
  "m_description": "",
  "createdAt": "2014-10-22 11:14:26",
  "updatedAt": "2014-10-22 11:14:26",
  "objectId": 154,
  "isSys": false,
  "name": "a44455"
}
```

类型	名称	描述
String	createDate	创建时间
String	updateData	修改时间
Int	objectId	对象ID
String	m_description	描述
String	name	角色名称

Boolean	isSys	是否为系统角色
---------	-------	---------

3.9.3 更新角色对象

访问接口方式 POST

访问接口协议 Https

访问相对地址: /openapi/1.0/roles/{objectId}

Head 参数列表参照文档 3.1 需要将登入成功后的 token 传入 head 中, 服务器验证临时令牌, 如果验证不通过将会返回 500 错误异常. 可以参照 3.10 失败错误返回值. 访问表单总表接口返回值.

提交参数名: formJson **参数格式:** json

内容: (根据打开表单的方式获取到角色对象, 根据角色对象修改需要改变的值, 如: 需要修改姓名将 json 中 “name” 值进行修改后提交给服务器, 服务器将进行保存.)

```
{  
  
  "m_description": "描述",  
  
  "objectId": 154,  
  
  "name": "名称"  
}
```

3.10 数据接口操作

3.10.1 使用全局数据规范

访问接口方式 GET

访问接口协议 Https

访问相对地址:openapi/1.0/global/{数据接口名称}/query

Head 参数列表参照文档 **3.1** 需要将登入成功后的 token 传入 head 中, 服务器验证临时令牌, 如果验证不通过将会返回 500 错误异常. 可以参照 3.10 失败错误返回值. 访问表单总表接口返回值

3.10.1.1 成功访问全局数据接口返回值

```
{  
  
  "results": [  
  
    {  
  
      "字段1": "33333",  
  
      "字段2": "44",  
  
      "字段3": "33",  
  
      "字段4": "33"  
    },  
  
    {  
  
      "字段1": "33333",  
  
      "字段2": "555",  
  
    }  
  ]  
}
```

```

    "字段 3": "33",

    "字段 4": "33"

  }

]

}

```

类型	名称	描述
List<JSONObject>	列表选择数据接口列表	列表集合
Object	数据接口设计的字段	数据接口设置的字段值

3.10.2 使用指定模板数据规范

访问接口方式 GET

访问接口协议 Https

访问相对地址:openapi/1.0/{模板名称}/{数据接口名称}/query

Head 参数列表参照文档 **3.1** 需要将登入成功后的 token 传入 head 中, 服务器验证临时令牌, 如果验证不通过将会返回 500 错误异常. 可以参照 3.10 失败错误返回值. 访问表单总表接口返回值

3.10.2.1 成功访问全局数据接口返回值

```
{
```

```
"results": [  
  
  {  
  
    "字段 1": "2222",  
  
    "字段 2": "11"  
  
  },  
  
  {  
  
    "字段 1": "222ggggg2",  
  
    "字段 2": "11"  
  
  },  
  
  {  
  
    "字段 1": "222ggggfffg2",  
  
    "字段 2": "11"  
  
  },  
  
  {  
  
    "字段 1": "2222111fg2",  
  
    "字段 2": "11"  
  
  },  
  
  {  
  
    "字段 1": "33333",
```

```

        "字段2": "33"
    }
]
}
    
```

类型	名称	描述
List<JSONObject>	指定模板数据接口	列表集合
JSONObject	数据接口设计的字段	数据接口设置的字段值

3.10.3 访问参数调用全局数据接口

访问接口方式 POST

访问接口协议 Https

访问相对地址:openapi/1.0/global/{数据接口名称}/query

Head 参数列表参照文档 **3.1** 需要将登入成功后的 token 传入 head 中, 服务器验证临时令牌, 如果验证不通过将会返回 500 错误异常. 可以参照 3.10 失败错误返回值. 访问表单总表接口返回值

调用数据接口请求 Post 传递参数:

类型	参数名称	格式
----	------	----

String	formJson	json
JSONObject	PageInfo	分页信息
类型	名称	描述
Int	pageIndex	当前页面索引
Int	rowIndex	当前记录行索引
Int	pageCount	页数
Int	pageSize	每页最大显示记录数量
Int	Total	记录总数
Boolean	isUseRowIndex	使用行数进行分页,类似微博一样的分页
List<JSONObject>	paramList	参数列表
类型	名称	描述
String	Param	参数名称
Object	Value	参数值

List<JSONObject>			sortList			排序列表		
类型			名称			描述		
String			fieldName			排序字段名称		
Boolean			isDesc			是否为降序		
List<JSONObject>			Filter			过滤列表		
类型			名称			描述		
String			filterField			过滤字段名称		
List<JSONObject>			Expresionlist			过滤表表达式列表		
类型			名称			描述		
Int			Operator			参数匹配比较可以在文档 3.10.2 查看到		
Object			Value			匹配值		

	Boolean	IsAnd	连接下一个表达式的条件， 如果没有下一个表达式，则 不需要此参数
--	---------	-------	--

提交参数名: queryParams 参数格式: json

内容: (如果需要分页或者进行过滤,则需要提交参数 Json,比如说只需要进行分页,就只传入分页的 Json,只要进行过滤,就只传入需要过滤的 Json,不需要所有都传入.)

```
{  
  
  "pageInfo": {  
  
    "isUseRowIndex": false,  
  
    "pageCount": 1,  
  
    "pageIndex": 0,  
  
    "pageSize": 50,  
  
    "rowIndex": 0,  
  
    "total": 1  
  
  },  
  
  "paramList": [  
  
    {
```

```
    "param": "姓名",

    "value": "wynn"

  }

],

"sortList": [

  {

    "isDesc": false,

    "field": "you_xiang"

  }

],

"filter": [

  {

    "expressionList": [

      {

        "operator": 10,

        "isAnd": true,

        "value": "wynn"

      }

    ]

  }

],
```

```
    "filterField": "ying_wen_ming"

  }

]

}
```

3.11 上传附件接口

访问接口方式 POST

访问接口协议 Https

访问相对地址:openapi/1.0/{模板名称}/file

Head 参数列表参照文档 **3.1** 需要将登入成功后的 token 传入 head 中, 服务器验证临时令牌, 如果验证不通过将会返回 500 错误异常. 可以参照 3.10 失败错误返回值. 访问表单总表接口返回值

上传附件注意事项:上传附件是绑定在表单当中, 需要上传附件则必须模板开启了附件功能. 上传完附件后, 获取到附件的 ObjectId, 保存入表单附件表。

1. 上传附件流程, 调用 openapi/1.0/{模板名称}/file 接口, 传入以下参数进行附件上传.

上传附件请求 Post 参数列表:

类型	参数名称	格式
----	------	----

File	file	附件
String	fileType	附件类型

2. 上传完附件后，服务器返回 Json 格式

```
{  
  
  "objectId": 101,  
  
}
```

3. 将服务器返回的附件对象 ID，保存至所属表单中. 例如打开表单返回 Json 格式

```
{  
  
  "objectVersion": 1,  
  
  "objectLocked": false,  
  
  "createdAt": "2014-10-28 18:44:39",  
  
  "updatedAt": "2014-10-28 18:44:39",  
  
  "objectId": 301,  
  
  "测试 3": "",  
  
  "测试 2": "33",  
  
  "测试 1": "222222",  
  
  "测试 5": "",  
  
}
```

```
"测试4": "",
```

```
"附件": [
```

```
{
```

```
  "附件ID": "1",
```

```
  "objectId": 1,
```

```
  "recOrder": 1,
```

```
  "附件名": "SerialKey.lic",
```

```
  "上传人": "",
```

```
  "类型": "lic",
```

```
  "大小": 80
```

```
}
```

```
],
```

```
"明细": [
```

```
{
```

```
  "objectId": 201,
```

```
  "recOrder": 1,
```

```
  "测试8": "55",
```

```
  "测试7": "44",
```

```
  "测试6": "33"
```

```

    }
  ]
}

```

类型	名称	描述
String	objectVersion	表单版本, 每次保存, 版本都会发生改变, 保存时需要获取到最新的表单版本进行保存. 不然会提示, 错误信息, 版本不匹配.
Int	objectId	表单ID
Boolean	objectLocked	表单是否锁定, 如果锁定了的表单, 是不允许进行修改的.
String	createdAt	表单创建时间
String	updatedAt	表单修改时间
List<JSONObject>	明细	明细表的值信息情况, 这是一个列表, 每个JSONObject都是一条记录值, 由明细的字段与值名称与值进行组合的Json
List<Object>	附件	表单的附件表记录, 先将附件上传至服务器, 获取到附件的ObjectId, 写入记录至

		附件表.
String	附件ID	附件的ObjectId(根据上传附件后服务器返回的ObjectId进行赋值.)
Long	objectId	附件表Id
String	附件名	附件名称
String	上传者	上传附件用户
String	类型	附件类型
Int	大小	附件大小(单位: 字节)

上传完附件后，需要在保存表单中添加附件的相关记录, 使用以上打开表单为例添加一个附件记录.

例：上传了一个附件为 *.jpg 大小为 80 字节，上传附件返回的 ObjectId 为 201, 上传人为”小王”，提交的 Json 应该为.

```
{
  "objectVersion": 1,
  "objectLocked": false,
  "createdAt": "2014-10-28 18:44:39",
  "updatedAt": "2014-10-28 18:44:39",
```

```
"objectId": 301,

"测试3": "",

"测试2": "33",

"测试1": "222222",

"测试5": "",

"测试4": "",

"附件": [

  {

    "附件ID": "1",

    "objectId": 1,

    "recOrder": 1,

    "附件名": "SerialKey.lic",

    "上传者": "",

    "类型": "lic",

    "大小": 80

  },

  {

    "附件ID": "201",

    "objectId": 0,
```

```
    "recOrder": 1,

    "附件名": "**.jpg",

    "上传人": "",

    "类型": "lic",

    "大小": 32

  }

],

"明细": [

  {

    "objectId": 201,

    "recOrder": 1,

    "测试8": "55",

    "测试7": "44",

    "测试6": "33"

  }

]

}
```

在传入的 Json 格式中，红色标记的 Json 为添加的一条附件记录.赋值好后进行保存表单.

3.12 下载附件接口

访问接口方式 GET

访问接口协议 Https

访问相对地址:openapi/1.0/{模板名称}/file/{objectId}

Head 参数列表参照文档 **3.1** 需要将登入成功后的 token 传入 head 中, 服务器验证临时令牌, 如果验证不通过将会返回 500 错误异常. 可以参照 3.10 失败错误返回值. 访问表单总表接口返回值.

传入上传后服务器返回的附件 ObjectId(注意: 是上传附件后服务器返回的 ObjectId), 进行获取附件, 将会返回 File 对象. 可以直接进行下载.

3.13 发送短信

访问接口方式 GET

访问接口协议 http

访问相对地址: openapi/1.0/sendSms/new

Head 参数列表参照文档 **3.1** 需要将登入成功后的 token 传入 head 中, 服务器验证临时令牌, 如果验证不通过将会返回 500 错误异常. 可以参照 3.10 失败错误返回值.

在 Head 中访问的 URL, 与 AppKey, AppName, 账号密码需要向云表申请.

JSON 格式为:

```
{  
  
  "className": "短信",  
  
  "明细": [  
  
    {
```

```

        "latoId": "0",

        "msg": "",

        "smsTemplate": "",

        "toNumber": "",

        "smsId": ""

    }

]

}
    
```

类型		参数名称	格式																		
String		formJson	json																		
<table border="1"> <thead> <tr> <th>类型</th> <th>名称</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>String</td> <td>className</td> <td></td> </tr> <tr> <td>List<JSONObject></td> <td>明细</td> <td>允许处理多条短信记录</td> </tr> <tr> <td colspan="3"> <table border="1"> <thead> <tr> <th>类型</th> <th>名称</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>Int</td> <td>latoId</td> <td>唯一值, 默认为 0</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>				类型	名称	描述	String	className		List<JSONObject>	明细	允许处理多条短信记录	<table border="1"> <thead> <tr> <th>类型</th> <th>名称</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>Int</td> <td>latoId</td> <td>唯一值, 默认为 0</td> </tr> </tbody> </table>			类型	名称	描述	Int	latoId	唯一值, 默认为 0
类型	名称	描述																			
String	className																				
List<JSONObject>	明细	允许处理多条短信记录																			
<table border="1"> <thead> <tr> <th>类型</th> <th>名称</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>Int</td> <td>latoId</td> <td>唯一值, 默认为 0</td> </tr> </tbody> </table>			类型	名称	描述	Int	latoId	唯一值, 默认为 0													
类型	名称	描述																			
Int	latoId	唯一值, 默认为 0																			

String	Msg	短信内容
String	smsTemplate	短信模板
String	toNumber	发送号码
String	smsId	短信记录 ID, 可以根据这个值查询返回的状态, 查看是否发送成功.

3.14 失败返回值

3.14.1 令牌过期或服务器错误返回值

服务器如果失败, 则会返回 500 错误, 错误 Json 格式:

```
{  
  
  "Exception": {  
  
    "code": 1,  
  
    "msg": "连接超时, 或其他人登入, 请重新登入!",  
  
    "detail": "连接超时, 或其他人登入, 请重新登入!"  
  
  }  
  
}
```

类型	名称	描述
JSON	Exception	Json对象
Int	code	错误编码可以根据4.0 查看异常编码与内容
String	msg	异常摘要内容(抛出的异常服务器经过了Unicode编码)
String	detail	异常详细信息(抛出的异常服务器经过了Unicode编码)

4 异常列表

编码:300100 -- 数据表中存在记录，新增字段不能设置为主键.

编码:300101 -- 获取的模板 Json 出现错误

编码:300102 -- 不能删除或修改系统模板.

编码:300103 -- 不能删除系统模板.

编码:300104 -- 文件夹底下有模板，不能进行删除操作

编码:300105 -- 模板中有数据，不能进行删除.

- 编码:300106 -- 找不到指定模板
- 编码:300107 -- 不能查询成品模板
- 编码:300108 -- 不能修改成品模板
- 编码:300109 -- 不能删除成品模板
- 编码:300110 -- 不能使用多个数据源做为一个查询. 请重新选择模板.
- 编码:300111 -- 查询的数据表不属于该模板
- 编码:300112 -- 初始化系统模板错误
- 编码:300113 -- 模板名称或字段名称不能使用数字开头
- 编码:300114 -- 保存失败, 没有保存模板权限
- 编码:300115 -- 没查询到父模板
- 编码:300116 -- 模板名称不能为空. 请输入模板名称
- 编码:300117 -- 已有此模板名称! 请重新输入
- 编码:300118 -- 创建或者修改数据表结构时出错了
- 编码:300119 -- 数据表表名过长. 请重新输入数据表名称
- 编码:300120 -- 权限列表不允许出现重复名称!
- 编码:300121 -- 没有找到样式文件
- 编码:300122 -- 不能使用系统字段
- 编码:300123 -- 字段名称不允许存在特殊字符
- 编码:300124 -- 超出了模板的限制数

编码:300125 -- 没有找到表名表名

编码:300126 -- 没有找到外部数据源名称

编码:300127 -- 没有查询到该模板

编码:300128 -- 没有查询到[{0}]模板

编码:300129 -- 没有查询到要删除或修改的模板

编码:300130 -- 没有查询到要查询的模板

编码:300131 -- 没有查询到要删除或修改的流程实例的

编码:300132 -- 没有查询到要查询的流程实例的

编码:300133 -- 暂时没有{0}的索引类型

编码:300134 -- 创建索引时查询不到{0}字段.

编码:300135 -- 已经超出了限制索引数量, 只能创建 3 个或者 3 个以下的索引.

编码:300136 -- 删除模板失败.

编码:300137 -- 模板已被他人修改, 请关闭后重新打开编辑。

编码:300138 -- 保存[{0}]模板附件错误

编码:300139 -- 模板中找不到表的定义

编码:300140 -- 无效的表名

编码:300141 -- 找不到指定路径

编码:300142 -- 没有找到客户信息模板

编码:300143 -- 解析模板样式失败. 请检查模板是否发布成了浏览器版本

编码:300144 -- 有绑定 {0} 模板的策略，不能进行删除

编码:305139 -- 执行流程撤销错误

编码:305140 -- 流程模板没有绑定启动模板

编码:305141 -- 启动流程时发生了异常

编码:305142 -- 不能对表单明细执行启动流程操作

编码:305143 -- 查询到流程记录中不止一条记录

编码:305144 -- 查询到流程记录中不存在记录

编码:305145 -- 执行[启动流程]表达式

编码:305146 -- 在执行[启动流程]中没有填写执行人

编码:305147 -- 初始化流程模块出错

编码:305148 -- 请先部署流程

编码:305149 -- 流程步骤已被他人修改，撤销失败

编码:305150 -- 没有查询出想对应的流程模板明细，请联系软件开发商！

编码:305151 -- 没有查询到下一步执行人，请重新指定或者添加相对应条件的执行人

编码:305152 -- 同一个任务号查询出了多个任务，请联系软件开发商！

编码:305153 -- 指派人列表为空

编码:305154 -- workflow 分配 ID 出现错误

编码:305155 -- 找不到指定的任务

编码:305156 -- 任务指派人关系列表为空

编码:305157 -- 流程已经结束, 不能重新激活

编码:305158 -- 已有流程使用下步执行表单, 这种设计是不合理的, 请联系流程设计人员.

编码:305159 -- 交办人没有设置组织机构

编码:305160 -- 流程已经被停止, 不能进行操作此流程, 如需操作请联系流程管理员进行激活流程

编码:305161 -- 流程已启动成功, 不能重复启动.

编码:305162 -- 记录流程历史记录时, 流程实例发生了变化, 请联系软件提供商.

编码:305163 -- 删除流程错误

编码:305164 -- 获取指派人发生错误

编码:305165 -- 停止流程错误:

编码:305166 -- 待办事项查询出错, 查询不到待办事项.

编码:305167 -- 没有查找到历史记录

编码:305168 -- 没找到任务模板

编码:305169 -- 流程摘要不能为空

编码:305170 -- 没有实现 {0} 类型

编码:310169 -- 没有查询到要删除或修改的数据规范

编码:310170 -- 没有查询到指定数据规范

编码:310171 -- 数据规范名称不完整

编码:310172 -- 不允许添加和修改相同名称的数据规范

编码:310173 -- 已存在相同数据规范名称

编码:310174 -- 保存失败, 没有保存数据规范权限

编码:310175 -- 找不到数据规范

编码:310176 -- 数据接口名称为空

编码:310177 -- 不能删除或修改系统级的数据规范

编码:310178 -- 请传入全局数据接口

编码:315 -- 表单错误

编码:315178 -- 没有查询到要删除或修改的角色

编码:315179 -- 没有查询到要查询的角色

编码:315180 -- 没有查询到要删除或修改的用户分类

编码:315181 -- 没有查询到要查询的用户分类

编码:315182 -- 没有查询到要删除或修改的用户

编码:315183 -- 没有查询到要查询的用户

编码:315184 -- 没有查询到要删除或修改的组织机构

编码:315185 -- 没有查询到需要查询的组织机构

编码:315186 -- 不能讲系统管理员角色添加到岗位中

编码:315187 -- 已存在相同代码组织机构. 请重新输入!

编码:315188 -- 已存在相同用户名称, 请重新输入

编码:315189 -- 用户帐号不能进行修改

编码:315190 -- 已存在相同角色名称, 请重新输入!

编码:315191 -- 已存在相同目录, 请重新输入!

编码:315192 -- 不能添加相同的 EDI 名称

编码:315193 -- 创建计划任务用户不存在

编码:315194 -- 组织机构父节点 ID 为本节点, 不能进行保存. 请选择非本节点为父节点

编码:315195 -- 找不到当前登录用户所在的部门

编码:315196 -- 找不到指定的部门, 请检查该部门是否已经被删除.

编码:315197 -- 没有找到指定的岗位

编码:315198 -- 查询出多个角色, 角色名称出现同名情况

编码:315199 -- 没找到岗位, 查看是否删除

编码:315200 -- 不能有相同名称的消息服务

编码:315201 -- 组织机构创建失败

编码:315202 -- 这个用户是空的, 请管理员授权

编码:320200 -- 没有权限操作表单

编码:320201 -- 当前登入用户没有满足表单操作的策略

编码:320202 -- 系统表单不能够解锁

编码:320203 -- 请填写用户邮箱.

编码:320204 -- 保存 EDI 设置没设置邮箱.

编码:320205 -- 服务器已经有相同的 EDI 邮箱设置, 请设置其他邮箱

编码:320206 -- 没有发生变化, 服务器将不会执行任何操作

- 编码:320207 -- 没有查询到指定的表单
- 编码:320208 -- 请先保存表单再进行审核
- 编码:320209 -- 此表单审核过后不允许进行复审.
- 编码:320210 -- 不能进行重复审核
- 编码:320211 -- 不允许并发数小于 0
- 编码:320212 -- 不允 {0} 小于 0
- 编码:320213 -- 没有查询到主表
- 编码:320214 -- 没有查询到主表数据
- 编码:320215 -- 匹配符不能为空
- 编码:320216 -- 匹配不到相对应的操作符[{0}]
- 编码:320217 -- 您访问的表单不存在, 或者您没有权限访问该表单
- 编码:320218 -- 查询出多条数据, 请检查您的查询策略
- 编码:320219 -- 没有找到相对应的数据. 请检查查询条件是否错误
- 编码:320220 -- 数据项名称无效[{0}]
- 编码:320221 -- 执行锁定表达式出现错误
- 编码:320222 -- 执行条件判断[{0}]时出错
- 编码:320223 -- 无法将文本转换成时间类型: {0}
- 编码:320224 -- 执行数据规范[{0}]时出错
- 编码:320225 -- 查询语句语法错误

编码:320226 -- 表达式语法存在错误

编码:320227 -- 赋值操作出现字段与值不匹配情况.

编码:320228 -- 执行查询时没有查询到任何的数据源列表, 请检查是否指定了数据源.

编码:320229 -- 执行情况[{0}]时出现错误

编码:320230 -- 更新记录[{0}]时出现错误

编码:320231 -- 请检查传入的值与模板格式是否匹配

编码:320232 -- 没有获取到 FormId

编码:325231 -- 新插入记录不应该指定 ID 值

编码:325232 -- 已有人修改此表单, 请重新加载表单

编码:325233 -- 表单已被他人修改, 请关闭后重新打开编辑。

编码:325235 -- 一次只允许改变一个状态属性

编码:325236 -- 当前登入用户不能操作此步骤

编码:325237 -- 找不到系统变量[{0}]

编码:325238 -- 字符串转[{0}]换成 Double 类型失败

编码:325239 -- 数据表[{0}]中不存在名称为[{1}]的字段

编码:325240 -- 模板[{0}]中不存在指定的数据表: {1}

编码:325241 -- 电话号码[{0}]不正确, 请输入正确的号码!

编码:325242 -- 添加记录[{0}]时出现错误

编码:325243 -- 计算[{0}]的赋值表达式时出错

编码:325244 -- 表达式计算出错:[{0}]

编码:325245 -- 保存表单错误

编码:325246 -- 查询不到需要修改的记录

编码:325247 -- 同级文件夹是不能够重名

编码:325248 -- 无法识别的系统变量

编码:325249 -- 编号的长度不正确

编码:330246 -- 找不到指定的表单

编码:330247 -- 服务器找不到您要修改的表单，是否已经被他人删除或者请查看表单的编辑策略是否满足.

编码:330248 -- 找不到指定的

编码:330249 -- 找不到字段名称

编码:330250 -- 数据源中引用的数据规范 ID 为空

编码:330251 -- 数据源中引用的查询方法名称为空

编码:330252 -- 没有查询到附件

编码:330253 -- 没有找到相对应的历史节点

编码:330254 -- 获取的交易订单列表模板 Json 出现错误

编码:335254 -- 不能删除系统用户分类

编码:335255 -- 用户不能删除自己.

编码:335256 -- 文件夹底下有文件夹，不能进行删除操作.

编码:335257 -- 系统角色，不能进行删除或修改.

编码:335258 -- 不能删除系统用户

编码:335260 -- 该表单由工作流程产生,不能被删除,请先联系流程管理员将对应的流程删除之后,再来删除本表单,流程名称为:

编码:335261 -- 表单工作流还没有完成,所以不能进行删除

编码:335262 -- 没有删除系统数据的权限

编码:340263 -- json 解析出错

编码:340264 -- 获取的邮箱 Edi Json 出现错误

编码:345 -- 登入异常

编码:345267 -- 检查到超出限制模板数量,现在模板数量

编码:345268 -- 您不允许从该 {0} 地址登入.

编码:345269 -- 登录失败, 请检查账号密码是否正确

编码:345270 -- 名称与 url 数量不匹配

编码:345271 -- 在该帐套中没查询到用户对象

编??345272 -- 此企业帐套已经锁定

编码:345273 -- 客户端版本过低,请升级客户端

编码:345275 -- 没找到此帐套

编码:345276 -- 企业 ID 重复

编码:345277 -- 没有查询到该企业 ID

编码:345278 -- 地址错误

编码:345279 -- 此帐户出现异常

编码:345280 -- 企业帐套已经被锁定

编码:345281 -- 此用户已被锁定

编码:345282 -- 输入原密码错误!请输入正确原密码!

编码:345283 -- 服务器最大用户数为[{0}],当前登入账号不在用户列表之内,如需增加用户数,请及时联系您的软件服务商.

编码:345284 -- 传入的密码不符合密码规范,请传入正确的密码.

编码:345285 -- 您现在的客户端版本[{0}]过低,与服务器版本不匹配,请升级客户端版本至[{1}]以上版本.请联系您的软件提供商获取客户端升级包.

编码:345286 -- 错误:没找到 Token

编码:345287 -- {0}输入的应用 ID 与登入的用户应用 ID 不匹配

编码:345288 -- 链接超时,请重新登入

编码:350287 -- 获取指定的函数列表失败

编码:350288 -- 获取函数错误

编码:350289 -- 传入函数类型错误!不能获取函数列表

编码:350290 -- 时间转换错误

编码:350291 -- 计算函数'日期间隔'时出错,第一个参数无法识别:

编码:350292 -- 无法识别的业务函数

编码:350293 -- 字符串引号不匹配

编码:350294 -- 取子串的时候起始长度已经超过了字符串的长度.

编码:350295 -- 没有值进行对比

编码:350296 -- 传递类型错误

编码:355298 -- 暂时不支持超过两张表关联

编码:355300 -- 虚拟表不能执行该方法

编码:355301 -- 参数名和参数值不匹配

编码:355302 -- context 对象为空

编码:355303 -- 目标对象名称为空

编码:355304 -- 没有获取到这张表.

编码:355305 -- 没有查询到[{0}]对象.

编码:355306 -- 外部数据不能够执行修改操作. 不能当作目标对象.

编码:355307 -- 文本不能使用单引号请改用双引号

编码:355308 -- 无法识别的标识, 请检查名称是否拼写错误

编码:355309 -- 匹配条件不能为空

编码:355310 -- bizDataObject is null

编码:355311 -- parent bizDataObject is null

编码:355312 -- 不允许来自不同数据源的表关联查询

编码:355313 -- 执行业务公式[{0}]时发生错误:

编码:355314 -- 无法识别的属性{0}

编码:360315 -- 连接属性设置错误, 端口号或者地址或者数据库没有填写

编码:360316 -- 请选择短信通道

编码:360318 -- 所有号码发送成功;

编码:360319 -- 等待发送

编码:365320 -- 导入用户分类时出现错误

编码:365321 -- 导入用户角色时出现错误

编码:365322 -- 导入数据规范时出现错误

编码:365323 -- 导入文件夹时出现错误

编码:365324 -- 导入模板时出现错误

编码:365325 -- 导入模板保存模板获取 ID 时出错

编码:365326 -- 您传入的采集信息出现错误, 请传入正确的采集信息!

编码:365327 -- 请检查是否更新过了秘钥, 必须使用解密时使用的密码才能进行操作

编码:365328 -- 序列号不正确无法导入应用模板库, 请重新进行导入.

编码:365329 -- 导入数据规范保存时出现错误

编码:365330 -- 不允许在云服务器中生成序列号.

编码:365331 -- 不允许在云服务器中采集服务端信息.

编码:365332 -- 云端服务器不允许导入加密的应用模板库

编码:365333 -- 只允许采集本服务器的 C2V 文件

编码:365334 -- 解密失败

编码:365335 -- 解压失败

编码:365336 -- 请选择需要导入的模板库文件

编码:365337 -- 导入模板时没有获取到新的模板 ID

编码:365338 -- 请登入 admin 用户才能进行导出

编码:370334 -- 找不到指定的应用空间

编码:370335 -- 服务器最大应用空间数为[{0}], 当前使用的应用空间已超出授权范围。如需增加应用空间数量, 请及时联系您的软件服务商

编码:375336 -- 使用锁无法识别.

编码:375337 -- 查询不到硬件锁

编码:375338 -- 只允许主锁开发锁用户才能导出应用模板库.

编码:375339 -- 只允许主锁用户才能导出加密的应用模板库.

编码:375340 -- 加密锁未激活! 请联系软件开发商!

编码:375341 -- 没有查询到锁记录, 请插入锁.

编码:380342 -- 连接数据库错误

编码:380343 -- 不能创建数据源

编码:385343 -- 请同步本地时间, 超过了服务器接受的时间范围

编码:385344 -- appName 不可用

编码:385345 -- 签名验证不匹配, 请传入正确的签名

编码:385346 -- 外部接口暂时没有开放[{0}]

编码:390001 -- 未知错误异常

编码:395001 -- 请检查客户端版本是否与服务器版本匹配, 解析失败

编码:400001 -- 业务公式自定义异常错误

