

目 录

1.认证文档

1.1 应用接入流程

1.2 认证数据流程

2.认证接口说明

2.1 根据code获取token

2.2 刷新token

2.3 app登录接口

2.3 app注销接口

2.5 校验token是否有效

2.6 获取当前用户信息

2.7 用户注销

2.8 web端重定向到应用

2.9 app端获取所有应用

2.10 app端切换应用

2.11 修改账户密码

3.权限相关接口（md5验签模式）

3.1 获取用户信息

3.2 获取部门下的子部门

3.3 获取部门下的所有人员

3.4 获取角色下的所有人员

3.5 获取用户的功能权限

3.6 获取用户的表权限

3.7 获取用户的表字段权限

3.8 获取用户的表字段权限及附加条件

3.9 获取指定机构的下级节点

3.10 根据机构编号获取机构信息

4.权限相关接口（应用token模式）

4.1 获取应用的access_token

4.2 获取用户信息

4.3 获取部门下的子部门

4.4 获取部门下的所有人员

- 4.5 获取角色下所有人员
- 4.6 获取用户的功能权限
- 4.7 获取用户的表权限
- 4.8 获取用户的表字段权限
- 4.9 获取用户的表字段权限及附加条件
- 4.10 根据站点id和姓名获取用户
- 4.11 获取指定机构的下级节点
- 4.12 根据机构编号获取机构信息
- 4.13 分页查询机构信息
- 4.14 根据编号获取用户信息
- 4.15 获取多个用户编号对应的用户信息
- 4.16 根据站点名称获取站点信息
- 4.17 获取站点用户列表
- 4.18 新增部门
- 4.19 修改部门
- 4.20 删除部门
- 4.21 新增用户
- 4.22 修改用户
- 4.23 删除用户
- 5.部门增量推送信息
 - 5.1 新增/导入部门推送信息
 - 5.2 修改部门推送信息
 - 5.3 删除部门推送信息
- 6.用户增量推送信息
 - 6.1 新增/导入用户推送信息
 - 6.2 修改用户推送信息
 - 6.3 删除用户推送信息
- 7.行政区划增量推送信息
 - 7.1 新增行政区划推送信息
 - 7.2 修改行政区划推送信息
 - 7.3 删除行政区划推送信息
- 8.权限系统操作
- 9.SDK/demo下载

10. FAQ

11. code值对应表

12. 短信相关接口

12.1 获取所有可用模板

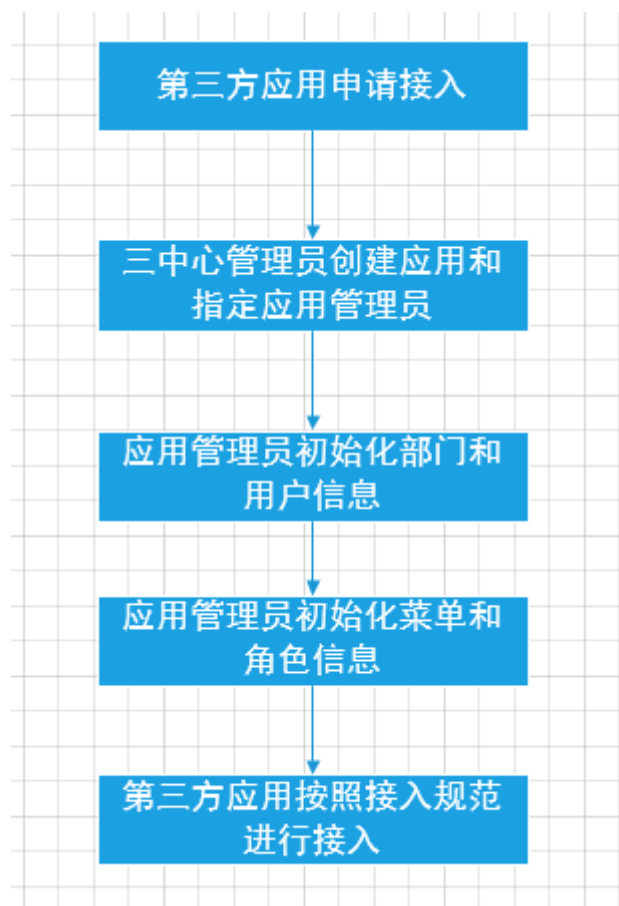
12.2 发送短信

12.3 发送定时短信

12.4 短信验证码校验

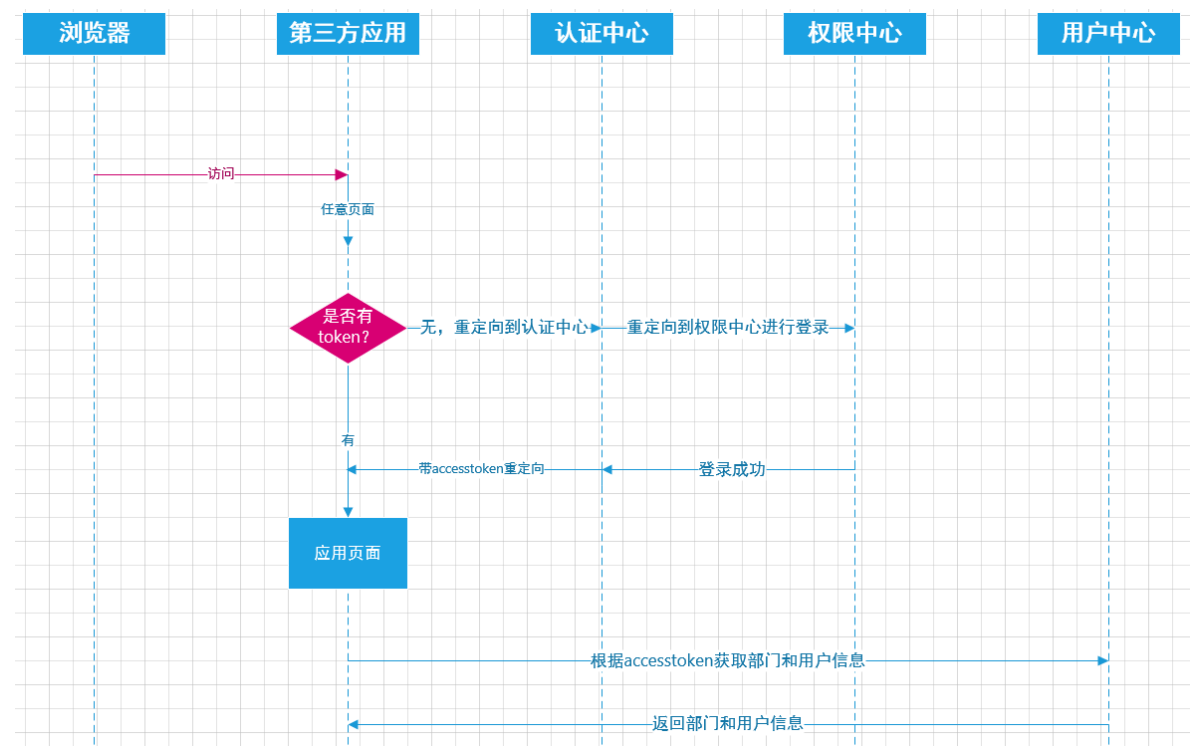
1.认证文档

1.1 应用接入流程



1.2 认证数据流程

app端的登录是根据接口传输用户名密码进行登录（接口的说明在流程3接口说明中体现）



2.认证接口说明

所有接口POST方式请求时，参数用application/x-www-form-urlencoded格式传输

2.1 根据code获取token

三中心门户跳转页进行应用跳转时，进行回调地址回调并附加code参数，此接口可以根据code获取token。

URL:

http://认证地址/oauth/token?client_id={client_id}&client_secret={client_secret}&code=\${code}&grant_type=authorization_code

HTTP请求方式:

get

参数信息:

参数名称	参数类型	必需	说明
client_id	String	是	client_id
client_secret	String	是	client_secret
code	String	是	回调地址中附加的code参数
grant_type	String	是	固定为authorization_code

```
curl -d 'client_id=${client_id}&client_secret=${client_secret}&code=${code}&grant_type=authorization_code' http://认证地址/oauth2/oauth/token
```

返回正确结果如下:

```
{
  "access_token": "919d21a2-82f2-4bd0-941f-b904065a3c9c",
  "token_type": "bearer",
  "refresh_token": "1e12b1b8-43f3-4407-b187-d9e1379d274c",
  "expires_in": 43196,
  "scope": "247"
}
```

返回错误结果如下:

```
{
  "code": "500",
  "message": "Token was not recognised",
  "data": null
}
```


2.2 刷新token

token会有有效期存在时间，expires_in为存活时间值，在token快过期时，可以通过刷新token获取到新的token

URL:

http://认证地址/oauth/token?grant_type=refresh_token&refresh_token={refresh_tokenvalue}&client_id={client_idvalue}&client_secret={client_secretvalue}

HTTP请求方式:

get

参数信息:

参数名称	参数类型	必需	说明
refresh_token	String	是	获取token时同时获取refresh_token
grant_type	String	是	固定用refresh_token
client_id	String	是	client_id
client_secret	String	是	client_secret

```
curl -d 'client_id=${client_id}&client_secret=${client_secret}&refresh_token=${refresh_token}&grant_type=refresh_token' http://认证地址//oauth2/oauth/token
```

返回正确结果如下:

```
{
  "access_token": "a7b62f9f-13a8-4856-aeb4-0dc62c51f1b3",
  "token_type": "bearer",
  "refresh_token": "b73ca711-4d9b-4981-bd61-415fe8d7c0d9",
  "expires_in": 43199,
  "scope": "269"
}
```

返回错误结果如下:

```
{
  "error": "invalid_client",
  "error_description": "Bad client credentials"
}
```

2.3 app登录接口

app端进行登录的接口。

URL: http://认证地址/oauth/token

HTTP请求方式: get/post

参数信息:

参数名称	参数类型	必需	说明
grant_type	String	是	授权方式，app端使用password方式
username	String	是	用户名
password	String	是	密码，需使用MD5加密
client_id	String	是	流程1.1申请到的client-id
client_secret	String	是	流程1.1申请到的client_secret

```
curl -d 'username=${username}&password=${password}&grant_type=password&client_id=${clientId}&client_secret=${clientSecret}' http://认证地址/oauth2/oauth/token
```

返回结果如下:

```
{
  "access_token": "919d21a2-82f2-4bd0-941f-b904065a3c9c",
  "token_type": "bearer",
  "refresh_token": "1e12b1b8-43f3-4407-b187-d9e1379d274c",
  "expires_in": 43196,
  "scope": "247"
}
```

返回结果参数说明:

参数名称	参数类型	说明
access_token	String	token凭证
token_type	String	token类型
refresh_token	String	刷新token
expires_in	String	token还可存活时间，单位是秒
scope	String	token绑定的站点id，不同的token此值不同

2.3 app登录接口

2.3 app注销接口

app端用户注销用户，先访问自身服务后端接口，后端接口访问此注销接口，app端不可直接访问，以防跨域。

URL:
http://认证地址/nonSessionBased/logout?token=tokenvalue

HTTP请求方式: get

参数信息:

参数名称	参数类型	必需	说明
token	String	是	用户token

```
curl -d 'token=${token}' http://认证地址/nonSessionBased/logout
```

返回正确结果如下:

```
{
  "code": "200",
  "message": "操作成功",
  "data": {
    "loginUrl": "http://10.12.107.33/oatuh2/login",
    "loginPage": "http://10.12.107.33/login"
  }
}
```

返回错误结果如下:

```
{
  "code": "500",
  "message": "Token was not recognised",
  "data": null
}
```

2.5 校验token是否有效

校验请求中 accessToken 的值是否有效。

URL： http://认证地址/oauth/check_token?token=tokenvalue

HTTP请求方式： get

参数信息：

参数名称	参数类型	必需	说明
token	String	是	token凭证

```
CURL: curl -d 'token=${token}' http://认证地址/oauth/check_token
```

返回结果如下：

```
{
  "aud": [
    "resource1"
  ],
  "user_name": "yqcs4",
  "scope": [
    "247"
  ],
  "active": true,
  "exp": 1576145398,
  "authorities": [
    "ADMIN"
  ],
  "client_id": "client1"
}
```

返回结果参数说明：

参数名称	参数类型	说明
user_name	String	用户名
scope	String	token绑定的站点id，不同的token此值不同
active	boolean	token 是否有效
exp	Long	token的失效时间戳，单位为秒
authorities	String[]	当前token对应的用户的权限角色
client_id	String	应用的clientId

2.5 校验token是否有效

2.6 获取当前用户信息

URL: http://认证地址/user/getUserInfo

HTTP请求方式: get/post

参数信息:

参数名称	参数类型	必需	说明
token	String	是	token凭证

```
curl -d 'token=${token}' http://认证地址/user/getUserInfo
```

返回结果如下:

```
{
  "code": "200",
  "message": "获取用户信息成功",
  "data": {
    "userId": 16,
    "userNo": "test",
    "userName": "测试用户",
    "deptId": 2,
    "deptName": "技术处",
    "sex": null,
    "ownRoles": "测试角色,普通用户",
    "currentSiteId": 266,
    "currentSiteName": "测试站点",
    "platformId": 72,
    "groupId": 8866,
    "fullPath": "组织/部门A"
  }
}
```

返回结果参数说明:

参数名称	参数类型	说明
userId	Long	用户ID
userNo	String	用户账户
userName	String	用户名
deptId	Long	部门ID
deptName	String	部门名称

2.6 获取当前用户信息

参数名称	参数类型	说明
sex	Short	性别
ownRoles	String	用户拥有的角色
currentSiteId	Long	当前站点id
currentSiteName	String	站点名称
platformId	Long	平台id
groupId	Long	用户组织id
fullPath	String	组织全路径名称

2.7 用户注销

web端用户直接重定向到此路径(<http://认证地址/logout>)，退出成功会重定向到认证登录地址，app端在此接口地址(http://认证地址/logout?json_response=true)

URL:

<http://认证地址/logout> (web重定向此地址)

http://认证地址/logout?json_response=true (app用此地址访问接口返回json数据)

HTTP请求方式:

get

参数信息:

参数名称	参数类型	必需	说明
json_response	String	否	app端需要此参数,web端不需要

```
curl -d 'json_response=true' http://认证地址/logout
```

返回结果如下:

app端:

```
{
  "totalCount":1,
  "data":
  {
    "loginUrl":"http://localhost:8081/oauth2/login",
    "loginPage":"http://localhost:9091/login/"
  },
  "success":true,
  "code":null,
  "message":"退出成功",
  "lastTime":"2020-05-14 14:17:36",
  "url":null
}
```

web端: 将重定向至登录页

2.8 web端重定向到应用

web端用户直接重定向到此地址

URL:

http://认证地址/sessionBased/switchSite/{clientId}/{siteId} (web重定向此地址)

HTTP请求方式:

get

参数信息:

参数名称	参数类型	必需	说明
clientId	String	是	应用的站点clientId
siteId	String	是	应用的站点id

返回结果: web端重定向到选择租户页面

2.9 app端获取所有应用

URL：http://认证地址/nonSessionBased/siteList

HTTP请求方式：

get/post

参数信息：

参数名称	参数类型	必需	说明
token	String	是	token凭证
json_response	String	是	app端需要此参数

```
curl -d 'token=${token}' http://认证地址/nonSessionBased/siteList
```

返回结果如下：

```
{
  "code": "200",
  "message": "获取用户站点列表成功",
  "data": {
    "userInfo": {
      "userId": 16,
      "userName": "测试用户",
      "isPlatformManager": false,
      "url": "http://www.baidu.com"
    },
    "publicSites": [
      {
        "projectId": 2,
        "projectNo": "测试站点",
        "projectName": "测试站点",
        "remark": "",
        "isSiteManager": false,
        "tokenUrl": "/error"
      }
    ],
    "innerSites": [
      {
        "projectId": 1,
        "projectNo": "platform",
        "projectName": "平台管控",
        "remark": "",
        "isSiteManager": false,
        "tokenUrl": "/error"
      }
    ]
  }
}
```

2.10 app端切换应用

门户app进入第三方app应用时调用，获取到需要进入的第三方app的token信息，才能进入

URL: http://认证地址/nonSessionBased/switchSite

HTTP请求方式: get/post

参数信息:

参数名称	参数类型	必需	说明
token	String	是	app端需要此参数
siteId	String	是	应用的站点id， 在3.6流程接口中获取到的projectId值
json_response	String	否	app端需要此参数，返回json数据

```
curl -d 'token=${token}&siteId=${siteId}' http://10.10.71.121:8081/oauth2/nonSessionBased/switchSite
```

返回结果如下:

```
{
  "code": "200",
  "message": "切换站点成功",
  "data": {
    "access_token": "5eea3800-51ff-4458-9f10-030a05e7c16f",
    "token_type": "bearer",
    "refresh_token": "c23d098c-d331-4bf2-a034-be50d731e797",
    "expires_in": 43199,
    "scope": "247"
  }
}
```

返回结果参数说明: 获取的是相关租户应用的token信息。

2.11 修改账户密码

通过使用用户的 `access token` 作为用户凭证，对用户账户的密码进行更改

URL: `http://认证地址/nonSessionBased/userService/changePwd`

HTTP请求方式: `post`

参数信息:

参数名称	参数类型	必需	说明
token	String	是	用户的 <code>access token</code>
oldPwd	String	是	原密码
newPwd	String	是	新密码

```
curl -d 'token=${token}&oldPwd=${oldPwd}&newPwd=${newPwd}' http://认证地址/nonSessionBased/userService/changePwd
```

成功返回结果示例:

```
{
  "code": "200",
  "message": "成功",
  "data": "密码更改成功"
}
```

失败返回结果示例

```
{
  "code": "500",
  "message": "该token(token value)属于由客户端授权模式获取的token, 不包含用户信息",
  "data": null
}
```

```
{
  "code": "500",
  "message": "无效的token",
  "data": null
}
```

```
{
  "code": "204",
  "message": "原密码错误!",
  "data": null
}
```

成功/失败 `code`为 200（成功）、其它`code`值（失败）失败时有`message`字段返回

返回参数data内容说明:

字段代码	字段名称	字段类型
code	返回code码	String
message	返回code码的说明	String
date	更为详细的补充说明	String

3.权限相关接口（md5验签模式）

此类接口与个人用户信息无关，所以无需token验证，部门、人员同步等接口，接口有验签规则.

注意：权限地址需要找认证维护人员要到，注意不是认证地址

3.1 获取用户信息

登陆租户系统后获取当前访问人员的权限信息包含个人基本信息、部门信息、租户信息、角色信息

URL: http://权限地址/v1/sys/user4sec/findById

HTTP请求方式: get

验签规则: MD5

参数信息

参数代码	参数名称	必选	字段类型	说明
projectId	租户ID	否	U16	没有租户ID传空串
userId	用户ID	是	U16	不允许为空
AccessKeyId	AppID	是	String	client-id, 在1.1流程中申请到的client-id
timeStamp	时间戳 (毫秒值)	是	U32	时间戳(毫秒值)
signature	签名	是	String	参照参数说明

参数说明:

signature加密需要注意参数顺序, signature加密需要注意参数顺序,
MD5(params=AccessKeyId=appIdKey&AccessKeySecret=accessKeySecretValue&projectId=requestProjectId&timeStamp=requestTimestamp&userId=requestUserId), 如果'租户ID'为null或者空串直接写为空串""进行拼接即可。accessKeySecretValue 为在1.1流程中申请到的client_secret

```
curl -d 'timeStamp=${time}&signature=${sign}&projectId=${projectId}&userId=${userId}&AccessKeyId=${keyId}' http://权限地址/v1/sys/user4sec/findById
```

返回结果如下:

```
{
  "data": {
    "originalUser": {
      "userId": 16,
      "groupId": 8866,
      "userGuid": "f99fdcf1-88b1-a35f-4e87-caf289e35143",
      "userNo": "test",
      "userName": "测试用户",
      "cardNo": "",
      "deptId": 886292,
      "dataStatus": 1,
      "json": "{}",
      "version": 274,
      "creator": "yqcs4",
      "createdTime": "2018-12-26 11:53:02",
```


3.1 获取用户信息

```
{
  "ownRoles": "中级用户",
  "postId": 16161616,
  "userType": "",
  "lawEnforcementId": "执法证号"
},
{
  "objId": "16",
  "userId": "16",
  "allPathName": "海关总署/杭州海关\\技术处",
  "dispName": "云擎测试4",
  "email": "null",
  "tel": "null",
  "groupId": "8866",
  "group": {
    "groupId": 8866,
    "pid": 1,
    "orgGuid": "bba02f17-b961-4f89-9b87-b2fbb0067268",
    "groupNo": "2900",
    "groupName": "杭州海关",
    "fullPath": "海关总署/杭州海关",
    "dataStatus": 1,
    "version": 2,
    "creator": "admin",
    "createTime": "2019-01-17 23:22:18"
  },
  "deptId": "886292",
  "dept": {
    "deptId": 886292,
    "groupId": 8866,
    "orgGuid": "9e6c55aa-7dd5-4463-83f4-7263844489c8",
    "deptName": "技术处",
    "fullPath": "技术处",
    "dataStatus": 1,
    "version": 3,
    "creator": "admin",
    "createTime": "2019-01-17 23:22:26",
    "areaId": 111,
    "deptLevel": 2222
  },
  "roles": [],
  "postId": 16161616,
  "userTypes": [
    "呵呵呵呵呵呵",
    "哦哦哦",
    "呵呵呵呵呵呵"
  ],
  "lawEnforcementId": "执法证号"
},
{
  "success": true,
  "lastTime": "2019-12-20 12:27:56"
}
```

失败返回结果示例

```
{
  "success": false,
  "code": 401,
  "message": "无效的应用Id!"
}
```

成功/失败 success为 true（成功）、false（失败）失败是有message字段返回

返回参数data内容说明：字段值为空时，字段不返回

字段代码	字段名称	字段类型	说明
originalUser	人员	参考user-model	
userId	用户ID	U16	
allPathName	全路径	String	
dispName	显示名称	String	
email	邮箱	String	
lastName	LastName	String	
tel	电话	String	
groupId	机构ID	String	
group	机构	参考Group-Model	
deptId	部门ID	String	
dept	部门	参考Deptment-Model	
Roles	角色	List 参考 role-model	
postId	职务标识	U16	
userTypes	用户类型	jsonArr,用户所有类型	
law_enforcement_id	执法人员编号	String	

Deptment-Model

字段代码	字段名称	字段类型
rojectId	目ID	16
eptId	部门ID	U16
roupId	组织ID	U16
pid	父ID	U32
deptCode	部门编号	String

3.1 获取用户信息

字段代码	字段名称	字段类型
deptName	部门名称	String
deptFunction	部门职能	String
deptType	部门类型	U8
telephone	联系电话	String
buildDate	成立日期	Date
dataStatus	数据状态	U8
json	JSON字段	String
version	版本	U16
creator	创建人	String
createdTime	创建时间	Time
updater	更新人	String
updatedAtTime	更新时间	Time
emark	备注	String
areald	行政区划标识	U8
dept_level	部门等级标识	U8

Group-Model

字段代码	字段名称	字段类型	字段类型
groupId	组织ID	U16	
districtId	行政区划ID	U16	
pid	父ID	U32	
groupNo	机构编号	String	
groupName	机构名称	String	
fullName	构全称	String	

字段代码	字段名称	字段类型	字段类型
shortName	机构简称	tring	
fullPath	全路径	String	
orderNo	序号	U16	
address	联系地址	String	
linkMan	联系人	String	
telephone	联系电话	String	
postCode	邮编	String	
dataStatus	数据状态	U8	0- 停用 1- 启用
json	JSON字段	String	
version	版本	U16	
creator	创建人	String	
createdTime	创建时间	Time	
updator	新人	String	
updatedTime	更新时间	Time	
remark	备注	String	

Role-Model

字段代码	字段名称	字段类型	说明
roleId	角色ID	U16	
roleCode	角色代码	String	
roleName	角色名称	String	
dataStatus	数据状态	U8	

User-Model

字段代码	字段名称	字段类型	说明
userId	用户ID	U16	
groupId	组织ID	U16	
userNo	用户编号	String	
userName	用户名称	String	
cardNo	员工卡号	String	
sex	性别	U8	
deptId	部门ID	U16	
telephone	联系电话	String	
address	联系地址	String	
email	邮箱	String	
birthday	出生年月	Date	
idCard	身份证号	String	
age	年龄	U8	
theme	界面主题	String	
authStartDate	权限开始日期	Date	
authEndDate	权限结束日期	Date	
dataStatus	数据状态	U8	
json	JSON字段	String	
version	版本	U16	
creator	创建人	String	
createdTime	创建时间	Time	
updater	更新人	String	
updatedTime	更新时间	Time	

3.1 获取用户信息

字段代码	字段名称	字段类型	说明
remark	备注	String	
officeTel	办公室电话	String	
ownRoles	已有角色	String	

3.2 获取部门下的子部门

获取某一个部门下的子部门，不包含子孙部门

URL: http://权限地址/v1/sys/datac/datacUser/getDirectDeptsByDeptId

HTTP请求方式: post、get

验签规则: MD5

参数信息:

参数名称	参数类型	必需	说明
AccessKeyId	String	是	是 client-id，在1.1流程中申请到的client-id
timeStamp	String	是	是 时间戳(毫秒值)
signature	String	是	签名
projectId	String	是	站点id
deptId	String	是	部门ID，允许为空值， 获取所有部门信息设定deptId为""

参数说明:

signature加密需要注意参数顺序(),

MD5(params=AccessKeyId=\${key}&AccessKeySecret=\${secret}&deptId=\${deptId}&projectId=\${projectId}&timeStamp=\${time})

accessKeySecretValue 为在1.1流程中申请到的client_secret

```
curl -d 'AccessKeyId=${keyId}&timeStamp=${time}&signature=${sign}&projectId=${projectId}&deptId=${deptId}' http://权限地址/v1/sys/datac/datacUser/getDirectDeptsByDeptId
```

成功返回结果示例:

```
{
  "data": [
    {
      "deptId": 12072,
      "groupId": 1,
      "pid": 1,
      "deptCode": "12072",
      "deptName": "城建集团",
      "fullPath": "市级/市数字城管/城建集团",
      "orderNo": 22,
      "dataStatus": 1,
      "json": {},
      "creator": "zsadmin",
      "createTime": "2019-12-05 00:00:00"
    }
  ],
}
```

3.2 获取部门下的子部门

```
{
  "deptId": 12073,
  "groupId": 1,
  "pid": 1,
  "deptCode": "12073",
  "deptName": "宜宾博世安防",
  "fullPath": "市级/市数字城管/宜宾博世安防",
  "orderNo": 23,
  "dataStatus": 1,
  "json": {},
  "creator": "zsadmin",
  "createTime": "2019-12-05 00:00:00"
},
{
  "deptId": 12074,
  "groupId": 1,
  "pid": 1,
  "deptCode": "12074",
  "deptName": "临港经开区指挥中心",
  "fullPath": "市级/市数字城管/临港经开区指挥中心",
  "orderNo": 24,
  "dataStatus": 1,
  "json": {},
  "creator": "zsadmin",
  "createTime": "2019-12-05 00:00:00"
},
{
  "deptId": 12107,
  "groupId": 1,
  "pid": 1,
  "deptCode": "12107",
  "deptName": "翠屏区指挥中心",
  "fullPath": "市级/市数字城管/翠屏区指挥中心",
  "orderNo": 25,
  "dataStatus": 1,
  "json": {},
  "creator": "zsadmin",
  "createTime": "2019-12-05 00:00:00"
},
{
  "deptId": 12180,
  "groupId": 1,
  "pid": 1,
  "deptCode": "12180",
  "deptName": "珙县指挥中心",
  "fullPath": "市级/市数字城管/珙县指挥中心",
  "orderNo": 26,
  "dataStatus": 1,
  "json": {},
  "creator": "zsadmin",
  "createTime": "2019-12-05 00:00:00"
},
{
  "deptId": 12217,
  "groupId": 1,
  "pid": 1,
  "deptCode": "12217",
  "deptName": "叙州区指挥中心",
  "fullPath": "市级/市数字城管/叙州区指挥中心",
  "orderNo": 27,
  "dataStatus": 1,
  "json": {}
}
```


3.2 获取部门下的子部门

```
{
  "creator": "zsadmin",
  "createdTime": "2019-12-05 00:00:00",
},
{
  "deptId": 12292,
  "groupId": 1,
  "pid": 1,
  "deptCode": "12292",
  "deptName": "南溪区指挥中心",
  "fullPath": "市级/市数字城管/南溪区指挥中心",
  "orderNo": 28,
  "dataStatus": 1,
  "json": {},
  "creator": "zsadmin",
  "createdTime": "2019-12-05 00:00:00"
}
],
"success": true
}
```

失败返回结果示例

```
{
  "success": false,
  "code": 401,
  "message": "无效的应用Id!"
}
```

成功/失败 success为 true（成功）、false（失败）失败时有message字段返回

返回参数data内容说明： 字段值为空时，字段不返回

字段代码	字段名称	字段类型
deptId	部门ID	U16
pid	父部门ID	U16
deptCode	部门编号	String
deptName	部门名称	String
orderNo	排序	U16
dataStatus	数据状态 1有效 0无效	U8
creator	创建人	U16
createdTime	创建时间	Time
json	JSON字段	String
groupId	组织id	U16

3.2 获取部门下的子部门

3.3 获取部门下的所有人员

获取某一个部门下的所有人员仅限于当前部门下人员不包含子部门的人员。

URL: http://权限地址/v1/sys/datac/datacUser/getDirectUsersByDeptId

HTTP请求方式: get

验签规则: MD5

参数信息:

参数名称	参数类型	必需	说明
projectId	String	是	站点id
deptId	String	是	部门ID 不可为空值
AccessKeyId	String	是	是 client-id, 在1.1流程中申请到的client-id
timeStamp	String	是	是 时间戳(毫秒值)
signature	String	是	签名

参数说明:

signature加密需要注意参数顺序,

MD5(params=AccessKeyId=\${key}&AccessKeySecret=\${secret}&deptId=\${deptId}&projectId=\${projectId}&timeStamp=\${time})。

accessKeySecretValue 为在1.1流程中申请到的client_secret

```
curl -d 'AccessKeyId=${keyId}&timeStamp=${time}&signature=${sign}&projectId=${projectId}&deptId=${deptId}' http://权限地址/v1/sys/datac/datacUser/getDirectUsersByDeptId
```

成功返回结果示例:

```
{
  "data": [
    {
      "creator": "admin",
      "userNo": "abc",
      "deptId": 14017,
      "dataStatus": 1,
      "json": "{}",
      "createTime": "2020-05-27 12:43:11",
      "userName": "userA",
      "userId": 1001722
    },
    {
      "creator": "admin",
      "userNo": "def",
      "sex": 1,
    }
  ]
}
```

3.3 获取部门下的所有人员

```
{
  "deptId": 14017,
  "dataStatus": 1,
  "json": "{}",
  "createTime": "2020-05-27 11:33:06",
  "userName": "userB",
  "userId": 1001721
},
{
  "success": true
}
```

失败返回结果示例

```
{
  "success": false,
  "code": 401,
  "message": "无效的应用Id!"
}
```

成功/失败 success为 true（成功）、false（失败）失败时有message字段返回

返回参数data内容说明： 字段值为空时，字段不返回

字段代码	字段名称	字段类型
userId	用户ID	U16
userGuid	来源标识ID	String
userNo	员工号/账号	String
userName	用户名	String
cardNo	卡号	String
sex	性别	String
deptId	部门ID	U16
telephone	电话	String
address	地址	String
email	邮箱	String
birthday	出生日期	Date
idCard	身份证号	String
age	年龄	Int

3.3 获取部门下的所有人员

字段代码	字段名称	字段类型
theme	主题	String
dataStatus	数据状态 1有效 0无效	U8
creator	创建人	U16
createdTime	创建时间	Date
updater	更新人	U16
updatedTime	更新时间	Date
remark	备注	String
json	JSON字段	String
authStartDate	权限开始日期	Date
endAuthEndDate	权限结束日期	Date
officeTel	办公室电话	String
userGrade	用户等级	U8
type	用户类型	U8
iconFileId	头像id	String
iconFilePath	头像路径	String
enabled	活动用户标识： 0-非当前活动用户 1-当前活动用户	U8

备注：

接口返回数据有缓存缓存时效默认30分钟

3.4 获取角色下的所有人员

获取角色编码、站点id获取该角色下所有人员。

URL: http://权限地址/v1/sys/user4sec/findUsersById

HTTP请求方式: get

验签规则: MD5

参数信息:

参数名称	参数类型	必需	说明
roleCode	String	是	角色编码
projectId	String	是	站点id
AccessKeyId	AppID	是	String
timeStamp	时间戳(毫秒值)	是	U32
signature	签名	是	String

参数说明:

signature加密需要注意参数顺序，signature加密需要注意参数顺序，
MD5(params=AccessKeyId=\${apKeyId}&AccessKeySecret=\${accessKeySecretValue}&projectId=\${projectId}&roleCode=\${code}&timeStamp=\${requestTimestamp})
accessKeySecretValue 为在1.1流程中申请到的client_secret。

```
curl -G -d 'AccessKeyId=${keyId}&projectId=${projectId}&roleCode=${role}&timeStamp=${time}&signature=${sign}' http://权限地址/v1/sys/user4sec/findUsersById
```

成功返回结果示例:

```
{
  "code": "200",
  "message": "200",
  "data": [
    {
      "deptName": "测试机构",
      "deptId": 886530,
      "userName": "田乐宇",
      "userId": 31386
    },
    {
      "deptName": "总署机关",
      "deptId": 887612,
      "userName": "徐丹平",
      "userId": 8090
    }
  ]
}
```

3.4 获取角色下的所有人员

```
    }  
  ]  
}
```

失败返回结果示例

```
{  
  "code": "500",  
  "message": "您传输的参数有误",  
  "data": null  
}
```

成功/失败 success为 true（成功）、false（失败）失败时有message字段返回

返回参数data内容说明： 字段值为空时，字段不返回

字段代码	字段名称	字段类型
deptName	部门名称	String
deptId	部门ID	int
userId	用户id	int
userName	用户名	String

3.5 获取用户的功能权限

登陆租户系统后获取当前访问人员具有权限的菜单和功能

URL: http://权限地址/v1/sys/user4sec/getProjectFunsByUserIdAndProjectId

HTTP请求方式: get

验签规则: MD5

参数信息

参数代码	参数名称	必选	字段类型	说明
projectId	租户ID	否	U16	没有租户ID传空串
userId	用户ID	是	U16	不允许为空
AccessKeyId	AppID	是	String	client-id, 在1.1流程中申请到的client-id
timeStamp	时间戳 (毫秒值)	是	U32	时间戳(毫秒值)
signature	签名	是	String	参照参数说明

参数说明:

signature加密需要注意参数顺序, signature加密需要注意参数顺序,
MD5(params=AccessKeyId=appIdKey&AccessKeySecret=accessKeySecretValue&projectId=requestProjectId&timeStamp=requestTimestamp&userId=requestUserId), 如果'租户ID'为null或者空串直接写为空串""进行拼接即可, accessKeySecretValue 为在1.1流程中申请到的client_secret。

```
curl -d 'timeStamp=${time}&signature=${sign}&projectId=${projectId}&userId=${userId}&AccessKeyId=${keyId}' http://权限地址/v1/sys/user4sec/getProjectFunsByUserIdAndProjectId
```

成功返回结果如下:

```
{
  "data": [
    {
      "id": "60ee8395cf404d4da578febbd6fad5c0_2",
      "text": "测试菜单",
      "leaf": false,
      "expanded": true,
      "children": [
        {
          "id": "b958a2c9cbc44fc3aacbb52fdce8ef09_2",
          "text": "测试子菜单",
          "leaf": true,
          "expanded": true,
          "nodeType": 1,

```



```
        "systemType": 0
      }
    ],
    "nodeType": 1,
    "systemType": 0
  }
],
"success": true
}
```

失败返回结果示例

```
{
  "success": false,
  "code": 401,
  "message": "无效的应用Id!"
}
```

返回参数data内容说明：字段值为空时，字段不返回

字段代码	字段名称	字段类型	说明
id	功能编号	U16	
text	功能名称	String	
url	链接	String	
expanded	是否展开	U8	
orderNo	序号	U16	
helpUrl	帮助地址	String	
nodeType	功能类型	S8	0-顶级菜单 1-节点 2-叶子 3-按钮
icon	图标	String	
systemType	系统类型	S8	
leaf	是否叶子节点	S8	True: 是 false 否
openNew	是否弹出菜单	S8	True: 是 false 否
children	子对象		当前对象列表

3.6 获取用户的表权限

根据用户ID、ClientId或者租户ID、数据源ID获取用户具有权限的表。

URL: http://权限地址/v1/sys/user4sec/getDataPermissionsOfTableByUser

HTTP请求方式: get/post

验签规则: MD5

参数信息:

参数代码	参数名称	必选	字段类型	说明
datasource	数据源ID	是	String	
projectId	租户ID	否	U16	没有传空串
userId	用户ID	是	String	不允许为空
AccessKeyId	AppID	是	String	client-id, 在1.1流程中申请到的client-id
timeStamp	时间戳 (毫秒值)	是	U32	时间戳(毫秒值)
signature	签名	是	String	参考参数说明

参数说明:

signature加密需要注意参数顺序, signature加密需要注意参数顺序,
MD5(params=AccessKeyId=appIdKey&AccessKeySecret=accessKeySecretValue&datasource=requestDataSource&projectId=requestProjectId&timeStamp=requestTimestamp&userId=requestUserId), 如果'数据源ID'或者'租户ID'为null或者空串直接写为空串""进行拼接即可, accessKeySecretValue 为在1.1流程中申请到的client_secret。

```
curl -d 'timeStamp=${time}&signature=${sign}&projectId=${projectId}&userId=${userId}&AccessKeyId=${keyId}&datasource=${master}' http://权限地址/v1/sys/user4sec/getDataPermissionsOfTableByUser
```

返回结果如下:

```
{
  "data": [
    "9f723116332044a4bd049e50d93f7016",
    "7ce2f6424969482ab425f62ded2ca078",
    "8a80f4075bb0440b9ca4ddff30521388",
    "e27c6ae2a98f41dfa2c84cba9ee4eac4",
    "b87b67ab8d4742579b32c88268bc2df3",
    "83ed3aa946784ebf8ac1a2aac0d0af3d",
    "92a10c587d4348ba98a5ab9608c96ae2",
    "29620c81342f47c087ea7a37a2d645e8",
  ]
}
```

3.6 获取用户的表权限

```
    "ee16c0321ea84ba2938cac33c14ce3ee",  
    "8ac3b68918ee4fca80f08c5f6095e520",  
    "e0c2c50b8caa4f998bb00071887c8a97",  
    "31b31e49624b45c9b60ef87d5d956334",  
    "0b3f7bd529b04fa68508ad0e1f8af891",  
    "cf093b9ac03a44f38ed338ee8f587f82"  
  ],  
  "success": true,  
  "lastTime": "2019-10-15 15:49:00"  
}
```

3.7 获取用户的表字段权限

根据用户ID、ClientId或者租户ID、数据源ID获取用户具有的某张表的字段权限。

URL: http://权限地址/v1/sys/user4sec/getDataPermissionsOfFieldByUserAndTable

HTTP请求方式: get

验签规则: MD5

参数信息:

参数代码	参数名称	必选	字段类型	说明
datasource	数据源ID	是	String	
projectId	租户ID	否	U16	没有传空串
tableValue	表ID	是	String	
userId	用户ID	是	String	不允许为空
AccessKeyId	ApplD	是	String	client-id, 在1.1流程中申请到的client-id
timeStamp	时间戳 (毫秒值)	是	U32	时间戳(毫秒值)
signature	签名	是	String	参考参数说明

参数说明:

signature加密需要注意参数顺序, signature加密需要注意参数顺序,
MD5(params=AccessKeyId=appldKey&AccessKeySecret=accessKeySecretValue&datasource=requestDatasource&projectId=requestProjectId&tableValue=tableValueRequest&timeStamp=requestTimestamp&userId=requestUserId)
accessKeySecretValue 为在1.1流程中申请到的client_secret。

```
curl -d 'timeStamp=${time}&signature=${sign}&projectId=${projectId}&userId=${userId}&AccessKeyId=${keyId}&datasource=${source}&tableValue=${value}' http://权限地址/v1/sys/user4sec/getDataPermissionsOfFieldByUserAndTable
```

成功返回结果示例:

```
{
  "data": [
    "VALID_DATE",
    "ID_NUMBER",
    "LAW_MAN",
    "LAW_MAN_TEL",
    "REG_CO",
    "RG_FUND",
    "ACCO_NO",
```

3.7 获取用户的表字段权限

```
    "EN_FULL_CO",
    "BROKER_TYPE",
    "BUSI_TYPE",
    "FULL_NAME",
    "ADDR_CO",
    "CO_CLASS",
    "APPR_ID",
    "APPR_DEP",
    "ACCO_BANK",
    "TEL_CO",
    "RG_DATE",
    "EN_ADDR_CO"
  ],
  "success": true,
  "lastTime": "2019-10-15 15:55:18"
}
```

失败返回结果示例

```
{
  "success": false,
  "code": 401,
  "message": "无效的应用Id!"
}
```

成功/失败 success为 true（成功）、false（失败）失败是有message字段返回

3.8 获取用户的表字段权限及附加条件

根据用户ID、ClientId或者租户ID、数据源ID以及机构编码获取用户具有表的字段权限和字段授权条件。

URL： http://权限地址/v1/sys/user4sec/getDataPermissionsOfTableField

HTTP请求方式： get

验签规则： MD5

参数信息：

参数代码	参数名称	必选	字段类型	说明
groupNo	机构编号	是	String	
datasource	数据源ID	是	String	不允许为空
projectId	租户ID	否	U16	没有传空串
tableValue	表ID	是	String	
userId	用户ID	是	String	不允许为空
AccessKeyId	AppID	是	String	client-id， 在1.1流程中申请到的client-id
timeStamp	时间戳 (毫秒值)	是	U32	时间戳(毫秒值)
signature	签名	是	String	参考参数说明

参数说明：

signature加密需要注意参数顺序，signature加密需要注意参数顺序，
MD5(params=AccessKeyId=appIdKey&AccessKeySecret=accessKeySecretValue&datasource=requestDatasource&groupNo=requestGroupno&projectId=requestProjectId&tableValue=tableValueRequest&timeStamp=requestTimestamp&userId=requestUserId)
'租户ID'为null或者空串直接写为空串""进行拼接即可，
accessKeySecretValue 为在1.1流程中申请到的client_secret。

```
curl -d 'AccessKeyId=${keyId}&datasource=${source}&groupNo=${groupNo}&projectId=${projectId}&tableValue=${vaule}&timeStamp=${time}&userId=${userId}&signature=${sign}' http://权限地址/v1/sys/user4sec/getDataPermissionsOfTableField
```

成功返回结果示例：

```
{
  "data": {
```

3.8 获取用户的表字段权限及附加条件

```
{
  "showFields": [
    "VALID_DATE",
    "EN_FULL_CO",
    "BROKER_TYPE",
    "BUSI_TYPE",
    "FULL_NAME",
    "ADDR_CO",
    "EXAM_SCORE",
    "CUSTOMS_CODE",
    "CONTAC_CO",
    "CREDIT_MAR",
    "LICENSE_ID",
    "MAIL_CO",
    "CURR_CODE",
    "TRADE_CO",
    "INV_FUND_T",
    "CO_CLASS",
    "APPR_ID",
    "APPR_DEP",
    "ACCO_BANK",
    "TEL_CO",
    "RG_DATE",
    "EN_ADDR_CO"
  ],
  "fieldWheres": [{
    "fieldName": "CUSTOMS_CODE",
    "fieldOp": "LIKE",
    "fieldCon": "29%",
    "dataGroupId": "ccuap_org"
  }]
},
"success": true,
"lastTime": "2019-10-15 16:00:23"
}
```

**

失败返回结果示例**

```
{
  "success": false,
  "code": 401,
  "message": "无效的应用Id!"
}
```

成功/失败 success为 true（成功）、false（失败）失败是有message字段返回

3.9 获取指定机构的下级节点

获取某一个机构下的子机构，不包含子孙机构

URL: http://权限地址/v1/sys/datac/datacUser/findNextChildGroupNodeTree

HTTP请求方式: post、get

验签规则: MD5

参数信息:

参数名称	参数类型	必需	说明
AccessKeyId	String	是	是 client-id，在1.1流程中申请到的client-id
timeStamp	String	是	是 时间戳(毫秒值)
signature	String	是	签名
groupId	String	是	机构id，允许为空值，获取一级机构信息 groupId传参""

参数说明:

signature加密需要注意参数顺序(),

MD5(params=AccessKeyId=\${key}&AccessKeySecret=\${secret}&groupId=\${groupId}&timeStamp=\${time})

accessKeySecretValue 为在1.1流程中申请到的client_secret

```
curl -d 'AccessKeyId=${keyId}&timeStamp=${time}&signature=${sign}&groupId=${groupId}' http://权限地址/v1/sys/datac/datacUser/findNextChildGroupNodeTree
```

成功返回结果示例:

```
{
  "data": [
    {
      "groupId": 2,
      "pid": 1,
      "groupNo": "111",
      "groupName": "111",
      "fullPath": "全部/111",
      "dataStatus": 1,
      "json": {},
      "version": 0,
      "creator": "tc_admin",
      "createTime": "2020-06-16 14:16:12"
    }
  ],
  "success": true,
  "lastTime": "2020-06-16 14:52:12"
}
```


失败返回结果示例

```
{
  "success": false,
  "code": 401,
  "message": "无效的应用Id!"
}
```

成功/失败 success为 true（成功）、false（失败）失败时有message字段返回

返回参数data内容说明： 字段值为空时，字段不返回

字段代码	字段名称	字段类型
groupId	组织ID	U16
districtId	行政区划ID	U16
pid	父ID	U32
groupNo	机构编号	String
groupName	机构名称	String
fullName	机构全称	String
shortName	机构简称	String
fullPath	全路径	String
orderNo	序号	U16
address	联系地址	String
linkMan	联系人	String
telephone	联系电话	String
postCode	邮编	String
dataStatus	数据状态	U8
json	JSON字段	String
version	版本	U16
creator	创建人	String
createdTime	创建时间	Time

3.9 获取指定机构的下级节点

字段代码	字段名称	字段类型
updater	更新人	String
updatedAt	更新时间	Time
remark	备注	String

3.10 根据机构编号获取机构信息

根据机构编号获取机构详情

URL: http://权限地址/v1/sys/datac/datacUser/findGroupByNo

HTTP请求方式: post、get

验签规则: MD5

参数信息:

参数名称	参数类型	必需	说明
AccessKeyId	String	是	是 client-id, 在1.1流程中申请到的client-id
timeStamp	String	是	是 时间戳(毫秒值)
signature	String	是	签名
groupNo	String	是	机构编号

参数说明:

signature加密需要注意参数顺序(),

MD5(params=AccessKeyId=\${key}&AccessKeySecret=\${secret}&groupNo=\${groupNo}&timeStamp=\${time})

accessKeySecretValue 为在1.1流程中申请到的client_secret

```
curl -d 'AccessKeyId=${keyId}&timeStamp=${time}&signature=${sign}&groupNo=${groupNo}' http://权限地址/v1/sys/datac/datacUser/findGroupByNo
```

成功返回结果示例:

```
{
  "data": {
    "groupId": 1,
    "orgGuid": "94480300-fca2-4268-86c2-4c13963bf4cb",
    "groupNo": "0000",
    "groupName": "全部",
    "fullPath": "全部",
    "dataStatus": 1,
    "json": {},
    "version": 1,
    "creator": "初始化",
    "createdTime": "2020-05-28 13:52:07"
  },
  "success": true,
  "lastTime": "2020-06-16 14:59:37"
}
```

失败返回结果示例

```
{
  "success": false,
  "code": 401,
  "message": "无效的应用Id!"
}
```

成功/失败 success为 true（成功）、false（失败）失败时有message字段返回

返回参数data内容说明： 字段值为空时，字段不返回

字段代码	字段名称	字段类型
groupId	组织ID	U16
districtId	行政区划ID	U16
pid	父ID	U32
groupNo	机构编号	String
groupName	机构名称	String
fullName	机构全称	String
shortName	机构简称	String
fullPath	全路径	String
orderNo	序号	U16
address	联系地址	String
linkMan	联系人	String
telephone	联系电话	String
postCode	邮编	String
dataStatus	数据状态	U8
json	JSON字段	String
version	版本	U16
creator	创建人	String
createdTime	创建时间	Time
updater	更新人	String

3.10 根据机构编号获取机构信息

字段代码	字段名称	字段类型
updatedTime	更新时间	Time
remark	备注	String

4.权限相关接口（应用token模式）

此类接口与个人用户信息无关，需要通过`access_token`来认证。

注意：权限地址需要找认证维护人员要到，注意不是认证地址

4.1 获取应用的access_token

通过该接口获取到与用户信息无关的access_token值，然后用这个值来请求以下接口

URL: http://认证地址/oauth/token

HTTP请求方式: get/post

参数信息:

参数名称	参数类型	必需	说明
grant_type	String	是	授权方式，client_credentials
client_id	String	是	流程1.1申请到的client-id
client_secret	String	是	流程1.1申请到的client_secret

curl 请求模版，请自行替换参数

```
curl -d "grant_type=client_credentials&client_id=${clientId}&client_secret=${clientSecret}" "http://认证地址/oauth/token"
```

返回结果如下:

```
{
  "access_token": "919d21a2-82f2-4bd0-941f-b904065a3c9c",
  "token_type": "bearer",
  "refresh_token": "1e12b1b8-43f3-4407-b187-d9e1379d274c",
  "expires_in": 43196,
  "scope": "247"
}
```

返回结果参数说明:

参数名称	参数类型	说明
access_token	String	token凭证
token_type	String	token类型
expires_in	String	token还可存活时间，单位是秒
scope	String	token绑定的站点id，不同的token此值不同

4.2 获取用户信息

登陆租户系统后获取当前访问人员的权限信息包含个人基本信息、部门信息、租户信息、角色信息

URL: http://权限地址/v1/sys/user4sec/findById

HTTP请求方式: get

验签规则: access_token

参数信息

参数代码	参数名称	必选	字段类型	说明
access_token	令牌	是	String	在4.1取到的access_token值
projectId	租户ID	必选	U16	租户ID
userId	用户ID	是	U16	

```
curl -d 'access_token=${token}&projectId=${projectId}&userId=${userId}' http://权限地址/v1/sys/user4sec/findById
```

参数说明:

如果'用户ID'或者'租户ID'为null或者空串直接写为空串""进行拼接即可。

成功返回结果如下:

```
{
  "data": {
    "originalUser": {
      "userId": 4,
      "groupId": 1,
      "userNo": "zjl",
      "userName": "userZjl",
      "deptId": 1,
      "dataStatus": 1,
      "json": {},
      "version": 0,
      "creator": "admin",
      "createTime": "2020-05-26 16:42:17"
    },
    "objId": "4",
    "userId": "4",
    "allPathName": "全部\\测试部门\\测试",
    "dispName": "zjl",
    "email": "null",
    "tel": "null",
    "groupId": "1",
    "group": {
      "groupId": 1,

```


4.2 获取用户信息

```
{
  "orgGuid": "94480300-fca2-4268-86c2-4c13963bf4cb",
  "groupNo": "0000",
  "groupName": "全部",
  "fullPath": "全部",
  "dataStatus": 1,
  "json": {},
  "version": 1,
  "creator": "初始化",
  "createTime": "2020-05-26 15:56:55"
},
{
  "deptId": "1",
  "dept": {
    "deptId": 1,
    "groupId": 1,
    "deptCode": "BM-TEST",
    "deptName": "测试部门",
    "fullPath": "测试部门",
    "dataStatus": 1,
    "json": {},
    "version": 0,
    "creator": "admin",
    "createTime": "2020-05-26 16:14:44"
  },
  "roles": [{
    "originalRole": {
      "projectId": 151,
      "roleId": 103,
      "roleCode": "LOG_AUDITOR_ROLE",
      "roleName": "日志审计员角色",
      "dataStatus": 1,
      "json": "{}",
      "version": 16,
      "creator": "yqcs4",
      "createTime": "2019-05-10 10:22:10",
      "grantUsers": "云擎测试"
    },
    "objId": "103",
    "code": "LOG_AUDITOR_ROLE",
    "name": "日志审计员角色",
    "funcPermissions": []
  }]
},
{
  "success": true,
  "lastTime": "2019-12-09 10:24:45"
}
```

失败返回结果示例

```
{
  "success": false,
  "code": 401,
  "message": "无效的应用Id!"
}
```

成功/失败 success为 true（成功）、false（失败）失败时有message字段返回

返回结果字段说明：

字段代码	字段名称	字段类型
originalUser	人员	参考user-model
userId	用户ID	U16
allPathName	全路径	String
dispName	显示名称	String
email	邮箱	String
firstName	FirstName	String
lastName	LastName	String
tel	电话	String
groupId	机构ID	String
group	机构	参考Group-Model
dept	部门	参考Deptment-Model
Roles	角色	List 参考 role-model

Deptment-Model

字段代码	字段名称	字段类型	必须	说明
projectId	项目ID	U16		
deptId	部门ID	U16	是	
groupId	组织ID	U16		
pid	父ID	U32		
deptCode	部门编号	String		
deptName	部门名称	String	是	
deptFunction	部门职能	String		
deptType	部门类型	U8		
telephone	联系电话	String		

字段代码	字段名称	字段类型	必须	说明
buildDate	成立日期	Date		
dataStatus	数据状态	U8	是	
json	JSON字段	String		
version	版本	U16		
creator	创建人	String	是	
createdTime	创建时间	Time	是	
updater	更新人	String		
updatedAt	更新时间	Time		
remark	备注	String		

Group-Model

字段代码	字段名称	字段类型	必须	说明
groupId	组织ID	U16	是	
districtId	行政区划ID	U16		
pid	父ID	U32		
groupNo	机构编号	String		
groupName	机构名称	String	是	
fullName	机构全称	String		
shortName	机构简称	String		
fullPath	全路径	String		
orderNo	序号	U16		
address	联系地址	String		
linkMan	联系人	String		
telephone	联系电话	String		

字段代码	字段名称	字段类型	必须	说明
postCode	邮编	String		
dataStatus	数据状态	U8	是	0- 停用 1- 启用
json	JSON字段	String		
version	版本	U16		
creator	创建人	String	是	
createdTime	创建时间	Time	是	
updater	更新人	String		
updatedAt	更新时间	Time		
remark	备注	String		

Role-Model

字段代码	字段名称	字段类型	必须	说明
projectId	项目ID	U16		
roleId	角色ID	U16	是	
roleCode	角色代码	String		
roleName	角色名称	String	是	
dataStatus	数据状态	U8	是	
json	JSON字段	String		
version	版本	U16		
creator	创建人	String	是	
createdTime	创建时间	Time	是	
updater	更新人	String		
updatedAt	更新时间	Time		
remark	备注	String		

字段代码	字段名称	字段类型	必须	说明
grantUsers	授予用户	String		

User-Model

字段代码	字段名称	字段类型	必须	说明
userId	用户ID	U16	是	
groupId	组织ID	U16		
userNo	用户编号	String	是	
userName	用户名称	String	是	
cardNo	员工卡号	String		
password	密码	String		
desPassword	DES密码	String		
sex	性别	U8		
deptId	部门ID	U16		
telephone	联系电话	String		
address	联系地址	String		
email	邮箱	String		
birthday	出生年月	Date		
idCard	身份证号	String		
age	年龄	U8		
theme	界面主题	String		
authStartDate	权限开始日期	Date		
authEndDate	权限结束日期	Date		
dataStatus	数据状态	U8	是	
json	JSON字段	String		

4.2 获取用户信息

字段代码	字段名称	字段类型	必须	说明
version	版本	U16	是	
creator	创建人	String	是	
createdTime	创建时间	Time	是	
updater	更新人	String		
updatedTime	更新时间	Time		
remark	备注	String		
officeTel	办公室电话	String		
ownRoles	已有角色	String		

4.3 获取部门下的子部门

获取某一个部门下的子部门，不包含子孙部门

URL: http://权限地址/v1/sys/datac/datacUser/getDirectDeptmentsByDeptId

HTTP请求方式: post、get

参数信息:

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值
projectId	String	是	站点id
deptId	String	是	部门ID，允许为空值， 获取所有部门信息设定deptId为""

其中access_token在header中，其他参数在paramter中

参数说明:

如果'部门ID'为null或者空串直接写为空串""进行拼接即可。

```
curl -d 'access_token=${token}&projectId=${projectId}&deptId=${deptId}' http://权限认证地址/v1/sys/datac/datacUser/getDirectDeptmentsByDeptId
```

成功返回结果示例:

```
{
  "data": [
    {
      "deptId": 12072,
      "groupId": 1,
      "pid": 1,
      "deptCode": "12072",
      "deptName": "城建集团",
      "fullPath": "市级/市数字城管/城建集团",
      "orderNo": 22,
      "dataStatus": 1,
      "json": {},
      "creator": "zsadmin",
      "createTime": "2019-12-05 00:00:00"
    },
    {
      "deptId": 12073,
      "groupId": 1,
      "pid": 1,
      "deptCode": "12073",
      "deptName": "宜宾博世安防",

```

4.3 获取部门下的子部门

```
    "fullPath": "市级/市数字城管/宜宾博世安防",
    "orderNo": 23,
    "dataStatus": 1,
    "json": {},
    "creator": "zsadmin",
    "createdTime": "2019-12-05 00:00:00"
  },
  {
    "deptId": 12074,
    "groupId": 1,
    "pid": 1,
    "deptCode": "12074",
    "deptName": "临港经开区指挥中心",
    "fullPath": "市级/市数字城管/临港经开区指挥中心",
    "orderNo": 24,
    "dataStatus": 1,
    "json": {},
    "creator": "zsadmin",
    "createdTime": "2019-12-05 00:00:00"
  },
  {
    "deptId": 12107,
    "groupId": 1,
    "pid": 1,
    "deptCode": "12107",
    "deptName": "翠屏区指挥中心",
    "fullPath": "市级/市数字城管/翠屏区指挥中心",
    "orderNo": 25,
    "dataStatus": 1,
    "json": {},
    "creator": "zsadmin",
    "createdTime": "2019-12-05 00:00:00"
  },
  {
    "deptId": 12180,
    "groupId": 1,
    "pid": 1,
    "deptCode": "12180",
    "deptName": "珙县指挥中心",
    "fullPath": "市级/市数字城管/珙县指挥中心",
    "orderNo": 26,
    "dataStatus": 1,
    "json": {},
    "creator": "zsadmin",
    "createdTime": "2019-12-05 00:00:00"
  },
  {
    "deptId": 12217,
    "groupId": 1,
    "pid": 1,
    "deptCode": "12217",
    "deptName": "叙州区指挥中心",
    "fullPath": "市级/市数字城管/叙州区指挥中心",
    "orderNo": 27,
    "dataStatus": 1,
    "json": {},
    "creator": "zsadmin",
    "createdTime": "2019-12-05 00:00:00"
  },
  {
    "deptId": 12292,
    "groupId": 1,
```


4.3 获取部门下的子部门

```
        "pid": 1,
        "deptCode": "12292",
        "deptName": "南溪区指挥中心",
        "fullPath": "市级/市数字城管/南溪区指挥中心",
        "orderNo": 28,
        "dataStatus": 1,
        "json": {},
        "creator": "zsadmin",
        "createTime": "2019-12-05 00:00:00"
    },
    ],
    "success": true
}
```

失败返回结果示例

```
{
  "success": false,
  "code": 401,
  "message": "无效的应用Id!"
}
```

成功/失败 success为 true（成功）、false（失败）失败时有message字段返回

返回参数data内容说明： 字段值为空时，字段不返回

字段代码	字段名称	字段类型
deptId	部门ID	U16
pid	父部门ID	U16
deptCode	部门编号	String
deptName	部门名称	String
orderNo	排序	U16
dataStatus	数据状态 1有效 0无效	U8
creator	创建人	U16
createTime	创建时间	Time
json	JSON字段	String
groupId	组织id	U16

4.4 获取部门下的所有人员

获取某一个部门下的所有人员仅限于当前部门下人员不包含子部门的人员。

URL: http://权限地址/v1/sys/datac/datacUser/getDirectUsersByDeptId

HTTP请求方式: post、get

参数信息:

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值
projectId	String	是	站点id
deptId	String	是	部门ID 不可为空值

其中access_token在header中，其他参数在paramter中

```
curl -d 'access_token=${token}&projectId=${projectId}&deptId=${deptId}' http://权限认证/v1/sys/datac/datacUser/getDirectUsersByDeptId
```

成功返回结果示例:

```
{
  "data": [
    {
      "creator": "admin",
      "userNo": "abc",
      "deptId": 14017,
      "dataStatus": 1,
      "json": "{}",
      "createTime": "2020-05-27 12:43:11",
      "userName": "userA",
      "userId": 1001722
    },
    {
      "creator": "admin",
      "userNo": "def",
      "sex": 1,
      "deptId": 14017,
      "dataStatus": 1,
      "json": "{}",
      "createTime": "2020-05-27 11:33:06",
      "userName": "userB",
      "userId": 1001721
    }
  ],
  "success": true
}
```

失败返回结果示例

```
{
  "success": false,
  "code": 401,
  "message": "无效的应用Id!"
}
```

成功/失败 success为 true（成功）、false（失败）失败时有message字段返回

返回参数data内容说明： 字段值为空时，字段不返回

字段代码	字段名称	字段类型
userId	用户ID	U16
userGuid	来源标识ID	String
userNo	员工号/账号	String
userName	用户名	String
cardNo	卡号	String
sex	性别	String
deptId	部门ID	U16
telephone	电话	String
address	地址	String
email	邮箱	String
birthday	出生日期	Date
idCard	身份证号	String
age	年龄	Int
theme	主题	String
dataStatus	数据状态 1有效 0无效	U8
creator	创建人	U16
createdTime	创建时间	Date
updater	更新人	U16

4.4 获取部门下的所有人员

字段代码	字段名称	字段类型
updateTime	更新时间	Date
remark	备注	String
json	JSON字段	String
authStartDate	权限开始日期	Date
endAuthEndDate	权限结束日期	Date
officeTel	办公室电话	String
userGrade	用户等级	U8
type	用户类型	U8
iconFileId	头像id	String
iconFilePath	头像路径	String
enabled	活动用户标识： 0-非当前活动用户 1-当前活动用户	U8

4.5 获取角色下所有人员

获取角色编码、站点id获取该角色下所有人员。

URL: http://权限地址/v1/sys/user4sec/findUsersById

HTTP请求方式: get

参数信息:

参数名称	参数类型	必需	说明
roleCode	String	是	角色编码
projectId	String	是	站点id
access_token	String	是	4.1中获取的access_token

```
curl -G -d 'access_token=${token}&roleCode=${code}&projectId=${projectId}' http://权限地址/v1/sys/user4sec/findUsersById
```

成功返回结果示例:

```
{
  "code": "200",
  "message": "200",
  "data": [
    {
      "deptName": "测试机构",
      "deptId": 886530,
      "userName": "田乐宇",
      "userId": 31386
    },
    {
      "deptName": "总署机关",
      "deptId": 887612,
      "userName": "徐丹平",
      "userId": 8090
    }
  ]
}
```

失败返回结果示例

```
{
  "code": "500",
  "message": "您传输的参数有误",
  "data": null
}
```

成功/失败 success为 true（成功）、false（失败）失败时有message字段返回

4.5 获取角色下所有人员

返回参数data内容说明：字段值为空时，字段不返回

字段代码	字段名称	字段类型
deptName	部门名称	String
deptId	部门ID	int
userId	用户id	int
userName	用户名	String

4.6 获取用户的功能权限

登陆租户系统后获取当前访问人员具有权限的菜单和功能

URL: http://权限地址/v1/sys/user4sec/getProjectFunsByUserIdAndProjectId

HTTP请求方式: get

参数信息

参数代码	参数名称	必选	字段类型	说明
access_token	令牌	是	String	在4.1取到的access_token值
projectId	租户ID	否	U16	projectId可以为空
userId	用户ID	是	U16	

```
curl -d 'access_token=${token}&projectId=${projectId}&userId=${userId}' http://权限认证/v1/sys/user4sec/getProjectFunsByUserIdAndProjectId
```

成功返回结果如下:

```
{
  "data": [
    {
      "id": "60ee8395cf404d4da578febbd6fad5c0_2",
      "text": "测试菜单",
      "leaf": false,
      "expanded": true,
      "children": [
        {
          "id": "b958a2c9cbc44fc3aacbb52fdce8ef09_2",
          "text": "测试子菜单",
          "leaf": true,
          "expanded": true,
          "nodeType": 1,
          "systemType": 0
        }
      ],
      "nodeType": 1,
      "systemType": 0
    }
  ],
  "success": true
}
```

失败返回结果示例

```
{
  "success": false,
}
```

```
{
  "code": 401,
  "message": "无效的应用Id!"
}
```

返回参数data内容说明：字段值为空时，字段不返回

字段代码	字段名称	字段类型	说明
id	功能编号	U16	
text	功能名称	String	
url	链接	String	
expanded	是否展开	U8	
orderNo	序号	U16	
helpUrl	帮助地址	String	
nodeType	功能类型	S8	0-顶级菜单 1-节点 2-叶子 3-按钮
icon	图标	String	
systemType	系统类型	S8	
leaf	是否叶子节点	S8	True: 是 false 否
openNew	是否弹出菜单	S8	True: 是 false 否
children	子对象		当前对象列表

4.7 获取用户的表权限

根据用户ID或者租户ID、数据源ID获取用户具有权限的表。

URL: http://权限地址/v1/sys/user4sec/getDataPermissionsOfTableByUser

HTTP请求方式: get

验签规则: access_token

参数信息:

参数代码	参数名称	必选	字段类型	说明
access_token	令牌	是	String	在4.1取到的access_token值
datasource	数据源ID	是	String	
projectId	租户ID	否	U16	没有传空串
userId	用户ID	是	String	

参数说明:

如果'用户ID'、'数据源ID'或者'租户ID'为null或者空串直接写为空串""进行拼接即可。

```
curl -d 'access_token=${token}&datasource=${source}&projectId=${projectId}&userId=${userId}' http://权限地址/v1/sys/user4sec/getDataPermissionsOfTableByUser
```

成功返回样例如下:

```
{
  "data": [
    "9f723116332044a4bd049e50d93f7016",
    "7ce2f6424969482ab425f62ded2ca078",
    "8a80f4075bb0440b9ca4ddff30521388",
    "e27c6ae2a98f41dfa2c84cba9ee4eac4",
    "b87b67ab8d4742579b32c88268bc2df3",
    "83ed3aa946784ebf8ac1a2aac0d0af3d",
    "92a10c587d4348ba98a5ab9608c96ae2",
    "29620c81342f47c087ea7a37a2d645e8",
    "ee16c0321ea84ba2938cac33c14ce3ee",
    "8ac3b68918ee4fca80f08c5f6095e520",
    "e0c2c50b8caa4f998bb00071887c8a97",
    "31b31e49624b45c9b60ef87d5d956334",
    "0b3f7bd529b04fa68508ad0e1f8af891",
    "cf093b9ac03a44f38ed338ee8f587f82"
  ],
  "success": true,
  "lastTime": "2019-10-15 15:49:00"
}
```

失败返回结果示例

4.7 获取用户的表权限

```
{  
  "success": false,  
  "code": 401,  
  "message": "无效的应用Id!"  
}
```

成功/失败 success为 true（成功）、false（失败）失败时有message字段返回

4.8 获取用户的表字段权限

根据用户ID或者租户ID、数据源ID获取用户具有的某张表的字段权限。

URL: http://权限地址/v1/sys/user4sec/getDataPermissionsOfFieldByUserAndTable

HTTP请求方式: 'get'

验签规则: access_token

参数信息:

参数代码	参数名称	必选	字段类型	说明
access_token	令牌	是	String	在4.1取到的access_token值
datasource	数据源ID	是	String	
projectId	租户ID	否	U16	没有传空串
tableValue	表ID	是	String	
userId	用户ID	是	String	

参数说明:

如果'用户ID'、'数据源ID'或者'租户ID'为null或者空串直接写为空串""进行拼接即可。

```
curl -d 'access_token=${token}&datasource=${master}&projectId=${projectId}&userId=${userId}&tableValue=${value}' http://权限认证/v1/sys/user4sec/getDataPermissionsOfFieldByUserAndTable
```

成功返回样例如下:

```
{
  "data": [
    "VALID_DATE",
    "ID_NUMBER",
    "LAW_MAN",
    "LAW_MAN_TEL",
    "REG_CO",
    "RG_FUND",
    "ACCO_NO",
    "EN_FULL_CO",
    "BROKER_TYPE",
    "BUSI_TYPE",
    "FULL_NAME",
    "ADDR_CO",
    "CO_CLASS",
    "APPR_ID",
    "APPR_DEP",
    "ACCO_BANK",
    "TEL_CO",
    "RG_DATE",
  ]
}
```

4.8 获取用户的表字段权限

```
      "EN_ADDR_CO"  
    ],  
    "success": true,  
    "lastTime": "2019-10-15 15:55:18"  
  }  
}
```

失败返回结果示例

```
{  
  "success": false,  
  "code": 401,  
  "message": "无效的应用Id!"  
}
```

成功/失败 success为 true（成功）、false（失败）失败时有message字段返回

4.9 获取用户的表字段权限及附加条件

根据用户ID或者租户ID、数据源ID以及机构编码获取用户具有表的字段权限和字段授权条件。

URL: http://权限地址/v1/sys/user4sec/getDataPermissionsOfTableField

HTTP请求方式: get

验签规则: access_token

参数信息:

参数代码	参数名称	必选	字段类型	说明
access_token	令牌	是	String	在4.1取到的access_token值
groupNo	机构编号	是	String	
datasource	数据源ID	是	String	
projectId	租户ID	否	U16	没有传空串
tableValue	表ID	是	String	
userId	用户ID	是	String	

参数说明:

如果'用户ID'、'数据源ID'或者'租户ID'为null或者空串直接写为空串""进行拼接即可。

```
curl -d 'access_token=${token}&datasource=${master}&projectId=${projectId}&userId=${userId}&tableValue=${value}&groupNo=${groupNo}' http://认证地址/v1/sys/user4sec/getDataPermissionsOfTableField
```

成功返回样例如下:

```
{
  "data": {
    "showFields": [
      "VALID_DATE",
      "EN_FULL_CO",
      "BROKER_TYPE",
      "BUSI_TYPE",
      "FULL_NAME",
      "ADDR_CO",
      "EXAM_SCORE",
      "CUSTOMS_CODE",
      "CONTAC_CO",
      "CREDIT_MAR",
      "LICENSE_ID",
      "MAIL_CO",
      "CURR_CODE",
      "TRADE_CO",
    ]
  }
}
```

4.9 获取用户的表字段权限及附加条件

```
{
  "INV_FUND_T",
  "CO_CLASS",
  "APPR_ID",
  "APPR_DEP",
  "ACCO_BANK",
  "TEL_CO",
  "RG_DATE",
  "EN_ADDR_CO"
],
  "fieldWheres": [{
    "fieldName": "CUSTOMS_CODE",
    "fieldOp": "LIKE",
    "fieldCon": "29%",
    "dataGroupId": "ccuap_org"
  }]
},
"success": true,
"lastTime": "2019-10-15 16:00:23"
}
```

失败返回结果示例

```
{
  "success": false,
  "code": 401,
  "message": "无效的应用Id!"
}
```

成功/失败 success为 true（成功）、false（失败）失败时有message字段返回

4.10 根据站点id和姓名获取用户

根据站点id和姓名分页获取用户列表

URL: http://权限地址/v1/sys/user4sec/queryUser

HTTP请求方式: post、get

参数信息:

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值
projectId	String	是	站点id
userName	String	否	用户姓名
currPage	int	是	页码
pageLimit	int	是	条数

其中access_token在header中，其他参数在parameter中

参数说明:

成功返回结果示例:

```
{
  "data": [
    {
      "created_time": "2020-06-02 09:27:40",
      "deptName": "浦沿街道测试",
      "groupName": "全部",
      "userNo": "app_user",
      "userName": "应用用户",
      "userId": 1001726
    }
  ],
  "success": true
}
```

失败返回结果示例

```
{
  "success": false,
  "code": 401,
  "message": "无效的应用Id!"
}
```

成功/失败 success为 true（成功）、false（失败）失败时有message字段返回

返回参数data内容说明：字段值为空时，字段不返回

字段代码	字段名称	字段类型
deptName	部门名称	String
groupName	机构名称	String
userNo	用户编号	String
userName	用户名	String
created_time	创建时间	Time

4.11 获取指定机构的下级节点

获取某一个机构下的子机构，不包含子孙机构

URL: http://权限地址/v1/sys/datac/datacUser/findNextChildGroupNodeTree

HTTP请求方式: post、get

参数信息:

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值
groupId	String	是	机构id，允许为空值，获取一级机构信息 groupId传参""

其中access_token在header中，其他参数在paramter中

参数说明:

如果'机构ID'为null或者空串直接写为空串""进行拼接即可。

```
curl -d 'access_token=${token}&groupId=${groupId}' http://权限认证地址/v1/sys/datac/datacUser/findNextChildGroupNodeTree
```

成功返回结果示例:

```
{
  "data": [
    {
      "groupId": 2,
      "pid": 1,
      "groupNo": "111",
      "groupName": "111",
      "fullPath": "全部/111",
      "dataStatus": 1,
      "json": {},
      "version": 0,
      "creator": "tc_admin",
      "createdTime": "2020-06-16 14:16:12"
    }
  ],
  "success": true,
  "lastTime": "2020-06-16 14:52:12"
}
```

失败返回结果示例

4.11 获取指定机构的下级节点

```
{
  "success": false,
  "code": 401,
  "message": "无效的应用Id!"
}
```

成功/失败 success为 true（成功）、false（失败）失败时有message字段返回

返回参数data内容说明： 字段值为空时，字段不返回

字段代码	字段名称	字段类型
groupId	组织ID	U16
districtId	行政区划ID	U16
pid	父ID	U32
groupNo	机构编号	String
groupName	机构名称	String
fullName	机构全称	String
shortName	机构简称	String
fullPath	全路径	String
orderNo	序号	U16
address	联系地址	String
linkMan	联系人	String
telephone	联系电话	String
postCode	邮编	String
dataStatus	数据状态	U8
json	JSON字段	String
version	版本	U16
creator	创建人	String
createdTime	创建时间	Time
updater	更新人	String

4.11 获取指定机构的下级节点

字段代码	字段名称	字段类型
updateTime	更新时间	Time
remark	备注	String

4.12 根据机构编号获取机构信息

根据机构编号获取机构详情

URL: http://权限地址/v1/sys/datac/datacUser/findGroupByNo

HTTP请求方式: post、get

验签规则: MD5

参数信息:

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值
groupNo	String	是	机构编号

其中access_token在header中，其他参数在paramter中

参数说明:

如果'机构ID'为null或者空串直接写为空串""进行拼接即可。

```
curl -d 'access_token=${token}&groupNo=${groupNo}' http://权限认证地址/v1/sys/datac/datacUser/findGroupByNo
```

****成功返回结果示例: ****

```
`` json
{
  "data": {
    "groupId": 1,
    "orgGuid": "94480300-fca2-4268-86c2-4c13963bf4cb",
    "groupNo": "0000",
    "groupName": "全部",
    "fullPath": "全部",
    "dataStatus": 1,
    "json": {},
    "version": 1,
    "creator": "初始化",
    "createdTime": "2020-05-28 13:52:07"
  },
  "success": true,
  "lastTime": "2020-06-16 14:59:37"
}
```

失败返回结果示例

```
{
  "success": false,
  "code": 401,
}
```

```
    "message": "无效的应用Id!"
}
```

成功/失败 success为 true（成功）、false（失败）失败时有message字段返回

返回参数data内容说明： 字段值为空时，字段不返回

字段代码	字段名称	字段类型
groupId	组织ID	U16
districtId	行政区划ID	U16
pid	父ID	U32
groupNo	机构编号	String
groupName	机构名称	String
fullName	机构全称	String
shortName	机构简称	String
fullPath	全路径	String
orderNo	序号	U16
address	联系地址	String
linkMan	联系人	String
telephone	联系电话	String
postCode	邮编	String
dataStatus	数据状态	U8
json	JSON字段	String
version	版本	U16
creator	创建人	String
createdTime	创建时间	Time
updater	更新人	String
updatedTime	更新时间	Time

4.12 根据机构编号获取机构信息

字段代码	字段名称	字段类型
remark	备注	String

4.13 分页查询机构信息

分页查询机构详情

URL: http://权限地址/v1/sys/datac/datacUser/queryGroup

HTTP请求方式: get

验签规则: MD5

参数信息:

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值
start	int	否	开始的记录数
limit	int	否	最后的记录数

其中access_token在header中，其他参数在paramter中

参数说明:

如果'机构ID'为null或者空串直接写为空串""进行拼接即可。

```
curl -d 'access_token=${token}&start=${start}&limit=${limit}' http://权限认证地址/v1/sys/datac/datacUser/queryGroup
```

****成功返回结果示例:****

```
``` json
{
 "totalCount": 2,
 "data": [
 {
 "groupId": 1,
 "orgGuid": "94480300-fca2-4268-86c2-4c13963bf4cb",
 "groupNo": "0000",
 "groupName": "全部",
 "fullPath": "全部",
 "dataStatus": 1,
 "json": {},
 "version": 1,
 "creator": "初始化",
 "createTime": "2020-05-28 13:52:07"
 }
],
 "success": true,
 "lastTime": "2020-07-09 15:11:20"
}
```

**失败返回结果示例**

```
{
 "success": false,
 "code": 401,
 "message": "无效的应用Id!"
}
```

**成功/失败** success为 true（成功）、false（失败）失败时有message字段返回

**返回参数data内容说明：** 字段值为空时，字段不返回

字段代码	字段名称	字段类型
groupId	组织ID	U16
districtId	行政区划ID	U16
pid	父ID	U32
groupNo	机构编号	String
groupName	机构名称	String
fullName	机构全称	String
shortName	机构简称	String
fullPath	全路径	String
orderNo	序号	U16
address	联系地址	String
linkMan	联系人	String
telephone	联系电话	String
postCode	邮编	String
dataStatus	数据状态	U8
json	JSON字段	String
version	版本	U16
creator	创建人	String
createdTime	创建时间	Time
updater	更新人	String



#### 4.13 分页查询机构信息

字段代码	字段名称	字段类型
updateTime	更新时间	Time
remark	备注	String

## 4.14 根据编号获取用户信息

根据编号获取用户信息

**URL:** http://权限地址/v1/sys/user4sec/findByUserNo

**HTTP请求方式:** `get`

**验签规则:** access\_token

**参数信息**

参数代码	参数名称	必选	字段类型	说明
access_token	令牌	是	String	在4.1取到的access_token值
userNo	用户编号	是	String	

```
curl -d 'access_token=${token}&userNo=${userNo}' http://权限地址/v1/sys/user4sec/findByUserNo
```

**成功返回结果如下:**

```
{
 "data": {
 "userId": 1001723,
 "groupId": 1,
 "userNo": "tc_admin",
 "userName": "管理员",
 "deptId": 14017,
 "dataStatus": 1,
 "json": {},
 "version": 10,
 "creator": "zsr",
 "createdTime": "2020-05-28 15:38:37",
 "ownRoles": "JQB_TEST,112,平台管理员,站点管理员"
 },
 "success": true,
 "lastTime": "2019-12-09 10:24:45"
}
```

**失败返回结果示例**

```
{
 "success": false,
 "code": 401,
 "message": "无效的应用Id!"
}
```

**成功/失败** success为 true（成功）、false（失败）失败时有message字段返回

**返回结果字段说明：**

参数名称	参数类型	说明
userId	Long	用户ID
userName	String	用户姓名
userNo	String	用户编号（登录名）
cardNo	String	用户卡号
sex	Short	性别（0:未说明，1:男，2:女）
deptId	Long	部门ID
projectId	Long	站点ID
status	Boolean	true: 启用，false: 禁用
version	Integer	版本号
telephone	String	手机号
address	String	联系人地址
email	String	联系人邮箱
birthday	Date	出生年月
userGrade	Short	用户等级
postId	Integer	职务标识
type	Short	用户类型
remark	String	备注
officeTel	String	办公室电话
iconFilePath	String	头像路径
createTime	Date	创建时间

## 4.15 获取多个用户编号对应的用户信息

获取多个用户编号对应的用户信息

**URL:** http://权限地址/v1/sys/user4sec/findByUserNos

**HTTP请求方式:** get

**验签规则:** access\_token

### 参数信息

参数代码	参数名称	必选	字段类型	说明
access_token	令牌	是	String	在4.1取到的access_token值
userNos	用户编号	是	String	多个用户编号用,分隔

```
curl -d 'access_token=${token}&userNos=${userNos}' http://权限地址/v1/sys/user4sec/findByUserNos
```

**成功返回结果如下:**

```
{
 "data": [
 {
 "userId": 1001747,
 "groupId": 1,
 "projectId": 2,
 "userNo": "12121212",
 "userName": "21212121212",
 "deptId": 14017,
 "dataStatus": 1,
 "json": {},
 "version": 0,
 "creator": "tc_admin",
 "createTime": "2020-07-27 15:36:41",
 "userGrade": 4
 },
 {
 "userId": 1001746,
 "groupId": 1,
 "projectId": 2,
 "userNo": "212121",
 "userName": "2121212",
 "deptId": 14017,
 "dataStatus": 1,
 "json": {},
 "version": 0,
 "creator": "tc_admin",
 "createTime": "2020-07-10 18:00:20"
 }
],
}
```

```
{
 "success": true,
 "lastTime": "2020-07-27 15:39:27"
}
```

**失败返回结果示例**

```
{
 "success": false,
 "code": 401,
 "message": "无效的应用Id!"
}
```

**成功/失败** success为 true（成功）、false（失败）失败时有message字段返回

**返回结果字段说明：**

参数名称	参数类型	说明
userId	Long	用户ID
userName	String	用户姓名
userNo	String	用户编号（登录名）
cardNo	String	用户卡号
sex	Short	性别（0:未说明，1:男，2:女）
deptId	Long	部门ID
projectId	Long	站点ID
status	Boolean	true：启用，false：禁用
version	Integer	版本号
telephone	String	手机号
address	String	联系人地址
email	String	联系人邮箱
birthday	Date	出生年月
userGrade	Short	用户等级
postId	Integer	职务标识
type	Short	用户类型

#### 4.15 获取多个用户编号对应的用户信息

参数名称	参数类型	说明
remark	String	备注
officeTel	String	办公室电话
iconFilePath	String	头像路径
createTime	Date	创建时间

## 4.16 根据站点名称获取站点信息

根据站点名称分页获取站点信息

**URL:** http://权限地址/v1/sys/user4sec/queryWebsite

**HTTP请求方式:** post、get

**参数信息:**

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值
websiteName	String	否	站点名称
currPage	int	是	页码
pageLimit	int	是	条数

其中access\_token在header中，其他参数在parameter中

**参数说明:**

**成功返回结果示例:**

```
{
 "totalCount": 10,
 "data": [
 {
 "projectName": "阿里集成测试站点",
 "projectId": 24
 }
],
 "success": true,
 "lastTime": "2020-07-29 17:39:46"
}
```

**失败返回结果示例**

```
{
 "success": false,
 "code": 401,
 "message": "无效的应用Id!"
}
```

**成功/失败** success为 true（成功）、false（失败）失败时有message字段返回

**返回参数data内容说明:** 字段值为空时，字段不返回

#### 4.16 根据站点名称获取站点信息

字段代码	字段名称	字段类型
projectId	Long	站点ID
projectName	站点名称	String



## 4.17 获取站点用户列表

根据站点id获取用户列表

**URL:** http://权限地址/v1/sys/user4sec/getProjectUserList

**HTTP请求方式:** get

**参数信息:**

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值
projectId	String	是	站点id

其中access\_token在header中，其他参数在parameter中

**参数说明:**

**成功返回结果示例:**

```
{
 "data": [
 {
 "officeTel": "12121",
 "deptName": "用于测试",
 "address": "12121",
 "userNo": "hwcs",
 "deptId": 14017,
 "telephone": "15012345678",
 "userName": "环卫平台测试",
 "cardNo": "212121",
 "userId": 1000009,
 "email": "2121212"
 }
],
 "success": true
}
```

**失败返回结果示例**

```
{
 "success": false,
 "code": 401,
 "message": "无效的应用Id!"
}
```

**成功/失败** success为 true（成功）、false（失败）失败时有message字段返回

**返回参数data内容说明:** 字段值为空时，字段不返回

#### 4.17 获取站点用户列表

字段代码	字段名称	字段类型
officeTel	办公室电话	String
deptName	部门名称	String
address	地址	String
userNo	用户编号	String
deptId	部门id	String
telephone	电话	String
userName	用户名称	String
cardNo	员工卡号	String
userId	用户id	Long
email	邮箱	String

## 4.18 新增部门

新增部门

**URL:** http://权限地址/v1/sys/user4sec/addDept

**HTTP请求方式:** post

**参数信息:**

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值
projectId	String	是	站点id
deptId	int	否	部门ID
groupId	int	否	组织ID
orgGuid	varchar	否	机构标识
pid	int	否	父ID
deptCode	varchar	是	部门编号
deptName	varchar	是	部门名称
fullName	varchar	否	机构全称
shortName	varchar	否	机构简称
fullPath	varchar	否	全路径
orderNo	int	否	排序号
deptFunction	varchar	否	部门职能
deptType	smallint	否	部门类型(DEPT_TYPE)
telephone	varchar	否	联系电话
buildDate	date	否	成立日期
dataStatus	smallint	是	数据状态（ENABLED）
json	varchar	否	JSON字段

#### 4.18 新增部门

参数名称	参数类型	必需	说明
version	int	否	版本
creator	varchar	否	创建人
createdTime	datetime	否	创建时间
updater	varchar	否	更新人
updatedAt	datetime	否	更新时间
remark	varchar	否	备注
fullIdPath	varchar	否	ID全路径
areald	varchar	否	地区ID

其中access\_token在header中，其他参数在parameter中

##### 成功返回结果示例：

```
{
 "message": "添加部门成功!",
 "success": true
}
```

##### 失败返回结果示例

```
{
 "success": false,
 "code": 401,
 "message": "无效的应用Id!"
}
```

**成功/失败** success为 true（成功）、false（失败）失败时有message字段返回

## 4.19 修改部门

修改部门

**URL:** http://权限地址/v1/sys/user4sec/updateDept

**HTTP请求方式:** post

**参数信息:**

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值
projectId	String	是	站点id
deptId	int	是	部门ID
groupId	int	是	组织ID
orgGuid	varchar	否	机构标识
pid	int	否	父ID
deptCode	varchar	是	部门编号
deptName	varchar	是	部门名称
fullName	varchar	是	机构全称
shortName	varchar	是	机构简称
fullPath	varchar	是	全路径
orderNo	int	否	排序号
deptFunction	varchar	否	部门职能
deptType	smallint	否	部门类型(DEPT_TYPE)
telephone	varchar	否	联系电话
buildDate	date	否	成立日期
dataStatus	smallint	是	数据状态（ENABLED）
json	varchar	否	JSON字段

#### 4.19 修改部门

参数名称	参数类型	必需	说明
version	int	是	版本
creator	varchar	是	创建人
createdTime	datetime	是	创建时间
updater	varchar	否	更新人
updatedTime	datetime	否	更新时间
remark	varchar	否	备注
fullIdPath	varchar	是	ID全路径
areald	varchar	否	地区ID

其中access\_token在header中，其他参数在paramter中

**成功返回结果示例：**

```
{
 "message": "修改部门成功!",
 "success": true
}
```

**失败返回结果示例**

```
{
 "success": false,
 "code": 401,
 "message": "无效的应用Id!"
}
```

**成功/失败** success为 true（成功）、false（失败）失败时有message字段返回

## 4.20 删除部门

### 删除部门

**URL:** http://权限地址/v1/sys/user4sec/deleteDept

**HTTP请求方式:** get

**参数信息:**

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值
projectId	String	是	站点id
deptId	int	是	部门ID

其中access\_token在header中，其他参数在paramter中

**成功返回结果示例:**

```
{
 "message": "删除部门节点成功!",
 "success": true
}
```

**失败返回结果示例**

```
{
 "success": false,
 "code": 401,
 "message": "无效的应用Id!"
}
```

**成功/失败** success为 true（成功）、false（失败）失败时有message字段返回

## 4.21 新增用户

新增用户

**URL:** http://权限地址/v1/sys/user4sec/addUser

**HTTP请求方式:** post

**参数信息:**

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值
projectId	String	是	站点id
groupId	bigint	是	组织ID
userId	int	否	用户ID
deptId	int	是	部门ID
userGuid	varchar	否	用户标识
enabled	smallint	否	活动用户标识：用户标识对应的活动用户标识0-非当前活动用户1-当前活动用户
userNo	varchar	是	用户编号
userName	varchar	是	用户名称
cardNo	varchar	否	员工卡号
password	varchar	否	密码
desPassword	varchar	否	DES密码（可逆）
sex	smallint	否	性别（SEX）
telephone	varchar	否	联系电话
officeTel	varchar	否	办公室电话
address	varchar	否	联系地址
email	varchar	否	邮箱



参数名称	参数类型	必需	说明
birthday	date	否	出生年月
idCard	varchar	否	身份证号
age	smallint	否	年龄
theme	varchar	否	界面主题
authStartDate	date	否	权限开始日期
authEndDate	date	否	权限结束日期
dataStatus	smallint	否	数据状态（ENABLED）
json	varchar	否	JSON字段
version	int	否	版本
creator	varchar	否	创建人
createdTime	datetime	否	创建时间
updater	varchar	否	更新人
updatedTime	datetime	否	更新时间
remark	varchar	否	备注
ownRoles	varchar	否	拥有角色
userGrade	smallint	是	员工等级（USER_GRADE）
type	smallint	否	用户类型（USER_TYPE）
iconFileId	varchar	否	头像id
iconFilePath	varchar	否	头像路径

其中access\_token在header中，其他参数在paramter中

**成功返回结果示例：**

```
{
 "message": "添加用户成功!",
 "success": true
}
```

#### 失败返回结果示例

```
{
 "success": false,
 "code": 401,
 "message": "无效的应用Id!"
}
```

**成功/失败** success为 true（成功）、false（失败）失败时有message字段返回

## 4.22 修改用户

修改用户

**URL:** http://权限地址/v1/sys/user4sec/updateUser

**HTTP请求方式:** post

**参数信息:**

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值
projectId	String	是	站点id
groupId	bigint	是	组织ID
userId	int	是	用户ID
deptId	int	是	部门ID
userGuid	varchar	否	用户标识
enabled	smallint	否	活动用户标识：用户标识对应的活动用户标识0-非当前活动用户1-当前活动用户
userNo	varchar	是	用户编号
userName	varchar	是	用户名称
cardNo	varchar	否	员工卡号
password	varchar	否	密码
desPassword	varchar	否	DES密码（可逆）
sex	smallint	否	性别（SEX）
telephone	varchar	否	联系电话
officeTel	varchar	否	办公室电话
address	varchar	否	联系地址
email	varchar	否	邮箱

参数名称	参数类型	必需	说明
birthday	date	否	出生年月
idCard	varchar	否	身份证号
age	smallint	否	年龄
theme	varchar	否	界面主题
authStartDate	date	否	权限开始日期
authEndDate	date	否	权限结束日期
dataStatus	smallint	是	数据状态（ENABLED）
json	varchar	否	JSON字段
version	int	是	版本
creator	varchar	是	创建人
createdTime	datetime	是	创建时间
updater	varchar	否	更新人
updatedTime	datetime	否	更新时间
remark	varchar	否	备注
ownRoles	varchar	否	拥有角色
userGrade	smallint	是	员工等级（USER_GRADE）
type	smallint	否	用户类型（USER_TYPE）
iconFileId	varchar	否	头像id
iconFilePath	varchar	否	头像路径

其中access\_token在header中，其他参数在paramter中

**成功返回结果示例：**

```
{
 "message": "修改用户成功!",
 "success": true
}
```

### 失败返回结果示例

```
{
 "success": false,
 "code": 401,
 "message": "无效的应用Id!"
}
```

**成功/失败** success为 true（成功）、false（失败）失败时有message字段返回

## 4.23 删除用户

### 删除用户

**URL:** http://权限地址/v1/sys/user4sec/deleteUser

**HTTP请求方式:** get

**参数信息:**

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值
projectId	String	是	站点id
deptId	int	是	部门ID
userId	int	是	用户ID

其中access\_token在header中，其他参数在parameter中

**成功返回结果示例:**

```
{
 "message": "删除用户成功!",
 "success": true
}
```

**失败返回结果示例**

```
{
 "success": false,
 "code": 401,
 "message": "无效的应用Id!"
}
```

**成功/失败** success为 true（成功）、false（失败）失败时有message字段返回

## 5.部门增量推送信息

**topic:** datac\_user\_dept\_topic

## 5.1 新增/导入部门推送信息

### 新建/导入部门推送信息

\*推送结果\*

```
{
 "data": [
 {
 "createTime": 1583174064000,
 "dataStatus": 1,
 "deptCode": "csbm",
 "deptId": 14021,
 "deptName": "测试部门",
 "orderNo": 1,
 "projectId": 2,
 "version": 0
 }
],
 "lastTime": "2020-06-11 15:40:59",
 "objectType": "dept",
 "operation": "created",
 "projectId": 2,
 "success": true
}
```

说明：  
监听的mq的Topic为“datac\_user\_dept\_topic”,推送信息中 **data** 为变更数据，各站点根据 **projectId** 过滤各自数据进行同步处理。

结果说明：

参数名称	参数类型	说明
code	String	返回结果状态码
data	Object	变更数据
message	String	返回提示语
lastTime	String	同步时间
objectType	String	对象类型（dept: 部门，user: 用户）
operation	String	操作类型（created: 新建/导入，updated:修改，deleted:删除）
projectId	Long	站点id

部门信息：



参数名称	参数类型	说明
deptId	Long	部门ID
deptName	String	部门名称
pid	Long	父节点ID
projectId	Long	站点ID
deptCode	String	部门编码
orderNo	String	部门排序号
deptType	Short	部门类型
telephone	String	联系电话
dataStatus	Short	（1:启用，0:禁用）
version	Integer	版本号
createTime	Date	创建时间
remark	String	描述
areaId	String	区划ID

## 5.2 修改部门推送信息

### 新建/修改部门推送信息

\*推送结果\*

```
{
 "data": {
 "createTime": 1591861660000,
 "dataStatus": 1,
 "deptCode": "dept_720665563809972224",
 "deptId": 888221,
 "deptName": "测试部门_change",
 "projectId": 2,
 "remark": "",
 "telephone": "",
 "version": 1
 },
 "lastTime": "2020-06-11 15:47:53",
 "objectType": "dept",
 "operation": "updated",
 "projectId": 2,
 "success": true
}
```

说明：  
监听的mq的Topic为“datac\_user\_dept\_topic”,推送信息中 **data** 为变更数据，各站点根据 **projectId** 过滤各自数据进行同步处理。

结果说明：

参数名称	参数类型	说明
code	String	返回结果状态码
data	Object	变更数据
message	String	返回提示语
lastTime	String	同步时间
objectType	String	对象类型（dept: 部门，user: 用户）
operation	String	操作类型（created: 新建/导入，updated:修改，deleted: 删除）
projectId	Long	站点id

部门信息：

参数名称	参数类型	说明
deptId	Long	部门ID
deptName	String	部门名称
pid	Long	父节点ID
projectId	Long	站点ID
deptCode	String	部门编码
orderNo	String	部门排序号
deptType	Short	部门类型
telephone	String	联系电话
dataStatus	Short	（1:启用，0:禁用）
version	Integer	版本号
createTime	Date	创建时间
remark	String	描述
areaId	String	区划ID

## 5.3 删除部门推送信息

### 部门删除

#### 推送结果

```
{
 "data": {
 "deptId": [
 888221
]
 },
 "lastTime": "2020-06-11 15:48:44",
 "objectType": "dept",
 "operation": "deleted",
 "projectId": 2,
 "success": true
}
```

说明：  
监听的mq的Topic为“datac\_user\_dept\_topic”，推送信息中 **data** 为变更数据，各站点根据 **projectId** 过滤各自数据进行同步处理。

#### 结果说明：

参数名称	参数类型	说明
code	String	返回结果状态码
data	Object	变更数据
message	String	返回提示语
lastTime	String	同步时间
objectType	String	对象类型（dept：部门，user：用户）
operation	String	操作类型（created：新建/导入，updated:修改，deleted:删除）
projectId	Long	站点id

#### 信息实体说明：

参数名称	参数类型	说明
deptId	List	部门ID集合

### 5.3 删除部门推送信息

说明：

部门删除需要根据站点ID（projectId）和部门ID（deptID）作为唯一主键进行删除。

## 6.用户增量推送信息

**topic:** datac\_user\_dept\_topic

## 6.1 新增/导入用户推送信息

### 用户新增/导入推送信息

\*推送结果\*

```
{
 "data": [
 {
 "createTime": 1584638284000,
 "deptId": 14021,
 "projectId": 2,
 "status": true,
 "telephone": "13211111111",
 "userId": 1000037,
 "userName": "郑锐",
 "userNo": "zhengrui",
 "version": 0
 }
],
 "lastTime": "2020-06-11 15:51:26",
 "objectType": "user",
 "operation": "created",
 "projectId": 2,
 "success": true
}
```

说明：  
监听的mq的Topic为“datac\_user\_dept\_topic”,推送信息中 **data** 为变更数据，各站点根据 **projectId** 过滤各自数据进行同步处理。

结果说明：

参数名称	参数类型	说明
code	String	返回结果状态码
data	Object	变更数据
message	String	返回提示语
lastTime	String	同步时间
objectType	String	对象类型（dept: 部门，user: 用户）
operation	String	操作类型（created: 新建/导入，updated:修改，deleted:删除）
projectId	Long	站点id

用户信息说明：

参数名称	参数类型	说明
userId	Long	用户ID
userName	String	用户姓名
userNo	String	用户编号（登录名）
cardNo	String	用户卡号
sex	Short	性别（0:未说明，1:男，2:女）
deptId	Long	部门ID
projectId	Long	站点ID
status	Boolean	true: 启用，false: 禁用
version	Integer	版本号
telephone	String	手机号
address	String	联系人地址
email	String	联系人邮箱
birthday	Date	出生年月
userGrade	Short	用户等级
postId	Integer	职务标识
type	Short	用户类型
remark	String	备注
officeTel	String	办公室电话
iconFilePath	String	头像路径
createTime	Date	创建时间



## 6.2 修改用户推送信息

### 用户删除

#### 推送结果

```
{
 "data": {
 "createTime": 1591861984000,
 "deptId": 14021,
 "projectId": 2,
 "status": true,
 "userId": 1001738,
 "userName": "张三",
 "userNo": "zhangsan",
 "version": 1
 },
 "lastTime": "2020-06-11 15:53:31",
 "objectType": "user",
 "operation": "updated",
 "projectId": 2,
 "success": true
}
```

说明：  
监听的mq的Topic为“datac\_user\_dept\_topic”,推送信息中 **data** 为变更数据，各站点根据 **projectId** 过滤各自数据进行同步处理。

#### 结果说明：

参数名称	参数类型	说明
code	String	返回结果状态码
data	Object	变更数据
message	String	返回提示语
lastTime	String	同步时间
objectType	String	对象类型（dept: 部门，user: 用户）
operation	String	操作类型（created: 新建/导入，updated:修改，deleted: 删除）
projectId	Long	站点id

#### 用户信息说明：

参数名称	参数类型	说明
userId	Long	用户ID
userName	String	用户姓名
userNo	String	用户编号（登录名）
cardNo	String	用户卡号
sex	Short	性别（0:未说明，1:男，2:女）
deptId	Long	部门ID
projectId	Long	站点ID
status	Boolean	true: 启用，false: 禁用
version	Integer	版本号
telephone	String	手机号
address	String	联系人地址
email	String	联系人邮箱
birthday	Date	出生年月
userGrade	Short	用户等级
postId	Integer	职务标识
type	Short	用户类型
remark	String	备注
officeTel	String	办公室电话
iconFilePath	String	头像路径
createTime	Date	创建时间

## 6.3 删除用户推送信息

### 用户删除

#### 推送结果

```
{
 "data": {
 "userDepts": [
 {
 "deptId": 14021,
 "userId": 1000037
 }
]
 },
 "lastTime": "2020-06-11 16:02:38",
 "objectType": "user",
 "operation": "deleted",
 "projectId": 2,
 "success": true
}
```

说明：  
监听的mq的Topic为“datac\_user\_dept\_topic”,推送信息中 **data** 为变更数据，各站点根据 **projectId** 过滤各自数据进行同步处理。

#### 结果说明：

参数名称	参数类型	说明
code	String	返回结果状态码
data	Object	变更数据
message	String	返回提示语
lastTime	String	同步时间
objectType	String	对象类型（dept: 部门，user: 用户）
operation	String	操作类型（created: 新建/导入，updated:修改，deleted:删除）
projectId	Long	站点id

#### 用户部门关联实体说明：

参数名称	参数类型	说明
------	------	----

### 6.3 删除用户推送信息

参数名称	参数类型	说明
userId	Long	用户ID
deptId	Long	部门ID

说明：

用户删除需要部门ID（deptId）和用户ID（userId）作为唯一主键进行删除。

## 7.行政区划增量推送信息

## 7.1 新增行政区划推送信息

### 行政区划新增推送信息

\*推送结果\*

```
{
 "data": {
 "districtId": 20,
 "districtName": "湖北省",
 "districtNo": "hubeisheng",
 "orderNo": 2,
 "pid": 19,
 "poltype": 1
 },
 "lastTime": "2020-07-13 16:36:58",
 "objectType": "district",
 "operation": "created",
 "success": true
}
```

说明：  
监听的mq的Topic为“datac\_district\_topic”,推送信息中 **data** 为变更数据，行政区划不区分站点

结果说明：

参数名称	参数类型	说明
code	String	返回结果状态码
data	Object	变更数据
message	String	返回提示语
lastTime	String	同步时间
objectType	String	对象类型（dept: 部门，user: 用户）
operation	String	操作类型（created: 新建/导入，updated:修改，deleted: 删除）
projectId	Long	站点id

用户信息说明：

参数名称	参数类型	说明
districtId	Long	行政区划ID

参数名称	参数类型	说明
districtName	String	行政区划名称
districtNo	String	行政区划编码
orderNo	String	排序号
poltype	Short	行政级别,0-全国、1-省级、2-地市、3-县市、4-乡镇、5-社区(村)
pid	Long	父ID

## 7.2 修改行政区划推送信息

### 行政区划修改推送信息

\*推送结果\*

```
{
 "data": {
 "districtId": 19,
 "districtName": "中国",
 "districtNo": "zhongug",
 "orderNo": 1,
 "poltype": 0,
 "remark": "排第一位"
 },
 "lastTime": "2020-07-13 16:44:58",
 "objectType": "district",
 "operation": "updated",
 "success": true
}
```

说明：  
监听的mq的Topic为“datac\_district\_topic”,推送信息中 **data** 为变更数据，行政区划不区分站点

结果说明：

参数名称	参数类型	说明
code	String	返回结果状态码
data	Object	变更数据
message	String	返回提示语
lastTime	String	同步时间
objectType	String	对象类型（dept: 部门，user: 用户）
operation	String	操作类型（created: 新建/导入，updated:修改，deleted: 删除）
projectId	Long	站点id

用户信息说明：

参数名称	参数类型	说明
districtId	Long	行政区划ID



参数名称	参数类型	说明
districtName	String	行政区划名称
districtNo	String	行政区划编码
orderNo	String	排序号
poltype	Short	行政级别,0-全国、1-省级、2-地市、3-县市、4-乡镇、5-社区(村)
pid	Long	父ID

## 7.3 删除行政区划推送信息

### 行政区划删除推送信息

\*推送结果\*

```
{
 "data": {
 "districtId": 20
 },
 "lastTime": "2020-07-13 16:46:16",
 "objectType": "district",
 "operation": "deleted",
 "success": true
}
```

说明：  
监听的mq的Topic为“datac\_district\_topic”，推送信息中 **data** 为变更数据，行政区划不区分站点

结果说明：

参数名称	参数类型	说明
code	String	返回结果状态码
data	Object	变更数据
message	String	返回提示语
lastTime	String	同步时间
objectType	String	对象类型（dept：部门，user：用户）
operation	String	操作类型（created：新建/导入，updated:修改，deleted:删除）
projectId	Long	站点id

用户信息说明：

参数名称	参数类型	说明
districtId	Long	删除的行政区划ID

# 8.权限系统操作

## 最终想要的效果：

下游应用获取认证中心的token后（见接口：[4.1 获取应用的access\\_token](#)），  
调用认证中心接口4.6（见接口：[4.6 获取用户的功能权限集合](#)）

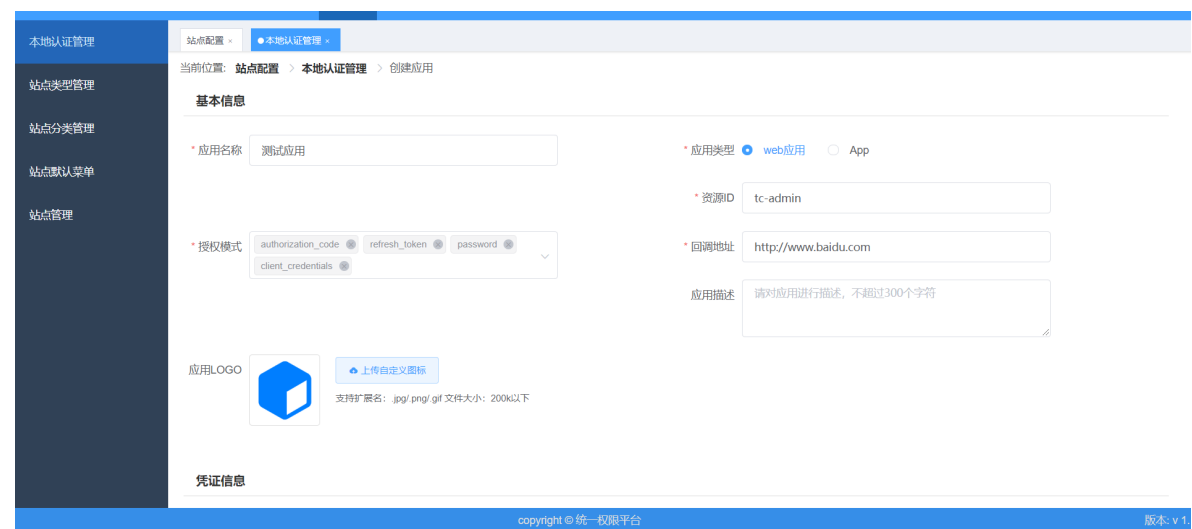
## 背景知识：

要实现这个业务闭环需要平台管理员和站点管理员相互配合操作：

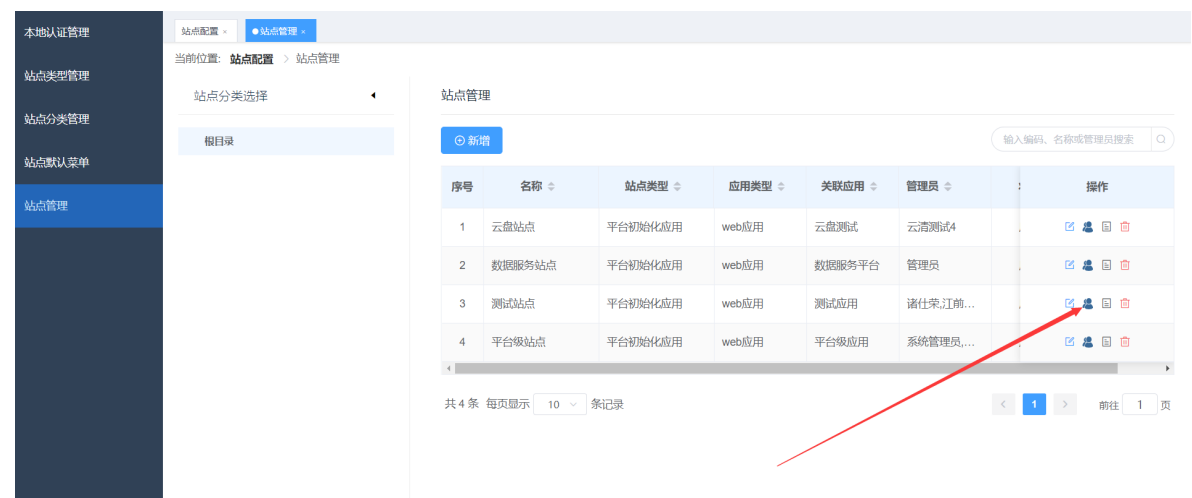
- 1、平台管理员创建应用、站点，并分配站点管理员；
- 2、站点管理员配置站点菜单、部门、人员和角色；

## 配置步骤：

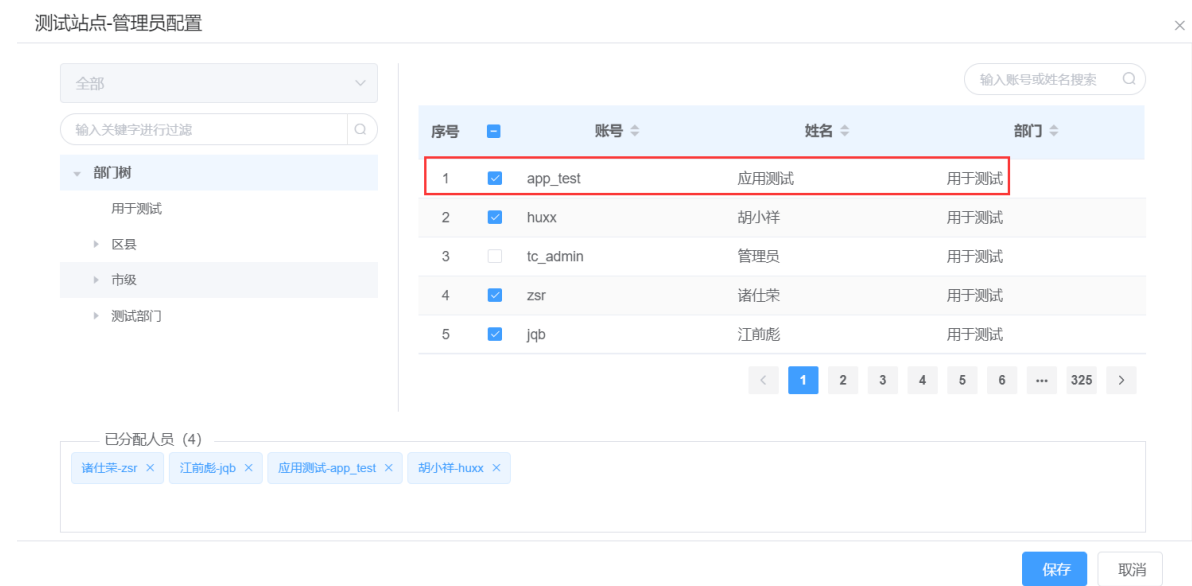
- 1、平台管理员创建应用



- 2、平台管理员创建站点，并分配站点管理员



8.权限系统操作



3、使用站点管理员登陆，点击齿轮进入权限配置



4、站点管理员配置站点菜单



8.权限系统操作

5、站点管理员配置站点部门



6、站点管理员配置站点人员



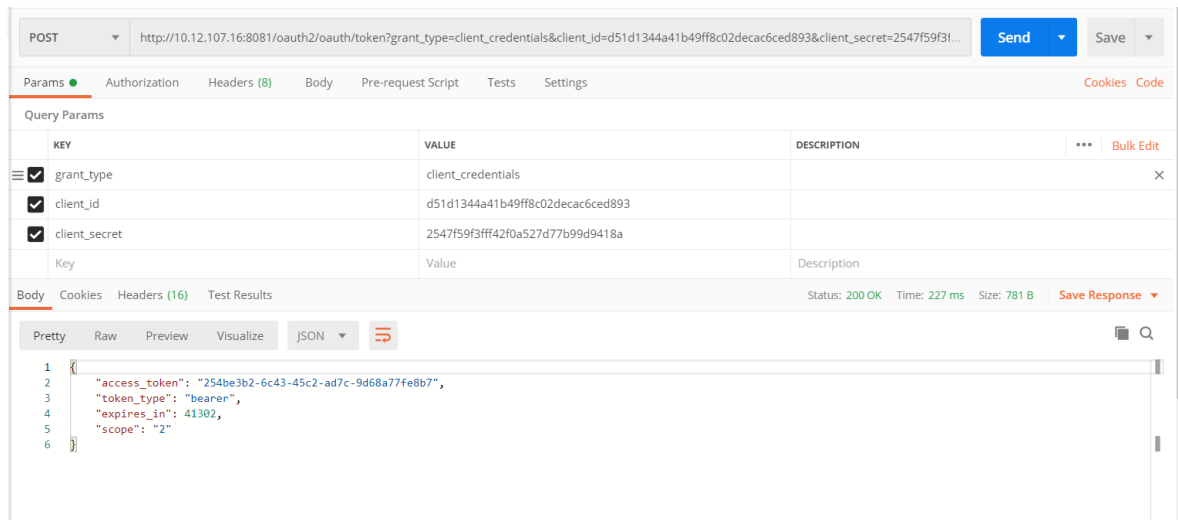
7、站点管理员配置站点角色及相应权限



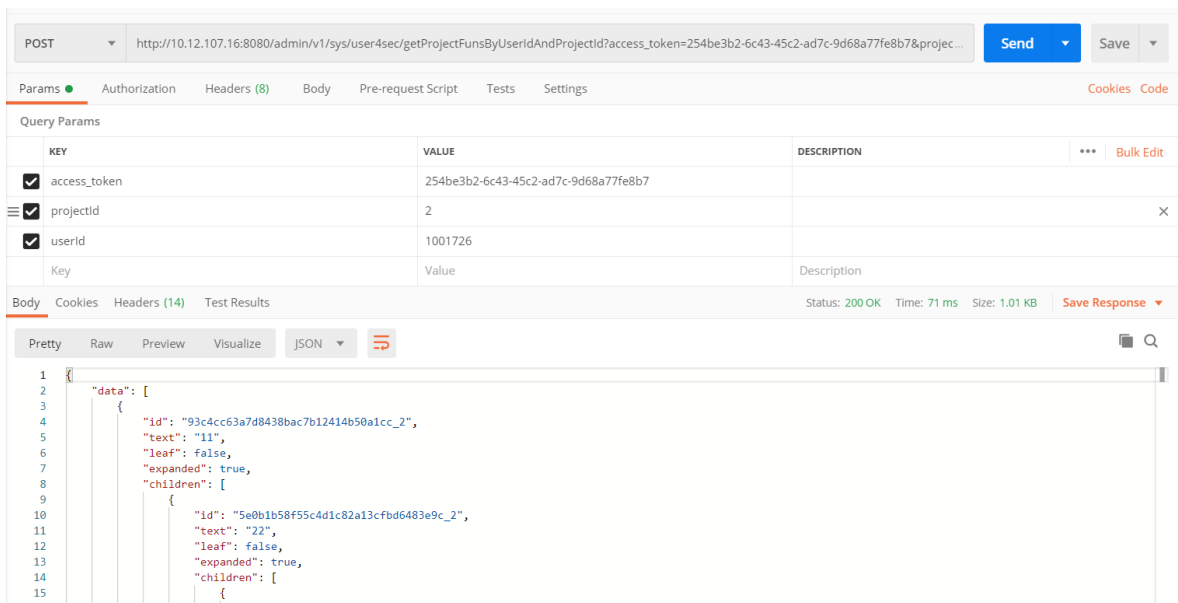
## 8.权限系统操作

### 接口调用:

## 1、获取应用access\_token



## 2、获取用户的功能权限



## 9.SDK/demo下载

## 10. FAQ

### 常见问题自查

#### 1、认证页面输入用户名、密码登录后，提示“用户名/密码错误，请重新输入”

- a、确认用户名、密码输入正确；
- b、使用平台管理员登录进入权限后台重置对应用户密码；
- c、上述处理仍无法登录，请确认三中心oauth2认证服务连接到正确的数据库

#### 2、部门、用户数据有调整未推送到MQ

- a、确认子系统服务连接的MQ与三中心admin权限服务一致；
- b、确认对应站点已开启MQ推送（站点管理-编辑站点信息 查看是否开启）；
- c、平台管理员进入平台的权限后台新增数据是不会推送到MQ的，只有编辑的数据已被若干站点引用才会推送；

#### 3、正常获取token后，接口调用时返回“无接口访问权限，请联系管理员”

在 站点管理-应用功能 中添加该接口后，进入 站点管理-应用资源 菜单,选择资源id为“tc-admin”的数据，点击关联按钮，勾选上步中添加的功能接口

#### 4、正常获取token后，接口调用时返回“Invalid token does not contain resource id (tc-admin)”

检查站点对应用的资源id配置，默认情况下权限服务在oauth2中注册的资源id为“tc-admin”，请将此值配置到相应应用的属性中

#### 5、平台管理员登录后进入平台的权限后台给新用户分配平台管理员角色，该用户登录后在认证中心右上角无“平台级站点”按钮

平台级站点也是一个站点，只是在认证中心有特殊处理。在站点管理中找到平台级站点，点击站点管理员配置选择相应用户即可。

#### 6、内网外部署配置，平台级站点或相应子系统跳转异常

- 1、本地认证回调地址中配置两个（内网地址在前，外网地址在后）
- 2、权限服务（tc-admin）oauth2.properties配置项 tc.front.useReal 设为false
- 3、认证服务（tc-oauth2）global.properties配置项 tc.intranet.ip 设为内网ip
- 4、子系统回调接口调用/oauth/token时，传参redirect\_uri（取三中心跳转过来时state的值），state没有就不传redirect\_uri



## 11. code值对应表

以下表格为响应体code值解释

code值	含义
-1	未登录
-2	无效的Token
200	请求成功
204	请求成功（请求数据有误）
401	token无效/无效的应用Id/无效的时间戳/无效的签名
403	无访问权限
405	请求方式不允许
406	参数不符合要求
408	请求超时
500	系统异常

## 12. 短信相关接口

## 12.1 获取所有可用模板

获取所有可用模板

**URL:** http://权限地址/v1/sys/sms/getSmsTemplateList

**HTTP请求方式:** get

**参数信息:**

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值

其中access\_token在header中，其他参数在paramter中

**成功返回结果示例:**

```
{
 "data": [
 {
 "template": "(%s)验证码: %s。尊敬的用户，您正在使用短信验证码登陆，验证码将于五分钟后失效，为保障您的信息安全，请勿泄露。（如非本人操作，请忽略本短信）",
 "templateType": 1,
 "templateName": "测试",
 "reqExample": "curl --location --request POST 'localhost:8080/admin/v1/sys/sms/send' --header 'access_token: aa093950-8955-487a-84cb-2e45deda9e68' --header 'Content-Type: application/x-www-form-urlencoded' --data-urlencode 'templateId=2' --data-urlencode 'phoneNum=13750858517' --data-urlencode 'test=测试信息' --data-urlencode 'code=123456' --data-urlencode 'projectId=2'",
 "templateId": 2
 }
],
 "success": true,
 "lastTime": "2020-10-09 10:45:03"
}
```

**失败返回结果示例**

```
{
 "success": false,
 "code": 401,
 "message": "无效的应用Id!"
}
```

**成功/失败** success为 true（成功）、false（失败）失败时有message字段返回

**返回参数data内容说明:** 字段值为空时，字段不返回

字段代码	字段名称	字段类型
------	------	------

字段代码	字段名称	字段类型
template	模板	String
templateType	模板类型	int
templateName	模板名称	String
reqExample	请求样例	String
templateId	模板id	int

## 12.2 发送短信

发送短信

**URL:** http://权限地址/v1/sys/sms/send

**HTTP请求方式:** post

**参数信息:**

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值
projectId	String	是	站点id
templateId	String	是	模板id
phoneNum	String	是	手机号

其中access\_token在header中，其他参数在parameter中。  
模板中配置的变量请一同以表单方式提交，调用方式参考模板返回的请求样例。

**成功返回结果示例:**

```
{
 "message": "短信发送成功!",
 "success": true
}
```

**失败返回结果示例**

```
{
 "success": false,
 "code": 401,
 "message": "无效的应用Id!"
}
```

**成功/失败** success为 true（成功）、false（失败）失败时有message字段返回

## 12.3 发送定时短信

发送定时短信

**URL:** http://权限地址/v1/sys/sms/sendScheduled

**HTTP请求方式:** post

**参数信息:**

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值
projectId	String	是	站点id
templateId	String	是	模板id
phoneNum	String	是	手机号
cronExpression	String	是	调度表达式

其中access\_token在header中，其他参数在paramter中。  
模板中配置的变量请一同以表单方式提交，调用方式参考模板返回的请求样例。

**成功返回结果示例:**

```
{
 "message": "定时短信发送成功!",
 "success": true
}
```

**失败返回结果示例**

```
{
 "success": false,
 "code": 401,
 "message": "无效的应用Id!"
}
```

**成功/失败** success为 true（成功）、false（失败）失败时有message字段返回

## 12.4 短信验证码校验

### 短信验证码校验

**URL:** http://权限地址/v1/sys/sms/checkCode

**HTTP请求方式:** post

**参数信息:**

参数名称	参数类型	必需	说明
access_token	String	是	在4.1取到的access_token值
code	String	是	验证码
phoneNum	String	是	手机号

其中access\_token在header中，其他参数在paramter中。

**成功返回结果示例:**

```
{
 "data": true,
 "success": true
}
```

**失败返回结果示例**

```
{
 "success": false,
 "code": 401,
 "message": "无效的应用Id!"
}
```

**成功/失败** success为 true（成功）、false（失败）失败时有message字段返回