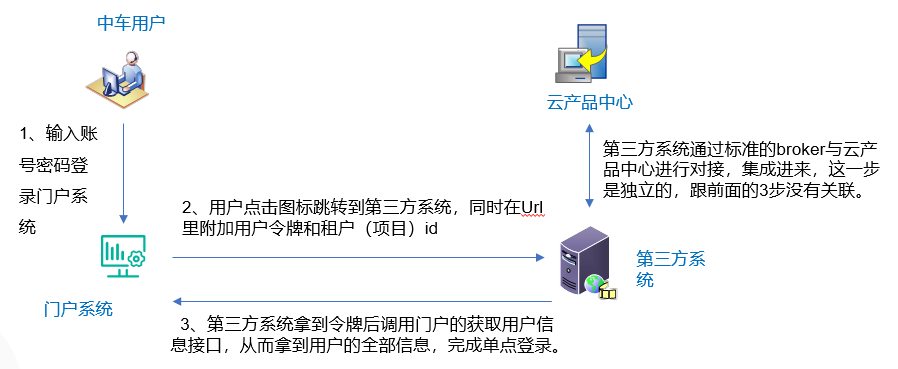
# 一、单点登录的流程及事例



第一步先访问门户，返回令牌和租户id.

跳转示例：http://ip:port/index?sessionId=aegaegtatetaetettaettaetwt&projectId=fadsf4d25

sessionId:当前登录用户令牌

projectId：项目id或租户id

门户获取用户信息接口:

<http://ip:port/indnet/portalUser/getUserPermit> post请求

参数为空，注意要把获取到的令牌放到请求的Header里。

格式为：

Authorization:令牌的值

调用成功后，门户返回的格式如下：

{

"state": true,

"message": "",

"value": {

"userId": "用户id",

"account": "用户帐号",

"userName": "用户名",

"email": "邮箱",

"orgId": "公司id",

"orgName": "\*\*公司名",

"deptId": "部门id",

"deptName": "部门名",

"roles": [

{

"id": "角色id",

"name": "角色名",

"code": "角色编码",

"enabled": "是否可用0-不可用，1可用",

"isCommon": "是否公共角色0-否1是",

"orgId": "所属组织"

}

],

"roleCodes": "所有角色编码",

"roleName": "所有角色名",

"menuCheck": [

"菜单名1",

"菜单名1"

],

"authorization": "当前token",

"userKey": "加密key",

"userCheckKey": "加密key",

//用户所属租户信息 数组类型

"tenants": [

{

                "id": "租户id",

                "name": "租户名",

                "description": "隆轩项目D",

                "orgId": "所属组织id",

                "status": 1, // 是否启用0-否；1-是

                 //用户所属租户下的项目信息 数组类型

"projects": [

                    {

                        "id": "项目id",

                        "name": "项目名",

                        "code": "项目编码",

                        "status": 1, // 是否启用0-否；1-是

                        "userProRoles": [用户在此项目的角色编码],

                    }

]

}

]

}

}

第三方系统调用门户获取用户token接口:

<http://ip:port/indnet/portalUser/getUserToken>

请求类型:post

入参：

{

“sysUser”:”第三方系统识别用户，这个由门户系统给出”,

“sysPassword”:”第三方系统识别用户的密码，门户会给出原密码，但是要做sm3加密，前端的加密方法和加密类可以找门户的研发要”,

“userAccount”：“真正登录的用户账号”

}

调用成功后，门户返回的格式如下：

{

"state": true,

"message": "",

"value": {

"token": "用户token",

}

}

# 二、门户提供的可供调用的API接口

## 请求方式

门户提供接口，供其他系统调用，调用方式为POST

## 查询验证用户是否在IAM存在

请求url：http(s)://ip:port/user/checkUserIam

其中ip和port为门户的ip或域名，port为端口号

请求头：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数编码 | 参数值 | 是否必须 | 说明 |
| Authorization | Token | 是 | 门户登录时生成的token值 |
| Content-Type | application/json | 否 |  |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数编码 | 参数值 | 是否必须 | 说明 |
| userId | String | 是 | 用户编码 |
| name | String | 是 | 用户名 |

响应编码：200

响应报文代表：如果存在则返回用户信息，如果为空则不存在

响应编码非200值，data会有对应错误信息

## 用户信息查询

用户登录完门户后会生成tokenId，当用户打开第三方平台的时候，把tokenId放到请求的header参数传到paas平台，会视提供通过tokenId查询用户信息的接口，该接口返回用户信息、权限信息、租户信息、组织机构信息等。

请求url：http(s)://ip:port/user/getUserInfo

其中ip和port为门户的ip或域名，port为端口号

请求头：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数编码 | 参数值 | 是否必须 | 说明 |
| Authorization | Token | 是 | 门户登录时生成的token值 |
| Content-Type | application/json | 否 |  |

请求参数：无

响应编码：200

响应报文：

{

"state": true,

"message": "",

"value": {

"userId": "用户id",

"account": "用户帐号",

"userName": "用户名",

"email": "邮箱",

"orgId": "公司id",

"orgName": "\*\*公司名",

"deptId": "部门id",

"deptName": "部门名",

"roles": [

{

"id": "角色id",

"name": "角色名",

"code": "角色编码",

"enabled": "是否可用0-不可用，1可用",

"isCommon": "是否公共角色0-否1是",

"orgId": "所属组织"

}

],

"roleCodes": "所有角色编码",

"roleName": "所有角色名",

"menuCheck": [

"菜单名1",

"菜单名1"

],

"authorization": "当前token",

"userKey": "加密key",

"userCheckKey": "加密key",

//用户所属租户信息 数组类型

"tenants": [

{

                "id": "租户id",

                "name": "租户名",

                "description": "隆轩项目D",

                "orgId": "所属组织id",

                "status": 1, // 是否启用0-否；1-是

                 //用户所属租户下的项目信息 数组类型

"projects": [

                    {

                        "id": "项目id",

                        "name": "项目名",

                        "code": "项目编码",

                        "status": 1, // 是否启用0-否；1-是

                        "userProRoles": [用户在此项目的角色编码],

                    }

]

}

]

}

}

## 消息中心接收消息接口

请求url：http(s)://ip:port/user/messageInfo/

其中ip和port为门户的ip或域名，port为端口号

请求头：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数编码 | 参数值 | 是否必须 | 说明 |
| Authorization | Token | 是 | 门户登录时生成的token值 |
| Content-Type | application/json | 否 |  |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数编码 | 参数值 | 是否必须 | 说明 |
| infoType | String | 是 | 消息类型 |
| infoResource | String | 是 | 消息来源 |
| infoTitle | String | 是 | 消息标题 |
| data | String | 是 | 消息体 |
| userId | List<string> | 是 | 接收消息的用户 |

响应编码：200

响应报文：

{

code:200

message:””

}

响应编码非200值，message会有对应错误信息

# 三、门户的同步信息接口

## 请求方式

第三方系统需要提供一个信息接收接口url，门户会在相关信息发生变更的时候调用配置好的第三方接口url，并把相关的信息作为入参传递过去，请求方式为POST。

响应统一为：

响应编码：200

响应报文：

{

code:200

message:””

}

## 请求参数

请求头参数

|  |  |  |
| --- | --- | --- |
| 参数编码 | 参数值 | 说明 |
| Authorization | Basic secretKey | 会视门户登录时生成的token值 |
| Content-Type | application/json |  |

请求参数

|  |  |  |
| --- | --- | --- |
| 参数编码 | 参数值 | 说明 |
| flag | String | 11:项目增加  12:项目删除  13:项目修改  21:菜单权限增加或修改  22: 菜单权限删除  31:组织机构增加  32:组织机构删除  33:组织机构修改  41:角色增加  42:角色删除  43:角色修改  51:用户信息增加  52:用户信息删除  53:用户信息修改  61:租户信息增加  62:租户信息删除  63:租户信息修改  71:子公司信息增加或修改  72:子公司信息删除  511:用户和角色关系的增加或修改  512:用户和角色关系的删除  521:用户和组织关系的增加或修改  522:用户和组织关系的删除  531:用户和项目关系的增加  532:用户和项目关系的删除  533:用户和项目关系的修改 |
| data | Object | data数据部分根据flag值确定 |

### 2.1操作项目对象

#### 增加项目

当flag=11时，增加项目时data对象

|  |  |  |  |
| --- | --- | --- | --- |
| 参数编码 | 参数类型 | 必传 | 说明 |
| projectId | Long | 是 | 项目唯一标识id，会视在创建时生成，根据标识id存在与否进行插入或更新 |
| projectCode | String | 是 | 项目编码 |
| projectName | String | 是 | 项目名称 |
| createTime | String | 否 | 创建时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| createUserId | String | 否 | 创建人登录id,该字段来自用户唯一标识 |
| updateTime | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |
| tenantId | String | 是 | 创建人所属的租户id |
| projectStatus | String | 是 | 0：注销 1：启用 2：保存 |

入参示例：

{

"flag": "11",

"data": {

"projectId":123456789,

"projectCode": "项目唯一code",

"projectName": "项目名字",

"createTime": "20230209133000",

"createUserId": "123456",

"updateTime":"20230209123030",

"updateUserId":"456789",

"tenantId": "123456789",

"projectStatus": "1"

}

}

#### 删除项目

当flag=12时，data对象结构如下

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 参数类型 | 是否必须 | 说明 |
| projectId | String | 是 | 项目唯一标识id，由会视生成的 |
| updateTime | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |

入参示例：

{

"flag": "12",

"data": {

"projectId": "项目唯一id",

"updateTime":"20230209123030",

"updateUserId":"456789"

}

}

#### 修改项目

当flag=13时，修改项目时data对象

|  |  |  |  |
| --- | --- | --- | --- |
| 参数编码 | 参数类型 | 必传 | 说明 |
| projectId | Long | 是 | 项目唯一标识id，会视在创建时生成，根据标识id存在与否进行插入或更新 |
| projectCode | String | 否 | 项目编码 |
| projectName | String | 否 | 项目名称 |
| createTime | String | 否 | 创建时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| createUserId | String | 否 | 创建人登录id,该字段来自用户唯一标识 |
| updateTime | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |
| tenantId | String | 否 | 创建人所属的租户id |
| projectStatus | String | 否 | 1：开启，0：关闭 |

入参示例：

{

"flag": "13",

"data": {

"projectId":123456789,

"projectCode": "项目唯一code",

"projectName": "项目名字",

"createTime": "20230209133000",

"createUserId": "123456",

"updateTime":"20230209123030",

"updateUserId":"456789",

"tenantId": "123456789",

"projectStatus": "1"

}

}

### 2.2操作菜单和权限相关信息

#### 新增修改菜单权限

当flag=21时新增菜单机构信息，data对象结构如下，表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 参数类型 | 是否必须 | 说明 |
| menuinfo | Object | 是 |  |
| resourceNew | Object | 是 |  |

menuinfo的对象结构，

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 参数类型 | 是否必须 | 说明 |
| ord | long | 否 | 排序 |
| menuDesc | String | 否 | 菜单描述 |
| url | String | 是 | 菜单url |
| menuCode | String | 是 | 菜单编码，唯一标识id，由会视生成的 |
| menuName | String | 是 | 菜单名字 |
| menuinfoCode | String | 否 | 上级菜单id，该字段来自上级menuCode字段 |
| tenantId | String | 是 | 租户id |
| icon | String | 否 | 菜单图标 |
| isDisable | int | 否 | 菜单是否失效标志0:失效，1有效 |
| serviceAppCode | String | 否 | 菜单所属的服务提供方的appCode |
| createUserId | String | 否 | 创建者登录id，该字段来自用户唯一标识 |
| createTime | String | 否 | 创建时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateTime | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |
| isOpen | int | 否 | 是否在新标签页打开默认值0 |
| isHotMenu | Int | 否 |  |
| menuinfoTypeId | int | 否 | 菜单类型id （1：系统菜单，2：应用菜单） |
| roleinfoCodeList | List<String> | 是 | 该菜单拥有的角色，和4.4 关联 |

resourceNew中data数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 参数类型 | 是否必须 | 说明 |
| resouceCode | String | 是 | 菜单资源唯一编号 |
| resouceName | String | 否 | 资源名字 |
| resouceDesc | String | 否 | 资源描述 |
| resouceUrl | String | 否 | 资源url |
| parentResourcesCode | String | 否 | 上级父类资源Code，来自resouceCode |
| resourceType | String | 否 | 资源类型 |
| tenantId | String | 是 | 租户id |
| roleinfoCodeList | List<String> | 是 | 该菜单拥有的角色code，和4.4 关联 |
| createTime | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| createUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |
| updateTime | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |

入参示例：

{

"flag": "21",

"data": {

"menuinfo":{

"ord":1,

"menuDesc":"菜单描述",

"url":"菜单路径",

"menuCode":"唯一标识id",

"menuName":"菜单名字",

"menuinfoCode":"上级菜单menuCode",

"tenantId":"租户id",

"icon":"菜单图标",

"isDisable":1,

"serviceAppCode":"sdf",

"createUserId":"admin",

"createTime":"20230209133000",

"updateTime":"20230209133000",

"updateUserId":"123456",

"isOpen":0,

"isHotMenu":**null**,

"roleinfoCodeList":["123456","123"],

"menuinfoTypeId":12

},

"resourceNew":{

"resouceCode":"123456",

"resouceName":"asg",

"resouceDesc":"描述",

"resouceUrl":"456",

"parentResourcesCode":"123456",

"resourceType":"资源类型",

"appId":"1",

"tenantId":"租户id",

"roleinfoCodeList":["1","2"],

"createUserId":"admin",

"createTime":"20230209133000",

"updateTime":"20230209133000",

"updateUserId":"123456"

}

}

}

#### 删除菜单权限

当flag=22时删除菜单机构信息，data对象结构如下

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 参数类型 | 是否必须 | 说明 |
| menuinfo | Object | 是 |  |
| resourceNew | Object | 是 |  |

menuinfo的对象结构，

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 参数类型 | 是否必须 | 说明 |
| menuCode | String | 是 | 菜单编码，唯一标识id，由会视生成的 |
| roleinfoCodeList | List<String> | 否 | 删除该菜单拥有的角色，和4.4 关联 |
| updateTime | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |

resourceNew中数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 参数类型 | 是否必须 | 说明 |
| resouceCode | String | 是 | 菜单资源唯一编号 |
| roleinfoCodeList | List<String> | 否 | 删除该菜单拥有的角色id，和4.4 关联 |
| updateTime | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |

入参示例：

{

"flag": "21",

"data": {

"menuinfo":{

"menuCode":"唯一标识id",

"roleinfoCodeList":["123456"] ,

"updateTime":"20230209133000",

"updateUserId":"123456"

},

"resourceNew":{

"resouceCode":"123456",

"roleinfoCodeList":["1","2"],

"updateTime":"20230209133000",

"updateUserId":"123456"

}

}

}

### 2.3操作组织机构信息

#### 新增组织机构

当flag=31时，增加组织机构时data对象

|  |  |  |  |
| --- | --- | --- | --- |
| 参数编码 | 参数类型 | 必传 | 说明 |
| id | String | 是 | 主键id，由会视生成唯一id |
| orgName | String | 是 | 组织机构名字 |
| path | String | 否 | 机构路径，对应多级id |
| orgtypeId | String | 否 | 组织类型id（来自基础表，组织类型） |
| orginfoId | String | 否 | 上级组织机构id（都是由会试生成） |
| memo | String | 否 | 备注 |
| status | Long | 是 | 机构状态（1:可用/0:不可用） |
| createTime | String | 否 | 创建时间，格式yyyyMMddHHmmss,如2023年2月9日13：30:00则为20230209133000， |
| createUserId | String | 否 | 创建人 登录id，该字段来自用户唯一标识 |
| updateTime | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |
| tenantId | String | 否 | 创建人所属的租户id，不传租户默认就是tenant\_system |

入参示例：

{

"flag": "31",

"data": {

"id":"2132222",

"orgName": "组织机构名字",

"path": "/2/1",

"orgtypeId":"111",

"orginfoId":"112322",

"memo":"备注"，

"status": 1,

"createTime": "20230209133000",

"createUserId": "123456",

"updateTime":"20230209140000",

"updateUserId":"123456",

"tenantId": "tenant\_id\_1"

}

}

#### 删除组织机构

当flag=32时，data对象结构如下

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 参数类型 | 是否必须 | 说明 |
| id | String | 是 | 主键id |
| updateTime | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |

入参示例：

{

"flag": "32",

"data": {

"id": "12322",

"updateTime":"20230209140000",

"updateUserId":"123456"

}

}

#### 修改组织机构

当flag=33时，增加组织机构时data对象

|  |  |  |  |
| --- | --- | --- | --- |
| 参数编码 | 参数类型 | 必传 | 说明 |
| id | String | 是 | 主键id，由会视生成唯一id |
| orgName | String | 否 | 组织机构名字 |
| path | String | 否 | 机构路径，对应多级id |
| orgtypeId | String | 否 | 组织类型id（来自基础表，组织类型） |
| orginfoId | String | 否 | 上级组织机构id（都是由会试生成） |
| memo | String | 否 | 备注 |
| status | Long | 否 | 机构状态（1:可用/0:不可用） |
| createTime | String | 否 | 创建时间，格式yyyyMMddHHmmss,如2023年2月9日13：30:00则为20230209133000， |
| createUserId | String | 否 | 创建人 登录id，该字段来自用户唯一标识 |
| updateTime | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |
| tenantId | String | 否 | 创建人所属的租户id，不传租户默认就是tenant\_system |

入参示例：

{

"flag": "33",

"data": {

"id":"2132222",

"orgName": "组织机构名字",

"path": "/2/1",

"orgtypeId":"111",

"orginfoId":"112322",

"memo":"备注"，

"status": 1,

"createTime": "20230209133000",

"createUserId": "123456",

"updateTime":"20230209140000",

"updateUserId":"123456",

"tenantId": "tenant\_id\_1"

}

}

### 2.4操作角色信息

#### 新增角色信息

当flag=41时，新增角色时data对象

|  |  |  |  |
| --- | --- | --- | --- |
| 参数编码 | 参数类型 | 必传 | 说明 |
| roleCode | String | 是 | 角色编码 |
| roleName | String | 否 | 角色名称 |
| roleinfoCode | String | 否 | 上级角色Code |
| createTime | String | 否 | 创建时间，格式yyyyMMddHHmmss,如2023年2月9日13：30:00则为20230209133000， |
| createUserId | String | 否 | 创建人id，该字段来自用户唯一标识 |
| updateTime | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |
| tenantId | String | 是 | 创建人所属的租户id |

入参示例：

{

"flag": "41",

"data": {

"roleCode":"2132222",

"roleName": "组织机构名字",

"roleinfoCode":"112322",

"createTime": "20230209133000",

"createUserId": "123456789",

"upateTime":"20230209140000",

"upateUserId":"123456",

"tenantId": "tenant\_id\_1"

}

}

1. 测试数据
2. {
3. "flag":"41",
4. "data":{
5. "roleCode":"SUPER\_ADMIN",
6. "roleName":"超级管理员",
7. "roleinfoCode":"1",
8. "createTime":"20230209133000",
9. "createUserId":"1547409851726008320",
10. "upateTime":"20230209140000",
11. "upateUserId":"1547409851726008320",
12. "tenantId":"1547409848181821440"
13. }
14. }

#### 删除角色信息

当flag=42时，data对象结构如下

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 参数类型 | 是否必须 | 说明 |
| roleCode | String | 是 | 角色编码 |
| updateTime | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |

入参示例：

{

"flag": "42",

"data": {

"roleCode": "12322",

"upateTime":"20230209140000",

"upateUserId":"123456"

}

}

#### 修改角色信息

当flag=43时，修改角色时data对象

|  |  |  |  |
| --- | --- | --- | --- |
| 参数编码 | 参数类型 | 必传 | 说明 |
| roleCode | String | 是 | 角色编码 |
| roleName | String | 否 | 角色名称 |
| roleinfoCode | String | 否 | 上级角色Code |
| createTime | String | 否 | 创建时间，格式yyyyMMddHHmmss,如2023年2月9日13：30:00则为20230209133000， |
| createUserId | String | 否 | 创建人id，该字段来自用户唯一标识 |
| updateTime | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |
| tenantId | String | 否 | 创建人所属的租户id |

入参示例：

{

"flag": "43",

"data": {

"roleCode":"2132222",

"roleName": "组织机构名字",

"roleinfoCode":"112322",

"createTime": "20230209133000",

"createUserId": "123456789",

"upateTime":"20230209140000",

"upateUserId":"123456",

"tenantId": "tenant\_id\_1"

}

}

1. 测试数据
2. {
3. "flag": "43",
4. "data": {
5. "roleCode": "SUPER\_ADMIN",
6. "roleName": "超级管理员1",
7. "roleinfoCode": "1",
8. "createTime": "20230209133000",
9. "createUserId": "123456789",
10. "upateTime": "20230209140000",
11. "upateUserId": "1547409851726008320",
12. "tenantId": "1547409848181821440"
13. }
14. }

### 2.5操作用户（租户成员）对象

#### 新增用户信息

当flag=51时传入userinfo对象参数，新增用户信息

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必须 | 说明 |
| userName | String | 是 | 用户姓名 |
| sexId | Long | 否 | 性别 1:男，2:女 |
| loginId | String | 是 | 登录id，唯一id，由由会视生成 |
| password | String | 否 | 密码 |
| Email | String | 否 | 邮箱 |
| Mobile | String | 否 | 手机 |
| Memo | String | 否 | 备注 |
| isAdmin | Int | 否 | 是否管理员，1:是/0:否 |
| isNormal | Int | 否 | 是否正常，1:是/0:否 |
| updateDate | String | 否 | 更新日期,格式yyyyMMddHHmmss,如2023年2月9日13：30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人，该字段来自用户唯一标识 |
| createDate | String | 否 | 创建日期,格式yyyyMMddHHmmss,如2023年2月9日13：30:00则为20230209133000 |
| createUserId | String | 否 | 创建人，该字段来自用户唯一标识 |
| pwdExpireDate | String | 否 | 密码过期时间,格式yyyyMMddHHmmss,如2023年2月9日13：30:00则为20231209133000 |
| imgUrl | String | 否 | 用户头像 |
| reserve3 | String | 否 | 密码加密盐值 |
| tenantId | String | 否 | 租户id，关联2.6，不传的话，就默认系统tenant\_system |

入参示例：

1. {
2. "flag": "51",
3. "data": [{
4. "userName":"123",
5. "sexId":1,
6. "loginId": "用户唯一id",
7. "password":"123",
8. "email":"bdev@172.com",
9. "mobile":"16601330458",
10. "memo":"备注",
11. "isAdmin":0,
12. "isNormal":1,
13. "updateDate":"20230209133000",
14. "updateUserId":"456789",
15. "createDate":"20230209133000",
16. "createUserId":"123456",
17. "pwdExpireDate":"20231209133000",
18. "imgUrl":"/root/img/test.jpg",
19. "reserve3":"32",
20. "tenantId":"123456789"
21. },
22. {
23. "userName":"123",
24. "sexId":1,
25. "loginId": "用户唯一id",
26. "password":"123",
27. "email":"bdev@172.com",
28. "mobile":"16601330458",
29. "memo":"备注",
30. "isAdmin":0,
31. "isNormal":1,
32. "updateDate":"20230209133000",
33. "updateUserId":"456789",
34. "createDate":"20230209133000",
35. "createUserId":"123456",
36. "pwdExpireDate":"20231209133000",
37. "imgUrl":"/root/img/test.jpg",
38. "reserve3":"32",
39. "tenantId":"123456789"
40. }
41. ]
42. }

#### 删除用户信息

当flag= 52时传入userinfo对象参数，删除用户信息

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必须 | 说明 |
| loginId | String | 是 | 登录id，唯一id，由由会视生成 |
| updateDate | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |

入参示例：

1. {
2. "flag": "52",
3. "data": [
4. {
5. "loginId": "用户唯一id",
6. "updateDate":"20230209133000",
7. "updateUserId":"456789"
8. },
9. {
10. "loginId": "用户唯一id",
11. "updateDate":"20230209133000",
12. "updateUserId":"456789"
13. }]
14. }

#### 修改用户信息

当flag=53时传入userinfo对象参数，修改用户信息

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必须 | 说明 |
| userName | String | 是 | 用户姓名 |
| sexId | Long | 否 | 性别 1:男，2:女 |
| loginId | String | 否 | 登录id，唯一id，由由会视生成 |
| password | String | 否 | 密码 |
| Email | String | 否 | 邮箱 |
| Mobile | String | 否 | 手机 |
| Memo | String | 否 | 备注 |
| isAdmin | Int | 否 | 是否管理员，1:是/0:否 |
| isNormal | Int | 否 | 是否正常，1:是/0:否 |
| updateDate | String | 否 | 更新日期,格式yyyyMMddHHmmss,如2023年2月9日13：30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人，该字段来自用户唯一标识 |
| createDate | String | 否 | 创建日期,格式yyyyMMddHHmmss,如2023年2月9日13：30:00则为20230209133000 |
| createUserId | String | 否 | 创建人，该字段来自用户唯一标识 |
| pwdExpireDate | String | 否 | 密码过期时间,格式yyyyMMddHHmmss,如2023年2月9日13：30:00则为20231209133000 |
| imgUrl | String | 否 | 用户头像 |
| reserve3 | String | 否 | 密码加密盐值 |
| tenantId | String | 否 | 租户id，关联4.6 |

入参示例：

{

"flag": "53",

"data": {

"userName":"123",

"sexId":1,

"loginId": "用户唯一id",

"password":"123",

"email":"[bdev@172.com](mailto:bdev@172.com)",

"mobile":"16601330458",

"memo":"备注",

"isAdmin":0,

"isNormal":1,

"updateDate":"20230209133000",

"updateUserId":"456789",

"createDate":"20230209133000",

"createUserId":"123456",

"pwdExpireDate":"20231209133000",

"imgUrl":"/root/img/test.jpg",

"reserve3":"32",

"tenantId":"123456789"

}

}

### 2.6操作租户对象

#### 新增租户信息

当flag=61时增加租户时data对象

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必须 | 说明 |
| tenantId | String | 是 | 该用户所属租户id |
| tenantCode | String | 是 | 租户编码（可以和tenantId一样） |
| Remark | String | 否 | 备注 |
| tenantName | String | 是 | 租户名字 |
| isSystem | Int | 否 | 是否系统1:是/0:否 |
| groupCode | String | 是 | 子公司编码code，关联4.7 (对应租户的Org\_id) |
| createTime | String | 否 | 创建时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| createUserId | String | 否 | 创建人登录id,该字段来自用户唯一标识 |
| updateTime | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |
| tenantManageId | String | 是 | 租户管理员id,该字段来自用户唯一标识 |

入参示例：

{

"flag": "61",

"data": {

"tenantId": "123456789",

"tenantCode": "123456789",

"remark": "备注",

"tenantName": "租户名字",

"isSystem":0,

"groupCode":"456"，

"createTime":"20230214120000",

"createUserId":"123",

"updateTime":"20230214142020",

"updateUserId":"456",

"tenantManageId":"123456789"

}

}

1. 测试参数
2. {
3. "flag": "61",
4. "data": {
5. "tenantId": "1547409848181821440",
6. "tenantCode": "1547409848181821440",
7. "remark": "备注",
8. "tenantName": "租户名字测试1",
9. "isSystem":0,
10. "groupCode":"test1",
11. "createTime":"20230214120000",
12. "createUserId":"1547409851726008320",
13. "updateTime":"20230214142020",
14. "updateUserId":"1547409851726008320"
15. }
16. }

#### 删除租户信息

当flag=62时删除租户时data对象

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必须 | 说明 |
| tenantId | String | 是 | 该用户所属租户id |
| updateTime | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |

入参示例：

{

"flag": "62",

"data": {

"tenantId": "123456789",

"updateTime":"20230214142020",

"updateUserId":"456"

}

}

#### 修改租户信息

当flag=63时修改租户时data对象

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必须 | 说明 |
| tenantId | String | 是 | 该用户所属租户id |
| tenantCode | String | 否 | 租户编码（可以和tenantId一样） |
| Remark | String | 否 | 备注 |
| tenantName | String | 否 | 租户名字 |
| isSystem | Int | 否 | 是否系统1:是/0:否 |
| groupCode | String | 否 | 子公司编码code，关联4.7 |
| createTime | String | 否 | 创建时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| createUserId | String | 否 | 创建人登录id,该字段来自用户唯一标识 |
| updateTime | String | 否 | 更新时间，格式yyyyMMddHHmmss,如2023年2月9日13:30:00则为20230209133000 |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |
| tenantManageId | String | 否 | 租户管理员id,该字段来自用户唯一标识 |

入参示例：

{

"flag": "63",

"data": {

"tenantId": "123456789",

"tenantCode": "123456789",

"remark": "备注",

"tenantName": "租户名字",

"isSystem":0,

"groupCode":"456"，

"createTime":"20230214120000",

"createUserId":"123",

"updateTime":"20230214142020",

"updateUserId":"456",

"tenantManageId":"123456789"

}

}

### 2.7操作子公司对象

#### 新增修改子公司信息

当flag=71时增加或修改子公司时data对象

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必须 | 说明 |
| groupName | String | 是 | 子公司名字，该字段有唯一性限制 |
| groupCode | String | 是 | 子公司编码，由会视生成，唯一的子公司code |
| groupStatus | String | 是 | 0：未生效，1：生效 |
| createUserId | String | 否 | 创建人登录id，该字段来自用户唯一标识 |
| createTime | String | 否 | 创建时间，时间格式yyyyMMddHHmmss |
| updateTime | String | 否 | 更新时间，时间格式yyyyMMddHHmmss |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |

入参示例：

{

"flag": "71",

"data": {

"groupName": "123456789",

"groupCode": "123456",

"groupStatus":"1",

"createUserId": "23456",

"createTime": "20230209133000",

"updateTime": "20230209133000",

"updateUserId":"123456"

}

}

1. 测试参数
2. {
3. "flag": "71",
4. "data": {
5. "groupName": "子公司测试1",
6. "groupCode": "test1",
7. "createUserId": "1547409851726008320",
8. "createTime": "20230209133000",
9. "updateTime": "20230209133000",
10. "updateUserId":"1547409851726008320"
11. }
12. }

#### 删除子公司信息

当flag=72时删除子公司时data对象

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必须 | 说明 |
| groupCode | String | 是 | 分组编码，由租户生成，唯一的子公司code |
| updateTime | String | 否 | 更新时间，时间格式yyyyMMddHHmmss |
| updateUserId | String | 否 | 更新人登录id,该字段来自用户唯一标识 |

入参示例：

{

"flag": "72",

"data": {

"groupCode": "123456",

"updateTime": "20230209133000",

"updateUserId":"123456"

}

}

### 2.8操作用户和角色关系对象

#### 新增修改用户拥有的角色

当flag=511时新增修改用户拥有的角色时data对象

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必须 | 说明 |
| loginId | String | 是 | 用户唯一登录id |
| roleinfoCodeList | List<String> | 是 | 该用户所拥有的角色code，关联4.4 |

入参示例：

{

"flag": "511",

"data": {

"loginId": "123456789",

"roleinfoCodeList": ["123456789","56478"]

}

}

#### 删除用户拥有的角色

当flag=512时删除用户拥有的角色时data对象

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必须 | 说明 |
| loginId | String | 是 | 用户唯一登录id |
| roleinfoCodeList | List<int> | 是 | 该用户所拥有的角色code |

入参示例：

{

"flag": "512",

"data": {

"loginId": "123456789",

"roleinfoCodeList": ["123456789","56478"]

}

}

### 2.9操作用户和组织关系对象

#### 新增修改用户组织关系角色

当flag=521时新增修改用户拥有的角色时data对象

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必须 | 说明 |
| loginId | String | 是 | 用户唯一登录id |
| orginfoId | String | 是 | 该用户所拥有的组织机构id，关联4.3 |

入参示例：

{

"flag": "521",

"data": {

"loginId": "123456789",

"orginfoId": "123456789"

}

}

#### 删除用户组织关系

当flag=522 时删除用户拥有的角色时data对象

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必须 | 说明 |
| loginId | String | 是 | 用户唯一登录id |
| orginfoId | String | 是 | 该用户所拥有的组织机构id，关联4.3 |

入参示例：

{

"flag": "522",

"data": {

"loginId": "123456789",

"orginfoId": "123456789"

}

}

### 2.10操作用户和项目关系对象

#### 新增用户和项目关系角色

当flag=531时新增用户拥有的角色时data对象，注意此处需要数组形式

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必须 | 说明 |
| loginId | String | 是 | 用户唯一登录id |
| projectId | Long | 是 | 项目id，来自项目表的projectId，关联4.1 |
| projectRoleCodeList | List<String> | 是 | 此用户在项目中拥有的角色，关联4.4 |

入参示例：

1. {
2. "flag": "531",
3. "data": [{
4. "loginId": "123456789",
5. "projectId":456789,
6. "projectRoleCodeList": ["123456789","56478"]
7. },
8. { "loginId": "123456789",
9. "projectId":456789,
10. "projectRoleCodeList": ["123456789","56478"]
11. },
12. ]
13. }

#### 删除用户和项目的角色

当flag=532时删除用户拥有的角色时data对象

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必须 | 说明 |
| loginId | String | 是 | 用户唯一登录id |
| projectId | String | 是 | 项目id，来自项目表的projectId，关联4.1 |
| projectRoleCodeList | List<String> | 是 | 此用户在项目中拥有的角色，关联4.4 |

入参示例：

1. {
2. "flag": "532",
3. "data": [{
4. "loginId": "123456789",
5. "projectId":"456789",
6. "projectRoleCodeList": ["123456789","56478"]
7. },
8. {
9. "loginId": "123456789",
10. "projectId":"456789",
11. "projectRoleCodeList": ["123456789","56478"]
12. }]
13. }

#### 修改用户和项目关系角色

当flag=533时修改用户拥有的角色时data对象

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必须 | 说明 |
| loginId | String | 是 | 用户唯一登录id |
| projectId | Long | 是 | 项目id，来自项目表的projectId，关联4.1 |
| projectRoleCodeList | List<String> | 是 | 此用户在项目中拥有的角色，关联4.4 |

入参示例：

{

"flag": "533",

"data": {

"loginId": "123456789",

"projectId":456789,

"projectRoleCodeList": ["123456789","56478"]

}

}

### 2.11 根据用户ID查询当前用户所属租户

请求路径：/portalTenant/getTenantInfoListByUserId?userId=1607259779849191424

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 是否必须 | 说明 |
| userId | String | 是 | 用户id |

入参示例：

{

"state": **true**,

"message": "",

"value": [

{

"id": "1627953643387940864",//租户ID

"name": "时代租户BB",//租户名称

"description": "时代租户A改成B测试 租户名重复",//租户描述

"orgId": "13762560000625",//所属子公司ID （同groupCode）

"status": 1,//启用状态 0：注销 1：启用 2:保存

"isDelete": 0,//是否删除 0：未删除 1：已删除

"createDate": "2023-02-21 08:49:29",

"createUserId": "1554750555509882880",

"createOrgId": "b4026391df8443ed9a499befd322dac8",

"updateDate": "2023-02-21 12:08:52",

"userId": "1607259779849191424",//用户ID

"isKeeper": 1,//是否是管理员 0：不是 1：是

"total": 0,

"size": 20,

"current": 1,

"pkVal": "1627953643387940864"

}

]

}

## 3.响应报文

result响应报文参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数编码 | 参数类型 | 必传 | 说明 |
| success | boolean | 是 | 是否成功 |
| message | String | 是 | 错误信息 |

成功示例：

{

"success": true,

"message": "成功"

}

失败示例：

{

"success": false,

"message": "租户信息不存在"

}