

用户服务

简介:用户服务

HOST:127.0.0.1:28692

联系人:

Version:1.0

接口路径:/v2/api-docs?group=用户服务

用户服务

统一账号-租户、套餐、账号相关

根据用户名获取账号信息
创建账号
根据手机号获取账号信息
检索账号
创建账号
查询某账号所属租户列表
创建账号并绑定至某租户
用户认证
将已存在账号，加入到某租户
新增套餐
更新套餐
查询所有套餐
查询某租户类型的套餐列表(不包含初始化套餐)
获取某套餐信息
根据手机号判断账号是否存在
根据用户名判断账号是否存在
根据手机号判断账号是否在某租户内
根据用户名判断账号是否在某租户内
根据接入客户端获取应用
发送短信验证码
用户登录
创建租户并关联账号
创建租户
检索租户
获取租户信息
初始化租户套餐
切换租户套餐
将某账号从租户内移除，租户管理员不可移除

统一账号-租户、套餐、账号相关V2.0

根据用户名获取账号信息
创建账号
根据手机号获取账号信息
检索账号
创建账号
查询某账号所属租户列表
创建账号并绑定至某租户
用户认证
将已存在账号，加入到某租户
新增套餐
更新套餐
查询所有套餐

查询某租户类型的套餐列表(不包含初始化套餐)
获取某套餐信息
根据手机号判断账号是否存在
根据用户名判断账号是否存在
根据手机号判断账号是否在某租户内
根据用户名判断账号是否在某租户内
根据接入客户端获取应用
发送短信验证码
用户登录
创建租户并关联账号
创建租户
检索租户
获取租户信息
初始化租户套餐
切换租户套餐
将某账号从租户内移除, 租户管理员不可移除

统一账号-角色权限

绑定角色
获取权限组下所有权限
查根据roleCode获取角色(不包含权限)
修改租户内角色(不包含权限)
查询某租户下角色(不分页, 不包含权限)
创建租户内角色 (不包含权限)
查找某租户下, 多个角色的权限信息集合
查找某租户下, 多个角色的权限信息集合, 并根据权限组组装成为权限树
根据角色编码删除租户内角色, 如果有账号绑定该角色, 禁止删除
角色设置权限(清除+设置)
获取租户内的某个角色信息(包含权限信息)
授权, 获取某用户在多个租户下的权限信息
解绑角色

统一账号-角色权限V2.0

绑定角色
获取权限组下所有权限
查根据roleCode获取角色(不包含权限)
修改租户内角色(不包含权限)
查询某租户下角色(不分页, 不包含权限)
创建租户内角色 (不包含权限)
查找某租户下, 多个角色的权限信息集合
查找某租户下, 多个角色的权限信息集合, 并根据权限组组装成为权限树
根据角色编码删除租户内角色, 如果有账号绑定该角色, 禁止删除
角色设置权限(清除+设置)
获取租户内的某个角色信息(包含权限信息)
授权, 获取某用户在多个租户下的权限信息
解绑角色

统一账号-资源相关

创建应用
修改应用
分页条件查找应用
根据权限加载指定应用的资源
根据应用编码查找应用
删除应用
根据条件查询菜单列表
根据权限编码列表获取菜单
根据应用编码查询菜单列表
根据ID查询菜单
获取多个租户下权限所对应的的资源信息, 并按应用拆分

统一账号-资源相关V2.0

创建应用

- 修改应用
- 分页条件查找应用
- 根据权限加载指定应用的资源
- 根据应用编码查找应用
- 删除应用
- 根据条件查询菜单列表
- 根据权限编码列表获取菜单
- 根据应用编码查询菜单列表
- 根据ID查询菜单
- 获取多个租户下权限所对应的的资源信息，并按应用拆分

统一账号-租户、套餐、账号相关

根据用户名获取账号信息

接口地址: /user/account-information/name/{username}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	AccountDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
id	账号id	integer(int64)	integer(int64)
mobile	手机号	string	
password	密码	string	
salt	盐	string	
username	用户名	string	

响应示例:

```
1 {
2   "id": 0,
3   "mobile": "",
4   "password": "",
5   "salt": "",
6   "username": ""
7 }
```

创建账号

接口地址: /user/accounts

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1 {
2   "mobile": "",
3   "password": "",
4   "username": ""
5 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
createAccountRequest	创建账号请求	body	true	CreateAccountRequestDTO	CreateAccountRequestDTO
mobile	手机号		false	string	
password	密码		false	string	
username	用户名		false	string	

响应状态:

状态码	说明	schema
200	OK	AccountDTO
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
id	账号id	integer(int64)	integer(int64)
mobile	手机号	string	
password	密码	string	
salt	盐	string	
username	用户名	string	

响应示例:

```
1 {
2   "id": 0,
3   "mobile": "",
4   "password": "",
5   "salt": "",
6   "username": ""
7 }
```

根据手机号获取账号信息

接口地址: /user/accounts-information/mobile/{mobile}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
mobile	手机号	path	true	string	

响应状态:

状态码	说明	schema
200	OK	AccountDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
id	账号id	integer(int64)	integer(int64)
mobile	手机号	string	
password	密码	string	
salt	盐	string	
username	用户名	string	

响应示例:

```
1 {
2   "id": 0,
3   "mobile": "",
4   "password": "",
5   "salt": "",
6   "username": ""
7 }
```

检索账号

接口地址: /user/accounts/page

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1 {
2   "mobile": "",
3   "tenantId": 0,
4   "username": ""
5 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
order	顺序	query	true	string	
pageNo	页码	query	true	integer(int32)	
pageSize	每页记录数	query	true	integer(int32)	
sortBy	排序字段	query	true	string	
accountQuery	账号查询条件	body	false	AccountQueryDTO	AccountQueryDTO
mobile	手机号		false	string	
tenantId	所属租户		false	integer	
username	用户名		false	string	

响应状态:

状态码	说明	schema
200	OK	PageVO«AccountDTO»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
counts		integer(int64)	integer(int64)
first		boolean	
items		array	AccountDTO
id	账号id	integer	
mobile	手机号	string	
password	密码	string	
salt	盐	string	
username	用户名	string	
itemsSize		integer(int32)	integer(int32)
page		integer(int32)	integer(int32)
pageSize		integer(int32)	integer(int32)
pages		integer(int32)	integer(int32)

响应示例:

```
1 {
2   "counts": 0,
3   "first": true,
4   "items": [
5     {
6       "id": 0,
7       "mobile": "",
8       "password": "",
9       "salt": "",
10      "username": ""
11    }
12  ],
13  "itemsSize": 0,
14  "page": 0,
15  "pageSize": 0,
16  "pages": 0
17 }
```

创建账号

接口地址: /user/accounts/password

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:


```
1 {
2     "accountId": 0,
3     "password": "",
4     "userName": ""
5 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
accountPwdRequest	重置账号密码	body	true	ChangeAccountPwdDTO	ChangeAccountPwdDTO
accountId	账号Id		false	integer	
password	密码		false	string	
userName	账号名		false	string	

响应状态:

状态码	说明	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

```
1 |
```

查询某账号所属租户列表

接口地址: /user/accounts/tenant-list/{username}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	TenantDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
bundleCode	套餐编码	string	
id	租户id	integer(int64)	integer(int64)
name	租户名称	string	
tenantTypeCode	租户类型编码	string	

响应示例:

```
1  [
2    {
3      "bundleCode": "",
4      "id": 0,
5      "name": "",
6      "tenantTypeCode": ""
7    }
8  ]
```

创建账号并绑定至某租户

接口地址: /user/accounts/tenants/{tenantId}

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1  {
2    "mobile": "",
3    "password": "",
4    "username": ""
5  }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
createAccountRequest	创建账号请求	body	true	CreateAccountRequestDTO	CreateAccountRequestDTO
mobile	手机号		false	string	
password	密码		false	string	
username	用户名		false	string	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

1	
---	--

用户认证

接口地址: /user/authentication

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

1	{
2	"authenticationType": "",
3	"certificate": "",
4	"principal": "",
5	"smsKey": ""
6	}

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
authenticationInfo	认证请求信息	body	true	AuthenticationInfo	AuthenticationInfo
authenticationType	认证类型: 短信快捷认证、用户名密码认证、二维码认证等		false	string	
certificate	凭证: 如密码, 短信认证码等		false	string	
principal	认证主体: 如用户名, 手机号等		false	string	
smsKey	短信认证时需要传key到短信微服务校验		false	string	

响应状态:

状态码	说明	schema
200	OK	AccountDTO
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
id	账号id	integer(int64)	integer(int64)
mobile	手机号	string	
password	密码	string	
salt	盐	string	
username	用户名	string	

响应示例:

```
1 {
2   "id": 0,
3   "mobile": "",
4   "password": "",
5   "salt": "",
6   "username": ""
7 }
```

将已存在账号，加入到某租户

接口地址: /user/bind/accounts/{username}/tenants/{tenantId}

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
tenantId	租户id	path	true	integer(int64)	
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

```
1 |
```

新增套餐

接口地址: `/user/bundles`

请求方式: `POST`

请求数据类型: `application/json`

响应数据类型: `*/*`

接口描述:

请求示例:

```
1  {
2      "ability": "",
3      "code": "",
4      "comment": "",
5      "id": 0,
6      "initialize": true,
7      "name": "",
8      "numberOfApp": 0,
9      "numberOfConcurrent": 0,
10     "numberOfInvocation": 0,
11     "tenantTypeCode": ""
12 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
bundleDTO	套餐信息	body	true	BundleDTO	BundleDTO
ability	套餐包含功能描述,JSON格式的角色与权限		false	string	
code	套餐编码		false	string	
comment	套餐说明		false	string	
id	主键, 套餐id		false	integer	
initialize	是否为初始化套餐,1表示是初始化套餐, 0表示不是		false	boolean	
name	套餐名称		false	string	
numberOfApp	允许创建应用数量		false	integer	
numberOfConcurrent	并发数/秒		false	integer	
numberOfInvocation	API调用次数/月		false	integer	
tenantTypeCode	租户类型编码		false	string	

响应状态:

状态码	说明	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

更新套餐

接口地址: /user/bundles

请求方式: PUT

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1  {
2      "ability": "",
3      "code": "",
4      "comment": "",
5      "id": 0,
6      "initialize": true,
7      "name": "",
8      "numberOfApp": 0,
9      "numberOfConcurrent": 0,
10     "numberOfInvocation": 0,
11     "tenantTypeCode": ""
12 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
bundleDTO	套餐信息	body	true	BundleDTO	BundleDTO
ability	套餐包含功能描述,JSON格式的角色与权限		false	string	
code	套餐编码		false	string	
comment	套餐说明		false	string	
id	主键, 套餐id		false	integer	
initialize	是否为初始化套餐,1表示是初始化套餐, 0表示不是		false	boolean	
name	套餐名称		false	string	
numberOfApp	允许创建应用数量		false	integer	
numberOfConcurrent	并发数/秒		false	integer	
numberOfInvocation	API调用次数/月		false	integer	
tenantTypeCode	租户类型编码		false	string	

响应状态:

状态码	说明	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

查询所有套餐

接口地址: /user/bundles/bundle-list

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	BundleDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
ability	套餐包含功能描述,JSON格式的角色与权限	string	
code	套餐编码	string	
comment	套餐说明	string	
id	主键, 套餐id	integer(int64)	integer(int64)
initialize	是否为初始化套餐,1表示是初始化套餐, 0表示不是	boolean	
name	套餐名称	string	
numberOfApp	允许创建应用数量	integer(int32)	integer(int32)
numberOfConcurrent	并发数/秒	integer(int32)	integer(int32)
numberOfInvocation	API调用次数/月	integer(int32)	integer(int32)
tenantTypeCode	租户类型编码	string	

响应示例:

```
2      {
3          "ability": "",
4          "code": "",
5          "comment": "",
6          "id": 0,
7          "initialize": true,
8          "name": "",
9          "numberOfApp": 0,
10         "numberOfConcurrent": 0,
11         "numberOfInvocation": 0,
12         "tenantTypeCode": ""
13     }
14 ]
```

查询某租户类型的套餐列表(不包含初始化套餐)

接口地址: /user/bundles/tenant-types/{tenantType}/bundle-list

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
tenantType	租户类型	path	true	string	

响应状态:

状态码	说明	schema
200	OK	BundleDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
ability	套餐包含功能描述,JSON格式的角色与权限	string	
code	套餐编码	string	
comment	套餐说明	string	
id	主键, 套餐id	integer(int64)	integer(int64)
initialize	是否为初始化套餐,1表示是初始化套餐, 0表示不是	boolean	
name	套餐名称	string	
numberOfApp	允许创建应用数量	integer(int32)	integer(int32)
numberOfConcurrent	并发数/秒	integer(int32)	integer(int32)
numberOfInvocation	API调用次数/月	integer(int32)	integer(int32)
tenantTypeCode	租户类型编码	string	

响应示例:

```

1  [
2      {
3          "ability": "",
4          "code": "",
5          "comment": "",
6          "id": 0,
7          "initialize": true,
8          "name": "",
9          "numberOfApp": 0,
10         "numberOfConcurrent": 0,
11         "numberOfInvocation": 0,
12         "tenantTypeCode": ""
13     }
14 ]

```

获取某套餐信息

接口地址: /user/bundles/{bundleCode}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
bundleCode	租户套餐	path	true	string	

响应状态:

状态码	说明	schema
200	OK	BundleDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
ability	套餐包含功能描述,JSON格式的角色与权限	string	
code	套餐编码	string	
comment	套餐说明	string	
id	主键, 套餐id	integer(int64)	integer(int64)
initialize	是否为初始化套餐,1表示是初始化套餐, 0表示不是	boolean	
name	套餐名称	string	
numberOfApp	允许创建应用数量	integer(int32)	integer(int32)
numberOfConcurrent	并发数/秒	integer(int32)	integer(int32)
numberOfInvocation	API调用次数/月	integer(int32)	integer(int32)
tenantTypeCode	租户类型编码	string	

响应示例:

```
1  {
2      "ability": "",
3      "code": "",
4      "comment": "",
5      "id": 0,
6      "initialize": true,
7      "name": "",
8      "numberOfApp": 0,
9      "numberOfConcurrent": 0,
10     "numberOfInvocation": 0,
11     "tenantTypeCode": ""
12 }
```

根据手机号判断账号是否存在

接口地址: /user/exist/accounts/mobile/{mobile}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
mobile	手机号	path	true	string	

响应状态:

状态码	说明	schema
200	OK	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

1

根据用户名判断账号是否存在

接口地址: /user/exist/accounts/name/{username}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

1

根据手机号判断账号是否在某租户内

接口地址: /user/exist/accounts/tenants/{tenantId}/mobiles/{mobile}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
mobile	手机号	path	true	string	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

1

根据用户名判断账号是否在某租户内

接口地址: /user/exist/accounts/{username}/tenants/{tenantId}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
tenantId	租户id	path	true	integer(int64)	
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

1

根据接入客户端获取应用

接口地址: /user/getApplicationDTOByClientId/{clientId}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
clientId	客户端id	path	true	string	

响应状态:

状态码	说明	schema
200	OK	ApplicationDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	应用编码,跨租户唯一	string	
id	id	integer(int64)	integer(int64)
name	应用名称	string	
tenantId	应用所属租户	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": "",
3   "id": 0,
4   "name": "",
5   "tenantId": 0
6 }
```

发送短信验证码

接口地址: /user/getMessage

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
phone	手机号	query	true	string	

响应状态:

状态码	说明	schema
200	OK	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

1 |

用户登录

接口地址: /user/login

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1 | {
2 |     "authenticationType": "",
3 |     "certificate": "",
4 |     "principal": "",
5 |     "smsKey": ""
6 | }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
loginRequest	登录请求信息	body	true	LoginRequestDTO	LoginRequestDTO
authenticationType	认证类型：短信快捷认证、用户名密码认证、二维码认证等		false	string	
certificate	凭证：如密码，短信认证码等		false	string	
principal	认证主体：如用户名，手机号等		false	string	
smsKey	短信认证时需要传key		false	string	

响应状态:

状态码	说明	schema
200	OK	LoginInfoDTO
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
id	账号id	integer(int64)	integer(int64)
mobile	手机号	string	
resources	账号所属多个租户下，不同应用的资源，如：{租户A: [{应用1权限信息},{应用2权限信息}],租户B:[{}]}...	ResourceDTO	ResourceDTO
appRes	应用包含资源，按资源类型分包括菜单等资源信息 如 menu:JSONObject	object	
applicationCode	应用编码	string	
applicationName	应用名称	string	
tenantAuthorizationInfoMap	账号所属多个租户下的权限	AuthorizationInfoDTO	AuthorizationInfoDTO
rolePrivilegeMap	授权信息，角色-权限关系,key为角色编码，value为权限集合	object	
tenants	账号所属多个租户	array	TenantDTO
bundleCode	套餐编码	string	
id	租户id	integer	
name	租户名称	string	
tenantTypeCode	租户类型编码	string	
username	用户名	string	

响应示例:

```

1  {
2      "id": 0,
3      "mobile": "",
4      "resources": {
5          "additionalProperties1": {
6              "appRes": {},
7              "applicationCode": "",
8              "applicationName": ""
9          }
10     },
11     "tenantAuthorizationInfoMap": {
12         "additionalProperties1": {
13             "rolePrivilegeMap": {}
14         }
15     },
16     "tenants": [
17         {
18             "bundleCode": "",
19             "id": 0,
20             "name": "",
21             "tenantTypeCode": ""
22         }
23     ],
24     "username": ""
25 }
```

创建租户并关联账号

接口地址: /user/tenantRelateAccount

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1 {
2     "bundleCode": "",
3     "mobile": "",
4     "name": "",
5     "password": "",
6     "tenantTypeCode": "",
7     "username": ""
8 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
createTenantRequest	创建租户请求信息	body	true	CreateTenantRequestDTO	CreateTenantRequestDTO
bundleCode	初始化套餐编码，用于初始化套餐ABILITY，若为空则使用该租户类型的初始化套餐		false	string	
mobile	租户管理员登录手机号		false	string	
name	租户名称		false	string	
password	租户管理员登录密码		false	string	
tenantTypeCode	租户类型编码		false	string	
username	租户管理员登录用户名		false	string	

响应状态:

状态码	说明	schema
200	OK	TenantDTO
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
bundleCode	套餐编码	string	
id	租户id	integer(int64)	integer(int64)
name	租户名称	string	
tenantTypeCode	租户类型编码	string	

响应示例:

```
1 {  
2   "bundleCode": "",  
3   "id": 0,  
4   "name": "",  
5   "tenantTypeCode": ""  
6 }
```

创建租户

接口地址: /user/tenants

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1 {  
2   "bundleCode": "",  
3   "mobile": "",  
4   "name": "",  
5   "password": "",  
6   "tenantTypeCode": "",  
7   "username": ""  
8 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
createTenantRequest	创建租户请求信息	body	true	CreateTenantRequestDTO	CreateTenantRequestDTO
bundleCode	初始化套餐编码，用于初始化套餐ABILITY，若为空则使用该租户类型的初始化套餐		false	string	
mobile	租户管理员登录手机号		false	string	
name	租户名称		false	string	
password	租户管理员登录密码		false	string	
tenantTypeCode	租户类型编码		false	string	
username	租户管理员登录用户名		false	string	

响应状态:

状态码	说明	schema
200	OK	TenantDTO
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
bundleCode	套餐编码	string	
id	租户id	integer(int64)	integer(int64)
name	租户名称	string	
tenantTypeCode	租户类型编码	string	

响应示例:

```
1 {
2   "bundleCode": "",
3   "id": 0,
4   "name": "",
5   "tenantTypeCode": ""
6 }
```

检索租户

接口地址: /user/tenants/page

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1 {
2     "name": "",
3     "tenantTypeCode": ""
4 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
order	顺序	query	true	string	
pageNo	页码	query	true	integer(int32)	
pageSize	每页记录数	query	true	integer(int32)	
sortBy	排序字段	query	true	string	
tenantQuery	租户查询条件	body	false	TenantQueryDTO	TenantQueryDTO
name	租户名称		false	string	
tenantTypeCode	租户类型		false	string	

响应状态:

状态码	说明	schema
200	OK	PageVO«TenantDTO»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
counts		integer(int64)	integer(int64)
first		boolean	
items		array	TenantDTO
bundleCode	套餐编码	string	
id	租户id	integer	
name	租户名称	string	
tenantTypeCode	租户类型编码	string	
itemsSize		integer(int32)	integer(int32)
page		integer(int32)	integer(int32)
pageSize		integer(int32)	integer(int32)
pages		integer(int32)	integer(int32)

响应示例:

```
1  {
2      "counts": 0,
3      "first": true,
4      "items": [
5          {
6              "bundleCode": "",
7              "id": 0,
8              "name": "",
9              "tenantTypeCode": ""
10         }
11     ],
12     "itemsSize": 0,
13     "page": 0,
14     "pageSize": 0,
15     "pages": 0
16 }
```

获取租户信息

接口地址: /user/tenants/{id}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
id	id	path	true	integer(int64)	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	TenantDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
bundleCode	套餐编码	string	
id	租户id	integer(int64)	integer(int64)
name	租户名称	string	
tenantTypeCode	租户类型编码	string	

响应示例:

```
1 {
2   "bundleCode": "",
3   "id": 0,
4   "name": "",
5   "tenantTypeCode": ""
6 }
```

初始化租户套餐

接口地址: /user/tenants/{tenantId}/bundles/{bundleCode}

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
bundleCode	套餐编码	path	true	string	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

1

切换租户套餐

接口地址: /user/tenants/{tenantId}/bundles/{bundleCode}

请求方式: PUT

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
bundleCode	套餐编码	path	true	string	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

1

将某账号从租户内移除，租户管理员不可移除

接口地址: /user/unbind/accounts/{username}/tenants/{tenantId}

请求方式: DELETE

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
tenantId	租户id	path	true	integer(int64)	
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	
204	No Content	
401	Unauthorized	
403	Forbidden	

响应参数:

暂无

响应示例:

统一账号-租户、套餐、账号相关V2.0

根据用户名获取账号信息

接口地址: /user/v2/account-information/name/{username}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	R«AccountDTO»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	AccountDTO	AccountDTO
id	账号id	integer	
mobile	手机号	string	
password	密码	string	
salt	盐	string	
username	用户名	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```

1  {
2      "code": 0,
3      "data": {
4          "id": 0,
5          "mobile": "",
6          "password": "",
7          "salt": "",
8          "username": ""
9      },
10     "extra": {},
11     "isError": true,
12     "isSuccess": true,
13     "msg": "",
14     "path": "",
15     "timestamp": 0
16 }
```

创建账号

接口地址: /user/v2/accounts

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1 {
2   "mobile": "",
3   "password": "",
4   "username": ""
5 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
createAccountRequest	创建账号请求	body	true	CreateAccountRequestDTO	CreateAccountRequestDTO
mobile	手机号		false	string	
password	密码		false	string	
username	用户名		false	string	

响应状态:

状态码	说明	schema
200	OK	R«AccountDTO»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	AccountDTO	AccountDTO
id	账号id	integer	
mobile	手机号	string	
password	密码	string	
salt	盐	string	
username	用户名	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1  {
2      "code": 0,
3      "data": {
4          "id": 0,
5          "mobile": "",
6          "password": "",
7          "salt": "",
8          "username": ""
9      },
10     "extra": {},
11     "isError": true,
12     "isSuccess": true,
13     "msg": "",
14     "path": "",
15     "timestamp": 0
16 }
```

根据手机号获取账号信息

接口地址: `/user/v2/accounts-information/mobile/{mobile}`

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
mobile	手机号	path	true	string	

响应状态:

状态码	说明	schema
200	OK	R«AccountDTO»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	AccountDTO	AccountDTO
id	账号id	integer	
mobile	手机号	string	
password	密码	string	
salt	盐	string	
username	用户名	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": 0,
3   "data": {
4     "id": 0,
5     "mobile": "",
6     "password": "",
7     "salt": "",
8     "username": ""
```

```
9      },
10     "extra": {},
11     "isError": true,
12     "isSuccess": true,
13     "msg": "",
14     "path": "",
15     "timestamp": 0
16 }
```

检索账号

接口地址: /user/v2/accounts/page

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1 {
2     "mobile": "",
3     "tenantId": 0,
4     "username": ""
5 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
order	顺序	query	true	string	
pageNo	页码	query	true	integer(int32)	
pageSize	每页记录数	query	true	integer(int32)	
sortBy	排序字段	query	true	string	
accountQuery	账号查询条件	body	false	AccountQueryDTO	AccountQueryDTO
mobile	手机号		false	string	
tenantId	所属租户		false	integer	
username	用户名		false	string	

响应状态:

状态码	说明	schema
200	OK	R«PageVO«AccountDTO»»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	PageVO«AccountDTO»	PageVO«AccountDTO»
counts		integer	
first		boolean	
items		array	AccountDTO
id	账号id	integer	
mobile	手机号	string	
password	密码	string	
salt	盐	string	
username	用户名	string	
itemsSize		integer	
page		integer	
pageSize		integer	
pages		integer	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

1	{
---	---

```
2      "code": 0,
3      "data": {
4          "counts": 0,
5          "first": true,
6          "items": [
7              {
8                  "id": 0,
9                  "mobile": "",
10                 "password": "",
11                 "salt": "",
12                 "username": ""
13             }
14         ],
15         "itemsSize": 0,
16         "page": 0,
17         "pageSize": 0,
18         "pages": 0
19     },
20     "extra": {},
21     "isError": true,
22     "isSuccess": true,
23     "msg": "",
24     "path": "",
25     "timestamp": 0
26 }
```

创建账号

接口地址: /user/v2/accounts/password

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1 {
2     "accountId": 0,
3     "password": "",
4     "userName": ""
5 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
accountPwdRequest	重置账号密码	body	true	ChangeAccountPwdDTO	ChangeAccountPwdDTO
accountId	账号Id		false	integer	
password	密码		false	string	
userName	账号名		false	string	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1  {
2      "code": 0,
3      "data": true,
4      "extra": {},
5      "isError": true,
6      "isSuccess": true,
7      "msg": "",
8      "path": "",
9      "timestamp": 0
10 }
```

查询某账号所属租户列表

接口地址: /user/v2/accounts/tenant-list/{username}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	R«List«TenantDTO»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	array	TenantDTO
bundleCode	套餐编码	string	
id	租户id	integer	
name	租户名称	string	
tenantTypeCode	租户类型编码	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": 0,
3   "data": [
4     {
5       "bundleCode": "",
6       "id": 0,
7       "name": "",
8       "tenantTypeCode": ""
9     }
10  ],
```

```
11     "extra": {},
12     "isError": true,
13     "isSuccess": true,
14     "msg": "",
15     "path": "",
16     "timestamp": 0
17 }
```

创建账号并绑定至某租户

接口地址: /user/v2/accounts/tenants/{tenantId}

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1 {
2     "mobile": "",
3     "password": "",
4     "username": ""
5 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
createAccountRequest	创建账号请求	body	true	CreateAccountRequestDTO	CreateAccountRequestDTO
mobile	手机号		false	string	
password	密码		false	string	
username	用户名		false	string	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1  {
2      "code": 0,
3      "data": true,
4      "extra": {},
5      "isError": true,
6      "isSuccess": true,
7      "msg": "",
8      "path": "",
9      "timestamp": 0
10 }
```

用户认证

接口地址: /user/v2/authentication

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1  {
2      "authenticationType": "",
3      "certificate": "",
4      "principal": "",
5      "smsKey": ""
6  }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
authenticationInfo	认证请求信息	body	true	AuthenticationInfo	AuthenticationInfo
authenticationType	认证类型: 短信快捷认证、用户名密码认证、二维码认证等		false	string	
certificate	凭证: 如密码, 短信认证码等		false	string	
principal	认证主体: 如用户名, 手机号等		false	string	
smsKey	短信认证时需要传key到短信微服务校验		false	string	

响应状态:

状态码	说明	schema
200	OK	R«AccountDTO»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	AccountDTO	AccountDTO
id	账号id	integer	
mobile	手机号	string	
password	密码	string	
salt	盐	string	
username	用户名	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```

1  {
2      "code": 0,
3      "data": {
4          "id": 0,
5          "mobile": "",
6          "password": "",
7          "salt": "",
8          "username": ""
9      },
10     "extra": {},
11     "isError": true,
12     "isSuccess": true,
13     "msg": "",
14     "path": "",
15     "timestamp": 0
16 }

```

将已存在账号，加入到某租户

接口地址: /user/v2/bind/accounts/{username}/tenants/{tenantId}

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
tenantId	租户id	path	true	integer(int64)	
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": 0,
3   "data": true,
4   "extra": {},
5   "isError": true,
6   "isSuccess": true,
7   "msg": "",
8   "path": "",
9   "timestamp": 0
10 }
```

新增套餐

接口地址: /user/v2/bundles

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1  {
2      "ability": "",
3      "code": "",
4      "comment": "",
5      "id": 0,
6      "initialize": true,
7      "name": "",
8      "numberOfApp": 0,
9      "numberOfConcurrent": 0,
10     "numberOfInvocation": 0,
11     "tenantTypeCode": ""
12 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
bundleDTO	套餐信息	body	true	BundleDTO	BundleDTO
ability	套餐包含功能描述,JSON格式的角色与权限		false	string	
code	套餐编码		false	string	
comment	套餐说明		false	string	
id	主键, 套餐id		false	integer	
initialize	是否为初始化套餐,1表示是初始化套餐, 0表示不是		false	boolean	
name	套餐名称		false	string	
numberOfApp	允许创建应用数量		false	integer	
numberOfConcurrent	并发数/秒		false	integer	
numberOfInvocation	API调用次数/月		false	integer	
tenantTypeCode	租户类型编码		false	string	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1  {
2      "code": 0,
3      "data": true,
4      "extra": {},
5      "isError": true,
6      "isSuccess": true,
7      "msg": "",
8      "path": "",
9      "timestamp": 0
10 }
```

更新套餐

接口地址: /user/v2/bundles

请求方式: PUT

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1  {
2      "ability": "",
3      "code": "",
4      "comment": "",
5      "id": 0,
6      "initialize": true,
7      "name": "",
8      "numberOfApp": 0,
9      "numberOfConcurrent": 0,
10     "numberOfInvocation": 0,
11     "tenantTypeCode": ""
12 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
bundleDTO	套餐信息	body	true	BundleDTO	BundleDTO
ability	套餐包含功能描述JSON格式的角色与权限		false	string	
code	套餐编码		false	string	
comment	套餐说明		false	string	
id	主键, 套餐id		false	integer	
initialize	是否为初始化套餐,1表示是初始化套餐, 0表示不是		false	boolean	
name	套餐名称		false	string	
numberOfApp	允许创建应用数量		false	integer	
numberOfConcurrent	并发数/秒		false	integer	
numberOfInvocation	API调用次数/月		false	integer	
tenantTypeCode	租户类型编码		false	string	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": 0,
3   "data": true,
4   "extra": {},
5   "isError": true,
6   "isSuccess": true,
7   "msg": "",
8   "path": "",
9   "timestamp": 0
10 }
```

查询所有套餐

接口地址: /user/v2/bundles/bundle-list

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	R«List«BundleDTO»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	array	BundleDTO
ability	套餐包含功能描述,JSON格式的角色与权限	string	
code	套餐编码	string	
comment	套餐说明	string	
id	主键, 套餐id	integer	
initialize	是否为初始化套餐,1表示是初始化套餐, 0表示不是	boolean	
name	套餐名称	string	
numberOfApp	允许创建应用数量	integer	
numberOfConcurrent	并发数/秒	integer	
numberOfInvocation	API调用次数/月	integer	
tenantTypeCode	租户类型编码	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1  {
2      "code": 0,
3      "data": [
4          {
5              "ability": "",
6              "code": "",
7              "comment": "",
8              "id": 0,
9              "initialize": true,
10             "name": "",
11             "numberOfApp": 0,
12             "numberOfConcurrent": 0,
13             "numberOfInvocation": 0,
14             "tenantTypeCode": ""
```

```
15     }
16   ],
17   "extra": {},
18   "isError": true,
19   "isSuccess": true,
20   "msg": "",
21   "path": "",
22   "timestamp": 0
23 }
```

查询某租户类型的套餐列表(不包含初始化套餐)

接口地址: /user/v2/bundles/tenant-types/{tenantType}/bundle-list

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
tenantType	租户类型	path	true	string	

响应状态:

状态码	说明	schema
200	OK	R«List«BundleDTO»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	array	BundleDTO
ability	套餐包含功能描述,JSON格式的角色与权限	string	
code	套餐编码	string	
comment	套餐说明	string	
id	主键, 套餐id	integer	
initialize	是否为初始化套餐,1表示是初始化套餐, 0表示不是	boolean	
name	套餐名称	string	
numberOfApp	允许创建应用数量	integer	
numberOfConcurrent	并发数/秒	integer	
numberOfInvocation	API调用次数/月	integer	
tenantTypeCode	租户类型编码	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```

1  {
2      "code": 0,
3      "data": [
4          {
5              "ability": "",
6              "code": "",
7              "comment": "",
8              "id": 0,
9              "initialize": true,
10             "name": "",
11             "numberOfApp": 0,
12             "numberOfConcurrent": 0,
13             "numberOfInvocation": 0,
14             "tenantTypeCode": ""
15         }
16     ],

```

```
17     "extra": {},
18     "isError": true,
19     "isSuccess": true,
20     "msg": "",
21     "path": "",
22     "timestamp": 0
23 }
```

获取某套餐信息

接口地址: /user/v2/bundles/{bundleCode}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
bundleCode	租户套餐	path	true	string	

响应状态:

状态码	说明	schema
200	OK	R«BundleDTO»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	BundleDTO	BundleDTO
ability	套餐包含功能描述,JSON格式的角色与权限	string	
code	套餐编码	string	
comment	套餐说明	string	
id	主键, 套餐id	integer	
initialize	是否为初始化套餐,1表示是初始化套餐, 0表示不是	boolean	
name	套餐名称	string	
numberOfApp	允许创建应用数量	integer	
numberOfConcurrent	并发数/秒	integer	
numberOfInvocation	API调用次数/月	integer	
tenantTypeCode	租户类型编码	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```

1  {
2      "code": 0,
3      "data": {
4          "ability": "",
5          "code": "",
6          "comment": "",
7          "id": 0,
8          "initialize": true,
9          "name": "",
10         "numberOfApp": 0,
11         "numberOfConcurrent": 0,
12         "numberOfInvocation": 0,
13         "tenantTypeCode": ""
14     },
15     "extra": {},
16     "isError": true,

```

```
17     "isSuccess": true,
18     "msg": "",
19     "path": "",
20     "timestamp": 0
21 }
```

根据手机号判断账号是否存在

接口地址: /user/v2/exist/accounts/mobile/{mobile}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
mobile	手机号	path	true	string	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2     "code": 0,
3     "data": true,
4     "extra": {},
5     "isError": true,
6     "isSuccess": true,
7     "msg": "",
8     "path": "",
9     "timestamp": 0
10 }
```

根据用户名判断账号是否存在

接口地址: /user/v2/exist/accounts/name/{username}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": 0,
3   "data": true,
4   "extra": {},
5   "isError": true,
6   "isSuccess": true,
7   "msg": "",
8   "path": "",
9   "timestamp": 0
10 }
```

根据手机号判断账号是否在某租户内

接口地址: /user/v2/exist/accounts/tenants/{tenantId}/mobiles/{mobile}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
mobile	手机号	path	true	string	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1  {
2      "code": 0,
3      "data": true,
4      "extra": {},
5      "isError": true,
6      "isSuccess": true,
7      "msg": "",
8      "path": "",
9      "timestamp": 0
10 }
```

根据用户名判断账号是否在某租户内

接口地址: /user/v2/exist/accounts/{username}/tenants/{tenantId}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
tenantId	租户id	path	true	integer(int64)	
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": 0,
3   "data": true,
4   "extra": {},
5   "isError": true,
6   "isSuccess": true,
7   "msg": "",
8   "path": "",
9   "timestamp": 0
10 }
```

根据接入客户端获取应用

接口地址: /user/v2/getApplicationDTOByClientId/{clientId}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
clientId	客户端id	path	true	string	

响应状态:

状态码	说明	schema
200	OK	R«ApplicationDTO»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	ApplicationDTO	ApplicationDTO
code	应用编码,跨租户唯一	string	
id	id	integer	
name	应用名称	string	
tenantId	应用所属租户	integer	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": 0,
3   "data": {
4     "code": "",
5     "id": 0,
```

```
6         "name": "",
7         "tenantId": 0
8     },
9     "extra": {},
10    "isError": true,
11    "isSuccess": true,
12    "msg": "",
13    "path": "",
14    "timestamp": 0
15 }
```

发送短信验证码

接口地址: /user/v2/getMessage

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
phone	手机号	query	true	string	

响应状态:

状态码	说明	schema
200	OK	R«string»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1  {
2      "code": 0,
3      "data": "",
4      "extra": {},
5      "isError": true,
6      "isSuccess": true,
7      "msg": "",
8      "path": "",
9      "timestamp": 0
10 }
```

用户登录

接口地址: /user/v2/login

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1  {
2      "authenticationType": "",
3      "certificate": "",
4      "principal": "",
5      "smsKey": ""
6  }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
loginRequest	登录请求信息	body	true	LoginRequestDTO	LoginRequestDTO
authenticationType	认证类型：短信快捷认证、用户名密码认证、二维码认证等		false	string	
certificate	凭证：如密码，短信认证码等		false	string	
principal	认证主体：如用户名，手机号等		false	string	
smsKey	短信认证时需要传key		false	string	

响应状态:

状态码	说明	schema
200	OK	R«LoginInfoDTO»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	LoginInfoDTO	LoginInfoDTO
id	账号id	integer	
mobile	手机号	string	
resources	账号所属多个租户下,不同应用的资源,如: {租户A:[{应用1权限信息},{应用2权限信息}],租户B:[{},{...}]}	ResourceDTO	ResourceDTO
appRes	应用包含资源,按资源类型分包括菜单等资源信息 如 menu:JSONObject	object	
applicationCode	应用编码	string	
applicationName	应用名称	string	
tenantAuthorizationInfoMap	账号所属多个租户下的权限	AuthorizationInfoDTO	AuthorizationInfoDTO
rolePrivilegeMap	授权信息,角色-权限关系,key为角色编码,value为权限集合	object	
tenants	账号所属多个租户	array	TenantDTO
bundleCode	套餐编码	string	
id	租户id	integer	
name	租户名称	string	
tenantTypeCode	租户类型编码	string	
username	用户名	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```

1  {
2      "code": 0,
3      "data": {
4          "id": 0,
5          "mobile": "",
6          "resources": {
7              "additionalProperties1": {
8                  "appRes": {},
9                  "applicationCode": "",
10                 "applicationName": ""
11             }
12         }
13     }
14 }
```

```

12         },
13         "tenantAuthorizationInfoMap": {
14             "additionalProperties1": {
15                 "rolePrivilegeMap": {}
16             }
17         },
18         "tenants": [
19             {
20                 "bundleCode": "",
21                 "id": 0,
22                 "name": "",
23                 "tenantTypeCode": ""
24             }
25         ],
26         "username": ""
27     },
28     "extra": {},
29     "isError": true,
30     "isSuccess": true,
31     "msg": "",
32     "path": "",
33     "timestamp": 0
34 }

```

创建租户并关联账号

接口地址: /user/v2/tenantRelateAccount

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```

1  {
2      "bundleCode": "",
3      "mobile": "",
4      "name": "",
5      "password": "",
6      "tenantTypeCode": "",
7      "username": ""
8  }

```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
createTenantRequest	创建租户请求信息	body	true	CreateTenantRequestDTO	CreateTenantRequestDTO
bundleCode	初始化套餐编码，用于初始化套餐ABILITY，若为空则使用该租户类型的初始化套餐		false	string	
mobile	租户管理员登录手机号		false	string	
name	租户名称		false	string	
password	租户管理员登录密码		false	string	
tenantTypeCode	租户类型编码		false	string	
username	租户管理员登录用户名		false	string	

响应状态:

状态码	说明	schema
200	OK	R«TenantDTO»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	TenantDTO	TenantDTO
bundleCode	套餐编码	string	
id	租户id	integer	
name	租户名称	string	
tenantTypeCode	租户类型编码	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1  {
2      "code": 0,
3      "data": {
4          "bundleCode": "",
5          "id": 0,
6          "name": "",
7          "tenantTypeCode": ""
8      },
9      "extra": {},
10     "isError": true,
11     "isSuccess": true,
12     "msg": "",
13     "path": "",
14     "timestamp": 0
15 }
```

创建租户

接口地址: /user/v2/tenants

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1  {
2      "bundleCode": "",
3      "mobile": "",
4      "name": "",
5      "password": "",
6      "tenantTypeCode": "",
7      "username": ""
8  }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
createTenantRequest	创建租户请求信息	body	true	CreateTenantRequestDTO	CreateTenantRequestDTO
bundleCode	初始化套餐编码，用于初始化套餐ABILITY，若为空则使用该租户类型的初始化套餐		false	string	
mobile	租户管理员登录手机号		false	string	
name	租户名称		false	string	
password	租户管理员登录密码		false	string	
tenantTypeCode	租户类型编码		false	string	
username	租户管理员登录用户名		false	string	

响应状态:

状态码	说明	schema
200	OK	R«TenantDTO»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	TenantDTO	TenantDTO
bundleCode	套餐编码	string	
id	租户id	integer	
name	租户名称	string	
tenantTypeCode	租户类型编码	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1  {
2      "code": 0,
3      "data": {
4          "bundleCode": "",
5          "id": 0,
6          "name": "",
7          "tenantTypeCode": ""
8      },
9      "extra": {},
10     "isError": true,
11     "isSuccess": true,
12     "msg": "",
13     "path": "",
14     "timestamp": 0
15 }
```

检索租户

接口地址: /user/v2/tenants/page

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1  {
2      "name": "",
3      "tenantTypeCode": ""
4  }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
order	顺序	query	true	string	
pageNo	页码	query	true	integer(int32)	
pageSize	每页记录数	query	true	integer(int32)	
sortBy	排序字段	query	true	string	
tenantQuery	租户查询条件	body	false	TenantQueryDTO	TenantQueryDTO
name	租户名称		false	string	
tenantTypeCode	租户类型		false	string	

响应状态:

状态码	说明	schema
200	OK	R«PageVO«TenantDTO»»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	PageVO«TenantDTO»	PageVO«TenantDTO»
counts		integer	
first		boolean	
items		array	TenantDTO
bundleCode	套餐编码	string	
id	租户id	integer	
name	租户名称	string	
tenantTypeCode	租户类型编码	string	
itemsSize		integer	
page		integer	
pageSize		integer	
pages		integer	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```

1  {
2      "code": 0,
3      "data": {
4          "counts": 0,
5          "first": true,
6          "items": [
7              {
8                  "bundleCode": "",
9                  "id": 0,
10                 "name": "",
11                 "tenantTypeCode": ""
12             }
13         ],
14         "itemsSize": 0,

```

```
15         "page": 0,
16         "pageSize": 0,
17         "pages": 0
18     },
19     "extra": {},
20     "isError": true,
21     "isSuccess": true,
22     "msg": "",
23     "path": "",
24     "timestamp": 0
25 }
```

获取租户信息

接口地址: /user/v2/tenants/{id}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
id	id	path	true	integer(int64)	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«TenantDTO»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	TenantDTO	TenantDTO
bundleCode	套餐编码	string	
id	租户id	integer	
name	租户名称	string	
tenantTypeCode	租户类型编码	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```

1  {
2      "code": 0,
3      "data": {
4          "bundleCode": "",
5          "id": 0,
6          "name": "",
7          "tenantTypeCode": ""
8      },
9      "extra": {},
10     "isError": true,
11     "isSuccess": true,
12     "msg": "",
13     "path": "",
14     "timestamp": 0
15 }

```

初始化租户套餐

接口地址: /user/v2/tenants/{tenantId}/bundles/{bundleCode}

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
bundleCode	套餐编码	path	true	string	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

1	{
2	"code": 0,
3	"data": true,
4	"extra": {},
5	"isError": true,
6	"isSuccess": true,
7	"msg": "",
8	"path": "",
9	"timestamp": 0
10	}

切换租户套餐

接口地址: /user/v2/tenants/{tenantId}/bundles/{bundleCode}

请求方式: PUT

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
bundleCode	套餐编码	path	true	string	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2     "code": 0,
3     "data": true,
4     "extra": {},
5     "isError": true,
6     "isSuccess": true,
7     "msg": "",
8     "path": "",
9     "timestamp": 0
10 }
```

将某账号从租户内移除，租户管理员不可移除

接口地址: /user/v2/unbind/accounts/{username}/tenants/{tenantId}

请求方式: DELETE

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
tenantId	租户id	path	true	integer(int64)	
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
204	No Content	
401	Unauthorized	
403	Forbidden	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": 0,
3   "data": true,
4   "extra": {},
5   "isError": true,
6   "isSuccess": true,
7   "msg": "",
8   "path": "",
9   "timestamp": 0
10 }
```

统一账号-角色权限

绑定角色

接口地址: /user/bind/tenants/{tenantId}/accounts/{username}/roles

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
roleCodes	角色编码	query	true	array	string
tenantId	租户id	path	true	integer(int64)	
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

1

获取权限组下所有权限

接口地址: /user/privilege-groups/{privilegeGroupId}/privilege-list

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
privilegeGroupId	权限组的id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	PrivilegeDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	权限编码	string	
id	权限id	integer(int64)	integer(int64)
name	权限名称	string	
privilegeGroupId	所属权限组id	integer(int64)	integer(int64)

响应示例:

```
1  [
2    {
3      "code": "",
4      "id": 0,
5      "name": "",
6      "privilegeGroupId": 0
7    }
8  ]
```

查根据roleCode获取角色(不包含权限)

接口地址: /user/roles/tenants/{tenantId}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
roleCodes	角色编码	query	true	array	string
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	RoleDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	角色编码	string	
id	角色id	integer(int64)	integer(int64)
name	角色名称	string	
privilegeCodes	角色包含权限列表	array	
tenantId	角色所属租户	integer(int64)	integer(int64)

响应示例:

```
1  [
2    {
3      "code": "",
4      "id": 0,
5      "name": "",
6      "privilegeCodes": [],
7      "tenantId": 0
8    }
9  ]
```

修改租户内角色(不包含权限)

接口地址: /user/tenants/roles

请求方式: PUT

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1  {
2    "code": "",
3    "id": 0,
4    "name": "",
5    "privilegeCodes": [],
6    "tenantId": 0
7  }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
role	角色信息	body	true	RoleDTO	RoleDTO
code	角色编码		false	string	
id	角色id		false	integer	
name	角色名称		false	string	
privilegeCodes	角色包含权限列表		false	array	string
tenantId	角色所属租户		false	integer	

响应状态:

状态码	说明	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

1

查询某租户下角色(不分页，不包含权限)

接口地址: `/user/tenants/{tenantId}/roles`

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	RoleDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	角色编码	string	
id	角色id	integer(int64)	integer(int64)
name	角色名称	string	
privilegeCodes	角色包含权限列表	array	
tenantId	角色所属租户	integer(int64)	integer(int64)

响应示例:

```
1  [  
2    {  
3      "code": "",  
4      "id": 0,  
5      "name": "",  
6      "privilegeCodes": [],  
7      "tenantId": 0  
8    }  
9  ]
```

创建租户内角色（不包含权限）

接口地址: /user/tenants/{tenantId}/roles

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1  {  
2    "code": "",  
3    "id": 0,  
4    "name": "",  
5    "privilegeCodes": [],  
6    "tenantId": 0  
7  }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
role	角色信息	body	true	RoleDTO	RoleDTO
code	角色编码		false	string	
id	角色id		false	integer	
name	角色名称		false	string	
privilegeCodes	角色包含权限列表		false	array	string
tenantId	角色所属租户		false	integer	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

1

查找某租户下，多个角色的权限信息集合

接口地址: /user/tenants/{tenantId}/roles/privilege-list

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
roleCodes	多个角色编码	query	true	array	string
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	PrivilegeDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	权限编码	string	
id	权限id	integer(int64)	integer(int64)
name	权限名称	string	
privilegeGroupId	所属权限组id	integer(int64)	integer(int64)

响应示例:

```
1  [
2    {
3      "code": "",
4      "id": 0,
5      "name": "",
6      "privilegeGroupId": 0
7    }
8  ]
```

查找某租户下，多个角色的权限信息集合，并根据权限组组装成为权限树

接口地址: /user/tenants/{tenantId}/roles/role-privilege-list

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
roleCodes	多个角色的编码	query	true	array	string
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	PrivilegeTreeDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
children	子节点集合	array	PrivilegeTreeDTO
children	子节点集合	array	PrivilegeTreeDTO
group		boolean	
id	节点唯一标识, 权限组使用id, 权限使用code, 约束code不能为纯数值保证不重复。	string	
name	节点名称	string	
parentId	父节点标识	string	
sort	排序	integer	
status	状态	integer	
group		boolean	
id	节点唯一标识, 权限组使用id, 权限使用code, 约束code不能为纯数值保证不重复。	string	
name	节点名称	string	
parentId	父节点标识	string	
sort	排序	integer(int32)	integer(int32)
status	状态	integer(int32)	integer(int32)

响应示例:

1	{
2	"children": [

```
3      {
4          "children": [],
5          "group": true,
6          "id": "",
7          "name": "",
8          "parentId": "",
9          "sort": 0,
10         "status": 0
11     }
12 ],
13 "group": true,
14 "id": "",
15 "name": "",
16 "parentId": "",
17 "sort": 0,
18 "status": 0
19 }
```

根据角色编码删除租户内角色，如果有账号绑定该角色，禁止删除

接口地址: /user/tenants/{tenantId}/roles/{roleCode}

请求方式: DELETE

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
roleCode	角色编码	path	true	string	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	
204	No Content	
401	Unauthorized	
403	Forbidden	

响应参数:

暂无

响应示例:

角色设置权限(清除+设置)

接口地址: /user/tenants/{tenantId}/roles/{roleCode}/privileges

请求方式: PUT

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
privilegeCodes	权限编码	query	true	array	string
roleCode	角色编码	path	true	string	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

获取租户内的某个角色信息(包含权限信息)

接口地址: /user/tenants/{tenantId}/roles/{roleCode}/role-privilege

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
roleCode	roleCode	path	true	string	
roleCodes	角色编码	path	true	string	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	RoleDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	角色编码	string	
id	角色id	integer(int64)	integer(int64)
name	角色名称	string	
privilegeCodes	角色包含权限列表	array	
tenantId	角色所属租户	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": "",
3   "id": 0,
4   "name": "",
5   "privilegeCodes": [],
6   "tenantId": 0
7 }
```

授权，获取某用户在多个租户下的权限信息

接口地址: /user/tenants/{username}/privileges

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
tenantIds	多个租户的id	query	true	array	integer
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

1

解绑角色

接口地址: /user/unbind/tenants/{tenantId}/accounts/{username}/roles

请求方式: PUT

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
roleCodes	角色编码	query	true	array	string
tenantId	租户id	path	true	integer(int64)	
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

1

统一账号-角色权限V2.0

绑定角色

接口地址: /user/v2/bind/tenants/{tenantId}/accounts/{username}/roles

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
roleCodes	角色编码	query	true	array	string
tenantId	租户id	path	true	integer(int64)	
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": 0,
3   "data": true,
4   "extra": {},
5   "isError": true,
6   "isSuccess": true,
7   "msg": "",
8   "path": "",
9   "timestamp": 0
10 }
```

获取权限组下所有权限

接口地址: /user/v2/privilege-groups/{privilegeGroupId}/privilege-list

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
privilegeGroupId	权限组的id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«List«PrivilegeDTO»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	array	PrivilegeDTO
code	权限编码	string	
id	权限id	integer	
name	权限名称	string	
privilegeGroupId	所属权限组id	integer	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

1	{
2	"code": 0,
3	"data": [
4	{
5	"code": "",
6	"id": 0,
7	"name": "",
8	"privilegeGroupId": 0
9	}
10],
11	"extra": {},
12	"isError": true,
13	"isSuccess": true,
14	"msg": "",
15	"path": "",
16	"timestamp": 0
17	}

查根据roleCode获取角色(不包含权限)

接口地址: /user/v2/roles/tenants/{tenantId}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
roleCodes	角色编码	query	true	array	string
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«List«RoleDTO»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	array	RoleDTO
code	角色编码	string	
id	角色id	integer	
name	角色名称	string	
privilegeCodes	角色包含权限列表	array	string
tenantId	角色所属租户	integer	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1  {
2      "code": 0,
3      "data": [
4          {
5              "code": "",
6              "id": 0,
7              "name": "",
8              "privilegeCodes": [],
9              "tenantId": 0
10         }
11     ],
12     "extra": {},
13     "isError": true,
14     "isSuccess": true,
15     "msg": "",
16     "path": "",
17     "timestamp": 0
18 }
```

修改租户内角色(不包含权限)

接口地址: /user/v2/tenants/roles

请求方式: PUT

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1  {
2      "code": "",
3      "id": 0,
4      "name": "",
5      "privilegeCodes": [],
6      "tenantId": 0
7  }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
role	角色信息	body	true	RoleDTO	RoleDTO
code	角色编码		false	string	
id	角色id		false	integer	
name	角色名称		false	string	
privilegeCodes	角色包含权限列表		false	array	string
tenantId	角色所属租户		false	integer	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": 0,
3   "data": true,
4   "extra": {},
5   "isError": true,
6   "isSuccess": true,
7   "msg": "",
8   "path": "",
9   "timestamp": 0
10 }
```

查询某租户下角色(不分页，不包含权限)

接口地址: /user/v2/tenants/{tenantId}/roles

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«List«RoleDTO»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	array	RoleDTO
code	角色编码	string	
id	角色id	integer	
name	角色名称	string	
privilegeCodes	角色包含权限列表	array	string
tenantId	角色所属租户	integer	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

1	{
2	"code": 0,
3	"data": [
4	{
5	"code": "",
6	"id": 0,
7	"name": "",
8	"privilegeCodes": [],
9	"tenantId": 0
10	}
11],
12	"extra": {},
13	"isError": true,
14	"isSuccess": true,
15	"msg": "",
16	"path": "",
17	"timestamp": 0

创建租户内角色（不包含权限）

接口地址: `/user/v2/tenants/{tenantId}/roles`

请求方式: `POST`

请求数据类型: `application/json`

响应数据类型: `*/*`

接口描述:

请求示例:

```
1 {  
2   "code": "",  
3   "id": 0,  
4   "name": "",  
5   "privilegeCodes": [],  
6   "tenantId": 0  
7 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
role	角色信息	body	true	RoleDTO	RoleDTO
code	角色编码		false	string	
id	角色id		false	integer	
name	角色名称		false	string	
privilegeCodes	角色包含权限列表		false	array	string
tenantId	角色所属租户		false	integer	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": 0,
3   "data": true,
4   "extra": {},
5   "isError": true,
6   "isSuccess": true,
7   "msg": "",
8   "path": "",
9   "timestamp": 0
10 }
```

查找某租户下，多个角色的权限信息集合

接口地址: /user/v2/tenants/{tenantId}/roles/privilege-list

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
roleCodes	多个角色编码	query	true	array	string
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«List«PrivilegeDTO»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	array	PrivilegeDTO
code	权限编码	string	
id	权限id	integer	
name	权限名称	string	
privilegeGroupId	所属权限组id	integer	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

1	{
2	"code": 0,
3	"data": [
4	{
5	"code": "",
6	"id": 0,
7	"name": "",
8	"privilegeGroupId": 0
9	}
10],
11	"extra": {},
12	"isError": true,
13	"isSuccess": true,
14	"msg": "",
15	"path": "",
16	"timestamp": 0
17	}

查找某租户下，多个角色的权限信息集合，并根据权限组组装成为权限树

接口地址: /user/v2/tenants/{tenantId}/roles/role-privilege-list

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
roleCodes	多个角色的编码	query	true	array	string
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«PrivilegeTreeDTO»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	PrivilegeTreeDTO	PrivilegeTreeDTO
children	子节点集合	array	PrivilegeTreeDTO
group		boolean	
id	节点唯一标识，权限组使用id，权限使用code，约束code不能为纯数值保证不重复。	string	
name	节点名称	string	
parentId	父节点标识	string	
sort	排序	integer	
status	状态	integer	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示信息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1  {
2      "code": 0,
3      "data": {
4          "children": [],
5          "group": true,
6          "id": "",
7          "name": "",
8          "parentId": "",
9          "sort": 0,
10         "status": 0
11     },
12     "extra": {},
13     "isError": true,
14     "isSuccess": true,
15     "msg": "",
16     "path": "",
17     "timestamp": 0
18 }
```

根据角色编码删除租户内角色，如果有账号绑定该角色，禁止删除

接口地址: /user/v2/tenants/{tenantId}/roles/{roleCode}

请求方式: DELETE

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
roleCode	角色编码	path	true	string	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
204	No Content	
401	Unauthorized	
403	Forbidden	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1  {
2      "code": 0,
3      "data": true,
4      "extra": {},
5      "isError": true,
6      "isSuccess": true,
7      "msg": "",
8      "path": "",
9      "timestamp": 0
10 }
```

角色设置权限(清除+设置)

接口地址: /user/v2/tenants/{tenantId}/roles/{roleCode}/privileges

请求方式: PUT

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
privilegeCodes	权限编码	query	true	array	string
roleCode	角色编码	path	true	string	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1  {
2      "code": 0,
3      "data": true,
4      "extra": {},
5      "isError": true,
6      "isSuccess": true,
7      "msg": "",
8      "path": "",
9      "timestamp": 0
10 }
```

获取租户内的某个角色信息(包含权限信息)

接口地址: /user/v2/tenants/{tenantId}/roles/{roleCode}/role-privilege

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
roleCode	roleCode	path	true	string	
roleCodes	角色编码	path	true	string	
tenantId	租户id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«RoleDTO»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	RoleDTO	RoleDTO
code	角色编码	string	
id	角色id	integer	
name	角色名称	string	
privilegeCodes	角色包含权限列表	array	string
tenantId	角色所属租户	integer	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1  {
2      "code": 0,
3      "data": {
4          "code": "",
5          "id": 0,
6          "name": "",
7          "privilegeCodes": [],
8          "tenantId": 0
9      },
10     "extra": {},
11     "isError": true,
12     "isSuccess": true,
13     "msg": "",
14     "path": "",
15     "timestamp": 0
16 }
```

授权，获取某用户在多个租户下的权限信息

接口地址: /user/v2/tenants/{username}/privileges

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
tenantIds	多个租户的id	query	true	array	integer
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	R«Map«long,AuthorizationInfoDTO»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	AuthorizationInfoDTO	AuthorizationInfoDTO
rolePrivilegeMap	授权信息, 角色-权限关系,key为角色编码, value为权限集合	object	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1  {
2      "code": 0,
3      "data": {
4          "additionalProperties1": {
5              "rolePrivilegeMap": {}
6          }
7      },
8      "extra": {},
9      "isError": true,
10     "isSuccess": true,
11     "msg": "",
12     "path": "",
13     "timestamp": 0
14 }
```

解绑角色

接口地址: `/user/v2/unbind/tenants/{tenantId}/accounts/{username}/roles`

请求方式: PUT

请求数据类型: `application/json`

响应数据类型: `*/*`

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
roleCodes	角色编码	query	true	array	string
tenantId	租户id	path	true	integer(int64)	
username	用户名	path	true	string	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": 0,
3   "data": true,
4   "extra": {},
5   "isError": true,
6   "isSuccess": true,
7   "msg": "",
8   "path": "",
9   "timestamp": 0
10 }
```

统一账号-资源相关

创建应用

接口地址: /user/apps

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1 {
2   "code": "",
3   "id": 0,
4   "name": "",
5   "tenantId": 0
6 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
application	应用信息	body	true	ApplicationDTO	ApplicationDTO
code	应用编码,跨租户唯一		false	string	
id	id		false	integer	
name	应用名称		false	string	
tenantId	应用所属租户		false	integer	

响应状态:

状态码	说明	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

```
1 |
```

修改应用

接口地址: /user/apps

请求方式: PUT

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1 {
2   "code": "",
3   "id": 0,
4   "name": "",
5   "tenantId": 0
6 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
application	应用信息	body	true	ApplicationDTO	ApplicationDTO
code	应用编码,跨租户唯一		false	string	
id	id		false	integer	
name	应用名称		false	string	
tenantId	应用所属租户		false	integer	

响应状态:

状态码	说明	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

```
1 |
```

分页条件查找应用

接口地址: /user/apps/page

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1 {
2   "name": ""
3 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
pageNo	页码	query	true	integer(int32)	
pageSize	每页记录数	query	true	integer(int32)	
query	应用查询参数	body	true	ApplicationQueryParams	ApplicationQueryParams
name	应用名称(模糊匹配)		false	string	

响应状态:

状态码	说明	schema
200	OK	PageVO«ApplicationDTO»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
counts		integer(int64)	integer(int64)
first		boolean	
items		array	ApplicationDTO
code	应用编码,跨租户唯一	string	
id	id	integer	
name	应用名称	string	
tenantId	应用所属租户	integer	
itemsSize		integer(int32)	integer(int32)
page		integer(int32)	integer(int32)
pageSize		integer(int32)	integer(int32)
pages		integer(int32)	integer(int32)

响应示例:

```
1  {
2      "counts": 0,
3      "first": true,
4      "items": [
5          {
6              "code": "",
7              "id": 0,
8              "name": "",
9              "tenantId": 0
10         }
11     ],
12     "itemsSize": 0,
13     "page": 0,
14     "pageSize": 0,
15     "pages": 0
16 }
```

根据权限加载指定应用的资源

接口地址: /user/apps/privileges/resources/{applicationCode}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
applicationCode	应用的编码	path	true	string	
privilegeCodes	权限集合	query	true	array	string

响应状态:

状态码	说明	schema
200	OK	ResourceDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
appRes	应用包含资源，按资源类型分包括菜单等资源信息 如 menu:JSONObject	object	
applicationCode	应用编码	string	
applicationName	应用名称	string	

响应示例:

1	[
2	{
3	"appRes": {},
4	"applicationCode": "",
5	"applicationName": ""
6	}
7]

根据应用编码查找应用

接口地址: /user/apps/{applicationCode}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
applicationCode	应用信息	path	true	string	

响应状态:

状态码	说明	schema
200	OK	ApplicationDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	应用编码,跨租户唯一	string	
id	id	integer(int64)	integer(int64)
name	应用名称	string	
tenantId	应用所属租户	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": "",
3   "id": 0,
4   "name": "",
5   "tenantId": 0
6 }
```

删除应用

接口地址: /user/apps/{applicationCode}

请求方式: DELETE

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
application	应用信息	path	true	string	
applicationCode	applicationCode	path	true	string	

响应状态:

状态码	说明	schema
200	OK	
204	No Content	
401	Unauthorized	
403	Forbidden	

响应参数:

暂无

响应示例:

1	
---	--

根据条件查询菜单列表

接口地址: /user/menus/page

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

1	{
2	"applicationCode": "",
3	"status": 0,
4	"title": ""
5	}

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
pageNo	页码	query	true	integer(int32)	
pageSize	每页记录数	query	true	integer(int32)	
params	菜单查询参数	body	true	MenuQueryDTO	MenuQueryDTO
applicationCode	所属应用		false	string	
status	状态		false	integer	
title	菜单标题		false	string	

响应状态:

状态码	说明	schema
200	OK	PageVO«MenuDTO»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
counts		integer(int64)	integer(int64)
first		boolean	
items		array	MenuDTO
application	所属应用	string	
applicationCode	所属应用编码	string	
code	页面标识	string	
comment	说明	string	
icon	菜单图标	string	
id	菜单id	integer	
parentId	菜单父级id	integer	
privilegeCode	绑定的权限编码	string	
sort	排序	integer	
status	状态	integer	
title	菜单标题	string	
url	菜单跳转url	string	
itemsSize		integer(int32)	integer(int32)
page		integer(int32)	integer(int32)
pageSize		integer(int32)	integer(int32)
pages		integer(int32)	integer(int32)

响应示例:

```
1  {
2      "counts": 0,
3      "first": true,
4      "items": [
5          {
6              "application": "",
7              "applicationCode": "",
8              "code": "",
9              "comment": "",
10             "icon": "",
11             "id": 0,
12             "parentId": 0,
13             "privilegeCode": "",
14             "sort": 0,
15             "status": 0,
16             "title": "",
17             "url": ""
18         }
19     ]
20 }
```

```
19     ],
20     "itemSize": 0,
21     "page": 0,
22     "pageSize": 0,
23     "pages": 0
24 }
```

根据权限编码列表获取菜单

接口地址: /user/menus/privileges

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
privileges	权限编码	query	true	array	string

响应状态:

状态码	说明	schema
200	OK	MenuDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
application	所属应用	string	
applicationCode	所属应用编码	string	
code	页面标识	string	
comment	说明	string	
icon	菜单图标	string	
id	菜单id	integer(int64)	integer(int64)
parentId	菜单父级id	integer(int64)	integer(int64)
privilegeCode	绑定的权限编码	string	
sort	排序	integer(int32)	integer(int32)
status	状态	integer(int32)	integer(int32)
title	菜单标题	string	
url	菜单跳转url	string	

响应示例:

```
1  [
2      {
3          "application": "",
4          "applicationCode": "",
5          "code": "",
6          "comment": "",
7          "icon": "",
8          "id": 0,
9          "parentId": 0,
10         "privilegeCode": "",
11         "sort": 0,
12         "status": 0,
13         "title": "",
14         "url": ""
15     }
16 ]
```

根据应用编码查询菜单列表

接口地址: /user/menus/{applicationCode}/menu-list

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
applicationCode	应用编码	path	true	string	

响应状态:

状态码	说明	schema
200	OK	MenuDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
application	所属应用	string	
applicationCode	所属应用编码	string	
code	页面标识	string	
comment	说明	string	
icon	菜单图标	string	
id	菜单id	integer(int64)	integer(int64)
parentId	菜单父级id	integer(int64)	integer(int64)
privilegeCode	绑定的权限编码	string	
sort	排序	integer(int32)	integer(int32)
status	状态	integer(int32)	integer(int32)
title	菜单标题	string	
url	菜单跳转url	string	

响应示例:

```
1  [
2    {
3      "application": "",
4      "applicationCode": "",
5      "code": "",
6      "comment": "",
7      "icon": "",
8      "id": 0,
9      "parentId": 0,
10     "privilegeCode": "",
11     "sort": 0,
12     "status": 0,
```



```
13         "title": "",
14         "url": ""
15     }
16 ]
```

根据ID查询菜单

接口地址: /user/menus/{id}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
id	菜单Id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	MenuDTO
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
application	所属应用	string	
applicationCode	所属应用编码	string	
code	页面标识	string	
comment	说明	string	
icon	菜单图标	string	
id	菜单id	integer(int64)	integer(int64)
parentId	菜单父级id	integer(int64)	integer(int64)
privilegeCode	绑定的权限编码	string	
sort	排序	integer(int32)	integer(int32)
status	状态	integer(int32)	integer(int32)
title	菜单标题	string	
url	菜单跳转url	string	

响应示例:

```
1 {
2   "application": "",
3   "applicationCode": "",
4   "code": "",
5   "comment": "",
6   "icon": "",
7   "id": 0,
8   "parentId": 0,
9   "privilegeCode": "",
10  "sort": 0,
11  "status": 0,
12  "title": "",
13  "url": ""
14 }
```

获取多个租户下权限所对应的的资源信息，并按应用拆分

接口地址: /user/tenants/apps/privileges/resources

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
tenantAuthorizationInfoMap	多个租户下的资源信息	body	true	Map<Long, AuthorizationInfoDTO>	Map<Long, AuthorizationInfoDTO>

响应状态:

状态码	说明	schema
200	OK	
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

暂无

响应示例:

1	
---	--

统一账号-资源相关V2.0

创建应用

接口地址: /user/v2/apps

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

1	{
2	"code": "",
3	"id": 0,
4	"name": "",
5	"tenantId": 0
6	}

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
application	应用信息	body	true	ApplicationDTO	ApplicationDTO
code	应用编码,跨租户唯一		false	string	
id	id		false	integer	
name	应用名称		false	string	
tenantId	应用所属租户		false	integer	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2     "code": 0,
3     "data": true,
4     "extra": {},
5     "isError": true,
6     "isSuccess": true,
7     "msg": "",
8     "path": "",
9     "timestamp": 0
10 }
```

修改应用

接口地址: /user/v2/apps

请求方式: PUT

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1 {
2     "code": "",
3     "id": 0,
4     "name": "",
5     "tenantId": 0
6 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
application	应用信息	body	true	ApplicationDTO	ApplicationDTO
code	应用编码,跨租户唯一		false	string	
id	id		false	integer	
name	应用名称		false	string	
tenantId	应用所属租户		false	integer	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": 0,
3   "data": true,
4   "extra": {},
5   "isError": true,
6   "isSuccess": true,
7   "msg": "",
8   "path": "",
9   "timestamp": 0
10 }
```

分页条件查找应用

接口地址: /user/v2/apps/page

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1 | {
2 |     "name": ""
3 | }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
pageNo	页码	query	true	integer(int32)	
pageSize	每页记录数	query	true	integer(int32)	
query	应用查询参数	body	true	ApplicationQueryParams	ApplicationQueryParams
name	应用名称(模糊匹配)		false	string	

响应状态:

状态码	说明	schema
200	OK	R«PageVO«ApplicationDTO»»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	PageVO«ApplicationDTO»	PageVO«ApplicationDTO»
counts		integer	
first		boolean	
items		array	ApplicationDTO
code	应用编码,跨租户唯一	string	
id	id	integer	
name	应用名称	string	
tenantId	应用所属租户	integer	
itemsSize		integer	
page		integer	
pageSize		integer	
pages		integer	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```

1  {
2      "code": 0,
3      "data": {
4          "counts": 0,
5          "first": true,
6          "items": [
7              {
8                  "code": "",
9                  "id": 0,
10                 "name": "",
11                 "tenantId": 0
12             }
13         ],

```



```
14         "itemsSize": 0,
15         "page": 0,
16         "pageSize": 0,
17         "pages": 0
18     },
19     "extra": {},
20     "isError": true,
21     "isSuccess": true,
22     "msg": "",
23     "path": "",
24     "timestamp": 0
25 }
```

根据权限加载指定应用的资源

接口地址: /user/v2/apps/privileges/resources/{applicationCode}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
applicationCode	应用的编码	path	true	string	
privilegeCodes	权限集合	query	true	array	string

响应状态:

状态码	说明	schema
200	OK	R«List«ResourceDTO»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	array	ResourceDTO
appRes	应用包含资源，按资源类型分包括菜单等资源信息 如 menu:JSONObject	object	
applicationCode	应用编码	string	
applicationName	应用名称	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```

1  {
2      "code": 0,
3      "data": [
4          {
5              "appRes": {},
6              "applicationCode": "",
7              "applicationName": ""
8          }
9      ],
10     "extra": {},
11     "isError": true,
12     "isSuccess": true,
13     "msg": "",
14     "path": "",
15     "timestamp": 0
16 }

```

根据应用编码查找应用

接口地址: /user/v2/apps/{applicationCode}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
applicationCode	应用信息	path	true	string	

响应状态:

状态码	说明	schema
200	OK	R«ApplicationDTO»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	ApplicationDTO	ApplicationDTO
code	应用编码,跨租户唯一	string	
id	id	integer	
name	应用名称	string	
tenantId	应用所属租户	integer	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": 0,
3   "data": {
4     "code": "",
5     "id": 0,
6     "name": "",
7     "tenantId": 0
8   },
9   "extra": {},
10  "isError": true,
11  "isSuccess": true,
12  "msg": "",
```

```
13     "path": "",
14     "timestamp": 0
15 }
```

删除应用

接口地址: /user/v2/apps/{applicationCode}

请求方式: DELETE

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
application	应用信息	path	true	string	
applicationCode	applicationCode	path	true	string	

响应状态:

状态码	说明	schema
200	OK	R«boolean»
204	No Content	
401	Unauthorized	
403	Forbidden	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	boolean	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2     "code": 0,
3     "data": true,
4     "extra": {},
5     "isError": true,
6     "isSuccess": true,
7     "msg": "",
8     "path": "",
9     "timestamp": 0
10 }
```

根据条件查询菜单列表

接口地址: /user/v2/menus/page

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求示例:

```
1 {
2     "applicationCode": "",
3     "status": 0,
4     "title": ""
5 }
```

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
pageNo	页码	query	true	integer(int32)	
pageSize	每页记录数	query	true	integer(int32)	
params	菜单查询参数	body	true	MenuQueryDTO	MenuQueryDTO
applicationCode	所属应用		false	string	
status	状态		false	integer	
title	菜单标题		false	string	

响应状态:

状态码	说明	schema
200	OK	R«PageVO«MenuDTO»»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	PageVO«MenuDTO»	PageVO«MenuDTO»
counts		integer	
first		boolean	
items		array	MenuDTO
application	所属应用	string	
applicationCode	所属应用编码	string	
code	页面标识	string	
comment	说明	string	
icon	菜单图标	string	
id	菜单id	integer	
parentId	菜单父级id	integer	
privilegeCode	绑定的权限编码	string	
sort	排序	integer	
status	状态	integer	
title	菜单标题	string	
url	菜单跳转url	string	
itemsSize		integer	
page		integer	
pageSize		integer	
pages		integer	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1  {
2      "code": 0,
3      "data": {
4          "counts": 0,
5          "first": true,
6          "items": [
7              {
8                  "application": "",
9                  "applicationCode": "",
10                 "code": "",
11                 "comment": "",
12                 "icon": "",
13                 "id": 0,
14                 "parentId": 0,
15                 "privilegeCode": "",
16                 "sort": 0,
17                 "status": 0,
18                 "title": "",
19                 "url": ""
20             }
21         ],
22         "itemsSize": 0,
23         "page": 0,
24         "pageSize": 0,
25         "pages": 0
26     },
27     "extra": {},
28     "isError": true,
29     "isSuccess": true,
30     "msg": "",
31     "path": "",
32     "timestamp": 0
33 }
```

根据权限编码列表获取菜单

接口地址: /user/v2/menus/privileges

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
privileges	权限编码	query	true	array	string

响应状态:

状态码	说明	schema
200	OK	R«List«MenuDTO»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	array	MenuDTO
application	所属应用	string	
applicationCode	所属应用编码	string	
code	页面标识	string	
comment	说明	string	
icon	菜单图标	string	
id	菜单id	integer	
parentId	菜单父级id	integer	
privilegeCode	绑定的权限编码	string	
sort	排序	integer	
status	状态	integer	
title	菜单标题	string	
url	菜单跳转url	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1 {
2   "code": 0,
3   "data": [
4     {
```

```
5         "application": "",
6         "applicationCode": "",
7         "code": "",
8         "comment": "",
9         "icon": "",
10        "id": 0,
11        "parentId": 0,
12        "privilegeCode": "",
13        "sort": 0,
14        "status": 0,
15        "title": "",
16        "url": ""
17    }
18    ],
19    "extra": {},
20    "isError": true,
21    "isSuccess": true,
22    "msg": "",
23    "path": "",
24    "timestamp": 0
25 }
```

根据应用编码查询菜单列表

接口地址: /user/v2/menus/{applicationCode}/menu-list

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
applicationCode	应用编码	path	true	string	

响应状态:

状态码	说明	schema
200	OK	R«List«MenuDTO»»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	array	MenuDTO
application	所属应用	string	
applicationCode	所属应用编码	string	
code	页面标识	string	
comment	说明	string	
icon	菜单图标	string	
id	菜单id	integer	
parentId	菜单父级id	integer	
privilegeCode	绑定的权限编码	string	
sort	排序	integer	
status	状态	integer	
title	菜单标题	string	
url	菜单跳转url	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```

1  {
2      "code": 0,
3      "data": [
4          {
5              "application": "",
6              "applicationCode": "",
7              "code": "",
8              "comment": "",
9              "icon": "",
10             "id": 0,
11             "parentId": 0,
12             "privilegeCode": "",
13             "sort": 0,
14             "status": 0,
15             "title": "",
16             "url": ""

```

```
17     }
18     ],
19     "extra": {},
20     "isError": true,
21     "isSuccess": true,
22     "msg": "",
23     "path": "",
24     "timestamp": 0
25 }
```

根据ID查询菜单

接口地址: /user/v2/menus/{id}

请求方式: GET

请求数据类型: *

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
id	菜单Id	path	true	integer(int64)	

响应状态:

状态码	说明	schema
200	OK	R«MenuDTO»
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	MenuDTO	MenuDTO
application	所属应用	string	
applicationCode	所属应用编码	string	
code	页面标识	string	
comment	说明	string	
icon	菜单图标	string	
id	菜单id	integer	
parentId	菜单父级id	integer	
privilegeCode	绑定的权限编码	string	
sort	排序	integer	
status	状态	integer	
title	菜单标题	string	
url	菜单跳转url	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```
1  {
2      "code": 0,
3      "data": {
4          "application": "",
5          "applicationCode": "",
6          "code": "",
7          "comment": "",
8          "icon": "",
9          "id": 0,
10         "parentId": 0,
11         "privilegeCode": "",
12         "sort": 0,
13         "status": 0,
14         "title": "",
15         "url": ""
16     },
```

```
17     "extra": {},
18     "isError": true,
19     "isSuccess": true,
20     "msg": "",
21     "path": "",
22     "timestamp": 0
23 }
```

获取多个租户下权限所对应的的资源信息，并按应用拆分

接口地址: /user/v2/tenants/apps/privileges/resources

请求方式: POST

请求数据类型: application/json

响应数据类型: */*

接口描述:

请求参数:

参数名称	参数说明	in	是否必须	数据类型	schema
tenantAuthorizationInfoMap	多个租户下的资源信息	body	true	Map<Long, AuthorizationInfoDTO>	Map<Long, AuthorizationInfoDTO>

响应状态:

状态码	说明	schema
200	OK	R«Map«long,List«ResourceDTO»»»
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
code	响应编码:0/200-请求处理成功	integer(int32)	integer(int32)
data	响应数据	ResourceDTO	ResourceDTO
appRes	应用包含资源，按资源类型分包括菜单等资源信息 如 menu:JSONObject	object	
applicationCode	应用编码	string	
applicationName	应用名称	string	
extra	附加数据	object	
isError		boolean	
isSuccess		boolean	
msg	提示消息	string	
path	请求路径	string	
timestamp	响应时间戳	integer(int64)	integer(int64)

响应示例:

```

1  {
2      "code": 0,
3      "data": {
4          "additionalProperties1": {
5              "appRes": {},
6              "applicationCode": "",
7              "applicationName": ""
8          }
9      },
10     "extra": {},
11     "isError": true,
12     "isSuccess": true,
13     "msg": "",
14     "path": "",
15     "timestamp": 0
16 }

```