中国移动通信企业文档

**福建移动智慧工厂**

Document No:

**单点登录接入规范**

2019年5月

文档修订记录

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 版本编号 | \*变化  状态 | 简要说明（变更内容和变更范围） | 日期 | 变更人 | 批准日期 | 批准人 |
| V1.0 | 建立 | 新增智慧工厂平台的单点登录接入规范 | 2019/5/9 | 朱英弘 |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

\*变化状态：建立，修改，增加，删除

**目 录**

[1. 前言 4](#_Toc8634504)

[2. 接口概况 5](#_Toc8634505)

[2.1. 智慧工厂平台单点登录接口简介 5](#_Toc8634506)

[2.2. 接口协议统一原则 5](#_Toc8634507)

[2.3. 接口数据格式统一原则 5](#_Toc8634508)

[3. 应用环境说明 6](#_Toc8634509)

[3.1. 测试环境 6](#_Toc8634510)

[3.2. 正式环境 6](#_Toc8634511)

[4. 接口调用方式介绍 7](#_Toc8634512)

[4.1. Http调用方式详解 7](#_Toc8634513)

[5. 服务报文规范 9](#_Toc8634514)

[5.1消息的约束 9](#_Toc8634515)

[5.2消息请求结构 9](#_Toc8634516)

[6. 接口说明 16](#_Toc8634517)

[6.1. 单点认证接口 16](#_Toc8634518)

[6.2. 所有租户用户关系列表获取 18](#_Toc8634519)

[6.3. 单个用户信息列表获取 22](#_Toc8634520)

[7. 附录 25](#_Toc8634521)

[7.1. 请求来源 25](#_Toc8634522)

[7.2. Http响应状态代码 26](#_Toc8634523)

[7.3. 错误码定义 26](#_Toc8634524)

# 前言

本文档主要用于定义第三方产品与智慧工厂平台之间的单点登录接入的接口规范，作为程序开发的依据。

# 接口概况

## 智慧工厂平台单点登录接口简介

智慧工厂平台单点登录接口主要用于将智慧工厂平台的能力放给第三方的开发工具及产品。

## 接口协议统一原则

所有服务的接口均基于HTTP协议。与SOAP或其他标准不同，REST要求用来操纵资源的方法或动词不是任意的。这意味着RESTful服务的开发人员只能使用HTTP支持的方法，比如GET、PUT、POST、DELETE等等。因此不需要使用WSDL等服务描述语言。

## 接口数据格式统一原则

1. 报文统一采用UTF-8进行编码。

2. 请求与响应的公共信息（请求时间，请求源等）存储在Http协议头部信息中。

3. 请求报文与响应报文中的业务数据，数据采用JSON格式表示，并且符合相应的JSON Schema。服务提供方和服务消费方必须同时使用同一种格式进行数据交互。JSON不同含义的段落应定义明确含义的字段名称，相同内容的数据应采用数组来进行描述，双方可根据JSON名称和路径进行精确定位，不应根据字段的顺序来获取字段值，字段值不受字段顺序调整的影响。

# 应用环境说明

## 测试环境

测试环境是智慧工厂平台提供给外部平台的线上运行测试环境，测试环境下的数据均为测试账号和测试数据，主要用于接口协议及接口报文测试，外部平台可以在测试环境下测试接口的功能。

## 正式环境

正式环境是智慧工厂平台提供给外部平台的正式线上运行环境。正式环境下的数据均为正式账号和正式数据。外部平台可以在正式环境下调用平台接口的功能，根据外部平台的申请的请求来源标识，在正式环境下由智慧工厂平台对外部平台进行认证校验，对未申请请求来源标识的的外部平台的访问被视为非法服务调用。

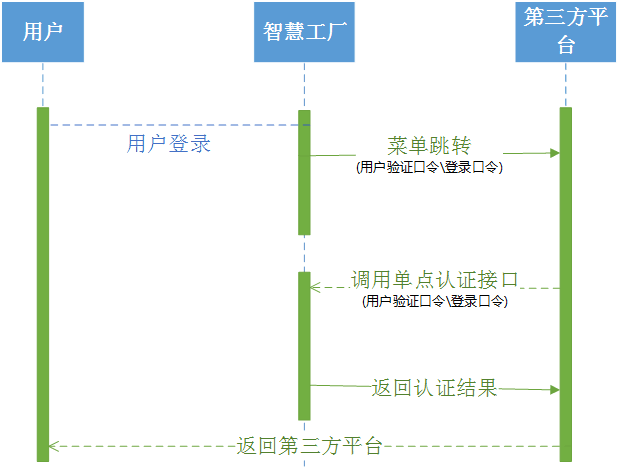
# 接口调用方式介绍

## Http调用方式详解

### 调用介绍

智慧工厂平台API（接口）采用 REST 风格，只需将所需参数拼装成http请求，即可调用，故支持 http 协议请求的程序语言，均可调用开放平台API，例如php、C#、asp、java、delphi 等。

API用户需要按照开放平台规范拼装一个正确的url，通过http请求到智慧工厂平台，就能够拿到需要的数据。调用原理示意图如下（注：）



1. 用户在智慧工厂平台登录
2. 用户进入对应的第三方平台宣传页，点击立即使用
3. 打开第三方平台页面，同时带上验证口令
4. 第三方平台接收到验证口令，同时调用单点认证接口，验证口令有效性以判断是否可以登录

### 调用入口

环境地址，即调用接口(API)时，都通过访问该地址，来获取该接口需要获取的数据。

**测试环境：**

HTTP://10.47.248.23:32005//nl-edc-cct-system/v1/sso/

**正式环境：**

HTTP://10.48.134.112:32005/nl-edc-cct-system/v1/sso/

# 服务报文规范

为保证对服务的接口消息结构进行统一的规范和定义，接口消息结构的统一将在数据接口层面上形成统一标准，实现在各模块之间与外部系统的数据交互的标准化和规范化。

REST架构风格的WEB服务1.2协议，而非仅仅是传输协议使用。因此报文的定义http协议进行扩展。

## 5.1消息的约束

表格中对元素约束（出现次数）的表示方法的描述：

|  |  |
| --- | --- |
| **符号** | **含义** |
| ? | 0..1，可选项 |
| \* | 0..n，可以没有，也可以有多项 |
| + | 1..n，至少有1项，也可以有多项 |
| 1 | 数字1，代表必须且只能填1项 |

## 5.2消息请求结构

REST请求就是HTTP或HTTPS的请求，请求消息结构也就是http请求的消息结构。消息请求结构包括消息请求行、消息请求头和消息请求正文。

请求行：描述请求动作。

请求头：描述http协议本身以及自定义的公共信息。

请求正文：描术请求的业务信息。

请求信息以各种“参数”的形式存在，分布在消息结构中，包括有请求路径参数，自定义头参数，查询字符串参数，请求体参数。

### 5.2.1消息请求行

请求行（Request Line）分为三个部分：请求方法、请求地址和协议及版本，以CRLF(rn)结束。

#### 5.2.1.1请求方法

HTTP/1.1定义的请求方法有8种：GET、POST、PUT、DELETE、PATCH、HEAD、OPTIONS、TRACE。

RESTful架构风格规定，数据的元操作，即CRUD（create，read，update和delete，即数据的增删查改）操作，需要分别对应于HTTP方法。具体要求如下：

|  |  |  |
| --- | --- | --- |
| **CRUD操作** | **说明** | **Http Method** |
| select | 从服务器取得资源（一项或多项） | get |
| create | 在服务器新建一个资源 | post |
| update | 在服务器更新资源 | put |
| delete | 从服务器删除资源 | delete |

注意:

1. JSR规范中对于更新操作定义有put，update两种操作，适用于不同的更新场景。本规范只定义了put，考虑到使用的方便性，以及要兼容低版本的情况（patch的注解在JDK1.8中才支持，对应javax.ws.rs-api-2.1.jar）。

2. Http协议本身没有对get时URL长度有做限制。如果是后台服务之间的调用，采用get方法不会有长度限制的问题存在。如果是浏览器端请求，并且长度可能用超过特定浏览器的限制，则需要变通用post方式提交。

#### 5.2.1.2请求地址与路径

REST接口服务的URL由以下几个部分组成：

**schema://host:port/path?queryString**

|  |  |  |
| --- | --- | --- |
| **元素名** | **说明** | **示例** |
| schema | 协议名称。通常是http或者https |  |
| host | 主机ip、主机名称或者域名 |  |
| port | 端口 | 8080 |
| ? | 用来分隔资源地址和查询字符串符号 |  |
| path | 定义资源的地址，使用"/"符号来分隔逻辑上的层次结构 | /cmcc-sys-xxx/v1/users |
| queryString | 查询字符串，使用"&"符号来分隔查询条件 |  |

资源地址通常使用ContextPath、ServletPath、VersionPath和PathInfo来细分资源地址。如下图所示：



* ContextPath是上下文名称，通常和Rest服务的web.xml配置有关。路径上下文要求能简洁的描述服务的来源与作用，建议至少包括有“服务提供商英文缩写”、“系统名称英文缩写”和“子系统（模块）名称英文缩写”的内容，并以中划线"-"相连接，如"cmcc-sys-xxx"。
* ServletPath是servlet的名称，通常和Rest服务的web.xml的配置有关。本规范中ServletPath用来表示api版本的信息，用来区分不同版本的服务接口，如"/v1"。
* PathInfo是资源路径信息，由主资源信息、子资源信息、参数等组成。

##### 5.2.1.2.1请求地址规范

1. 必须使用斜杆(/)分隔来表示分层关系。

2. 不要在URI中使用尾部的斜杆(/)。

3. 小写字母优先选用。

##### 5.2.1.2.2资源（包括路径参数）规范

1. 资源名称为名词，简明展示所请求的资源信息。资源名的命名采用英文。在URI中不要出现CRUD等动词，如getAllUsers，deleteRecord之类的定义。英文统一采用小写，如有联合单词，建议采用lowerCamelCase命名法。

2. 资源名称无特别情况下使用复数名词。如：/books，/settings。

3. 路径参数与查询参数的处理：只有主键查询，使用路径参数，如"/books/1"。当多参数时，建议采用查询参数的方式，如"/books?id=1&name=java"。

4. 使用子资源表达关系。如果一个资源与另外一个资源有关系，使用子资源。如："/shops/3/books"，"/shops/3/books/1"。

### 5.2.2消息请求头

Http header部分的内容遵循http协议的定义。

#### 5.2.2.1Http标准的报文头信息

|  |  |  |  |
| --- | --- | --- | --- |
| **元素名称** | **约束** | **中文描述** | **取值说明** |
| Authorization | ? | HTTP授权的授权证书 | bearer xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx |
| Content-Type | 1 | 请求的与实体对应的MIME信息 | text/plain、text/html、application/json、application/x-www-form-urlencoded等 |
| Accept | 1 | 指定客户端能够接收的内容类型 | application/json |
| Date | 1 | 请求发送的日期和时间 | Date: Mon, 25 Dep 2017 08:12:31 GMT |
| Cache-Control | 1 | 指定请求和响应遵循的缓存机制 | Cache-Control: no-cache |
| Content-Length | 1 | 请求Body长度 | 数值 |
| …… |  |  |  |

注意：从Http协议本身的协议头中就可以得到认证信息等有效信息。

#### 5.2.2.2自定义的报文头信息

为了描述请求的公共信息，在头部信息加入部分自定义头信息。定义要求所下：

1. 所有自定义header名中首字符大写。

2. 名称以"X"开头；单词之间采用连接符"-"。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **自定义名称** | **约束** | **类型** | **中文描述** | **取值说明** |
| X-UserId | 1 | String | 请求用户编号 |  |
| X-Req-Source | ? | String | 请求来源 | 参见附表 |
| X-Req-Module | ? | String | 请求的系统功能模块 |  |
| X-Password | ? | String | 请求的密码 |  |
| X-Req-Time | 1 | String | 请求时间 | 格式：YYYYMMDDHH24MISS |
| X-Req-Seq | ? | String | 请求流水号 | 请求的唯一编号，可以是UUID码 |
| X-Inner-KeyId | ? | String | 在内部认证时使用，指的是DES密钥对应的唯一编码 |  |
| X-Inner-Key | ? | String | 指的是DES算法的密钥 |  |

注意：从Http协议本身的协议头中就可以得到认证信息等有效信息。

### 5.2.3消息请求正文

#### 5.2.3.1get方式提交

get提交时，Content-Type采用"text/plain"或"text/html"。业务参数通过URL提交参数（字符串）。

如下所示，其中"/cmcc-sys-xxx/v1/books?bookid=1"为资源地址，"books"为资源名：

http://localhost:8080/cmcc-sys-xxx/v1/books?bookid=1

#### 5.2.3.2post方式提交

当post方式提交时，Content-Type采用"application/json"或"application/x-www-form-urlencoded"（处理表单数据时可采用"application/x-www-form-urlencoded"）。业务参数通过http body提交，一般以json方式组织。数据结构的描述大体如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **父参数名称** | **参数名称** | **约束** | **类型** | **中文描述** | **取值说明** |
|  | json对象名 | \* |  | json对象 |  |
| json对象名 | json属性1 | 1 |  | 属性1 |  |
| json对象名 | json属性2 | 1 |  | 属性2 |  |
| json对象名 | …… | 1 |  |  |  |

如下所示，其中"cmcc-sys-xxx/v1/books?bookid=1"为资源地址，"books"为资源名；业务数据是json结构：

http://localhost:8080/cmcc-sys-xxx/v1/books

其中，业务数据的json格式可以是json对象或者json对象数组：

{"bookId":1,"bookName":"MyBookName","publisher":"MyPublicsher"}

或者：

[{"bookId":1,"bookName":"MyBookName1","publisher":"MyPublicsher1"},{"bookId":2,"bookName":"MyBookName2","publisher":"MyPublicsher2"}]

#### 5.2.3.3 put、delete方式提交

当put、patch、delete方式提交时，既可以采用路径参数、查询参数的方式，也可以采用请求体参数的方式。建议当参数结构简单时，比如就一个id时，就采用路径参数、查询参数的方式；当参数结构复杂时，采用请求体参数，用json格式表示。

注意：

1. 至于post，put，delete的Content-Type均可以采用"multipart/form-data"或者"application/x-www-form-urlencoded"，在不方便使用"application/json"时（如上传文件时），注意在调用接口时要说明好Content-Type采用"multipart/form-data"（支持二进制文件上传、更新）还是"application/x-www-form-urlencoded"（默认）。可参见“示例”章节中post、put、delete的例子。

#### 5.2.3.4其他情况

当上述的标准的HTTP方法无法满足业务需求时，比如对于图书资源，除了基本的CRUD之外，若需要公布像借阅、折旧、电子版下载等实际生活中的更新操作的接口时，仅公布一个PUT方法就不够用了。这些操作是动词性的，无法简单地使用一个book名词就能定位。在路径变量难以准确描述的情况下：

1. 服务实现通过不同的重载函数；

2. 考虑使用动词作为查询参数；

3. 引入RPC风格的POST方法，辅助完成复杂业务的接口设计，也就是采用REST和RPC混合型的Web服务方式。

# 接口说明

## 单点认证接口

### 请求报文

|  |  |
| --- | --- |
| **接口描述** | 单点认证接口 |
| **请求方式** | POST |
| **请求路径** | http://ip:port/nl-edc-cct-system/v1/sso |
| **content-type** | application/json;charset=UTF-8 |
| **accept** | application/json;charset=UTF-8 |
| **返回数据结构** | json |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **父元素**  **名称** | **元素**  **名称** | **约**  **束** | **类型** | **中文**  **描述** | **取值**  **说明** |
|  | verifyCode | 1 | String | 用户验证口令 | 取自门户http传输过去的post请求参数:verify\_code |
|  | randomSeq | 1 | String | 登录口令 | 取自门户http传输过去的post请求参数:random\_seq |

### 应答报文

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **父元素**  **名称** | **元素**  **名称** | **约**  **束** | **类型** | **中文描述** | **取值说明** |
|  | success | 1 | String | 成功：ture 失败false |  |
|  | message | 1 | String | 登录返回信息 |  |
|  | data | 1 |  |  |  |
| data | userId | 1 | String | 用户ID |  |
| data | userName | 1 | String | 用户名称 |  |
| data | nickName | 1 | String | 用户别名 |  |
| data | orgId | 1 | String | 机构ID |  |
| data | orgName | 1 | String | 机构名称 |  |
| data | homeCity | 1 | String | 归属地市ID |  |
| data | homeCityName | 1 | String | 归属地市 |  |
| data | homeCounty | 1 | String | 归属县市ID |  |
| data | homeCountyName | 1 | String | 归属县市 |  |
| data | mobilePhone | 1 | String | 手机号 |  |
| data | mailAddr | 1 | String | 邮件地址 |  |

### 请求报文示例

curl -X POST \

http://10.47.248.23:32005/nl-edc-cct-system/v1/sso \

-H 'Content-Type: application/json' \

-d '{

"verifyCode": "a04f4faa1bd67d22d9d558b535363080",

"randomSeq": "9aiUjs3lIPVHOg8TJbH2KX6SUaAIXUTqglUHekzypbU="

}

### 应答报文示例

{

"success": true,

"message": "登陆成功",

"data": {

"userId" : "111",

"userName" : "张三",

"nickName" : "张三",

"orgId" : "机构ID",

"orgName" : "机构名称",

"homeCity" : "归属市ID",

"homeCityName" : "归属市名称",

"homeCounty": "归属县ID",

"homeCountyName": "归属县名称",

"mobilePhone": "手机号",

"mailAddr": "邮件地址"

}

}

## 所有租户用户关系列表获取

### 请求报文

|  |  |
| --- | --- |
| **接口描述** | 所有租户用户关系列表获取 |
| **请求方式** | POST |
| **请求路径** | http://ip:port/nl-edc-cct-system/v1/qryTenant |
| **content-type** | application/json;charset=UTF-8 |
| **accept** | application/json;charset=UTF-8 |
| **返回数据结构** | json |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 参数  类型 | 父元素  名称 | 元素  名称 | 约  束 | 类型 | 中文  描述 | 取值  说明 |
|  |  |  |  |  |  |  |

### 应答报文

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 父元素  名称 | 元素  名称 | 约  束 | 类型 | 中文描述 | 取值说明 |
|  | success | 1 | Object | 成功：ture 失败false |  |
|  | message | 1 | String | 返回信息 |  |
|  | data | \* |  |  |  |
| data | departId | 1 |  | 租户ID |  |
| data | departName | 1 |  | 租户名称 |  |
| data | departLevel | 1 |  | 租户等级 |  |
| data | departType | 1 |  | 租户类型 | 1 租户 2 用户组 |
| data | user | \* |  | 租户下用户列表 |  |
| user | userId | 1 |  | 用户id |  |
| user | userName | 1 |  | 用户名称 |  |
| user | nickName | 1 |  | 用户别名 |  |
| user | mobilePhone | 1 |  | 手机号码 |  |
| user | mailAddr | 1 |  | 邮件地址 |  |
| user | isAdmin | 1 |  | 是否租户管理员 | 1 是 0 否 |
| data | children | \* |  | 下级租户列表 |  |
| children | departId | 1 |  | 租户ID |  |
| children | departName | 1 |  | 租户名称 |  |
| children | departLevel | 1 |  | 租户等级 |  |
| children | departType | 1 |  | 租户类型 |  |
| children | children | \* |  | 下级租户列表 |  |
| children | user | \* |  | 租户下用户列表 |  |

### 请求报文示例

curl -X POST \

http://10.47.248.23:32005/nl-edc-cct-system/v1/qryTenant \

-H 'Content-Type: application/json' \

-d '{

}'

### 应答报文示例

{

"success": "true",

"message": "成功",

"data": [{

"departId": "T100000031",

"departName": "福州分公司",

"departLevel": "1",

"departType": "1",

"children": [{

"departId": "C871A6D6ECEDF6037C377300E7AC11E8",

"departName": "市场部租户",

"departLevel": "2",

"departType": "1",

"children": null,

"user": [{

"userId": "1000561",

"userName": "李四",

"nickName": "1000561",

"mobilePhone": "15880037583",

"mailAddr": "15880037583@139.com",

"isAdmin": 1

}, {

"userId": "1000502",

"userName": "张三",

"nickName": "1000502",

"mobilePhone": "13591000502",

"mailAddr": null,

"isAdmin": 0

}]

}, {

"departId": "D2A7EAD1A7BAC3B57B392DB007E911E9",

"departName": "鼓楼公司",

"departLevel": "2",

"departType": "1",

"children": [{

"departId": "D2A7EAD1A7BAC3B5A3BFEF3007E911E9",

"departName": "新东网-东街口大数据项目",

"departLevel": "3",

"departType": "2",

"children": null,

"user": [{

"userId": "1011285",

"userName": "张三",

"nickName": "1011285",

"mobilePhone": "13559175807",

"mailAddr": "13559175807@139.com",

"isAdmin": "1"

}]

}],

"user": [{

"userId": "1011285",

"userName": "张三",

"nickName": "1011285",

"mobilePhone": "13559175807",

"mailAddr": "13559175807@139.com",

"isAdmin": "1"

}]

}],

"user": [{

"userId": "1000206",

"userName": "张三",

"nickName": "1000206",

"mobilePhone": "15080098194",

"mailAddr": "15080098194@139.com",

"isAdmin": "0"

}, {

"userId": "1000511",

"userName": "张三",

"nickName": "1000511",

"mobilePhone": "13950402433",

"mailAddr": "13950402433@139.com",

"isAdmin": "0"

}]

}, {

"departId": "EFC8B64819651EE0F531A400EBD611E8",

"departName": "市场部",

"departLevel": "1",

"departType": "1",

"children": [{

"departId": "F9A8B7A4B76CFC4EEDDAECF0F82B11E8",

"departName": "集团一级电子渠道",

"departLevel": "2",

"departType": "1",

"children": null,

"user": [{

"userId": "9990035",

"userName": "张三",

"nickName": "zhengzhihuan",

"mobilePhone": "13600810081",

"mailAddr": "zhengzhihuan@fj.chinamobile.com",

"isAdmin": "1"

}]

}, {

"departId": "F0EBB925277B7148CCD74BB0EE3111E8",

"departName": "福建移动在线便利店",

"departLevel": "2",

"departType": "1",

"children": null,

"user": [{

"userId": "2006305",

"userName": "张三",

"nickName": "sunxiaozhui",

"mobilePhone": "13606065096",

"mailAddr": "13606065096@139.com",

"isAdmin": "1"

}]

}],

"user": [{

"userId": "9993787",

"userName": "张三",

"nickName": "9993787",

"mobilePhone": "15005003200",

"mailAddr": "0",

"isAdmin": "1"

}]

}]

}

## 单个用户信息列表获取

### 请求报文

|  |  |
| --- | --- |
| **接口描述** | 单个用户信息列表获取 |
| **请求方式** | POST |
| **请求路径** | http://ip:port/nl-edc-cct-system/v1/qryUser |
| **content-type** | application/json;charset=UTF-8 |
| **accept** | application/json;charset=UTF-8 |
| **返回数据结构** | json |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 父元素  名称 | 元素  名称 | 约  束 | 类型 | 中文  描述 | 取值  说明 |
|  | userId |  | String | 用户id |  |

### 应答报文

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 父元素  名称 | 元素  名称 | 约  束 | 类型 | 中文描述 | 取值说明 |
|  | success | 1 | Object | 成功：ture 失败false |  |
|  | message | 1 | String | 返回信息 |  |
|  | data | 1 |  |  |  |
| data | userId | 1 | String | 用户ID |  |
| data | userName | 1 | String | 用户名称 |  |
| data | nickName | 1 | String | 用户别名 |  |
| data | orgId | 1 | String | 机构ID |  |
| data | orgName | 1 | String | 机构名称 |  |
| data | homeCity | 1 | String | 归属地市ID |  |
| data | homeCityName | 1 | String | 归属地市 |  |
| data | homeCounty | 1 | String | 归属县市ID |  |
| data | homeCountyName | 1 | String | 归属县市 |  |
| data | mobilePhone | 1 | String | 手机号 |  |
| data | mailAddr | 1 | String | 邮件地址 |  |

### 请求报文示例

curl -X POST \

http://10.47.248.23:32005/nl-edc-cct-system/v1/qryUser \

-H 'Content-Type: application/json' \

-d '{

"userId": "9991227"

}'

### 应答报文示例

{

"success": "true",

"message": "查询成功",

"data": {

"userId": 9991227,

"userName": "新大陆",

"nickName": "9991227",

"orgId": 9990002,

"orgName": "BOSS开发商",

"homeCity": 590,

"homeCityName": "省中心",

"homeCounty": null,

"homeCountyName": null,

"mobilePhone": "13333333",

"mailAddr": "133333333@139.com"

}

# 附录

## 请求来源

### 移动内部来源

请求来源的编码规则利用8位数字区分，其中前四位地域信息采用城市区号后补0来表示，如果是省信息采用省会城市区号最后一位补9来表示。例如：乌鲁木齐的区号是0991，编码则为9910；北京的区号是010，编码则为1000；江苏省省会城市区号是025，省编码则为2509；福建省省会城市区号为0591，省编码则为5919。其中后四位表示系统和子系统，五六两位代表系统，七八两位代表子系统。

|  |  |  |  |
| --- | --- | --- | --- |
| **系统标识** | **系统说明** | **请求来源** | **请求描述** |
| 01 | BASS | 59190100 |  |
| 02 | BOSS | 59190200 |  |
| 03 | CRM系统 | 59190300 |  |
| 04 | BOMC系统 | 59190400 |  |
| 05 | VGOP | 59190500 |  |
| …… |  |  |  |

详见《FJBASS-数据封装-请求来源.xls》。

### 移动外部来源

请求来源的编码规则利用8位数字区分，其中前四位地域信息采用城市区号后补0来表示。第4，5位表示行业的拼音缩写。其中后四位表示系统和子系统，五六两位代表系统，七八两位代表子系统。

|  |  |  |
| --- | --- | --- |
| **请求来源** | **系统说明** | **请求描述** |
| 5919XH0001 | 中国银行福建分行XXX系统 |  |
| …… |  |  |
|  |  |  |

## Http响应状态代码

常见状态代码、状态描述的说明如下：

200 OK：客户端请求成功。

400 Bad Request：客户端请求有语法错误，不能被服务器所理解。

401 Unauthorized：请求未经授权，这个状态代码必须和WWW-Authenticate报头域一起使用。

403 Forbidden：服务器收到请求，但是拒绝提供服务。

404 Not Found：请求资源不存在，举个例子：输入了错误的URL。

500 Internal Server Error：服务器发生不可预期的错误。

503 Server Unavailable：服务器当前不能处理客户端的请求，一段时间后可能恢复正常，举个例子：HTTP/1.1 200 OK（CRLF）。

## 错误码定义

应答消息结果定义如下：

|  |  |  |
| --- | --- | --- |
| **应答错误类型** | **应答错误代码** | **应答错误描述** |
| 900 | AuthErr | 鉴权错误 |
| 901 | Sqlcode | 数据库错误，错误代码由sqlcode指定 |
| 902 | Errno | 操作系统错误，错误代码由errno指定 |
| 903 | Errcode | Web容器错误，错误代码由errcode指定 |
| 904 | TimeoutErr | 超时错误 |
| 905 | MessageErr | 服务消息错误 |
| 906 | NetErr | 网络连接错误 |
| 907 | IOErr | IO错误 |
| … | … | … |
| 999 | OtherErr | 其他错误 |