统一认证平台接口说明

# 概述

## 说明

本文档用于统一认证平台接入用户信息相关接口和子系统向统一认证平台进行用户信息同步调用接口说明及接入指引。

## 用户信息同步API列表

测试环境访问地址：

内网：http://172.17.3.63:7031/openapi/v1/请求地址

内网对应外网：http://36.112.25.13:7031/openapi/v1/请求地址

|  |  |  |
| --- | --- | --- |
| 接口 | 请求方式 | 请求地址 |
| 用户删除 | DELETE | /user/delete |
| 用户新增 | POST | /user/patchUser |
| 用户修改 | POST | /user/update |
| 分配角色 | POST | /user/empower |
| 修改密码 | POST | /user/changePwd |
| 根据用户名查询用户 | GET | /user/getUserByUserName |
| 根据手机号查询用户 | GET | /user/getUserByMobile |
| 根据邮箱查询用户 | GET | /user/getUserByEmail |
| 查询手机号是否存在 | GET | /user/findByMobile |
| 查询邮箱是否存在 | GET | /user/findByEmail |
| 查询用户名是否存在 | GET | /user/findByUserName |
| 查询应用下所有用户 | GET | /user/getClientAllUser |

# API接口说明

1. 已经接入统一认证平台
2. 获得应用标识（系统码）和接口调用密钥
3. 配置接口访问接口访问地址以及密钥

## 统一请求头

应用标识（系统码）和接口调用密钥放到请求头。

Header包含两个参数clientCode和token，具体描述如下。

* clientCode：系统码xx
* token：根据body+params+secret计算后得到的签名内容 ，即sign(body, params,secret)。注意：参数说明如下。具体生成方式如2.1.1加密说明章节：

|  |  |  |
| --- | --- | --- |
| token生成携带字段说明 | 类型 | 说明 |
| body | String | API的请求body体 |
| params | String | 跟在API请求地址URI后的参数 |
| secret | String | 对接时，由认证平台SSO在“应用管理”根据对接系统配置的秘钥提供 |

### **2.1.1 加密说明**

请求参数要经过加密签名，签名内容放到请求头。

签名示例代码：

public static String sign*(*String body, Map*<*String, String*[]>* params,String secret*) {* StringBuilder sb = new StringBuilder*()*;  
 if *(*StringUtils.*isNotBlank(*body*)) {* sb.append*(*body*)*.append*(*'#'*)*;  
 *}* if *(*!CollectionUtils.*isEmpty(*params*)) {* params.entrySet*()* .stream*()* .sorted*(*Map.Entry.*comparingByKey())* .forEach*(*paramEntry -> *{* String paramValue = String.*join(*",", Arrays.*stream(*paramEntry.getValue*())*.sorted*()*.toArray*(*String*[]*::new*))*;  
 sb.append*(*paramEntry.getKey*())*.append*(*"="*)*.append*(*paramValue*)*.append*(*'#'*)*;  
 *})*;  
 *}* sb.append*(*secret*)*;  
 String createSign = DigestUtils.*sha256Hex(*sb.toString*())*;  
 return createSign;  
*}*public static void main*(*String*[]* args*) {* String body = "{\n" +  
 "\t\"name\": \"hjzgg\",\n" +  
 "\t\"age\": 26\n" +  
 "}";//请求体  
 Map*<*String, String*[]>* params = new HashMap*<>()*;//请求参数  
 params.put*(*"var3", new String*[]{*"3"*})*;  
 params.put*(*"var4", new String*[]{*"4"*})*;  
 String sec="123";//密钥  
  
 System.*out*.println*(sign(*body, params,sec*))*;  
*}*

## 统一响应说明

|  |  |  |  |
| --- | --- | --- | --- |
| 返回字段 | 字段类型 | 必选 | 说明 |
| status | int | Y | 返回状态,1 成功 -2 失败 -3 无权限 |
| message | String (128) | Y | 返回消息 |
| data | T | Y | 返回数据 |

# API 列表

## 用户新增

请求地址:/user/patchUser

请求方式：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 必填 | 说明 |
| roleCode | List<String> | false | 角色码-由用户统一认证平台上分配的角色码 |

请求体：

|  |  |  |  |
| --- | --- | --- | --- |
| simpleUserBO | 对象  SimpleUserBO | Y | 用户信息{userName:xx mobile:xx ...} |
| userName | String | Y | 用户名 |
| mobile | String | Y | 手机号 |
| email | String | Y | 邮箱 |
| remarks | String | false | 备注 |
| password | String | Y | 密码 |
| acctName | String | false | 账号名 |

响应参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 返回字段 | 字段类型 | 必选 | 说明 |
| status | int | Y | 返回状态,1 成功 -2 失败 |
| message | String (128) | Y | 返回消息 |
| data | SysUserBO | Y | SysUserBO对象 失败时为null |

状态-2失败结果说明：

|  |  |
| --- | --- |
| 结果描述 | 说明 |
| 传入用户为空 | 用户名为空 |
| 超级管理员无需新增 | 用户名root不允许添加 |
| 用户名重复 | 用户名已经存在 |
| 邮箱重复 | 邮箱已经存在 |
| 手机号重复 | 手机号已经存在 |
| 平台无该应用角色信息 | 用户统一认证平台内没有该子系统角色 |
| 角色码不属于该应用 | 角色码错误不属于本子系统角色不允许添加 |
| 不允许添加应用管理员 | 子应用管理员不允许通过接口创建 |
| 不允许添加管理员 | 管理员角色码ROLE\_ADMIN不允许添加 |
| 系统异常 | 系统内部错误 |

## 用户修改

请求地址:/user/update

请求方式：POST

请求体：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 必填 | 说明 |
| updateUser | UpdateUserBO | Y | 用户信息 |
| userName | String | Y | 用户名--用户名不可修改 |
| mobile | String | Y | 手机号 |
| emali | String | Y | 邮箱 |
| remarks | String | false | 备注 |
| password | String | Y | 密码---不可修改（密文） |
| acctName | String | false | 账号名 |

响应参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 返回字段 | 字段类型 | 必选 | 说明 |
| status | int | Y | 返回状态,1 成功 -2 失败 |
| message | String | Y | 返回消息 |
| data | Boolean | Y | 成功true 失败false |

状态-2失败结果说明：

|  |  |
| --- | --- |
| 结果描述 | 说明 |
| 用户不存在 | 用户名在统一认证平台不存在 |
| 密码不正确不允许修改 | 密码不能修改（密码修改另有接口），但需要传递密码 |
| 邮箱被其他用户占用 | 修改的邮箱已经存在 |
| 手机号被其他用户占用 | 修改的手机号已经存在 |
| 系统异常 | 系统内部错误 |

## 密码修改

请求地址:/user/changePwd

请求方式：POST

请求体：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 必填 | 说明 |
| changePwdBo | 对象  ChangePwdBO | Y |  |
| userName | String | Y | 用户名 |
| originPwd | String | Y | 原始密码 |
| newPwd | String | Y | 新密码 |

响应参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 返回字段 | 字段类型 | 必选 | 说明 |
| status | int | Y | 返回状态,1 成功 -2 失败 |
| message | String | Y | 返回消息 |
| data | Boolean | Y | 成功true 失败false |

状态-2失败结果说明：

|  |  |
| --- | --- |
| 结果描述 | 说明 |
| 用户不存在 | 用户名在统一认证平台不存在 |
| 修改密码和原始密码不能一样 | 新密码和旧密码一样 |
| 原始密码错误 | 原密码错误不允许修改 |
| 系统异常 | 系统内部错误 |

## 分配角色

请求地址:/user/empower

请求方式：POST

body请求体：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 必填 | 说明 |
| empowerBO | 对象  EmpowerBO | Y |  |
| userName | String | Y | 用户名 |
| override | Boolean | Y | 是否覆盖默认false true时会覆盖原有角色 false时会保留原有角色并追加新角色 |
| roleCode | List<String> | Y | 角色码--统一认证平台获得 |

响应参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 返回字段 | 字段类型 | 必选 | 说明 |
| status | int | Y | 返回状态,1 成功 -2 失败 |
| message | String | Y | 返回消息 |
| data | Boolean | Y | 成功true 失败false |

状态-2失败结果说明：

|  |  |
| --- | --- |
| 结果描述 | 说明 |
| 用户不存在 | 用户名在统一认证平台不存在 |
| 平台无该应用角色信息 | 用户统一认证平台内没有该子系统角色 |
| 角色码不属于该应用 | 角色码错误不属于本子系统角色不允许添加 |
| 不允许添加应用管理员 | 子应用管理员不允许通过接口创建 |
| 不允许添加管理员 | 管理员角色码ROLE\_ADMIN不允许添加 |
| 不允许添加超级管理员 | 用户名root不允许添加 |
| 系统异常 | 系统内部错误 |

## 用户删除

请求地址:/user/delete

请求方式：DELETE

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 必填 | 说明 |
| userName | String | Y | 用户名 |

响应参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 返回字段 | 字段类型 | 必选 | 说明 |
| status | int | Y | 返回状态,1 成功 -2 失败 |
| message | String | Y | 返回消息 |
| data | Boolean | Y | 成功true 失败false |

状态-2失败结果说明：

|  |  |
| --- | --- |
| 结果描述 | 说明 |
| 用户不存在 | 用户名在统一认证平台不存在 |
| 用户不属于该应用不允许删除 | 只能删除自己应用下的用户 |
| 系统异常 | 系统内部错误 |

## 根据用户名查询用户

请求地址:/user/getUserByUserName

请求方式：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 必填 | 说明 |
| userName | String | Y | 用户名 |

响应参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 返回字段 | 字段类型 | 必选 | 说明 |
| status | int | Y | 返回状态,1 成功 -2 失败 |
| message | String (128) | Y | 返回消息 |
| data | SysUserBO | Y | SysUserBO对象 失败时为null |

## 根据手机号查询用户

请求地址:/user/getUserByMobile

请求方式：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 必填 | 说明 |
| mobile | String | Y | 手机号 |

响应参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 返回字段 | 字段类型 | 必选 | 说明 |
| status | int | Y | 返回状态,1 成功 -2 失败 |
| message | String (128) | Y | 返回消息 |
| data | SysUserBO | Y | SysUserBO对象 失败时为null |

## 根据邮箱查询用户

请求地址:/user/getUserByEmail

请求方式：GET

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 必填 | 说明 |
| email | String | Y | 邮箱 |

响应参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 返回字段 | 字段类型 | 必选 | 说明 |
| status | int | Y | 返回状态,1 成功 -2 失败 |
| message | String (128) | Y | 返回消息 |
| data | SysUserBO | Y | SysUserBO对象 失败时为null |

## 查询手机号是否存在

请求地址:/user/findByMobile

请求方式：GET

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 必填 | 说明 |
| mobile | String | Y | 手机号 |

响应参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 返回字段 | 字段类型 | 必选 | 说明 |
| status | int | Y | 返回状态,1 成功 -2 失败 |
| message | String (128) | Y | 返回消息 |
| data | Boolean | Y | 已存在true 不存在false |

## 查询邮箱是否存在

请求地址:/user/findByEmail

请求方式：GET

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 必填 | 说明 |
| email | String | Y | 邮箱 |

响应参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 返回字段 | 字段类型 | 必选 | 说明 |
| status | int | Y | 返回状态,1 成功 -2 失败 |
| message | String (128) | Y | 返回消息 |
| data | Boolean | Y | 已存在true 不存在false |

## 查询用户名是否存在

请求地址:/user/findByUserName

请求方式：GET

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 必填 | 说明 |
| userName | String | Y | 用户名 |

响应参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 返回字段 | 字段类型 | 必选 | 说明 |
| status | int | Y | 返回状态,1 成功 -2 失败 |
| message | String (128) | Y | 返回消息 |
| data | Boolean | Y | 已存在true 不存在false |

## 查询应用下所有用户

请求地址:ssoApi/user/getClientAllUser

请求方式：GET

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 必填 | 说明 |
| currentPage | Integer | N | 第几页 |
| pageSize | Integer | N | 每页条数默认10 |

响应参数:

|  |  |  |  |
| --- | --- | --- | --- |
| 返回字段 | 字段类型 | 必选 | 说明 |
| status | int | Y | 返回状态,1 成功 -2 失败 |
| message | String (128) | Y | 返回消息 |
| data | R<P> | Y | 用户信息列表 |

# 接口调用

## Fegin方式

引入依赖

<!--统一认证平台接口-->  
*<*dependency*>  
 <*groupId*>*cn.caict*</*groupId*>  
 <*artifactId*>*sso-external-service*</*artifactId*>  
 <*version*>*0.0.2*</*version*>  
</*dependency*>*

Springboot启动类添加注解：

@EnableFeignClients*(*basePackages = "cn.caict.sso"*)*

调用示例：

@Autowired  
private IUserService iUserService;

@Test  
public void getUserByUserName*(){* R*<*SysUserBO*>* r = iUserService.getUserByUserName*(*"user1"*)*;  
 System.*out*.println*(*JSONUtil.*toJsonStr(*r.getData*()))*;  
*}*

如有Fegin版本问题可参考sso-external-service 项目，自己项目本地实现fegin调用。

版本问题参考：

<https://www.csdn.net/tags/MtjakgysNTgzMTgtYmxvZwO0O0OO0O0O.html>

## RestTemplate方式

@Test  
public void restTemplateTest*(){* RestTemplate restTemplate = new RestTemplate*()*;  
  
 SimpleUserBO simpleUserBO =new SimpleUserBO*()*;  
 simpleUserBO.setEmail*(*"123@qq.com"*)*;  
 simpleUserBO.setMobile*(*"18131621"*)*;  
 simpleUserBO.setPassword*(*"213132"*)*;  
 simpleUserBO.setUserName*(*"tttt"*)*;  
 // 默认基础角色  
 List*<*String*>* sysRoles = new ArrayList*<>()*;  
 sysRoles.add*(*"ROLE\_ENT"*)*;  
  
  
 String url = "http://172.17.3.63:7031/openapi/v1/user/patchUser?roleCode="+sysRoles;  
  
 //设置Http的Header  
 HttpHeaders headers = new HttpHeaders*()*;  
 headers.setContentType*(*MediaType.*APPLICATION\_JSON\_UTF8)*;  
  
 //设置访问参数  
 Map*<*String, String*[]>* params = new HashMap*<>()*;  
 params.put*(*"roleCode",new String*[]{*sysRoles.toString*()})*;  
  
 //得到发送的body签名（直接签名body中属性顺序可能不一致，造成签名失败）  
 HttpEntity*<*String*>* tmp = new HttpEntity*<*String*>(*JSONUtil.*toJsonStr(*simpleUserBO*))*;  
 String strBody=tmp.getBody*()*;  
 //签名  
 String sign= SignUtil.*sign(*strBody,params,secret*)*;  
 headers.add*(*RequestHeaderConstants.*ACCESS\_TOKEN*, sign*)*;  
 headers.add*(*RequestHeaderConstants.*CLIENT\_CODE*,clientCode*)*;  
  
 //设置访问的Entity  
 HttpEntity*<*String*>* entity = new HttpEntity*<*String*>(*JSONUtil.*toJsonStr(*simpleUserBO*)*, headers*)*;  
 ResponseEntity*<*String*>* result = null;  
 try *{* //发起一个POST请求  
 result = restTemplate.exchange*(*url, HttpMethod.*POST*, entity, String.class*)*;  
 JSONObject data = JSONObject.*parseObject(*result.getBody*())*;  
 System.*out*.println*(*JSONUtil.*toJsonStr(*data*))*;  
 *}* catch *(*Exception e*) {* System.*out*.println*(*"异常"*)*;  
 *}  
}*

## Postman方式

### 示例：用户新增user/patchUser

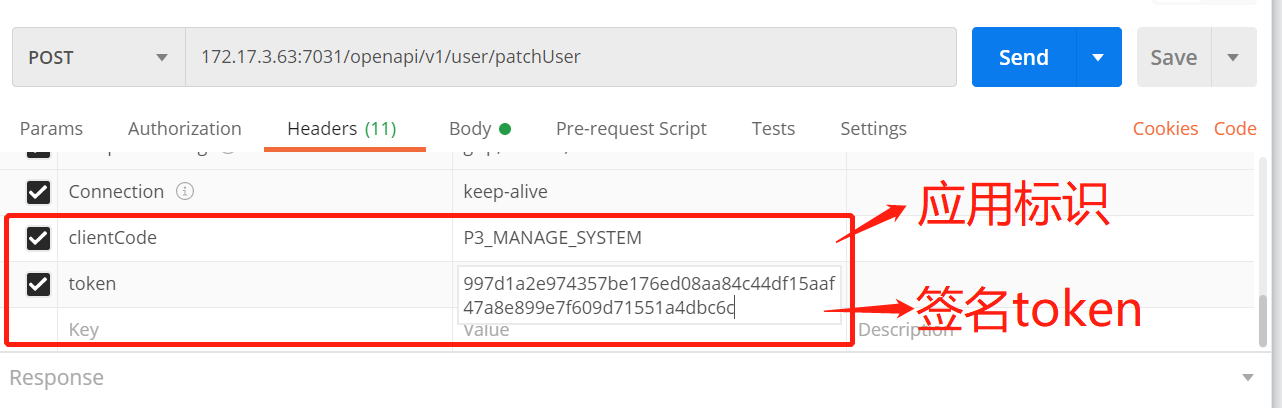
1. 签名

public static void main*(*String*[]* args*) {* //密钥  
 String secret="16952dbc378233518154c1ed73e7edf8";  
 //请求body-用户信息  
 SimpleUserBO simpleUserBO =new SimpleUserBO*()*;  
 simpleUserBO.setEmail*(*"12345678@qq.com"*)*;  
 simpleUserBO.setMobile*(*"18131515673"*)*;  
 simpleUserBO.setPassword*(*"123456"*)*;  
 simpleUserBO.setUserName*(*"zhangsan"*)*;  
 // 请求参数-默认用户基础角色  
 List*<*String*>* sysRoles = new ArrayList*<>()*;  
 sysRoles.add*(*"ROLE\_ENT"*)*;  
  
 //设置访问参数用于签名  
 Map*<*String, String*[]>* params = new HashMap*<>()*;  
 params.put*(*"roleCode",sysRoles.toArray*(*new String*[*sysRoles.size*()]) )*;  
 //设置访问请求body用于签名  
 String strBody=JSONUtil.*toJsonStr(*simpleUserBO*)*;  
 //可取此值放在postman请求体  
 System.*out*.println*(*strBody*)*;  
 //签名token  
 String sign= SignUtil.*sign(*strBody,params,secret*)*;  
 System.*out*.println*(*sign*)*;

1. 设置请求头

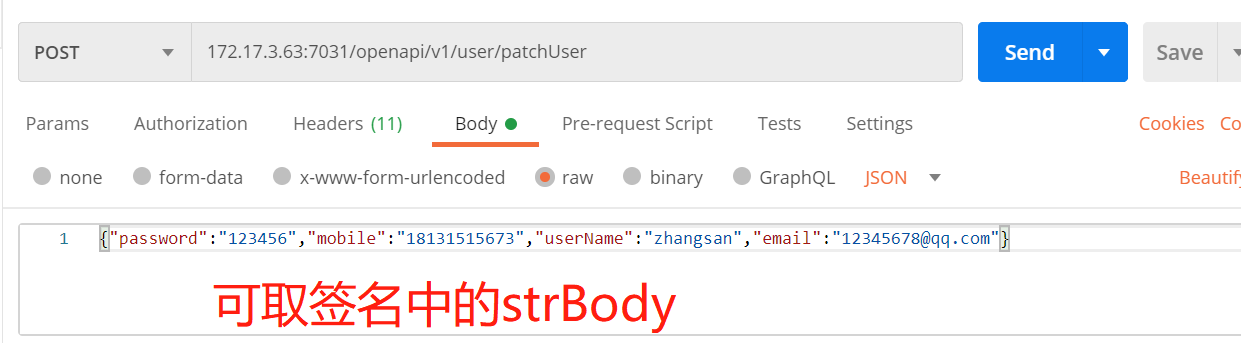
clientCode：应用标识

token：签名token

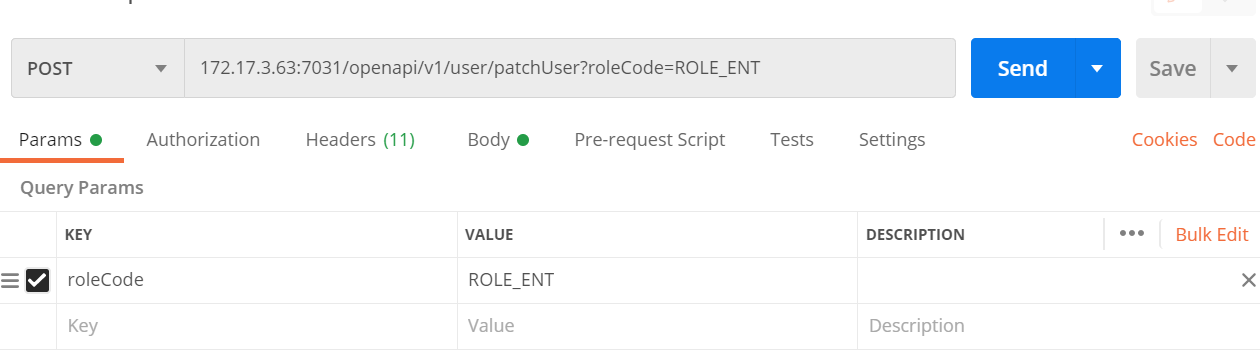


1. 设置请求体和请求参数

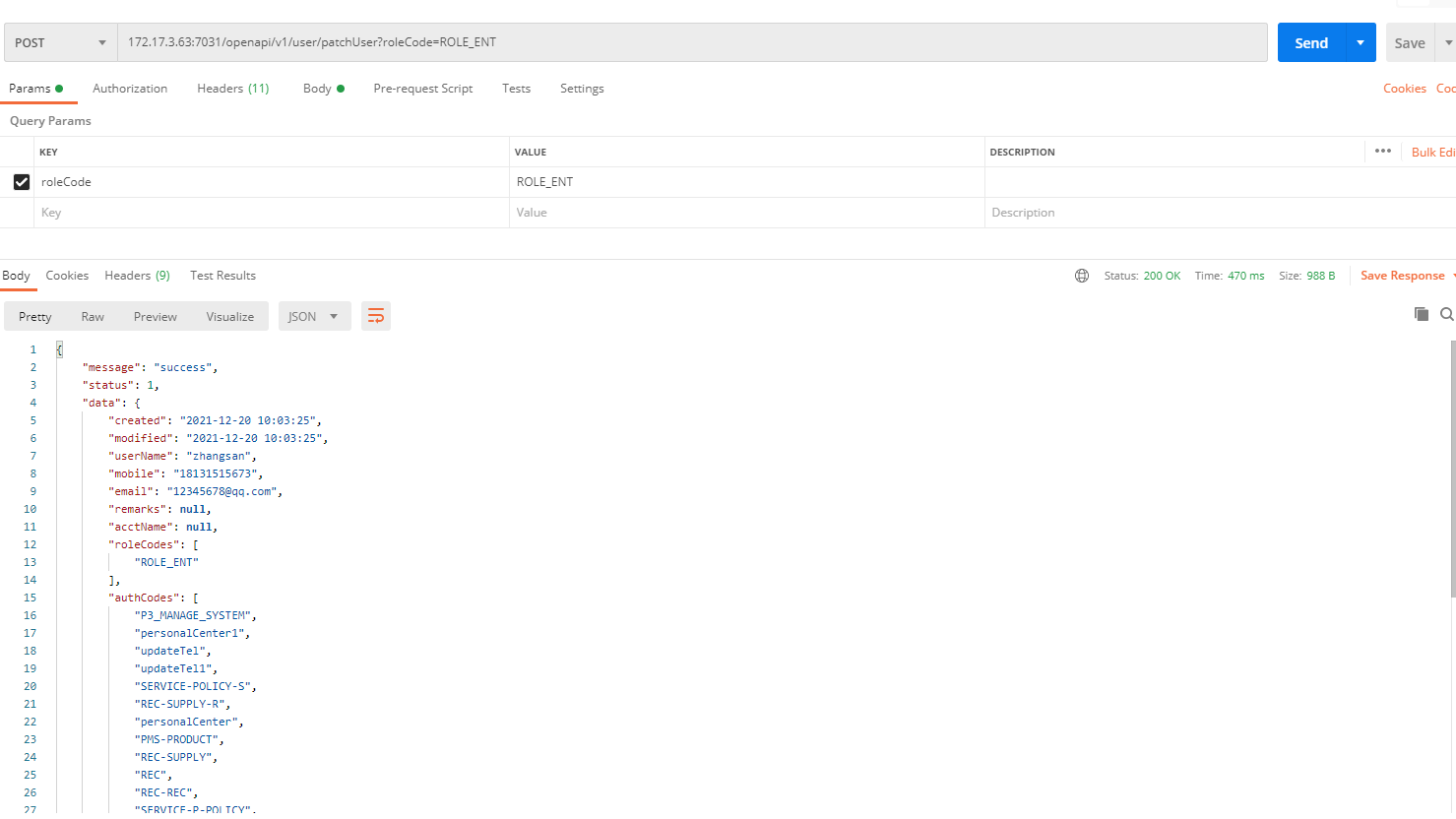
请求体



请求参数



1. 发送请求



### 示例：根据手机号查询用户信息/user/getUserByMobile

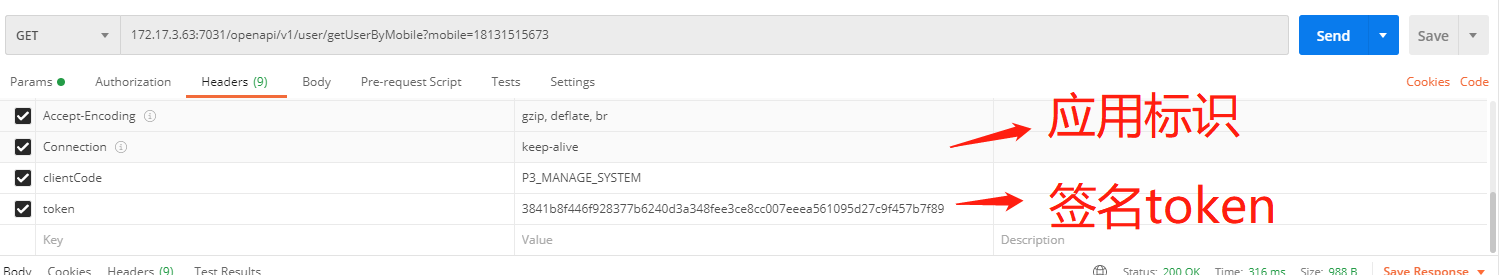
1. 签名

public static void main*(*String*[]* args*) {* //密钥  
 String secret="16952dbc378233518154c1ed73e7edf8";  
 //设置访问参数用于签名  
 Map*<*String, String*[]>* params = new HashMap*<>()*;  
 params.put*(*"mobile",new String*[]{*"18131515673"*})*;  
 //签名token  
 String sign= SignUtil.*sign(*null,params,secret*)*;  
 System.*out*.println*(*sign*)*;  
  
*}*

1. 设置请求头

clientCode：应用标识

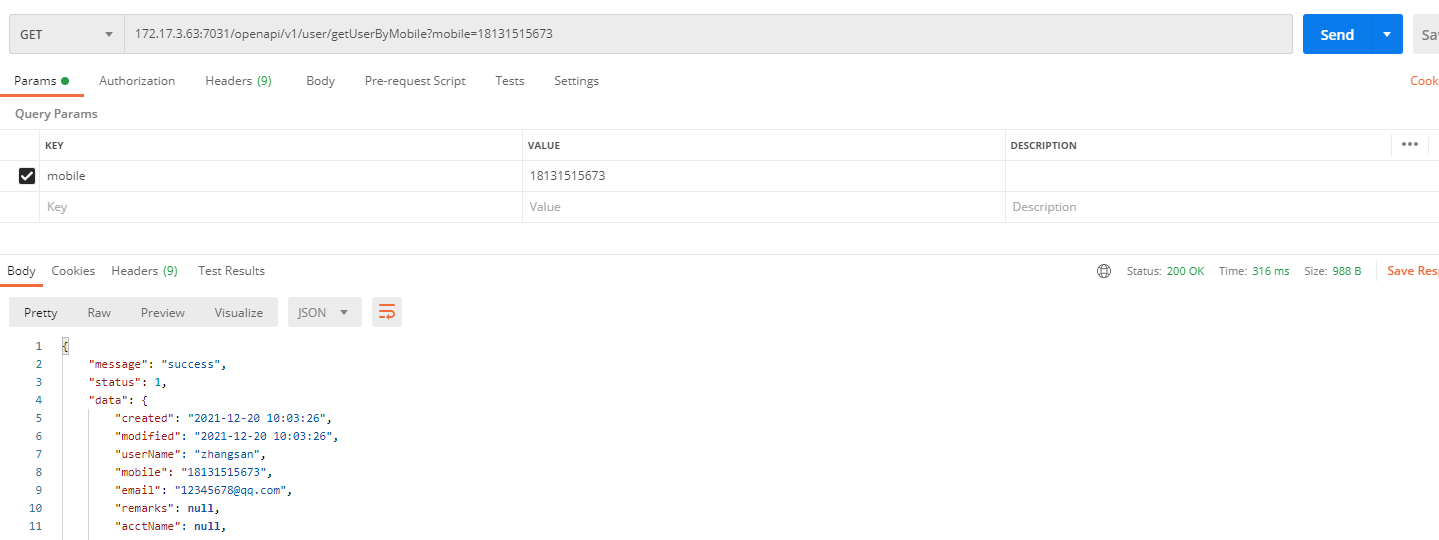
token：签名token



1. 设置请求参数



1. 发送请求

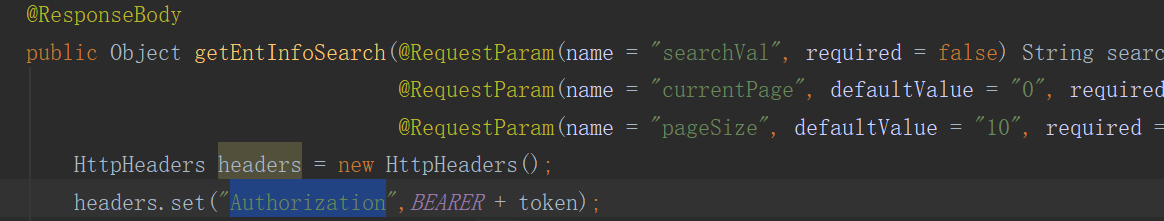


# 用户密码加密说明

加密方式必须为SHA256(明文密码)

# 其他用户信息相关接口说明

以下接口可以提供用户的一些基础信息和权限信息，如下所示为localhost本地访问的方式，如果是访问sso认证中心，请在请求头里面加上Authorization（如下图）。其中6.1.1-6.1.6为通用型接口，6.2.1-6.2.2为和P3平台共用用户的系统专用的接口，如非和P3平台共用用户的系统可忽略6.2章节中的内容。

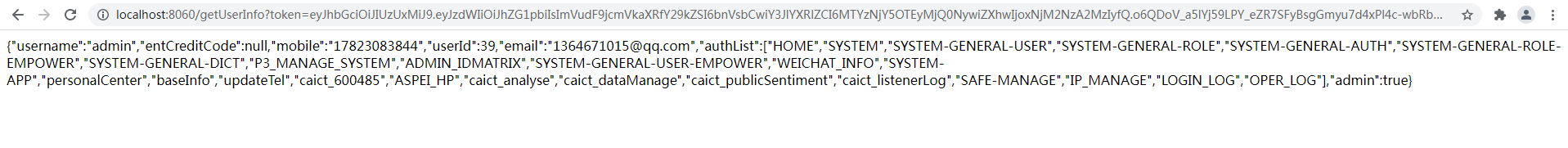


说明：如果接口返回的status的值为-2，则需要客户端需要调用单点登录的logout接口退出系统。若返回值为-2，一种情况访问sso未携带token，一种情况token已过期。

## 通用型接口

### 6.1.1获取用户信息和企信码接口--灾备数据托管、灾备门户

http://localhost:8060/getUserInfo?token=xxx



返回值说明:

|  |  |
| --- | --- |
| 字段 | 说明 |
| username | 用户名 |
| entCreditCode | 企信码（如果是和P3平台共用用户的系统，可以使用此字段，否则此字段为null值，可以忽略此字段） |
| mobile | 用户手机号 |
| email | 用户邮箱 |
| authList | 用户数据权限集合 |
| admin | 是否为管理员 |

### 6.1.2根据用户名获取数据权限

http://36.112.25.13:7031/authorization/getDataAuthByUsername?username= 用户名

请求参数:

|  |  |
| --- | --- |
| 字段 | 说明 |
| username | 用户名 |

**请求头: Cookie: token = 登录后后取到的token值（必须携带）**

响应参数:

|  |  |
| --- | --- |
| 字段 | 说明 |
| message | 响应说明 |
| status | 响应码： 1表示成功 -2:表示失败，在客户端应该退出登录，调用/logout接口 |
| data | list |
| --id | Id（后期会去掉，不要使用） |
| --userId | 用户id（后期会去掉，不要使用） |
| --authId | 权限id（后期会去掉，不要使用） |
| --authCode | 权限码 |

### 6.1.3根据用户名获取菜单

http://36.112.25.13:7031/authorization/findAuthByUsername?username= 用户名

请求参数:

|  |  |
| --- | --- |
| 字段 | 说明 |
| username | 用户名 |

**请求头: Cookie: token = 登录后后取到的token值（必须携带）**

响应参数:

|  |  |
| --- | --- |
| 字段 | 说明 |
| message | 响应说明 |
| status | 响应码： 1表示成功 -2:表示失败，在客户端应该退出登录，调用/logout接口 |
| data | list |
| authName | 菜单名 |
| authCode | 菜单编码 |
| path | 路径 |

### 6.1.4根据用户名获取用户id

http://36.112.25.13:7031/authorization/getUserId?username= 用户名

请求参数:

|  |  |
| --- | --- |
| 字段 | 说明 |
| username | 用户名 |

**请求头: Cookie: token = 登录后后取到的token值（必须携带）**

响应参数:

|  |  |
| --- | --- |
| 字段 | 说明 |
| message | 响应说明 |
| status | 响应码： 1表示成功 -2:表示失败，在客户端应该退出登录，调用/logout接口 |
| data | Id值 |

### 6.1.5刷新token

（1）JAVA版接入可替换ssoclient的jar包版本ssoclien-1.0.3.jar (支持token刷新)

（2）如需自己调用参考如下：

如token过期需要支持刷新token，拿旧的token和系统码去请求新的token

请求地址:http://ip:port/auth/sso/refreshToken?token=xxx&systemCode=xxx

请求参数: token =xxx systemCode=xx

请求方式：POST

响应参数:

|  |  |
| --- | --- |
| 参数 | 说明 |
| status | 200 表示成功 -1标识token已失效无法刷新 其他情况表示失败 |
| message | ok |
| data | 新的token |

获取到新的token后需要删除cookie中旧的token 并设置成新的token，拿新的token去请求。

### 6.1.6 获取登录用户信息-灾备数据托管服务

请求地址:http://ip:port/auth/sso/getUserInfo?systemCode=xxx

请求cookie: token=xx，**注意：Cookie必须携带token值，否则接口认证失败。**

请求参数: systemCode=xx

请求方式：POST

响应参数:

|  |  |
| --- | --- |
| 参数 | 说明 |
| username | 用户名 |
| mobile | 手机号 |
| email | 邮箱 |
| remarks | 备注 |

如果用户未登录，则返回为null

## P3专用型接口

### 6.2.1根据用户名获取企业信息

注意：此接口已经在sso-auth:1.0.5版本中移除，如需要请调用P3V2。

http://36.112.25.13:7031/authorization/getEntInfoByUserName?username=用户名

请求参数:

|  |  |
| --- | --- |
| 字段 | 说明 |
| username | 用户名 |

**请求头: Cookie: token = 登录后后取到的token值（必须携带）**

响应参数:

|  |  |
| --- | --- |
| 字段 | 说明 |
| message | 响应说明 |
| status | 响应码： 1表示成功 -2:表示失败，在客户端应该退出登录，调用/logout接口 |
| data | list |
| userId | 用户id |
| userName | 用户名 |
| entName | 企业名称 |
| entId | 企业id |
| businessAddress | 企业地址 |
| entCreditCode | 企信码 |
| dateIssue | 发照日期 |
| companyType | 企业类型 |
| industry | 行业类型 |
| entGeneralCreditCode | 社会信用码 |
| qrCodePath | 企信码路径 |
| provinceName | 所属省 |
| countyName | 所属城市 |

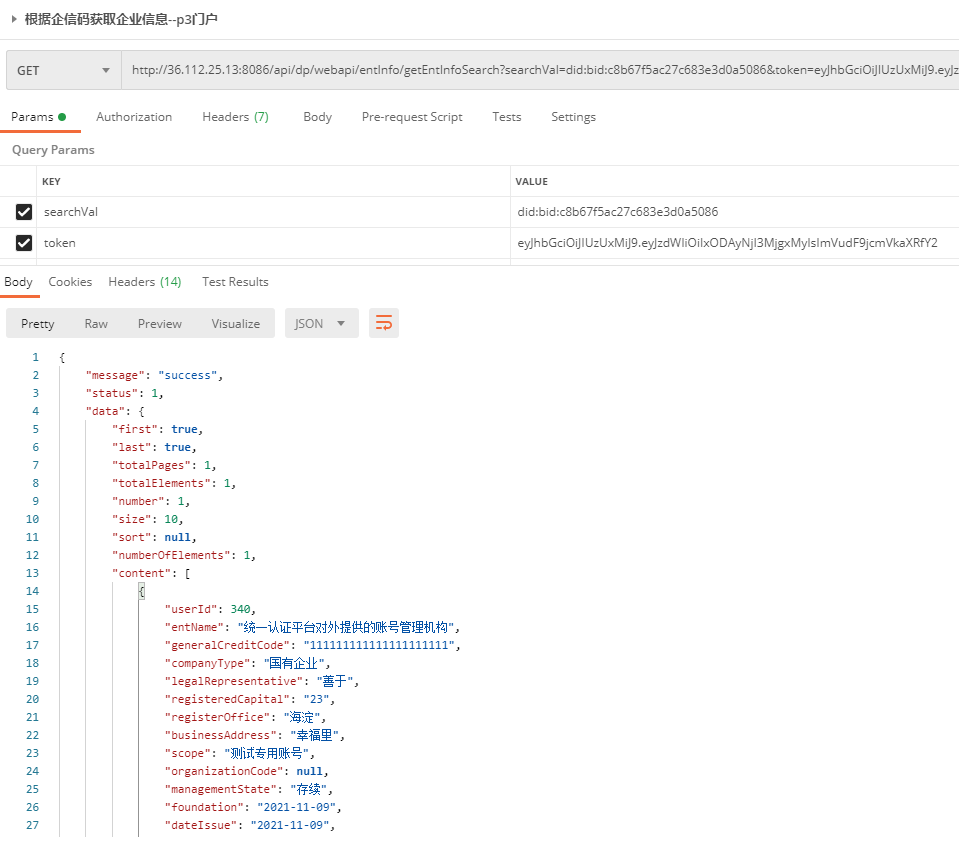
### 6.2.2根据企信码获取企业信息--p3门户

注意：此接口已经在sso-auth:1.0.5版本中移除，如需要请调用P3V2的接口地址。

http://ip:端口/api/dp/webapi/entInfo/getEntInfoSearch?searchVal=企信码&token=xxx

status为1则为成功返回，否则为-2则为系统异常。

特别注意： ip:端口为P3企信码管理平台的访问地址（测试环境36.112.25.13:8086）。



返回值说明:

|  |  |
| --- | --- |
| 字段 | 说明 |
| useId | 用户id |
| entName | 企业名称 |
| generalCreditCode | 社会信用码 |
| companyType | 企业类型 |
| legalRepresentative | 法定代表人 |
| registeredCaptial | 注册资本 |
| registerOffice | 等级机关 |
| businessAddress | 企业地址 |
| scope | 经营范围 |
| organizationCode | 组织机构代码 |
| managementState | 经营状态 |
| foundation | 成立日期 |
| dateIssue | 发照日期 |
| businessTerm | 营业期限 |
| entCreditCode | 企信码 |
| legalpIdno | 法人证件号 |
| provinceName | 省 |
| countyName | 市 |
| legalpIdimgFrontImg | 法人证件附件正面 |
| legalpIdimgReverseImg | 法人证件附件反面 |
| industry | 行业类型 |
| businessLicenseImg | 营业执照附件 |
| reWorkStatus | 复工状态 |
| qxmImg | 二维码路径 |
| auditor | 审核人 |
| entPhone | 企业电话 |
| creatTime | 创建时间 |
| email | 联系email |

### 6.2.3**获取当前登录者的企业资质信息--为灾备数据托管服务**

**请求地址：**[http://ip:port/api/dp/webapi/entInfo/entInfoForDh](http://localhost:8086/api/dp/webapi/entInfo/entInfoForDh)。**注意：ip:port为P3企信码后管台系统的请求路径**

**请求方式：g**et

**请求参数：**

将认证成功的token值存入http请求头的Cookie属性里，存放形式：**token=登录后后取到的token值。注意：Cookie必须携带token值，否则接口认证失败。**

**响应结果：**

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 说明 |
| message | 字符串 | 响应说明 |
| status | 整型 | 响应码： 1表示成功 -2:表示失败，在客户端应该退出登录，调用/logout接口 |
| data | List集合 | list |
| entName | 字符串 | 企业名称 |
| entGeneralCreditCode | 字符串 | 社会信用码 |
| companyType | 字符串 | 企业类型 |
| managementState | 字符串 | 经营状态 |
| foundation | 字符串 | 成立日期 |
| businessTerm | 字符串 | 营业期限 |
| dateIssue | 字符串 | 发照日期 |
| legalRepresentative | 字符串 | 法定代表人 |
| entPhone | 字符串 | 企业电话 |
| registeredCapital | 字符串 | 注册资金 |
| scope | 字符串 | 经营范围 |
| businessAddress | 字符串 | 企业地址 |
| workStatus | 字符串 | 复工状态 |
| industry | 字符串 | 行业类型 |
| entProvince | 字符串 | 所属省 |
| entCounty | 字符串 | 所属城市 |

**Postman对该接口的请求响应示例：**

