



一键取号接口文档

北京月付科技有限公司

2017年3月

修订

时间	版本号	维护人	维护内容
2017-03-21	V0.1	贾正	创建初稿

1.文档说明.....3

2.接入前准备.....3

3.接入流程.....4

4.合作方获取TOKEN8

5.取号.....9

6.合作方查询接口.....10

7.合作方客户端请求示例（javascript）12

8.合作方服务器端请求Token示例（JAVA）14

9.合作方服务器端请求Token示例（PYTHON）15

1.文档说明

本文档定义了取号服务通用接入标准，为实现合作方快速接入取号服务，其中包括接入流程，服务接口。

1.1文档编制目的

方便系统对接双方人员，清晰，快速理解接口的含义和接口对接的技术细节。

1.2术语

签名：客户端系统在调用服务端接口的时候，需要一个特殊的字段来识别传输数据的正确性和准确性，提高接口调用的安全性。

AppId：合作方唯一标识

Token：客户端系统在调用服务端接口时，需要验证请求来自于指定的客户端系统，提高接口调用的安全性。

reqId：合作方客户端请求唯一标识

timeStamp：时间戳

AES加密：对称加密算法

AESKey：AES加密解密使用的密钥

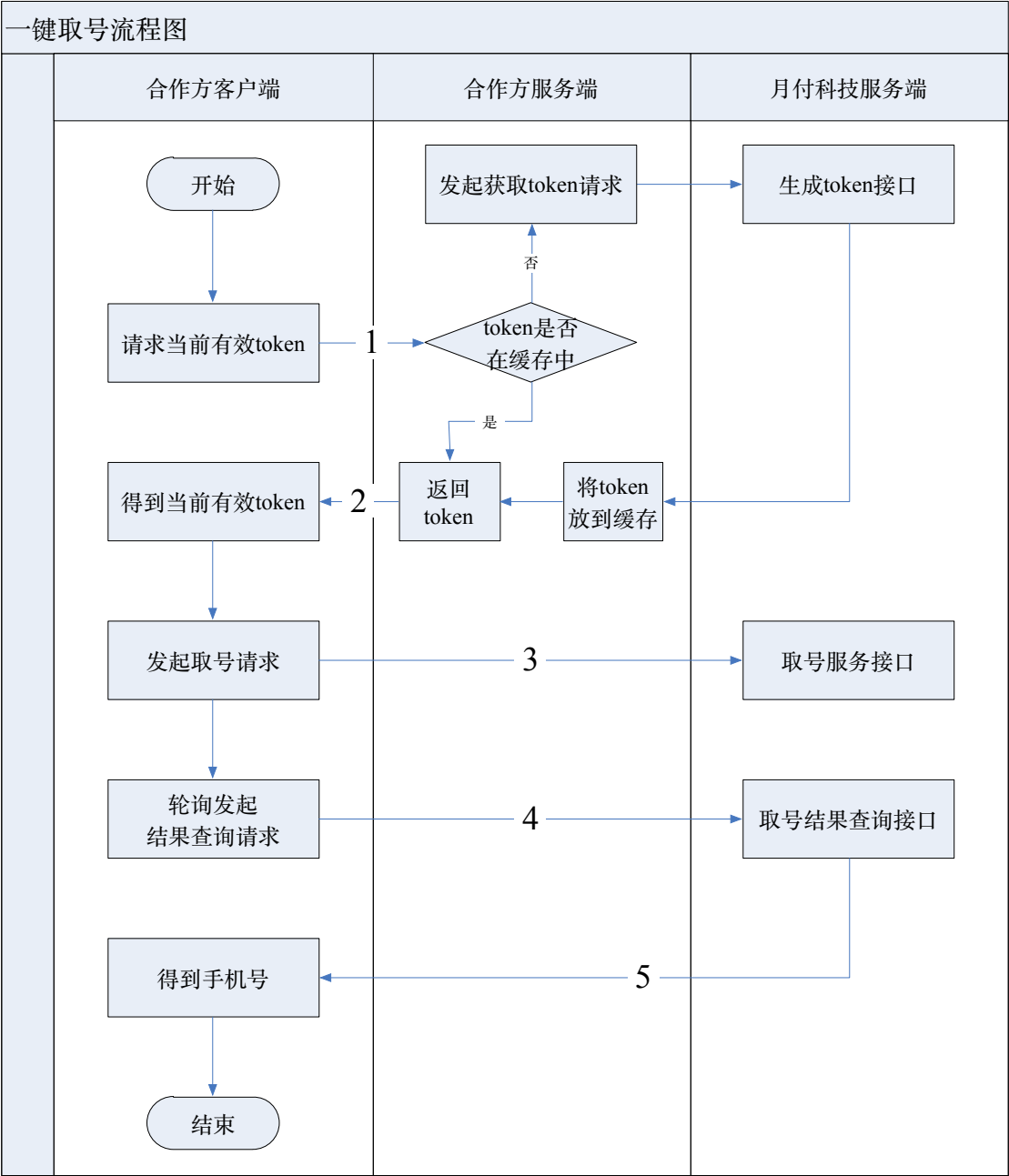
2.接入前准备

申请AppId：月付科技为合作方分配的一个唯一标识。

申请AESKey：月付科技为合作方分配的AES加解密密钥。

申请加入白名单：合作方需提供其获取token的服务器IP给月付科技，月付科技将该IP写入白名单。

3.接入流程



请求顺序	请求说明	说明及注意事项
1	请求token	合作方客户端向合作方服务器端请求token
i	异步请求token	合作方 服务器端 向月付服务端请求token (token值30分钟失效，需重发请求)
ii	返回token	月付服务端返回token
2	返回token	合作方 服务器端 对token、appId、时间戳、reqId进行aes加密，返回加密后的字符串
3	请求取号	合作方客户端调用月付取号接口，需携带sign和token， 需要使用jsonp接口
4	查询结果	合作方客户端调用月付查询接口，需携带sign和token， 需要使用jsonp接口
5	返回结果	月付服务端返回取号结果

注意：合作方客户端必须以4G网络访问取号接口

3.1 参数加密说明

包含参数：

数据字段	数据描述	数据格式	是否必填（Y/N）
appId	月付分配给合作方的接入ID	String	Y
reqId	合作方每次请求的唯一标识	String	Y
timeStamp	时间戳	String	Y
token	服务器验证标识	String	Y

1. 参数按ascii码排序

2. 对参数的json格式进行AES加密

3.2 加密demo（java）

```
// aes加解密密钥
String aesKey = "h1q9750o8141K864";
// appId 由月付分配
String appId = "wozhitest";
// 合作方请求唯一标识
String reqId = "5450312";
// 时间戳
String timeStamp = String.valueOf(new Date().getTime());
// token 从月付获取
String token = "49FF63FD025711D51DC744E836319D00";
// 准备生成签名(treemap 自动对参数排序)
TreeMap<String, String> param = new TreeMap<String, String>();
param.put("appId", appId);
param.put("reqId", reqId);
param.put("timeStamp", timeStamp);
param.put("token", token);
// 执行加密
String sign = simpleSign(param, aesKey);
```

```
/**
 * 加密
 * @param param
 * @return
 * @throws Exception
 */
private static String simpleSign (TreeMap<String, String> param,
    String aesKey) throws Exception{
    String paramStr = new Gson().toJson(param);
    String res = AesCharge.encryptToBase64(paramStr ,aesKey);
    return res;
}

public static String encryptToBase64(String data, String key){
    try {
        byte[] valueByte = encrypt(data.getBytes(CHAR_ENCODING),
            key.getBytes(CHAR_ENCODING));
        return new String(Base64.encode(valueByte));
    } catch (UnsupportedEncodingException e) {
        throw new RuntimeException("encrypt fail!", e);
    }
}

public static byte[] encrypt(byte[] data, byte[] key) {
    notEmpty(data, "data");
    notEmpty(key, "key");
    if(key.length!=16){
        throw new RuntimeException("Invalid AES key "
            + "length (must be 16 bytes)");
    }
    try {
        SecretKeySpec secretKey = new SecretKeySpec(key, "AES");
        byte[] enCodeFormat = secretKey.getEncoded();
        SecretKeySpec secKey =
            new SecretKeySpec(enCodeFormat,"AES");
        // 创建密码器
        Cipher cipher = Cipher.getInstance(AES_ALGORITHM);
        // 初始化
        cipher.init(Cipher.ENCRYPT_MODE, secKey);
        // 加密
        byte[] result = cipher.doFinal(data);
        return result;
    } catch (Exception e){
        throw new RuntimeException("encrypt fail!", e);
    }
}
```

4.合作方获取TOKEN

接口说明：合作方调用月付接口获取Token，需要在固定IP上发起此请求

调用方：合作方

被调用方：月付科技

调用地址：

正式地址：http://www.meiyuefu.com/getphone/token

测试地址：

传输协议：http post方式

调用参数：

数据字段	数据描述	数据格式	是否必填 (Y/N)
appId	月付分配给合作方的接入ID	String	Y

返回结果：json结果，结构如下

数据字段	数据描述	数据类型
resCode	成功或失败的code，"0"成功，其他失败	String
message	结果或失败具体原因	String
success	true/false	boolean

code映射表：

rescode	message
0	成功
1	用户IP不在白名单内
2	用户未开通功能
3	获取Token失败

返回样例:

成功:

```
{
  "resCode": "0", //
  "success": true,
  "message": "69B5B07FEBF98E77982D0C81FDD01F5E"
}
```

失败:

```
{
  "resCode": "1", //
  "message": "用户IP不在白名单内",
  "success": false
}
```

5.取号

接口说明: 合作方调用该接口, 获取手机号

调用方: 合作方

被调用方: 月付科技

调用地址:

正式地址: <http://www.meiyuefu.com/getphone>

测试地址:

传输协议: http get jsonp 方式

调用参数:

数据字段	数据描述	数据格式	是否必填 (Y/N)
appId	月付分配给合作方的接入ID	String	Y
sign	加密后的token等数据	String	Y

返回结果：需要合作方调用查询结果

6.合作方查询接口

接口说明：合作方调用该查询接口，返回手机号信息

调用方：合作方

被调用方：月付科技

调用地址：

正式地址： <http://www.meiyuefu.com/getphone/query>

测试地址：

传输协议：http get jsonp 方式

调用参数：

数据字段	数据描述	数据格式	是否必填 (Y/N)
appId	月付分配给合作公寓的接入ID	String	Y
sign	加密后的token等数据	String	Y

返回结果：json结果，结构如下

数据字段	数据描述	数据类型
resCode	成功或失败的code, "0"成功, 其他失败	String
message	失败具体原因	String
success	成功 true/失败 false	boolean
object	成功时返回结果的json数据	String

object结构

数据字段	数据描述	数据类型
------	------	------

phone	手机号	String
par	结果标识	String

code映射表：

code	msg
0	成功
1	缺少参数
2	token失效
3	获取结果失败

返回样例：

成功：

```
{
  "success": true,
  "resCode": "0",
  "object": {
    "par": "B0A63A589B1541DC9B3E4C8F25F073F7",
    "phone": "134****8682"
  }
}
```

失败：

```
{
  "success": false,
  "resCode": "1",
  "message": "缺少参数"
}
```

7.合作方客户端请求示例 (javascript)

7.1发起取号 (依赖jquery)

```
<script src="getPhoneNum.js"></script>
<script language="javascript">
var appId = "wozhitest";
var url = "http://quhao.meiyuefu.com/demo?appId="+appId;
$.get(url,function(obj){
    $.getPhoneNum({
        data:{
            sign:obj,
            appId:appId
        },
        success:function(data){
            $("#phone").text(data.object.phone)
        },
        error:function(data){
            $("#phone").text(data.message)
        }
    })
})
</script>
```

7.2 插件getPhoneNum.js（依赖jquery）

```
/*
author: caoke
time: 2017/3/20
说明: 通过4g网络获取用户的手机号码
demo:
$.getPhoneNum({
  //成功回调
  success: function(data){
    alert(JSON.stringify(data))
  },
  //网络错误及失败回调
  error: function(data){
    alert(JSON.stringify(data))
  }
})
*/
$.extend({
  getPhoneNum: function(config){
    var config = $.extend({success: function(){}}, error: function(){}}, config);
    var sign = config.data.sign;
    var appId = config.data.appId;
    //把客户端信息发送过去
    $.ajax({
      url: "http://www.meiyuefu.com/getphone",
      data: {
        appId: appId,
        sign: sign
      },
      dataType: "jsonp",
      complete: function () {
        _sync(5)
      }
    })
    //轮询结果，成功返回successCallback，失败继续轮询5次
    function _sync(num){
      setTimeout(function(){
        $.ajax({
          url: "http://www.meiyuefu.com/getphone/query",
          dataType: "jsonp",
          data: {
            appId: appId,
            sign: sign
          },
          success: function(data){
            if (data.success) {
              config.success(data)
            } else {
              if(num > 0){
                _sync(--num)
              } else {
                config.error(data)
              }
            }
          },
          error: config.error
        })
      }, 1000)
    }
  }
})
```

8.合作方服务器端请求Token示例（JAVA）

```
@Path("quhao.meiyuefu.com/demo")
public ActionResult test() throws Exception {
    // appId 由月付分配
    String appId = "wozhitest";
    // 合作方请求唯一标识
    String reqId = "5450312";
    // 时间戳
    String timeStamp = String.valueOf(new Date().getTime());
    // aes加解密密钥
    String aesKey = "h1q9750o8141K864";
    // 月付科技获取token的url
    String url = "http://meiyuefu.com/getphone/token";
    // 准备生成签名
    TreeMap<String, String> param = new TreeMap<String, String>();
    param.put("appId", appId);
    param.put("reqId", reqId);
    param.put("timeStamp", timeStamp);
    // 获取token
    String redisKey = "meiyuefu::access_token_key";
    String token = RedisInstance.get().get(redisKey);
    // 如果缓存中存在, 则直接返回tokens
    if(StringUtils.isNotBlank(token)){
        param.put("token", token);
    }else{
        Map<String, Object> httpParam =
            new TreeMap<String, Object>();
        httpParam.put("appId", appId);
        String res = HttpClientUtil.
            postHtmlForm(url, "utf-8", httpParam);
        logger.info("test res=" + res);
        ResultInfo resInfo = new Gson().fromJson(res, ResultInfo.class);
        param.put("token", resInfo.getMessage());
        if(StringUtils.isNotBlank(resInfo.getMessage())
            && resInfo.isSuccess()){
            // 存入缓存
            RedisInstance.get().set(redisKey, resInfo.getMessage());
            RedisInstance.get().expire(redisKey, 30 * 60);
        }
    }
    logger.info("test param=" + new Gson().toJson(param));
    String localSign = simpleSign(param, aesKey);
    logger.info("test localSign=" + localSign);
    return new JsonResult(localSign);
}
```

9.合作方服务器端请求Token示例（PYTHON）

```
#!/usr/bin/env python3
#coding: utf-8

import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.image import MIMEImage
import time
import datetime
import os
import sys
import MySQLdb
import hashlib
import urllib
import urllib2
import redis
import json
#####
m = hashlib.md5()
#####
#####
def requestUrl():
    ISOTIMEFORMAT = '%Y-%m-%d %H:%M:%S'
    currentTime = time.strftime( ISOTIMEFORMAT, time.localtime() )
    print currentTime
    url = "http://www.meiyuefu.com/getphone/token?apld=wozhitest"
    req = urllib2.Request(url)
    print req
    res_data = urllib2.urlopen(req)
    res = res_data.read()
    try:
        obj = json.loads(res)
        token = obj['message']
        if token != "":
            storage(token)
    except Exception,ex:
        print ex
    print url + ", " + res

def storage(token):
    #####
    new_r = redis.Redis("101.200.54.50", 6379, password="myuefu001#")
    #####
    key = "meiyuefu::access_token_key"
    new_r.set(key, token)
    new_r.expire(key, 30 * 60)
    #####

if __name__ == "__main__":
    requestUrl()
```