验证用户登录

v3/user/is\_login

## 1 功能说明

验证用户的登录态，判断openkey是否过期，没有过期则对openkey有效期进行续期（一次调用续期2小时）。

本接口的调用场景：  
（1） 用来验证用户登录态。  
验证登录态的必要性详见这里：[技术优化原则#1.3 需要考虑对登录态做校验](http://wiki.open.qq.com/wiki/%E6%8A%80%E6%9C%AF%E4%BC%98%E5%8C%96%E5%8E%9F%E5%88%99#1.3_.E9.9C.80.E8.A6.81.E8.80.83.E8.99.91.E5.AF.B9.E7.99.BB.E5.BD.95.E6.80.81.E5.81.9A.E6.A0.A1.E9.AA.8C)；  
（注：如果是多区多服应用，除调用本接口进行登录态验证外，还需要调用[v3/user/is\_area\_login](http://wiki.open.qq.com/wiki/v3/user/is_area_login)接口验证是否从选区页面进入应用）。

（2） 用来对openkey进行续期。  
用户登录平台后进入应用时，URL中会带有该用户的OpenID和openkey，该openkey具有2小时的有效期。如果用户在应用中一直在操作，但是2小时内没有触发OpenAPI的调用，则会导致openkey过期。  
因此开发者需要调用本接口来对openkey进行续期，一次调用续期2小时。本接口无调用次数限制。  
注：1.如果由于平台统一刷新登录态导致续期机制失效，开发者可调用[fusion2.dialog.relogin](http://wiki.open.qq.com/wiki/fusion2.dialog.relogin)接口弹出登录弹框让用户登录，重新获得openkey。   
2.openkey的最大可续期间隔为2小时，最大有效期是12-24小时，每天的早晚8点为集中过期时间。集中过期规则如下：以早晚8点为参考时间点，超过12小时的openkey全部失效。如，今天上午10点拿到的openkey，到今天晚上8点第一个集中过期时间点判断时，因不足12小时故不会过期；而到明天上午8点第二个集中过期时间点判断时，已超过12小时，因此从这个时刻起就过期了。

## 2 接口调用说明

### 2.1 URL

http://[域名]/v3/user/is\_login   
  
正式环境域名或测试环境IP详见：[API3.0文档#请求URL说明](http://wiki.open.qq.com/wiki/API3.0%E6%96%87%E6%A1%A3#.E8.AF.B7.E6.B1.82URL.E8.AF.B4.E6.98.8E)。

### 2.2 格式

json

### 2.3 HTTP请求方式

GET, POST

### 2.4 IP限制

TRUE

### 2.5 输入参数说明

各个参数请进行URL 编码，编码时请遵守 [RFC 1738](http://tools.ietf.org/html/rfc1738)

（1）公共参数  
发送请求时必须传入公共参数，详见[公共参数说明](http://wiki.open.qq.com/wiki/API3.0%E6%96%87%E6%A1%A3#.E5.85.AC.E5.85.B1.E5.8F.82.E6.95.B0.E8.AF.B4.E6.98.8E)。

（2）私有参数  
无。

### 2.6 请求示例

http://113.108.20.23/v3/user/is\_login?  
openid=B624064BA065E01CB73F835017FE96FA&  
openkey=5F154D7D2751AEDC8527269006F290F70297B7E54667536C&  
appid=2&  
sig=9999b41ad0b688530bb1b21c5957391c&  
pf=qzone&  
format=json&  
userip=112.90.139.30

### 2.7 返回参数说明

|  |  |
| --- | --- |
| **参数名称** | **描述** |
| **ret** | 返回码。详见[公共返回码说明#OpenAPI V3.0 返回码](http://wiki.open.qq.com/wiki/%E5%85%AC%E5%85%B1%E8%BF%94%E5%9B%9E%E7%A0%81%E8%AF%B4%E6%98%8E#OpenAPI_V3.0_.E8.BF.94.E5.9B.9E.E7.A0.81)。 |
| **msg** | 如果错误，返回错误信息。 |

### 2.8 正确返回示例

JSON示例:

Content-type: text/html; charset=utf-8  
{  
"ret":0,  
"msg":"用户已登录"  
}

### 2.9 错误返回示例

Content-type: text/html; charset=utf-8  
{  
"ret":1002,  
"msg":"用户没有登录态"  
}

您还没有登录，[立即登录](javascript:void(0);)

===============================================================================

## v3/user/get\_info

|  |
| --- |
| 目录 [[隐藏](javascript:toggleToc())]   * [What's New?](http://wiki.open.qq.com/wiki/v3/user/get_info#What.27s_New.3F) * [1 功能说明](http://wiki.open.qq.com/wiki/v3/user/get_info#1_.E5.8A.9F.E8.83.BD.E8.AF.B4.E6.98.8E) * [2 接口调用说明](http://wiki.open.qq.com/wiki/v3/user/get_info#2_.E6.8E.A5.E5.8F.A3.E8.B0.83.E7.94.A8.E8.AF.B4.E6.98.8E)   + [2.1 URL](http://wiki.open.qq.com/wiki/v3/user/get_info#2.1.09URL)   + [2.2 格式](http://wiki.open.qq.com/wiki/v3/user/get_info#2.2.09.E6.A0.BC.E5.BC.8F)   + [2.3 HTTP请求方式](http://wiki.open.qq.com/wiki/v3/user/get_info#2.3.09HTTP.E8.AF.B7.E6.B1.82.E6.96.B9.E5.BC.8F)   + [2.4 IP限制](http://wiki.open.qq.com/wiki/v3/user/get_info#2.4.09IP.E9.99.90.E5.88.B6)   + [2.5 输入参数说明](http://wiki.open.qq.com/wiki/v3/user/get_info#2.5.09.E8.BE.93.E5.85.A5.E5.8F.82.E6.95.B0.E8.AF.B4.E6.98.8E)   + [2.6 请求示例](http://wiki.open.qq.com/wiki/v3/user/get_info#2.6.09.E8.AF.B7.E6.B1.82.E7.A4.BA.E4.BE.8B)   + [2.7 返回参数说明](http://wiki.open.qq.com/wiki/v3/user/get_info#2.7.09.E8.BF.94.E5.9B.9E.E5.8F.82.E6.95.B0.E8.AF.B4.E6.98.8E)   + [2.8 错误返回码说明](http://wiki.open.qq.com/wiki/v3/user/get_info#2.8_.E9.94.99.E8.AF.AF.E8.BF.94.E5.9B.9E.E7.A0.81.E8.AF.B4.E6.98.8E)   + [2.9 正确返回示例](http://wiki.open.qq.com/wiki/v3/user/get_info#2.9.09.E6.AD.A3.E7.A1.AE.E8.BF.94.E5.9B.9E.E7.A4.BA.E4.BE.8B)   + [2.10 错误返回示例](http://wiki.open.qq.com/wiki/v3/user/get_info#2.10.09.E9.94.99.E8.AF.AF.E8.BF.94.E5.9B.9E.E7.A4.BA.E4.BE.8B) * [3 示例代码](http://wiki.open.qq.com/wiki/v3/user/get_info#3_.E7.A4.BA.E4.BE.8B.E4.BB.A3.E7.A0.81) * [4 接口调试](http://wiki.open.qq.com/wiki/v3/user/get_info#4_.E6.8E.A5.E5.8F.A3.E8.B0.83.E8.AF.95) |

## What's New?

2012年12月07日，本接口新增返回参数“is\_yellow\_high\_vip”，用于判断用户是否为豪华版黄钻用户。

## 1 功能说明

获取登录用户的信息，包括昵称、头像、性别等信息。  
本接口是全平台通用的，即发送请求后，可根据请求中传入的“pf”平台参数返回对应平台的用户信息，详见返回字段说明。  
例如：如果传入的pf为qzone，则返回的是其QQ空间的昵称和头像。

注意：  
1. 本接口返回的各种VIP（例如黄钻等）信息是经过缓存的，有一定的延时。  
如果需要VIP信息特别准确的场景（例如黄钻每日礼包场景中，非黄钻用户开通黄钻后，返回应用应该立即可领取礼包），请调用专门的VIP实时信息获取接口。  
目前为黄钻提供专门的黄钻实时信息获取接口：[v3/user/is\_vip](http://wiki.open.qq.com/wiki/v3/user/is_vip)，其它VIP实时信息获取接口为：[v3/user/total\_vip\_info](http://wiki.open.qq.com/wiki/v3/user/total_vip_info)。   
2. 本接口只返回用户基本个人资料。对于其它更丰富的用户个人资料，出于保护用户隐私的考虑，目前尚不开放。

## 2 接口调用说明

### 2.1 URL

http://[域名]/v3/user/get\_info   
  
正式环境域名或测试环境IP详见：[API3.0文档#请求URL说明](http://wiki.open.qq.com/wiki/API3.0%E6%96%87%E6%A1%A3#.E8.AF.B7.E6.B1.82URL.E8.AF.B4.E6.98.8E)。

### 2.2 格式

json

### 2.3 HTTP请求方式

GET, POST

### 2.4 IP限制

TRUE

### 2.5 输入参数说明

各个参数请进行URL 编码，编码时请遵守 [RFC 1738](http://tools.ietf.org/html/rfc1738)

（1）公共参数  
发送请求时必须传入公共参数，详见[公共参数说明](http://wiki.open.qq.com/wiki/API3.0%E6%96%87%E6%A1%A3#.E5.85.AC.E5.85.B1.E5.8F.82.E6.95.B0.E8.AF.B4.E6.98.8E)。

（2）私有参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **是否必须** | **类型** | **描述** |
| **charset** |  | string | 指定请求及响应的字符集，取值为gbk或utf-8（只有pf=qqgame或pf=3366时，可以输入该参数）。  默认值为utf-8，其他非法取值也认为是utf-8。 |
| **flag** |  | unsigned int | pf=qqgame时，必须输入该参数，指定需要获取QQGame中的哪些信息：  1：需要获取游戏昵称和性别； 2：需要获取蓝钻等级； 3：需要获取昵称和蓝钻等级； 4：需要获取照片秀标志。 |

### 2.6 请求示例

http://openapi.tencentyun.com/v3/user/get\_info?  
openid=B624064BA065E01CB73F835017FE96FA&  
openkey=5F154D7D2751AEDC8527269006F290F70297B7E54667536C&  
appid=2&  
sig=VrN%2BTn5J%2Fg4IIo0egUdxq6%2B0otk%3D&  
pf=qzone&  
format=json&  
userip=112.90.139.30

### 2.7 返回参数说明

|  |  |
| --- | --- |
| **参数名称** | **描述** |
| **ret** | 返回码。详见[公共返回码说明#OpenAPI V3.0 返回码](http://wiki.open.qq.com/wiki/%E5%85%AC%E5%85%B1%E8%BF%94%E5%9B%9E%E7%A0%81%E8%AF%B4%E6%98%8E#OpenAPI_V3.0_.E8.BF.94.E5.9B.9E.E7.A0.81)。 |
| **msg** | 如果错误，返回错误信息。 |
| **is\_lost** | 判断是否有数据丢失。如果应用不使用cache，不需要关心此参数。  0或者不返回：没有数据丢失，可以缓存。 1：有部分数据丢失或错误，不要缓存。 |
| **nickname** | 昵称。 |
| **gender** | 性别。 |
| **country** | 国家（当pf=qzone、pengyou或qplus时返回）。 |
| **province** | 省（当pf=qzone、pengyou或qplus时返回）。 |
| **city** | 市（当pf=qzone、pengyou或qplus时返回）。 |
| **figureurl** | 头像URL。详见：[前端页面规范#6. 关于用户头像的获取和尺寸说明](http://wiki.open.qq.com/wiki/%E5%89%8D%E7%AB%AF%E9%A1%B5%E9%9D%A2%E8%A7%84%E8%8C%83#6._.E5.85.B3.E4.BA.8E.E7.94.A8.E6.88.B7.E5.A4.B4.E5.83.8F.E7.9A.84.E8.8E.B7.E5.8F.96.E5.92.8C.E5.B0.BA.E5.AF.B8.E8.AF.B4.E6.98.8E)。 |
| **openid** | 用户QQ号码转化得到的ID（当pf=qplus时返回）。 |
| **qq\_level** | 用户QQ等级（当pf=qplus时返回）。 |
| **qq\_vip\_level** | 用户QQ会员等级（当pf=qplus时返回）。 |
| **qplus\_level** | 用户Q+等级（当pf=qplus时返回）。 |
| **is\_yellow\_vip** | 是否为黄钻用户（0：不是； 1：是）。  （当pf=qzone、pengyou或qplus时返回） |
| **is\_yellow\_year\_vip** | 是否为年费黄钻用户（0：不是； 1：是）。  （当pf=qzone、pengyou或qplus时返回） |
| **yellow\_vip\_level** | 黄钻等级，目前最高级别为黄钻8级（如果是黄钻用户才返回此参数）。  （当pf=qzone、pengyou或qplus时返回） |
| **is\_yellow\_high\_vip** | 是否为豪华版黄钻用户（0：不是； 1：是）。  （当pf=qzone、pengyou或qplus时返回） |
| **is\_blue\_vip** | 是否为蓝钻用户（0：不是； 1：是）。  （当pf=qqgame或3366时返回） |
| **is\_blue\_year\_vip** | 是否为年费蓝钻用户（0：不是； 1：是）。  （当pf=qqgame或3366时返回） |
| **blue\_vip\_level** | 蓝钻等级（如果是蓝钻用户才返回此参数）。  （当pf=qqgame或3366时返回） |
| **3366\_level** | 3366用户的大等级。  （当pf=3366时返回） |
| **3366\_level\_name** | 3366用户的等级名，如小游游、小游仙。  （当pf=3366时返回） |
| **3366\_grow\_level** | 3366用户的成长等级。  （当pf=3366时返回） |
| **3366\_grow\_value** | 3366用户的成长值。  （当pf=3366时返回） |
| **is\_super\_blue\_vip** | 是否是豪华蓝钻。  （当pf=qqgame或3366时返回） |

### 2.8 错误返回码说明

公共错误返回码：[公共返回码说明#OpenAPI V3.0 返回码](http://wiki.open.qq.com/wiki/%E5%85%AC%E5%85%B1%E8%BF%94%E5%9B%9E%E7%A0%81%E8%AF%B4%E6%98%8E#OpenAPI_V3.0_.E8.BF.94.E5.9B.9E.E7.A0.81)。   
本接口私有错误返回码：暂无。

### 2.9 正确返回示例

JSON示例:

Content-type: text/html; charset=utf-8  
{  
"ret":0,  
"is\_lost":0,  
"nickname":"Peter",  
"gender":"男",  
"country":"中国",  
"province":"广东",  
"city":"深圳",  
"figureurl":"http://imgcache.qq.com/qzone\_v4/client/userinfo\_icon/1236153759.gif",  
"is\_yellow\_vip":1,  
"is\_yellow\_year\_vip":1,  
"yellow\_vip\_level":7,  
"is\_yellow\_high\_vip": 0  
}

### 2.10 错误返回示例

Content-type: text/html; charset=utf-8  
{  
"ret":1002,  
"msg":"请先登录"  
}

## 3 示例代码

您可以直接下载并使用腾讯开放平台提供的SDK，并参考SDK里面的给出的示例代码进行接口调用。  
详见：[SDK下载#OpenAPI V3.0 SDK下载](http://wiki.open.qq.com/wiki/SDK%E4%B8%8B%E8%BD%BD#OpenAPI_V3.0_SDK.E4.B8.8B.E8.BD.BD)。

## 4 接口调试

您可以使用腾讯开放平台提供的API联调工具进行接口调试。  
请登录：<http://open.qq.com/tools>

您还没有登录，[立即登录](javascript:void(0);)

以上信息是否解决您的问题？

===============================================================================

### 1.3 需要考虑对登录态做校验

**进行登录态校验的必要性：**  
如果应用的代码中没有登录态验证的逻辑：  
-由于调用平台的任何后台接口时，平台都会对登录态进行验证，因此不存在用户信息泄漏的问题；   
-但是如果应用不调用腾讯平台的任何后台接口（虽然这种情况很少见），则应用本身存在安全隐患。攻击者只要知道用户的OpenID就可以登录该用户添加的任何一个应用，并且以主人态进行操作，攻击者可以利用该漏洞发送垃圾信息。  
-如果用户在应用中进行某些需要调用前台接口的操作（例如分享）时，接口判断其登录态失效，会直接返回登录态失效的提示语，这对于用户的体验很不好。因此强烈建议应用程序中包含有登录态校验逻辑，在发现其登录态失效时，及时弹出登录窗口提示用户登录。   
  
**进行登录态校验的方法：**  
用户进入应用时，URL中会带有该用户的OpenID和openkey，这时应用只要调用腾讯的任意后台API，都可以验证用户的登录态是否合法。  
但是如果仅仅为了出于验证登录态 目的而去随意调用某一个OpenAPI，则会给后台服务造成很大的压力 。  
因此更推荐应用在用户进入应用时通过以下方式进行登录态校验：  
（1） 方式1：调用[v3/user/get\_info](http://wiki.open.qq.com/wiki/v3/user/get_info) 接口来验证登录态。该接口即可验证登录态又可以获取登录用户的信息，1个应用一般都需要获取登录用户的信息，因此调用本接口可一举两得。  
（2）方式2：调用专门的登录态校验接口[v3/user/is\_login](http://wiki.open.qq.com/wiki/v3/user/is_login)。

**进行登录态校验的逻辑说明 ：**  
（1）用户进入应用，应用获得用户的openid，openkey。  
（2）应用按照上述方法，进行用户的登录态校验。  
（3）如果登录态失效，则弹出登录弹框让用户登录（需调用[fusion2.dialog.relogin](http://wiki.open.qq.com/wiki/fusion2.dialog.relogin)接口）；  
（4）如果登录态有效，应用生成自己的登录态信息写到cookie中。应用自己的登录态可以是用户的openid+timestamp+应用希望的信息的加密串，请千万不要写到qzoneapp.com下，请写到自己的域名下，保证各应用之间不会冲突。  
（5）之后所有的应用逻辑都只需要验证cookie中应用自己的登录态；调用平台的接口时还是验证openid，openkey。

### 1.4 需要考虑100-continue问题

在使用CURL做POST的时候，CURL并不会直接就发起POST请求，而是会分为2步：  
（1）发送一个请求，请求头部包含一个“Expect:100-continue”，询问httpserver是否愿意接受数据；  
（2）接收到Server返回的100-continue应答以后，把数据POST给httpserver。   
（CURL的行为详见RFC2616中的相关描述：<http://www.w3.org/Protocols/rfc2616/rfc2616-sec8.html#sec8.2.3> ）   
  
腾讯开放平台用来响应OpenAPI请求的httpserver出于性能上的考虑，不支持对100-continue的应答。需要在应用程序中进行处理，否则会导致请求发送失败，调用OpenAPI出错。   
腾讯开放平台提供的官方SDK中已经增加了相关处理，因此不会出现该问题。  
如果是开发者自己封装SDK，则需要在调用OpenAPI时增加对请求头部中“Expect:100-continue”的处理。以PHP为例：

curl\_setopt($ch, CURLOPT\_HTTPHEADER, array('Expect:'));

===============================================================================

## OpenAPI V3.0调用说明

### 请求URL说明

http://[域名]/[api\_name]

**域名和IP的说明：**  
正式环境下使用域名：openapi.tencentyun.com。测试环境下使用IP：119.147.19.43。  
使用CEE进行程序部署的应用，测试环境请使用虚拟IP：1.254.254.22。

注意：  
1. 原先已经接入的应用如果使用的是正式环境IP，建议改为域名访问方式，因为：  
（1）域名访问方式在调用用户信息以及关系链接口时平均访问速度更快，主要接口不受公网波动的影响。  
（2）域名访问方式扩展性更强。后续CEE支持域名访问后，则无论哪种部署方式，都可以使用域名访问，且无需再关注IP变更。  
2. 使用CEE进行程序部署的应用测试环境IP也与上述不同，请使用虚拟IP：1.254.254.22。正式环境域名与上面相同。  
3. 应用部署前必须确保已经将程序中的测试IP换成了正式域名，才能将应用上线到正式环境。  
4. 测试环境IP仅针对调试者QQ号有用，即只有调试者QQ号对应的OpenID会通过验证，非调试者QQ号将会返回-64的错误。调试者QQ号的设置详见[这里](http://wiki.open.qq.com/wiki/%E5%BA%94%E7%94%A8%E6%B5%8B%E8%AF%95%E7%8E%AF%E5%A2%83%E8%AF%B4%E6%98%8E#3.1_.E8.AE.BE.E7.BD.AE.E8.B0.83.E8.AF.95.E8.80.85QQ.E5.8F.B7)。

**api\_name的说明：**  
[api\_name]：见[API列表](http://wiki.open.qq.com/wiki/API%E5%88%97%E8%A1%A8)中带API_legend_3.png 标记的接口。例如接口名称为：v3/user/get\_info。   
  
**请求URL的示例：**  
1. 正式环境下访问OpenAPI V3.0：  
http://openapi.tencentyun.com/v3/user/get\_info   
2. 测试环境下访问OpenAPI V3.0：  
http://119.147.19.43/v3/user/get\_info

### OpenAPI超时时间说明

OpenAPI一般会在50ms以内返回数据，OpenAPI接口机设置的最长超时时间为3s。  
开发者可以根据上述说明自行设置OpenAPI调用的超时时间。

### 公共参数说明

注意：  
1. 所有参数都需进行URL 编码，编码时请遵守 [RFC 1738](http://tools.ietf.org/html/rfc1738)。  
2. 下列部分参数是由平台直接传给应用的，应用原样传给平台即可，即跳转到应用首页后的URL会带该参数，一定会带的参数有：openid,openkey,pf,pfkey，根据场景不同可能会传的参数有：invkey,iopenid,itime,source,app\_custom。   
以上参数已经被腾讯的统计后台占用，应用不能再用作自定义参数。

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **是否必须** | **类型** | **描述** |
| **openid** | 必须 | string | 与APP通信的用户key。从平台跳转到应用时会调用应用的CanvasURL，平台会在CanvasURL后带上本参数。由平台直接传给应用，应用原样传给平台即可。  根据APPID以及QQ号码生成，即不同的appid下，同一个QQ号生成的OpenID是不一样的。 注： （1）如果不知道如何解析CanvasURL以获取OpenID和Openkey，点击[这里](http://wiki.open.qq.com/wiki/OpenAPI%E8%B0%83%E7%94%A8%E7%9B%B8%E5%85%B3%E9%97%AE%E9%A2%98#4._.E5.A6.82.E4.BD.95.E8.8E.B7.E5.8F.96OpenID.E5.92.8COpenkey.E7.AD.89.E5.8F.82.E6.95.B0.EF.BC.9F)； （2）如果解析CanvasURL时获取不到OpenID和Openkey等参数，点击[这里](http://wiki.open.qq.com/wiki/OpenAPI%E8%B0%83%E7%94%A8%E7%9B%B8%E5%85%B3%E9%97%AE%E9%A2%98#5._.E4.B8.BA.E4.BD.95.E8.8E.B7.E5.8F.96.E4.B8.8D.E5.88.B0OpenID.E5.92.8COpenkey.E7.AD.89.E5.8F.82.E6.95.B0.EF.BC.9F)。 |
| **openkey** | 必须 | string | session key。从平台跳转到应用时会调用应用的CanvasURL，平台会在CanvasURL后带上本参数。由平台直接传给应用，应用原样传给平台即可。  注： （1）如果不知道如何解析CanvasURL以获取OpenID和Openkey，点击[这里](http://wiki.open.qq.com/wiki/OpenAPI%E8%B0%83%E7%94%A8%E7%9B%B8%E5%85%B3%E9%97%AE%E9%A2%98#4._.E5.A6.82.E4.BD.95.E8.8E.B7.E5.8F.96OpenID.E5.92.8COpenkey.E7.AD.89.E5.8F.82.E6.95.B0.EF.BC.9F)； （2）如果解析CanvasURL时获取不到OpenID和Openkey等参数，点击[这里](http://wiki.open.qq.com/wiki/OpenAPI%E8%B0%83%E7%94%A8%E7%9B%B8%E5%85%B3%E9%97%AE%E9%A2%98#5._.E4.B8.BA.E4.BD.95.E8.8E.B7.E5.8F.96.E4.B8.8D.E5.88.B0OpenID.E5.92.8COpenkey.E7.AD.89.E5.8F.82.E6.95.B0.EF.BC.9F)； （3）注意同一个用户如果在不同时间打开多个应用页面，页面返回的openkey是不一样的，这些openkey在各自的页面都可用。但是不要在当前页面使用另1个页面的openkey，否则会出错； （4）Openkey长度为不固定的字符串，不能为空。建议开发者不要检查openkey的长度，也不要在后台存储openkey，否则可能会导致用户无法登录; （5）openkey过期时间为两小时，需要用[v3/user/is\_login](http://wiki.open.qq.com/wiki/v3/user/is_login)接口续期。 |
| **appid** | 必须 | unsigned int | 应用的唯一ID。可以通过appid查找APP基本信息。 |
| **sig** | 必须 | string | 请求串的签名，以appkey作为密钥，具体签名算法见[腾讯开放平台第三方应用签名参数sig的说明](http://wiki.open.qq.com/wiki/%E8%85%BE%E8%AE%AF%E5%BC%80%E6%94%BE%E5%B9%B3%E5%8F%B0%E7%AC%AC%E4%B8%89%E6%96%B9%E5%BA%94%E7%94%A8%E7%AD%BE%E5%90%8D%E5%8F%82%E6%95%B0sig%E7%9A%84%E8%AF%B4%E6%98%8E)。 |
| **pf** | 必须 | string | 应用的来源平台。从平台跳转到应用时会调用应用的CanvasURL，平台会在CanvasURL后带上本参数。由平台直接传给应用，应用原样传给平台即可。  注： （1）由平台直接传给应用，应用原样传给平台即可。请不要应用手动构造pf值，也不需要对平台传递的pf进行有效性校验，以避免应用上线多平台时需要付出额外的修改成本，导致调用某些接口时由于参数需要根据pf值生成导致传入参数不合法。 （2）如果不知道如何解析CanvasURL以获取pf，点击[这里](http://wiki.open.qq.com/wiki/OpenAPI%E8%B0%83%E7%94%A8%E7%9B%B8%E5%85%B3%E9%97%AE%E9%A2%98#4._.E5.A6.82.E4.BD.95.E8.8E.B7.E5.8F.96OpenID.E5.92.8COpenkey.E7.AD.89.E5.8F.82.E6.95.B0.EF.BC.9F)； （3）如果解析CanvasURL时获取不到pf，点击[这里](http://wiki.open.qq.com/wiki/OpenAPI%E8%B0%83%E7%94%A8%E7%9B%B8%E5%85%B3%E9%97%AE%E9%A2%98#5._.E4.B8.BA.E4.BD.95.E8.8E.B7.E5.8F.96.E4.B8.8D.E5.88.B0OpenID.E5.92.8COpenkey.E7.AD.89.E5.8F.82.E6.95.B0.EF.BC.9F)； （4）建议应用不要在程序中对pf进行转换（例如将字符型转换为int型），避免对于值不固定的pf（例如漫游或游戏联盟的pf）支持造成障碍。 |
| **format** |  | string | 定义API返回的数据格式。  取值说明：为xml时表示返回的格式是xml；为json时表示返回的格式是json。 注意：json、xml为小写，否则将不识别。format不传或非xml，则返回json格式数据。 |
| **userip** |  | string | 用户的IP。  应用服务器得到的remote\_ip和x\_forward\_ip是10.x.x.x，由上一层腾讯路由转发。 http请求的头里有一个QVia（HTTP\_QVIA值），是个40个字节的串，前8个字节是IP，后面的是校验，将字节串前8个字节，分成4段，分别由16进制转成10进制，即可得到用户最终IP。 如果应用（例如新接入的应用、已经接入TGW的应用、non-hosting应用）获取不到QVia的值，可以直接通过标准http协议获取用户IP。 |

### 公共返回参数说明

公共返回参数如下，其余返回参数由各个API自行定义，请参考各OpenAPI的说明。

|  |  |
| --- | --- |
| **参数名称** | **描述** |
| **ret** | 返回码。具体返回码含义详见下文。 |
| **msg** | 如果错误，返回错误信息。 |
| **is\_lost** | 数据是否丢失，如果应用不考虑cache可以完全不关心。  0或者不返回：完全没有丢失，可以缓存。 1：有一部分数据错误，不要缓存 |

### 公共返回码说明

详见：[公共返回码说明#OpenAPI V3.0 返回码](http://wiki.open.qq.com/wiki/%E5%85%AC%E5%85%B1%E8%BF%94%E5%9B%9E%E7%A0%81%E8%AF%B4%E6%98%8E#OpenAPI_V3.0_.E8.BF.94.E5.9B.9E.E7.A0.81)。

## OpenAPI V3.0 SDK下载

详见：[SDK下载#OpenAPI V3.0 SDK下载](http://wiki.open.qq.com/wiki/SDK%E4%B8%8B%E8%BD%BD#OpenAPI_V3.0_SDK.E4.B8.8B.E8.BD.BD)。

您还没有登录，[立即登录](javascript:void(0);)

## 签名生成总体说明

本文档仅适用于OpenAPI V3.0接口的签名生成，由于是通用说明，本文中仅以/v3/user/get\_info的签名生成作为示例。  
  
签名值**sig**是将请求源串以及密钥根据一定签名方法生成的签名值，用来提高传输过程参数的防篡改性。  
签名值的生成共有3个步骤：**构造源串，构造密钥，生成签名值**。详见下面的描述。

### Step 1. 构造源串

**源串是由3部分内容用“&”拼接起来的：** HTTP请求方式 & urlencode(uri) & urlencode(a=x&b=y&...)

**源串构造步骤如下：**

第1步：将请求的URI路径进行URL编码（URI不含host，URI示例：/v3/user/get\_info）。  
请开发者关注：[URL编码注意事项](http://wiki.open.qq.com/wiki/%E8%85%BE%E8%AE%AF%E5%BC%80%E6%94%BE%E5%B9%B3%E5%8F%B0%E7%AC%AC%E4%B8%89%E6%96%B9%E5%BA%94%E7%94%A8%E7%AD%BE%E5%90%8D%E5%8F%82%E6%95%B0sig%E7%9A%84%E8%AF%B4%E6%98%8E#URL.E7.BC.96.E7.A0.81.E6.B3.A8.E6.84.8F.E4.BA.8B.E9.A1.B9)，否则容易导致后面签名不能通过验证。

第2步：将除“sig”外的所有参数按key进行字典升序排列。   
注：除非OpenAPI文档中特别标注了某参数不参与签名，否则除sig外的所有参数都要参与签名。

第3步：将第2步中排序后的参数(key=value)用&拼接起来，并进行URL编码。  
请开发者关注：[URL编码注意事项](http://wiki.open.qq.com/wiki/%E8%85%BE%E8%AE%AF%E5%BC%80%E6%94%BE%E5%B9%B3%E5%8F%B0%E7%AC%AC%E4%B8%89%E6%96%B9%E5%BA%94%E7%94%A8%E7%AD%BE%E5%90%8D%E5%8F%82%E6%95%B0sig%E7%9A%84%E8%AF%B4%E6%98%8E#URL.E7.BC.96.E7.A0.81.E6.B3.A8.E6.84.8F.E4.BA.8B.E9.A1.B9)，否则容易导致后面签名不能通过验证。   
  
第4步：将HTTP请求方式（GET或者POST）以及第1步和第3步中的字符串用&拼接起来。  
注：Java\_SDK\_V3.0.6仅支持POST方式，如果用GET可能导致一直计算sig不正确。   
  
**源串构造示例如下**  
（由于是通用说明，这里以/v3/user/get\_info作为示例，且示例中的请求串不可直接复制访问）

1. 原始请求信息：  
appkey：228bf094169a40a3bd188ba37ebe8723

HTTP请求方式：GET

请求的URI路径（不含HOST）：/v3/user/get\_info

请求参数：openid=11111111111111111&openkey=2222222222222222&appid=123456&pf=qzone&format=json&userip=112.90.139.30

2. 下面开始构造源串：

第1步：将请求的URI路径进行URL编码，得到：**%2Fv3%2Fuser%2Fget\_info**  
  
第2步：将除“sig”外的所有参数按key进行字典升序排列，排列结果为：appid，format，openid，openkey，pf，userip   
  
第3步：将第2步中排序后的参数(key=value)用&拼接起来：  
appid=123456&format=json&openid=11111111111111111&openkey=2222222222222222&pf=qzone&userip=112.90.139.30   
然后进行URL编码（ 编码时请关注[URL编码注意事项](http://wiki.open.qq.com/wiki/%E8%85%BE%E8%AE%AF%E5%BC%80%E6%94%BE%E5%B9%B3%E5%8F%B0%E7%AC%AC%E4%B8%89%E6%96%B9%E5%BA%94%E7%94%A8%E7%AD%BE%E5%90%8D%E5%8F%82%E6%95%B0sig%E7%9A%84%E8%AF%B4%E6%98%8E#URL.E7.BC.96.E7.A0.81.E6.B3.A8.E6.84.8F.E4.BA.8B.E9.A1.B9)，否则容易导致后面签名不能通过验证），编码结果为：  
**appid%3D123456%26format%3Djson%26openid%3D11111111111111111%26openkey%3D2222222222222222%26pf%3Dqzone%26  
userip%3D112.90.139.30**

第4步：将HTTP请求方式，第1步以及第3步中的到的字符串用&拼接起来，得到源串：

GET&%2Fv3%2Fuser%2Fget\_info&appid%3D123456%26format%3Djson%26openid%3D11111111111111111%26  
openkey%3D2222222222222222%26pf%3Dqzone%26userip%3D112.90.139.30

### Step 2. 构造密钥

得到密钥的方式：在应用的appkey末尾加上一个字节的“&”，即appkey&，例如：

228bf094169a40a3bd188ba37ebe8723&

### Step 3. 生成签名值

1. 使用HMAC-SHA1加密算法，使用Step2中得到的密钥对Step1中得到的源串加密。   
（注：一般程序语言中会内置HMAC-SHA1加密算法的函数，例如PHP5.1.2之后的版本可直接调用hash\_hmac函数。）

2. 然后将加密后的字符串经过Base64编码。   
（注：一般程序语言中会内置Base64编码函数，例如PHP中可直接调用 base64\_encode() 函数。）

3. 得到的签名值结果如下：

FdJkiDYwMj5Aj1UG2RUPc83iokk=