

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](#)；

（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](#) 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](#) 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](#)。

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](#)

(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](#)。

(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 。
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](#)

=====

v3/user/get_info

目录

[\[隐藏\]](#)

[What's New?](#)

[1 功能说明](#)

[2 接口调用说明](#)

[2.1 URL](#)

[2.2 格式](#)

[2.3 HTTP 请求方式](#)

[2.4 IP 限制](#)

[2.5 输入参数说明](#)

[2.6 请求示例](#)

[2.7 返回参数说明](#)

[2.8 错误返回码说明](#)

[2.9 正确返回示例](#)

[2.10 错误返回示例](#)

[3 示例代码](#)

[4 接口调试](#)

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](#)，其它 VIP 实时信息获取接口为：[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](#)。

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](#)
(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](#)。

(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	<p>pf=qqgame 时，必须输入该参数，指定需要获取 QQGame 中的哪些</p> <ol style="list-style-type: none"> 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 。
msg	如果错误，返回错误信息。
is_lost	<p>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。</p> <p>0 或者不返回：没有数据丢失，可以缓存。</p> <p>1：有部分数据丢失或错误，不要缓存。</p>
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 。
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](#)。

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](#)。

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](#)

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** 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**。

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** 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](#)。

api_name 的说明：

[api_name]：见 [API 列表](#) 中带 ■ 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](#)。
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带的参数有：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	<p>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。</p> <p>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。</p> <p>注：</p> <p>（1）如果不知道如何解析 CanvasURL 以获取 OpenID 和 Openkey，点击这里；</p> <p>（2）如果解析 CanvasURL 时获取不到 OpenID 和 Openkey 等参数，点击这里。</p>

openkey	必须	string	<p>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。</p> <p>注：</p> <p>（1）如果不知道如何解析 CanvasURL 以获取 OpenID 和 Openkey，点击这里；</p> <p>（2）如果解析 CanvasURL 时获取不到 OpenID 和 Openkey 等参数，点击这里；</p> <p>（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；</p> <p>（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，也不要后台存储 openkey，否则可能会导致用户无法登录；</p> <p>（5）openkey 过期时间为两小时，需要用 v3/user/is_login 接口续期。</p>
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 。
pf	必须	string	<p>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。</p> <p>注：</p> <p>（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。</p> <p>（2）如果不知道如何解析 CanvasURL 以获取 pf，点击这里；</p> <p>（3）如果解析 CanvasURL 时获取不到 pf，点击这里；</p> <p>（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。</p>
format		string	<p>定义 API 返回的数据格式。</p> <p>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。</p> <p>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。</p>
userip		string	<p>用户的 IP。</p> <p>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。</p> <p>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。</p> <p>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。</p>

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](#)。

OpenAPI V3.0 SDK 下载

详见：[SDK 下载#OpenAPI V3.0 SDK 下载](#)。

您还没有登录，[立即登录](#)

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](#)，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](#)，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=1111111111111111&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=1111111111111111&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](#)，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%3D1111111111111111%26openkey%3D2222222222222222%26pf%3Dqzone%26userip%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%3D1111111111111111%26openkey%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=
```

回调道具交换 URL 的协议说明

特别声明：

1. 本文档基于 V3 版 OpenAPI 协议。
2. 本接口仅适用于在支付接入中提交寄售接入申请并审核通过的应用。使用前请确保你的应用具有该 OpenAPI 的访问权限。

目录

[\[隐藏\]](#)

[1 什么是道具交换 URL?](#)

[2 协议参数](#)

[3 协议返回包](#)

[4 协议错误码](#)

[5 请求示例](#)

[6 返回示例](#)

[相关文档](#)

1 什么是道具交换 URL?

道具交换 URL 是提交申请时填写的“发货 URL”（详见：[支付接入申请说明#3. 支付接入申请时需填写的基本资料](#)），由应用指定。

开发者请按照下列协议进行发货 URL 的开发。

参数名称	是否必须		类型	描述
appid	openid	必须	string	在道具寄售系统中，购买道具的用户 openid。 与 APP 通信的用户 key，跳转到应用首页后，URL 后会带该参数。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。
	必须		string	应用的唯一 ID。可以通过 appid 查找 APP 基本信息。
ts	必须		string	linux 时间戳。 注意开发者的机器时间与腾讯计费开放平台的时间相差不能超过 15 分钟。
payitem	必须		string	物品购买信息。 (1) 接收标准格式为 ID*price*num，回传时 ID 为必传项。批量购买套餐物品则用";"分隔，字符串中不能包含" "特殊字符。 (2) ID 表示物品 ID，price 表示单价（以 Q 点为单位，单价最少不能少于 2Q 点，1Q 币=10Q 点。单价的制定需遵循 道具定价规范 ），num 表示购买数量。 示例： 批量购买套餐，套餐中包含物品 1 和物品 2。物品 1 的 ID 为 G001，该物品的单价为 10Q 点，购买数量为 1；物品 2 的 ID 为 G008，该物品的单价为 8Q 点，购买数量为 2，则 payitem 为： G001*10*1;G008*8*2 。
token	必须		string	应用调用 v3/pay/exchange_goods 接口成功返回的交易 token。 注意，交易 token 的有效期为 15 分钟，必须在获取到 token 后的 15 分钟内传递该 token，否则将会返回 token 不存在的错误。
billno			string	支付流水号（64 个字符长度。该字段和 openid 合起来是唯一

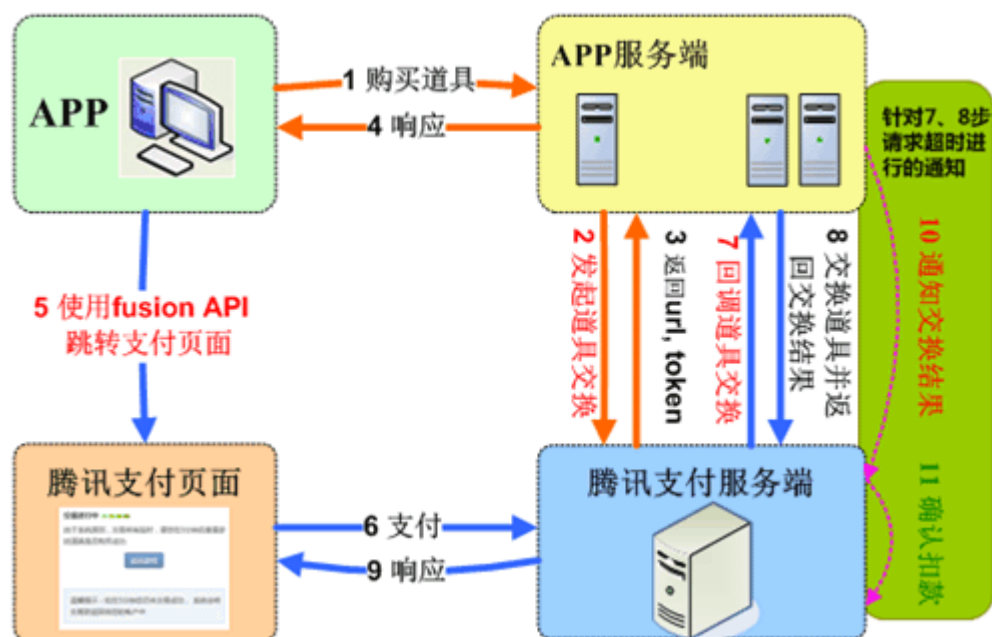
			的)。
version	必须	string	协议版本号，由于基于 V3 版 OpenAPI，这里一定返回“v3”。
zoneid	必须	string	<p>在支付营销分区配置说明页面，配置的分区 ID 即为这里的“zoneid”。</p> <p>如果应用不分区，则为 0。</p> <p>回调发货的时候，根据这里填写的 zoneid 实现分区发货。</p> <p>注：2013 年后接入的寄售应用，此参数将作为分区发货的重要参数，如果因为参数传错或为空造成的收入损失，由开发商自行承担。</p>
providetype	必须	string	<p>发货类型，这里请传入 3。</p> <p>0 表示道具购买，1 表示营销活动中的道具赠送，2 表示交叉营销任务集市中的奖励发放，3 表示道具寄售系统中的道具转移。</p>
amt	必须	string	Q 点/Q 币消耗金额或财付通游戏子账户的扣款金额。道具寄售目前暂不支持 Q 点/Q 币/财付通游戏子账户，因此该参数的值为 0。
seller_openid	必须	string	道具寄售系统中，售卖道具的用户的 openid。
fee	必须	string	<p>应用扣除的佣金，包括扣除公共游戏币账户和分区游戏币账户中的金额。</p> <p>佣金：用户在应用中售卖道具获得游戏币，需要付给应用一定的游戏币作为佣金。</p>
fee_acct	必须	string	佣金中扣掉 Q 币 Q 点的金额，道具寄售目前暂不支持 Q 点/Q 币或财付通子账户支付，因此该参数为 0。
fee_pubcoins	必须	string	<p>佣金中扣掉公共游戏币账户金额，包括充值得到的游戏币和赠送得到的游戏币。</p> <p>其中赠送得到的游戏币不参与结算，如无公共游戏币账户则为 0。</p>
fee_pubcoins_save	必须	string	<p>佣金中扣掉公共游戏币账户中通过充值得到的游戏币金额，该金额将参与结算。</p> <p>如无公共游戏币账户则为 0。</p>
fee_coins	必须	string	<p>佣金中扣掉分区游戏币账户金额，包括充值得到的游戏币和赠送得到的游戏币。</p> <p>其中赠送得到的游戏币不参与结算，如无分区游戏币账户则为 0。</p>
fee_coins_save	必须	string	<p>佣金中扣掉分区游戏币账户中通过充值得到的游戏币金额，该金额将参与结算。</p> <p>如无分区游戏币账户则为 0。</p>

uni_appamt	必须	string	购买的道具的价格。 注意，这里以 0.1Q 点为单位。即如果总金额为 18Q 点，则这里显示的数字是 180。请开发者关注，特别是对账的时候注意单位的转换。
cee_extend	必须	string	部署在 CEE_V2 的应用，必须传入该参数，用于将请求转发到对应的应用下的支付相关 Web 服务上进行处理。该参数不参与签名计算。 详见： CEE V2 访问云支付
sig	必须	string	请求串的签名，由需要签名的参数生成。 (1) 签名方法请见文档： 腾讯开放平台第三方应用签名参数 sig 的说明 。 (2) 按照上述文档进行签名生成时，需注意回调协议里多加了一个步骤： 在构造源串的第 3 步“将排序后的参数(key=value)用&拼接起来，并进行 URL 编码”之前，需对 value 先进行一次编码（编码规则为：除了 0~9 a~z A~Z !*() 之外其他字符按其 ASCII 码的十六进制加%进行表示，例如“-”编码为"%2D"）。 (3) 以每笔交易接收到的参数为准，接收到的所有参数除 sig 以外都要参与签名。为方便平台后续对协议进行扩展，请不要将参与签名的参数写死。 (4) 所有参数都是 string 型，进行签名时必须使用原始接收到的 string 型值。开发商出于本地记账等目的，对接收到的某些参数值先转为数值型再转为 string 型，导致字符串部分被截断，从而导致签名出错。如果要进行本地记账等逻辑，建议用另外的变量来保存转换后的数值。

道具交换 URL 用于提供给腾讯后台回调。在寄售系统中，用户 A 购买用户 B 的道具付费成功后，腾讯后台将回调该 URL 将道具从用户 B 转到用户 A 下。

本接口在以下场景使用：[道具寄售模式-道具寄售](#)

属于以下流程中的第 7 步：



2 协议参数

请开发者特别关注：

平台后续可能对协议进行扩展，因此请不要将参与签名的参数写死。

计算签名时，请以每笔交易接收到的参数为准，接收到的所有参数除 sig 以外都要参与签名。

3 协议返回包

应用的返回包应该包含如下参数：

ret: 返回码。

msg: 道具交换操作的结果，成功为“OK”，失败则表明错误原因（必须使用 utf8 编码）。

腾讯设置的调用第三方发货超时是 2 秒钟，请第三方注意超时时间设置不要超过 2 秒，否则腾讯后台将返回“系统繁忙”的错误消息。

4 协议错误码

应用的错误码应该从 0 开始，按照整数递增的方式进行定义，建议应用按照如下描述定义错误码：

0: 成功

1: 系统繁忙

2: token 已过期

3: token 不存在

4: 请求参数错误：（这里填写错误的具体参数）

5 请求示例

回调发货请求的签名生成较为复杂，许多应用在此出错，因此下面演示了 **sig** 签名的生成细节。开发者可以下面的示例来验证 **sig** 的详细过程，但不能直接复制。

第一步： 应用接收到腾讯支付后台发送过来的请求。

该请求通过后台发送，因此所有参数都未进行 URL 编码。但是该请求中所带的 **sig** 却进行了 URL 编码（这一点请开发者关注，在回调协议中这里比较特殊）。

请求示例如下：

```
http://ip/cgi-bin/demo_provide.cgi?amt=0&appid=15499&billno=-APPDJ10153-20120809-  
1150429539&fee=10&  
fee_acct=0&fee_coins=10&fee_coins_save=10&fee_pubcoins=0&fee_pubcoins_save=0&openid=00000  
000000000000000  
000000E1E0000&payitem=50005*2*10&providetype=3&seller_openid=00000000000000000000000000  
8FA509&  
token=2854C0C5BEC0AC942C020846C0D0B33129885&ts=1344484244&uni_appamt=200&version=v3  
&zoneid=1&  
sig=ZCKQN%2F0%2FBRNxzkrmK6GiwL1hyG8%3D
```

特别提示:

请求中的红色部分为应用指定的发货 URL，是提交支付接入申请时填写的“发货 URL”（详见：[支付接入申请说明#3. 支付接入申请时需填写的基本资料](#)）。

第二步： 应用对该请求进行解析，根据得到的请求源参数来计算 sig。

对上面的请求进行解析，得到源参数如下：

method:GET

```
url_path: /cgi-bin/demo_provide.cgi
```

假设 appkey 为: 56abfbcd12fe46f5ad85ad9f2faf36d7

请求源参数以及对应的值为：（注意：请以接收到的参数为准，除 **sig** 外的所有参数都需要参与签名）

amt: 0

appid: 15499

billno: -APPDJ10153-20120809-1150429539

fee: 10

```
fee acct: 10
```

```
fee_coins: 10
```

fee coins save: 10

```
fee pubcoins: 0
```

```
fee_pubcoins_save: 0
```

openid: 00000000000000000000000000000000E1E0000

```
payitem: 50005*2*10
```

在回调发货协议中，在进行签名生成时，回调协议里多加了一个步骤：
在构造源串第 3 步之前（sig 生成通用步骤说明详见[这里](#)），需对 value 先按照如下编码规则进行编码（注意这里不是 urlencode）：
除了 0~9 a~z A~Z !*() 之外其他字符按其 ASCII 码的十六进制加%进行表示，例如“-”编码为“%2D”。
payitem 中，单价如果有小数点“.”，请编码为“%2E”。

```
amt: 0
appid: 15499
billno: %252DAPPDJ10153%252D20120809%252D1150429539
fee: 10
fee_acct: 10
fee_coins: 10
fee_pubcoins: 0
fee_pubcoins_save: 0
openid: 00000000000000000000000000000000E1E0000
payitem: 50005*2*10
providetype: 3
selleropenid: 00000000000000000000000000000008FA509
token: 2854C0C5BEC0AC942C020846C0D0B33129885
ts: 1344484244
uni_appamt: 200
version: v3
zoneid: 1
```

则源串为: GET&%2Fcgi-bin%2Fdemo_provide.cgi&amt%3D0%26appid%3D15499%26billno%3D%252DAPPDJ10153%252D20120809

```
%252D1150429539%26fee%3D10%26fee_acct%3D0%26fee_coins%3D10%26fee_coins_save%3D10%26fee_pubcoins%3D0%26fee_pubcoins_save%3D0%26openid%3D00000000000000000000000000000000E1E0000%26payitem%3D50005%2A2%2A10%26providetype%3D3%26seller_openid%3D000000000000000000000000000000008FA509%26token%3D2854C0C5BEC0AC942C020846C0D0B33129885%26ts%3D1344484244%26uni_appamt%3D200%26version%3Dv3%26zoneid%3D1
```

密钥为：56abfbcd12fe46f5ad85ad9f2faf36d7&

生成的签名为：ZCKQN/0/BRNxzkrmK6GiwL1hyG8=

第三步： 应用将自己生成的 **sig** 与腾讯支付后台发送的请求中的 **sig** 进行比对。

注意： 应用接收到的腾讯支付平台发送过来的请求后，请先对请求中的 **sig** 值做一次 URL 解码；

然后再将解码后的 **sig** 值与应用自己生成的 **sig** 值进行比对。

在 PHP 中，通过 `$_GET['sig']` 获取 **sig** 值，PHP 语言本身对 **sig** 值进行了 URL 解码，不需要再次解码。

如果 **sig** 值一致，表示该请求合法，应用可以处理该请求并将结果返回给腾讯支付后台。

6 返回示例

正确返回示例：

```
Content-type: text/html; charset=utf-8
{
  "ret":0,
  "msg":"OK"
}
```

错误返回示例：

```
Content-type: text/html; charset=utf-8
{
  "ret":4,
  "msg":"请求参数错误：（sig）"
}
```

相关文档

上一步: [fusion2.dialog.buy](#)

下一步: [v3/pay/confirm_delivery](#)

其他相关文档: [支付接入#支付接入文档索引](#) | [支付测试环境说明](#) | [支付错误码说明](#) | [支付功能开发相关 FAQ](#)

您还没有登录, [立即登录](#)

以上信息是否解决您的问题?