

腾讯社交广告移动APP转化追踪API统计方案对接文档(API 方案一)

腾讯社交广告移动APP转化追踪API统计方案对接文档(API 方案一)

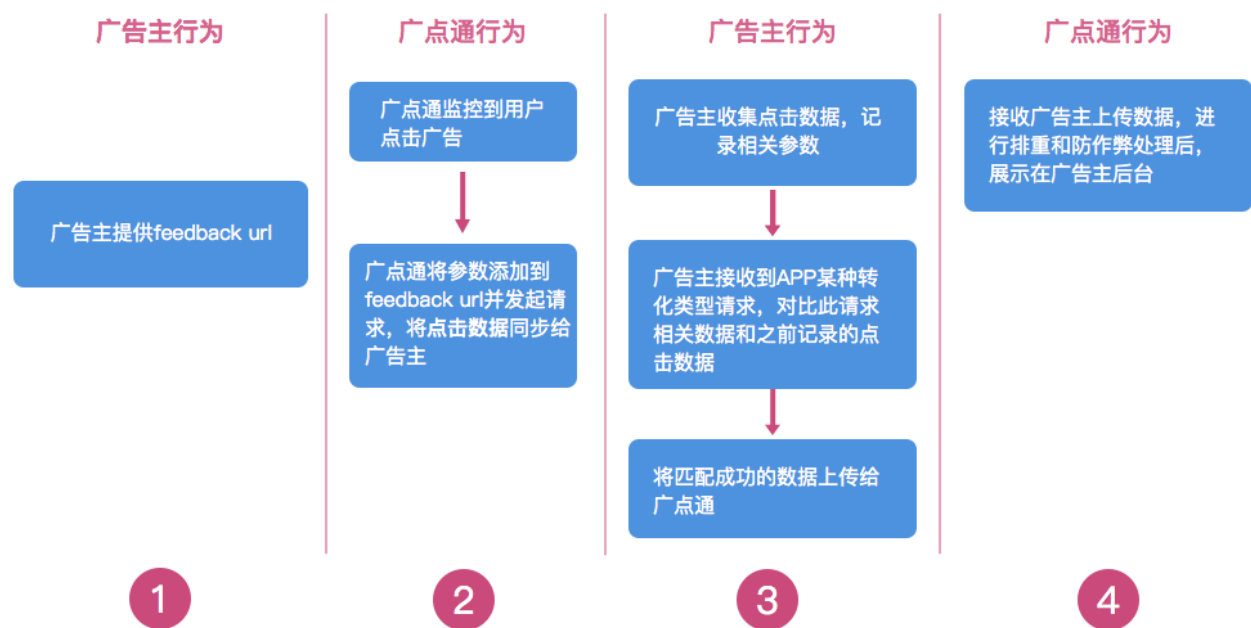
- 1.文档介绍
- 2.API功能逻辑介绍
- 3.接入流程
- 4.接收点击接口（接口一）
 - 4.1 接收点击接口介绍
 - 4.2 接收点击接口格式
 - 4.3 接受点击接口参数介绍
 - 4.4 feedback url 介绍
 - 4.5 muid加密方法
 - 4.6 接收点击接口响应信息
5. 转化数据上报接口（接口二）
 - 5.1 转化数据上报接口介绍
 - 5.2 转化数据上报接口格式
 - 5.3 转化数据上报接口参数介绍
 - 5.4 转化数据上报接口响应信息
 - 5.5 转化数据上报接口加密组装流程
 - 5.6加密密钥和签名密钥
 - 5.7 data加密方案
 - 5.7.1 data加密流程
 - 5.7.2 组合参数生成query_string
 - 5.7.3 利用query_string构造page
 - 5.7.4 将page进行urlencode得到encode_page
 - 5.7.5 将encode_page组装成property
 - 5.7.6 利用property生成signature
 - 5.7.7 利用signature和query_string生成base_data
 - 5.7.8 利用base_data和encrypt_key生成data
 - 5.8组装请求
 - 5.8.1 构造attachment
 - 5.8.2 构造最终请求
- 6.接入多种转化类型
 - 6.1 多种转化类型介绍
 - 6.2 接入多种转化类型
- 7 附录
 - 7.1 转化数据上报接口异常信息
 - 7.2 muid加密代码示例

- 7.2.1 muid加密代码示例(Python)
- 7.2.2 muid加密代码示例(Java)
- 7.2.3 muid加密代码示例(PHP)
- 7.3 简单异或加密代码示例
 - 7.3.1 简单异或加密代码示例(Python)
 - 7.3.2 简单异或加密代码示例(Java)
 - 7.3.3 简单异或加密代码示例(PHP)
- 7.4 关于文档

1.文档介绍

此文档面向广告主开发人员，用于描述API方案一的接入流程。

2.API功能逻辑介绍



3.接入流程

| 分工职责 | 开发人员 | 投放端操作人员 |
|------|------|---------|
| | | |

| 分工职责 | 开发人员 | 投放端操作人员 |
|------|---|--|
| 具体工作 | 1.提供feedback url 2.完成接口1，接口2的开发 3.确认接口2联调成功（联调前需要确认转化规则已启用） | 1.创建转化规则 2.提供转化规则相关信息给开发人员，包括：appid，账户id，密钥 3.完成接口1联调（联调前需确认feedback url可访问） 4.启用转化规则 |

4.接收点击接口（接口一）

4.1 接收点击接口介绍

- **接收点击接口**用于广点通将用户的点击行为信息发送给广告主
- **接收点击接口**由广告主自定义的feedback url 和广点通拼接的参数共同构成。
- 实现此接口需要广告主提供广告主自定义的feedback url，广点通按照规则拼接参数后调用**接收点击接口**。
- 广点通只会转发有设备号的点击，因此广点通投放端显示的点击可能与实际转发的点击数有差距。

4.2 接收点击接口格式

- 请求方：广点通点击后台
- 处理方：广告主后台
- 请求协议：http
- 请求方式：GET

接收点击接口示例：

<http://www.example.com/gdt.cgi?>

[muid={muid}](#)

[&click_time={click_time}](#)

[&appid={appid}](#)

[&click_id={click_id}](#)

[&app_type={app_type}](#)

[&advertiser_id={advertiser_id}](#)

实例如下

Android 应用:

```
http://www.example.com/gdt.cgi?
muid=40c7084b4845eebce9d07b8a18a055fc&click_time=1406276499&appid=000000&click_id=007210548a030059c
dfd1d4&app_type=android&advertiser_id=20000
```

iOS 应用:

```
http://www.example.com/gdt.cgi?
```

```
muid=40c7084b4845eebce9d07b8a18a055fc&click_time=1406276499&appid=000000&click_id=007210548a030059c  
dfd1d4&app_type=ios&advertiser_id=20000
```

4.3 接受点击接口参数介绍

| 名称 | 类型 | 描述 |
|---------------|--------|--|
| muid | string | 设备id，由IMEI (Android应用) md5生成，或是由IDFA (iOS应用) md5生成；具体加密方案在 4.5muid加密方案 有详细说明 |
| click_time | string | 点击发生的时间，由广点通系统生成，取值为标准时间戳， 秒级别 |
| click_id | string | 广点通后台生成的点击id，广点通系统中标识用户每次点击生成的唯一标识 |
| appid | int | Android应用为应用宝移动应用的id，或者iOS应用在Apple App Store的id；创建转化时，需填入此appid |
| advertiser_id | int | 广告主在广点通 (e.qq.com) 的账户id |
| app_type | string | app类型；取值为 android或ios (联盟Android为unionandroid) ；注意是小写；根据广告主在广点通 (e.qq.com) 创建转化时提交的基本信息关联； |

4.4 feedback url 介绍

feedback url 是由广告主自定义的用于广点通将点击数据传输给广告主的实现连接

广告主在广点通投放端 (e.qq.com) 创建转化时，需要输入feedback url

feedback URL要求：

- 可以有参数，但不能包含符号:#
- 不要占用如下参数：muid, click_time, click_id, app_type, appid , advertiser_id
- 可以用https
- 不能直接用ip，必须是正式的域名

示例如下：

<http://www.example.com/gdt.cgi> 备注：正确 (√)

<http://www.example.com/gdt.cgi?source=gdt> 备注：可以添加参数 (√)

[http://www.example.com/gdt.cgi?](http://www.example.com/gdt.cgi?muid=xxxxx&click_time=1406276499&appid=00000&click_id=00000&app_type=android&advertiser_id=0000)

[muid=xxxxx&click_time=1406276499&appid=00000&click_id=00000&app_type=android&advertiser_id=0000](http://www.example.com/gdt.cgi?muid=xxxxx&click_time=1406276499&appid=00000&click_id=00000&app_type=android&advertiser_id=0000)

备注：不能用特定的参数名称 (×)

4.5 muid加密方法

muid:设备id,由 IMEI(Android 应用)、IDFA(iOS 应用)md5生成。具体加密方案如下:

Android 设备-muid 加密规则:

IMEI 号(需转小写)，进行 md5 以后得到的 **32位全小写** md5 表现字符串。

iOS 设备-muid 加密规则:

IDFA 码(需转大写), 进行 md5 以后得到的 32位全小写 md5 表现字符串。

muid加密代码示例 (C++)

```
1.  bool GenerateMuid(const std::string& uni_id, const int32_t app_type, std::string* muid)
2.  {
3.      if (muid == NULL || uni_id.empty()) {
4.          return false;
5.      }
6.      muid->clear();
7.      std::string tmp;
8.      if (app_type == kAppTypeAndoirdImei) {
9.          LatinToLower(uni_id, &tmp);
10.     }
11.     else if (app_type == kAppTypeiOSIfa) {
12.         LatinToUpper(uni_id, &tmp);
13.     }
14.     else {
15.         return false;
16.     }
17.     md5sum32l(tmp, muid);    // 32bit lower
18.     return true;
19. }
```

[muid加密代码示例 \(Python \)](#)

[muid加密代码示例 \(PHP \)](#)

[muid加密代码示例 \(Java \)](#)

4.6 接收点击接口响应信息

响应格式：Json数据格式

响应内容：{"ret": 返回码, "msg": "错误提示"}

返回码：返回码为0标识正常接收，其他返回码标识错误

注意：返回值必须有『ret』,且参数值必须为0；否则视为不成功；返回值不能包含『302』

返回值中的『错误信息』如果是中文，必须使用UTF-8字符集，不能使用GBK等其他字符集

5. 转化数据上报接口 (接口二)

5.1 转化数据上报接口介绍

转化数据上报接口用于用户在后台核对数据后，将有效转化数据上报给广点通。

转化数据上报接口需广告主按照广点通提供的接口格式补齐全部参数后将数据上传给广点通后台。

5.2 转化数据上报接口格式

```
http://t.gdt.qq.com/conv/app/{appid}/conv?v={data}&conv_type={conv_type}&app_type={app_type}&advertiser_id={advertiser_id}
```

5.3 转化数据上报接口参数介绍

| 名称 | 类型 | 描述 |
|---------------|--------|---|
| appid | int | Android应用为应用宝移动应用的id，或者iOS应用在Apple App Store的id；如果是移动联盟推广的App，则由移动联盟系统分配appid |
| data | string | 加密处理后的用户数据，详见 5.7data加密方案 |
| conv_type | string | 转化行为标记参数，激活参数取MOBILEAPP_ACTIVITE。更多转化类型接入详见 6.接入多种转化类型统计 |
| app_type | string | app类型；取值为 android 或 ios；注意是 小写 |
| advertiser_id | int | 广告主在广点通（e.qq.com）的账户id（不是QQ号） |

5.4 转化数据上报接口响应信息

响应格式：Json数据格式

响应内容：{"ret": 返回码, "msg": "错误提示"}

返回码：返回码为0标识正常接收，其他返回码标识错误

注意：关于返回码的异常信息解释，详见[7.1转化数据上报接口异常信息](#)

5.5 转化数据上报接口加密组装流程

| 流程 | 输入参数 | 操作方法 | 输出参数 |
|---------------------------|---|--|--------------|
| 1.组合参数(加密流程) | muid conv_time click_id client_ip(选填) value(选填) | query_string : muid={muid}&conv_time={conv_time}&click_id={click_id}&client_ip={client_ip}&value={value} | query_string |
| 2.参数签名(加密流程) | appid query_string | page : http://t.gdt.qq.com/conv/app/{appid}/conv? {query_string} | page |
| 3.参数签名 urlencode(加密流程) | page | 对page整体进行urlencode | encode_page |
| 4.组装字符串 property(加密流程) | encode_page | property : {sign_key}&GET&{encode_page} | property |

| 流程 | 输入参数 | 操作方法 | 输出参数 |
|------------------------|--|---|------------|
| 5.加密生成 signature(加密流程) | property | md5加密获得 32 位的小写加密串 | signature |
| 6.参数加密(加密) | query_string signature | base_data : {query_string}&sign=urlencode({signature}) | base_data |
| 7.简单异或 +base64(加密流程) | base_data encrypt_key | data : base64(simple_xor({base_data}, {encrypt_key})) ※base64 编码后,需要去除换行符再进行 URL 编码 | data |
| 8.组装请求(组装流程) | conv_type app_type advertiser_id | attachment : conv_type={conv_type}&app_type={app_type}&advertiser_id={advertiser_id} ※此处的app_type必须为大写 | attachment |
| 9.最终请求(组装流程) | appid data attachment | http://t.gdt.qq.com/conv/app/{appid}/conv?v={data}&{attachment} | 最终请求 |

5.6加密密钥和签名密钥

而对于每一个广告主账户，广点通都会分配一个加密密钥（encrypt_key）和一个签名密钥（sign_key）。

获取方式：

广告主在广点通创建转化规则之后,系统会自动生成密钥，点击『查看』即可查看密钥。

※一个广告主账户下,无论不同的APP或不同的转化规则，密钥是相同的。



密钥新老兼容问题

若广告主在某移动APP接入多种转化类型功能上线前,已经创建了此APP 的激活统计, 则此激活统计使用的原密钥系统会进行保留,不影响该部分激活数据的上报和投放端报表展示。

若广告主后继创建此 APP 的注册、加入购物车和 付费行为的转化规则,则获取的密钥为一组新密钥（注册、加入购物车和付费行为的密钥是相同的）。

若广告主为了方便加密转化数据的统一管理,也可将之前的与旧密钥关联的激活转化删除,然后重新创建新的激活转化规则,获取与注册、加入购物车和付费行为相同的一组新密钥。

※广告主进行上报信息加密时,请注意**密钥的新旧问题**。若使用密钥加密发生错误,则 广点通接收侧不能对上报信息进行正确解析,从而影响投放端报表对转化数据指标的准确显示。

5.7 data加密方案

5.7.1 data加密流程

1. 组合参数生成query_string
2. 利用query_string构造page
3. 将page进行urlencode得到encode_page
4. 将encode_page组装成property
5. 利用property生成signature
6. 利用signature和query_string生成base_data
7. 利用base_data和encrypt_key生成data

5.7.2 组合参数生成query_string

所需参数：

| 名称 | 类型 | 描述 |
|-----------|--------|--|
| click_id | string | 广点通点击跟踪 id(必填) |
| muid | string | 设备id，由IMEI（Android应用）md5生成，或是由IDFA（iOS应用）md5生成；具体加密方案在 5.6muid加密方案 有详细说明 |
| conv_time | string | 转化发生时间(必填) |
| client_ip | string | 转化发生用户的客户端IP(选填) |
| value | int | 转化价值，例如加入购物车和付费的金额，传入此参数后，会在投放端展示广告带来的金额，单位为分，1元=100（选填） |

※此处如果不填写 client_ip,可以直接在 query_string 中去除该参数

处理规则：

query_string=

```
muid={muid}&conv_time={conv_time}&click_id={click_id}&client_ip={client_ip}&value={value}
```


例如：

click_id:007210548a030059ccdfd1d4
muid: 0f074dc8e1f0547310e729032ac0730b
conv_time: 1422263664
client_ip: 10.11.12.13

输出为：

query_string：

```
click_id=007210548a030059ccdfd1d4&muid=0f074dc8e1f0547310e729032ac0730b&conv_time=1422263664&client_ip=10.11.12.13
```

5.7.3 利用query_string构造page

所需参数：

| 名称 | 类型 | 描述 |
|--------------|--------|---|
| appid | int | Android应用为应用宝移动应用的id，或者iOS应用在Apple App Store的id；如果是移动联盟推广的App，则由移动联盟系统分配appid |
| query_string | string | 由click_id、muid、conv_time和client_ip构造而成 |

处理规则：

page=

```
http://t.gdt.qq.com/conv/app/{appid}/conv?{query_string}
```

例如：

appid: 112233
query_string:
click_id=007210548a030059ccdfd1d4&muid=0f074dc8e1f0547310e729032ac0730b&conv_time=1422263664&client_ip=10.11.12.13

输出为：

page：

```
http://t.gdt.qq.com/conv/app/112233/conv?click_id=007210548a030059ccdfd1d4&muid=0f074dc8e1f0547310e729032ac0730b&conv_time=1422263664&client_ip=10.11.12.13
```

5.7.4 将page进行urlencode得到encode_page

所需参数：

| 名称 | 类型 | 描述 |
|------|--------|-----------------------------|
| page | string | 由query_string和appid按照规则构造生成 |

处理规则：

encode_page=对page字符串进行整体urlencode，得到encode_page

例如：

page:

```
http://t.gdt.qq.com/conv/app/112233/conv?
click_id=007210548a030059ccdfd1d4&muid=0f074dc8e1f0547310e729032ac0730b&conv_time=1422263664&client
ip=10.11.12.13
```

输出为：

encode_page：

```
http%3A%2F%2Ft.gdt.qq.com%2Fconv%2Fapp%2F112233%2Fconv%3Fclick_id%3D007210548a030059ccdfd1d4%26muid
3D0f074dc8e1f0547310e729032ac0730b%26conv_time%3D1422263664%26client_ip%3D10.11.12.13
```

5.7.5 将encode_page组装成property

所需参数：

| 名称 | 类型 | 描述 |
|-------------|--------|----------------------|
| encode_page | string | 由page进行整体urlencode生成 |
| sign_key | string | 广点通用户签名密钥 |

处理规则：

property={sign_key}&GET&{encode_page}

例如：

sign_key：test_sign_key

encode_page：

```
http%3A%2F%2Ft.gdt.qq.com%2Fconv%2Fapp%2F112233%2Fconv%3Fclick_id%3D007210548a030059ccdfd1d4%26muid
3D0f074dc8e1f0547310e729032ac0730b%26conv_time%3D1422263664%26client_ip%3D10.11.12.13
```

输出为：

property：

```
test_sign_key&GET&http%3A%2F%2Ft.gdt.qq.com%2Fconv%2Fapp%2F112233%2Fconv%3Fclick_id%3D007210548a030
59ccdfd1d4%26muid%3D0f074dc8e1f0547310e729032ac0730b%26conv_time%3D1422263664%26client_ip%3D10.11.1
```

5.7.6 利用property生成signature

所需参数：

| 名称 | 类型 | 描述 |
|----------|--------|-------------------------------|
| property | string | 由sign_key和encode_page按照规则构造生成 |

处理规则：

signature=对 property 进行 md5 获得 32 位的小写加密串

例如：

page:

```
test_sign_key&GET&http%3A%2F%2Ft.gdt.qq.com%2Fconv%2Fapp%2F112233%2Fconv%3Fclick_id%3D007210548a03059ccdfd1d4%26muid%3D0f074dc8e1f0547310e729032ac0730b%26conv_time%3D1422263664%26client_ip%3D10.11.1.13
```

输出为：

signature：

```
c2f87710541942364691e5e8adc84f3f
```

5.7.7 利用signature和query_string生成base_data

所需参数：

| 名称 | 类型 | 描述 |
|--------------|--------|--|
| signature | string | 由property进行md5生成 |
| query_string | string | 由click_id、muid、conv_time和client_ip构造而成 |

处理规则：

base_data= {query_string}&sign=urlencode({signature})

例如：

query_string:

click_id=007210548a030059ccdfd1d4&muid=0f074dc8e1f0547310e729032ac0730b&conv_time=1422263664&client_ip=10.11.12.13

signature: c2f87710541942364691e5e8adc84f3f

输出为：

base_data：

```
click_id=007210548a030059ccd1d4&muid=0f074dc8e1f0547310e729032ac0730b&conv_time=1422263664&client_ip=10.11.12.13&sign=c2f87710541942364691e5e8adc84f3f
```

5.7.8 利用base_data和encrypt_key生成data

所需参数：

| 名称 | 类型 | 描述 |
|-------------|--------|---------------------------------|
| base_data | string | 由query_string和signature按照规则构造生成 |
| encrypt_key | string | 广点通账户加密密钥 |

处理规则：

base_data和encrypt_key进行简单异或后的base64格式编码

※ base64 编码后,需要去除换行符再进行 URL 编码

data=base64(simple_xor({base_data}, {encrypt_key}))

例如：

base_data: click_id=007210548a030059ccd1d4&muid=0f074dc8e1f0547310e729032ac0730b&conv_time=1422263664&client_ip=10.11.12.13&sign=c2f87710541942364691e5e8adc84f3f

encrypt_key: test_encrypt_key

此处密钥包含了下划线是否要修改

输出为：

data:

```
FwkaFzQ6BwdPSUBDbVpVTEBdEkRsVV5WSxoTEDkPVB1AQx4BNgFTUxRJr0A7CF0cRNDQWtSXVJCHEdGZltWSxUGQ0NsVQxFERY
AgAfDBQRWEJAbVdcVUFPRkB5CAkQEqsHKzYVU1JCV0FFcVpXV0VWVQc2AgBeEUswTGhcVElBUUJNa1ddVUZPSUU6XgBBFQEQTGs
XQU=
```

简单异或加密代码示例(C++):

```
1.  bool SimpleXor(const std::string& info,const std::string& key, std::string* result)
2.  {
3.      if (result == NULL || info.empty() || key.empty())
4.      {
5.          return false;
6.      }
7.      result->clear();
8.      uint32_t i = 0;
9.      uint32_t j = 0;
```

```

10.         for (; i < info.size(); ++i) {
11.             result->push_back(static_cast<unsigned char>(info[i] ^ key[j]));
12.             j = (++j) % (key.length());
13.         }
14.         return true;
15.     }

```

[简单异或加密代码示例 \(Python \)](#)

[简单异或加密代码示例 \(PHP \)](#)

[简单异或加密代码示例 \(Java \)](#)

5.8 组装请求

5.8.1 构造attachment

所需参数：

| 名称 | 类型 | 描述 |
|---------------|--------|--|
| conv_type | string | 转化行为标记参数，激活参数取MOBILEAPP_ACTIVITE。更多转化类型接入详见 6.接入多种转化类型统计 |
| app_type | string | app类型;iOS 应用取值为 IOS，Android 应用取值为 ANDROID，联盟Android 应用取值为 UNIONANDROID，必须为大写 |
| advertiser_id | int | 广告主在广点通(e.qq.com)的账户id |

处理规则：

attachment=

```
conv_type={conv_type}&app_type={app_type}&advertiser_id={advertiser_id}
```

※此处的app_type必须为大写

例如：

conv_type: MOBILEAPP_ACTIVITE

app_type: ANDROID

advertiser_id: 10000

输出为：

attachment：

```
conv_type=MOBILEAPP_ACTIVITE&app_type=ANDROID&advertiser_id=10000
```

5.8.2 构造最终请求

所需参数：

| 名称 | 类型 | 描述 |
|------------|--------|---|
| appid | int | 广告主在广点通(e.qq.com)的账户id |
| data | string | 加密处理后的用户数据，详见 5.7data加密方案 |
| attachment | string | 由conv_type、app_type和advertiser_id按照规则构造生成 |

处理规则：

最终请求格式：

```
http://t.gdt.qq.com/conv/app/{appid}/conv?v={data}&{attachment}
```

例如：

appid: 112233

data:

```
FwkaFzQ6BwdPSUBDbVpVTEBdEkRsVV5WSxoTEDkPVB1AQx4BNgFTUxRJR0A7CF0cRQNDQWtSXVJCHEdGZ1tWSxUGQ0NsVQxFERY
AgAfDBQRWEJAbVdcVUFPRkB5CAkQEQsHKzYVU1JCV0FFcVpXV0VWVQc2AgBeEUsWTGhcVE1BUUJNa1ddVUZPSUU6XgBBFQEQTGs
XQU%3D
```

attachment: conv_type=MOBILEAPP_ACTIVITE&app_type=ANDROID&advertiser_id=10000

输出为：

最终请求：

```
http://t.gdt.qq.com/conv/app/112233/conv?
v=FwkaFzQ6BwdPSUBDbVpVTEBdEkRsVV5WSxoTEDkPVB1AQx4BNgFTUxRJR0A7CF0cRQNDQWtSXVJCHEdGZ1tWSxUGQ0NsVQxFE
YeAgAfDBQRWEJAbVdcVUFPRkB5CAkQEQsHKzYVU1JCV0FFcVpXV0VWVQc2AgBeEUsWTGhcVE1BUUJNa1ddVUZPSUU6XgBBFQEQT
sDXQU%3D&conv_type=MOBILEAPP_ACTIVITE&app_type=ANDROID&advertiser_id=10000
```

返回值为0代表请求成功。返回值异常详见[7.1转化数据上报接口异常信息](#)

6.接入多种转化类型

6.1 多种转化类型介绍

| 转化类型 | 定义 | conv_type 取值 |
|------|--|--------------------|
| 激活 | 用户下载应用后首次打开的操作行为。 | MOBILEAPP_ACTIVITE |
| 注册 | 用户在应用内完成注册,如手机号码验证、注册应用下账户、创建游戏角色、通过QQ、微信号等方式通过授权或其他类注册行为。 | MOBILEAPP_REGISTER |

| 转化类型 | 定义 | conv_type 取值 |
|-------|--|---------------------|
| 加入购物车 | 用户在应用内发起购买而未完成付费的行为,如加入购物车、下单等。支持传入金额。 | MOBILEAPP_ADDTOCART |
| 付费行为 | 用户在应用内进行付费,完成了交易的操作行为。支持传入金额。 | MOBILEAPP_COST |

※在不同 APP 内,除转化类型“激活”外,“注册”,“加入购物车”和“付费行为”的定义不尽相同,此处提供的定义内容仅供参考。请广告主参照该定义并结合自身 APP 特点,进行转化数据的上报。

6.2 接入多种转化类型

在已经接入激活的前提下,接入其他转化类型需要在广点通后台创建新的转化规则。

采用API方案一的情况下,需要保证其他转化类型和激活使用同一个feedback url,目前只有激活的feedback url能够收到广点通转发的点击信息。

接入多种转化类型的流程与接入激活转化完全一致。

对于接入多种转化类型时,新老密钥问题详见[5.7加密密钥和签名密钥](#)

7 附录

7.1 转化数据上报接口异常信息

| ret返回码 | 含义 |
|----------|---|
| -3 | 解密失败 或 签名校验不通过 |
| -4 or -5 | URL内path信息填写有误 |
| -12 | 获取密钥信息异常,可联系广点通工作人员解决 |
| -13 | app_type参数有误 |
| -14 | conv_time参数缺失 |
| -15 | muid参数有误 |
| -17 | 获取转化信息失败,可能是广告主id、app id或conv type有误导致找不到对应的转化;或者是相应转化类型的规则没有创建;也有可能是转化没有开启 |
| -19 | data参数缺失 |

7.2 muid加密代码示例

7.2.1 muid加密代码示例(Python)

```
1.  #!/usr/bin/env python
2.
3.  import sys
4.  from hashlib import md5 as md5sum
5.  from re import match as rematch
6.
7.  def generate_muid(device_id):
8.      if rematch('^[0-9A-Fa-f]{1,15}$', device_id) != None:
9.          # imei/meid
10.         return md5sum(device_id.lower()).hexdigest().lower()
11.     elif rematch(
12.         '^([0-9a-zA-Z]{1,})(([/\s-][0-9a-zA-Z]{1,}){4,})$', device_id) != None:
13.         # idfa
14.         return md5sum(device_id.upper()).hexdigest().lower()
15.     else:
16.         return None
17.
18. if __name__ == '__main__':
19.     if len(sys.argv) > 1:
20.         device_id = str(sys.argv[1])
21.         muid = generate_muid(device_id)
22.         print device_id + " ==> " + (muid if muid != None else "None")
23.     else:
24.         print "need device id"
```

7.2.2 muid加密代码示例(Java)

7.2.3 muid加密代码示例(PHP)

7.3 简单异或加密代码示例

7.3.1 简单异或加密代码示例(Python)

7.3.2 简单异或加密代码示例(Java)

7.3.3 简单异或加密代码示例(PHP)

7.4 关于文档

对于接入过程中存在的疑问，有如下渠道进行咨询：

QQ：2508973061 白璿

RTX：v_junbai, franciszhao