
微信广告移动 APP 转化数据统计接入指引

1 前言：	2
2 方案说明	2
3 接口一：接收微信广告提供的广告点击信息	3
3.1 Feedback url 格式：	3
3.2 微信广告请求格式	4
3.3 Feedback url 请求响应格式	5
3.4 Feedback url 请求响应内容	5
4 创建激活转化规则	5
4.1 创建规则入口	5
4.2 给应用创建规则	6
4.3 已创建的规则列表	9
4.4 联调转化规则	9
4.5 启动转化规则	10
4.6 编辑转化规则	10
4.7 解绑转化规则	11
5 转化数据上报接口	12
5.1 转化数据接口上报简介（原有方案）	13
5.2 转化数据接口上报简介（加密简化方案）	14
6 附：Muid 加密方案	19
7 附：V 参数加密方案	21

8 附：V 参数二次 urlencode	28
9 联系我们	28

1 前言：

转化数据是APP广告主衡量转化效果的重要指标。为了方便、准确的统计到广告主所推广移动APP的转化数据，微信广告为广告主提供了转化统计的方案如下，可追踪到的iOS 和Android APP转化行为，后续也将会逐渐支持更多转化类型。

说明：此文档说明与技术开发紧密相关，建议广告主运营人员在开发人员指导下阅读。

2 方案说明

方案简要：微信广告提供广告点击数据，广告主自建服务系统，将所要跟踪的转化类型数据与广告点击数据进行关联。

- 【Step1】：由广告主提供回调 url，即 feedback url：接收微信广告请求的点击数据；([接口一](#))
- 【Step2】：广告主在微信公众平台（mp.weixin.qq.com）的广告主-账户管理-工具箱-应用激活数据功能模块，使用 Feedback url 创建 iOS 或者 Android 应用转化规则，并自助联调及启动规则；
- 【Step3】：当用户点击广告时，微信广告的点击处理后台会获取相关参数，添加到广告主提供的 feedback url，发起请求，同步给广告主；
- 【Step4】：广告主需要搭建服务，对微信广告的请求 url 进行处理，并记下相关参数；([接口一](#))
- 【Step5】：当广告主收到 APP 激活转化类型请求时，比对转化类型数据与上述记

下的点击数据，经比对后，如果有微信广告在 5 天内带来的转化，则将微信广告点击追踪 ID 等信息回传微信广告。（[转化数据上报接口](#)）

3 接口一：接收微信广告提供的广告点击信息

广告主研发并提供 Feedback url，微信广告将会把广告点击信息添加到 Feedback url 中，请求给广告主。

3.1 Feedback url 格式：

请求方：微信广告点击后台

处理方：广告主后台

请求协议：http

请求方式：GET

请求地址：即feedback URL，将用于广告主在微信广告投放平台(mp.weixin.qq.com)

创建转化规则所提供请求的地址；

feedback URL要求：可以有参数，但请不要包含符号：#；也不要占用如下参数：muid,

click_time, click_id, app_type, appid，advertiser_id；如下：

`http://www.example.com/gdt.cgi` (√)

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

`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 (×)

3.2 微信广告请求格式

当用户点击广告时，微信广告点击处理后台会获取相关参数，添加到feedback url，发起请求，同步给广告主，请求格式如下：

- iOS应用：

```
http://www.example.com/gdt.cgi?muid=40c7084b4845eebce9d07b8a18a055f  
c&click_time=1406276499&appid=000000&click_id=007210548a030059ccdfd  
1d4&app_type=ios&advertiser_id=20000
```

- Android应用：

```
http://www.example.com/gdt.cgi?muid=40c7084b4845eebce9d07b8a18a055f  
c&click_time=1406276499&appid=000000&click_id=007210548a030059ccdfd  
1d4&app_type=android&advertiser_id=20000
```

3.2.1 请求中的参数说明：

- muid：设备 id，由 IDFA（iOS 应用）MD5 生成，或是由 IMEI（Android 应用）MD5 生成；具体加密方案在第 6 部分（muid 加密方案）详细说明；
- click_time：点击发生的时间，由微信广告系统生成，取值为标准时间戳，秒级别；
- click_id：微信广告后台生成的点击 id，微信广告系统中标识用户每次点击生成的唯一标识；
- appid：Android 应用为应用宝移动应用的 id，或者 iOS 应用在 Apple App Store 的 id；根据广告主在微信广告投放平台（mp.qq.weixin.com）创建转化时提交的基本信息关联；
- advertiser_id：广告主在微信广告（mp.qq.weixin.com）的账户 id；

-
- app_type : app 类型 ; 取值为 ios 或者 android ; **请兼容大小写** ; 根据广告主在微信广告投放平台 (mp.qq.weixin.com) 创建转化时提交的基本信息关联 ;

3.3 Feedback url 请求响应格式

Json 数据格式

3.4 Feedback url 请求响应内容

{"ret": 返回码, "msg": "错误提示"}

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,0**；否则视为不成功；返回值不能有 302

正确返回格式例子：

```
{  
  
  ret: 0,  
  
  errmsg: "ok"  
  
}
```

4 创建激活转化规则

广告主按文档中接口一指引, 完成 Feedback url 的研发, 然后可以在微信广告投放端 (mp.qq.weixin.com) 广告主-账户管理-工具箱-获取移动应用激活数据, 为移动应用创建转化。

4.1 创建规则入口

- 1) 登录微信广告投放平台 (mp.qq.weixin.com)。



2) 找到推广模块，进入广告主菜单。



3) 在账户管理菜单，点击【获取移动应用激活数据】，进入对应功能页面。

4.2 给应用创建规则（以 iOS 为例）

广告主

广告概览 公众号广告 朋友圈广告 账户管理 公告消息

财务管理 开具发票 银行转账记录 资质管理 工具箱 消息推送

工具箱 / 移动应用激活数据

+绑定移动应用

绑定移动应用，并经过联调后即可获取移动应用广告的激活转化数据 [使用教程](#)

应用名称	应用类型	转化类别	联调状态	转化状态	操作
唯品会-android-注册	Android	注册	联调成功	未启用	启用 编辑 解绑
唯品会-ios-付费	iOS	付费	尚未联调	--	联调 编辑 解绑
唯品会-android-加入购物车	Android	加入购物车	联调成功	未启用	启用 编辑 解绑
唯品会-ios-激活	iOS	激活	尚未联调	--	联调 编辑 解绑
唯品会-android-激活	Android	激活	联调成功	启用中	暂停转化 编辑 解绑
携程旅行-付费	iOS	付费	尚未联调	--	联调 编辑 解绑
达令全球好货	Android	付费	联调失败	--	联调 编辑 解绑

1) 点击【+绑定移动应用】按钮，开始创建转化规则，可以选择 iOS 或者 Android 应用类型。

应用信息

1. 填写信息

2. 获取开发密钥

应用类型

☒ iOS应用

☐ Android应用

应用信息

Apple ID

请输入Apple ID

校验

[如何获取Apple ID?](#)

应用名称

未获取应用名称

转化类别

激活

激活

注册

付费

加入购物车

Feedback URL

请输入Feedback URL

填写后，可用

请求信息

下一步

取消

填写需要创建数据转化规则的 Apple ID 校验，选择转化类别，并填写按此文档的接口一指引所研发完成的 Feedback url。

应用信息

1. 填写信息

2. 获取开发密钥

应用类型

☒ iOS应用

☐ Android应用

应用信息

Apple ID

417200582

校验

[如何获取Apple ID?](#)

应用名称

唯品会 - 全球精选 正品特卖

转化类别

激活

Feedback URL

np/data/collector?monitor_spot_cpde=qusmv14c&app_t

填写后，可用于接收广告点击的请求信息

下一步

取消

2) 填写如上图所示,确认校验正确的 Apple ID 及正确的 Feedback url ,点击下一步。

iOS应用信息

1. 填写信息

2. 获取开发密钥

encrypt_key

BAAAAAAAAAAAAANQv

sign_key

0ef10b563a8e32f3

广告主需要进一步完成接口的开发工作，并进行联调测试，才能统计激活数据。

[使用教程](#)

上一步

确定

3) 获得开发密钥 encrypt_key 及 sign_key，将用于转化数据上报接口：点击数据关联转化上报，详情可继续见文档下文。

点击确认，即完成该应用的激活转化数据规则创建。

4.3 已创建的规则列表

应用名称	应用类型	转化类别	联调状态	转化状态	操作
携程旅行-iOS-加入购物车	iOS	加入购物车	尚未联调	--	联调 编辑 解绑
唯品会-android-注册	Android	注册	联调成功	未启用	启用 编辑 解绑
唯品会-iOS-付费	iOS	付费	尚未联调	--	联调 编辑 解绑
唯品会-android-加入购物车	Android	加入购物车	联调成功	未启用	启用 编辑 解绑
唯品会-iOS-激活	iOS	激活	尚未联调	--	联调 编辑 解绑
唯品会-android-激活	Android	激活	联调成功	未启用	启用 编辑 解绑

- 1) 在列表中,可查看到已创建的转化规则,可以对规则进行联调、启动、暂停、编辑、解绑操作。
- 2) 规则需要在完成联调及启用后,才能正式生效。

4.4 联调转化规则

应用名称	应用类型	转化类别	联调状态	转化状态	操作
携程旅行-iOS-加入购物车	iOS	加入购物车	尚未联调	--	联调 编辑 解绑

- 1) 新转化规则创建后,需要进行联调,联调环节是模拟微信广告请求所创建的Feedback url 是否正常被接收。
- 2) 点击联调,即可开始模拟请求联调。

iOS联调

联调设备号

1E2DFA89-496A-47FD-9941-DF1FC4E648

请填写联调设备的idfa码

确定

取消

- 3) 在联调设备号中填入测试或正式所用的IDFA码 (iOS应用) 或IMEI码 (Android)

应用)，点击确认，进行自助联调。联调设备号示例：

- iOS：1E2DFA89-496A-47FD-9941-DF1FC4E6484A；
- Android：354649050046412

4) 注意：联调时需要确保 feedback url 可以访问，否则联调失败；

应用名称	应用类型	转化类别	联调状态	转化状态	操作
唯品会-android-注册	Android	注册	联调成功	未启用	启用 编辑 解绑

5) 联调成功后，联调状态将变为联调成功，对转化状态操作启用。

4.5 启动转化规则

应用名称	应用类型	转化类别	联调状态	转化状态	操作
唯品会-android-注册	Android	注册	联调成功	未启用	启用 编辑 解绑

1) 当转化规则联调成功后，需要启用才能正式生效，微信广告系统才会开始进行点击信息请求。

2) 点击启用，即完成启用操作，此时转化状态会变为联调成功。

应用名称	应用类型	转化类别	联调状态	转化状态	操作
唯品会-android-注册	Android	注册	联调成功	启用中	暂停转化 编辑 解绑

3) 启用后，即完成整体激活转化规则的创建，微信广告系统将开始把广告点击信息添加到 Feedback url 中，请求给广告主，广告主可观察是否已收到请求。

4.6 编辑转化规则

操作
启用 编辑 解绑

1) 已创建的转化规则，可随时编辑，但编辑后将需要重新进行联调及启用规则。

应用信息

1. 填写信息

2. 获取开发密钥

应用类型

☐ iOS应用

☒ Android应用

应用信息

腾讯开放平台 ID

1101072624

[如何获取腾讯开放平台 ID?](#)

应用名称

唯品会

转化类别

注册

Feedback URL

https://mp.weixin.qq.com/dmp/data/collector?monitor_s

填写后，可用于接收广告点击的请求信息

下一步

取消

2) 编辑仅可对 Feedback url 进行修改，应用信息、转化类别不可修改。

4.7 解绑转化规则

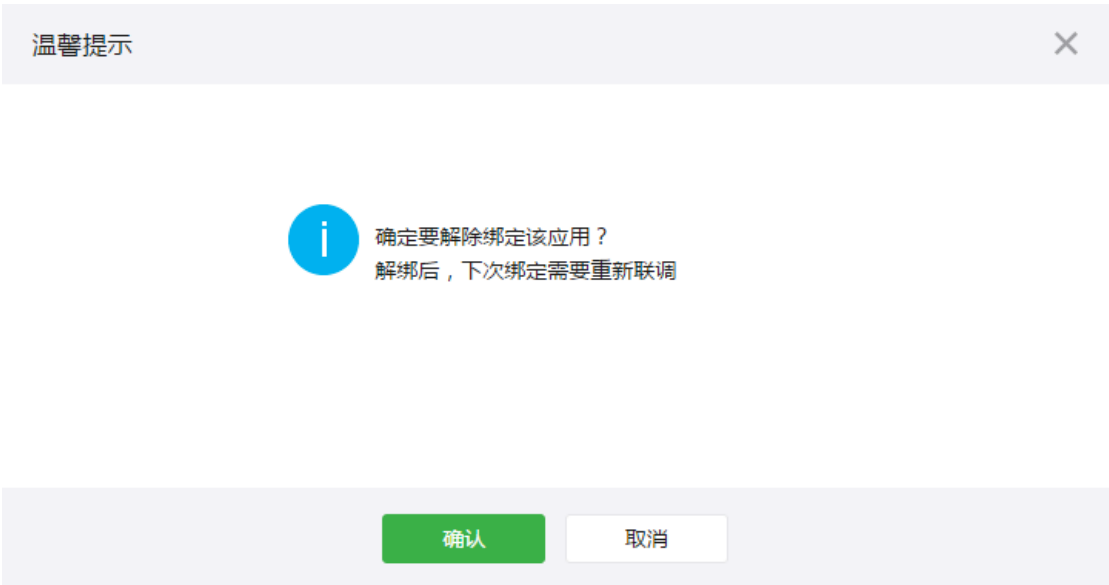
操作

启用

编辑

解绑

1) 已创建的转化规则，可随时进行解绑操作，解绑意味着删除，解绑后将不会再收到该转化规则的请求。



2) 点击解绑，完成二次确认操作，即完成解绑。

5 转化数据上报接口

实现转化数据上报接口，广告主可以将与点击数据进行关联后的转化类型数据上报给微信广告，微信广告会将该转化类型数据与广告关联，方便广告主跟踪广告粒度的转化效果。

上报方式：广告主必须在收到APP发生该种转化类型数据后**实时**将关联后的效果上报给微信广告，以免出现统计无效的情况。

上报方案：

- 对于已经接入微信广告原有iOS应用激活数据上报接口的用户，推荐使用[原有方案](#)继续上报Android应用激活数据，无需重新开发；
- 对于第一次接入微信广告上传应用激活数据的用户，推荐使用[加密简化方案](#)上报iOS或者Android应用的激活数据，新方案简化了加密流程，无需对data进行简单异或加密。

5.1 转化数据接口上报简介（原有方案）

1) 请求格式：

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

2) 参数说明：

- appid：数值，iOS应用在Apple App Store的id，Android为应用宝ID；广告主在微信广告投放平台（mp.qq.weixin.com）创建转化之后，系统会自动生成该id；
- data：为加密的数据结构，字符串，详细描述见本文第7部分；（[V参数加密方案](#)）
- conv_type：为转化类型，枚举值。转化类型与conv_type取值的对应关系如下；

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
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。	MOBILEAPP_ADDTOCART
付费	用户在应用内进行付费，完成了交易的操作行为。	MOBILEAPP_COST

- app_type：为应用类型，枚举值，现阶段有IOS和ANDROID；**注意要大写**；
- uid：数值，广告主在微信广告投放平台（mp.qq.weixin.com）的账户id；广告主在微信广告投放平台（mp.qq.weixin.com）创建转化之后，系统会自动生成该

id，可在收到的点击信息请求中获得；

3) 请求方式：

HTTP GET方法

4) 相应格式：

Json 数据格式

5) 相应内容：

{"ret": 返回码, "msg": "错误提示"}

返回码为0标识正常接收，其他返回码标识错误。

返回码	说明
0	成功
-1	请求非法参数
-2	参数解析失败
-3	参数解码失败
-12	获取密钥失败
-13	非法的应用类型
-14	非法的转化时间
-15	非法的移动设备标识
-17	获取转化规则失败

5.2 转化数据接口上报简介（加密简化方案）

1) 接口格式：

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

2) 请求方式：

HTTP POST方法

3) 参数处理流程：

- 选取特定参数拼接成字符串后进行md5,输出32位小写字符串
- 使用x-www-form-urlencoded方式构造post参数。

4) 拼接参数介绍：

- app_type：字符串类型，app类型; iOS应用取值为IOS，Android 应用取值为ANDROID，**建议为大写（已兼容大小写）**
- click_id：字符串类型，点击ID，来自发生转化行为（**例如激活**）的设备muid与微信广告转发点击信息的匹配结果
- client_ip：字符串类型，转化发生用户的客户端IP，可以为空
- conv_time：字符串类型，转化发生时间，取值为标准时间戳，秒级别
- muid：字符串类型，设备id，由IDFA（iOS应用）md5生成，或是由IMEI（Android应用）md5生成；具体加密方案在[Muid加密方案](#)有详细说明
- sign_key：字符串类型，广告主在微信广告投放平台（mp.qq.weixin.com）创建转化规则之后，系统会自动生成密钥。

参数处理规则

按顺序拼接成一个字符串，最终输出字符串：

```
app_type=IOS&click_id=007210548a030059ccdfd1d4&client_ip=10.11.12.13&
conv_time=1422263664&muid=0f074dc8e1f0547310e729032ac0730b&sign_k
ey=08ebe39d34c421b8
```

顺序需与示例保持一致。如果有的字段值为空，也需要填写，保证是这6个参数。例如

client_ip为空，字符串为：

```
app_type=IOS&click_id=007210548a030059ccdfd1d4&client_ip=&conv_time=1422263664&muid=0f074dc8e1f0547310e729032ac0730b&sign_key=08ebe39d34c421b8
```

对字符串md5示例

示例字符串：

```
app_type=IOS&click_id=007210548a030059ccdfd1d4&client_ip=10.11.12.13&conv_time=1422263664&muid=0f074dc8e1f0547310e729032ac0730b&sign_key=08ebe39d34c421b8
```

md5输出为：

```
5494af8f21f4083c5fcea60105c91253
```

5) 构造 post 参数

- click_id //字符串类型，点击ID（必填）

来自发生转化行为（例如激活）的设备muid与微信广告转发点击信息的匹配结果

- muid //字符串类型，设备id（必填）

由iOS应用在Apple App Store的id，或是由IMEI（Android应用）md5生成；具体加密方案在[muid加密方案](#)有详细说明

- appid //整型，应用ID，iOS应用在Apple App Store的id，Android为应用宝ID（必填）
- conv_time //字符串类型，转化发生时间取值为标准时间戳，秒级别（必填）
- client_ip //字符串类型，转化发生用户的客户端IP（选填）

- encstr //字符串类型，上一步生成的MD5（必填）
- encver //字符串类型，加密版本，目前为1.0（必填）
- advertiser_id //整型，广告主在微信广告的账户id（必填）
- app_type //字符串类型，app类型;iOS应用取值为IOS，Android 应用取值为 ANDROID，**建议为大写（已兼容大小写）**
- conv_type //字符串类型，转化行为标记参数

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
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。	MOBILEAPP_ADDTOCART
付费	用户在应用内进行付费，完成了交易的操作行为。	MOBILEAPP_COST

例如：

clid_id: 不填

appid: 112233

muid: 0f074dc8e1f0547310e729032ac0730b

conv_time: 1422263664

client_ip: 10.11.12.13

encstr: d96ce45044a716af56db404ecd3363b7

encver: 1.0

advertiser_id: 20345

app_type:IOS

conv_type: MOBILEAPP_ACTIVITE

输出为：

最终请求：

<https://t.gdt.qq.com/conv/app/112233/conv>

post body:

click_id=&appid=112233&muid=0f074dc8e1f0547310e729032ac0730b&c
onv_time=1422263664&client_ip=10.11.12.13&encstr=d96ce45044a716af
56db404ecd3363b7&encver=1.0&advertiser_id=20345&app_type=IOS&c
onv_type=MOBILEAPP_ACTIVITE

如需上报『加入购物车』和『付费』的金额，则参考如下示例（付费）：

post body:

click_id=&appid=112233&muid=0f074dc8e1f0547310e729032ac0730b&c
onv_time=1422263664&client_ip=10.11.12.13&encstr=d96ce45044a716af
56db404ecd3363b7&encver=1.0&advertiser_id=20345&app_type=IOS&c
onv_type=MOBILEAPP_COST&value=100

6 附：Muid 加密方案

muid：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++)：

```
bool GenerateMuid(  
    const std::string& uni_id, const int32_t app_type, std::string* muid) {  
    if (muid == NULL || uni_id.empty()) {  
        return false;  
    }  
    muid->clear();  
    std::string tmp;  
    if (app_type == kAppTypeAndoirdImei) {  
        LatinToLower(uni_id, &tmp);  
    } else if (app_type == kAppTypeiOSIfa) {  
        LatinToUpper(uni_id, &tmp);  
    } else {  
        return false;  
    }  
    md5sum32l(tmp, muid); // 32bit lower  
    return true;  
}
```

测试用例：

iOS设备号加密测试用例：

原始IDFA码：1E2DFA89-496A-47FD-9941-DF1FC4E6484A

加密之后：40c7084b4845eebce9d07b8a18a055fc

Android 设备号加密测试用例：

原始 IMEI 号：354649050046412

加密之后：b496ec1169770ea274a2b4f42ca4fb71

7 附：V 参数加密方案

整个加密方案分为了四个部分：**组装参数、参数签名、参数加密、组装请求。**

而对于每一个广告主账户我们会分配一个加密密钥 `encrypt_key` 和一个签名密钥 `sign_key`：

加密密钥 `encrypt_key` 和签名密钥 `sign_key` 获取方式：广告主在微信广告投放平台（`mp.qq.weixin.com`）创建转化规则之后，系统会自动生成密钥。

加密密钥 `encrypt_key` 和签名密钥 `sign_key` 粒度：广告主，即：对一个广告主账户下，不同的 APP、不同的转化规则，都会生成相同的一组密钥。

1) 组合参数

首先需要获取下列取值：

- `click_id` // 微信广告点击跟踪 ID（必选）

微信广告系统中标识用户每次点击生成的唯一标识；

- `muid` // 微信广告点击设备标识 ID（必选）

用户设备的 IMEI 或 idfa 进行 MD5SUM 以后得到的 32 位全小写 MD5 表现字符串；

- `conv_time` // 转化发生时间（必选）

转化发生的时间的标准时间戳, **秒级别, 不能是毫秒**; 由于转化效果统计有时间期限限制, 目前系统要求: 转化与点击时间间隔最长为 5 天

- client_ip // 转化发生 IP (可选) 转化发生用户的客户端 IP, 选填;

正确取值以后, 我们将上述的内容按照下列方式拼接成一个合法的 query_string :

{key1}=urlencode({value1})&{key2}=urlencode({value2})

注:

1. 此处如果不填写 client_ip, 可以直接在 query_string 中去除该参数;
2. 此处组合参数无顺序要求。

例:

click_id : 007210548a030059ccdfd1d4

muid: 0f074dc8e1f0547310e729032ac0730b

conv_time: 1422263664

client_ip: 10.11.12.13

变为

click_id=007210548a030059ccdfd1d4&muid=0f074dc8e1f0547310e72
9032ac0730b&conv_time=1422263664&client_ip=10.11.12.13

2) 参数签名

按照上述组合参数的规则组成了一个完整的 query_string 串以后, 此时我们可以得到一个基础的请求结构, 我们先按照如下结构组成一个新的字符串 page :

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

例：

appid: 112233

query_string:

click_id=007210548a030059ccdfd1d4&muid=0f074dc8e1f0547310e72

9032ac0730b&conv_time=1422263664&client_ip=10.11.12.13

变为

http://t.gdt.qq.com/conv/app/112233/conv?click_id=007210548a0300

59ccdfd1d4&muid=0f074dc8e1f0547310e729032ac0730b&conv_time

=1422263664&client_ip=10.11.12.13

通过得到的 page 字符串，我们进行整体 urlencode 得到 encode_page，然后按照下述描述组装成新的字符串 property。

{sign_key}&GET&{encode_page}

例：

sign_key: test_sign_key

变为：

test_sign_key&GET&http%3A%2F%2Ft.gdt.qq.com%2Fconv%2Fapp%2

F112233%2Fconv%3Fclick_id%3D007210548a030059ccdfd1d4%26mui

d%3D0f074dc8e1f0547310e729032ac0730b%26conv_time%3D142226

3664%26client_ip%3D10.11.12.13

我们对 property 进行 md5sum 获得 32 位的小写加密串即为 signature。

例：

property:

test_sign_key&GET&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

变为

c2f87710541942364691e5e8adc84f3f

3) 参数加密

按照上述组合参数的规则组成的 query_string 以及参数签名得到的 signature 组合，通过下述方式得到新的 base_data。

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

例：

query_string:

click_id=007210548a030059ccdfd1d4&muid=0f074dc8e1f0547310e729032ac0730b&conv_time=1422263664&client_ip=10.11.12.13
signature: c2f87710541942364691e5e8adc84f3f

变为

click_id=007210548a030059ccdfd1d4&muid=0f074dc8e1f0547310e729032ac0730b&conv_time=1422263664&client_ip=10.11.12.13&sign=c2f87710541942364691e5e8adc84f3f

通过上一个步骤得到的 `base_data` , 我们与 `encrypt_key` 进行简单异或 , 最终得到 `base64` 表达形式就是 `data` (注意 : 不能分页)。此时按照接口约定中的请求格式向微信广告这边发送请求。

`base64(simple_xor({base_data}, {encrypt_key}))`

注 : `base64` 编码后 , 需要去除换行符再进行 URL 编码

简单异或加密代码示例 (C++):

```
bool SimpleXor(
    const std::string& info,
    const std::string& key,
    std::string* result) {
    if (result == NULL || info.empty() || key.empty()){
        return false;
    }
    result->clear();
    uint32_t i = 0;
    uint32_t j = 0;
    for (; i < info.size(); ++i) {
        result->push_back(static_cast<unsigned char>(info[i] ^ key[j]));
        j = (++j) % (key.length());
    }
    return true;
}
```

简单异或加密代码示例 (Python):

```
这边有一个python的示例代码可以看下
def SimpleXor(source, key):
    retval = ""
    j = 0
    for ch in source:
        retval = retval + chr(ord(ch)^ord(key[j]))
        j = j + 1
        j = j % (len(key))
    return retval
```

例：

base_data:

click_id=007210548a030059ccdfd1d4&muid=0f074dc8e1f0547310e729032ac0730b&conv_time=1422263664&client_ip=10.11.12.13&sign=c2f87710541942364691e5e8adc84f3f

encrypt_key: test_encrypt_key

变为

FwkaFzQ6BwdPSUBDbVpVTEBdEkRsVV5WSxoTEDkPVB1AQx4BNgFTUxRJR0A7CF0cRQNDQWtSXVJCHEdGZltWSxUGQ0NsVQxFERYeAgAfDBQRWEJAbVdcVUFPRkB5CAkQEqsHKzYVU1JCV0FFcVpXV0VWVQc2AgBeEUsWTGhcVEIBUUJNa1ddVUZPSUU6XgBBFQEQTGsDXQU=

4) 组装请求：获取以下参数取值。

- conv_type // 转化类型（必选）

转化类型	conv_type 取值
------	--------------

激活	MOBILEAPP_ACTIVITE
注册	MOBILEAPP_ACTIVITE
加入购物车	MOBILEAPP_ADDTOCART
付费	MOBILEAPP_COST

- app_type // 转化应用类型（必选）

转化发生的 APP 类型，**iOS 应用取值为 IOS，Android 应用取值为 ANDROID**

- advertiser_id // 广告主 ID（必选）

广告主在微信广告系统中的广告主标识 ID

正确取值以后，我们将上述的内容按照下列方式拼接成一个合法的 attachment。

{key1}=urlencode({value1})&{key2}=urlencode({value2})

例：

conv_type: MOBILEAPP_ACTIVITE

app_type: IOS

advertiser_id: 10000

变为

conv_type=MOBILEAPP_ACTIVITE&app_type=IOS&advertiser_id=1000

0

获取到正确的 attachment 以后，我们便可以按照下列形式，拼接出最后的请求。

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

最终请求的形式为：

```
http://t.gdt.qq.com/conv/app/112233/conv?v=FwkaFzQ6BwdPSUBDb  
VpVTEBdEkRsVV5WSxoTEDkPVB1AQx4BNgFTUxRJR0A7CF0cRQNDQW  
tSXVJCHEdGZltWSxUGQ0NsVQxFERYeAgAfDBQRWEJAbVdcVUFPRkB5  
CAkQEQsHKzYVU1JCV0FFcVpXV0VWVQc2AgBeEUsWTGhcVEIBUUNa  
1ddVUZPSUU6XgBBFQEQTGsDXQU%3D&conv_type=MOBILEAPP_ACT  
IVITE&app_type=IOS&advertiser_id=10000
```

8 附：V 参数二次 urlencode

根据 V 参数加密方案处理得到最终请求的形式后，需要对 V 参数整体再做 urlencode 处理，才可以正常的上报请求。

将以下部分做 urlencode：

```
v=FwkaFzQ6BwdPSUBDbVpVTEBdEkRsVV5WSxoTEDkPVB1AQx4BNgFTUxRJR  
0A7CF0cRQNDQWtSXVJCHEdGZltWSxUGQ0NsVQxFERYeAgAfDBQRWEJAbVd  
cVUFPRkB5CAkQEQsHKzYVU1JCV0FFcVpXV0VWVQc2AgBeEUsWTGhcVEIBUU  
JNa1ddVUZPSUU6XgBBFQEQTGsDXQU%3D
```

9 联系我们

若对接入有任何疑问，可关注微信广告助手公众号联系我们。



微信广告助手 (ad_helper)