OPPO广告主转化数据接入说明

**方案设计**

[1、 背景 2](#_Toc28021)

[2、 接口说明 2](#_Toc6643)

[2.1 转化数据同步接口 2](#_Toc2366)

[2.2 签名算法 3](#_Toc19258)

[2.3 加密算法 4](#_Toc19228)

1. **背景**

OCPC智能出价是OPPO广告系统通过优化转化率预估模型，帮助广告主提升推广精准度，降低转化成本的一种投放方式；而转化率的预估模型，需要广告主回传相关转化数据；该接口文档主要描述了接口的定义，对接方式以及相关注意事项。

1. **接口说明**

接口采用 https + post + json的方式提交数据;

编码方式统一使用UTF-8;

Salt: e0u6fnlag06lc3pl

Base64Key: XGAXicVG5GMBsx5bueOe4w==

**2.1 转化数据同步接口**

接口地址：https://api.ads.heytapmobi.com/api/uploadActiveData

请求信息

请求头(header):

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **说明** | **是否必须** |
| signature | string | 根据签名算法计算得到的字符串，见签名算法2.2 | 是 |
| timestamp | string | 时间戳转换为字符串（long毫秒） | 是 |

数据字段(post body)：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **说明** | **是否必须** |
| imei | string | 客户端imei经过AES加密后的值,加密说明见2.3 | **三选一** |
| ouId | string | 开发者ID经过AES加密后的值,加密说明见2.3 |
| androidId | string | 安卓ID经过AES加密后的值,加密说明见2.3 |
| mac | string | 客户端mac经过AES加密后的值,加密说明见2.3 | 否 |
| clientIp | string | 事件发生时的客户端ip，如210.210.210.210 | 是 |
| timestamp | long | 事件发生的时间戳(毫秒)，如1522221766623 | 是 |
| pkg | string | 包名，如com.xxx | 是 |
| dataType | int | 转化数据类型：1、激活，2、注册，3、付费，4、次留，7、自定义目标 | 是 |
| customType | int | 自定义目标类型：dataType填了7之后补充，枚举值与客户沟通后补充 | 否 |
| channel | int | 渠道：1、OPPO，2、一加，0、其他 | 是 |
| type | int | Imei原始加密类型  1： md5加密 0： 无加密（默认为0） | 否 |
| appType | int | 应用类别：1应用 2游戏 3快应用 0其他，默认1应用 | 否 |
| payAmount | long | 付费金额（单位：分*）* | 否 |
| ascribeType | int | 归因类型：1:广告主归因，0：OPPO归因（默认或者不填即为0） | 否 |
| adId | long | 广告主回传转化数据时，附带已经归因好的广告id | 否 |
| requestId | string | 请求id | 否 |

请求示例:

imei:123456789 ===AES加密===> mWxEB4Il3Q3MsW0/gVPqSw==

mac:123456 ===AES加密===> 13tTYyshcNalBAxdvreL6A==

timestamp:1571995483916

clientIp:111.111.111.111

pkg:com.xxx.xxx

payAmount:0

appType:1

dataType:1

channel:1

type:0

请求报文：

(1)header：

signature: 67843bc0e7e7b09cc369beabf05e9d30

timestamp: 1571995483916

(2)body：

{"payAmount":0,"appType":1,"clientIp":"111.111.111.111","dataType":1,"channel":1,"imei":"mWxEB4Il3Q3MsW0/gVPqSw==","type":0,"pkg":"com.xxx.xxx","mac":"13tTYyshcNalBAxdvreL6A==","timestamp":1571995483916}

返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **说明** | **是否必须** |
| ret | int | 返回码，0正常，1001参数校验失败，1002未知错误  注意：如果是签名错误的话会返回http status 403 | 是 |
| msg | string | 如果有错误，则返回错误信息 | 否 |

返回示例：

{

"ret":0

}

**2.2 签名算法**

使用md5对数据进行签名: md5( postData + timestamp + salt)。

timestamp: 时间戳

postData: post的JSON格式数据

salt: OPPO提供

如当前时间戳为1522221766623, post的json字符串（不要格式化）为{"imei":"123456789"},OPPO提供的salt字符为oppo\_salt\_demo，则

{"imei":"123456789"}1522221766623oppo\_salt\_demo

=>

"b757ea95cbba030ebf43520b14a3229b"

**注：签名后结果需转为小写**

**2.3 加密算法**

请求中的imei与mac字段均需通过AES算法进行加密，并且将加密后的二进制数据通过base64编码为字符串。为了方便配置，示例中的密钥是经过base64处理后的字符串(base64Key)。base64Key在接入时由OPPO提供。

**import** java.security.GeneralSecurityException;

**import** java.security.Key;

**import** javax.crypto.Cipher;

**import** javax.crypto.spec.SecretKeySpec;

**import** org.apache.commons.codec.binary.Base64;

**public** **class** Demo {

**public** **static** String encode(**byte**[] data, String base64Key) **throws** GeneralSecurityException {

**final** Key dataKey = **new** SecretKeySpec(Base64.*decodeBase64*(base64Key), "AES");

Cipher cipher = Cipher.*getInstance*("AES/ECB/PKCS5Padding");

cipher.init(Cipher.***ENCRYPT\_MODE***, dataKey);

**byte**[] encryptData = cipher.doFinal(data);

**return** Base64.*encodeBase64String*(encryptData);

}

}

php加密示例（仅供参考）：

public function encrypt($input, $base64Key){

return openssl\_encrypt($input, ‘AES-128-ECB’,base64\_decode($base64Key),0,’’);

}

1. **其他说明**

**3.1 签名校验说明**

若签名校验失败，直接返回http status code : 403。