

IOS 积分墙激活回调接口

1. 使用说明

本接口主要用来对开发者接入积分墙SDK后获取激活回调信息的。

- a) 协议:HTTP GET
- b) 编码:UTF-8
- c) 地址:通过开发者后台进行配置。
- d) key:在设置回调地址后会生成。注:管理程序→积分回调设置→服务器端回调

虚拟货币设置 | 积分回调设置 | 过滤URL设置 | 二次确认设置

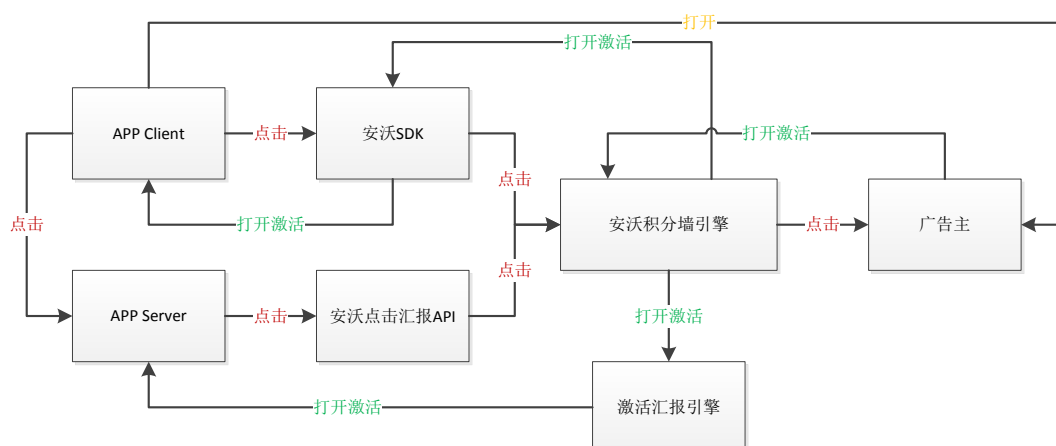
本功能设置仅对积分墙类广告形式有效, 设置积分回调方式。

积分墙回调方式: ☒ 客户端回调 ☐ 服务器端回调

应用市场URL: 服务器地址: (示例: http://www.adwo.com/)

确认提交

2. 调用时序图



文字说明:

- a) 应用客户端通过安沃SDK或安沃点击汇报API来通知安沃积分墙引擎打开广告动作。
- b) 安沃积分墙引擎会把点击请求发送给广告主。
- c) 应用客户端在打开激活下载的广告后, 广告主会把打开激活的信息回传给安沃积分墙引擎。
- d) 我积分墙引擎会直接返回给SDK或通过激活汇报引擎把相关信息汇报给应用服务端。

注:SDK及SDK文档请到官网下载, 安沃点击汇报API及激活汇报API请找安沃工作人员索取。

3. 安全签名认证

签名算法是基于 MD5 小写 32 位 16 进制字符串

- a) 所有参数/值名称按照字母升序排列, 并拼接
- b) 最后将 key 拼接并进行 MD5

Signstring = MD5("k1=v1k2=v2k3v3.....kn=vnkey=value")

注:key在设置回调地址后会生成。

4. 接口列表

a) 授予积分

建议接口地址: /ad/adpoint		
参数说明	appid	开发者应用 ID: 如 kc 电话对应的 ID
	adname	下载应用名称
	adid	下载应用 ID
	device	mac (mac无冒号)
	idfa	idfa
	point	赠送积分, 整型
	ts	时间戳(时间戳+device/idfa为激活回调唯一标识, 可联合做流水号使用, 开发者需要通过此字段联合device/idfa来判断回调是否重复)
	sign	数字签名
特殊参数	keyword	关键字 注: 不做为签名条件, 参数值由开发者自己协定 (SDK 实例: NSArray *array = [NSArray arrayWithObjects:@"keyword1", nil]; 接收到的字符串为: "keyword1")
处理成功 (服务器返回状态码为200即为回调成功, 其他状态码则重复发送)		

5. 加密拼装实例:

adid=valueadname=valueappid=valuedevice=valueidfa=valuepoint=valuets=timekey=audc28ls

加密后字符串(以上面拼装实例为例):

aadb5aec576076313cc612252b70d779

6. 常见问题 (Q&A)

Q: 是否可以传递或回调自定义的参数 (如userid等)?

A: 可以通过SDK注入参数或API 的keywords参数 (详见具体文档) 来进行自定义参数的传递, 我们在接到激活动作的时候keyword这个参数我们会原封不动的给回传回去。

Q: 如果要传多个自定义参数怎么办?

A: 因为keyword参数是字符串, 所以开发者可以把多个参数拼成字符串后传递给我们, 在我们回传回去的时候按照自己的规则进行解析即可, 如 (param1, param2, param3或 { paramName1: paramValue1, paramName1: paramValue2})

Q: 安沃激活汇报的发送机制是怎样的?

A: 我们会在接到广告主打开激活确认数据的第一时间给开发者返回回去, 如果开发者

的服务器状态为200我们 即认为是成功，如果为其他状态码如500、502、404我们即认为是发送失败，我们会把此次请求重新加入到发送队列，重新发送。

Q: 为什么加密后的签名和参数中sign值对不上?

A: 请仔细确认参数排序是否正确，特殊参数keyword是不参与加密条件的，是否在参数按规定顺序拼接好后没有在最后拼接key=xxx加密key。

Q: 如果我们想判定激活是否重复激活?

A: 返回参数中的ts参数可以作为订单编号，此数值为激活的唯一标识，多次发送也不会发生变换。

JAVA版加密算法

```
private static String MD5(String plain) {
    StringBuffer buf = new StringBuffer("");
    try {
        MessageDigest md =
MessageDigest.getInstance("MD5");
        md.update(plain.getBytes());
        byte b[] = md.digest();
        int i;
        for (int offset = 0; offset < b.length; offset++) {
            i = b[offset];
            if (i < 0) {
                i += 256;
            }
            if (i < 16) {
                buf.append("0");
            }
            buf.append(Integer.toHexString(i));
        }
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }
    return buf.toString();
}
```