# 单机SDK接入服务端说明书

## 支付成功服务器通知参数定义

### 接口描述

支付成功后，魅族SDK服务器异步通知CP服务器，CP需要提供回调URL，填写到开发者社区支付成功回调地址。在服务器异步通知参数列表中，除去sign、sign\_type两个参数外，凡是通知返回回来的参数皆是要验签的参数,为了应对字段扩展，接入业务验签时，通过request.getParameters()获取所有参数。

### 接口URL和参数说明

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 查询版本号 | | 请求地址 | 由游戏开发者配置 | | |
| 调用方式 | http post | | |
| 请求方 | 魅族SDK服务器 | | |
| 响应方 | 游戏服务端 | | |
| 同步or异步 | 同步 | | |
| 请求内容： | | | | | |
| 字段名称 | 字段类型 | | | 必填 | 字段说明 |
| notify\_time | String | | | Y | 通知时间：yyyy-MM-dd HH:mm:ss |
| notify\_id | String(32) | | | Y | 通知id |
| order\_id | Long | | | Y | 订单id |
| app\_id | Long | | | Y | 应用id |
| uid | String | | | Y | 用户id，如果uid为空，则用udid |
| partner\_id | long | | | Y | 商户id |
| cp\_order\_id | String | | | Y | 游戏订单id |
| product\_id | String | | | Y | 产品id |
| buy\_amount | int | | | N | 购买数量 |
| total\_price | float | | | Y | 购买总价 |
| trade\_status | String | | | Y | 交易状态：  1：待支付（订单已创建）  2：支付中  3：已支付  4：取消订单  5：未知异常取消订单 |
| create\_time | Date | | | Y | 订单时间 |
| pay\_time | Date | | | Y | 支付时间：CP需要保存该时间进行后续的对账，对账时间以支付成功时间为统计 |
| pay\_type | int | | | N | 支付类型：1 不定金额充值，0 购买 |
| user\_info | String | | | N | 用户自定义信息 |
| sign | String | | | Y | 参数签名 |
| sign\_type | String | | | Y | 签名类型，常量md5 |

### 响应内容

CP发货成功/失败后，需要响应魅族的回调

|  |  |  |  |
| --- | --- | --- | --- |
| 响应内容： | | | |
| 协议：http + json | | | |
| 字段名称 | 字段类型 | 必填 | 字段说明 |
| code | String | Y | 200成功发货  120013尚未发货  120014 发货失败  900000未知异常 |
| message | String | N |  |
| value | String | N |  |
| redirect | String | N |  |

示例：response.getWriter().print("{\"code\":200}");

## 订单查询服务器接口

### 接口描述

CP游戏服务器收到SDK服务器支付成功通知后，如果CP服务器没加支付通知IP白名单，为了安全考虑，CP应该再去查一次订单状态，确保订单状态和金额的一致性。

目标：该接口保证，在支付成功并回调CP后，通过该接口查询出来的状态和支付成功回调时的订单状态一致。

### 接口URL和参数说明

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 查询订单接口 | | 请求地址 | https://api.game.meizu.com/game/order/v3/query | | |
| 调用方式 | http post | | |
| 请求方 | CP服务器 | | |
| 响应方 | 游戏中心 | | |
| 同步or异步 | 同步 | | |
| Content-Type | application/x-www-form-urlencoded; charset=UTF-8 | | |
| 请求内容: | | | | | |
| 字段名称 | 字段类型 | | | 必填 | 字段说明 |
| app\_id | long | | | Y | 游戏id |
| order\_id | String | | | Y | 魅族SDK服务器生成的订单号（全局唯一） |
| ts | long | | | Y | Unix\_timestamp.eg: 1396424644 |
| sign\_type | String | | | Y | 支持MD5 |
| sign | String | | | Y | 签数签名 |

### 响应内容

在收到响应数据时，需要对value对象里所有的key-value进行按消息签名规则进行拼接，进行MD5签名进行验证。以后可能扩展其他key-value，所以必须遍历所有的属性进行拼接（除了sign\_type和sign）

|  |  |  |  |
| --- | --- | --- | --- |
| 响应内容： | | | |
| 协议：http+json | | | |
| 字段名称 | 字段类型 | 必填 | 字段说明 |
| **公共参数** | | | |
| code | Int | Y | 错误代码 |
| message | String | Y | 错误信息 |
| **value业务数据** | | | |
| appId | long | Y | 游戏Id |
| cpOrderId | String | Y | 游戏订单Id |
| orderId | String | Y | SDK服务器订单Id |
| partnerId | String | Y | 开发者Id |
| uid | long | Y | 用户Id |
| productId | String | Y | 产品Id |
| productSubject | String | Y | 定单标题 |
| productBody | String | Y | 产品详情 |
| buyAmount | Int | Y | 购买数量 |
| totalPrice | Double | Y | 总价 |
| tradeStatus | Int | Y | 交易状态，1：待支付（订单已创建）2：支付中3：已支付4：取消订单5：未知 |
| createTime | Long | Y | 创建时间 |
| payTime | Long | Y | 支付时间 |
| deliverStatus | Int | Y | 发货状态，1：支付未完成，2：待发货：3：已发货，4：发货失败 |
| imei | String | Y | 设备IMEI |
| udid | String | Y | 设备唯一标识 |
| guid | String | Y | 游戏账号唯一标识 |
| sign | String | Y | 参数签名 |

## 签名与验签

魅族SDK服务器要求每次http请求都要对关键信息进行签名（防止参数值篡改），通信对端接收到消息后，也必须对消息进行验签。目前魅族SDK服务器仅支持MD5签名算法，交互过程中，该字段sign\_type仅能取值：MD5。签名密钥：appSecret（从魅族游戏开发者社区获取appid、appkey、appSecret）

## 消息签名

1. 签名算法：

目前仅支持MD5

2. 拼接参数串：

指消息中除“签名类型（sign\_type）”和“参数签名(sign)”外的所有属性。

3. 被签名字符串：

指拼接参数串和合作密钥信息的拼接结果，拼接方式如下所示：

key1=value1&key2=value2&...&keyn=valuen:appsecret

4. 签名结果：

对被签名字符串做签名运算的结果，即：

MD5签名：sign= md5\_hex(key1=value1&key2=value2&...&keyn=valuen:appsecret)

5. 注意事项：

（1）被签名字符串中的关键信息需要按照key值做升序排列；

（2）空值（null值）不参与签名运算；

（3）将被签名字符串转成字节数组时必须指定编码为utf-8。

（4）拼接的value值不能进行URL Encode

6. 返回的应答或通知消息

返回的应答或通知消息可能会由于升级增加参数，请验证应答签名时注意允许这种情况。

举例： 调用某个接口，接口有如下字段： app\_id、total\_price、product\_id、user\_info、uid

实际调用接口时，各字段的值：

app\_id=996000，total\_price=1，product\_id=comxxmz123，user\_info=test&123，uid=,

正确的签名原始串是：

app\_id=996000&product\_id=comxxmz123&total\_price=1&uid=&user\_info=test&123

常见的错误有：

示例1：app\_id=996000&product\_id=comxxmz123&total\_price=1&uid=&user\_info=test%26123

(user\_info的值做了URL Encode处理)

示例2：app\_id=996000&total\_price=1&uid=&user\_info=(product\_id参数未包含)

示例3：app\_id=996000&product\_id= comxxmz123&total\_price=1&user\_info=test&123&uid= (key=value未按照ascii排序)

MD5签名计算公式：

sign = Md5(拼接参数串:游戏密钥).toLowerCase()

如：

签名原始串是：

app\_id=996000&product\_id=comxxmz123&total\_price=1&uid=&user\_info=test&123

商户密钥是：8934e7d15453e97507ef794cf7b0519d

签名的结果为：

sign=md5(app\_id=996000&product\_id=comxxmz123&total\_price=1&uid=&user\_info=test&123:8934e7d15453e97507ef794cf7b0519d)= dfd52bb168e5781f5baf6cec1767862f