

# 魅族单机游戏 SDK 接入手册

V 1.3.0

版本	日期	说明
1.0	2014/5/13	初稿完成
1.2.0	2016/7/27	1、免登陆支付
1.3.0	2016/8/19	1、修复部分机型兼容问题

# 目 录

一、 环境准备.....	3
1.1. 接入须知.....	3
1.2. 接入所需参数说明 .....	3
二、 客户端接入说明.....	3
4.1.流程概述.....	3
4.2. SDK 接入方法.....	4
4.2.1. 开发环境配置.....	4
4.2.2. SDK 初始化.....	5
4.2.3. 补单调用代码.....	6
4.2.5. 支付调用代码.....	6



# 一、环境准备

## 1.1. 接入须知

- 弱联网游戏：不需要账号系统
- 游戏启动时不需要验证用户数据，用户进度数据保存在手机本地；
- 所有联运手游接入时的 package 名称必须以“.mz”结尾。
- 游戏提交到开发者平台，由开发者平台以魅族私钥重新签名，在魅族开发者平台中测试并发布。

## 1.2. 接入所需参数说明

参数	参数说明	参数说明
appID	游戏 ID	客户端和服务端接入时需要配置
appKey	游戏 key	客户端接入时需要配置
appSecret	密钥	签名时需要用到的密钥，不能泄露

上述参数请在魅族开发者平台中获取。

# 二、客户端接入说明

## 4.1. 流程概述

魅族游戏单机 SDK 主要提供两方面核心功能：支付和补单。游戏每次启动，都需要先调用补单 orderQueryConfirm (this,this)，补单接口主要是解决用户在上次支付成功，道具未进行实时发放的情况。补单成功后，会回调给游戏进行处理。注：游戏必须实现 MzPayListener 接口

## 4.2. SDK 接入方法

### 4.2.1. 开发环境配置

#### 4.2.1.1. 加入 SDK 的 jar 文件

- 1、 将提供的 MzOfflineGameSDK\_xxx.jar 放到游戏工程的 libs 目录。

#### 4.2.1.2. 配置 AndroidManifest 声明

魅族游戏 SDK 需要使用到以下权限，请在工程的 AndroidManifest.xml 中加入权限声明：

```
<activity
    android:name="com.meizu.atlas.server.am.ProxyActivity"
    android:configChanges="screenLayout|screenSize|orientation|keyboardHidden"
    android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen">
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />

        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />

        <data
            android:host="alipay"
            android:scheme="${applicationId}" />
    </intent-filter>
</activity>

<activity
    android:name="com.meizu.atlas.server.am.OrientationLProxyActivity"
    android:configChanges="screenLayout|screenSize|orientation|keyboardHidden"
    android:screenOrientation="Landscape"
    android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen" />

<activity
    android:name="com.meizu.atlas.server.am.OrientationPProxyActivity"
    android:configChanges="screenLayout|screenSize|orientation|keyboardHidden"
    android:screenOrientation="portrait"
    android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen" />

<activity
    android:name="com.meizu.atlas.server.am.TranslucentProxyActivity"
```

```

        android:configChanges="screenLayout|screenSize|orientation|keyboardHidden"
        android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen" />

<activity
    android:name="com.meizu.atlas.server.am.ConfigChangesProxyActivity"
    android:configChanges="screenLayout|screenSize|orientation|keyboardHidden"
    android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen" />

<activity
    android:name="com.meizu.atlas.server.am.TranslCfgChangesProxyActivity"
    android:configChanges="screenLayout|screenSize|orientation|keyboardHidden"
    android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen" />

<activity
    android:name="com.meizu.gamesdk.update.UpdateActivity"
    android:configChanges="screenLayout|screenSize|orientation|keyboardHidden"
    android:theme="@android:style/Theme.Translucent" />

<service
    android:name="com.meizu.atlas.server.am.stub.ProxyService"
    android:exported="false">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="com.meizu.atlas.server.am.PROXY_STUB" />
    </intent-filter>
</service>

```

注: **`#{applicationId}`** 需要改成游戏包名

## 4.2.2. SDK 初始化

使用 SDK 前，需要对 SDK 进行初始化。初始化需要魅族开发者平台申请的 appid 和 appkey 值作为参数。可以在 Application 的 onCreate()方法中调用。

```
MzGameCenterPlatform.init(context, appid, appkey);
```

注意：

init()方法在游戏应用启动时调用一次即可(推荐在 Application 的 onCreate()中调用)，该方法会记录游戏的 appid 和 appkey，没有其它特殊操作，不会阻塞线程。

### 4.2.3. 补单调用代码

调用登录接口登录成功后，会通过监听器返回登录状态和登录的用户信息

属性	类型	长度	名称	备注
activity	Activity			
mzPayListener	MzPayListener			

```
// TODO 调用登录接口。注意，该方法必须在应用的主线程中调用。
```

```
MzGameCenterPlatform.orderQueryConfirm(activity,mzPayListener)
```

### 4.2.5. 支付调用代码

签名说明：

拼接签名串：key=value&key=value，key 按照字母升序

除了 null 不参与签名，空值""是要参与签名的

sign=MD5(拼接签名串:appsecret)，签名后 sign 值全部小写

```
// TODO 创建订单
```

```
String cpUserInfo = "test";
String totalPrice = String.valueOf(amount);
String orderId = "" + System.currentTimeMillis();
String productId = "001";
String productSubject = orderName;
String productBody = "";
int payType = 0;
long createTime = System.currentTimeMillis();

StringBuilder builder = new StringBuilder();
final String equalStr = "=";
final String andStr = "&";
builder.append("app_id" + equalStr + appid + andStr);
builder.append("cp_order_id" + equalStr + orderId + andStr);
builder.append("create_time" + equalStr + createTime + andStr);
builder.append("pay_type" + equalStr + payType + andStr);
builder.append("product_body" + equalStr + productBody + andStr);
builder.append("product_id" + equalStr + productId + andStr);
builder.append("product_subject" + equalStr + productSubject + andStr);
builder.append("total_price" + equalStr + totalPrice + andStr);
builder.append("user_info" + equalStr + cpUserInfo);
builder.append(": " + GameConstants.GAME_SECRET);
String sign = MD5Utils.sign(builder.toString());
```

```

String signType = "md5";

Bundle payInfo = new Bundle();
// appid
payInfo.putString(MzPayParams.ORDER_KEY_ORDER_APPID, appId);
// cp自定义信息
payInfo.putString(MzPayParams.ORDER_KEY_CP_INFO, cpUserInfo);
// 金额
payInfo.putString(MzPayParams.ORDER_KEY_AMOUNT, totalPrice);
// 订单id
payInfo.putString(MzPayParams.ORDER_KEY_ORDER_ID, orderId);
// 产品id
payInfo.putString(MzPayParams.ORDER_KEY_PRODUCT_BODY, "");
// 产品id
payInfo.putString(MzPayParams.ORDER_KEY_PRODUCT_ID, productId);
// 产品描述
payInfo.putString(MzPayParams.ORDER_KEY_PRODUCT_SUBJECT, productSubject);

//支付方式, 默认值: "0" (即定额支付)
payInfo.putInt(MzPayParams.ORDER_KEY_PAY_TYPE, payType);
// 签名
payInfo.putString(MzPayParams.ORDER_KEY_SIGN, sign);
// 签名类型
payInfo.putString(MzPayParams.ORDER_KEY_SIGN_TYPE, signType);
// 是否关闭短信支付,默认是打开状态
payInfo.putBoolean(MzPayParams.ORDER_KEY_DISABLE_PAY_TYPE_SMS,
mSwCloseSms.isChecked());
payInfo.putLong(MzPayParams.ORDER_KEY_CREATE_TIME,createTiem);
// 优先支付渠道 0:支付宝 1:微信 2:银联 3:短信
payInfo.putInt(MzPayParams.ORDER_KEY_PAY_CHANNEL, 0);
// TODO 调用支付接口。注意,该方法必须在应用的主线程中调用。
MzGameCenterPlatform.singlePay(this, payInfo, new MzPayListener() {
    // TODO 支付结果回调,该回调跑在应用主线程。注意,该回调跑在应用主线程,不能在这里做耗时操作
    @Override
    public void onPayResult(int code, Bundle info, String errorMsg) {
        String orderid = "";
        if (info != null) {
            orderid = info.getString(MzPayParams.ORDER_KEY_ORDER_ID);
        }
        switch (code) {
            case PayResultCode.PAY_SUCCESS:
                displayMsg("支付成功:" + orderid);
                break;
            case PayResultCode.PAY_USE_SMS:

```

```

        // TODO 短息支付,CP 需要调起自己的短代支付
        displayMsg("短信支付:" + orderid);
        break;
    case PayResultCode.PAY_ERROR_CANCEL:
        // TODO 用户主动取消支付操作,不需要提示用户失败
        displayMsg("用户取消:" + orderid);
        break;
    case PayResultCode.PAY_ERROR_CODE_DUPLICATE_PAY:
        displayMsg("重复支付:" + orderid);
        break;
    case PayResultCode.PAY_ERROR_GAME_VERIFY_ERROR:
        displayMsg(errorMsg);
        break;
    default:
        // TODO 支付失败,包含错误码和错误消息。
        // TODO 注意,错误消息(errorMsg)需要由游戏展示给用户,提示失败原因
        displayMsg("支付失败:" + code + "," + errorMsg);
        break;
    }
    Log.i("MzGameSDK", errorMsg + code);
}
});

```