

小米游戏中心游戏接入手册

(网游技术分册)

V2.3.0

2013-08-22

修订记录：

日期	版本	修订记录
2013/02/25	2.0.0	Service 版本文档初稿，支持小号功能
2013/03/05	2.1.0	充值方式增加财付通安全支付和网页支付
2013/04/25	2.1.1	支付页面支持横屏，程序优化
2013/05/08	2.1.2	增加 Service 版自升级 增加 CP 服务端同步用户游戏角色信息，修改见 2.1.5.3
2013/06/07	2.2.0	1.大众支付方式增加联通短信充值（仅联通用户可见） 2.修改用户登陆时选择小号的逻辑，删除原有的小号列表页面，增加用户切换小号页面，和小号管理页面，如用户不切换默认使用上次登陆帐号登陆 3.增加查询用户信息接口，修改见 2.1.5.4 4.增加创建充值订单接口，修改见 2.1.5.5 3，4 两个接口是在网页端为游戏充值功能而设计，鉴于小米论坛的流量巨大，估计会有很多用户使用网页充值功能，建议游戏开发商务必实现这两个接口。
2013/06/19	2.2.1	修改 bug
2013/08/22	2.3.0	1.修改登陆成功后欢迎框样式 2.优化登陆逻辑，修复因服务器删除某些数据时候客户端无法登陆

术语定义：

术语名称	中文含义	定义
mibi	米币	小米公司发行的虚拟货币，用于移动支付
MIUI	米柚操作系统	
SDK	开发包	
cpOrderId	订单号	由开发者产生
productCode	商品代码	由小米游戏中心后台产生，仅单机业务需要申请商品代码
appId	App 的唯一编号	appId 由小米游戏中心后台生成，每个游戏业务有一个唯一编号，数字类型。
appKey	App 的密钥	调用 SDK 时需要
OfflineGame	单机游戏	包含弱联网类游戏
OnlineGame	网络游戏	

一、小米游戏平台

1.1.小米游戏平台介绍

小米游戏平台是小米公司推出的面向所有小米手机、MIUI 用户的游戏平台，整个游戏平台整合了 MIUI 系统、开发 SDK、互联网站、小米论坛等各方面优质资源，力求打造一个良好的安卓游戏生态系统。

小米游戏平台提供游戏下载、游戏搜索、游戏计费（支持单机游戏和网络游戏）等多种平台支撑能力，接入简单快捷，是众多游戏开发商最优的选择。

1.2.小米账户

小米账户是小米公司为所有小米网、MIUI、米聊而开发的小米通行证账户系统，目前注册用户已经超过 8000 万，小米账户主要采取绑定手机号码或邮箱的方式，用户注册了小米账户同时会享受网盘、小米云存储、语音助理、米聊、游戏中心等多项优质免费服务。

小米账户即可使用小米 ID 登录，也可使用手机号码、邮箱作为用户名登录，十分方便快捷。



1.3.米币系统

米币是小米公司针对虚拟商品支付而发行的一种通用代币，类似于 Q 币。具有可在小米公司所有虚拟物品平台流通的特性。用户只要通过小米账号进行充值（用人民币购买米币），就可以在小米公司旗下各移动产品或合作产品平台中支付使用。

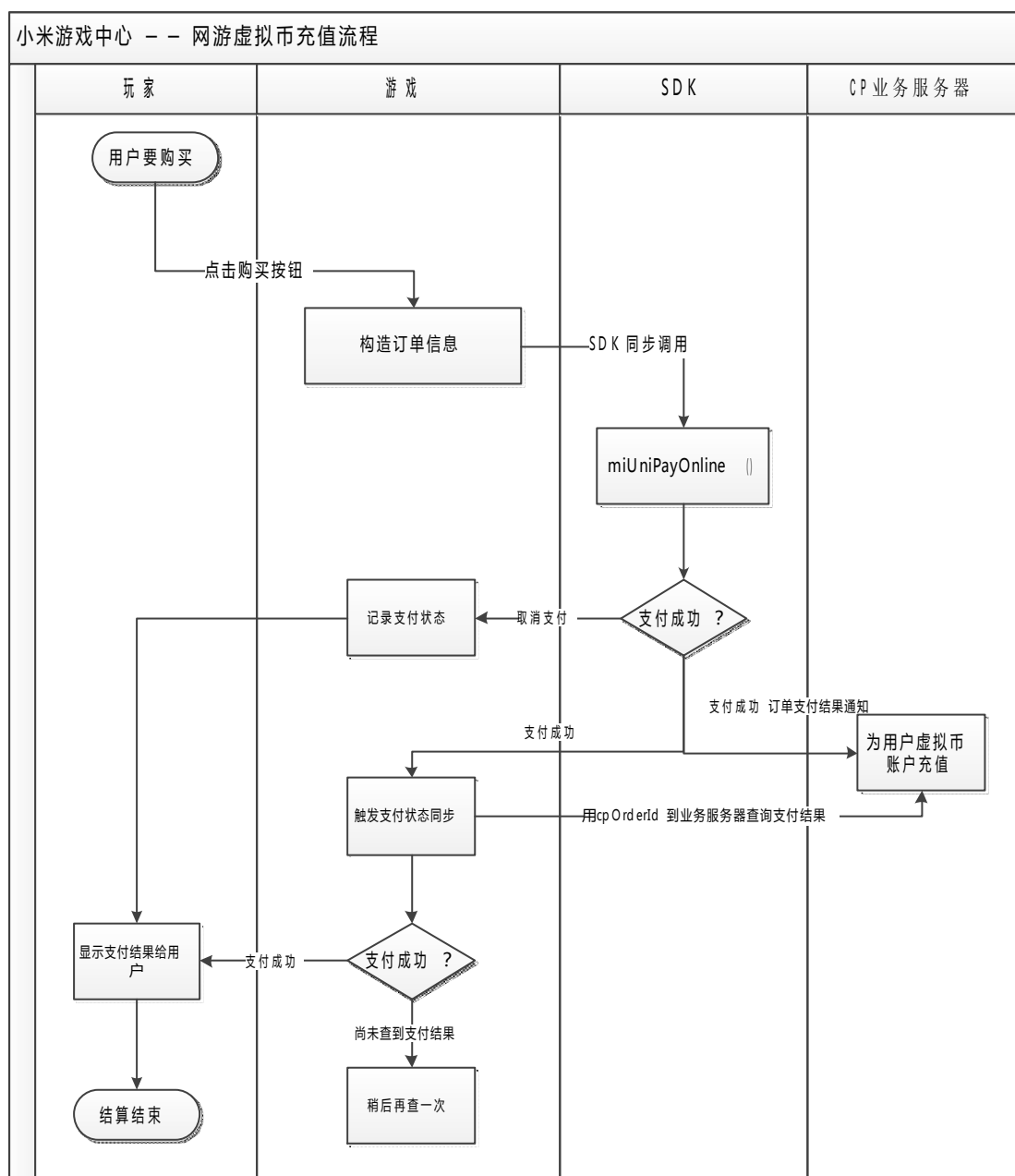
1 元=1 米币，米币最小单位是 1 分，即可定价为 0.01 米币。

开发者将所在应用中用户消费的米币数额作为应用收入的结算依据。

注意：

- 1、UID 不是小米 ID，但与小米 ID 有对照关系；
- 2、请开发者一定要使用这个 UID 作为用户标识，不要使用本机 IMEI 或 IMSI
- 3、未来游戏平台支持一个小米 ID 对应多个 UID，从而实现用户在单款游戏中创建多个角色的功能。

2.1.3 网游用户使用米币充值流程



2.1.4 SDK 调用方法

对于开发者来说，只需要引入以下代码即可完成网络游戏支付流程。

2.1.4.1.支付方式说明

	快捷支付	大众支付
参数定义	PayMode.simple	PayMode.custom
是否提示用户余额不足？	否	是
是否需要 CP 做充值卡充值页面？	需要 CP 做充值卡录入的 UI，然后把信息通过接口传给我们 SDK	不需要
优势分析	用户余额不足时，直接跳到支付宝充值页面，流程相对较短，但接入方工作会多一些	用户余额不足时会提示先充值，充值之后需要用户确认才能扣款，流程稍长，接入方工作量最少
案例	小米神仙道	大掌门
是否默认	否	是默认的支付方式

如果您不知道该选哪一种，就用‘大众支付’吧。

2.1.4.2.初始化

请先获取 AppId 和 AppKey，然后调用下面的 SDK 初始化代码进行初始化操作。

将 SDK 包中的 MiGameCenterSDKService.apk 放到应用工程的 assets 目录下，SDK 的 jar 包放到工程的 libs 下，在 buildpath 中引用，然后对 SDK 进行初始化,(注：检查游戏的包名是否与提供给小米后台数据配置的包名一致，appid 与 appkey 是否与申请的一致，如果不一致会导致成功调用登录和其它 SDK 接口失败)

Init 函数在 Application.onCreate 中调用！

```
MiAppInfo appInfo = new MiAppInfo();
appInfo.setAppId("请申请获得");
appInfo.setAppKey("请申请获得");
appInfo.setAppType(MiGameType.online); // 网游
//设置为大众支付方式（即米币余额不足时，先提示用户充值米币）
appInfo.setPayMode(PayMode.common);
//设置为快捷支付方式（即米币余额不足时，直接提示用户使用支付宝充值）
//appInfo.setPayMode(PayMode.simple); //若不设置，默认为大众支付
MiCommplatform.Init( this, appInfo );
```

2.1.4.3. 小米账户登录调用代码

```
MiCommplatform.getInstance().miLogin( context,
    new OnLoginProcessListener()
{
    @Override
    public void finishLoginProcess( int code, MiAccountInfo arg1)
    {
        switch( code )
        {
            case MiErrorCode.MI_XIAOMI_GAMECENTER_SUCCESS:
                // 登陆成功
                //获取用户的登陆后的 UID（即用户唯一标识）

                String uid = arg1.getUid();
                //获取用户的登陆的 Session（请参考 4.2.3.3 流程校验 Session 有效性）
                String session = arg1.getSessionId(); //若没有登录返回 null
                //请开发者完成将 uid 和 session 提交给开发者自己服务器进行 session 验证
                break;
            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_LOGIN_FAIL:
                // 登陆失败
                break;
            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_CANCEL:
                // 取消登录
                break;
            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_ACTION_EXECUTED:
                //登录操作正在进行中
                break;
            default:
                // 登录失败
                break;
        }
    }
});
```



```
    }  
    }  
});
```

可以通过实现 OnLoginProcessListener 接口来捕获登录结果。

2.1.4.4.虚拟币充值调用

```
MiBuyInfoOnline online = new MiBuyInfoOnline();  
online.setCpOrderId(UUID.randomUUID().toString()); //订单号唯一（不为空）  
online.setCpUserInfo(“cpUserInfo”); //此参数在用户支付成功后会透传给 CP 的服务器  
online.setMiBi( 10 ); //必须是大于 1 的整数，10 代表 10 米币，即 10 元人民币（不为空）  
  
MiCommplatform.getInstance().miUniPayOnline(activity, online,  
                                             new OnPayProcessListener()  
{  
    @Override  
    public void finishPayProcess( int code ) {  
        switch( code ) {  
            case MiErrorCode.MI_XIAOMI_GAMECENTER_SUCCESS:  
                //购买成功  
                break;  
            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_PAY_CANCEL:  
                //取消购买  
                break;  
            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_PAY_FAILURE:  
                //购买失败  
                break;  
            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_ACTION_EXECUTED:  
                //操作正在进行  
                break;  
            default:  
                //购买失败  
                break;  
        }  
    }  
});
```

参数说明：

参数名	用途	备注
cpOrderId	开发方订单号	必填参数。 20~100 字符以内，要求必须由开发者的业务服务器

		生成，因订单支付成功后游戏平台服务器会直接将支付结果通知给开发者的业务服务器，通知参数的cpOrderId 是重要信息。
cpUserInfo	给网游透传参数	必填参数。 用于透传用户信息，当用户支付成功后我们会将此参数透传给开发者业务服务器。（不能为 null 或 ""）
mibi	购买米币的数量	必填参数。 且数量是 int 型，即最少只能购买 1 米币对应的虚拟货币。

2.1.4.5 充值卡充值(需要调用方自备 UI 界面，且不支持单机游戏)

注意：若您选择大众支付模式，无需调用此接口。

充值卡类型：

- | CardType.CMCC 中国移动充值卡（含移动地方充值卡）
- | CardType.UNICOM 中国联通充值卡
- | CardType.TELECOM 中国电信充值卡

充值卡面额

通道名称	卡号位数	密码位数	支付面额
全国移动神州行充值卡	17	18	10、20、30、50、100、300、500
全国联通一卡充	15	19	20、30、50、100、200、300、500
全国电信卡	19	18	20、30、50、100
福建移动呱呱通充值卡	16	17	10、20、30、50、100
江苏移动充值卡	16	17	30、50、100
辽宁移动电话交费卡	16	21	10、20、30、50、100
浙江移动缴费券	10	8	10、20、30、50、100、200

代码示例：

```
CardBuyInfo info = new CardBuyInfo( "卡号", "密码",
    UUID.randomUUID().toString(), //cporderid
    "cpUserInfo", //透传的参数
    CardType.CMCC, //卡类型
    50 ); //充值卡面额（50 代表充值卡面额，单位是元）

MiCommplatform.getInstance().miCardPay( OnlineThrActivity.this, info, new OnCardPayProcessListener()
```

```

{

    @Override
    public void finishCardPayProcess( int arg0 )
    {
        switch( arg0 )
        {
            case MiErrorCode.MI_XIAOMI_GAMECENTER_SUCCESS:
                //成功
                handler.sendEmptyMessage( 10000 );
                break;

            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_PAY_FAILURE: //失败
            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_PAYFORCARD_FAILE: //充
值卡充值失败

            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_CARDNUMERROR: //卡号空
            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_CARDPASSERROR: //密码空
            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_CARDCARRISNULL: //运营商
空

            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_CARDNUMORPASSERROR: //
卡号或密码错误

            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_CARDMONEYERROR: //金
额错误

            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_CREATEORDERFAIL: //创
建充值订单错误

            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_CARDNOENOUGHMONY: //充值卡实
际金额小于所填面额

                //失败 以上错误都可以认为是充值失败

                handler.sendMessage( handler.obtainMessage( 20000, arg0 ) );
                break;

            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_ACTION_EXECUTED:
//操作正在进行中
                break;

            default:
                break;

        }
    }
}
};

```

2.1.4.5 帐号注销

注意：帐号注销，实际注销的是小号，不是登陆时使用的小米帐户

```
MiCommplatform.getInstance().miLogout( context,
    new OnLoginProcessListener()
{
    @Override
    public void finishLoginProcess( int code )
    {
        switch( code )
        {
            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_LOGINOUT_FAIL:
                // 注销失败
                break;
            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_LOGINOUT_SUCCESS:
                // 注销成功
                break;

            case MiErrorCode.MI_XIAOMI_GAMECENTER_ERROR_ACTION_EXECUTED:
                // 操作正在进行中
                break;
            default:
                break;
        }
    }
});
```

2.1.5 服务器接口

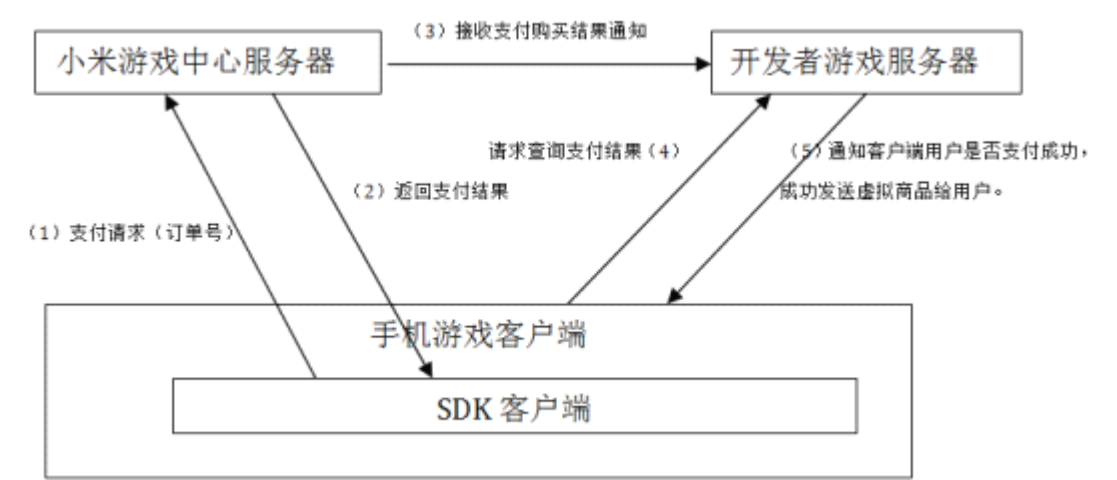
请开发者以小米游戏中心服务器通知的结果为准为用户进行虚拟币充值。

2.1.5.1.订单支付结果通知接口

2.1.5.1.1 流程说明：

在订单支付成功后，小米游戏平台服务器会将支付结果通知给开发者预先提供的服务器上。若开发者所提供的服务器地址不可用，在一定时间段内游戏平台服务器会按照周期进行轮训（前 10 次，每分钟通知 1 次；10 次后每小时通知 1 次）。

具体流程如下：



注：由于是异步通知模型，（3）和（4）不一定是按序号产生。因此（4）和（5）需要进行轮询处理或者使用接口进行支付结果查询。

相比后面提到的开发者主动查询订单的模式，我们推荐使用此模式。

2.1.5.1.2 接口及参数说明：

接口地址：请随《游戏 SDK 申请表》一起提交

请求方法：GET

请求参数说明：

参数名称	重要性	说明
appId	必须	appId

cpOrderId	必须	开发商订单 ID
cpUserInfo	可选	开发商透传信息
uid	必须	用户 ID
orderId	必须	游戏平台订单 ID
orderStatus	必须	订单状态 TRADE_SUCCESS 代表成功
payFee	必须	支付金额，单位为分，即 0.01 米币。
productCode	必须	商品代码
productName	必须	商品名称
productCount	必须	商品数量
payTime	必须	支付时间，格式 yyyy-MM-dd HH:mm:ss
signature	必须	签名,签名方法见后面说明

返回参数说明：

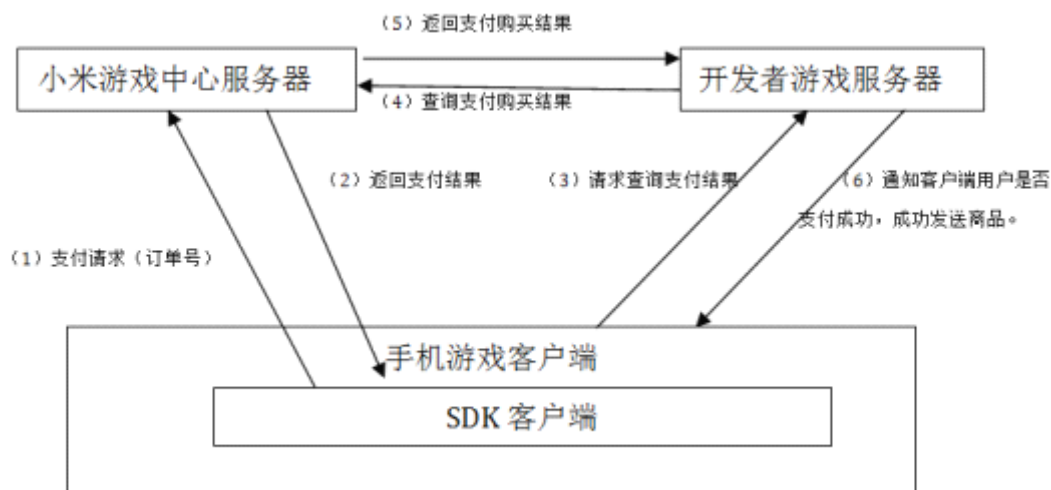
参数名称	重要性	说明
errcode	必须	状态码，200 成功 1506 cpOrderId 错误 1515 appId 错误 1516 uid 错误 1525 signature 错误 3515 订单信息不一致，用于和 CP 的订单校验
errMsg	可选	错误信息

注意：对于同一个订单号的多次通知，开发商要自己保证只处理一次发货。

2.1.5.2.主动查询订单支付状态接口

2.1.5.2.1 流程说明：

此接口由小米游戏中心为开发者提供。



2.1.5.2.2 接口及参数说明：

接口地址：<http://mis.migc.xiaomi.com/api/biz/service/queryOrder.do>

请求方法：GET

请求参数说明：

参数名称	重要性	说明
appId	必须	应用 ID
cpOrderId	必须	开发商订单 ID
uid	必须	用户 ID
signature	必须	签名，签名方法见后面说明

正确返回参数说明：

参数名称	重要性	说明
appId	必须	应用 ID
cpOrderId	必须	开发商订单 ID
cpUserInfo	可选	开发商透传信息
uid	必须	用户 ID

orderId	必须	订单 ID
orderStatus	必须	订单状态 TRADE_SUCCESS 代表成功 WAIT_BUYER_PAY 代表未支付 REPEAT_PURCHASE 订购关系已经存在
payFee	必须	支付金额，单位为分，即 0.01 米币
productCode	必须	商品代码
productName	必须	商品名称
productCount	必须	商品数量
payTime	必须	支付时间，格式 yyyy-MM-dd HH:mm:ss
signature	必须	签名,签名方法见后面说明

错误返回参数说明：

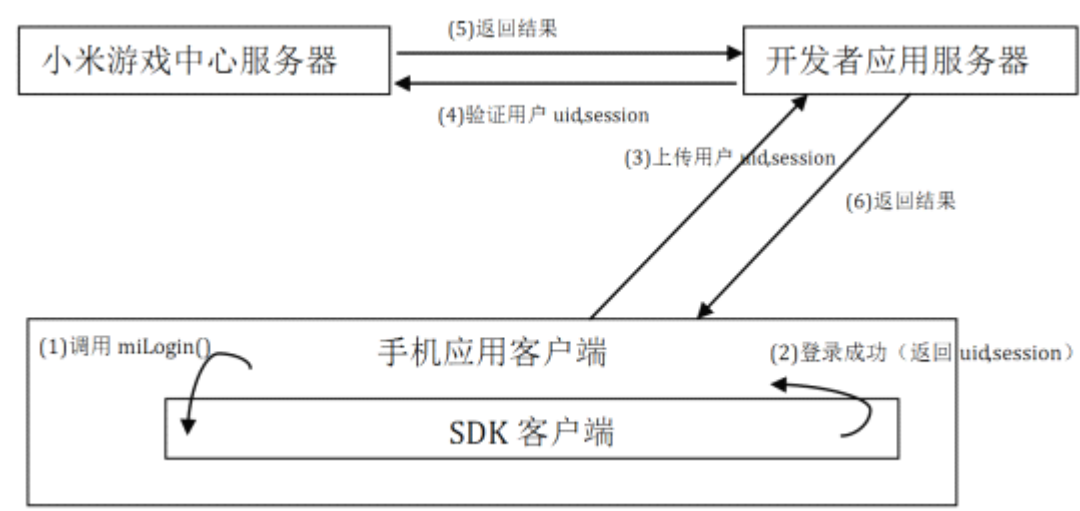
参数名称	重要性	说明
errcode	必须	状态码 1506 cpOrderId 错误 1515 appId 错误 1516 uid 错误 1525 signature 错误
errMsg	可选	错误信息描述

2.1.5.3.用户 session 验证接口

2.1.5.3.1 流程说明：

此接口由小米游戏中心为开发者提供，用于验证登录账户的有效性。

注意：用户的唯一标识是通过 SDK 获得的 uid，而不是 Session，Session 用于校验登录验证的有效性，必须经过 SDK、游戏中心服务器、开发者服务器进行三方验证，如果 Session 失效，需要重新调用 miLogin() 进行登录。



2.1.5.3.2 接口及参数说明：

接口地址：<http://mis.migc.xiaomi.com/api/biz/service/verifySession.do>

请求方法：GET

请求参数说明：

参数名称	重要性	说明
appId	必须	应用 ID
session	必须	用户 session ID
uid	必须	用户 ID
signature	必须	签名，签名方法见后面说明

返回参数说明：

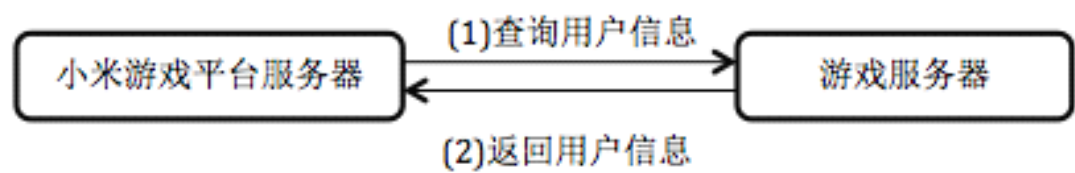
参数名称	重要性	说明
errcode	必须	状态码

		200 验证正确 1515 appId 错误 1516 uid 错误 1520 session 错误 1525 signature 错误
errMsg	可选	错误信息描述

2.1.5.4.小米游戏平台查询用户信息接口

此接口实现了小米游戏平台主动查询游戏开发者的用户信息。

2.1.5.4.1 流程说明：



2.1.5.4.2 接口及参数说明：

接口地址：由游戏开发者提供

请求方式：POST

请求参数说明：

参数名称	重要性	说明
appId	必须	应用 ID
uid	必须	用户 ID
signature	必须	签名，签名方法见后面说明

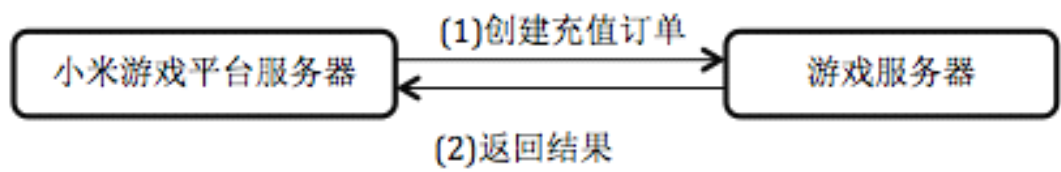
返回参数说明：

参数名称	父节点	重要性	说明
errcode	/	必须	状态码，取值见 3.3
errMsg	/	可选	错误信息描述
appId	/	必须	应用 ID
uid	/	必须	用户 ID
userInfos	/	必须	用户信息列表
userGameRoleId	userInfos	必须	用户在游戏角色 ID, 如果没有填用户 ID
userGameRoleName	userInfos	必须	用户在游戏中角色名称
userGameLv	userInfos	必须	用户在游戏中角色级别，如：1 级 填 1
userGameServerName	userInfos	必须	用户在游戏中角色所在服务器
userGameBalance	userInfos	必须	用户游戏帐户余额
userGameVip	userInfos	必须	游戏内 VIP 级别，VIP1，VIP2,VIP3 等待依次类推，VIP0 代表不是 VIP 用户。
userGamePartyName	userInfos	可选	角色所在帮会名称
signature	/	必须	签名，签名方法见后面说明

2.1.5.5 创建 CP 充值订单接口

此接口实现了小米游戏平台访问游戏开发者服务器创建充值订单接口。

2.1.5.5.1 流程说明：



2.1.5.5.2 接口及参数说明：

接口地址：由游戏开发者提供

请求方式：POST

请求参数说明：

参数名称	重要性	说明
tradeNo	必须	交易流水号，由小米游戏平台生成，20 位定长字符串
appId	必须	应用 ID
uid	必须	用户 ID
userGameRoleId	必须	用户在游戏角色 ID, 如果没有填用户 ID
userGameRoleName	必须	用户在游戏中角色名称
userGameServerName	必须	用户在游戏中角色所在服务器
payFee	必须	支付金额，单位为分，即 0.01 米币。
productCode	必须	商品代码
productName	必须	商品名称
productCount	必须	商品数量
payFee	必须	支付金额，单位为分，即 0.01 米币。
signature	必须	签名，签名方法见后面说明

返回参数说明：

参数名称	重要性	说明
errcode	必须	状态码，取值见 3.3
errMsg	可选	错误信息描述
tradeNo	必须	交易流水号，由小米游戏平台生成，20 位定长字符串
appId	必须	应用 ID
uid	必须	用户 ID
cpOrderId	必须	cp 订单号
signature	必须	签名，签名方法见后面说明

接口格式说明：

输入参数: ?参数 1=值 1&参数 2=值 2&....&参数 n=值 n,如果遇到文本参数值,需要根据情况对参数值做 UrlEncode。

返回参数：采用 json 格式，如:{"返回参数 1":"返回值 1","返回参数 2":"返回值 2",...."返回参数 n":"返回值 n"}

签名方法说明：

1、生成带签名字符串

表中各参数按字母顺序排序（不包含 signature），如果第一个字母相同，按第二个字母排序，依次类推。排序后拼接成 par1=val1&par2=val2&par3=val3 的格式,所生成的字符串即为待签名的字符串。没有值的参数请不要参与签名。由于有些数据根据 HTTP 协议需求,需要进行 URLencoding,这样接收方才可以接收到正确的参数,但如果这个参数参与签名,那么待签名字符串必须是字符串原值而非 URLencoding 的值。

2、签名算法

以 appKey 作为 key,使用 hmac-sha1 带密钥(secret)的哈希算法对代签字符串进行签名计算。签名的结果由 16 进制表示。hmac-sha1 带密钥(secret)哈希算法的实现参考附录。

三、FAQ

3.1.APK 打包及发布

需要注意，SDK 包是以 jar 包提供给开发者，此 jar 包本身已为混淆状态，您在混淆自己游戏的 APK 包时，需要在 proguard.cfg 里加入，以避免二次混淆。

```
public <init>(...);
}
-keepclassmembers enum * {
    public static **[] values();
    public static ** valueOf(java.lang.String);
}
-keep class * implements android.os.Parcelable {
    public static final android.os.Parcelable$Creator *;
}
```

3.2.网游服务器签名函数

Hmac - SHA1 算法 java 实现参考：

```
import javax.crypto.Mac;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;

public class HmacSHA1Encryption {
    private static final String MAC_NAME = "HmacSHA1";
    private static final String ENCODING = "UTF-8";
    /**
     * 使用 HMAC-SHA1 签名方法对 encryptText 进行签名
     * @param encryptText 被签名的字符串
     * @param encryptKey 密钥
     * @return 返回被加密后的字符串
     * @throws Exception
     */
    public static String HmacSHA1Encrypt( String encryptText,
String encryptKey ) throws Exception{
```

```

    byte[] data = encryptKey.getBytes( ENCODING );
    // 根据给定的字节数组构造一个密钥,第二参数指定一个密钥算法的名称
    SecretKey secretKey = new SecretKeySpec( data, MAC_NAME );
    // 生成一个指定 Mac 算法的 Mac 对象
    Mac mac = Mac.getInstance( MAC_NAME );
    // 用给定密钥初始化 Mac 对象
    mac.init( secretKey );
    byte[] text = encryptText.getBytes( ENCODING );
    // 完成 Mac 操作
    byte[] digest = mac.doFinal( text );
    StringBuilder sBuilder = bytesToHexString( digest );
    return sBuilder.toString();
}

```

```

/**
 * 转换成Hex
 *
 * @param byteArray
 */
public static StringBuilder bytesToHexString( byte[] byteArray ){
    if ( byteArray == null ){
        return null;
    }
    StringBuilder sBuilder = new StringBuilder();
    for ( byte b : byteArray ){
        String hv = String.format("%02x", b);
        sBuilder.append( hv );
    }
    return sBuilder;
}

```

```

/**
 * 使用 HMAC-SHA1 签名方法对对encryptText进行签名
 *
 * @param encryptData 被签名的字符串
 * @param encryptKey 密钥
 * @return 返回被加密后的字符串

```

```

* @throws Exception
*/
public static String hmacSHA1Encrypt( byte[] encryptData, String
encryptKey ) throws Exception{
    byte[] data = encryptKey.getBytes( ENCODING );
    // 根据给定的字节数组构造一个密钥,第二参数指定一个密钥算法的名称
    SecretKey secretKey = new SecretKeySpec( data, MAC_NAME );
    // 生成一个指定 Mac 算法的 Mac 对象
    Mac mac = Mac.getInstance( MAC_NAME );
    // 用给定密钥初始化 Mac 对象
    mac.init( secretKey );
    // 完成 Mac 操作
    byte[] digest = mac.doFinal( encryptData );
    StringBuilder sBuilder = bytesToHexString( digest );
    return sBuilder.toString();
}
}

```

3.3 错误代码对照表

代码	代码含义	代码	代码含义	代码	代码含义	代码	代码含义
200	成功	1514	CP 编号错误	1535	充值金额错误	3001	产品不存在或已经下线
1001	系统错误	1515	AppId 错误	1536	充值类型错误	3004	米粒账户金额错误
1002	数据库错误	1516	内部用户编号错误(uid 错误)	1537	充值订单号错误	3005	auth 验证失败
1003	缓存处理错误	1517	渠道号错误	1538	签名为空	3301	获取用户米粒账户错误
1004	http 请求错误	1518	账户状态错误	1539	分页查询最大 id	3302	订单创建错误
1005	重复请求	1519	产品 id 错误	1540	每页记录条数错误	3303	卡支付调用错误
1006	json 解析错误	1520	session 错误	1541	支付时间错误	3501	订单处理中
1007	RSA 签名公钥或私钥错误	1521	手机号码错误	1542	分页时多少页	3502	订单处理超时

1501	未输入请求参数	1522	产品单价错误	1543	开始时间	3510	订单失败，未知原因
1502	IP 鉴权错误	1523	购买数量错误	1544	结束时间	3511	订单不存在
1503	消息号错误	1524	没有 signature 参数	1545	订单请求时间错误	3512	订单已支付
1504	网元错误	1525	验证 signature 错误	1546	已下线的应用	3513	输入参数 miBi 错误
1505	小米 id 错误	1526	apkName 错误	2001	余额不足	3514	订购关系已经存
1506	cp 订单号错误	1527	apkSign 错误	2002	重复交易	4000	authToken 过期
1507	订单金额错误	1528	sdkVersionCode 错误	2003	签名错误	4001	authToken 非法
1508	支付方式错误	1529	卡面值错误	2004	重复充值	4002	serviceToken 过期
1509	订单 id 错误	1530	cpUserInfo 错误，透传参数	2005	充值订单不存在	4003	获取 serviceToken 失败
1510	支付订单 id 错误	1531	对账开始时间错误	2006	充值订单状态不存在	4004	验证 serviceToken 失败
1511	支付结果错误	1532	对账结束时间错误	2007	业务系统公钥不存在	4005	注销失败
1512	支付金额错误	1533	p 参数错误	2008	计费系统私钥不存在	4006	上传用户玩过的游戏失败
1513	单类型错误	1534	订单描述信息错误	2009	网元不存在	4007	修改用户信息错误
3515	订单信息不一致，用于和 CP 的订单校验						