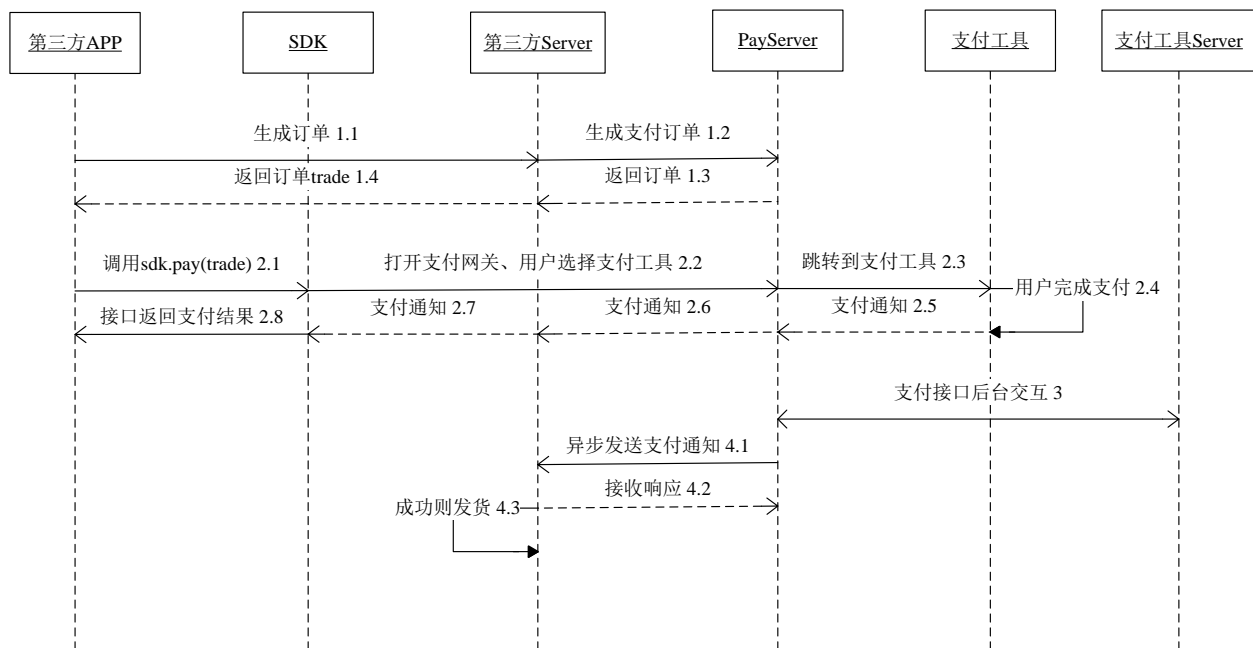


编号	章节名称	修订内容简述	修订日期	修订前版本号	修订后版本号	修订人
1		创建文档	20140130		1.0	yixinopen@188.com
		去掉 goodscount 参数	20140221		1.1	
		重构文档	20140226		1.2	
		增加手机话费支付更新商品信息说明。	20140311		1.3	
		增加 Android 客户端集成支付 SDK	20140312		1.4	
		修改 Android 客户端集成支付 SDK 说明	20150527		1.5	
		增加文字描述	20150909		1.5	
		更新银联 SDK	20150916		1.6	

易信游戏支付 android sdk 应用集成接入

1. 支付流程



1.1. 流程说明

第三方负责开发第三方 APP、第三方 Server 的逻辑。

SDK 是支付 android SDK 包。

PayServer 是支付平台服务器。支付工具及 Server 是诸如支付宝、银联等。

由流程图可知第三方 APP 负责调用第三方 Server “生成订单 1.1”，并处理返回值、封装成 trade 参数调用 sdk.pay 接口。用户支付完成后，回调”接口返回支付结果 2.8”。

第三方 Server 负责生成订单，并将订单信息持久化到数据库。

第三方 Server 负责处理支付通知，对支付成功的订单进行发货、更改数据库状态。

注意点：

- 1、调用支付接口时都必须传递易信用户的 access_token，用于提取用户名、验证。
- 2、任一步都可能中断，要保证最终数据的一致性、正确。比如悬挂订单最终要超时。

- 3、任一步都要签名，防篡改、欺骗、重放。
- 4、任一步都有可能多次重复调用，被调用者负责去重。

2. 服务器端接口

服务器端接口包括第三方 Server、PayServer 提供给对方和 android 端的接口。主要包括订单生成、支付通知两类接口。考虑到安全性，除了用户输入、UI 交互接口需要在 android 端完成，其他跟支付有关的接口尽量在服务器端进行调用。比如签名由服务器端进行，公钥不会暴露在 android 端。

2.1. 接口约定

2.1.1. 版本号

版本号用于接口扩展兼容。

版本号	发布日期	说明
1	20140220	

2.1.2. 商品名称参数 goodsname

第三方在后台系统“设置游戏可购买项”来配置商品的 Goods Name 列值（注意不是名称列）。

序号	名称 (描述)	Goods Name (当前游戏唯一，创建成功之后不可修改)	购买费用 (人民币, 元)	操作
1	钻石	coin	0.01	编辑

2.1.3. 支付工具类型参数 paytooltype

手机话费支付比较特殊，具体见“[手机话费支付更新商品金额信息](#)”。

paytooltype 值	支付工具	是否有扣点	第三方 Server 处理商品信息	是否支持
1	支付宝	无	检查 goodsname、goodsprice、goodsamount	支持

2	银联	无	检查 goodsname、goodsprice、goodsamount	支持
3	手机话费支付（移动）	有，扣 50%	更新 goodsname、goodsprice、goodsamount	暂不支持
4	手机话费支付（电信）	有，扣 50%	更新 goodsname、goodsprice、goodsamount	暂不支持
5	手机话费支付（联通）	有，扣 50%	更新 goodsname、goodsprice、goodsamount	暂不支持

2.1.4. 订单信息持久化和检查

第三方 Server 需要对订单信息包括订单 id、支付状态、各类时间戳等信息、接口参数进行持久化，建议使用数据库。这些信息用来对账。

非手机话费支付时，第三方 Server 需要对支付[页面通知接口](#)、[异步通知接口](#)接口返回的 goodsname、goodsprice、goodsamount 这几个字段跟自身持久化到数据库的值一一对比检查，具体见 [paytooltype 参数说明](#)。

2.1.5. 手机话费支付更新商品金额信息

用户购买道具时，第三方调用“1.2 生成支付订单”接口传递的 goodsname 是道具名称，返回的是该道具 goodsprice、goodsamount。但如果用户在支付网关页面选择“手机话费支付”，则实际购买的是金币，金额也是由用户在手机话费支付页面自由选择，此时与道具商品已不相关。用户可以在第三方 app 中使用金币购买此道具（需要第三方 app 支持）。

第三方 Server 需要判断支付[页面通知接口](#)、[异步通知接口](#)接口返回的 [paytooltype 参数](#)。如果支持手机话费支付，则根据返回参数 goodsname、goodsprice、goodsamount 更新自身订单表对应字段，具体见 [paytooltype 参数说明](#)。如果不支持话费支付，则抛出异常，避免因运营同事误操作打开话费支付，导致小金额话费购买任意大金额道具。

扣点说明：第三方在后台系统配置 1 个金币的价格，比如 0.01 元，目前支付平台扣点为 50%，则用户付 1 元钱只能买到 50 个金币。支付通知接口的 goodsamount 参数为已扣点金额，第三方依此来换算金币数量。

2.1.6. 支付通知并发问题

用户支付完成后，android 页面或 Payserver 会调用支付通知接口通知第三方 Server 支付成功。第三方 Server 响应时需要逻辑去重，特别是并发通知时，要避免重复发货、数据不一致等问题。只有当支付状态从未支付变为已支付时才发货，类似如下 sql 语句返回 true 时才发货，在并发情况下由数据库保证只发货一次：

```
update Trade set PayState=Payed where PayState=UnPay and Id=?;
```

2.1.7. 参数值编码

为方便处理，对所有参数值使用 UTF-8 进行 URLEncode 编码来处理特殊字符（如：空格、=、&、@等）。例如：ordertime=2014-01-01 12:12:12 编码后为 ordertime =2014-01-01+12%3A12%3A12。

由 RSA.sign()接口生成的签名串为 16 进制编码，无需再 URLEncode。

注意：接口响应的返回值内容不进行编码。

2.1.8. 公私钥和签名

公私钥生成使用 Demo 中的 RSA.genRSAKeyPair()接口，运行此类 main 函数即可生成一对。第三方在后台修改基本信息中设置公钥，私钥用来对参数进行签名。

待签名数据包含的参数值为使用 UTF-8 进行 URLEncode 编码后的值。例：

```
String signSrc = URLEncoder.encode("待签名参数", "UTF-8");
```

签名接口使用 android demo 工程的类 im.yixin.game.demo.rsa.RSA。RSA.sign()接口生成的签名串为 16 进制编码，无需再 URLEncode。

PayServer 返回数据中的 sign 需要使用支付平台公钥进行签名验证，Demo 中 RSA.verify 方法即可验证，支付平台公钥为：

```
30819f300d06092a864886f70d010101050003818d0030818902818100ac1b8d63bcaf49cdd0d1e79
c916aba0250421b3ee8eaf134f80843c5033e30a150b9e26e78025fde8e52538d4beb572940966b0c
80460d90a26c9119a0d28c4277024dbeb20e31403360aeca70da506a19d89e95512e5347be0eae9
b2c49da3150a93e3bc80817fa9a1d8170555e6117c86f84f13afc39944fb6bdfc85e3723b0203010001
```

2.1.9. 异常号

异常号	说明	用户页面提示文案
-1	服务器内部异常。未知异常，或无需详细说明的一般性异常。	支付请求失败，请重试
-2	登陆账号异常	账号登录失败，请重试
-3	生成签名失败	生成签名失败，请重试
-4	验证签名失败	验证签名失败，请重试
-5	商品不存在	商品不存在，请重试
-6	订单不存在	订单不存在，请重试
-7	支付方式错误	支付方式错误，请重试
-8	参数非法	参数非法
-9	超时	支付超时，请重新支付
-31	第三方订单号重复	订单重复
-51	支付工具通知的支付状态非法	无效订单
-101	数据库异常	支付数据请求失败，请重试

2.2. 生成订单接口

本接口为流程图中的“生成支付订单 1.2”接口，第三方 Server 在生成订单时调用 PayServer 的此接口。第三方处理时需要逻辑去重，避免重复支付、数据不一致等问题。

2.2.1. 接口定义

URL: <https://game.yixin.im/pay/games/{gameid}/trades>

提供方: PayServer

调用者: 第三方 Server

请求方式: post

路径参数: gameid 第三方 id, 由游戏中心分配。

订单唯一性: 相同的 thirdpart_orderid、thirdpart_ordertime 为第三方唯一订单号, 不能重复调用此接口生成订单。

参数	参数名称	参数说明	类型 (字符长度)	必传	传递者
v	接口版本号	接口版本号。将来版本兼容使用。 该参数请使用 SDK 中 YXPayConstants.YX_PAY_SDK_VERSION	String	是	sdk
thirdpart_orderid	第三方唯一订单 id	同一订单	String(64)	是	game
thirdpart_ordertime	第三方订单下单时间	格式(yyyy-MM-dd HH:mm:ss)	String(64)	是	game
goodsname	商品名称	小写英文字母、数字、下划线的组合	String(24)	是	game
access_token	易信 oauth token	易信登陆时返回, 用于账号验证。	String	是	game
goodscount	购买数量	为兼容老版支付, 不发默认当 1	Int	否	game
sign	签名	待签名数据包含的参数值顺序, goodscount 为 1 时, 签名允许不签进去: v+thirdpart_orderid+thirdpart_ordertime+goodsname +access_token+[goodscount]	String	是	game

2.2.2. 返回值字段

参数	参数名称	参数说明	类型
v	接口版本号		String
thirdpart_orderid	第三方唯一订单 id		String
thirdpart_ordertime	第三方订单下单时间		String
goodsname	商品名称	原值返回。	String
result	返回结果	正常 0、异常时返回异常号	int
trade_serialid	支付平台订单序列号		String
goodsprice	商品单价	两位小数点	String
goodsamount	支付总金额	两位小数点	String
pay_url	支付工具页面跳转 url	包含支付页面需要的参数。 第三方 Server 需要原样返回给客户端。	String
sign	签名	待签名数据包含的参数值顺序，goodscount 为 1 时，该签名没有签进去： v+thirdpart_orderid+thirdpart_ordertime+goodsname +result+trade_serialid+goodsprice+goodsamount+pay_url+[goodscount]	String

2.3. Android 页面同步通知

本接口为流程图中的“支付通知 2.6”接口。用户在 android 端完成支付时，sdk 打开的页面调用此接口来通知第三方 Server 支付状态。第三方 Server 响应此请求，如果支付成功则发货、修改支付状态。

需要第三方在后台系统中配置“设置支付成功的回调”链接 URL 地址。

第三方处理时需要逻辑去重，避免重复支付、数据不一致等问题，见“[支付通知并发问题](#)”。

提供方（处理此请求）：第三方 Server

调用者：android 端 sdk 打开的页面

请求方式：get

返回值：成功处理则 response.body 返回 success 这 7 个字符，失败返回 fail。

2.3.1. 请求参数

参数	参数名称	参数说明	类型(字符长度)
v	接口版本号		String
thirdpart_orderid	第三方唯一订单 id		String
thirdpart_ordertime	第三方订单下单时间		String
goodsname	商品名称	如果 paytooltype 是手机话费支付则需要 更新商品和金额信息	String
result	返回结果	正常 0、异常时返回异常号。返回异常时其他字段可能不返回。	int
trade_serialid	支付平台订单序列号		String
goodsprice	商品单价	两位小数点。如果 paytooltype 是手机话费支付则需要 更新商品和金额信息	String
goodsamount	支付金额	两位小数点。如果 paytooltype 是手机话费支付则需要 更新商品和金额信息	String
paystatus	支付结果	支付状态：0 未支付 1 已支付成功 2 关闭。result 返回 0 时才有此字段。	int
paytime	付款时间		long

paytooltype	支付工具类型		int
from	来源参数	值固定为 frontend	String
sign	签名	待签名数据包含的参数值顺序： v+thirdpart_orderid+thirdpart_ordertime+goodsname +result+trade_serialid+goodsprice +goodsamount +paystatus+paytime+paytooltype+from	String

2.4. PayServer 异步通知

本接口为流程图中的“异步发送支付通知 4.1”接口。**android** 页面通知可能因为网络故障导致通知失败，需要 **PayServer** 多次异步通知来保证通知到第三方 **Server**。

第三方可以将此接口和 **android** 页面通知接口在同一个接口中合并处理，此接口增加了 **notifyid**、**notifytime** 两个参数，并且 **from** 值不一样。

第三方处理时需要逻辑去重，避免重复支付、数据不一致等问题，见“[支付通知并发问题](#)”。

当第三方响应 **success** 时，**PayServer** 停止异步通知，否则按如下时间间隔不断通知直到收到 **success** 响应：40 秒、2 分钟、5 分钟、10 分钟、30 分钟、1 小时、2 小时、6 小时、15 小时。

提供方（处理此请求）：第三方 **Server**

调用者：**PayServer**

请求方式：**post**

返回值：成功处理则 **response.body** 返回 **success** 这 7 个字符，失败返回 **fail**。

2.4.1. 请求参数

参数	参数名称	参数说明	类型(字符长度)
v	接口版本号		String

thirdpart_orderid	第三方唯一订单 id		String
thirdpart_ordertime	第三方订单下单时间		String
goodsname	商品名称	如果 paytooltype 是手机话费支付则需要 更新商品和金额信息	String
result	返回结果	正常 0、异常时返回异常号。返回异常时其他字段可能不返回。	int
trade_serialid	支付平台订单序号		String
goodsprice	商品单价	两位小数点。如果 paytooltype 是手机话费支付则需要 更新商品和金额信息	String
goodsamount	支付金额	两位小数点。如果 paytooltype 是手机话费支付则需要 更新商品和金额信息	String
paystatus	支付结果	支付状态：0 未支付 1 已支付成功 2 关闭。result 返回 0 时才有此字段。	int
paytime	付款时间		long
paytooltype	支付工具类型		int
notifyid	通知内部 id	用于校验	long
notifytime	通知时间戳		long
from	来源参数	值固定为 backend	String
sign	签名	待签名数据包含的参数值顺序： v+thirdpart_orderid+thirdpart_ordertime+goodsname +result+tradeid+goodsprice	String

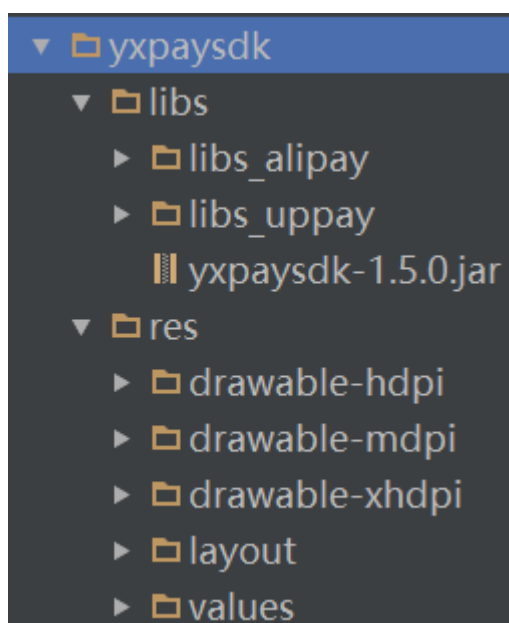
		+goodsamount+paystatus+paytime+paytooltype+notifyid+notifytime+from	
--	--	---	--

3. Android 客户端集成支付 SDK

3.1. 前提

集成支付 sdk 之前需要已集成《易信 Android 平台 SDK 使用新手指南》中的易信社交 sdk。服务器端支付接口需要从社交 sdk 中的用户 oauth 认证得到 access_token。

3.2. 将支付 SDK 导入工程



libs/yxpaysdk.jar 是支付 sdk 的 java 代码。

libs/libs_alipay 目录是接入支付宝需要集成的文件，如果不支持支付宝则忽略。。

libs/libs_uppay 目录是接入银联需要集成的文件，如果不支持银联则忽略。

res 为所需用到的相关资源，请复制到 android 工程的对应目录

3.3. 集成支付流程

参考“社交 SDK 范例代码”demo 工程中的 PayActivity。

3.3.1. 集成 paysdk

1. 将 yxpaysdk.jar 放入 android 工程的 libs 下
2. 如下部分复制到工程的 AndroidManifest.xml:

其中 android:screenOrientation 根据第三方 APP 横竖屏来配置。

注意必须配置 `android:configChanges="orientation|keyboardHidden|screenSize"`。

```
<!--yxpay sdk begin -->
<activity
    android:name="im.yixin.paysdk.paygate.PayGateActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:theme="@style/Android.Theme.NoTitleBar">
</activity>

<activity
    android:name="im.yixin.paysdk.paygate.WebViewPayActivity"
    android:configChanges="keyboardHidden|orientation|screenSize">
</activity>

<activity
    android:name="im.yixin.paysdk.WebviewActivity"
    android:configChanges="keyboardHidden|orientation|screenSize">
</activity>

<activity
    android:name="im.yixin.paysdk.paygate.PayFailedActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"/>
<!--yxpay sdk begin -->

添加权限
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
```

生成订单

第三方 APP 调用[支付流程图](#)中“生成订单 1.1”接口从第三方 Server 生成订单，并在“返回订单 1.4”得到订单信息。订单信息包括[生成订单接口返回值](#)中的支付平台订单序列号 `trade_serialid`、支付跳转 `pay_url` 等。

3.3.2. 调用支付接口

包括如下步骤:

- 1、生成 YXTrade 对象，其字段 `id`、`gateUrl` 分别对应订单信息中的 `trade_serialid`、`pay_url`。
- 2、生成支付类 YXPayApi，编写支付返回结果的回调接口 `YXPayDelegate.onTradeComplete`。

3、调用 `YXPayApi.pay(...)`，该接口负责打开支付工具。

参考代码如下，更完整的代码示例，请参考我们提供的 [demo](#)。

```
protected void onPostExecute(YXTrade trade) {
    if (trade == null) {
        Toast.makeText(PayActivity.this, "生成订单失败", Toast.LENGTH_SHORT).show();
        return;
    }

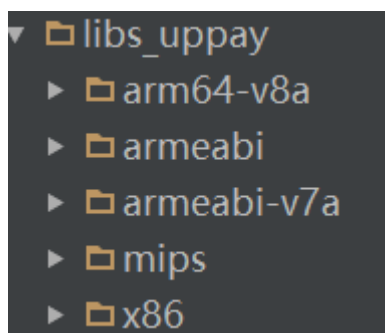
    if (trade.getResult() == YXTrade.RESULT_ERROR_INVALID_TOKEN) {
        Toast.makeText(PayActivity.this, "token验证失败，请先绑定OAUTH", Toast.LENGTH_SHORT).show();
        startActivity(new Intent(PayActivity.this, OauthActivity.class));
        finish();
        return;
    }

    // 调用SDK支付接口
    new YXPayApi(PayActivity.this, new YXPayDelegate() {
        @Override
        public void onTradeComplete(int resultCode) {
            // 注意回调在UI主线程中，对http请求需要异步调用
            if (resultCode == YXPayResultCode.SUCCESS) {
                Toast.makeText(PayActivity.this, "支付成功", Toast.LENGTH_SHORT).show();
            } else if (resultCode == YXPayResultCode.USER_CANCEL) {
                Toast.makeText(PayActivity.this, "用户取消", Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(PayActivity.this, "支付异常: " + resultCode, Toast.LENGTH_SHORT).show();
            }
        }
    }).pay(trade, DemoConstants.access_token);
};
```

3.4. 集成银联支付 SDK

UPPay 目录是接入银联需要集成的文件：

- 1、将目录 `libs_uipay` 下的 `jar` 放入 `android` 工程的 `libs` 下
- 2、将目录 `libs_uipay` 下的 `armeabi`，`armeabi-v7a` 等放入 `android` 工程的 `libs` 下，打包 `apk` 时，请将 `so` 一并打包入 `apk`



- 3、将如下部分复制到工程的 `AndroidManifest.xml`：

添加以下两个 `Activity`，其中 `android:screenOrientation` 根据第三方 APP 横竖屏来配置。

注意必须配置 `android:configChanges="orientation|keyboardHidden|screenSize"`，否则小米手机

支付收到短信后转屏有 bug。

```
<!--unionpay sdk begin -->
<activity
    android:name="com.unionpay.uppay.PayActivity"
    android:label="@string/app_name"
    android:screenOrientation="portrait"
    android:configChanges="orientation|keyboardHidden"
    android:excludeFromRecents="true"
    android:windowSoftInputMode="adjustResize"/>
<!--unionpay sdk end -->
```

3.5. 集成支付宝 SDK

libs_alipay 目录是接入支付宝需要集成的文件：

- 1、将 libs 下的 jar 放入 android 工程的 libs 下。
- 2、在 AndroidManifest.xml 配置如下内容

```
<!--alipay sdk begin -->
<activity
    android:name="com.alipay.sdk.app.H5PayActivity"
    android:configChanges="orientation|keyboardHidden|navigation"
    android:exported="false"
    android:screenOrientation="behind" >
</activity>
<activity
    android:name="com.alipay.sdk.auth.AuthActivity"
    android:configChanges="orientation|keyboardHidden|navigation"
    android:exported="false"
    android:screenOrientation="behind" >
</activity>
```

添加混淆规则

```
-libraryjars libs/alipaySDK-20150602.jar
-keep class com.alipay.android.app.IAlipay{*;}
-keep class com.alipay.android.app.IAlipay$Stub{*;}
-keep class com.alipay.android.app.IRemoteServiceCallback{*;}
-keep class com.alipay.android.app.IRemoteServiceCallback$Stub{*;}
-keep class com.alipay.sdk.app.PayTask{ public *;}
-keep class com.alipay.sdk.app.AuthTask{ public *;}
```