



CYPAY 安卓接入文档

CYPaySDK2.2.7

CYPAY 项目组

2014/12/15

目录

1.概述.....	5
2.获取身份.....	6
3.接入 SDK.....	7
3.1 简介.....	7
3.2 Eclipse 导入 SDK.....	8
3.3 应用加载 SDK.....	10
3.4 混淆代码.....	11
3.5 配置 AndroidManifest.xml.....	12
4.SDK 初始化.....	15
5.CYPAY 支付.....	16
5.1 流程介绍.....	16
5.2 支付方式介绍.....	17
5.2.1 普通支付方式.....	17
5.2.2 快捷短代支付.....	17
5.3 支付接口介绍.....	18
5.3.1 CYPay 支付接口.....	18
5.3.2 接收支付结果接口：.....	21

5.4 支付失败错误码 (PaymentErrorCode)	22
6.CYPAY 服务器.....	24
6.1 通知支付结果	24
6.2 确认接收消息	25
6.3 服务器端接收 CYPAY 支付结果回调示例代码.....	26
6.3.1 Servlet 方式处理 CYPAY 服务器端回调	26
6.3.2 Socket 方式处理 CYPAY 服务器端回调.....	27
6.4 订单查询	31
6.4.1 流程介绍	31
6.4.2 接口介绍	31
7.常见问题	33

版本号	修改时间	内容	修改人	确认人
1.0.0	2014-05-11	初稿	李臣	安秋亮； 丁宇； 张义金； 周启铸； 李志超
1.2.1	2014-05-23	增加横版界面支持，init requestconfig 接口变更	李臣	安秋亮
1.2.4	2014-05-28	AndroidManifest.xml 添加内 容，增加对 paypal 信用卡的支 持	李臣	安秋亮
2.0.0	2014-06-11	取消获取商品接口，pay 支付接 口参数变化	李臣	安秋亮； 丁宇；张 义金；周 启铸；李 志超
2.0.1	2014-07-04	支付接口国家简码可以传""	李臣	安秋亮
2.0.3	2014-07-17	服务器通知支付结果加入 cpUse rId 参数	李臣	周启铸， 安秋亮
2.1.0	2014-08-01	AndroidManifest.xml 中加入新的 渠道声明，添加快捷短信支付接 口	李臣	安秋亮， 丁宇
2.1.1	2014-08-12	优化信用卡支付界面	李臣	安秋亮， 丁宇
2.1.2	2014-09-01	修正印尼短信支付问题，添加新 的支付渠道	李臣	安秋亮， 丁宇

2.2.0	2014-09-17	增加用户中心，钱包充值，优化短代支付渠道和支付接口	李臣	安秋亮， 丁宇
2.2.1	2014-09-29	支付接口变更，优化印尼支付渠道。	李臣	安秋亮， 丁宇
2.2.2	2014-10-10	优化 CP 接入体验, AndroidManifest.xml 中填加 <code>android:launchMode="singleTop"</code>	李臣	安秋亮， 丁宇
2.2.4	2014-11-3	优化支付接口	李臣	安秋亮， 丁宇
2.2.5	2014-11-21	蓝色界面风格，更好的横屏支持，支付结果取消 Pending 状态 ,AndroidManifest.xml 中有修改，优先使用余额支付。	李臣	安秋亮， 丁宇
2.2.7	2014-12-15	支付界面直接展示信用卡支付，用户中心提供国际化语言。	李臣	安秋亮， 丁宇

1.概述

本文档描述安卓手机应用接入 CYPAY 支付平台的技术过程。CYPAY 支付平台为安卓应用提供支付服务。

本文档面向的读者是安卓应用开发者，包括客户端和服务端工程师。

接入 CYPAY 支付流程：

1. 向 CYPAY 支付平台申请身份标识，见 [2.获取身份](#)。
2. 加载 SDK，配置 XML，见 [3.接入 SDK 流程](#)。
3. 初始化，见 [4.初始化](#)。
4. 用户点击商品后，调用 CYPAYSDK 支付接口，见 [5.CYPAY 支付](#)。
5. 用户完成支付后，单机应用，CYPAYSDK 会通知客户端支付结果；联网应用，只有支付成功的订单，CYPAY 服务器会回调 URL 通知应用服务器，见 [5.CYPAY 支付](#)，[6.CYPAY 服务器](#)。

2.获取身份

开发者在调用接口之前, 需要先向 CYPAY 支付平台申请 app_key 和 app_secret. App_key 是应用的唯一身份标识, app_secret 相当于是应用的密钥。

如何申请 app_key 和 app_secret ?

请登录 : <http://merchant.wall-et.net/user/register.do>

注意 :

app_key 与应用是一一对应的关系, 若多个应用使用一个 app_key, 会导致统计收入无法按照应用区分。

3.接入 SDK

CYPay SDK 目前支持 Android2.3 API Level9 (含) 以上的系统版本。

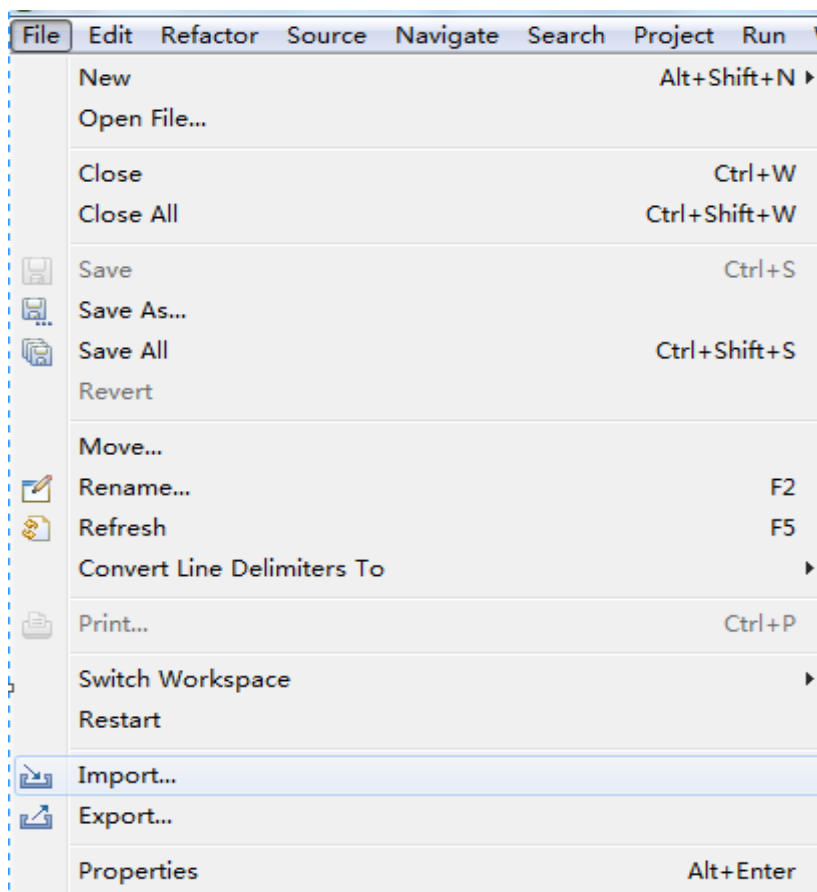
APP 客户端首先要加载 CYPAYSDK 工程 ,配置 AndroidManifest.xml,调用 CYPAYSDK 中的 init 接口初始化 , 当用户选择商品后调用 pay 接口进入支付。

3.1 简介

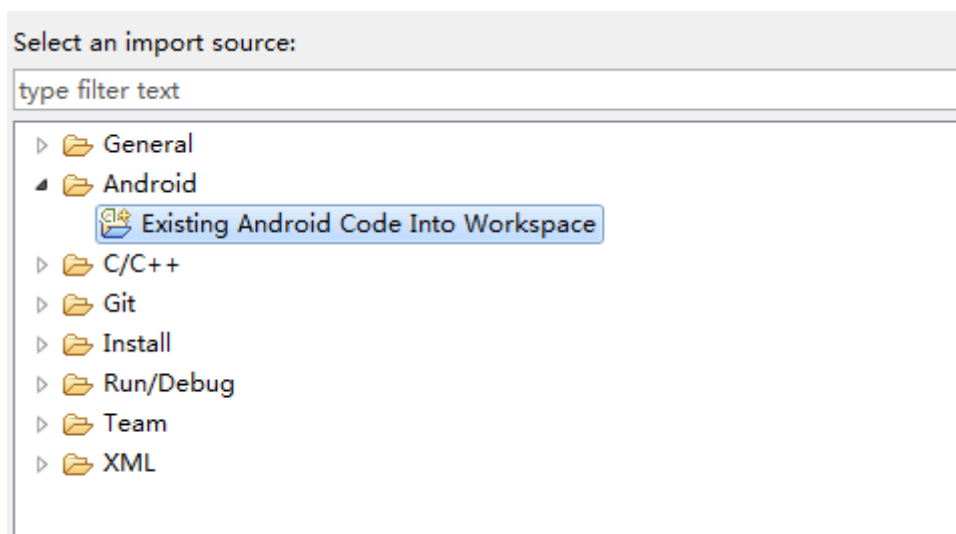
“cypay_sdk_2.*.*.zip” 解压后有三个目录 “docs”、“CYPaySDK” 和 “CYPaySDK_DEMO” , docs 存放接入指导文档和申请文档 , CYPaySDK_DEMO 是使用 CYPaySDK 的示例程序 , CYPaySDK 是要导入的 Library。

3.2 Eclipse 导入 SDK

Eclipse 的导入方法为：**打开 File 列表选择 Import...**



选择 Android 点击 Existing Android Code Into Workspace



点击 *Browse* 找到 *CyPaySDK* 文件夹后，选择 *Finish*

Import Projects

Select a directory to search for existing Android projects



Root Directory:

Projects:

Project to Import	New Project Name
<input checked="" type="checkbox"/> CyPaySDK	CyPaySDK

☐ Copy projects into workspace

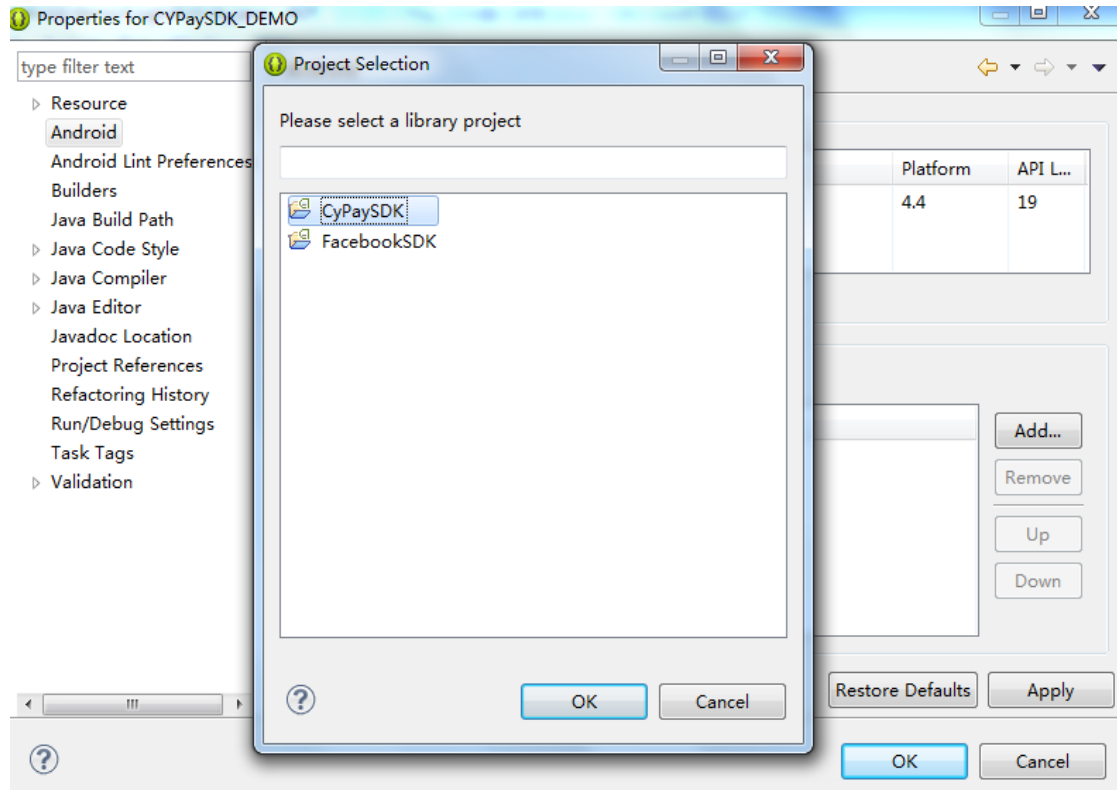
Working sets

☐ Add project to working sets

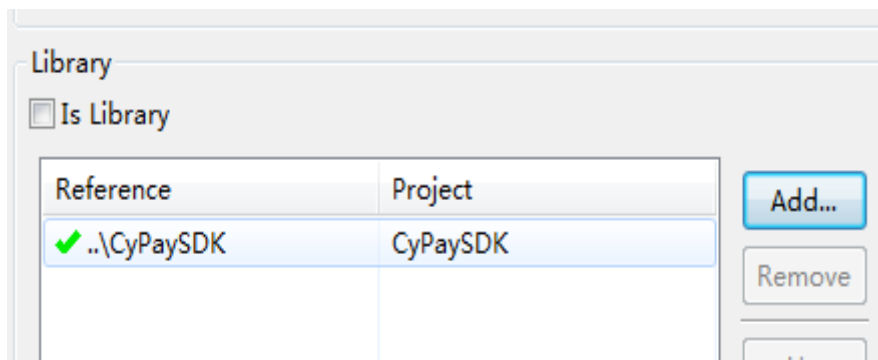
Working sets:

3.3 应用加载 SDK

项目属性中 Library 处(右键单击您的项目名-> Properties -> Android),
点击 Add... , 选中 CYPaySDK 点击 OK。



CYPaySDK 中的 jar 包将自动作为 Android Private Libraries 添加进您的
项目。



3.4 混淆代码

如果要混淆应用代码，请把 CYPAYSDK 中的类排除在混淆之外，可以添加

以下类到 proguard 配置。方法请参考：android-sdk/tools/proguard/

```
-keep class com.cypay.** {*; }
-keep class org.jsoup.** {*; }
-keep class com.mimopay.** {*; }
-keep class com.braintree.** {*; }
-keep class com.braintreegateway.** {*; }
-keep class mp.** {*; }
-keep class com.cypay.paysdk.questionnaire.QuestionnaireWebFragment$WebAppIn
terface{
    public <methods>;
}
-keep class com.android.easy2pay.** {*; }
-keep public class com.cypay.ui.MaxWidthLinearLayout{
    public <fields>;
    public <methods>;
}
-keep public class com.cypay.bean.Price{
    public <fields>;
    public <methods>;
}
-keep public class com.cypay.paysdk.topup.TopUpResultCallback{
    public <fields>;
    public <methods>;
}
```

3.5 配置 AndroidManifest.xml

添加权限，以下是 CYPAY SDK 使用的最低权限

```
<!-- 联网 -->

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

<!-- 手机信息 -->

<uses-permission android:name="android.permission.READ_PHONE_STATE" />

<!-- 用于收发短信 -->

<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.READ_SMS" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />

<!-- 读写 sd 卡的权限 -->

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

<!-- WIFI 状态 -->

<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<!--Fortumo -->
<permission
    android:name="应用的包名"
    android:label="Read payment status"
    android:protectionLevel="signature" />
<uses-permission android:name="应用的包名" />
<!--Fortumo end -->
```

添加 Activity 注意必须放入<application>元素区块内:

```
<!-- CYPay main & topUp & user center -->
<activity
    android:name="com.cypay.paysdk.CYPayMainActivity"
    android:configChanges="orientation|keyboardHidden|screenSize"
    android:label="@string/app_name"
    android:launchMode="singleTop"
    android:screenOrientation="portrait"
```

```
        android:theme="@android:style/Theme.Translucent.NoTitleBar"

        android:windowSoftInputMode="adjustResize|stateVisible|stateAlwaysHidden" >
        </activity>
        <activity
            android:name="com.cypay.paysdk.topup.TopUpFragmentActivity"
            android:configChanges="orientation|keyboardHidden|screenSize"
            android:theme="@android:style/Theme.Translucent.NoTitleBar"

        android:windowSoftInputMode="adjustResize|stateVisible|stateAlwaysHidden" >
        </activity>
        <activity
            android:name="com.cypay.paysdk.usercenter.ui.UserCenterActivity"
            android:configChanges="orientation|keyboardHidden|screenSize"
            android:theme="@android:style/Theme.Translucent.NoTitleBar"
            android:windowSoftInputMode="adjustPan|stateHidden" />
        <meta-data
            android:name="CYPAYSDK_KEY"
            android:value="@string/cypaysdk_app_key" />
        <!-- must use plaintext -->
        <!-- CYPay end -->
        <!--Fortumo -->
        <activity
            android:name="com.cypay.paysdk.channel.fortumo.FortumoActivity"
            android:configChanges="orientation|keyboardHidden|screenSize"
            android:screenOrientation="portrait"
            android:theme="@android:style/Theme.Translucent.NoTitleBar" />
        <activity
            android:name="mp.MpActivity"
            android:configChanges="orientation|keyboardHidden|screenSize"
            android:theme="@android:style/Theme.Translucent.NoTitleBar" />
        <receiver android:name="mp.MpSMSReceiver" >
        <intent-filter>
        <action android:name="android.provider.Telephony.SMS_RECEIVED" />
        </intent-filter>
        </receiver>
        <receiver
            android:name="com.cypay.paysdk.channel.fortumo.PaymentStatusReceiver"
            android:permission="com.cypay.paysdk.PAYMENT_BROADCAST_PERMISSION" >
        <intent-filter>
```

```
        <action android:name="mp.info.PAYMENT_STATUS_CHANGED" />
    </intent-filter>
</receiver>
<service android:name="mp.MpService" />
<service android:name="mp.StatusUpdateService" />
<!--Fortumo end -->
<!-- 2.1.0 add -->
<!-- CreditCardPayment -->
    <activity
        android:name="com.cypay.paysdk.channel.creditcard.CreditCardFragmentActivit
        y"
        android:configChanges="orientation|keyboardHidden|screenSize"
        android:theme="@android:style/Theme.Translucent.NoTitleBar"
        android:windowSoftInputMode="adjustPan|stateHidden" >
    </activity>
<!-- 2.1.2 add -->
<!-- Easy2Pay -->
    <activity
        android:name="com.cypay.paysdk.channel.easy2pay.Easy2PayActivity
        "
        android:configChanges="orientation|keyboardHidden|screenSize"
        android:screenOrientation="portrait"
        android:theme="@android:style/Theme.Translucent.NoTitleBar" />

    <activity android:name="com.android.easy2pay.Easy2PayScreen"
        android:theme="@android:style/Theme.Translucent.NoTitleBar" />
<!-- 2.2.7 add -->
    <activity
        android:name="com.cypay.advertisement.AdvertisementActivity"
        android:theme="@android:style/Theme.Translucent"
        android:configChanges="orientation|keyboardHidden|screenSize"
        android:label="@string/app_name"
        android:screenOrientation="portrait"

        android:windowSoftInputMode="adjustResize|stateVisible|stateAlwaysHidden" >
    </activity>
```

特别注意：

Fortumo 的 permission 内容中的 android:name 必须是应用的包名。

4.SDK 初始化

APP 客户端首先要对 CYPAYSDK 做初始化，推荐在 application 的 OnCreate 方法中调用 CYPay.init，不可重复调用。

接口说明：

```
public static void init(Context context);
```

参数	说明
Context context	ACTIVITY CONTEXT

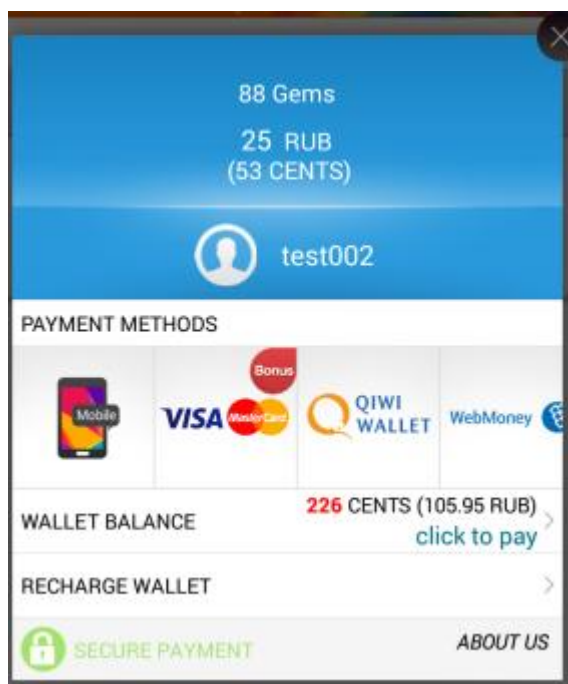
5.CYPAY 支付

5.1 流程介绍



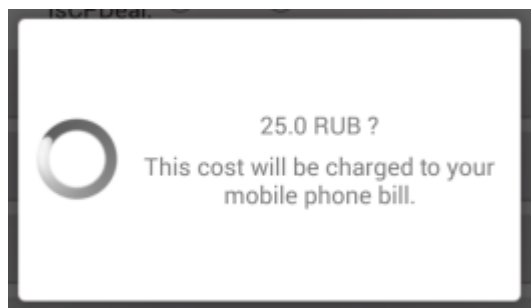
5.2 支付方式介绍

5.2.1 普通支付方式



CYPAY 支付平台提供安全，丰富的本地支付渠道，和信用卡等全球主流支付渠道，适合重度支付的游戏，网络游戏。参考 [5.3 支付接口介绍](#)。

5.2.2 快捷短代支付



快捷短代，为单机游戏，冲动型消费模式量身定做，支付轻松快捷，用户覆盖率高。参考 [5.3 支付接口介绍](#)。

5.3 支付接口介绍

5.3.1 CYPay 支付接口

在支付接口中可以指定支付国家和货币种类，CYPAY 会根据指定的国家显示出对应指定国家的本地支付渠道，并且 CYPAY 支持两种货币类型：

1. "USD"美元

当货币种类为美元时，指定支付的国家后，CYPAY 支付可以根据汇率转换成对应指定国家的本地货币进行支付。

2. 本地货币

当货币种类为本地货币时，国家简码和货币种类代码必须一致，如中国"CN"，货币种类代码为"CNY"人民币。

接口说明：

普通支付接口，调用出支付渠道页，用户选择渠道后支付。

```
public static void pay(Activity activity, Order order)
```

快捷短信支付接口，优先调用短信支付，当无SIM卡或者所在国家不支持短信支付时，商品定价不符合短信支付的计费点时，会转为普通支付方式。短信支付的计费点信息可以查询：<http://console.wall-et.net/country/coverage.jhtml>

```
public static void payWithMobile (Activity activity, Order order)
```

重要：选用此种支付方式时，需要应用显示的商品价格，符合用户设备所在国家的货币类型的定价，如商品 "88 Gems "在巴西定价 2.85 BRL，

印度定价 20 INR，应用可以调用 `getCountryIso` 函数获取设备的国家简码，函数源码参见“7 常见问题”，当判断国家为 BR 时，价格显示 2.85 BRL。为 IN 时，价格显示 20 INR。

参数	说明
Activity activity	APP 客户端的 Activity 实例
Order order	APP客户端传入，详情见下面的Order介绍

Order 介绍

成员	说明
String orderId	APP 客户端生成，每发起一次支付，需要提供一个订单号，订单号应是唯一的。 注意：订单号小于等于 32 字符。
ArrayList<Price> mPriceList	<p>一个商品在各个国家的定价信息，成员参考“Price 介绍”，建议对 CYPAY 支持的国家，按照短代计费点分别定价，定价请参考 http://console.wall-et.net/country/coverage.jhtml，代码请参考“接口实例”。</p> <p>CYPAY 也支持统一美元定价，实例：</p> <pre>Price priceTmp = new Price(); priceTmp.setProductName("88 Gems"); priceTmp.setCountry("DEFAULT"); priceTmp.setCurrency("USD"); priceTmp.setAmount(0.99); order.addPrice(priceTmp);</pre>

String cpOrderTime	APP 服务器下单时间，单机游戏传入当前时间，规则为“yyyy-MM-dd HH:mm:ss”
String appSecret	APP客户端在CYPAY支付平台申请的密钥
String cpUserId	有用户系统的应用传入用户ID，无用户系统传“ ”

Price 介绍

成员	说明
String productName	商品名称，如 “1000 金币”
String country	国家简码，如巴西是 “BR”
String currency	订单付款货币种类代码，如印尼“IDR”
double amount	订单付款金额，不能小于 0.01 美元

接口实例：

```
private String[] countryCodes = {"DEFAULT", "BR", "IQ", "IN",
    "ID", "TH", "RU", "TR", "VN", "MY", "PK", "SA", "EG"};
    //CYPAYSDK自动获取设备国家简码，判断一个商品在不同国家对应的货币
    类型和商品定价。
    //国家简码参数有“DEFAULT”时，会将没有单独定价的国家，选用DEFAULT
    对应的USD货币类型和订单金额。
private String[] currencyCodes = {"USD", "BRL", "IQD", "INR",
    "IDR", "THB", "RUB", "TRY", "VND", "MYR", "PKR", "SAR", "EGP"};
private double[] prices = { 0.99, 2.85, 600, 20,
    5000, 20, 25, 1, 5000, 1, 10, 5, 5};

{
    Order order = new Order();
    for(int i = 0; i < countryCodes.length; i++) {
        Price priceTmp = new Price();
        priceTmp.setProductName("88 Gems");
        priceTmp.setCountry(countryCodes[i]);
```

```
        priceTmp.setCurrency(currencyCodes[i]);
        priceTmp.setAmount(prices[i]);
        order.addPrice(priceTmp);
    }
    order.setOrderID(orderId);
    order.setCpUserId("");
    order.setIsCpDeal(false);
    order.setAppSecret("");
    Date curDate = new Date(System.currentTimeMillis());
    String str = sdf.format(curDate);
    order.setCpOrderTime(str);
    CYPay.payWithMobile(MainActivity.this, order);
}
```

5.3.2 接收支付结果接口：

支付完成后，CYPAY 服务器会回调 URL 通知 APP 服务器支付结果，没有配置 URL 的单机 APP，CYPAYSDK 会在 onActivityResult 消息中，通知 APP 客户端支付结果状态，requestCode 为 CYPay.REQUEST_PAY_CODE。

接口说明：

```
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == CYPay.REQUEST_PAY_CODE) {
        onPaymentResult(resultCode, data);
    }
}

private void onPaymentResult(int resultCode, Intent data) {
    if (resultCode == CYPay.RESULT_PAY_CODE_SUCCESS) {
        PaymentResult result = (PaymentResult) data
```

```

        .getSerializableExtra(CYPay.EXTRA_PAYMENT_RESULT);
    } else if (resultCode == CYPay.RESULT_PAY_CODE_ERROR) {
        PaymentErrorCode errorCode = (PaymentErrorCode) data
            .getSerializableExtra(CYPay.EXTRA_ERROR_CODE);
    }
}

```

PaymentResult 介绍

成员	说明
Order mOrder	用户支付时传入的 Order，参见“4.2.1CYPay 支付接口”的 Order 介绍
PaymentState mPaymentState	支付结果状态： <i>STATUS_FULL</i> ，//7 全额支付
double mRealAmount	用户实际支付的金额
String mRealCurrency	用户实际支付的货币种类代码
double mRate	对美元的汇率，货币种类相同时返回 1
String mCyOrderId	CYPaySDK 生成的订单号在 CYPaySDK 中唯一 —

5.4 支付失败错误码 (PaymentErrorCode)

常量	说明
PRE_PLACE_ORDER_FAILED	下单失败
PAYMENT_FAILED	支付失败
CONNECTION_ERROR	网络错误

USER_CANCELED	用户主动取消支付
CHECK_ORDER_FAILED	传入的 Order 信息错误
PAYMENT_SERVER_ERROR	服务器未知错误

6.CYPAY 服务器

6.1 通知支付结果

支付结果回调 URL 是在 CYPAY 支付平台配置的，请参见《CYPAY 支付平台接入申请》。

每当用户支付完成之后，CYPAY 服务器会向 APP 服务器发送 Http POST 请求商户配置的 callback_url，支持 Http/Https 协议，请求参数如下：

参数	说明
String orderId	返回 APP 客户端生成唯一订单号，见 “5.2.1CYPAY 支付接口” 的参数 orderId
String productName	商品名称
String amount	订单付款金额
String currency	订单付款货币种类代码，如印尼”IDR”
String cypayOrderId	CYPaySDK 生成的订单号，在 CYPaySDK 中唯一
String status	订单状态 status: 7 全额支付
String orderTime	订单更新时间，CYPAY 服务器时间
String realCurrency	用户实际支付的货币种类代码
String realAmount	用户实际支付的金额
String rate	对美元的汇率，货币种类相同时返回 1
String cpUserId	有用户系统的应用传入用户ID，无用户系统传” ”

String signature	加密串,MD5(orderId+cypayOrderId+amount+status + currency+secret)
-------------------------	--

实例：

cypayOrderId=201406061617071026501&status=0&orderTime=2014-06-11%16:09:21¤cy=USD&amount=1.0&realAmount=1.0&realCurrency=USD&rate=1.0&productName=baoshi&signature=a34e3f775b44cf8ce06d5ae1d7cddcbc&orderId=cp001

6.2 确认接收消息

在 APP 服务器接收到 CYPAY 服务器回调的消息后，请返回通知消息 “SUCCESS”。

如果 CYPAY 服务器没有接收到 APP 服务器返回的 “SUCCESS” 消息，则会间隔一段时间重新回调。

时间间隔：

10 秒、10 秒、10 秒、1 小时、10 秒、10 秒、10 秒、1 天。

如果轮询结束仍然没有接收到 “SUCCESS” 消息，则放弃。第三方服务器可调用订单查询接口主动查询订单状态。

6.3 服务器端接收 CYPAY 支付结果回调示例代码

6.3.1 Servlet 方式处理 CYPAY 服务器端回调

```
@Override
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        //APP客户端唯一订单号
        String orderId = request.getParameter("orderId");
        //产品名称
        String productName = request.getParameter("productName");
        //订单付款金额
        String amount = request.getParameter("amount");
        //订单付款币种
        String currency = request.getParameter("currency");
        //CYPAY生成的订单号，在CYPAY中唯一
        String cypayOrderId =
            request.getParameter("cypayOrderId");
        //订单状态 status :
        String status = request.getParameter("status");
        //订单更新时间，为CYPAY服务器时间
        String orderTime = request.getParameter("orderTime");
        //用户实际支付币种
        String realCurrency =
            request.getParameter("realCurrency");
        //用户实际支付金额
        String realAmount = request.getParameter("realAmount");
        //对美元的汇率，货币种类相同时返回 1
        String rate = request.getParameter("rate");
        //用户ID,无用户系统则为""
        String cpUserId = request.getParameter("cpUserId");
        //加密串 ,MD5(orderId+cypayOrderId+amount+status+
            currency+secret)
        String signature = request.getParameter("signature");

        //app密钥
        String secret = "testsecret";
        String signStr = orderId+cypayOrderId+amount+status+
            currency+secret;
```

```
//MD5(orderId+cypayOrderId+amount+status+
currency+secret)
String sign = StrMD5.getStrMD5(signStr);

if(sign.equals(signature)){
    /**游戏端业务逻辑处理***/
} else {
}
//调用returnToCYPAY方法向CYPAY服务器端返回处理结果，处理成功
只需返回字符串"SUCCESS"，处理失败请返回失败原因
returnToCYPAY(response, "");
}

private void returnToCYPAY(HttpServletResponse
response,String content) {
    PrintWriter out = null;
    try {
        out = response.getWriter();
        out.write(content);
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if(out!=null){
            out.close();
        }
    }
}
```

6.3.2 Socket 方式处理 CYPAY 服务器端回调

```
public static void main(String args[]) throws IOException {
    int port = 80;
    // 定义ServerSocket监听在端口80上
    ServerSocket server = new ServerSocket(port);
    Socket socket = server.accept();
    try {
        BufferedReader reader = new BufferedReader(new
```

```
InputStreamReader(  
    (socket.getInputStream())));  
  
    /**  
     * 解析HTTP请求，获取CYPAY请求内容  
     */  
    StringBuffer postContentBuffer =  
dealHttpRequest(reader);  
  
    /**  
     * 处理CYPAY请求内容  
     */  
    dealCYPAYPostContent(postContentBuffer, socket);  
    reader.close();  
} catch (Exception e) {  
    e.printStackTrace();  
} finally {  
    try {  
        // 关闭请求连接  
        socket.close();  
    } catch (IOException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}  
server.close();  
}  
  
private static StringBuffer dealHttpRequest(BufferedReader  
reader)  
    throws IOException {  
    // 存储请求内容  
    StringBuffer postContentBuffer = new StringBuffer();  
    // 请求内容长度  
    int contentLength = 0;  
    // 读取HTTP请求行以获取请求方法：POST或GET  
    String line = reader.readLine();  
    String[] requestLines = line.split(" ");  
    // 获取请求方法，GET 或者 POST  
    String method = requestLines[0];  
    // 获取请求路径，并检测路径是否正确，此处假设回调路径为  
    /cypayCallBack  
    String path = requestLines[1];
```

```
if (path.equals("/cypayCallBack")) {
    while ((line = reader.readLine()) != null) {
        // 获取HTTP请求内容长度
        if (line.startsWith("Content-Length")) {
            contentLength = Integer.parseInt(line.substring(
                line.indexOf(':') + 1).trim());
        }
        // HTTP协议中报头与正文之间以一空行隔开
        if ("".equals(line)) {
            break;
        }
    }
    // 读取CYPAY服务器端请求内容，CYPAY是以POST方式进行请求的
    if ("POST".equalsIgnoreCase(method) && contentLength >
0) {
        for (int i = 0; i < contentLength; i++) {
            postContentBuffer.append((char) reader.read());
        }
    }
    return postContentBuffer;
}

/**
 * 对POST过来的内容进行解析
 *
 * @param postContentBuffer
 * @throws IOException
 */
private static void dealCYPAYPostContent(StringBuffer
postContentBuffer,
    Socket socket) throws IOException {
    Map<String, String> paramMap = new HashMap<String,
String>();
    String postContent = postContentBuffer.toString();
    String[] contents = postContent.split("&");
    for (int i = 0; i < contents.length; i++) {
        String content = contents[i];
        String[] con = content.split("=");
        String key = con[0];
        String value = "";
        if (con.length == 2) {
```

```
        value = con[1];
    }
    paramMap.put(key, value);
}

// APP客户端唯一订单号
String orderId = paramMap.get("orderId");
// 产品名称
String productName = paramMap.get("productName");
// 订单付款金额
String amount = paramMap.get("amount");
// 订单付款币种
String currency = paramMap.get("currency");
// CYPAY生成的订单号, 在CYPAY中唯一
String cypayOrderId = paramMap.get("cypayOrderId");
// 订单状态 status
String status = paramMap.get("status");
// 订单更新时间, 为CYPAY服务器时间
String orderTime = paramMap.get("orderTime");
// 用户实际支付币种
String realCurrency = paramMap.get("realCurrency");
// 用户实际支付金额
String realAmount = paramMap.get("realAmount");
// 对美元的汇率, 货币种类相同时返回 1
String rate = paramMap.get("rate");
// 用户ID, 无用户系统则为""
String cpUserId = paramMap.get("cpUserId");
// 加密
串 ,MD5(orderId+cypayOrderId+amount+status+currency+secret)
String signature = paramMap.get("signature");
// app密钥
String secret = "testsecret";
String signStr = orderId + cypayOrderId + amount + status
+ currency + secret;
// MD5(orderId+cypayOrderId+amount+status+
currency+secret)
String sign = StrMD5.getStrMD5(signStr);
if (sign.equals(signature)) {
    /*****游戏服务器端业务逻辑处理*****/
    /*****
} else {
}
```

```
// 获取socket的outputStream，向CYPAY服务器端返回处理结果，处理成功只需返回字符串"SUCCESS"，处理失败请返回失败原因
PrintWriter out = new
PrintWriter(socket.getOutputStream());
    out.write("");
    out.flush();
    out.close();
}
```

6.4 订单查询

6.4.1 流程介绍

APP 服务器主动到 CYPAY 支付平台查询订单状态。

6.4.2 接口介绍

url :

<http://core.wall-et.net/order/check/checkOrder>

实例 :

<http://core.wall-et.net/order/check/checkOrder?orderId=eae9a6cce1c34c248d2660507c9ab4f3&cypayOrderId=201405042113001015505&signature=e42946fa33eb1cd3be30078b17160b5d>

参数	说明
orderId	返回 APP 客户端生成唯一订单号，见 “5.2.1CYPAY 支付接口” 的参数 orderId
cypayOrderId	CYPaySDK 生成的订单号，在 CYPaySDK 中唯一
signature	MD5 校验值，MD5(orderId + cypayOrderId + secret)

成功：

```
{  
  code: 0,  
  msg: "success",  
  data: {  
    cypayOrderId: "201406061617071026501",  
    amount: "1.0",  
    rate: "1.0",  
    status: "0",  
    realCurrency: "USD",  
    realAmount: "1.0",  
    orderTime: "2014-06-11 16:09:21",  
    productName: "baoshi",  
    signature: "a34e3f775b44cf8ce06d5ae1d7cddcbc",  
    orderId: "cp001",  
    currency: "USD"  
  }  
}
```

失败：

```
{"data": {}, "code": -1, "msg": "token md5 error"}
```

7.常见问题

1. 接入 SDK 后，编译时出现错误

错误原因：

SDK 工程中的 android-support-v4.jar 与应用工程中的版本不同，导致冲突。

解决方法：

用 APP 工程中的 android-support-v4.jar 替换到 SDK 工程中。

2. 调用 pay 接口后显示不出支付页或返回 PRE_PLACE_ORDER_FAILED

错误原因：

通常是 pay 接口参数传入不正确

a.传入国家简码与货币类型代码不匹配。

b.app_key 与 app_secret 不匹配。

c.商品价格小于 0.01 美元。

d.Activity 传入无效的值。

e.配置了用户体系的应用，没有传入 cpUserId。

解决方法：

输出 Log，排查 pay 接口参数传入值是否正确。

3. 如何获取设备的国家简码

```
public static String getCountryIso(Context context) {  
    TelephonyManager tm = (TelephonyManager) context.getSystemService(Context.TELEPHONY_SERVICE);  
    String iso = "";  
    if (tm != null) {  
        iso = tm.getSimCountryIso();  
    }  
    if (iso.equals("")) {  
        iso=context.getResources().getConfiguration().locale.getCountry()  
();  
    }  
    return iso.trim();  
}
```

注意：获取到的国家简码大小写都可能存在。

4. 测试支付

推荐使用 CYPAY 提供的支付渠道进行真实的支付验证，最低订单金额支持 0.02 美元。

如需测试账号进行支付验证，可以和 CYPAY 申请。