CYPAY Android Access Guideline

CYPaySDK2.2.7

CYPAY Project Team

2014/12/15

Table of Contents

Version	Date of Revision	Revisions	Written By	Confirmed By
1.0.0	2014-05-11	First Draft	Li chen	An Qiuliang, Ding Yu, Zhang Yijin, Zhou Qizhu, Li Zhichao
1.2.1	2014-05-23	Horizontal version，init requestconfig API changes	Li chen	An Qiuliang
1.2.4	2014-05-28	AndroidManifest.xml add content，support paypal credit card	Li chen	An Qiuliang
2.0.0	2014-06-11	Cancel item request api, pay api parameter changes	Li chen	An Qiuliang, Ding Yu, Zhang Yijin, Zhou Qizhu, Li Zhichao
2.0.3	2014-07-17	CYPAY Server Inform Payment Result “cpUserid”	Li Chen	An Qiuliang, Zhou Qizhu
2.1.0	2014-08-01	Disclaimer is added in AndroidManifest.xml. Direct Carrier Billing is added.	Li Chen	An Qiuliang, Ding Yu,Li Zhi Chao
2.1.1	2014-08-12	UI of Credit Card payment is improved	Li Chen	An Qiuliang, Ding Yu,Li Zhi Chao
2.1.2	2014-09-01	Correction of Indonesia SMS payment problems, add new payment channels	Li chen	An Qiuliang, Ding Yu
2.2.0	2014-09-17	Refined 1-click SMS payment and existing channels. Provided Account and Customer Service system along with the ability to recharge wallet balance	Li chen	An Qiuliang, Ding Yu
2.2.1	2014-09-29	Payment API change， Indonesian channel improved	Li chen	An Qiuliang, Ding Yu
2.2.2	2014-10-10	Please add android:launchMode=*"singleTop"* Into AndroidManifest.xml	Li chen	An Qiuliang, Ding Yu
2.2.4	2014-11-3	API improved	Li chen	An Qiuliang, Ding Yu
2.2.5	2014-11-21	Blue color UI， cross-screen support, Pending status is cancelled, changes of AndroidManifest.xml, Wallet balance has priority of being used for pay	Li chen	An Qiuliang, Ding Yu

2.2.7	2014-12-15	Credit Card is shown above in payment selection page, Five languages supported in Account Management	Li chen	An Qiuliang, Ding Yu

1.Overview

This guideline serves as a technical description of how Android apps will access the CYPAY payment platform, which provides payment service for Android apps.

This guideline is intended for Android app developers, including client engineers and server engineers.

1. Apply for unique ID from CYPAY，Please see 2 Acquire an ID.
2. Load SDK，set XML，Please see 3 Access SDK.
3. Initialization before CYPAYSDK API callback，Please see 4 Initialization.
4. After user click at item to buy, client build order, invoke CYPAYSDK payment API, Please see 5 CYPAY Payment.
5. After recharge is finished, for Client, CYPAYSDK will notify Client for the result; for Online APPs, only successful orders, CYPAY server will invoke URL to notify APP server, Please see 5 CYPAY Payment, 6 CYPAY Server .

2.Acquire an ID

Before calling the API, developers should apply for app_key and app_secret from the CYPAY payment platform. The app_key is the unique ID of the app, and the app_secret serves as the secret key of the app. It is recommended that developers don't save app_secret in the client, as there is a risk of leaking.

How to apply for app_key and app_secret?

Please see Access Application for CYPAY Payment Platform.

Note:

app_key is specifically used for only one app. If several apps use on e same key, then the income can not be divided by keys.

3.Access SDK

At present, CYPay SDK supports Android2.3 API Level9 or higher version.

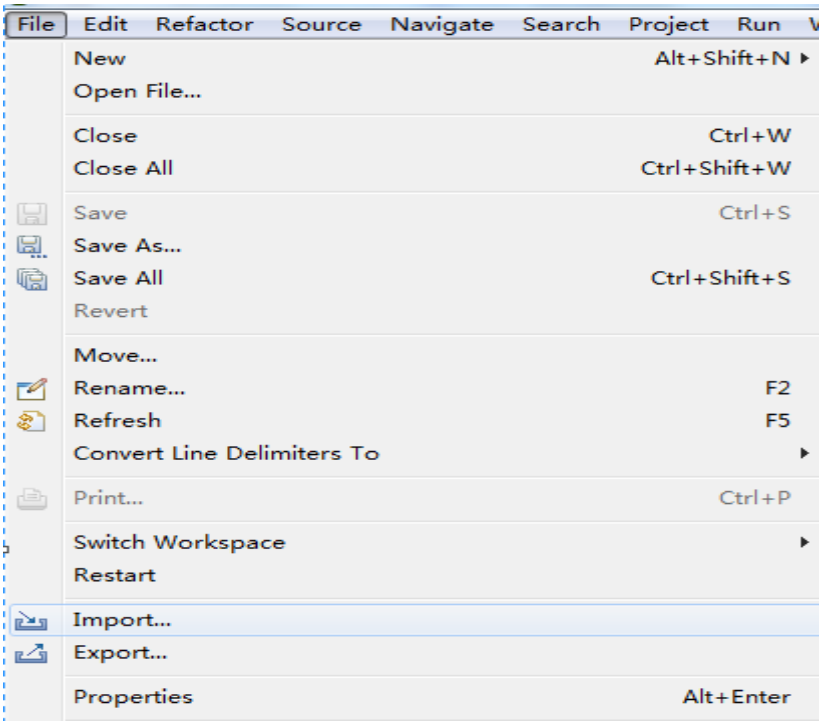
APP client must load CYPAYSDK, set AndroidManifest.xml, invoke init API initialization in CYPAYSDK, after user selecting items, pay API is invoked to begin payment.

3.1 Introduction

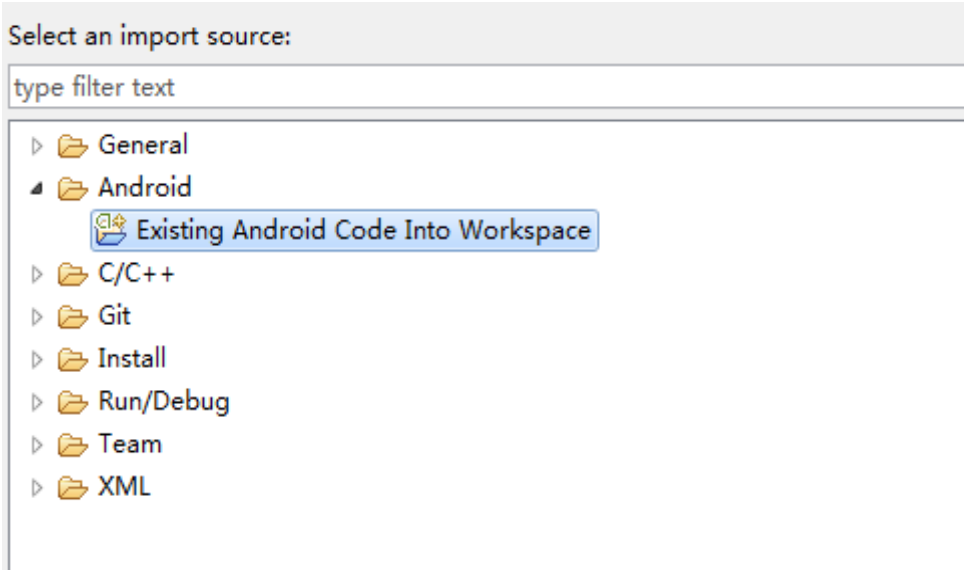
After unzipping "cypay_sdk_1.*.*.zip" you will find three folders: "docs", "CYPaySDK" and "CYPaySDK_DEMO". Docs stores access guideline documents; CYPaySDK_DEMO is a demo program on how to use CYPaySDK; CYPaySDK is the Library to be imported.

3.2 Import SDK in Eclipse

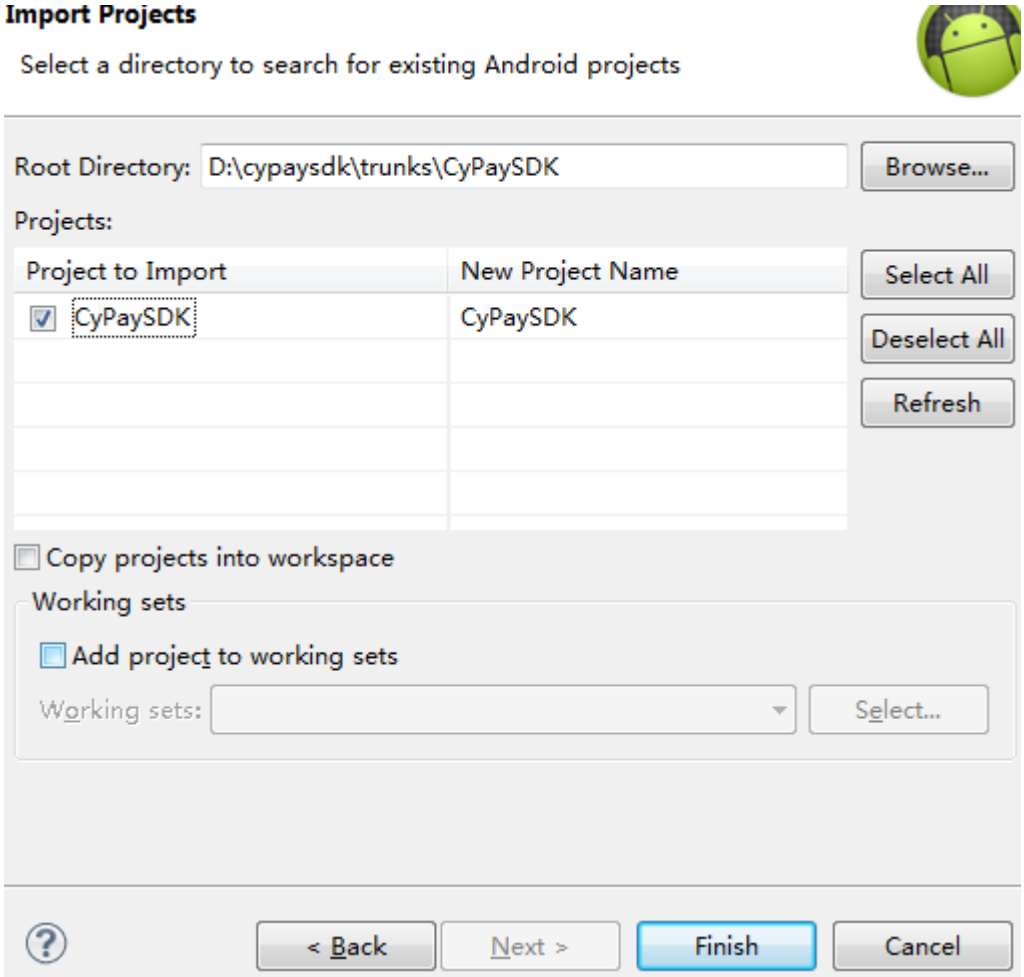
To import the SDK in Eclipse, please ***open File menu and choose Import…***



Choose Android, click Existing Android Code Into Workspace

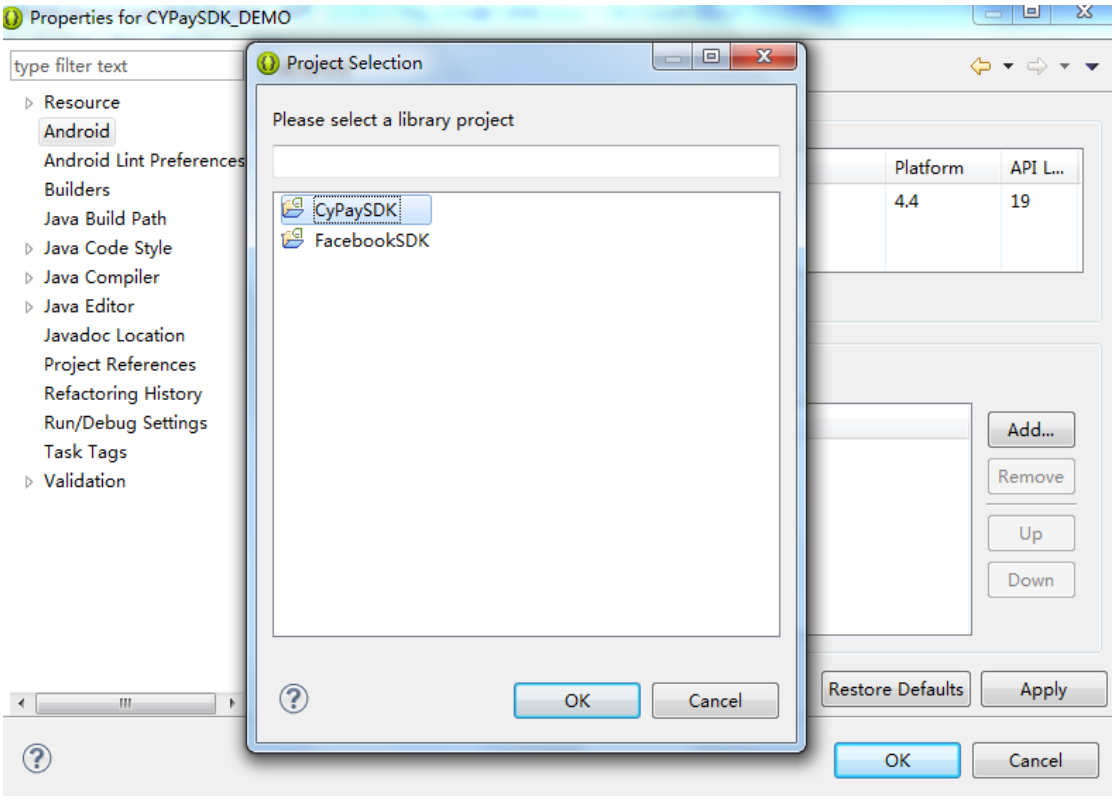


Click Browse, find CyPaySDK folder, choose Finish

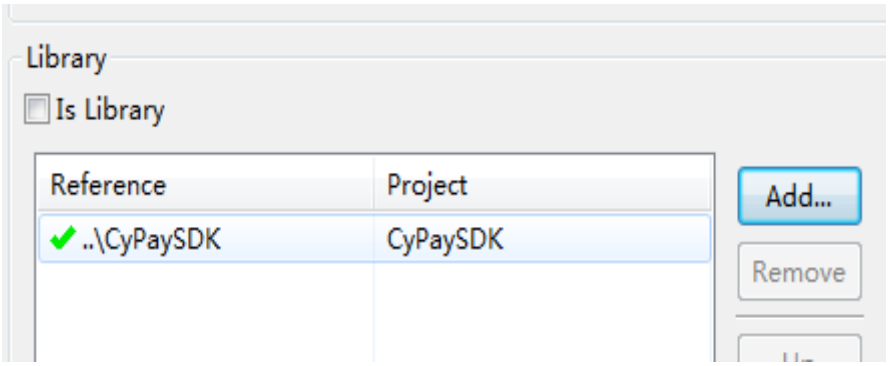


3.3 Add SDK in App

Find Library in project properties (right click project name -> Properties -> Android), click Add…, choose CYPaySDK, and click OK.



The jar package in CYPaySDK will be added to your project automatically as Android Private Libraries.



3.4 Code Obfuscation

If you want to obfuscate app code, please exclude classes in CYPAYSDK. You can add the following classes in proguard configuration. For more details, see: 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$WebAppInterface{
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 Configure AndroidManifest.xml

Add permissions. The following are the minimal permissions

```
    <!-- Network -->
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <!—Phone State -->
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <!—Send and Receive SMS -->
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.READ_SMS" />
```

```
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
     <!—Load and Write SD Card -->
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
     <!—WIFI State -->
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
  <!-- Fortumo -->
    <permission
        android:name=app package name
        android:label="Read payment status"
        android:protectionLevel="signature" />

    <uses-permission android:name= app package name />     <!-- Fortumo end -->
```

Activity must be put in <application> element block:

```
<!-- 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.PaymentStatusRece
iver”
            android:permission=”com.cypay.paysdk.PAYMENT_BROADCAST_PERMISSIO
N” >
            <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>
```

Please Note：

android:name in Permission of Fortumo must be the name of app packet.

4. SDK Initialization

The APP client will firstly initialize the CYPAY SDK. It is recommended that you call CYPay.init in OnCreate method of application. It can't be called repeatedly.

API Description:

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

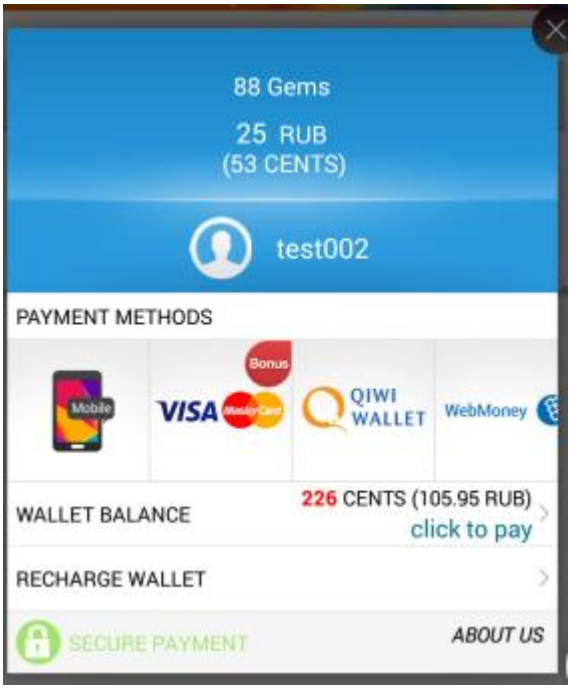
Parameter	Description
Context context	ACTIVITY CONTEXT

5. CYPAY Payment

5.1 Process Description

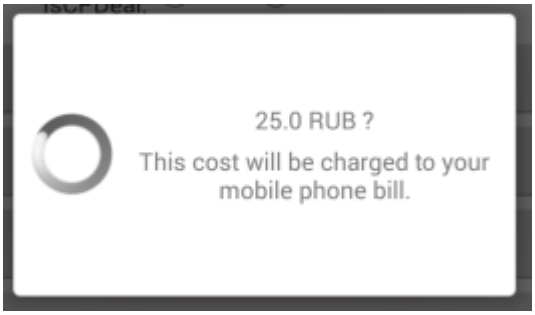
5.2 Payment method introduction

5.2.1 Common payment



CYPAY provide secure and main local channels, credit card payment, Paypal and other channels. It is for hardcore online games. Please refer to 5.3 Payment API Description

5.2.2 Direct carrier billing payment



Direct carrier billing payment, which is suitable for single game, is more suitable for impetuous consumption games. It can cover most carriers and pay conveniently. Please refer to 5.3 Payment API Description

5.3.1 CYPay Payment API

In recharge api, developers can set country and currency. CYPAY will show local payment channels according to the countries. CYPAY support two kind of currencies.

1. **USD**
   If the currency is USD, CYPAY can convert it to the local country's currency according to the exchange rate after you set the payment country.
2. **Local Currency**

If the currency is local currency, the country code and currency code must fit, e.g. "CN" for China and "CNY" for RMB.

API Description:

In General API, channel list page will be shown for users to select and pay.

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

Direct carrier billing API, evokes carrier billing firstly. If no SIM card inserted or carrier billing is not supported in that country, or price is not same as price points, common payment will be evoked. Price points of carrier billing can be found below

http://console.wall-et.net/country/coverage.jhtml

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

Notice, when Direct carrier billing payment is used, price of item shown in the app should match the device user's country carrier billing price points. For example, 88Gems' price is 2.85 BRL in Brazil while 20 INR in India. App can evoke getCountryIso for getting country code. Sample can be seen in 7 CP Integration FAQ. When get BR, price is 2.85 BRL while IN price is 20 INR.

Parameter	Description
Activity activity	APP client 's Activity sample
Order order	`Passed by APP client, please refer to order description`

Order Description

Member	Description
String orderId	Created by APP client. Each payment requires a order id, which is unique. Note: Order id can't exceed 32 characters.

ArrayList<Price> mPriceList	One item's price info in different countries. Please refer to Price Description, for countries which CYPAY carrier billing supported, please set price according to different countries price points. Price set, please review below link
	http://console.wall-et.net/country/coverage.jhtml
	For code please review API Sample
	
	CYPAY also support USD price set, sample
	Price priceTmp = **new** Price();
	priceTmp.setProductName("88 Gems");
	priceTmp.setCountry("DEFAULT");
	priceTmp.setCurrency("USD");
	priceTmp.setAmount(1);
	order.addPrice(priceTmp);

String cpOrderTime	Order time on the APP server. Client game should sent local time. The form should be "yyyy-MM-dd HH:mm:ss"
String appSecret	The secret key applied by APP client on CYPAY payment platform.
String cpUserId	For apps with a user system, send user ID; or send" "

Price Description

Member	Description
String productName	Product name, e.g. "1000 Gold"
String country	Country code, e.g. "ID" for Indonesia. Getting from SIM card is preferred. If it can not be get from SIM card you can get it from mobile settings.
String currency	Order payment currency code, e.g. "IDR" for Indonesia Rupiah.
double amount	Order payment amount. Can be less than 0.01 USD.

API Example:

```
private String[] countryCodes = {"DEFAULT", "BR", "IQ", "IN",
"ID", "TH","RU", "TR", "VN", "MY", "PK",  "SA",  "EG"};
   //CYPAY SDK get country code automatically, it can identify
an item's price in different countries.
   //If DEFAULT exists in country code, countries which has no
specific price,
   //will use USD as currency and DEFAULT as country code.
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.setAppSecret("");
     Date curDate = new Date(System.currentTimeMillis());
     String str = sdf.format(curDate);
     order.setCpOrderTime(str);
}
```

5.3.2 Payment Result Callback API:

When payment is finished, CYPAY server will call URL to notify APP server the payment result. If single APP without URL setting, in `onActivityResult`, `CYPAYSDK will notify APP client for payment result. requestCode is CYPay.REQUEST_PAY_CODE`。

API Description:

```
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 Description

Member	Description
Order mOrder	Order sent when user pays. See Order Description in "4.2.1 CYPay Payment API".
PaymentState mPaymentState	Payment result state: *STATUS_FULL*, //7 Full payment
double mRealAmount	Actual amount paid by user
String mRealCurrency	Currency code paid by user
double mRate	Exchange rate to USD. Return 1 if currency codes are the same.
String mCpOrderId	Order ids created by CYPaySDK are unique in CYPaySDK

5. 4 Payment Error Code (PaymentErrorCode)

Constant	Description
PRE_PLACE_ORDER_FAILED	Failed to place order
PAYMENT_FAILED	Payment failure
CONNECTION_ERROR	Network connection error
USER_CANCELED	Payment canceled by user
CHECK_ORDER_FAILED	Wrong order information
PAYMENT_SERVER_ERROR	Unknown server error

6.CYPAY Server

6.1 Inform Payment Result

The payment result callback URL is configured on CYPAY payment platform. Please see *CYPAY Payment Platform Access Application* for details.

Whenever the user completes a payment, CYPAY server will send a Http POST to get seller's callback_url configuration to the APP server, which supports Http and Https. The parameters are as follows:

Parameter	Description
String orderId	Return the unique order id created by APP client. See ordered parameter in "5.2.1 CYPAY Payment API"
String productName	Product name
String amount	Order payment amount
String currency	Order payment currency code, e.g. "IDR" for Indonesia Rupiah
String cypayOrderId	Order id created by CYPaySDK. Unique in CYPaySDK.
String status	Order status: 7 full payment
String orderTime	Order update time. CYPAY server time.
String realCurrency	Actual currency code paid by user
String realAmount	Actual amount paid by user
String Rate	Exchange rate to USD. Return 1 when currency codes are the same.
String cpUserId	For apps with a user system, send user ID; or send" "
String signature	Encrypted string, MD5(orderId+cypayOrderId+amount+status+ currency+secret)

Example:

cypayOrderId=20140606161707102650l&status=0&orderTime=2014-06-11%16:09:21&currency=USD&amount=1.0&realAmount=1.0&realCurrency=USD&rate=1.0&productName=baoshi&signature=a34e3f775b44cf8ce06d5ae1d7cddcbc&orderId=cp001

6.2 Reception Confirmation Information

After the app server has received the information called back by CYPAY server, please return the notification “SUCCESS”.

If CYPAY server doesn’t get the “SUCCESS” notification returned by the app server, it will callback again after some time.

Time interval:

10 seconds, 10 seconds, 10 seconds, 1 hour, 10 seconds, 10 seconds, 10 seconds, 1 day.

If it still can’t receive “SUCCESS”, it will abort. Third party servers can call the order query API to query order status.6.3 Order Query

6.3 Server Receive CYPAY Payment Result Callback Code Sample

6.3.1 Servlet Deal With CYPAY Server Callback

```
@Override
   protected void doPost(HttpServletRequest request,
HttpServletResponse response)
         throws ServletException, IOException {
      //APP client order NO which is unique
      String orderId = request.getParameter("orderId");
      //Item name in game
      String productName = request.getParameter("productName");
      //Amount when pay in the order
      String amount = request.getParameter("amount");
      //Order currency when payment is made
      String currency = request.getParameter("currency");
      //CYPAY order NO which is unique
      String cypayOrderId =
request.getParameter("cypayOrderId");
      //Order status :
      String status = request.getParameter("status");
      //Order update time which is CYPAY server time
   String orderTime = request.getParameter("orderTime");
      //Currency when user is finally recharged
      String realCurrency =
request.getParameter("realCurrency");
      //Amount when user is finally recharged
      String realAmount = request.getParameter("realAmount");
```

```
        //Exchange rate to USD, if USD return 1
        String rate = request.getParameter("rate");
        //User ID, if no user system, use""
        String cpUserId = request.getParameter("cpUserId");
        //Encrypted
string ,MD5(orderId+cypayOrderId+amount+status+ currency+secret)
        String signature = request.getParameter("signature");

        //app secret
        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)){
            /***********APP Service Logic ***************/
            /***************************************/
        }else{
        }
        //Evoke returnToCYPAY to return result to CYPAY server. If
success just return string "SUCCESS", if failed please return
reason why it failed
        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 Deal With CYPAY Server Callback

```
   public static void main(String args[]) throws IOException {
      int port = 80;
      // Define ServerSocket listening on port 80
      ServerSocket server = new ServerSocket(port);
      Socket socket = server.accept();
      try {
         BufferedReader reader = new BufferedReader(new
InputStreamReader(
               (socket.getInputStream())));
         /**
          * Resolve HTTP request, obtain CYPAY request content
          */
         StringBuffer postContentBuffer =
dealHttpRequest(reader);

         /**
          * Deal with CYPAY request content
          */
         dealCYPAYPostContent(postContentBuffer, socket);
         reader.close();
      } catch (Exception e) {
         e.printStackTrace();
      } finally {
         try {
            // Close request link
            socket.close();
         } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
         }
      }
      server.close();
   }

   private static StringBuffer dealHttpRequest(BufferedReader
reader)
         throws IOException {
      // Restore request content
      StringBuffer postContentBuffer = new StringBuffer();
      // Request content lengh
      int contentLength = 0;
```

```
        // Read HTTP request to get request method, POST or GET
    String line = reader.readLine();
        String[] requestLines = line.split(" ");
        // Get request method, POST or GET
        String method = requestLines[0];
        // Get request routine, check whether it is correct, consider
the callback routine is /cypayCallBack
        String path = requestLines[1];
        if (path.equals("/cypayCallBack")) {
            while ((line = reader.readLine()) != null) {
                // Get HTTP request content length
                if (line.startsWith("Content-Length")) {
                    contentLength = Integer.parseInt(line.substring(
                            line.indexOf(':') + 1).trim());
                }
                // One blank line must stand between HTTP protocol
header and body
                if ("".equals(line)) {
                    break;
                }
            }
            // Read CYPAY server request content, CYPAY request in
POST method
            if ("POST".equalsIgnoreCase(method) && contentLength >
0) {
                for (int i = 0; i < contentLength; i++) {
                    postContentBuffer.append((char) reader.read());
                }
            }
        }
        return postContentBuffer;
    }

    /**
     * Resolve POST content
     *
     * @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 client order NO which is unique
        String orderId = paramMap.get("orderId");
        //Item name in game
        String productName = paramMap.get("productName");
        //Amount when pay in the order
        String amount = paramMap.get("amount");
        //Order currency when payment is made
        String currency = paramMap.get("currency");
        //CYPAY order NO which is unique
        String cypayOrderId = paramMap.get("cypayOrderId");
        //order status
        String status = paramMap.get("status");
        //Order update time which is CYPAY server time
        String orderTime = paramMap.get("orderTime");
        //Currency when user is finally recharged
        String realCurrency = paramMap.get("realCurrency");
        //Amount when user is finally recharged
        String realAmount = paramMap.get("realAmount");
        //Exchange rate to USD, if USD return 1
        String rate = paramMap.get("rate");
        //User ID, if no user system, use""
        String cpUserId = paramMap.get("cpUserId");
        //Encrypted
MD5(orderId+cypayOrderId+amount+status+currency+secret)
        String signature = paramMap.get("signature");
        //app secret
        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)) {
            /***************** APP Service Logic ********/
            /*********************************************/
        } else {
        }
        //outputStream which get from socket,  return result to
CYPAY server. If result is successful, please return "SUCCESS"
, if failed please return reason.

        PrintWriter out = new
PrintWriter(socket.getOutputStream());
        out.write("");
        out.flush();
        out.close();
    }
```

6.4 Check Order

6.4.1 Process Description

The app server initiates a query for order status on CYPAY payment platform.

6.4.2 API Description

url:
http://core.wall-et.net/order/check/checkOrder
Example:
http://core.wall-et.net/order/check/checkOrder?orderId=eae9a6cce1c34c248d2660507c9ab4f3&cypayOrderId=20140504211300101550 5&signature=e42946fa33eb1cd3be30078b17160b5d

Parameter	Description
orderId	Returns unique order ID generated by the app client. See ordered parameter in ”5.2.1 CYPAY Payment API Description”.
cypayOrderId	Order ID created by CYPaySDK. Unique in CYPaySDK.
signature	MD5 checksum, MD5(orderId + cypayOrderId + secret

Success:

{
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"
}
}

Failed

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

7.CP Integration FAQ

1. After integration of SDK, error when compile

Reason, android-support-v4.jar version of SDK project file is different from APP project file's. Confliction is there.

Resolution, replace SDK project file's android-support-v4.jar by APP project file's

2. When using pay api, payment channels can not be shown or PRE_PLACE_ORDER_FAILED

Reason, usually it is caused by wrong parameter passing

a country code and currency not match

b app_key and app_secret not match

c price is less than 0.01 USD

d invalid value passed in Activity

e games which have account system with no cpUserId passed

Resolution, get the log, check whether correct parameter is passed in pay api

3. How to get device country code

```
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();
}
```

Note: country code get from device may capitalized

4. How to test payment

Please use real money for payment test, lowest price can be 0.02USD.

If test account is needed, please apply from CYPAY.