Citcon Pay Android SDK Documentation

Version 2.1.1

I.   Introduction

Citcon's Android SDK was designed for online merchants to integrate Citcon payment solutions effortlessly into their own iOS apps. By using the SDK, merchant developers can focus on business logics without having to understand the plumbing of payment transactions. The payment experience will be totally transparent and seamless to end consumers. This version of the SDK only supports payments through Alipay and WeChat Pay. Union Pay, credit cards and other payment methods will be added to future versions. The version of the SDK can process Alipay/WeChat Pay transactions using the following currencies: USD, CAD and CNY. Other currencies will be supported by future versions.

Although it's not necessary for merchant's engineering team to have in-depth knowledge of how Citcon Pay is integrated under the hood, an exhibition of the payment transaction flow will help developers better understand the integration process at the conceptual level, as illustrated in Figure.1 below.
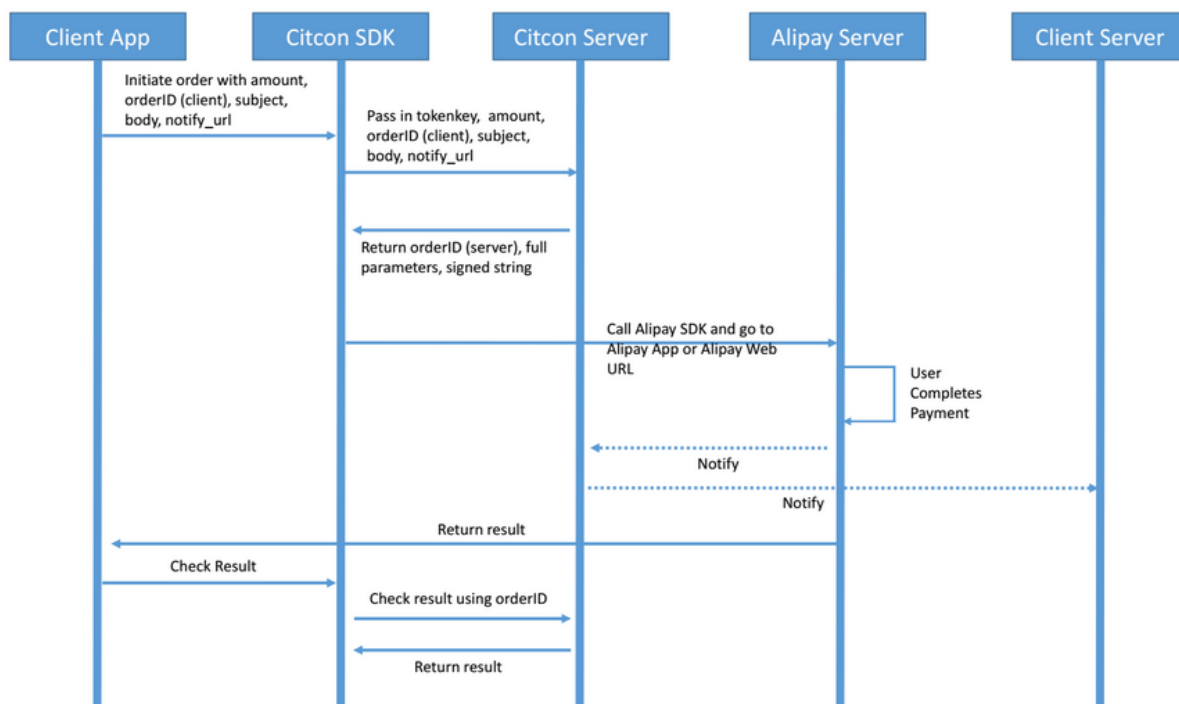

Figure 1 – Payment transaction flow from merchant app to payment processor through Citcon

The Citcon Android development SDK was developed in JAVA and targets Android 4.0.3 and above. The SDK is distributed as an Android Archive Library (aar).

II.   The Citcon Pay Framework

This section details the main components of the Citcon Pay Framework for Android development. The commonly used header files and their purposes will be listed here, and a step-by-step example of integrating the framework in a demo merchant app will be shown in the next section.

1. Library identity

    a. Name: citcon-sdk.aar

    b. Package Name: citcon.sdk

    c. Current version: 2.0

2. Required dependencies

    • com.android.support:appcompat-v7:25.3.1

    • com.android.support:multidex:1.0.1

3. Class references

    a. **CPaySDK**

    The CPaySDK class performs the most common payment related tasks: set up merchant token, send order to Citcon Pay and query the status of a specific transaction.

    Method summary

Name	Description
requestOrder	Sends order information to Citcon and initiates the payment transaction. Has an unique callback: gotOrder.
inquireOrder	Query the status of a specific payment transaction. Has an unique callback: inquiredOrder.
gotOrder	Callback of requestOrder. This method handles the outcome of the payment transaction initiation and continues the process by either Alipay or WeChat.
setMode	Set SDK running mode. If not set, default mode is CPayMode.PROD.

b. CPayOrder

Represents the order and payment information merchant wants to send to Citcon Pay for processing.

Constructor Property summary

Name	Type	Description	Example
referenceId	String	A reference merchant creates and assigns to the transaction.	123xyz
subject	String	A customer-defined description of the transaction	Gift for Mom
body	String	A more detailed customer-defined description of the transaction	A Blu-ray player and a few great movies
amount	String	Total charge amount of the transaction in cents.	245 ($2.45)
currency	String	The type of currency defined in a three-letter code	USD or CAD or CNY
vendor	String	The name of vendor that will process the payment. Only "Alipay" is supported in this version of the SDK	alipay or wechatpay
ipnUrl	String	The URL of a page Citcon Pay should post transaction status to. Normally this should be a page on the merchant's website.	http://www.xyz.com/notify.php
callbackUrl	String	The URL of a page Citcon Pay should redirect customer to after the payment transaction has completed.	http://xyz.com/confirm.php
allowDuplicate	boolean	Flag to control duplicate orders.	true / false

c. CPayOrderResult

Holds the status and message for a transaction returned by Citcon Pay service. An instance of the CPayOrderResult class can be inspected in the callback handler of the requestOrder method of CPaySDK.

Property summary

Name	Type	Description	Example
mRedirec-tUrl	String	Variable used in the WeChat sequence. Will only be popu-lated when paying by WeChat.	
mOrderId	String	ID of the Order from the pay-ment process.	123xyz
mOrder	CPayOrder	The actual Order object.	
mSignedStri ng	String	Variable returned by the Order initiation process. Used in the Alipay sequence.	
mOrderSpec	String	Variable returned by the Order initiation process. Used in the Alipay sequence.	
mMessage	String	Detailed description of the sta-tus of a transaction	Transaction succeeded
mStatus	String	The status code for the result	0
mCurrency	String	Order currency.	USD, CNY

d. CPayInquireResult

Holds the detailed status information for a transaction. An instance of the CPayInquireResult class can be inspected in the callback handler of the inquireOrder method of CPaySDK.

Property summary

Name	Type	Description	Example
mId	String	An unique identifier of the transaction. This ID is generated by Citcon Pay.	123456679
mReference	String	A reference identifying the transaction. This ID is generated by merchant	Abc123
mType	String	The type of the transaction: "charge" or "refund"	charge
mAmount	String	The total amount of the transaction in cents	225 ($2.45)
mCurrency	String	The name the of the currency in a three-letter code	USD or CAD or CNY
mTime	String	The timestamp for the transaction	2017/06/22 1:23:12 PM
mStatus	String	The status of the transaction	success
mNote	String	The note of the transaction.	

III.   Example

In this section, a demo merchant app making Alipay payments through Citcon Pay is demonstrated step-by-step using JAVA. The source code of this demo app is also provided as part of the SDK distribution package.

1. Create a new Android Studio Application project.

2. Copy the citcon-sdk.aar file in the /libs folder.

3. Modify the AndroidManifest.xml of your project, add the following for wechat pay:

    <application ...>

```xml
<activity-alias

    android:name=".wxapi.WXPayEntryActivity"

    android:exported="true"

    android:targetActivity="sdk.PaymentActivity" />

</application>


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

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

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

4. Modify the project-level build.gradle:

```
repositories {

        mavenCentral()

        flatDir {

        dirs 'libs'

        }}
```

5. Modify the app-level build.gradle:

```
dependencies {

compile(name:citcon-sdk', ext:'aar')

}
```

6. In your calling activity, add the following:

```java
  @Override
public void onResume()
{
        super.onResume();
        CPaySDK.getInstance(MainActivity.this, AUTH_TOKEN).onResume();
        CPaySDK.setMode(CPayMode.PROD);
}
```

The AUTH_TOKEN is the merchant token and needs to be populated accordingly.

7. Initiate a payment transaction after the lifecycle phase at step 5 has run:

```java
  CPayOrder order = new CPayOrder("1ZLLJULOCRW3LAU",
     "Test",
```

```java
                "This is a test transaction",
                "200",
                "USD",
                "alipay",
                "http://www.xyz.com/listen.php",
                "http://www.xyz.com/confirmation.php",
                true);
        CPaySDK.getInstance().requestOrder(order, new OrderResponse<CPayOrderResult>()
        {
            @Override
            public void gotOrderResult(final CPayOrderResult orderResult)
            {
                if(orderResult != null)
                {
                            // optionally call the Inquire API
                    CPaySDK.getInstance().inquireOrder(orderResult, new InquireResponse<CPayInquireRe-
sponse<CPayInquireResult>()
                    {
                        @Override
                        public void gotInquireResult(CPayInquireResult response)
                        {
                            if(response != null)
                            {
                                        //process the Inquire API response
                            }
                        }
                    });
                }
            }});
```