Citcon UPI SDK (Android)

Version 4.24.2

Version No.	Modify Activity	Modify Description	Editor	Modify Date
1.0.0	Creation	Braintree	Raymond Zhuang	Dec 2021
3.1.2	Update	UnionPay, Alipay and WeChat Pay	Raymond Zhuang	Apr 2022
4.22.07	Update	Integrated UPI & Online in one ; TOSS, FOMO	Raymond Zhuang	Aug 2022
4.22.09	Update	Support Alipay+ Pay	Andy	Oct 2022
4.23.00	Update	Add non-Citcon WeChat SDK broadcast callback	Andy	Jan 2023
4.24.00	Update	Add paypal/venmo and cashapp	Andy	July 2024
4.24.2	Update	Add CashApp Onfile	Andy	Dec 2024

I. Introduction

Citcon UPI SDK was designed for online merchant to integrate Citcon payment solutions effortlessly into their own app. By using the SDK, merchant developers can focus on businesss logic without having to understand the plumbing of payment transactions. This version of the SDK supports credit cards, PayPal , Venmo, AliPay, WeChatPay , UnionPay, TOSS and FOMO.

An exhibition of the payment transaction flow will help developers better understand the integration process at the conceptual level, as illustrated in Figure.1 blow.

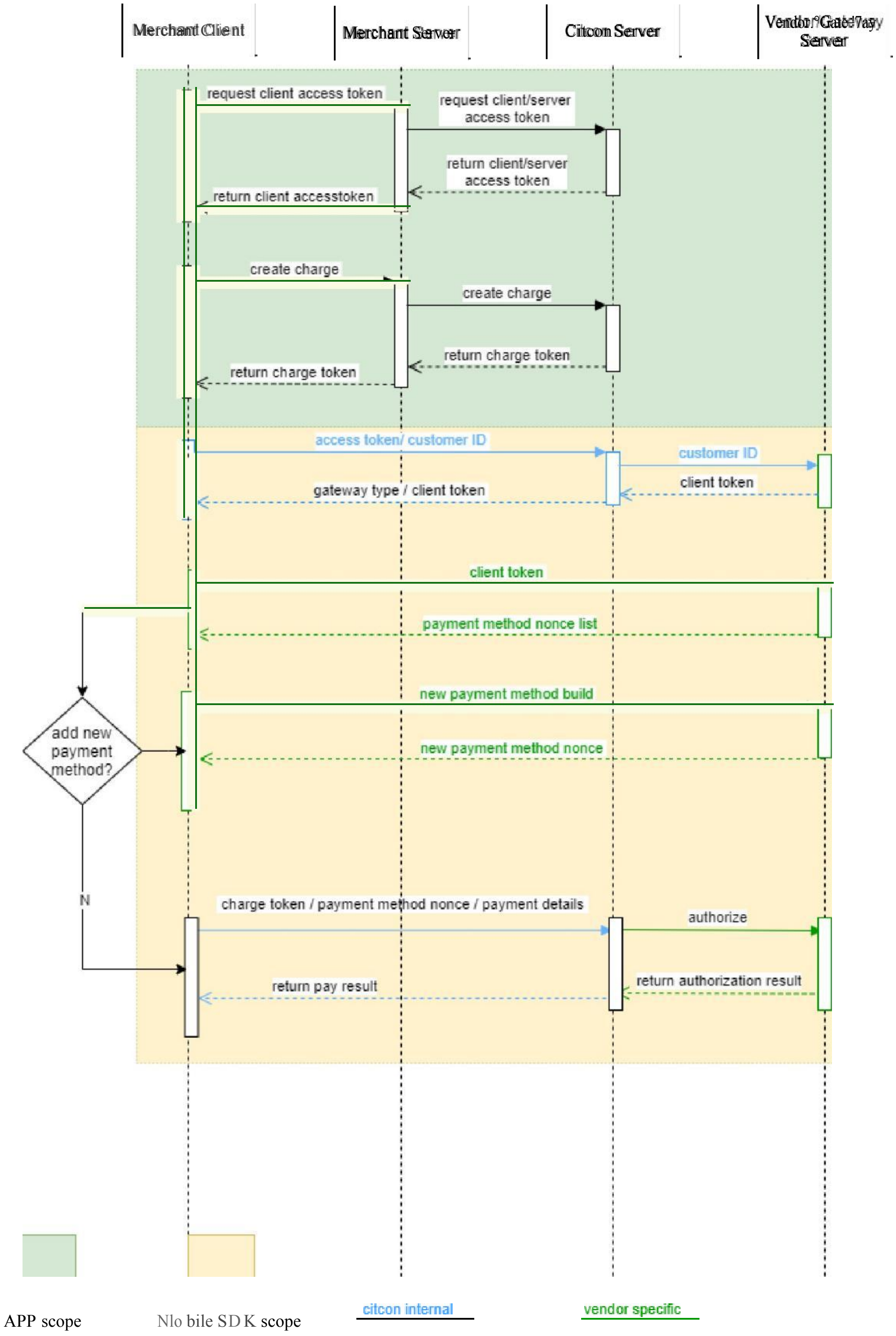

Merchant Client
Merchant Server
Citcon Server
Vendor/Gateway Server
request client access token
request client/server access token
return client/server access token
return client accesstoken
create charge
create charge
return charge token
return charge token
access token/ customer ID
customer ID
client token
gateway type / client token
client token
payment method nonce list
new payment method build
add new payment method?
new payment method nonce
N
charge token / payment method nonce / payment details
authorize
return authorization result
return pay result
APP scope
Nlo bile SD K scope
citcon internal
vendor specific

Fig.1 - Payment transaction flow from the merchant app to payment processor through Citcon

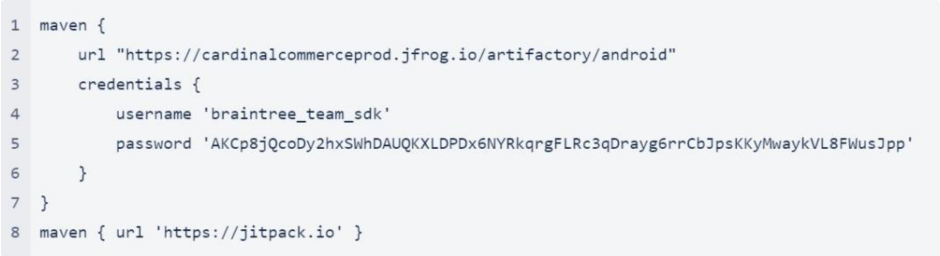
This version of the SDK targets Android 5.0 and above. It is distributed as an imported dependency.

II. Integration Steps

This section will show you step-by-step how to integrate Citcon UPI SDK. The source code of the demo has also been included in the distribution package.

!!!Attention: Do not use "_"(underscore) in project name. (for example: ~~demo_app~~).

1. Setup your project build.gradle at the end of repositories:



```
maven {
    url "https://cardinalcommerceprod.jfrog.io/artifactory/android"
    credentials {
        username 'braintree_team_sdk'
        password 'AKCp8jQcoDy2hxSWhDAUQKXLDPDx6NYRkqrgFLRc3qDrayg6rrCbJpsKKyMwaykVL8FWusJpp'
    }
}
maven { url 'https://jitpack.io' }
```

2. Add the dependency in your app build.gradle

Full module:

```
dependencies {
        implementation 'com.citcon.sdk:cpay:4.24.2'
        ...
}
```

Partial module:

```
dependencies {
        implementation('com.citcon.sdk:cpay:4.24.0') {
      // paypal module
      exclude group: "com.paypal.android", module: "paypal-web-payments"
      exclude group: "com.paypal.android", module: "fraud-protection"
      // braintree module
      exclude group: "com.braintreepayments.api", module: "braintree"
      // cashapp module
      exclude group: "app.cash.paykit", module: "core"
      ...
   }
}
```

Note: Currently, PayPal and Braintree cannot be integrated simultaneously. When you integrate Citcon SDK, you must to handle this using the 'exclude' command based on the specific situation.

3. **Get Mandatory Parameter from backend server**

   Generate Access Token and Charge Token on your server. The reference has to be get from your server as well. those 3 parameters are mandatory to be set up in next step.

4. Initialize SDK request

Braintree Java Example:

```
CPayRequest request = CPayRequest.PaymentBuilder.INSTANCE
                        .accessToken(accessToken)
                        .chargeToken(chargeToken)
                        .reference(reference)
                        .consumerID(consumerID)
                        .request3DSecureVerification(false)
                        .paymentMethod(CPayMethodType.PAYPAL)
                        .build(CPayENVMode.UAT);
```

Braintree Kotlin Example:

```
fun buildRequest(type: CitconPaymentMethodType): CPayRequest { return
   CPayRequest.PaymentBuilder
                .accessToken(mAccessToken.value!!)
                .chargeToken(mChargeToken.value!!)
                .reference(mReference)
                .consumerID(mConsumerID)
                .request3DSecureVerification(true)
                .consumer(demo3DSsetup()!!)
                .paymentMethod(CPayMethodType.UNKNOWN)//credit card
                .build(CPayENVMode.UAT)
}
```

UPI: AliPay/WeChatPay/UnionPay Java Example:

```
CPayRequest request = CPayRequest.UPIOrderBuilder.INSTANCE
        .accessToken(mAccessToken)
        .reference(mReference)
        .consumerID(consumerID)
        .currency(mCurrency)
        .amount(mAmount)
        .callbackURL(callbackURL)
        .ipnURL(IPNURL)
        .mobileURL("https://exampe.com/mobile")
        .cancelURL("https://exampe.com/cancel")
        .failURL("https://exampe.com/fail")
        .setAllowDuplicate(true)
        .paymentMethod(CPayMethodType.ALI)//AliPay
        .country(Locale.CANADA)
        .setExpiry(SystemcurrentTimeMillis() + mTimeout)
        .build(CPayENVMode.UAT);
```

UPI: AliPay/WeChatPay/UnionPay Kotlin Example:

```
fun buildDropInRequest(type: CitconPaymentMethodType): CPayRequest{
        return CPayRequest.UPIOrderBuilder
                    .accessToken(mAccessToken.value!!)
                    .reference(mReference)
                    .consumerID(mConsumerID)
                    .currency(mCurrency)
                    .amount(mAmount)
                    .callbackURL(mCallback)
                    .ipnURL("https://exampe.com/ipn")
                    .mobileURL("https://exampe.com/mobile")
                    .cancelURL("https://exampe.com/cancel")
                    .failURL("https://exampe.com/fail")
                    .setAllowDuplicate(true)
                    .paymentMethod(CPayMethodType.WECHAT)
                    .country(Locale.US)
                    .setExpiry(SystemcurrentTimeMillis() + mTimeout)
                    .build(CPayENVMode.UAT)
         }
```

ONLINE: AliPay/WeChatPay/UnionPay Java Example:

```
CPayRequest request = CPayRequest.CPayOrderBuilder.INSTANCE
        .token(mToken)
        .reference(mReference)
        .currency(mCurrency)
        .amount(mAmount)
        .paymentMethod(CPayMethodType.ALI)
        .setAllowDuplicate(true)
        .build(CPayENVMode.UAT);
```

UPI: PayPal/Venmo Minimum Example:

```
List<CPayGoodsData> goodsList = new ArrayList<CPayGoodsData>();
CPayGoodsData goodsData = new CPayGoodsData();
goodsData.setName("shoes");
goodsData.setQuantity(1);
goodsData.setUnitAmount(1);
goodsData.setProductType("physical");
goodsList.add(goodsData);
CPayGoodsData[] goods = goodsList.toArray(new CPayGoodsData[goodsList.size()]);
CPayShipping shipping = new CPayShipping();
shipping.setCity("CA");
shipping.setZip("95134");
shipping.setCountry("US");
CPayRequest.UPIOrderBuilder order = CPayRequest.UPIOrderBuilder.INSTANCE
        .accessToken(mAccessToken)
        .chargeToken(mChargeToken)
        .reference(mReference).amount("1")
        .country(Locale.US).currency("USD")
        .enableAutoCapture(true)
        .paymentMethod(CPayMethodType.PAYPAL)
        .ipnURL("https://www.merchant.com/ipn")
        .callbackURL("citcon://cpay.sdk")
        .cancelURL("citcon://cpay.sdk")
        .failURL("citcon://cpay.sdk")
        .goods(goods)
        .shipping(shipping)
        .build(CPayENVMode.UAT);
```

UPI: PayPal/Venmo Full Example:

```
// create goods
List<CPayGoodsData> goodsList = new ArrayList<CPayGoodsData>();
CPayGoodsData goodsData = new CPayGoodsData();
goodsData.setName("shoes");
goodsData.setSku("shoes");
goodsData.setUrl("https://www.ttshop.com");
goodsData.setQuantity(4);
goodsData.setUnitAmount(1);
goodsData.setUnitTaxAmount(1);
goodsData.setTotalDiscountAmount(1);
goodsData.setProductType("physical");
goodsList.add(goodsData);
CPayGoodsData[] goods = goodsList.toArray(new CPayGoodsData[goodsList.size()]);

// create shipping
CPayShipping shipping = new CPayShipping();
shipping.setFirstName("first");
shipping.setLastName("last");
shipping.setPhone("1-888-254-4887");
shipping.setEmail("test@citcon.cn");
shipping.setStreet("3 Main St");
shipping.setStreet2("");
shipping.setCity("CA");
shipping.setState("San Jose");
shipping.setZip("95134");
shipping.setCountry("US");
shipping.setType("shipping");
shipping.setAmount(1);

// create blling
CPayBillingAddr billingAddr = new CPayBillingAddr();
billingAddr.setCountry("US");
billingAddr.setCity("Columbus");
billingAddr.setState("OH");
billingAddr.setStreet("2425 Example Rd");
billingAddr.setStreet2("");
billingAddr.setZip("43221");

// create consumer
CPayConsumer consumer = CPayRequest.ConsumerBuilder.INSTANCE
.reference(mReference)
.firstName("first")
.lastName("last")
.email("test@citcon.cn")
.phone("+8615167186161")
.firstInteractionTime(1663311480)
.firstInteractionTime(1663312480)
.registrationIp("23.12.32.21")
.riskLevel("medium")
.totalTransactionCount(1)
.billingAddress(billingAddr)
.build();

// build order object
```

```
CPayRequest.UPIOrderBuilder order = CPayRequest.UPIOrderBuilder.INSTANCE
        .accessToken(mAccessToken)
        .chargeToken(mChargeToken)
        .reference(mReference)
        .currency("USD")
        .amount(8)
        .vertical("Household goods, shoes, clothing, tickets")
        .ipnURL("https://www.merchant.com/ipn")
        .callbackURL("citcon://cpay.sdk")
        .cancelURL("citcon://cpay.sdk")
        .failURL("citcon://cpay.sdk")
        .mobileURL("citcon://cpay.sdk")
        .note("test order")
        .paymentMethod(CPayMethodType.PAYPAL)
        .country(Locale.US)
        .goods(goods)
        .consumer(consumer)
        .billingAddr(billingAddr);
        .enableAutoCapture(true)
        .shipping(shipping)
        .build(CPayENVMode.UAT);
```

UPI: CASHAPP Example:

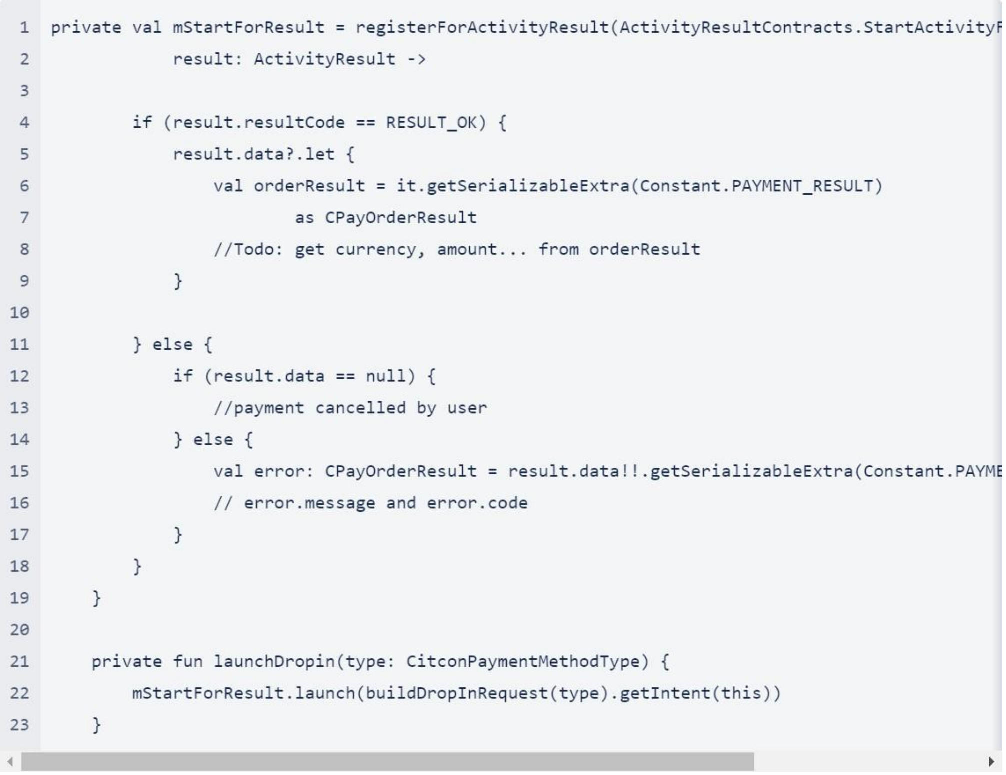
```
CPayRequest.UPIOrderBuilder order = CPayRequest.UPIOrderBuilder.INSTANCE
        .accessToken(mAccessToken)
        .chargeToken(mChargeToken)
        .reference(mReference)
        .country(Locale.US)
        .currency("USD")
        .amount("8")
        .enableAutoCapture(true)
          .requestToken(false)
        .paymentMethod(CPayMethodType.CASHAPP)
        .ipnURL("https://www.merchant.com/ipn")
        .callbackURL("citcon://cpay.sdk")
        .cancelURL("citcon://cpay.sdk")
        .failURL("citcon://cpay.sdk")
        .build(CPayENVMode.UAT);
```

5. Start Citcon Payment UI*

Parameter:
CPayRequest.getIntent()
Java Example:

```
startActivityForResult(request.getIntent(this), DROP_IN_REQUEST);
```

* onActivityResult has been deprecated recently, new implemented method is in this Kotlin example:



```
private val mStartForResult = registerForActivityResult(ActivityResultContracts.StartActivityF
            result: ActivityResult ->

        if (result.resultCode == RESULT_OK) {
            result.data?.let {
                val orderResult = it.getSerializableExtra(Constant.PAYMENT_RESULT)
                        as CPayOrderResult
                //Todo: get currency, amount... from orderResult
            }

        } else {
            if (result.data == null) {
                //payment cancelled by user
            } else {
                val error: CPayOrderResult = result.data!!.getSerializableExtra(Constant.PAYME
                // error.message and error.code
            }
        }
    }

    private fun launchDropin(type: CitconPaymentMethodType) {
        mStartForResult.launch(buildDropInRequest(type).getIntent(this))
    }
```

6. Get Payment Result

Java Example:

```
public void onActivityResult(int requestCode, int resultCode, Intent data)
   { super.onActivityResult(requestCode, resultCode, data);

   if (resultCode == RESULT_OK) {
      CPayResult order = (CPayResult) data.getSerializableExtra(Constant.PAYMENT_RESULT);
   } else {
      String message;
      if(data != null)
      {
         CPayResult error = (CPayResult) data.getSerializableExtra(Constant.PAYMENT_RESULT);
      } else {
         //payment cancelled by user";
      }
   }
}
```

** onActivityResult has been deprecated recently, new implemented method is in this Kotlin example

```
private val mStartForResult = registerForActivityResult(ActivityResultContracts.StartActivityForResult())
         { result: ActivityResult ->

      if (result.resultCode == RESULT_OK)
         { result.data?.let {
            val orderResult =
                  it.getSerializableExtra(Constant.PAYMENT_RESULT) as
                  CPayResult
            //Todo: get currency, amount... from orderResult
         }

      } else {
         if (result.data == null) {
            //payment cancelled by user
         } else {
            val error: CPayResult =
result.data!!.getSerializableExtra(Constant.PAYMENT_RESULT) as CPayResult
            // error.message and error.code
         }
      }
   }
```

```
private fun launchDropin(type: CitconPaymentMethodType)
   { mStartForResult.launch(buildDropInRequest(type).getIntent(this))
}
```

7. Using 3DS

In order to use 3DS2, you will need to create a CPayConsumer object with relevant customer and transaction data in order to minimize the need for issuing banks to present authentication challenges to customers

Java Example:

```
private CPayConsumer demo3DSsetup() {

        CPayBillingAddr billingAddr = CPayRequest.BillingAdressBuilder.INSTANCE

       .city("Chicago")

        .state("IL")

        .street("555 Smith St")

        .postCode("12345")

        .country("US")

        .build();


        return CPayRequest.ConsumerBuilder.INSTANCE

        .firstName("Alex")

        .lastName("Smith")

        .email("google@gmal.com")

        .phone("1112223344")

        .billingAddress(billingAddr)
```

```
            .build();

            }
```

8. No-Citcon Wechat SDK Broadcast

Sometimes our integrators have already integrated WeChatpay SDK, and they want to keep their own WeChatpay module. In this case, there will be a conflict in the callback class registration of WeChat payment.

a. define a simple broadcast class

```
package com.citconpay.cupay;

import android.annotation.SuppressLint;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.widget.Toast;

import com.tencent.mm.opensdk.constants.ConstantsAPI;
import com.tencent.mm.opensdk.modelpay.PayResp;

public class WXHandlerBroadcastReceiver extends BroadcastReceiver {
    private static final String TAG = "WXHandlerBroadcastReceiver";

    @SuppressLint("LongLogTag")
    @Override
    public void onReceive(Context context, Intent intent) {
        // onReceive: The function cannot do time-consuming things, reference value: within 10s

        //jump to activity
        //Intent i = new Intent(context, MainActivity.class);
        //i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        //context.startActivity(i);

        //get content
        String stage = intent.getStringExtra("stage");
        Bundle object = intent.getBundleExtra("object");

        Log.d(TAG, "---> onReceive: "+ stage +",object: "+ object);

        if ("onCreate".equals(stage)) {
            Bundle savedInstanceState = object;
            if (savedInstanceState != null) {

            }
        } else if ("onNewIntent".equals(stage)) {
            Intent newIntent = object.getParcelable("intent");
            if (newIntent != null) {

            }

        } else if ("onReq".equals(stage)) {
            if (object != null) {

            }

        } else if ("onResp".equals(stage)) {
            if (object != null) {
                int type = object.getInt("_wxapi_command_type");
                if (type == ConstantsAPI.COMMAND_PAY_BY_WX) {
                    PayResp resp = new PayResp();
                    resp.fromBundle(object);
                    if (resp.prepayId != null) {

                    }
                }
            }

        }

        Toast.makeText(context, "Broadcast Receiver: "+stage, Toast.LENGTH_SHORT).show();
    }
}
```

b. register broadcast

```
IntentFilter filter = new IntentFilter();
filter.setPriority(1000);
filter.addAction(Constant.CUPAT_WXHANDLE_BROADCAST_ACTION);
registerReceiver(new WXHandlerBroadcastReceiver(), filter);
```

or in **application** element of **AndroidManifest.xml**

```
<receiver android:name=".WXHandlerBroadcastReceiver" android:exported="true">
    <intent-filter>
        <action android:name="citcon.cupay.wxhandle" />
    </intent-filter>
</receiver>
```

9. Using a button UI in your App

PayPal Button

a. Add the payment-buttons package dependency in your App’s build.gradle file:

```
dependencies {
  implementation "com.paypal.android:payment-buttons:1.4.0"
}
```

b. Add PayPalButton to your App’s project layout XML:

```
<com.paypal.android.paymentbuttons.PayPalButton
  android:id="@+id/paypal_button"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"/>
```

c. Handle the button event:

```
val payPalButton = findViewById<PayPalButton>(R.id.paypal_button)
 payPalButton.setOnClickListener {
   // start PayPal
 }
```

CashApp Pay Button

a. Add the payment-buttons package dependency in your App’s build.gradle file:

```
dependencies {
  implementation 'app.cash.paykit:core:2.4.0'
}
```

b. The light and dark variants of the CashAppPayButton can be used using the style parameter to your App’s project layout XML, as follows:

```
// Light style
<app.cash.paykit.core.ui.CashAppPayButton
      style="@style/CAPButtonStyle.Light"
      android:id="@+id/cashapp_button"
      android:layout_height="54dp"
      android:layout_width="match_parent"
      />

// Dark style
<app.cash.paykit.core.ui.CashAppPayButton
      style="@style/CAPButtonStyle.Dark"
      android:id="@+id/cashapp_button"
      android:layout_height="54dp"
      android:layout_width="match_parent"
      />
```

c. Handle the button event:

```
val button = findViewById<PayPalButton>(R.id.cashapp_button)
button.setOnClickListener {
  // start CashApp
}
```

III. The Citcon UPI SDK

This section details the main components of the Citcon UPI SDK for Android development.

The commonly used header files and their purposes will be listed here, and a step-by-step example of integrating the SDK in a demo merchant app will be shown in the next section.

Class references

a. CPayDropInRequest:

The **CPayRequest** class set up most parameters to start up Citcon Payment UI.

Class Methods:

- accessToken (Mandatory)
  accessToken(String accessToken)
  - Parameters:
    - accessToken - Which gets from the server is used to identify merchants.
- chargeToken (Mandatory)
  chargeToken(String chargeToken)
  - Parameters:
    - accessToken - Which gets from the server is used to identify transactions.
- consumerID (Mandatory)
  accessToken(String consumerID)
  - Parameters:
    - consumerID - This ID is used to identify users on vaulting payment methods.
- reference (Mandatory)
  reference(String reference)
  - Parameters:
    - reference - This is the unique transaction reference id.

- paymentMethod (Mandatory)
  paymentMethod(**CitconPaymentMethodType** type)
  - Parameters:
    - type - this type is used to bring up different payment method UI.
- collectDeviceData (Optional)
  collectDeviceData(boolean collectDeviceData)
  - Parameters:
    - collectDeviceData - true if Drop-in should collect and return device data for fraud prevention.

- paypalRequest (Optional)
  paypalRequest(PayPalRequest request)
  - Parameters:
    - request - The PayPal Request PayPalRequest for the transaction. If no amount is set, PayPal will default to the billing agreement (Vault) flow. If the amount is set, PayPal will follow the one time payment (Checkout) flow.

- disablePayPal **(Optional)**
  disablePayPal()
  - Disables PayPal in Drop-in.

- disableVenmo **(Optional)**
  disableVenmo()
  - Disables Venmo in Drop-in.

- disableCard **(Optional)**
  disableCard()
  - Disables Card in Drop-in.

- request3DSecureVerification **(Optional)**
  request3DSecureVerification(boolean requestThreeDSecure)

  If 3D Secure has been enabled in the control panel and an amount is specified in amount(String) or a ThreeDSecureRequest is provided, Drop-In will request a 3D Secure verification for any new cards added by the user.
  - Parameters:
    - requestThreeDSecure - true to request a 3D Secure verification as part of Drop-In, false to not request a 3D Secure verification. Defaults to false.

- maskCardNumber **(Optional)**
  maskCardNumber(boolean maskCardNumber)

  - Parameters:
    - maskCardNumber - true to mask the card number when the field is not focused. See CardEditText for more details. Defaults to false.

- maskSecurityCode **(Optional)**

  maskSecurityCode(boolean maskSecurityCode)

  - Parameters:
    - maskSecurityCode - true to mask the security code during input. Defaults to false.

- getIntent **(Optional)**

  public Intent getIntent(android.content.Context context)

  - Get an Intent that can be used in FragmentActivity.startActivityForResult(Intent, int) to launch DropInActivity and the Drop-in UI.
  - Parameters:
    - context -
      Returns:
      Intent containing all of the options set in CPayDropInRequest.

- isPayPalEnabled **(Optional)**

  public boolean isPayPalEnabled()

- getPayPalRequest **(Optional)**

  public com.braintreepayments.api.models.PayPalRequest getPayPalRequest()

- isVenmoEnabled **(Optional)**

  public boolean isVenmoEnabled()

- isCardEnabled **(Optional)**

  public boolean isCardEnabled()

- getGooglePaymentRequest **(Optional)**

  public com.braintreepayments.api.models.GooglePaymentRequest getGooglePaymentRequest()

- getCardholderNameStatus **(Optional)**

  public int getCardholderNameStatus()

- isSaveCardCheckBoxShown **(Optional)**

  public boolean isSaveCardCheckBoxShown()

b. CPayMethodType: enum

Type definition of payment method.

Name	Description
PAYPAL	Paypal
CREDIT	Credit card. (Eg. American Express, Discover, Visa, Mastercard and so on)
PAY_WITH_VENMO	Venmo
ALI	AliPay
WECHAT	WeChat Pay
UNIONPAY	Union Pay
ALI_HK	AliPay HongKong
KAKAO	
SAMSUNG	
LG	
LPAY	
TOSS	
BANKTRANSFER	
ATOME	
PAYNOW	
NETSPAY	

GRABPAY	
SHOPEEPAY	
DANA	
GCASH	
RABBIT LINE PAY	
TNG	
BPI	
TRUE MONEY	
ALIPAY+	
CASHAPP	CASHAPP PAY

c. CPayBillingAddr:

Name	Type	Required-/ Optiona- l/Conditi- onal	Description	Example
Street	String?	O	Street name	
Street2	String?	O	Street name	
City	String?	O		

State	String?	O		
Zip	String?	O	Post code	
Country	String?	O		

d. CPayConsumer:

Name	Type	Required-/ Optiona- l/Conditi- onal	Description	Example
reference	String?	O	Transaction id	
firstName	String?	O		
lastName	String?	O		
phone	String?	O		
email	String?	O		
billingAddress	CPayBillingAddr	O		

e. CPayResult: (Kotlin data class)

Holds the result and message for a transaction returned by Citcon Pay service.

Property summary

Name	Type	Required-/ Optiona-l/ Conditi-onal	Description	Example
resultCode	Int	R	resultCode in onActivityResult	RESULT_OK
message	String	C	Detailed description of the status of a transaction. The message maybe null when result is "success".	bad request
result	String	R	The result of the transaction	success
code**	String	R	The status code for the result	0
transactionId	String	C	An unique identifier of the transaction. This ID is generated by Citcon Pay. It will be null when result is fail.	123xyz
amount	String	C	The total amount of an transaction. It will be null when result is fail.	10
currency	String	C	Currency type defined by three-letter code.	USD
reference	String	C	A reference identifying the transaction. This ID is generated by merchant.	456wyz

time	String	C	The timestamp for the transaction.	12321321312
paymentMethod	CitconPaymentMehodType	R	Payment method type of this order	PAYPAL
country	String	C	Two letter country code	US

**Status code

Code	Description
0	Success
-1	Fail. For details, please refer to the message.
-2	Cancel
-3	Load configuration fail
4000	Bad request
4010	Unauthorized
4100	Duplicate request
4101	Unknown error
4102	Invalid header
4103	Not found
5100	Internal server error
others	Other payment error