Payrix Mobile

Software Development Kit (SDK)

Android Developer Guide

Version 1.0.1

Revision Date: 07.08.2020

PAYRIX

TABLE OF CONTENTS

Overview

The Payrix Mobile SDK for Android consists of two Android modules, PayCard and PayCore, that allow a developer to access credit cards readers and perform payment processing requests, respectively. Using this SDK one can create a full mobile App that handles card swipes and the process those transactions against a designated payment gateway.

The Payrix Android SDK is distributed using Android Archive (.aar) files. This document fully details the process to follow to implement the SDK into your App.

Requirements

The Payrix Android SDK was designed and developed to leverage the following requirements:

- Android 9 (Pie) – API 28 or later. Please note that this requirement is driven by Google. As of August 2019, Google will not allow App updates that use a version below API 28.
- Android Studio version 3.5 or later.
- Bluetooth and Audio Card Readers (or Manual Card Entry)
- Network accessibility (WIFI or Cellular)

Installing the SDK (Step-by-Step)

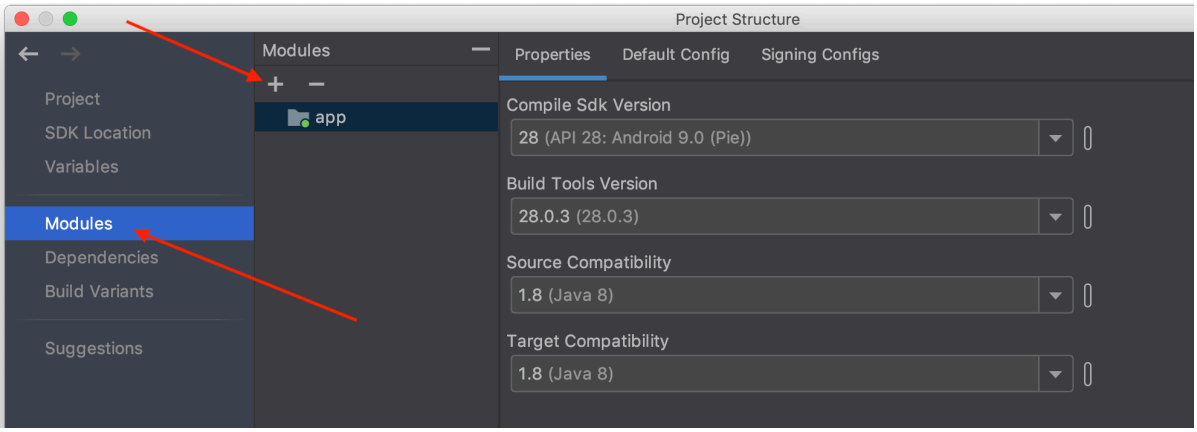
1. Download the Payrix Android SDK using the following link:
   https://gitlab.com/payrix/public/payrix_sdk_android

   The link will take you to a Git structured repository where you will be able to download a copy of the repository files. There are 3 items: ReadME file that provides a brief description, this Developer's Guide, and the folder containing the SDK libraries.

   The SDK folder named **PayrixAndroidSDK** contains 2 files:
   - paycard.aar
   - paycore.aar

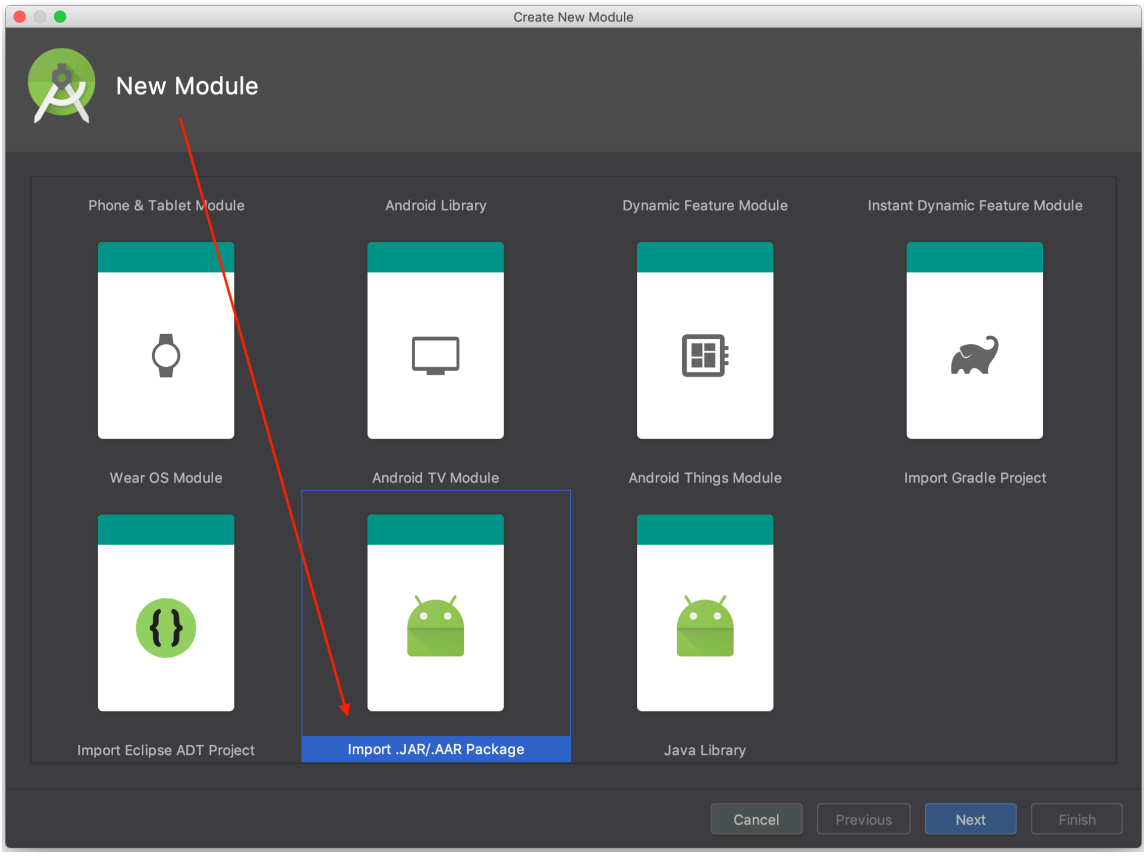
   These 2 libraries provide card reader and Payrix transaction processing features respectively.

2. Open your App in Android Studio, and click File, then Project Structure. The following screen will be displayed:

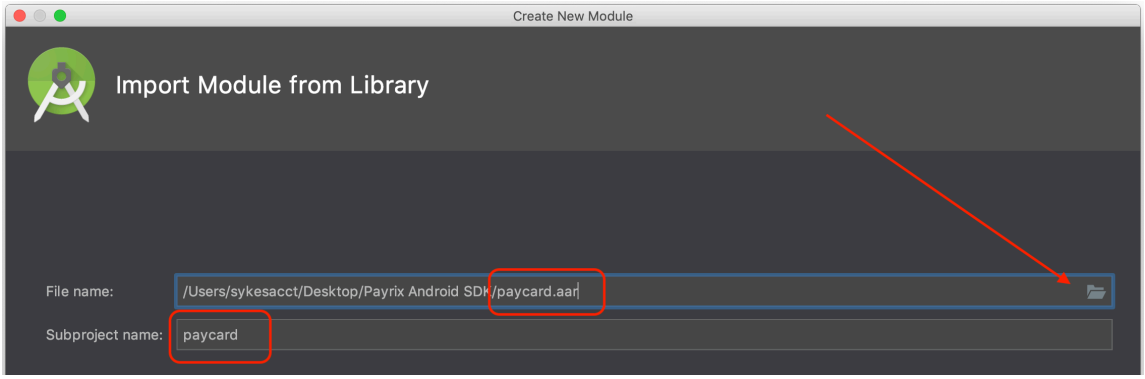


   - Select Modules in left pane.
   - Then in Modules pane click "+" to add new modules.

3. The next screen is the New Module assistant. Here you scroll down and select **Import .JAR / .AAR Package**. Then click Next.

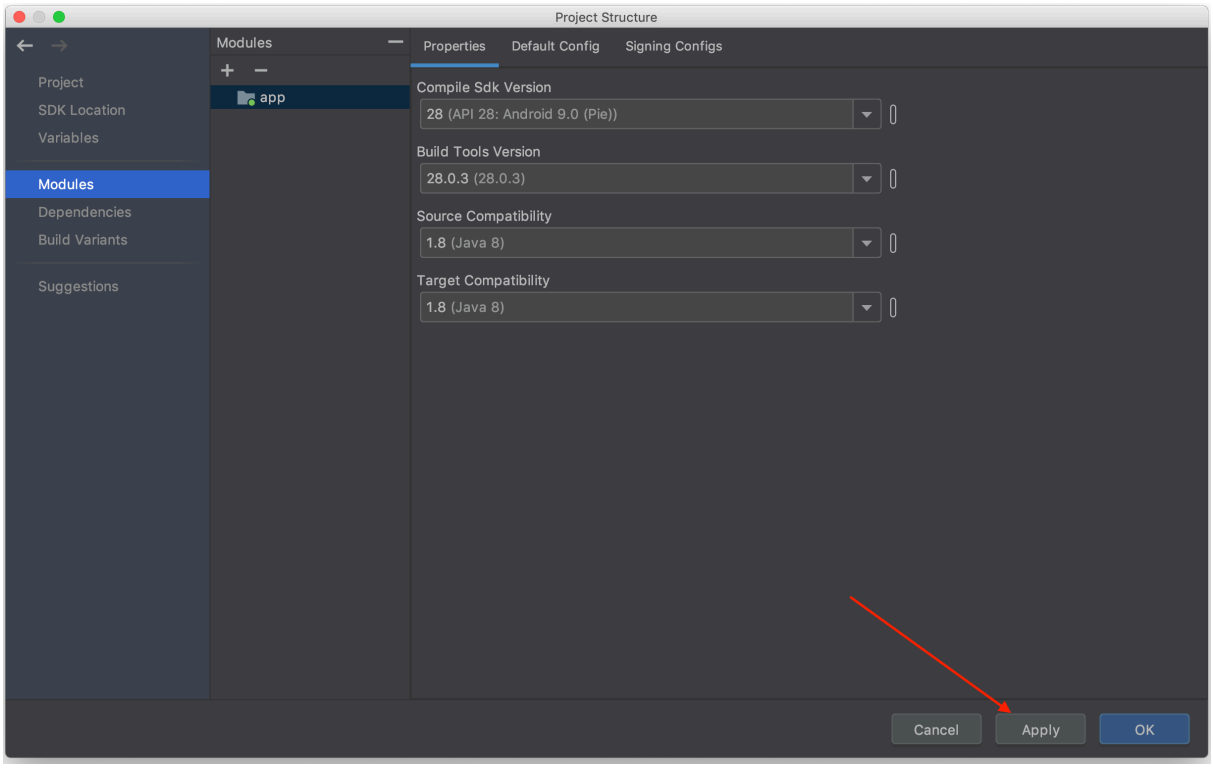


4. Next locate the .AAR file for PayCard which is in the folder you downloaded at the start of this process, by clicking the folder icon. Select the paycard.aar file. The subproject name will automatically default to paycard.

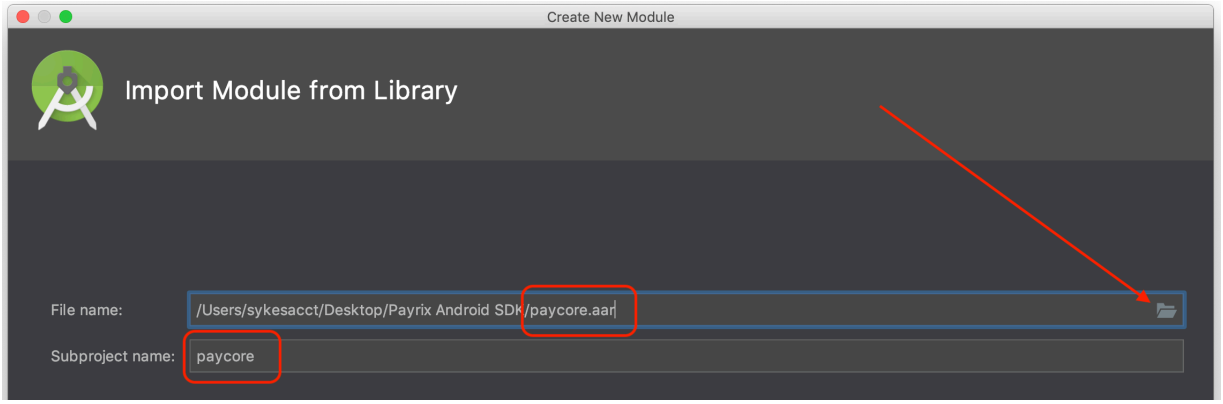


Click the Finish button at the bottom of the screen.

5. To complete the import of the PayCard module, click Apply as shown on the following screen.



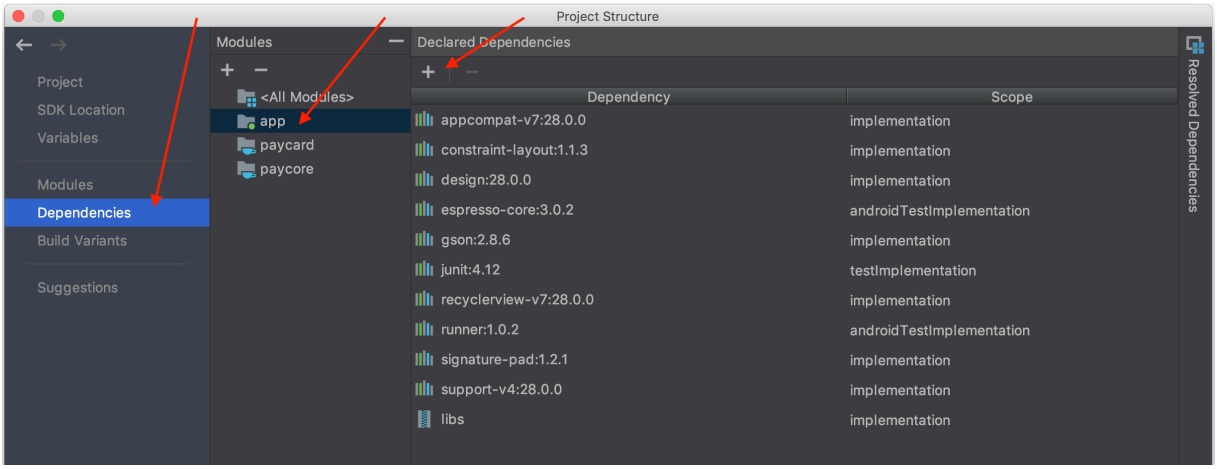
6. Now Repeat Step 2 through 4, but except on Step 4 select paycore.aar instead as show here:



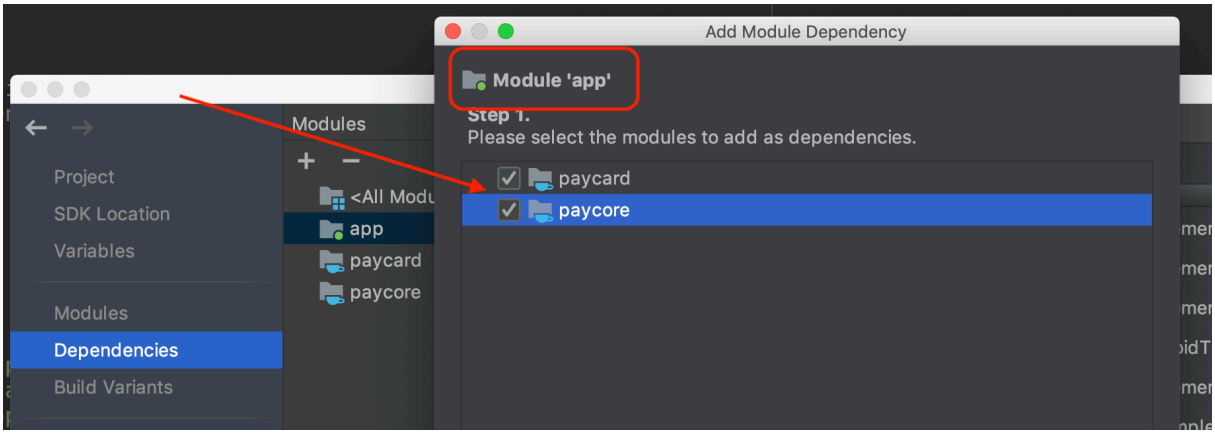
Click the Finish button at the bottom of the screen.

7. Reviewing the following screen, you will see that you have successfully added both the paycard and paycore modules to your app project.

8. Next we need to make the SDK classes and methods available to the project by establishing the dependencies. Select **Dependencies** in left pane, and **app** in center Modules pane. Then click "+" to add new dependencies to your project app, as shown on the following screen.

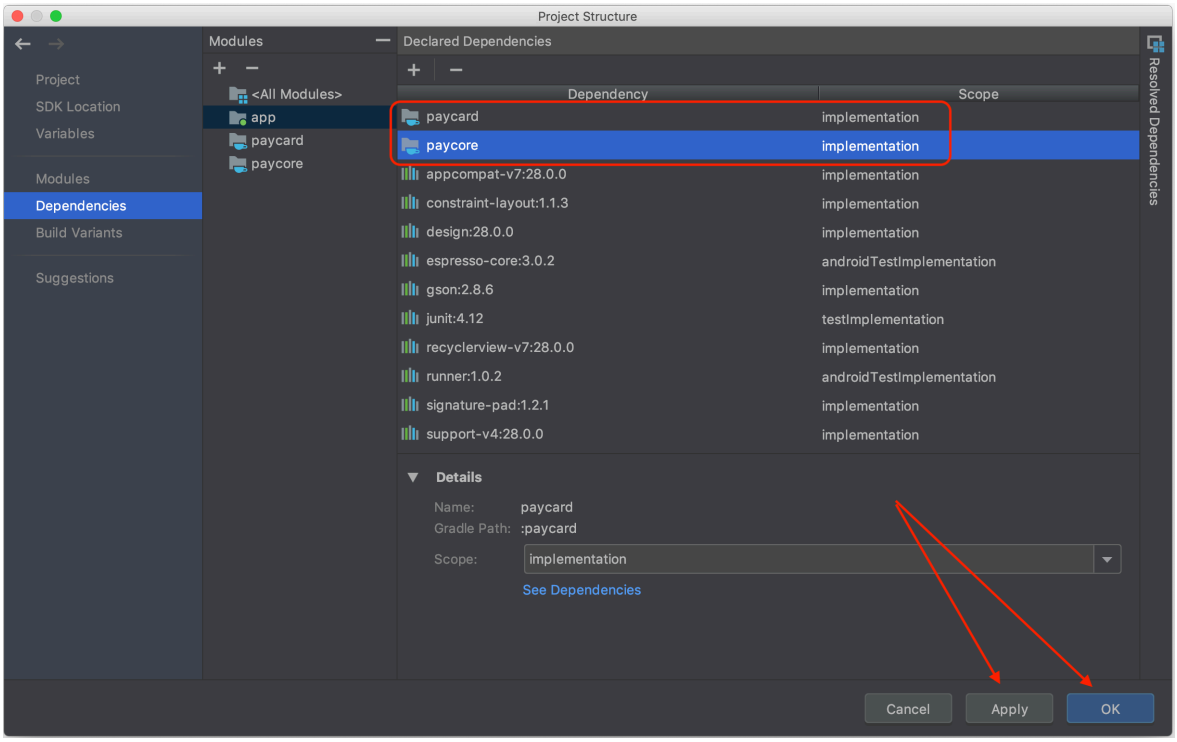


The following "Add Module Dependency" popup selection will appear.

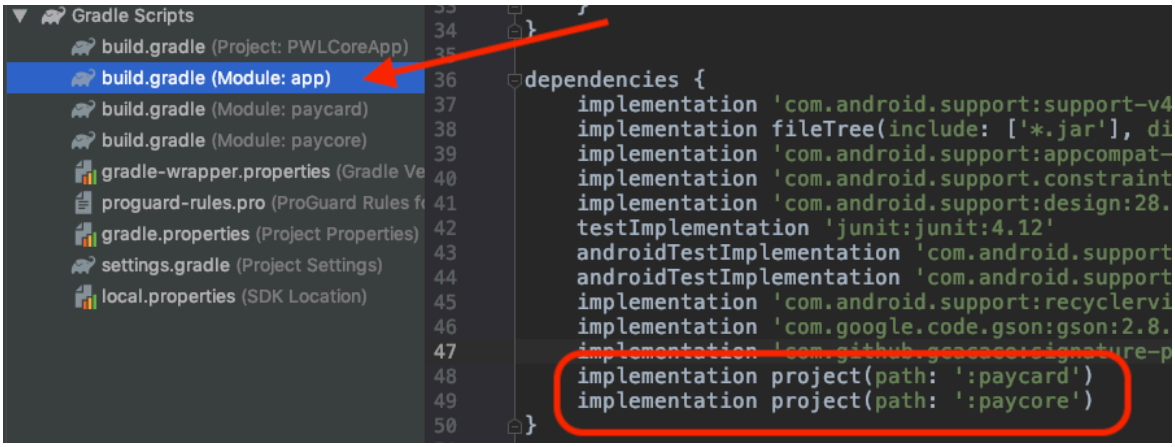


Select both **paycard** and **paycore** and click OK at the bottom of the screen.

9. When you are returned to the following screen you will notice 2 new dependencies (paycard, paycore), so now just click Apply (not always required) and OK to complete the SDK installation.



10. As a final verification, you can open the project's app build gradle file, and in the dependencies section you will see 2 new implementation statements, as shown in the following screenshot.



This completes the SDK installation process.

Using the Payrix Mobile Android SDK

There are two primary modules that make up the Payrix Mobile Android SDK. The first is PayCard which handles card reader management and communications, and the second is PayCore which handles the payment transaction processing with the appropriate gateway.

General App Setup Information

Android Manifest Recommended Permissions

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.USE_BIOMETRIC" />
<uses-permission android:name="android.permission.NFC" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

build.gradle (app) Recommendations

```
android {
    compileSdkVersion 28
    buildToolsVersion '28.0.3'
    defaultConfig {
        applicationId "com.payrix.pwlcoreapp"
        minSdkVersion 28
        targetSdkVersion 28
        versionCode 48
        versionName "Version 1.1.0 (Build 1912.04)"
        ndk.abiFilters 'armeabi-v7a', 'arm64-v8a', 'x86', 'x86_64'
```

Google Store New Minimum

Needed for bbPOS Readers

PayCard Services

Class	Method	Purpose	Delegate / Callback
PayCardRDRMgr	getInstance	Access to PayCardRDRMgr singleton class	delegate: Set to the class that will handle method specific callbacks.
	startPayCardRDRMgr	Start Reader Services.	N/A
	scanForReaders	Scans for available Bluetooth readers.	didFindRDRDevices didReceiveBTScanTimeOut
	connectBTReader	Connects a Bluetooth device by Device ID.	didSuccessfulBTConnect didReceiveCardReaderConnectionFailed
	connectAudioReader	Connects to an Audio Card Reader that is plugged into the audio port.	didReceiveAudioConnectedNotice didReceiveCardReaderConnectionFailed
	disconnectAudioReader	Disconnects an Audio Card Reader from transaction processing capability (even if still plugged in)	didReceiveAudioDisconnectedNotice
	detectConnectionType	Detects and returns the type of reader connection: • Audio • Bluetooth • USB (not Supported) • None	Returned value type enum: PayCardConnectionMode
	isAudioreaderConnected	Check if Audio Reader is attached and has card transaction processing capabilities active.	Returned value type Boolean: True = Reader Connected False = Reader Not Connected
	isBTReaderConnected	Check if a Bluetooth reader is connected.	Returned value type Boolean: True = Reader Connected False = Reader Not Connected
	disconnectBTReader	Disconnect the Bluetooth reader device from mobile device and transaction processing capability.	didReceiveBTDisconnect
	getDeviceData	Retrieves device specific information about card reader in use.	didReceiveDeviceInfo **Note:** Not all device services offer access to this information and the information may vary.
	stopScan	Stop scanning for Bluetooth readers.	N/A
	General Error Handling For unexpected issues	General Error Handling None specific errors are trapped and returned in this callback.	didReceiveCardReaderError

Class	Method	Purpose	Delegate / Callback
PayCardMaster	getInstance	Access to PayCardMaster singleton class	delegate: Set to the class that will handle method specific callbacks.
	startPayCardMaster	Initial setup of card reader device drivers.	N/A
	doReadCard	Stage 1: Request PayCard initiate the card reading process	didReceiveReaderModeUpdate: • Returns an enum value found in PayCardDeviceMode that tells app how to proceed. Example "Swipe Card" didReceiveCardReaderIssue
		Stage 2: Card is Swiped (or other action taken)	didReceiveSwipeSuccess didReceiveReaderModeUpdate
		Stage 3: (EMV) Do Card Transaction Confirmation	didReceiveReaderModeUpdate: • DeviceMode_Confirm • See doCardConfirm method
		Stage 4: (EMV) Complete transaction on Gateway to complete EMV transaction request.	requestForHostEMVProcess: • See responseFromHostEMVProcess method
	doDetermineCardType	A utility to take the card number and return the card brand such as Visa or MasterCard.	Returned value type String: Supported Card Brand Name.
	doCardConfirm (EMV)	Performs an EMV transaction confirmation.	requestForHostEMVProcess
	responseFromHostEMVProcess	Sends final EMV Tag (8A, 91, 71, 72) data to reader to complete transaction.	didCompleteEMVCardTransaction didReceiveFinalEMVBatchData
	- General Utility Callback	Message Capture. Helpful during development and debugging.	didReceiveMessageToDisplay

PayCore Services

Class	Method	Purpose	Delegate / Callback
PayCoreMaster	getInstance	Access to PayCoreMaster singleton class	delegate: Set to the class that will handle method specific callbacks.
	validateLoginCredentials	Login Authentication	didReceiveLoginResponse
	verifySessionKey	Verifies the session is still valid	Response in completion handler
	getMerchantId	Retrieves the Merchant ID of signed in Merchant	didReceiveMerchantIDInfo
	doCardReaderTransaction	Initiate payment transaction with Host-API gateway processing with Card Reader input.	didReceiveTransactionResponse
	doManualCardTransaction	Initiate payment transaction with Host-API gateway processing with Manual Entry input.	didReceiveTransactionResponse
	addSignatureToTransaction	Allows the adding a signature to the active payment transaction	N/A
	retrieveTransactionsFor	Retrieves transactions by merchant.	didReceiveTransactionResponse
	processRefund	Request a refund on original Sale transaction.	didReceiveRefundResponse
	checkTXNRefundEligible	Determines if a Sale transaction is refund eligible.	didReceiveRefundEligibleStatus
	doCheckNetworkConnection	This method checks the network availability and responds with a Boolean	Returned value type Boolean: True = Connection Available False = Connection Not Available

Feedback and Support

To provide feedback on this document and any of the specifications contained within, contact us at: support@payrix.com

Document Revision History

Revision Date	Author	Revision Description
12.04.2019	Mobile Design & Engineering Steven Sykes	Original Document
06.30.2020	Mobile Design & Engineering Steven Sykes	Revised to refer to GitLab Repository link for client download of SDK.
07.08.2020	Mobile Design & Engineering Steven Sykes	Added Android Manifest and Gradle setup information