



Apple Pay at Global Payments Asia-Pacific

Revision History

Version	Date	Author	Notes
1.0	3-AUG-2016	Match Cheng	Initial Document
1.1	19-AUG-2016	Match Cheng	Added Report / Transaction Query
1.2	26-SEP-2016	Match Cheng	Added Apple Pay JS (Apple Pay on the Web) References
1.3	11-OCT-2016	Match Cheng	Section 5 – F.A.Q. Added
1.4	09-NOV-2016	Match Cheng	Revised section 3

Table of Contents

- 1.0 INTRODUCTION OF APPLE PAY AT GLOBAL PAYMENTS..... 3**
 - 1.1 INTRODUCTION3
 - 1.2 ABOUT CYBERSOURCE.....3
 - 2.1 PRE-READ MATERIALS.....4
 - 2.2 PRE-REQUISITES5

- 2.0 CONFIGURE GLOBAL PAYMENTS APPLE PAY ENVIRONMENT..... 6**
 - 2.1 MERCHANT – REGISTER AN APPLE MERCHANT ID.6
 - 2.2 GLOBAL PAYMENTS – GENERATE THE CSR6
 - 2.3 MERCHANT – CONFIGURE THE CERTIFICATE FOR YOUR MERCHANT ID6
 - 2.4 MERCHANT – CREATE THE TRANSACTION SECURITY KEYS.....7

- 3.0 INTEGRATION & TESTING..... 8**
 - 3.1 IOS – CREATE THE PAYMENT REQUESTS.....8
 - 3.1 SERVER SIDE CODE.....8
 - 3.3 TESTING 10
 - 3.4 REFUND 10
 - 3.4 SPECIAL NOTE – SHIPPING / BILLING ADDRESS..... 10
 - 3.6 DISABLE OF BILLING ADDRESS..... 10
 - 3.6 SPECIAL NOTE – POS_ENVIRONMENT..... 10

- 4.0 MISCELLANEOUS..... 13**
 - 4.1 REPORT DOWNLOAD 13

- 5.0 F.A.Q..... 14**
 - 5.1 HOW CAN I CARRY OUT REFUND WITH APPLE PAY? 14
 - 5.2 CAN I OBTAIN THE FULL D-PAN? 14

1.0 Introduction of Apple Pay at Global Payments

1.1 Introduction

Global Payments Inc. (NYSE: GPN), a leading worldwide provider of payment technology services, is pleased to offer merchants in Hong Kong and Singapore the ability to accept credit and debit card payments via Apple Pay™, which is transforming mobile payments with an easy, secure and private way to pay that's fast and convenient.

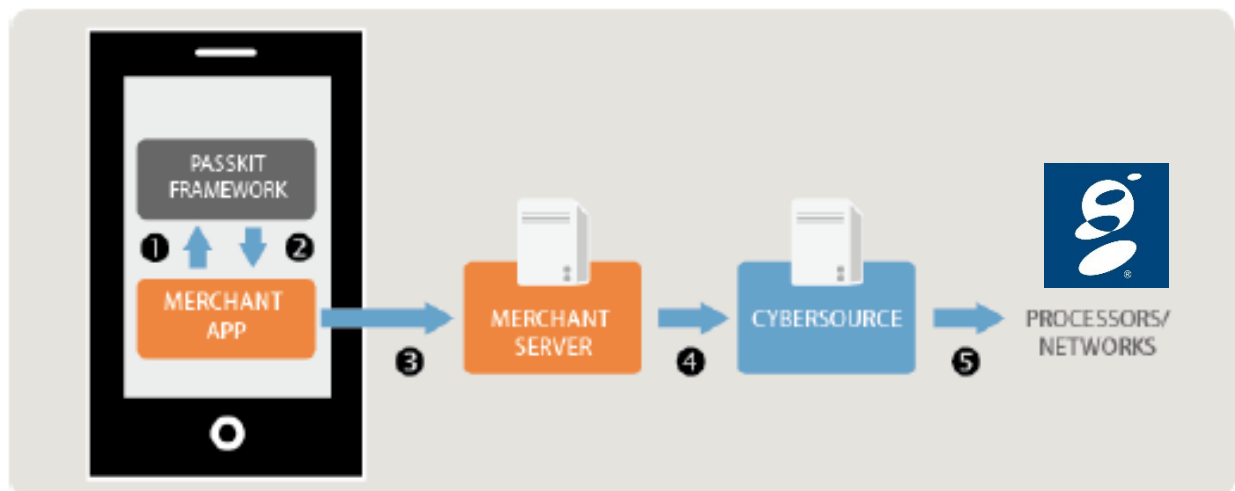
Security and privacy is at the core of Apple Pay. When you use a credit or debit card with Apple Pay, the actual card numbers are not stored on the device, nor on Apple servers. Instead, a unique Device Account Number is assigned, encrypted and securely stored in the Secure Element on your device. Each transaction is authorized with a one-time unique dynamic security code.

Online shopping in apps accepting Apple Pay is as simple as the touch of a finger with Touch ID, so there's no need to manually fill out lengthy account forms or repeatedly type in shipping and billing information. When paying for goods and services within apps, Apple Pay is compatible with iPhone SE, iPhone 6s, iPhone 6s Plus, iPhone 6, iPhone 6 Plus, iPad Air 2, iPad mini 3, iPad mini 4 and iPad Pro.

This guide will provide additional information on integration to ensure our merchants could complete the integration timely.

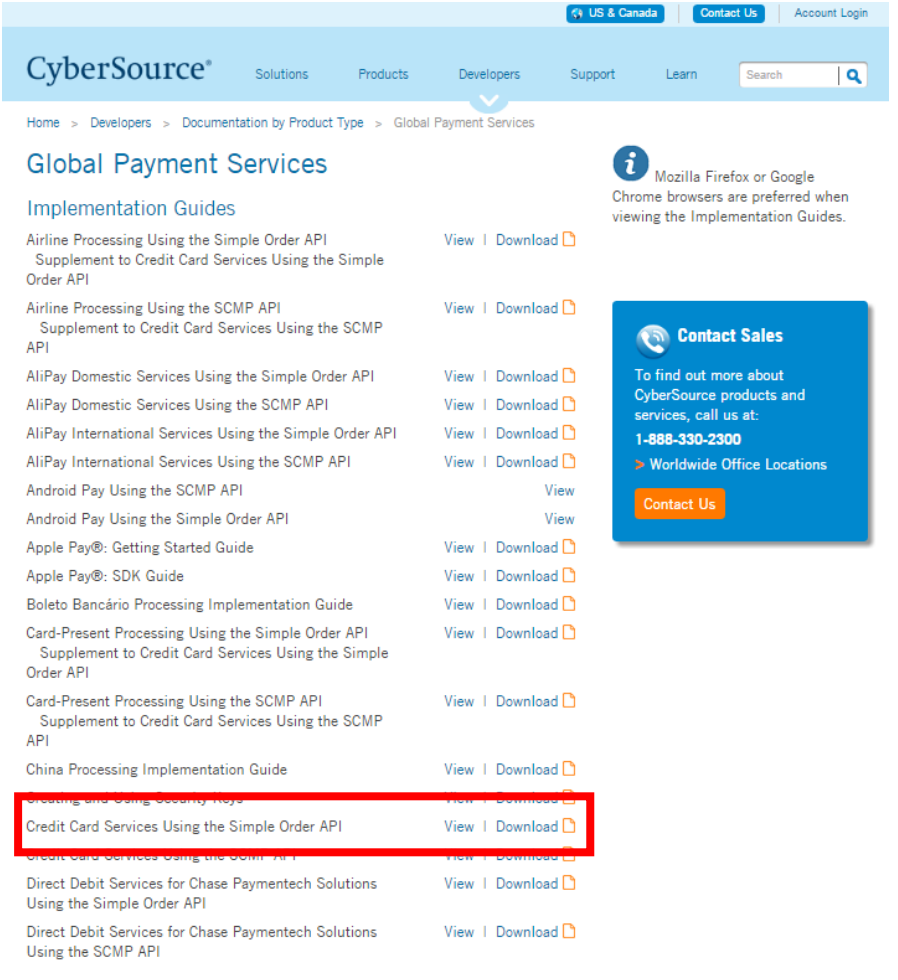
1.2 About CyberSource

CyberSource is the payment gateway providers for front-end integration. Payment requests will be submitted thru CyberSource and then it will be routed to Global Payments for payment processing.



2.1 Pre-Read Materials

Before you start, please ensure the following materials are with you to start the integration.

Document / Link	Download
Global Payments – Getting Started with Apple Pay	This document.
CyberSource – Apple Pay Using the Simple Order API	http://apps.cybersource.com/library/documentation/dev_guides/apple_payments/SO_API/Apple_Pay_SO_API.pdf
CyberSource Credit Card Services – Using the Simple Order API	<p>1: https://www.cybersource.com 2: Click Developers 3: Under ‘Documentation by Product Type’, click Global Payment Services. 4: Search for Credit Card Services Using the Simple Order API’ 5: You can use either the online HTML version or PDF version.</p>  <p>Shortcut URL: http://apps.cybersource.com/library/documentation/dev_guides/CC_Svcs_SO_API/Credit_Cards_SO_API.pdf</p>
Getting Started with Apple Pay	<p>1: https://developer.apple.com 2: Click ‘Getting Started with Apple Pay’</p>

	Shortcut URL: https://developer.apple.com/apple-pay/get-started/
Apple Pay Programming Guide	1: https://developer.apple.com 2: Click 'Apple Pay Programming Guide' Shortcut URL: https://developer.apple.com/library/ios/ApplePay_Guide/
Apple Pay Domain Verification (For Apple Pay on the web)	https://developer.apple.com/support/apple-pay-domain-verification/
Apple Pay JS (for Web Support)	1: https://developer.apple.com 2: Click 'Apple Pay JS' Shortcut URL: https://developer.apple.com/reference/applepayjs/
Apple Pay Best Practices Guide	Please request this document from Global Payments if you do not have it.

2.2 Pre-requisites

We will also assume you the followings:

Document / Link	Description
CyberSource MID	A CyberSource merchant MID, please contact your sales manager at Global Payments to obtain a CyberSource Merchant MID. We will then provide you a CyberSource MID
Apple Developer Account	Obtain thru Apple Developer
iOS App	Required for Apple Pay in-app payment.
Web Site	Required for Apple Pay on the web payment.

2.0 Configure Global Payments Apple Pay Environment

This sub-title of this session is arranged in a manner that 'Actor' – 'Task'. For example, 'Merchant – Obtain the Apple Merchant Apple ID' would be an action done by the merchant.

2.1 Merchant – Register an Apple Merchant ID.

Visit developer.apple.com → Apple Pay Programming Guide → Configuring Your environment. Then please following the configuring your environment and send Global Payments back the merchant ID.

Configuring Your Environment

A merchant ID identifies you to Apple Pay as being able to accept payments. The public key and certificate associated with your merchant ID is used as part of the payment process to encrypt payment information. Before your app can use Apple Pay, you need to register a merchant ID and configure its certificate.

To register a merchant ID

1. In Member Center, select [Certificates, Identifiers & Profiles](#).
2. Under Identifiers, select Merchant IDs.
3. Click the Add button (+) in the upper-right corner.
4. Enter a description and identifier, and click Continue.
5. Review the settings, and click Register.
6. Click Done.

Shortcut URL:

https://developer.apple.com/library/ios/ApplePay_Guide/Configuration.html#//apple_ref/doc/uid/TP40014764-CH2-SW1

2.2 Global Payments – Generate the CSR

Please send the Apple Merchant ID to Global Payments (apecommerce@globalpay.com), Global Payments will send you back the CSR file.

2.3 Merchant – Configure the certificate for your merchant ID

Visit developer.apple.com → Apple Pay Programming Guide → Configuring Your environment. Then please following the configuring your environment and send Global Payments back the merchant ID.

To configure a certificate for your merchant ID

1. In Member Center, select [Certificates, Identifiers & Profiles](#).
2. Under Identifiers, select Merchant IDs.
3. Select the merchant ID from the list, and click Edit.
4. Click Create Certificate, follow the instructions to obtain or generate your certificate signing request (CSR), and click Continue.
5. Click Choose File, select your CSR, and click Generate.
6. Download the certificate by clicking Download, and click Done.

If you see a warning in Keychain Access that the certificate was signed by an unknown authority or that it has an invalid issuer, make sure you have the WWDR intermediate certificate - G2 and the Apple Root CA - G2 installed in your keychain. You can download them from apple.com/certificateauthority.

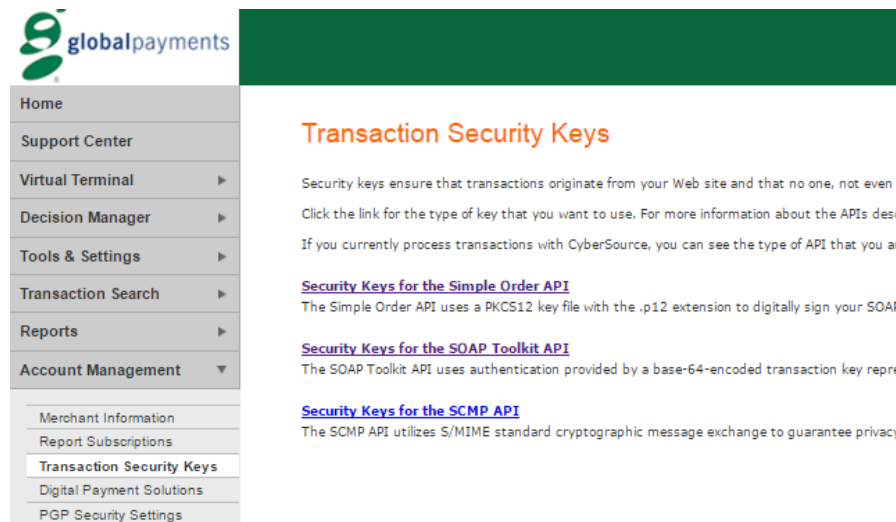
Shortcut URL:

https://developer.apple.com/library/ios/ApplePay_Guide/Configuration.html#//apple_ref/doc/uid/TP40014764-CH2-SW1

2.4 Merchant – Create the Transaction Security Keys

Security keys ensure that transactions originate from your Web site and that no one, not even CyberSource, can run transactions on your behalf by using your keys. Apple Pay requires Simple Order API. The transaction security key could be obtained thru CyberSource Business Center (EBC)

- 1: Login to EBC and Under Account Management.
- 2: Click transaction Security Keys.
- 3: Click Security Keys for Simple Order API.



- 4: Click 2028-Bit Key and then save the .12 file

3.0 Integration & Testing

3.1 iOS – Create the Payment Requests

Please refer to the Apple Pay Programming Guide:

https://developer.apple.com/library/ios/ApplePay_Guide/CreateRequest.html#//apple_ref/doc/uid/TP40014764-CH3-SW2

Payment data must be sent to CyberSource in base64Encoded format:

```
NSString *base64Encoded = [mypaydata base64EncodedStringWithOptions:0];
NSLog(@"Value of paymentdata = %@", stPaymentData);
NSLog(@"Value of base64 paymentdata = %@", base64Encoded); // this is the
one needed by CYBS.
```

3.1 Server Side Code

The encrypted Payment data above should be sent to your e-commerce back-end system.

Populate the field (encryptedPayment_data) with the encrypted payment data value obtained from the paymentData property of the PKPaymentToken object.

The code in the below is provided for reference only, both Global Payments and CyberSource do not provide support on the code below:

```
public static String runApplepay() {
    String requestID = null;
    HashMap request = new HashMap();

    String args[] = {"cybs.properties"};
    //payment properties
    Properties props=Utility.readProperties( args );

    request.put( "merchantReferenceCode", "Apple Pay" );
    request.put( "paymentSolution", "001");

    request.put( "billTo_firstName", "Steve" );
    request.put( "billTo_lastName", "Jobs" );
    request.put( "billTo_street1", "Global Street" );
    request.put( "billTo_city", "Singapore" );
    request.put( "billTo_state", "SG" );
    request.put( "billTo_postalCode", "123456" );
    request.put( "billTo_country", "SG" );
    request.put( "billTo_email", "nobody@globalpayments.com" );
    request.put( "billTo_ipAddress", "127.0.0.1" );
    request.put( "billTo_phoneNumber", "650-965-6000" );
    request.put( "pos_environment", "4" );

    request.put( "purchaseTotals_grandTotalAmount", "1.00");
    request.put( "encryptedPayment_descriptor",
"Rk1lEPUNPTU1PTi5BUFBMRS5JTkFQUC5QQV1NRU5U");
}
```


Getting Started Guide

```
request.put( "encryptedPayment_data", "xxxxx encrypted payment data
from iOS xxxxx");
request.put("encryptedPayment_encoding", "Base64");
request.put( "ccAuthService_run", "true" );

// add more fields here per your business needs

Try {
//displayMap( "CREDIT CARD AUTHORIZATION REQUEST:", request );

Calendar cal = Calendar.getInstance();
SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
String curtime = sdf.format(cal.getTime());

// run transaction now
long cybsStartTime = System.currentTimeMillis();
HashMap reply = Client.runTransaction( request, props );
long cybsEndTime = System.currentTimeMillis();

System.out.println("Auth timer " + (cybsEndTime - cybsStartTime));

displayMap( "CREDIT CARD AUTHORIZATION REPLY:", reply );

// if the authorization was successful, obtain the request id
// for the follow-on capture later.
String decision = (String) reply.get( "decision" );
if ("ACCEPT".equalsIgnoreCase( decision ))
{
    requestID = (String) reply.get( "requestID" );

    //writing(curtime,requestID,cybsEndTime - cybsStartTime);
}
}
catch (ClientException e)
{
    System.out.println( e.getMessage() );
    if (e.isCritical())
    {
        handleCriticalException( e, request );
    }
}
catch (FaultException e)
{
    System.out.println( e.getMessage() );
    if (e.isCritical())
    {
        handleCriticalException( e, request );
    }
}

return( requestID );
}
```

3.3 Testing

Currently, Testing will be carried out in a reserved production environment, where a live Visa Card and MasterCard must be used for testing.

The Apple Sandbox environment does not work with CyberSource Sandbox environment now. Integration of the two sandbox are being evaluated now but it is not supported at this moment.

When using a live credit card during the integration / testing stage, the fund won't be transferred to the merchant bank's account until account underwriting / review is completed. The fund will be remain in the merchant account.

Once the testing is completed, the merchant can use the refund feature to return the money back to the credit card holders.

Merchant should also send the screen dumps of successful / failure transactions to Global Payments for final review.

3.4 Refund

Refund could be done via CyberSource Business Center (EBC).

3.4 Special Note – Shipping / Billing Address

PKContact.postalAddress.isoCountryCode sometimes do not return values from Apple PassKit API (when the interface language is not English), while CyberSource API only accept isoLanguage in 2 characters ISO standard country code.

Checking must be done or a manual mapping of country from Passkit SDK to iso standard country is required until the problem is fixed by Apple.

The following fields are usually screwed:

billTo/shipTo state: String (2)

billTo/shipTo country: String (2)

billTo/shipTo phoneNumber: String (15)

billTo/shipTo postalCode: String (9)

billTo/shipTo street1/2: String (40)

3.6 Disable of Billing Address

It is possible to disable the billing address. Please contact Global Payments for the configuration. Details could be referred to 'Relaxed Requirements for Address Data and Expiration Date' in CyberSource Credit Card Services Simple Order API'.

3.6 Special Note – pos_environment

Please always set an API field pos_environment=4 in the authorization request.

3.7 Sample Request – Gateway Decryption (Visa)

```

<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
<merchantID>demomerchant</merchantID>
<merchantReferenceCode>demorefnum</merchantReferenceCode>
<billTo>
  <firstName>James</firstName>
  <lastName>Smith</lastName>
  <street1>1295 Charleston Road</street1>
  <city>Test City</city>
  <state>CA</state>
  <postalCode>99999</postalCode>
  <country>US</country>
  <email>demo@example.com</email>
</billTo>
<purchaseTotals>
  <currency>USD</currency>
  <grandTotalAmount>5.00</grandTotalAmount>
</purchaseTotals>
<card>
  <cardType>001</cardType>
</card>
<ccAuthService run="true">
  <commerceIndicator>internet</commerceIndicator>
</ccAuthService>
<encryptedPayment>
  <data>ABCDEFabcdefABCDEFabcdef0987654321234567</data>
  <descriptor>Rk1EPUNPTU1PTi5BUFBMRS5JTkFQUC5QQV1NRU5U</descriptor>
  <encoding>Base64</encoding>
</encryptedPayment>
<paymentSolution>001</paymentSolution>

```

3.8 Sample Request – Gateway Decryption (MasterCard)

```

<requestMessage xmlns="urn:schemas-cybersource-com:transaction-data-1.121">
<merchantID>demomerchant</merchantID>
<merchantReferenceCode>demorefnum</merchantReferenceCode>
<billTo>
  <firstName>James</firstName>
  <lastName>Smith</lastName>
  <street1>1295 Charleston Road</street1>
  <city>Test City</city>
  <state>CA</state>
  <postalCode>99999</postalCode>
  <country>US</country>
  <email>demo@example.com</email>
</billTo>
<purchaseTotals>
  <currency>USD</currency>
  <grandTotalAmount>5.00</grandTotalAmount>
</purchaseTotals>
<card>
  <cardType>002</cardType>
</card>
<ccAuthService run="true">
  <commerceIndicator>spa</commerceIndicator>
</ccAuthService>
<encryptedPayment>
  <data>ABCDEFabcdefABCDEFabcdef0987654321234567</data>
  <descriptor>Rk1EPUNPTU1PTi5BUFBMRS5JTkFQUC5QQV1NRU5U</descriptor>
  <encoding>Base64</encoding>
</encryptedPayment>
<paymentSolution>001</paymentSolution>
</requestMessage>

```


4.0 Miscellaneous

4.1 Report Download

Report Download API could be used to programmatically download transactions reports from CyberSource. Please note that a report download user account must be created first.

Document / Link	Download
CyberSource Reporting Developer Guide	1: https://www.cybersource.com 2: Click Developers 3: Click Reporting & Reconciliation 4: Click Reporting Developer Guide Shortcut URL: http://apps.cybersource.com/library/documentation/dev_guides/Reporting_Developers_Guide/reporting_dg.pdf

5.0 F.A.Q.

5.1 How can I carry out refund with Apple Pay?

Refund could be done via CyberSource Business Center (EBC) or via API. Please note that a CYBS transaction could be referenced by one of the following criteria where Transaction ID or Purchase Identifier are preferred:

- Transaction ID (generated by CyberSource).
- Purchase Identifier (provided by merchant).
- Name / Contact of the buyers.
- D-PAN.

5.2 Can I obtain the full D-PAN?

No – you could only obtain the first 6 and last 4 digits of the D-PAN.