Payeezy iOS SDK Integration Guide

April 2015

Payeezy DEVELOPERS

First Data. beyond the transaction

Contents

Quickly integrate Payeezy iOS SDK from First Data into your iOS applications

If you want to enable secure and convenient payments to your payment applications, this guide will get you up and running quickly. Payeezy handles all the heavy lifting of the complex tokenization and protects your customers' transactions. It is simple to create a developer test account and apply for a merchant account through our developer portal.

Payeezy℠, from First Data, is part of the Small Business Solutions Suite, which includes Clover™, Insightics℠, Perka™ and TransArmor®. The Payeezy eCommerce Solution empowers SMBs to expand their horizons and easily grow their business online or via mobile by reaching new customers no matter where they are.

This documentation refers to the iOS integration guide included within the overall Payeezy eCommerce Solution. Henceforth, all references to Payeezy are in relation to Payeezy APIs.

Minimum technical requirements

The iOS mobile SDK requires iOS SDK 6 and XCode 5.1 and above

Payeezy iOS SDK initialization

Please go through prerequisites, to get an overview of our developer portal. Link for prerequisites https://collaboration.1dc.com/sites/ecommerce/Payeezy/payeezy_integration/final/pdf/get_started_with_payeezy042015.pdf
Build iOS app to make purchases directly in your app.

Steps to integrate code with Payeezy SDK

Step1: Download the Payeezy iOS SDK from GitHub

Step 2: Define merchant Token, Apikey and ApiSecret (.m file)

Step 3: Initialize PayeezySDK object with KApiKey, KApiSecret, KToken and KURL

Step 4: Call transaction method and pass the required parameters

For test credit card/eCheck (telecheck) test data, refer https://collaboration.1dc.com/sites/ecommerce/Payeezy/payeezy_integration/final/pdf/payeezy_testdata042015.pdf

Getting started

Using GitHub

Clone Payeezy SDK using with HTTPS or Subversion
Using clone command: `git clone https://github.com/payeezy/payeezy_ios.git`

Or simply download zip file: https://github.com/payeezy/payeezy_ios/archive/master.zip

To know more about GitHub, click http://github.com

Then add the Payeezy library to your Xcode project:

1. In the menubar, click on 'File' then 'Add files to "Project"...'.
2. Select the 'Payeezy' directory in the downloaded repository.
3. Make sure 'Copy items into destination group's folder (if needed)' is checked.
4. Click 'Add'.
5. In your project settings, go to the "Build Phases" tab, and make sure MobileCoreServices.framework , Foundation.framework, Security.framework, SystemConfiguration.framework and PassKit.framework are included in the "Link Binary With Libraries" section. If you're targeting versions of iOS before iOS8, make sure PassKit.framework is listed as "Optional."

Define merchant Token, Apikey and ApiSecret (file name)

```
#define KApiKey    @"test_apikey_bA8hIqpzuAVW6itHqXsXwSl6JtFWPCA0"
#define KApiSecret @"test_apitsecret_YmI4YzA1NmRkZmQzMzA1ZmIZjYzwMWIzZThkMWU2NGRjZmI4OWE5NGRiMzM4NA=="
#define KToken     @"test_merchant_token_fdoa-a480ce8951daa73262734cf102641994c1e55e7cdf4c02b6"
#define KURL       @"https://api-cert.payeezy.com/v1/transactions"
```

Initialize PayeezySDK object with KApiKey, KApiSecret, KToken and KURL

```
// initialize Payeezy object with key,token and secret value
PayeezySDK* myClient = [[PayeezySDK alloc]initWithApiKey:KApiKey apiSecret:KApiSecret merchantToken:KToken url:KURL];
```

Initialize NSDictionary to capture input parameter: this is optional

```
/* */
// credit card info
 NSDictionary* credit_card = @{
                                    @"type":@"visa",
                                   @"cardholder_name":@"Jacob Test",
                                   @"card_number":@"4012000033330026",
                                   @"exp_date":@"0416",
                                   @"cvv":@"123"
                                 };
 // Transaction info
 NSDictionary* transaction_info = @{
                                   @"currencyCode":@"USD",
                                   @"amount":amount,
                                    @"merchantRefForProcessing":@"abc1412096293369"
                                   };
```

```
// initialize Payeezy object with key,token and secret value
  PayeezySDK* myClient = [[PayeezySDK alloc]initWithApiKey:KApiKey apiSecret:KApiSecret merchantToken:KToken url:KURL];


// call authorize method with payment/creditcard data
  [myClient submitAuthorizeTransactionWithCreditCardDetails:credit_card[@"type"]
        cardHolderName:credit_card[@"cardholder_name"] cardNumber:credit_card[@"card_number"]
        cardExpirymMonthAndYear:credit_card[@"exp_date"] cardCVV:credit_card[@"cvv"]
        currencyCode:transaction_info[@"currencyCode"] totalAmount:amount
        merchantRefForProcessing:transaction_info[@"merchantRefForProcessing"]
        completion:^(NSDictionary *dict, NSError *error) {

                NSString *authStatusMessage = nil;

                // handle error and response
                if (error == nil)
                {
                  authStatusMessage = [NSString stringWithFormat:@"Transaction
                  Successful\rType:%@\rTransaction ID:%@\rTransaction Tag:%@\rCorrelation Id:%@\rBank
                  Response Code:%@",
                          [dict objectForKey:@"transaction_type"],
                          [dict objectForKey:@"transaction_id"],
                          [dict objectForKey:@"transaction_tag"],
                          [dict objectForKey:@"correlation_id"],
                          [dict objectForKey:@"bank_resp_code"]];
                }
                else
                {
                  authStatusMessage = [NSString stringWithFormat:@"Error was encountered processing
                  transaction: %@", error.debugDescription];
                }
```

Call transaction method and pass the required parameters. For more information on request and response parameters refer to apple doc located @

Submitting/Generating transaction (method of payments) with example

Payeezy supports the following method of payments

- Credit Card Payments
- PayPal Transactions
- Gift Card (via ValueLink) Transactions
- eCheck (via TeleCheck) Transactions

For API processing details, click here https://developer-qa.payeezy.com/integration

Additional capabilities API (if applicable)

- Partner Reporting API - Use our powerful query engine to retrieve payment records. Supports complex filtering, sorting, pagination and more. This is exclusively for Third Party Partners and applicable for a **live** environment only.

For Reporting API processing details, click here https://developer-qa.payeezy.com/payeeyz_ref_docs/apis/get/transactions-0

Security related (HMAC, Token generation) with example

GENERATE HMAC

Construct the data param by appending the parameters below in the same order as shown. a. apikey - API key of the developer. b. nonce - secure random number. c. timestamp - epoch timestamp in milliseconds. d. token - Merchant Token. e. payload - Actual body content passed as post request. ***Compute HMAC SHA256 hash on the above data param using the key below*** f. apiSecret - Consumer Secret token for the given api key ***Calculate the base64 of the hash which would be our required Authorization header value.***

```
// import required header files
#import <CommonCrypto/CommonDigest.h>;
#import <CommonCrypto/CommonHMAC.h>;

NSString* apiKey = @"<your api key>";
NSString* apiSecret = @"<your consumer secret>";
NSString* token = @"<Merchant Token>";
NSString* payload = @"<For POST - Request body / For GET - empty string>";
NSString* nonce = @"<Crypographically strong random number>";
NSString* timestamp = @"<Epoch timestamp in milli seconds>";
NSString *hmacData =
[NSString stringWithFormat:@"%@%@%@%@%@%@",apiKey,nonce,timestamp,token,payload];

// Make sure the HMAC hash is in hex
unsigned char outputHMAC[CC_SHA256_DIGEST_LENGTH];
const char* keyChar = [apiSecret cStringUsingEncoding:NSUTF8StringEncoding];
const char* dataChar = [hmacData cStringUsingEncoding:NSUTF8StringEncoding];
CCHmac(kCCHmacAlgSHA256, keyChar, strlen(keyChar), dataChar, strlen(dataChar), outputHMAC);
NSData* hmacHash =
[[NSData alloc] initWithBytes:outputHMAC length:sizeof(outputHMAC)];

// The conversion (NSData to HexString) shown below is just an example.Please use other appropriate methods to do the
conversion.
NSString* hmacHashHexString =
[[hmacHash description] stringByReplacingOccurrencesOfString:@" " withString:@""];


// Authorization : base64 of hmac hash -->
NSString* authorization =
[[hmacHashHexString dataUsingEncoding:NSUTF8StringEncoding] base64EncodedStringWithOptions:0];
NSLog(@"Authorization Generated: %@",authorization);
```

Reference docs for Payeezy iOS SDK

Reference docs
(https://collaboration.1dc.com/sites/ecommerce/Payeezy/payeezy_integration/final/htmldocs/index.html)