iPay88 iOS (MT) Mobile Payment SDK v1.0.10

(Malaysia Gateway)

Contents

1 Mobile Payment SDK – iOS

1.1 Pre-requites when develop iOS Apps that use iPay88 iOS Mobile SDK channel

Please provide your Merchant Code to Ipay88 to verify merchant identity.

Ipay88 need to configure your Merchant Code into Ipay88 server to allow you can use Ipay88 Mobile Payment channel.

2 iPay88 IOS Mobile Payment Library

This section provides details about the Mobile Payments Library API, and it provides instructions and examples for integrating the library with merchant iOS application.

2.1 Mobile Payments Library API Reference

The flow of the library is as follow:

1. Your application initializes the library.
2. Your UI view controller should add the UI view returned by the library as its subview.
3. After buyers complete their payments, the library returns a callback to your application with the status of the payment and the Transaction Id.

2.2 Adding the Library and Importing Header Files

1. In XCode, open your build target and go to “Build Phases” tab.
2. Under “Link Binary With Libraries”, add the “libipay88sdk.xcframework” static library.
3. Add the headers files (“Ipay.h” & “IpayPayment.h”) into your project.

Also, import the appropriate headers into your application classes. The following classes must be imported:

```
#import “Ipay.h”
#import “IpayPayment.h”
```

2.3 Start the Library UIView

The Library uses the native UIWebView to start the checkout flow, and to communicate completion back to you. You can implement PaymentResultDelegate to be informed immediately upon successful completion of a payment.

To start the Ipay88 payment, you must initialize the Ipay object first.

You must use the checkout method and give it the IpayPayment object. IpayPayment handles simple payments, which support single receivers of payments with one transaction and a few details.

In the following example, the buyer checks out with a simple payment for a single recipient:

```
paymentsdk = [[Ipay alloc] init];
paymentsdk.delegate = self;
IpayPayment *payment = [[IpayPayment alloc] init];
[payment setPaymentId:@"2"];
[payment setMerchantKey:@"apple88KEY"];
[payment setMerchantCode:@"M09999"];
[payment setRefNo:@"ORD1188"];
[payment setAmount:@"1.00"];
[payment setCurrency:@"MYR"];
[payment setProdDesc:@"Payment for ORD1188"];
[payment setUserName:@"John Woo"];
[payment setUserEmail:@"johnwoo@yahoo.com"];
[payment setRemark:@"ORD1188"];
[payment setLang:@"ISO-8859-1"];
[payment setCountry:@"MY"];
[payment setBackendPostURL:@"http://www.mywebsite.com/backend_response.php"];
[payment  setappdeeplink:@" app://open.my.app/receipt/RefNo=A00000001"];
[payment setActionType:@"BT"];
[payment setTokenId:@"qwerty987abc7966"];
[payment setPromoCode:@"PROMOTION"];
[payment setFixpaymentId:@"16"];
UIView *paymentView = [paymentsdk checkout:payment];
[self.view addSubview:paymentView];
```

2.4 Method in the Mobile Payments Library

The “checkout” method

This method returns a UIView(an instance of UIWebView).

```
- (UIView *)checkout:(IpayPayment *)payment;
```

The interface for providing details on when a payment is completed is defined in <PaymentResultDelegate> protocol. This interface provides you with a way to be notified immediately when a payment has completed:

```
@protocol PaymentResultDelegate <NSObject>

- (void)paymentSuccess:(NSString *)refNo withTransId:(NSString *)transId withAmount:(NSString *)amount
  withRemark:(NSString *)remark withAuthCode:(NSString *)authCode withTokenId:(NSString *)tokenId
  withCCName:(NSString *)ccName withCCNo:(NSString *)ccNo
withS_bankname:(NSString *)s_bankname withS_country:(NSString *)s_country;

- (void)paymentFailed:(NSString *)refNo withTransId:(NSString *)transId withAmount:(NSString *)amount
  withRemark:(NSString *)remark withErrDesc:(NSString *)errDesc withTokenId:(NSString *)tokenId
  withCCName:(NSString *)ccName withCCNo:(NSString *)ccNo
withS_bankname:(NSString *)s_bankname withS_country:(NSString *)s_country;

- (void)paymentCancelled:(NSString *)refNo withTransId:(NSString *)transId withAmount:(NSString
  *)amount withRemark:(NSString *)remark withErrDesc:(NSString *)errDesc withTokenId:(NSString
  *)tokenId withCCName:(NSString *)ccName withCCNo:(NSString *)ccNo
withS_bankname:(NSString *)s_bankname withS_country:(NSString *)s_country;

@end
```

Requery method

```
- (void)requery:(IpayPayment *)payment
```

Note: only the refNo, merchantCode and amount properties of the IpayPayment object are required for requery(the rest can be left empty).

The interface for providing details on when a requery is completed is defined in <PaymentResultDelegate> protocol. This interface provides you with a way to be notified immediately when a requery has completed:

```
@protocol PaymentResultDelegate <NSObject>

- (void)requerySuccess:(NSString *)refNo withMerchantCode:(NSString *)merchantCode
  withAmount:(NSString *)amount withResult:(NSString *)result;
- (void)requeryFailed:(NSString *)refNo withMerchantCode:(NSString *)merchantCode
  withAmount:(NSString *)amount withErrDesc:(NSString *)errDesc;

@end
```

2.5 Custom Objects in the Mobile Payments Library

The Mobile Payments Library includes custom objects for passing information between the library and your application during checkout.

IpayPayment

This object is passed to the library. This object contains all the values for a payment.

Method	Description
setMerchantKey(String merchantKey)	(Required) – Merchant Key that provided by Ipay88. E.g. apple88KEY
setMerchantCode(String merchantCode)	(Required) – Merchant Code that provided by Ipay88. E.g. M09999
setPaymentId(String paymentId)	(Required) – PaymentId is the value to request payment method to appear on Ipay88 landing page. Please refer to **Appendix I** for available PaymentId **Note:** Ensure the Merchant Account had subscribed to the payment option in iPay88 before specify it in the APP
setRefNo(String refNo)	(Required) – Reference number for merchant reference purposes, should be unique for each transaction.
setAmount(String amount)	(Required) – Final Amount to pay and in 2 decimal point, E.g. **10.90** *For testing, please use 1.00
setCurrency(String currency)	(Required) – Currency code that based on standard ISO. E.g. MYR
setProdDesc(String prodDesc)	(Required) – Simple Product Description. E.g. Ticket for Concert. **Note:** Special characters is not allowed.
setUserName(String userName)	(Required) – Customer name in merchant's system. E.g. John Woo
setUserEmail(String userEmail)	(Required) – Customer email address in merchant's system with valid email format. E.g. johnwoo@yahoo.com
setRemark(String remark)	(Optional) – Remark for particular transaction. **Note:** Special characters is not allowed.
setLang(String lang)	(Optional) – Language. *Default - ISO-8859-1
setCountry(String country)	(Required) – Set this field to connect iPay88 gateway to selected country. *Default – MY *Available values: MY / PH / ID / BD **Note:** Please ensure merchant have merchant account before connect to specified country gateway
setBackendPostURL(String url)	(Required) – Specify a valid merchant callback URL when payment success. E.g. http://www.myshop.com/backend_page.php **Note:** Refer to section "2.7 Payment Response Parameters return to BackendPostURL" for detail of parameters return from iPay88
setappdeeplink(String setappdeeplink)	(Optional) – App Deeplink path for merchant mobile app. For ewallet payment. Refer to 2.9 Appdeeplink Implementation for detail
setActionType(String ActionType)	(Optional) – Type of action. ActionType = ""

setTokenId(String TokenId)	(Optional) – Bound credit card TokenId. TokenId = “”
setPromoCode(String PromoCode)	(Optional) – Promotion Code
setFixpaymentId(String setFixpaymentId)	(Optional) – FixpaymentID is highest priority payment method. iPay88 will take FixpaymentID method to appear on iPay88 landing page. No payment selection allowed on iPay88 Landing page. Please refer to **Appendix I** for available PaymentId **Note:** Ensure the Merchant Account had subscribed to the payment option in iPay88 before specify it in the APP

2.6 Merchant Tokenization

This is a feature where merchant will redirect customer to iPay88 OPSG to perform bind card (BC), bind card & charge (BCC) and subsequent charge activity. It's a feature where customer credit card information (encrypted credit card number, card holder name, expiry month, expiry year, card issuing bank and country) will be stored at iPay88 for future payment.

Customer will be able to make payment without the hassle to enter credit card information every time he/she making any purchases. Customer only have to enter the CVV/CVV2 number

2.6.1 Bind Card (BC)

In bind card process, MYR 1.00 value will be used for pre-authorization only. It will not be capture and charge to customer credit card. For bind card activity, PaymentId = 55, Amount = 1.00 and ActionType = "BT" MUST be use. Refer to the below table for the full payment request parameters.

Additional values for **IpayPayment** object whenever send payment request

Method	Description
setPaymentId(String paymentId)	*(Required)* – **PaymentId = 55**
setActionType(String actionType)	*(Required)* – **ActionType** = **BT**
setTokenId(String TokenId)	*(Option)* – TokenId = ""
setAmount (String amount)	*(Required)* – Amount = 1.00 (Amount must MYR 1.00 for Bind Card purpose)

TokenId and credit card information will be return to merchant for future credit card payment once it's successfully processed. Refer to section 2.6.3 for usage of the TokenId.

2.6.2 Bind Card & Charge (BCC)

BCC and BC are almost the same, the different are BC use **Amount=1.00** and using **PaymentId=55** for pre-authorization purpose only. BCC will be using actual payment via credit card with successful transaction only card will be bind into iPay88 system.

For bind card & charge activity, PaymentId = 2 and ActionType = "BT" MUST be use.

Refer to the below table for additional payment request parameters.

Method	Description
setPaymentId(String paymentId)	*(Required)* – **PaymentId = 2**
setActionType(String actionType)	*(Required)* – **ActionType** = **BT**
setTokenId(String TokenId)	*(Option)* – TokenId = ""

2.6.3 Subsequent Charge (SC)

With the TokenId generated for merchant when a customer credit card successfully binds into iPay88 system, credit card information will be returned as well for merchant to identify the TokenId is bind for which credit card if customer have bind multiple cards.

Below are the additional payment request parameters will be assigned to IpayPayment object for subsequent charge after card is bind with iPay88 OPSG:

Method	Description
setPaymentId(String paymentId)	*(Required)* – **PaymentId = 2**
setActionType(String actionType)	*(Required)* – **ActionType** = **SC**
setTokenId(String TokenId)	*(Required)* – Unique id assigned by iPay88 for the bind card

2.7 Payment Response Parameters return to BackendPostURL

HTTPS POST response from iPay88 to merchant BackendPostURL when payment success.

Parameter Name	Data Type	Size	M/O	Description
MerchantCode	String	20	M	Merchant Code provided by iPay88 and use to uniquely identify the Merchant. E.g. M09999
PaymentId	Integer		M	Please refer to Appendix I for possible PaymentId value return to BackPostURL.
RefNo	String	50	M	Reference number for merchant reference purposes, should be unique for each transaction.
Amount	Currency		M	Payment amount with two decimals and thousand symbols. E.g. 1,278.90
Currency	String	5	M	Currency code that based on standard ISO. E.g. MYR
Remark	String	100	O	Remark for particular transaction.
TransId	String	30	O	iPay88 Transaction ID. E.g. T019988877700
AuthCode	String	50	O	Bank reference number. Note: Sometime bank may not return reference number to gateway
Status	String	1	M	Use to indicate payment status “1” – Success “0” – Fail
ErrDesc	String	100	O	Payment status description (Refer to Appendix I)
Signature	String	100	M	SHA256 signature (Refer to “2.8 BackendPostURL Response page signature” on how the Signature generated)
CCName	String	200	O	Applicable for credit card payment only. Credit card holder name
CCNo	String	16	O	Applicable for credit card payment only. Masked credit card number. First six and last four of credit card number. Eg: 492159xxxxxx4941
S_bankname	String	100	O	Applicable for credit card payment only. Credit card issuing bank name
S_country	String	100	O	Applicable for credit card payment only. Credit card issuing country

** Refer to next page for detail of BackendPostURL implementation…

BackendPostURL (Backend Post Feature)

Backend POST feature is server to server technology where it does not depend on user's web browser to return payment response data to merchant website.

With this feature implemented, merchant server still can get the payment status on the backend (asynchronously) even if merchant mobile application (app) fails to get status via iPay88 SDK which may be due to user accidentally closed web browser, internet connection timeout and etc.

Implementation

On the merchant server, create a page (BackendPostURL) to accept backend post response parameters from iPay88 gateway. The list of parameters return is listed on section *"2.7 Payment Response Parameters return to BackendPostURL"*.

On the BackendPostURL page, write out the word 'RECEIVEOK' only (without quote) as an acknowledgement after update order to *PAID* on merchant system.

iPay88 will re-try post success status to merchant BackendPostURL up to 3 times on different time interval if no 'RECEIVEOK' acknowledgement from merchant server.

Example:
In ASP code > response.write "RECEIVEOK"
In PHP code > echo "RECEIVEOK";

Merchant Developer Note on BackendPostURL implementation

a) iPay88 server will ONLY return status if the transaction is payment success. No status will return if the payment is failed.

b) Ensure no HTML code/tag on merchant BackendPostURL page.

c) Ensure merchant BackendPostURL contain coding/function to update order besides using *PaymentResultDelegate(paymentSuccess)* method.

d) Ensure merchant BackendPostURL implement verification check on the received response data to prevent user hijack merchant system, e.g. implement Response Signature comparison check.

e) Ensure implement checking to determine either " *PaymentResultDelegate(paymentSuccess)* method" or "BackendPostURL" when update the order, so merchant system will not update status more than 1 time. Note: When payment success, iPay88 will return success payment status to "*SDK*" and "*BackendPostURL*".

f) Ensure merchant BackendPostURL should only reply 'RECEIVEOK' when note (b), (c), (d) and (e) implemented properly.

g) BackendPostURL is not a replacement for *PaymentResultDelegate* method to get payment success status thus merchant still require to implement *PaymentResultDelegate* on mobile application (app) to obtain payment status.

Sample (BackendURL Payment response)

2.7.1 ASP sample code

```
<%
MerchantCode = Request.Form("MerchantCode")
PaymentId     = Request.Form("PaymentId")
RefNo        = Request.Form("RefNo")
Amount       = Request.Form("Amount")
eCurrency    = Request.Form("Currency")
Remark       = Request.Form("Remark")
TransId       = Request.Form("TransId")
AuthCode      = Request.Form("AuthCode")
Status        = Request.Form("Status")
ErrDesc       = Request.Form("ErrDesc")
Signature     = Request.Form("Signature")
%>
<Add your programming code here>

IF Status=1 THEN
       COMPARE Return Signature with Generated Response
       Signature
       // update order to PAID
       response.write "RECEIVEOK"
ELSE
// update order to FAIL
```

2.7.2 PHP sample code

```
<?PHP
$merchantcode     = $_REQUEST["MerchantCode"];
$paymentid        = $_REQUEST["PaymentId"];
$refno            = $_REQUEST["RefNo"];
$amount           = $_REQUEST["Amount"];
$ecurrency        = $_REQUEST["Currency"];
$remark           = $_REQUEST["Remark"];
$transid          = $_REQUEST["TransId"];
$authcode         = $_REQUEST["AuthCode"];
$status           = $_REQUEST["Status"];
$errdesc          = $_REQUEST["ErrDesc"];
$signature        = $_REQUEST["Signature"];
PHP?>
<Add your programmingcode here>

IF ($status=1) {
      COMPARE Return Signaturewith GeneratedResponse Signature
      // update order to PAID
      echo "RECEIVEOK";
}
ELSE  {
// update order to FAIL
}
```

2.8 BackendPostURL Response page signature

This appdeeplink parameter is for merchant mobile app to mobile app link only and to handle payment method like TNG eWallet, Boost Wallet and Shopeepay.
Once the user completes the payment flow, user will redirect from eWallet app back to merchant mobile app by calling merchant mobile app deeplink, do note that when redirect back to merchant mobile app there no response data will return.
Upon return to merchant app deeplink URL, use requery to get payment status (iPay88 will also return success status to merchants BackendURL and then merchants can update the order to Paid

If the Merchant request is successful, the response message will contain as SHA-256 hashed signature. The hash signature for the response is a hash of the following fields:

1. **MerchantKey** (Provided by iPay88 OPSG and share between iPay88 and merchant only)
2. **MerchantCode**
3. **PaymentId**
4. **RefNo**
5. **Amount**
6. **Currency**
7. **Status**

The fields must be set in the following order,

(MerchantKey & MerchantCode & PaymentId & RefNo & Amount & Currency & Status)

For Example:

MerchantKey = “appleKEY”
MerchantCode = “M09999”
PaymentId = “2”
RefNo = “ORD1188”
Amount = “1.00” (**Note**: Remove the “**.**” and “**,**” in the string before hash)
Currency = “MYR”
Status = “1”

The hash would be calculated on the following string:

appleKEYM099992ORD1188100MYR1

The resulting has signature value equals to (using SHA256 algorithm)

c5cb0a17411ecd341e1c463b0443722452142a828fd94fc55074ebbd40c8d71f

To ensure the signature generated was correct, visit the link below for signature comparison.

https://payment.ipay88.com.my/epayment/testing/testsignature_response_256.asp

2.8.1 Sample function code to generate iPay88 Signature

Using .NET you can use the available libraries to perform this:

```
Public Shared Function GenerateSHA256String(ByVal inputString) As String
    Dim sha256 As SHA256 = SHA256Managed.Create()
    Dim bytes As Byte() = Encoding.UTF8.GetBytes(inputString)
    Dim hash As Byte() = sha256.ComputeHash(bytes)

    Dim stringBuilder As New StringBuilder()

    For i As Integer = 0 To hash.Length - 1
    stringBuilder.Append(hash(i).ToString("X2")) Next

    Return stringBuilder.ToString().ToLower

End Function
```

Using PHP you can use the following code to perform this:

```
<?PHP function
iPay88_signature($source)
{
       return hash('sha256', $source);
}
?>
```

2.9 Appdeeplink Integration

This appdeeplink parameter is for merchant mobile app to mobile app link only and to handle payment method like TNG eWallet, Boost Wallet and Shopeepay.
Once the user completes the payment flow, user will redirect from eWallet app back to merchant mobile app by calling merchant mobile app deeplink, do note that when redirect back to merchant mobile app there no response data will return.
Upon return to merchant app deeplink URL, use requery to get payment status (iPay88 will also return success status to merchants BackendURL and then merchants can update the order to Pai

3 Appendix I

3.1 PaymentId

If PaymentId not post via request, gateway will choose the predefined default payment method. By default, payment method can be re-select by customer from iPay88 payment.

All payment method below is for MYR currency ONLY.

Online Banking Payment Method	PaymentId
FPX TPA	16
Maybank2U	6
Alliance Online (Personal)	8
AmBank	10
RHB Bank	14
Hong Leong Bank	15
CIMB Clicks	20
Public Bank	31
Bank Rakyat	102
Affin Bank	103
Pay4Me (Delay payment)	122
BSN	124
Bank Islam	134
UOB Bank	152
Bank Muamalat	166
OCBC Bank	167
Standard Chartered Bank	168
Maybank2E	178
HSBC Bank	198
Kuwait Finance House	199
Agro Bank	405
Bank of China	1261
China UnionPay Online Banking (MYR)	18

Credit Card Payment Method	PaymentId
Credit Card (MYR)	2
Credit Card (MYR) Pre-Auth	55
Public Bank EPP (Instalment Payment)	111
Maybank EzyPay (Visa/Mastercard Instalment Payment)	112
Maybank EzyPay (AMEX Instalment Payment)	115
HSBC (Instalment Payment)	157
CIMB Easy Pay (Instalment Payment)	174
Hong Leong Bank EPP (Instalment Payment)	179
OCBC Instalment	430
RHB (Instalment Payment)	534
Ambank EPP	606
Standard Chartered Bank Instalment	727

Ewallet	PaymentId
Boost Wallet	210
MCash	244
NETS QR Online	382
GrabPay Online	523
Touch 'n Go eWallet	538
Maybank PayQR Online	542
ShopeePay Online	801

Extra Note:

- For Pay4ME and CIMB Virtual Account payment, iPay88 will initially return Status=6 (Pending Payment) to merchant as this payment method are not real-time payment.
- Once user make payment, iPay88 will return the Status=1 (Success) to merchant BackendURL thus merchant need to ensure had implement BackendURL.
- Generated CIMB Virtual Account number is valid for 7 days and user should make payment within these 7 days.

3.2 Currency

Currency Description	Currency
Malaysian Ringgit	MYR

3.3 Error Description

Error Message	Error Description
Duplicate reference number	Reference number must be unique for each transaction.
Invalid merchant	The merchant code does not exist.
Invalid parameters	Some parameter posted to iPay88 is invalid or empty
Overlimit per transaction	You exceed the amount value per transaction. * For Testing account, only amount MYR 1.00 is allowed.
Payment not allowed	The Payment method you requested is not allowed for this merchant code, please contact ipay88 to enable your payment option.
Permission not allow	Referrer URL in for your account registered in Ipay88 does not match. Please register your request and response URL with iPay88.
Signature not match	The Signature generated is incorrect.
Status not approved	Account was suspended or not active.