



## Mobile SDK Specification (iOS)

---

Merchant Integration Guide

Version 1.0

**Important:** Please note that this document is strictly private and confidential and is not intended for public circulation.

Last updated 14 February 2017

1. Introduction -----	2
1.1. Prerequisite-----	2
1.2. System Requirements -----	2
2. SDK Integration -----	3
2.1. Configure Xcode project -----	3
2.2. Configure SDK -----	4
• Import Library .....	4
• Setup delegate. ....	4
• Prepare request details. ....	4
• Allocate eGHLPayment class.....	5
• Load the SDK. ....	5
• Possible Error code & Error Data. ....	5
• Possible Status Data. ....	5
• Detecting user exiting / canceling the payment process. ....	6
3. Appendix -----	7
3.1. Response format for payment request -----	7
3.2. Response format for query request -----	11
3.3. Payment/Capture Transaction Status-----	11
3.4. Query Transaction Status-----	12
3.5. Currency Code -----	12
3.6. Language Code -----	12
3.7. Language Code -----	13

# 1. Introduction

Mobile Software Development Kit (SDK) allows integration with Merchant mobile application that would like to accept online payment by 3D and non-3D credit card, direct debit and e-Wallet payments.

This Merchant Integration Guide provides merchants with the necessary technical information to integrate their applications (Merchant Systems) with Payment Gateway.

The manual contains message format required between Payment Gateway and Merchant System for various payment transaction types, namely Payment, Query and Reversal. It is intended as a technical guide for merchant developers and system integrators who are responsible for designing or programming the respective applications to integrate with Payment Gateway.

## 1.1. Prerequisite

All merchants who would like to integrate with Payment Gateway must obtain a valid payment account from eGHL. Upon payment account generated, eGHL will provide merchant a Service ID and Merchant Password.

## 1.2. System Requirements

- Ready for iOS 8 and above
- Minimum iOS version 6 and above. Older version support can be provided on request for custom make
- iOS based devices that support iOS version 8.0

## 2. SDK Integration

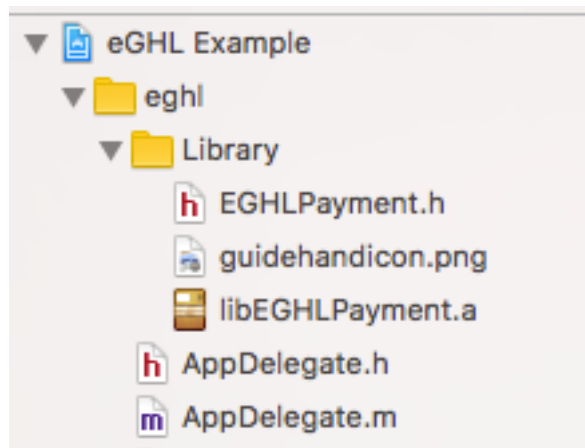
This section explains details on eGHL SDK integration. SDK is a library that will be added to the Xcode project.

The followings are the process of integration:

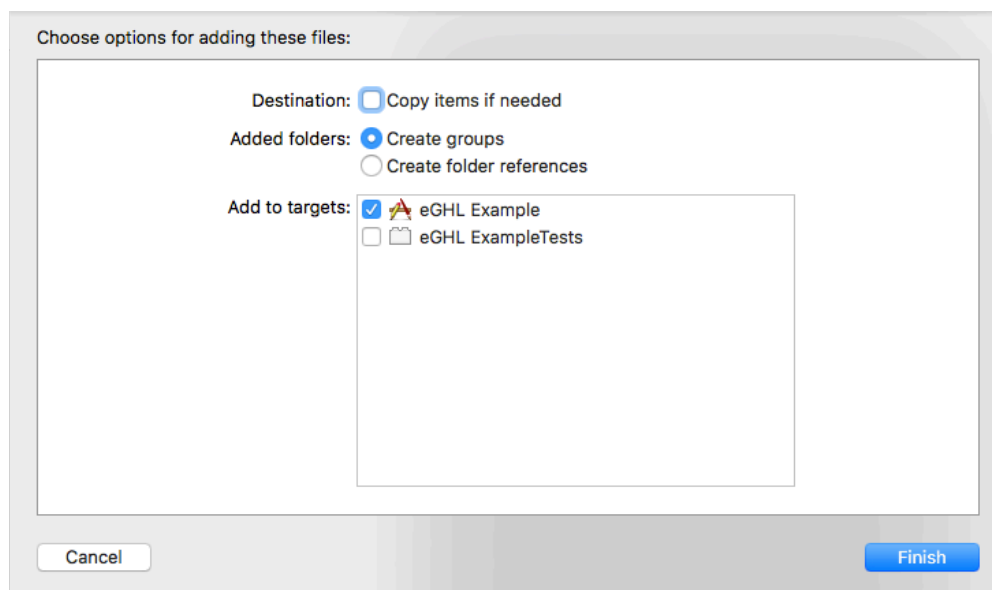
- 1) Configure Xcode project (Adding the library, header file(s) & images).
- 2) Configure SDK.
- 3) Merchant app necessary information to the SDK
- 4) SDK will handle the payment process.
- 5) After payment is completed, SDK will return the result for transaction via delegate.

### 2.1. Configure Xcode project

- Drag & drop Library Folder / all the files within it to your Xcode project.



- Uncheck the box "Copy items if needed" then press finish button on bottom right. This will reduce space in your App.



## 2.2. Configure SDK

- **Import Library**

Import SDK Public header.

```
#import "EGHLPayment.h"
```

- **Setup delegate.**

Add delegate at your *based view controller*\* header(.h)/implementation(.m) file.

```
@interface ViewController () <EGHLDelegate>
```

\* In eGHL example, *based view controller* is referring to ShowViewController

- **Prepare request details.**

Add delegate at your *based view controller*\* header(.h)/implementation(.m) file.

A. Payment request

```
PaymentRequestPARAM * payparam = [[PaymentRequestPARAM alloc] init]

payparam.Amount = ... (Type: NSString)
payparam.MerchantName = ... (Type: NSString)
payparam.CustEmail = ... (Type: NSString)
payparam.CustName = ... (Type: NSString)
payparam.ServiceID = ... (Type: NSString)
payparam.Password = ... (Type: NSString)
payparam.CurrencyCode = ... (Type: NSString)

payparam.PaymentID = ... (Type: NSString)
payparam.OrderNumber = ... (Type: NSString)

payparam.MerchantReturnURL = ... (Type: NSString)
payparam.MerchantCallbackURL = ... (Type: NSString)

payparam.CustPhone = ... (Type: NSString)
payparam.LanguageCode = ... (Type: NSString)
payparam.PageTimeout = ... (Type: NSString)
payparam.PaymentDesc = ... (Type: NSString)

payparam.IssuingBank = ... (Type: NSString)
payparam.PymtMethod = ... (Type: NSString)

payparam.RealHost = ... (Type: NSString)
```

B. Query request

```
PaymentRequestPARAM * payparam = [[PaymentRequestPARAM alloc] init]

payparam.PymtMethod = ... (Type: NSString)
payparam.ServiceID = ... (Type: NSString)
payparam.PaymentID = ... (Type: NSString)
payparam.Amount = ... (Type: NSString)
payparam.CurrencyCode = ... (Type: NSString)
```

- **Allocate eGHLPayment class.**

```
EGHLPayment * eghlpay = [[EGHLPayment alloc] init]
```

- **Load the SDK.**

- A. Payment request

```
[eghlpay EGHLRequestSale:payparam successBlock:^(NSString* SuccessData) {  
    NSLog(@"SuccessData:%@", SuccessData);  
} failedBlock:^(NSString* errorCode, NSString* errorData) {  
    NSLog(@"errorcode:%@ errordata:%@", errorCode, errorData);  
}];
```

- Payment request result

To get result for each payment request, you have to add the following delegate function in base view controller

- (void)QueryResult:(PaymentRespPARAM \*)result

- }

- \* Please refer *PaymentRespPARAM* class for possible fields

- \* Please refer appendix for fields specification

- B. Query request

```
[eghlpay EGHLRequestQuery:payparam successBlock:^(PaymentRespPARAM* ParamData) {  
    } failedBlock:^(NSString* errorCode, NSString* errorData) {  
    }  
}];
```

- **Possible Error code & Error Data.**

Error code	Error Data
-1009	The Internet connection appears to be offline.
-1001	The request timed out.
-1	ParametersEmpty

- **Possible Status Data.**

Success Data
LoadSuccess

- **Detecting user exiting / canceling the payment process.**

In based view controller (in GHL example would be ShowViewController)

```
- (void)viewDidLoad {
    [super viewDidLoad];

    UIButton * button = [[UIButton alloc] initWithFrame:CGRectMake(0,0,100,100)];
    [button setImage:[UIImage imageNamed:@"navigationItemBack"]
    forState:UIControlStateNormal];
    [button setTitleColor:[UIColor colorWithRed:0.04 green:0.38 blue:1 alpha:1.0]
    forState:UIControlStateNormal];

    [button addTarget:self action:@selector(backButtonPressed:)
    forControlEvents:UIControlEventTouchUpInside];
    [button setClipsToBounds:YES];

    [button setTitle:@"Back" forState:UIControlStateNormal];
    [button sizeToFit];

    [button.titleLabel setFont:[UIFont boldSystemFontOfSize:17]];

    self.navigationItem.leftBarButtonItem = [[UIBarButtonItem alloc] initWithCustomView:button];
    self.navigationController.interactivePopGestureRecognizer.delegate =
    (id<UIGestureRecognizerDelegate>)self;
}

- (void)backButtonPressed:(id)sender{
    UIAlertView * alertExit = [[UIAlertView alloc] initWithTitle:@"Are you sure you want to quit"
    message:@"Pressing BACK button will close and abandon the payment session." delegate:self
    cancelButtonTitle:@"Cancel" otherButtonTitles:@"Exit", nil];

    [alertExit setTag:100];
    [alertExit show];
}

#pragma mark - UIAlertView delegate

- (void)alertView:(UIAlertView *)alertView clickedButtonAtIndex:(NSInteger)buttonIndex
{
    switch ([alertView tag]) {
        case 100:
        {
            if (buttonIndex == 1) {
                [self.navigationController popViewControllerAnimated:YES];
            }
        }
        break;

        default:
            break;
    }
}
```

### 3. Appendix

#### 3.1. Response format for payment request

No.	Field	Data Type	Max Length	Req?	Description
1.	TransactionType	A	7	Y	Follows request
2.	PymtMethod	A	3	Y	Payment Method <b>CC</b> – Credit Card (Online 3D/Non3D) <b>MO</b> – Credit Card (MOTO – Mail Order Telephone Order) <b>DD</b> – Direct Debit <b>WA</b> – e-Wallet <b>CC</b> – Credit Card
3.	ServiceID	AN	3	Y	Follows request
4.	PaymentID	AN	20	Y	Follows request
5.	OrderNumber	AN	20	Y	Follows request
6.	Amount	N	12(2)	Y	Follows request for transaction not entitled for promotion  For transaction entitled for promotion, Amount will not be the same as the Amount provided in Payment Request. Amount will be the actual payment amount which is the final amount after deducting promotion discount. The original Amount will be available in PromoOriAmt
7.	CurrencyCode	N	3	Y	Follows request
8.	HashValue	AN	100	Y	Message digest value calculated by Payment Gateway in <b>hexadecimal</b> string using <b>SHA256</b> hash algorithm  Re: Section <a href="#">2.13.1.2</a>



9.	HashValue2	AN	100	Y	<b>HIGHLY RECOMMENDED</b> to verify this message digest value calculated by Payment Gateway in <b>hexadecimal</b> string using <b>SHA256</b> hash algorithm  Re: Section <a href="#">2.13.1.2</a>
10.	TxnID	AN	30	Y	Unique Transaction ID or Reference Code assigned by Payment Gateway for this transaction
11.	IssuingBank	AN	30	Y	Follows request if this field is provided in request If not provided in request, For PymtMethod "CC", this field is the Bank Name keyed in by customer on eGHL Payment Info Collection Page For PymtMethod "DD", this field is the Bank Name chosen by customer to perform Direct Debit transaction
12.	TxnStatus	N	4	Y	Re: Section <a href="#">3.2</a>
13.	AuthCode	AN	12	N	Authorization Code returned by bank
14.	TxnMessage	AN	255	N	Message that briefly explains the response
15.	TokenType	A	3	N	Token Type  If merchant is subscribed to eGHL One-Click Payment feature, upon payment approved, TokenType will be " <b>OCP</b> ", together with token value in <b>Token</b> field
16.	Token	ANS	50	C	Token Value  If Token Type is " <b>OCP</b> ", Token field will hold the Token Value for One-Click Payment purposes
17.	Param6	ANS	50	C	Follows request
18.	Param7	ANS	50	C	Follows request

19.	CardHolder	AN	30	C	Cardholder's Name Only available as per request
20.	CardNoMask	AN	19	C	Credit Card Number used for payment authorization, first 6 and last 4 digits are in clear, the rests are masked with "X". e.g. 444433XXXXXX1111 Only available as per request
21.	CardExp	N	6	C	Expiry date of credit card. Date format is <b>YYYYMM</b> , e.g. 201312 for year 2013 December Only available as per request
22.	CardType	A	10	C	Credit Card Type e.g. VISA/MASTERCARD/AMEX/ JCB/DINERS Only available as per request
23.	SettleTAID	N	10	C	Terminal Account ID identifying the respective TID used to process the transaction and to be used for TxnType SETTLE. Only available for MOTO transaction (PymtMethod MO)
24.	TID	N	8	C	Actual terminal ID assigned by the bank to perform the transaction Only available for MOTO transaction (PymtMethod MO)
25.	EPPMonth	N	2	C	Follows request
26.	EPP_YN	N	1	C	Identifier for installment entitlement <b>0</b> – Not entitled for installment <b>1</b> – Entitled for installment

27.	PromoCode	AN	10	C	<p>Promotion Code</p> <p>Follows request if PromoCode is provided in Payment Request</p> <p>If PromoCode is not provided in Payment Request but the transaction is entitled for promotion, PromoCode will be available in Payment Response</p>
28.	PromoOriAmt	N	12(2)	C	<p>Follows request's Amount for transaction entitled for promotion</p> <p>Original amount before deducting promotion discount. Only available for transaction entitled for promotion</p>

### 3.2. Response format for query request

No.	Field	Data Type	Max Length	Req?	Description
1.	TxnExists	N	1	Y	<p>An identifier to indicate whether the transaction being queried exists in Payment Gateway.</p> <p><b>0 –</b> Transaction being queried <b>exists</b> in Payment Gateway. Merchant System can proceed to refer to the rest of other fields for details, e.g. TxnStatus</p> <p><b>1 –</b> Transaction being queried <b>does not exist</b> in Payment Gateway. In other words, Payment Gateway is not able to find any transaction that is matched with all Query request fields submitted by merchant system <b>TxnStatus</b> will be <b>-1</b></p> <p><b>2 –</b> There was some kind of internal <b>error</b> occurred during query processing. Merchant System can retry sending query request <b>TxnStatus</b> will be <b>-2</b></p>
2.	QueryDesc	AN	255	N	Description of query result

### 3.3. Payment/Capture Transaction Status

TxnStatus	Description
0	Transaction successful
1	Transaction failed
2	Transaction pending

### 3.4. Query Transaction Status

TxnStatus	Description
0	Transaction successful (for transaction type SALE)
1	Transaction failed
2	Sale pending, retry Query
10	Transaction refunded
15	Transaction authorized (for transaction type AUTH)
16	Transaction captured
31	Reversal pending, merchant system can retry Reversal if merchant system initiated the Reversal request or else merchant system can retry Query
9	Transaction reversed
-1	Transaction not exists/not found
-2	Internal system error

### 3.5. Currency Code

Currency Code	Currency
MYR	Malaysia Ringgit
SGD	Singapore Dollar
THB	Thai Baht
CNY	China Yuan (Ren Min Bi)
PHP	Philippine Peso

### 3.6. Language Code

Language Code	Language
EN	English (default language)
BM	Malaysia
TH	Thai
CN	Chinese

### 3.7. Language Code

Country Code	Country
MY	Malaysia
TH	Thailand
SG	Singapore
PH	Philippines
CN	China
ID	Indonesia