PAYMENTS I SIMPLIFIED

Internet Payment Gateway

Merchant Integration Guide
Version 2.8p

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

TABLE OF CONTENTS

Document Version History

This section details the changes that were made to this document.
The first author is the creator of this document.

#	Version	Date	Author	Description

1.	2.1	22 Aug 2013	eGHL	• Added the following mandatory fields:
				  • CustName
				  • CustEmail
				  • CustPhone
				• Added the following non-mandatory field:
				  • CustMAC
				• Modified the following conditional field to be mandatory:
				  • CustIP

2.	2.2	4 Sept 2013	eGHL	• Added Reversal Pending status (31) in section 3.3 (Query Transaction Status)
3.	2.3	6 Jun 2014	eGHL	• Added MerchantApprovalURL and MerchantUnApprovalURL in section 2.1 (Payment Request) • Added MerchantApprovalURL and MerchantUnApprovalURL in section 2.7.1.1 (Payment Request Hashing)
4.	2.3	9 Jul 2014	eGHL	• Added SessionID in section 2.1 (Payment Request)
5.	2.4	17 Jul 2014	eGHL	• Added TokenType and Token in section 2.1 (Payment Request) and section 2.2 (Payment Response)
6.	2.5	16 Oct 2014	eGHL	• Added MerchantCallBackURL in section 2.1 (Payment Request) and section 2.7.1.1 (Payment Request Hashing)
7.	2.6	26 Nov 2014	eGHL	• Added B4TaxAmt and TaxAmt in section 2.1 (Payment Request)
8.	2.7	6 Apr 2015	eGHL	

				• Added transaction type AUTH in TransactionType field of section 2.1 (Payment Request) • Added transaction type CAPTURE in section 2.5 (Capture Request) • Added Param6 and Param7 in section 2.1 (Payment Request) and section 2.2 (Payment Response) • Added AUTH and CAPTURE in section 3.1 (Transaction Type) • Added status 15 for Auth, 16 for Captured in section 3.3 (Query Transaction Status)
9.	2.8	25 May 2015	eGHL	• Added transaction type REFUND in section 2.9 (Refund Request) • Added status 10 for Refunded in section 3.3 (Query Transaction Status)
10.	2.8a	5 Oct 2015	eGHL	• Added CardNoMask, CardExp, CardHolder and CardType in section 2.2 (Payment Response)
11.	2.8b	2 Nov 2015	eGHL	• Added "MO" as one of the available options for PymtMethod parameter in section 2.1 (Payment Request) to identify Mail Order Telephone Order Credit Card payment method
12.	2.8c	13 Jan 2016	eGHL	• Added a NOTE for Token value format to be put in hash calculation on section 2.13.1.1
13.	2.8d	17 Feb 2016	eGHL	• Updated the NOTE for Token value format to be put in hash calculation on section 2.13.1.1
14.	2.8e	2 May 2016	eGHL	• Added SettleTAID and TID field in section 2.2 (Payment Response) • Added transaction type SETTLE in section 2.11 (Settlement Request) and section 2.12 (Settlement Response) • Added Settle Request Hashing in section 2.13.1.4 • Added Settle Response Hashing in section 2.13.1.5 • Added SETTLE in section 3.1 (Transaction Type) • Added Settlement Transaction Status in section 3.5
15.	2.8f	16 May 2016	eGHL	

				• Added HashValue2 in section 2.2 (Payment Response) • Added HashValue2 in section 2.13.1.2 • Added EPPMonth in section 2.1 (Payment Request) and section 2.2 (Payment Response) • Added EPP_YN in section 2.2 (Payment Response)
16.	2.8g	27 Jun 2016	eGHL	• Added PromoCode in section 2.1 (Payment Request) • Added PromoCode and PromoOriAmt in section 2.2 (Payment Response) • Added a note for Amount in section 2.2 (Payment Response) related to promotion
17.	2.8h	5 Dec 2016	eGHL	• Added note for Production environment only allows TLS 1.2 communication protocol from incoming server-to-server integration for Query, Reversal, Capture, Refund and Settlement • Added TxnStatus 2 for Pending in Appendix section 3.2 (Payment/Capture Transaction Status)
18.	2.8i	16 Jan 2017	eGHL	• Masterpass Express Checkout Integration (MPE)
19.	2.8i	26 Feb 2017	eGHL	• Added ReqToken, PairingToken in 2nd MPE request message • Change 1st MPE response message Express Checkout.
20.	2.8j	4 Apr 2017	eGHL	• Masterpass Standard Checkout Integration (MSC) Note: Only for merchant have Masterpass Express Checkout and want to skip eGHL Payment Page
21.	2.8k	6 May 2017	eGHL	• Added new TokenType N3D
22.	2.8l	3 Aug 2017	eGHL	• Added PymtMethod "OTC" in section 2.1 (Payment Request) • Appendix 3.7 language code change based on ISO 639-1 code
23.	2.8m	2 Oct 2017	eGHL	• Added ECI, CAVV, 3DXID fields in section 2.1 (Payment Request) • Added sample of server-to-server payment request message in section 2.1 (Payment Request) • Added sample of server-to-server payment response message in section 2.2 (Payment Response)
24.	2.8n	30 Oct 2017	eGHL	

				• Added TokenType value SOP and Token SOP description in section 2.1 (Payment Request) • Added section 2.13.1.6 Optimize/SOP Hash
25.	2.8o	24 Oct 2018	eGHL	• Added section 2.16 FPX e-Mandate

1 Introduction

Payment Gateway provides a web interface that allows integration with Merchant System which 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 online applications to integrate with Payment Gateway.

1.1 Pre-requisite

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.

2 Payment API Reference Fields (HTTP GET and POST)

2.1 Payment Request (Merchant System → Payment Gateway)

The following fields are the Payment information expected from Merchant System to Payment Gateway in order to perform an online payment transaction:

No.	Field	Data Type	Max Length	Req?	Description
1.	TransactionType	A	7	Y	SALE – Direct captured for credit card payment; Payment request for other payment methods AUTH – For credit card payment, authorize the availability of funds for a transaction but delay the capture of funds until a later time. This is often useful for merchants who have a delayed order fulfillment process. Authorize & Capture also enables merchants to modify the original authorization amount due to order changes occurring after the initial order is placed, such as taxes, shipping or item availability Re: Section 2.5 for Transaction Type CAPTURE message format
2.	PymtMethod	A	3	Y	Payment Method CC – Credit Card (Online 3D/Non3D) MO – Credit Card (MOTO – Mail Order Telephone Order) DD – Direct Debit (not applicable to TransactionType AUTH) WA – e-Wallet (not applicable to Transaction Type AUTH) OTC – Over The Counter (not applicable to TransactionType AUTH) ANY – All payment method(s) registered with eGHL

3.	ServiceID	AN	3	Y	Merchant Service ID given by eGHL
4.	PaymentID	AN	20	Y	Unique transaction ID/reference code assigned by merchant for this payment transaction (No duplicate PaymentID is allowed)
5.	OrderNumber	AN	20	Y	Reference number / Invoice number for this order PaymentID must be unique but OrderNumber can be the same under different PaymentID, indicating multiple payment attempts are made on a particular order If Order Number is not applicable, please provide the same value as PaymentID
6.	PaymentDesc	AN	100	Y	Order's descriptions
7.	MerchantReturnURL	AN	255	Y	Merchant system's browser redirect URL which receives payment response from eGHL when transaction is completed (approved/declined/system error/ cancelled by buyer on eGHL Payment Page) If MerchantApprovalURL is provided, when payment is approved, MerchantApprovalURL will be used instead of MerchantReturnURL

					If MerchantUnApprovalURL is provided, when payment is declined, MerchantUnApprovalURL will be used instead of MerchantReturnURL For server-to-server integration for non-3D payment transaction, please provide value as s2s. Note: Replace "&" with ";" if any. e.g. https://merchantdomain/index.php?field1=value1&field2=value2 to https://merchantdomain/index.php?field1=value1;field2=value2
8.	Amount	N	12(2)	Y	Payment amount in 2 decimal places regardless whether the currency has decimal places or not. Please exclude "," sign. e.g. 1000.00 for IDR Invalid format: 1,000.00 or 1000
9.	CurrencyCode	A	3	Y	3-letter ISO4217 of Payment Currency Code Re: Section 3.6
10.	HashValue	AN	100	Y	Message digest value calculated by Merchant System in hexadecimal string using SHA256 hash algorithm Re: Section 2.7.1.1
11.	CustIP	AN	20	Y	Customer's IP address captured by merchant system

12.	CustName	AN	50	Y	Customer Name
13.	CustEmail	AN	60	Y	Customer's Email Address
14.	CustPhone	AN	25	Y	Customer's Contact Number
15.	B4TaxAmt	N	12(2)	N	Original amount before tax is incurred, in 2 decimal places regardless whether the currency has decimal places or not. As for final payment amount after tax is incurred, is to be specified in Amount field Please exclude "," sign. e.g. 1000.00 for IDR Invalid format: 1,000.00 or 1000
16.	TaxAmt	N	12(2)	N	Tax amount incurred Please exclude "," sign. e.g. 1000.00 for IDR Invalid format: 1,000.00 or 1000
17.	MerchantName	AN	25	N	Merchant's business name
18.	CustMAC	AN	50	N	Machine ID (MAC Address) of customer's computer/device which was used to make payment
19.	MerchantApprovalURL	AN	255	N	URL to link to merchant's website when payment is approved

					If not provided, MerchantReturnURL will be used Note: Replace "&" with ";" if any. e.g. https://merchantdomain/index.php?field1=value1&field2=value2 to https://merchantdomain/index.php?field1=value1;field2=value2
20.	MerchantUnApprovalURL	AN	255	N	URL to link to merchant's website when payment is declined If not provided, MerchantReturnURL will be used Note: Replace "&" with ";" if any. e.g. https://merchantdomain/index.php?field1=value1&field2=value2 to https://merchantdomain/index.php?field1=value1;field2=value2
21.	MerchantCallBackURL	AN	255	N	Server-to-server URL as an additional link to merchant's website to be informed of transaction status This is useful when browser redirect URLs (MerchantReturnURL/MerchantApprovalURL/MerchantUnApprovalURL) were not able to receive payment response due to buyer's Internet

					connectivity problem or buyer closed browser Upon receiving response from Gateway, MerchantCallBackURL is to return an acknowledgement message "OK" to the Gateway or else Gateway will continue to send response to this URL for a maximum of 3 times Note: Replace "&" with ";" if any. e.g. https://merchantdomain/index.php?field1=value1&field2=value2 to https://merchantdomain/index.php?field1=value1;field2=value2
22.	LanguageCode	A	2	N	ISO 639-1 language Code for eGHL Payment Info Collection Page Re: Section 3.7
23.	PageTimeout	N	4	N	eGHL Payment Info Collection Page timeout in seconds Applicable for merchant system which would like to bring forward to Payment Gateway, the time remaining before product/order is released For example, a movie ticket sales page shows time remaining countdown from 15 minutes till 5 minutes. Upon customer's clicking "checkout / proceed / pay" button, merchant system can then pass the value of (5 minutes x 60 seconds=300) seconds in this field to Gateway which will then continue the countdown from 5 minutes. Upon timeout, all entry fields

					and buttons on the Collection Page will be disabled
24.	CardHolder	AN	30	N	Cardholder's Name For PymtMethod "CC", if not provided, Payment Gateway will prompt this field on eGHL Payment Info Collection Page
25.	CardNo	N	19	N	Credit Card Number used for payment authorization For PymtMethod "CC", if not provided, Payment Gateway will prompt this field on eGHL Payment Info Collection Page If merchant has own payment page to collect card information, merchant system is required to be PCI compliant
26.	CardExp	N	6	N	Expiry date of credit card. Date format is YYYYMM, e.g. 201312 for year 2013 December For PymtMethod "CC", if not provided, Payment Gateway will prompt this field on eGHL Payment Info Collection Page
27.	CardCVV2	N	4	N	3-4 digits Card Verification Value available on the back of credit card For PymtMethod "CC", if not provided, Payment Gateway will prompt this field on eGHL Payment Info Collection Page
28.	IssuingBank	AN	30	N	For PymtMethod "CC", this field indicates Bank which issued the credit card used for this transaction

					If not provided, Payment Gateway will prompt this field on eGHL Payment Info Collection Page For PymtMethod "DD", this field indicates Direct Debit banks/payment switches. If not provided, the list of Direct Debit/payment switches supported will be shown on eGHL Payment Info Collection Page
29.	BillAddr	AN	100	N	Billing Address (excludes postcode, town/city, state and country) For PymtMethod "CC", if not provided, Payment Gateway will prompt this field on eGHL Payment Info Collection Page
30.	BillPostal	AN	15	N	Billing Postcode For PymtMethod "CC", if not provided, Payment Gateway will prompt this field on eGHL Payment Info Collection Page
31.	BillCity	A	30	N	Billing Town/City For PymtMethod "CC", if not provided, Payment Gateway will prompt this field on eGHL Payment Info Collection Page
32.	BillRegion	A	30	N	Billing Region/State For PymtMethod "CC", if not provided, Payment Gateway will prompt this field on eGHL Payment Info Collection Page
33.	BillCountry	A	2	N	Billing Country Code Re: Section 3.8

					For PymtMethod "CC", if not provided, Payment Gateway will prompt this field on eGHL Payment Info Collection Page
34.	ShipAddr	AN	100	N	Shipping Address (excludes postcode, town/city, state and country)
35.	ShipPostal	AN	15	N	Shipping Postcode
36.	ShipCity	A	30	N	Shipping Town/City
37.	ShipRegion	A	30	N	Shipping Region/State
38.	ShipCountry	A	2	N	Shipping Country Code Re: Section 3.8
39.	SessionID	AN	100	N	Session ID
40.	TokenType	A	3	N	Token Type OCP – One-click Payment MPE – MasterPass Express Checkout N3D – For Non-3D transaction SOP – For Optimize/SOP Payment
41.	Token	ANS	50	C	Token Value If TokenType is specified, Token is expected to have value For TokenType OCP, this filed is expecting the token value returned by eGHL in payment response For TokenType MPE, this field is expecting consumer's Login ID to the merchant system For TokenType N3D, the request will bypass eGHL payment page. Card details will be fetched automatically.

					Account from acquirer is CVV disabled. For TokenType SOP, this field is expecting one-time token value returned by eGHL-SOP library.
42.	Param6	ANS	50	N	Additional data from merchant system that will be passed back to merchant in payment response
43.	Param7	ANS	50	N	Additional data from merchant system that will be passed back to merchant in payment response
44.	EPPMonth	N	2	N	Number of months for the installment
45.	PromoCode	AN	10	N	Promotion Code registered in eGHL. If provided, the transaction will be entitled for the specific promotion
46.	ECI	AN	02	C	ECI is 3D Secure Field. It is a value that is returned from the Directory Server to indicate the authentication result. This field is for server-to-server integration and merchant who has own MPI.
47.	CAVV	ANS	28	C	CAVV is 3D Secure Field. The format is 28-char Base64 encoded string. e.g. AAABA2dGFgAAAAABEUYWAAAAAA= If non 3D Secure transaction, left the field to be blank.

					This field is for server-to-server integration and merchant who has own MPI.
48.	3DXID	ANS	28	C	XID is 3D Secure Field. The format is 28-char Base64 encoded string. e.g. ejU4ZlB0Q2NmTUpQdndtdGxHWDA= If non 3D Secure transaction, left the field to be blank. This field is for server-to-server integration and merchant who has own MPI.
49	RecurringCriteria	ANS	40	C	Recurring Information: Frequency|Duration|RecurringStartDt|RecurringAmount (Please use pipe (|) as a separator) - Frequency : - The type of frequency for debiting from the payer's account. - Option available are as below : D – Day M – Month Y - Year (Please indicate the number of Day/Month/Year eg; 3D) - Duration : - The duration for debiting from the payer's account. - Option available are as below : D – Day M – Month Y – Year (Please indicate the number of

					Day/Month/Year eg; 3D) - RecurringStartDt : - The effective date for the first debiting from payer's account. - YYYYMMDD - RecurringAmount : - The amount that can be debiting from the payer's account. - Payment amount in 2 decimal places regardless whether the currency has decimal places or not. Please exclude "," sign. e.g. 1000.00 for IDR Invalid format: 1,000.00 or 1000 Please refer to below example; - **5D|12m|20181025|1.00** Does not support: Slash(/) Ampersand(&) Apostrophe(')

Req? – Required? (Mandatory fields?)
Y – Yes
N – No
C – Conditional

Sample HTML Form Post Payment Request

```
<form name="frmPayment" method="post" action="https://<URL to be provided by eGHL>">
<input type="hidden" name="TransactionType" value="SALE">
<input type="hidden" name="PymtMethod" value="ANY">
<input type="hidden" name="ServiceID" value="A00">
<input type="hidden" name="PaymentID" value="ABCDEFGH130820142128">
<input type="hidden" name="OrderNumber" value="IJKLMN">
<input type="hidden" name="PaymentDesc" value="Booking No: IJKLMN, Sector: KUL-BKI,
First Flight Date: 26 Sep 2012">
<input type="hidden" name="MerchantName" value="Merchant A">
<input type="hidden" name="MerchantReturnURL"
value="https://merchA.merchdomain.com/pymtresp.aspx">
<input type="hidden" name="MerchantCallbackURL"
value="https://merchA.merchdomain.com/pymtrespcallback.aspx">
<input type="hidden" name="Amount" value="228.00">
<input type="hidden" name="CurrencyCode" value="MYR">
<input type="hidden" name="CustIP" value="192.168.2.35">
<input type="hidden" name="CustName" value="Jason">
<input type="hidden" name="CustEmail" value="Jasonabc@gmail.com">
<input type="hidden" name="CustPhone" value="60121235678">
<input type="hidden" name="HashValue" value="hash value generated">
<input type="hidden" name="MerchantTermsURL"
value="http://merchA.merchdomain.com/terms.html">
<input type="hidden" name="LanguageCode" value="en">
<input type="hidden" name="PageTimeout" value="780">
</form>
```

Sample of Server-to-server Payment Request

```
TransactionType=SALE&PymtMethod=CC&ServiceID=FY&PaymentID=A3BHPF20171001018074
&OrderNumber=A3BHPF&PaymentDesc=-
&MerchantReturnURL=S2S&Amount=299.48&CurrencyCode=MYR&HashValue=D8374858C6D7D
EAF12E1EDA44859C01AAF189197314CC662ACF0B0BB2D742D76&CustIP=-&CustName=-
&CustEmail=-&CustPhone=-&Cardholder=TESTER
&CardNo=379186123459794&CardExp=202012&CardCVV2=1234&
ECI=05&CAVV=AAABA2dGFgAAAAABEUYWAAAAAAA=&
3DXID=ejU4ZlB0Q2NmTUpQdndtdGxHWDA=
```

2.2 Payment Response (Payment Gateway → Merchant System)

Upon payment process completion, the following fields will be returned from Payment Gateway to Merchant System's MerchantReturnURL in order to complete an end-to-end payment process:

No.	Field	Data Type	Max Length	Req?	Description
1.	TransactionType	A	7	Y	Follows request
2.	PymtMethod	A	3	Y	Payment Method CC – Credit Card (Online 3D/Non3D) MO – Credit Card (MOTO – Mail Order Telephone Order) DD – Direct Debit WA – e-Wallet CC – 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 hexadecimal string using SHA256 hash algorithm Re: Section 2.13.1.2
9.	HashValue2	AN	100	Y	HIGHLY RECOMMENDED to verify this message digest value calculated by

					Payment Gateway in hexadecimal string using SHA256 hash algorithm Re: Section 2.13.1.2
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 3.2
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 "OCP", together with token value in Token field
16.	Token	ANS	50	C	Token Value If Token Type is "OCP", 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 YYYYMM, 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 0 – Not entitled for installment 1 – Entitled for installment
27.	PromoCode	AN	10	C	Promotion Code

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

Sample VB.net Code to Retrieve Payment Response Fields

```
Imports System.Web

Var1 = Server.UrlDecode(HttpContext.Current.Request("PaymentID")
Var2 = Server.UrlDecode(HttpContext.Current.Request("OrderNumber")
Var3 = Server.UrlDecode(HttpContext.Current.Request("Amount")
Var4 = Server.UrlDecode(HttpContext.Current.Request("CurrencyCode")
Var5 = Server.UrlDecode(HttpContext.Current.Request("TxnID")
Var6 = Server.UrlDecode(HttpContext.Current.Request("PymtMethod")
Var7 = Server.UrlDecode(HttpContext.Current.Request("TxnStatus")
Var8 = Server.UrlDecode(HttpContext.Current.Request("AuthCode")
Var9 = Server.UrlDecode(HttpContext.Current.Request("TxnMessage")
Var10 = Server.UrlDecode(HttpContext.Current.Request("IssuingBank")
Var11 = Server.UrlDecode(HttpContext.Current.Request("HashValue")
Var12 = Server.UrlDecode(HttpContext.Current.Request("HashValue2")
```

Sample of Server-to-server Payment Response

TransactionType=SALE&PymtMethod=CC&ServiceID=FY&PaymentID=A3BHPF20171001018074
&OrderNumber=A3BHPF&Amount=299.48&CurrencyCode=MYR&HashValue=fec00c8be232ee73f
cbccc09da56b8c76db34181b6c63e0bd922e2712127ef38&HashValue2=3305131d44484c8faea9
92550e2f9c746c4071f158e055c1615feaacfd56e573&TxnID=FYA3BHPF20171001018074&Issuin
gBank=MBBAMEX2_3D&TxnStatus=0&AuthCode=256140&BankRefNo=41214869&TxnMessage
=Transaction+Successful

2.3 Query Request (Merchant System → Payment Gateway)

The following fields are the Payment information expected from Merchant System to Payment Gateway in order to query payment status.

No MerchantReturnURL involved. Merchant System can get query response on the same session.

NOTE: The protocol used to communicate with Payment Gateway for test environment is not limited but production environment only allows TLS 1.2 protocol for PCI compliance and security concerns. For Status Inquiry integration, it is recommended to test out the TLS 1.2 connectivity with production environment before going LIVE.

No.	Field	Data Type	Max Length	Req?	Description
1.	TransactionType	A	7	Y	QUERY
2.	PymtMethod	A	3	Y	Payment Method submitted in the original Payment Request being queried
3.	ServiceID	AN	3	Y	Merchant Service ID given by eGHL
4.	PaymentID	AN	20	Y	Unique Transaction ID or Reference Code assigned by Merchant System for the original Payment Request being queried
5.	Amount	N	12(2)	Y	Payment Amount submitted in Payment Request
6.	CurrencyCode	A	3	Y	Payment Currency Code submitted in Payment Request
7.	HashValue	AN	100	Y	Message digest value calculated by Merchant System in hexadecimal string using SHA256 hash algorithm Re: 2.7.1.3

Sample VB.net Code HTTP POST Query Request And Receive Query Response

Reference: http://msdn.microsoft.com/en-us/library/debx8sh9.aspx

```
Imports System
Imports System.IO
Imports System.Net
Imports System.Text
Namespace Examples.System.Net
    Public Class WebRequestPostExample

        Public Shared Sub Main()
            ' Create a request using a URL that can receive a post.
            Dim request As WebRequest = WebRequest.Create("https://<to be provided by eGHL>")
            ' Set the Method property of the request to POST.
            request.Method = "POST"
            ' Create POST data and convert it to a byte array.
            Dim postData As String =
"TransactionType=QUERY&PymtMethod=CC&ServiceID=A07&PaymentID=
ABCDEFGH130820142128&Amount=228.00&CurrencyCode=MYR&HashValue=hash value generated"
            Dim byteArray As Byte() = Encoding.UTF8.GetBytes(postData)
            ' Set the ContentType property of the WebRequest.
            request.ContentType = "application/x-www-form-urlencoded"
            ' Set the ContentLength property of the WebRequest.
            request.ContentLength = byteArray.Length
            ' Get the request stream.
            Dim dataStream As Stream = request.GetRequestStream()
            ' Write the data to the request stream.
            dataStream.Write(byteArray, 0, byteArray.Length)
            ' Close the Stream object.
            dataStream.Close()

            ' Get the response.
            Dim response As WebResponse = request.GetResponse()
            ' Display the status.
            Console.WriteLine(CType(response, HttpWebResponse).StatusDescription)
            ' Get the stream containing content returned by the server.
            dataStream = response.GetResponseStream()
            ' Open the stream using a StreamReader for easy access.
            Dim reader As New StreamReader(dataStream)
            ' Read the content.
            Dim responseFromServer As String = reader.ReadToEnd()
            ' Display the content.
            Console.WriteLine(responseFromServer)

            ' +++++ Parse Query Response Here ++++++++++++++++++++++++++++++++++++++++
            ' +++++ Sample of Query Response Is Available On The Next Section +++++

            ' Clean up the streams.
            reader.Close()
            dataStream.Close()
            response.Close()
        End Sub
    End Class
End Namespace
```

2.4 Query Response (Payment Gateway → Merchant System)

Upon query process completion, Payment Gateway will return all fields same as Payment Response fields together with the following additional fields.

For Query transaction status, please refer to section 3.3.

No MerchantReturnURL involved. Merchant System can get query response on the same session.

No.	Field	Data Type	Max Length	Req?	Description
1.	TxnExists	N	1	Y	An identifier to indicate whether the transaction being queried exists in Payment Gateway. 0 – Transaction being queried exists in Payment Gateway. Merchant System can proceed to refer to the rest of other fields for details, e.g. TxnStatus 1 – Transaction being queried does not exist 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 TxnStatus will be -1 2 – There was some kind of internal error occurred during query processing. Merchant System can retry sending query request TxnStatus will be -2
2.	QueryDesc	AN	255	N	Description of query result
3.	TotalRefundAmount	N	12(2)	N	Applicable for PymtMethod CC only. Total successfully refunded amount
The rest of Query response fields are available in section 2.2 Payment Response.					

Sample Query Response (A single string will be returned to Merchant System on the same Query session)

TxnExists=0&QueryDesc=Exists&ServiceID=D01&PymtID=ABC1234567890&Amount=1234.00&CurrencyCode=THB&TxnStatus=0&……

Sample Query Response (TxnExists=1)
TxnExists=1&QueryDesc=Transaction Not Exists&TransactionType=QUERY&PymtMethod=CC&ServiceID=TIS&PaymentID=JEFSIT13110700004&OrderNumber=&Amount=0.14&CurrencyCode=MYR&TxnID=&IssuingBank=&TxnStatus=-1&AuthCode=&BankRefNo=&TxnMessage=&HashValue=1d54f8eb9ff6c92c09737f61b8a68afed0ff1725a5c22efbc3b7c5bb50a2cd0c

Sample Query Response (TxnExists=2)
TxnExists=2&QueryDesc=Invalid Service ID&TransactionType=QUERY&PymtMethod=CC&ServiceID=TI1&PaymentID=JEFSIT131107000003&OrderNumber=&Amount=0.14&CurrencyCode=MYR&TxnID=&IssuingBank=&TxnStatus=-2&AuthCode=&BankRefNo=&TxnMessage=&HashValue=9ee1609200924444f924993361a138be3da59d8545db14e5985f787e1254c4c7

2.5 Capture Request (Merchant System → Payment Gateway)

The following fields are the Payment information expected from Merchant System to Payment Gateway in order to capture the original authorization transaction (payment request with transaction type AUTH):

NOTE: The protocol used to communicate with Payment Gateway for test environment is not limited but production environment only allows TLS 1.2 protocol for PCI compliance and security concerns. For Capture integration, it is recommended to test out the TLS 1.2 connectivity with production environment before going LIVE.

No.	Field	Data Type	Max Length	Req?	Description
1.	TransactionType	A	7	Y	CAPTURE
2.	PymtMethod	A	3	Y	Payment Method submitted in the original Payment Request being captured
3.	ServiceID	AN	3	Y	Merchant Service ID given by eGHL
4.	PaymentID	AN	20	Y	Unique Transaction ID or Reference Code assigned by Merchant System for the original authorization transaction
5.	Amount	N	12(2)	Y	Transaction amount to be captured in 2 decimal places, e.g. 100.00 Should not be exceeding the original authorization request amount
6.	CurrencyCode	N	3	Y	Original authorization transaction's ISO4217 3-letter currency code
7.	HashValue	AN	40	Y	Message digest value calculated by Merchant System in hexadecimal string using SHA256 hash algorithm Re: Section 2.13.1.3

2.6 Capture Response (Merchant System → Payment Gateway)

Upon capture process completion, the following fields will be returned from Payment Gateway to Merchant System in order to complete an end-to-end capture process.

No MerchantReturnURL involved. Merchant System can get Capture response on the same session.

No.	Field	Data Type	Max Length	Req?	Description
1.	TransactionType	A	7	Y	Follows request
2.	PymtMethod	A	3	Y	Follows request
3.	ServiceID	AN	3	Y	Follows request
4.	PaymentID	AN	20	Y	Follows request
5.	Amount	N	12(2)	Y	Follows request
6.	CurrencyCode	N	3	Y	Follows request
7.	TxnStatus	N	4	Y	Capture status Re: Section 3.2
8.	HashValue	AN	100	Y	Message digest value calculated by Payment Gateway in hexadecimal string using SHA256 hash algorithm Re: Section 2.13.1.2
9.	TxnMessage	AN	255	N	Message that briefly explains the response
10.	TxnID	AN	30	N	Unique Transaction ID or Reference Code assigned by Payment Gateway for the original authorization transaction, available only if Payment Gateway received the original authorization transaction

11.	IssuingBank	AN	30	N	This field is the Bank Name keyed in by customer on eGHL Payment Info Collection Page Available only if Payment Gateway received the original authorization transaction
12.	AuthCode	AN	12	N	Available only if Payment Gateway received the original authorization transaction and only if bank returned the Auth Code

Sample Capture Response (A single string will be returned to Merchant System on the same Capture session)

ServiceID=D01&PymtID=ABC1234567890&TxnStatus=0&……

2.7 Reversal Request (Merchant System → Payment Gateway)

Prior to sending Reversal, recommended Merchant System to send a Query request to Payment Gateway to get the actual payment status. Based on Query response, Merchant System will be able to know whether the payment transaction is successful in order to further determining whether to send a Reversal request to reverse the original payment.

Under the following situations, Merchant System can send Reversal request to Payment Gateway to reverse payment previously sent to Payment Gateway:

1. Error Handling
    a. Timeout – Merchant System successfully sent out Payment request to Payment Gateway but Merchant System is not able to receive response within timeout period.

    b. Others – Any other errors, encountered by Merchant System, that could have direct impact to the payment, such as power failure, database errors, hardware failure and system failure (bugs, services interrupted).

2. Product Fulfillment Failure
    Merchant System received payment approved status from Payment Gateway but Merchant System somehow failed to complete product fulfillment process for the respective booking, reservation or order made by the consumer due to certain scenarios such as inventory availability problem or unforeseen price changes problem.

Some tips for Reversal handling by Merchant System:

1. Reversal is not allowed to be used by merchants for Refund purposes.

2. Reversal process can only be performed before bank's settlement. As such, if any of the above scenarios occurred, it is advisable to reverse a transaction within one hour of its original Sale time. Reversal of a transaction already settled by bank will be rejected with TxnStatus=1 (Failed). Merchant System should not attempt Reversal anymore.

3. Reversal amount must be the same as the original Payment amount. Partial Reversal will be rejected with TxnStatus=-1 (Not Found). Merchant System should only attempt Reversal with the original Payment amount and currency code.

4. Merchant System can retry performing Reversal of the respective transaction upon receiving Query TxnStatus = 31 or Reversal TxnStatus=2 (Reversal is pending processing) or TxnStatus=-2 (Internal system error) or when Merchant System encountered communication error with Payment Gateway during Reversal processing, such as Merchant System failed to send Reversal request or timeout Payment Gateway for Reversal response. Under these circumstances, in order to obtain Reversal response, Merchant System can either send Query to check whether status received is TxnStatus=9 (Transaction Reversed) or perform Reversal request again and each Reversal retry attempt is recommended to be at least 5 minutes apart in order to save server and network resources.

5. If Merchant System somehow failed to receive Payment response from Payment Gateway, the original Payment request could be failed or never received by Payment Gateway. For such Reversal, Merchant System will receive successful Reversal response with TxnStatus=0 (Transaction Reversed). Merchant System should not attempt Reversal anymore.

The following fields are the Payment information expected from Merchant System to Payment Gateway in order to reverse the original sale/payment:

NOTE: The protocol used to communicate with Payment Gateway for test environment is not limited but production environment only allows TLS 1.2 protocol for PCI compliance and security concerns. For Reversal integration, it is recommended to test out the TLS 1.2 connectivity with production environment before going LIVE.

No.	Field	Data Type	Max Length	Req?	Description
1.	TransactionType	A	5	Y	RSALE – Reversal
2.	PymtMethod	A	3	Y	Payment Method submitted in the original Payment Request being reversed
3.	ServiceID	AN	3	Y	Merchant Service ID given by eGHL
4.	PaymentID	AN	20	Y	Unique Transaction ID or Reference Code assigned by Merchant System for the original Payment transaction
5.	Amount	N	12(2)	Y	Original Payment transaction amount to be reversed in 2 decimal places, e.g. 100.00 No partial reversal is allowed
6.	CurrencyCode	N	3	Y	Original Payment transaction's ISO4217 3-letter currency code
7.	HashValue	AN	40	Y	Message digest value calculated by Merchant System in hexadecimal string using SHA256 hash algorithm Re: Section 2.13.1.3

Sample VB.net Code HTTP POST Reversal Request And Receive Reversal Response

Reference: http://msdn.microsoft.com/en-us/library/debx8sh9.aspx

```
Imports System
Imports System.IO
Imports System.Net
Imports System.Text
Namespace Examples.System.Net
   Public Class WebRequestPostExample

      Public Shared Sub Main()
         ' Create a request using a URL that can receive a post.
         Dim request As WebRequest = WebRequest.Create("https:// <to be provided by eGHL>")
         ' Set the Method property of the request to POST.
         request.Method = "POST"
         ' Create POST data and convert it to a byte array.
         Dim postData As String =
"TransactionType=RSALE&PymtMethod=CC&ServiceID=A07&PaymentID=
ABCDEFGH130820142128&Amount=228.00&CurrencyCode=MYR&HashValue=hash value generated"
         Dim byteArray As Byte() = Encoding.UTF8.GetBytes(postData)
         ' Set the ContentType property of the WebRequest.
         request.ContentType = "application/x-www-form-urlencoded"
         ' Set the ContentLength property of the WebRequest.
         request.ContentLength = byteArray.Length
         ' Get the request stream.
         Dim dataStream As Stream = request.GetRequestStream()
         ' Write the data to the request stream.
         dataStream.Write(byteArray, 0, byteArray.Length)
         ' Close the Stream object.
         dataStream.Close()

         ' Get the response.
         Dim response As WebResponse = request.GetResponse()
         ' Display the status.
         Console.WriteLine(CType(response, HttpWebResponse).StatusDescription)
         ' Get the stream containing content returned by the server.
         dataStream = response.GetResponseStream()
         ' Open the stream using a StreamReader for easy access.
         Dim reader As New StreamReader(dataStream)
         ' Read the content.
         Dim responseFromServer As String = reader.ReadToEnd()
         ' Display the content.
         Console.WriteLine(responseFromServer)

         ' +++++ Parse Reversal Response Here ++++++++++++++++++++++++++++++++++++
++
         ' +++++ Sample of Reversal Response Is Available On The Next Section +++++

         ' Clean up the streams.
         reader.Close()
         dataStream.Close()
         response.Close()
      End Sub
   End Class
End Namespace
```

2.8 Reversal Response (Payment Gateway → Merchant System)

Upon reversal process completion, the following fields will be returned from Payment Gateway to Merchant System in order to complete an end-to-end payment reversal process.

No MerchantReturnURL involved. Merchant System can get Reversal response on the same session.

No.	Field	Data Type	Max Length	Req?	Description
1.	TransactionType	A	5	Y	Follows request
2.	PymtMethod	A	3	Y	Payment Method CC – Credit Card DD – Direct Debit WA – e-Wallet
3.	ServiceID	AN	3	Y	Follows request
4.	PaymentID	AN	20	Y	Follows request
5.	Amount	N	12(2)	Y	Follows request
6.	CurrencyCode	N	3	Y	Follows request
7.	TxnStatus	N	4	Y	Reversal status Re: Section 3.4
8.	HashValue	AN	100	Y	Message digest value calculated by Payment Gateway in hexadecimal string using SHA256 hash algorithm Re: Section 2.13.1.2
9.	TxnMessage	AN	255	N	Message that briefly explains the response
10.	TxnID	AN	30	N	Unique Transaction ID or Reference Code assigned by Payment Gateway for the original Payment transaction,

					available only if Payment Gateway received the original Payment transaction
11.	IssuingBank	AN	30	N	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 Available only if Payment Gateway received the original Payment transaction
12.	AuthCode	AN	12	N	Available only if Payment Gateway received the original Payment transaction and only if bank returned the Auth Code

Sample Reversal Response (A single string will be returned to Merchant System on the same Reversal session)

ServiceID=D01&PymtID=ABC1234567890&TxnStatus=0&.....

2.9 Refund Request (Merchant System → Payment Gateway)

The following fields are the Payment information expected from Merchant System to Payment Gateway in order to refund the original sale/payment:

NOTE: The protocol used to communicate with Payment Gateway for test environment is not limited but production environment only allows TLS 1.2 protocol for PCI compliance and security concerns. For Refund integration, it is recommended to test out the TLS 1.2 connectivity with production environment before going LIVE.

No.	Field	Data Type	Max Length	Req?	Description
1.	TransactionType	A	5	Y	REFUND – Refund
2.	PymtMethod	A	3	Y	Payment Method submitted in the original Payment Request being refunded
3.	ServiceID	AN	3	Y	Merchant Service ID given by eGHL
4.	PaymentID	AN	20	Y	Unique Transaction ID or Reference Code assigned by Merchant System for the original Payment transaction
5.	Amount	N	12(2)	Y	Transaction amount to be refunded in 2 decimal places, e.g. 100.00 Should not be exceeding the original sale request amount
6.	CurrencyCode	N	3	Y	Original Payment transaction's ISO4217 3-letter currency code
7.	HashValue	AN	40	Y	Message digest value calculated by Merchant System in hexadecimal string using SHA256 hash algorithm Re: Section 2.13.1.3

2.10 Refund Response (Payment Gateway → Merchant System)

Upon refund process completion, the following fields will be returned from Payment Gateway to Merchant System in order to complete an end-to-end refund process.

No MerchantReturnURL involved. Merchant System can get Refund response on the same session.

No.	Field	Data Type	Max Length	Req?	Description
1.	TransactionType	A	7	Y	Follows request
2.	PymtMethod	A	3	Y	Follows request
3.	ServiceID	AN	3	Y	Follows request
4.	PaymentID	AN	20	Y	Follows request
5.	Amount	N	12(2)	Y	Follows request
6.	CurrencyCode	N	3	Y	Follows request
7.	TxnStatus	N	4	Y	Capture status Re: Section 3.4
8.	HashValue	AN	100	Y	Message digest value calculated by Payment Gateway in hexadecimal string using SHA256 hash algorithm Re: Section 2.13.1.2
9.	TxnMessage	AN	255	N	Message that briefly explains the response
10.	TxnID	AN	30	N	Unique Transaction ID or Reference Code assigned by Payment Gateway for the original Payment transaction, available only if Payment Gateway received the original Payment transaction
11.	IssuingBank	AN	30	N	

					This field is the Bank Name keyed in by customer on eGHL Payment Info Collection Page Available only if Payment Gateway received the original Payment transaction
12.	AuthCode	AN	12	N	Available only if Payment Gateway received the original Payment transaction and only if bank returned the Auth Code

Sample Refund Response (A single string will be returned to Merchant System on the same Refund session)

ServiceID=D01&PymtID=ABC1234567890&TxnStatus=0&.....

2.11 Settlement Request (Merchant System → Payment Gateway)

The following fields are the Payment information expected from Merchant System to Payment Gateway in order to settle the transaction batch of a particular Terminal Account ID (SettleTAID):

NOTE: The protocol used to communicate with Payment Gateway for test environment is not limited but production environment only allows TLS 1.2 protocol for PCI compliance and security concerns. For Settlement integration, it is recommended to test out the TLS 1.2 connectivity with production environment before going LIVE.

No.	Field	Data Type	Max Length	Req?	Description
1.	TransactionType	A	5	Y	SETTLE – Settlement Note: For PymtMethod MOTO only
2.	ServiceID	AN	3	Y	Merchant Service ID given by eGHL
3.	SettleTAID	N	10	Y	Terminal Account ID to be settled, available in SALE payment response
4.	SettleAmount	N	12(2)	Y	Total transaction amount to be settled in 2 decimal places, e.g. 100.00 Gateway will proceed to settle only if Gateway's amount and transaction count tallied with SettleAmount and SettleTxnCount.
5.	SettleTxnCount	N	3	Y	Number of transactions to be settled Note: Must not exceed 999 Gateway will proceed to settle only if Gateway's amount and transaction count tallied with SettleAmount and SettleTxnCount.
6.	HashValue	AN	40	Y	Message digest value calculated by Merchant System in hexadecimal string using SHA256 hash algorithm Re: Section 2.13.1.4

2.12 Settlement Response (Payment Gateway → Merchant System)

Upon settlement process completion, the following fields will be returned from Payment Gateway to Merchant System in order to complete an end-to-end settlement process.

No MerchantReturnURL involved. Merchant System can get Settlement response on the same session.

No.	Field	Data Type	Max Length	Req?	Description
1.	TransactionType	A	5	Y	Follows request
2.	ServiceID	AN	3	Y	Follows request
3.	SettleTAID	N	8	Y	Follows request
4.	GatewaySettleAmount	N	12(2)	C	Total transaction amount for the settlement batch of SettleTAID on Gateway in 2 decimal places, e.g. 100.00 Available only if no internal error
5.	GatewaySettleTxnCount	N	3	C	Number of transactions for the settlement batch of SettleTAID on Gateway Note: Not exceeding 999, available only if no internal error
6.	TxnStatus	N	4	Y	Re: Section 3.5
7.	TxnMessage	AN	255	N	Message that briefly explains the response
8.	HashValue	AN	40	Y	Message digest value calculated by Merchant System in hexadecimal string using SHA256 hash algorithm Re: Section 2.13.1.5

2.13 Additional Information

2.13.1 Usage of Hash Value

A hash value (or simply hash), also called a message digest, is a number generated from a text string. The hash is substantially smaller than the text string itself, and is generated by a formula or hash algorithm in such a way that it is extremely unlikely that some other texts will produce the same hash value.

For online payment processing, hashing plays an important role to ensure the transmitted request and response messages have not been tampered with, in order to achieve data integrity.

For transaction request, Merchant System is required to use SHA256 hash algorithm to generate a hash value from a combination of Merchant Password and certain transaction fields, and then includes the hash value in HashValue field before sending the request to Payment Gateway which will then generate a hash value based on the same method and then verify these two hash values. If both hash value matched, Payment Gateway will further process the payment request or else it will discard the request message and will treat it as an invalid message.

Likewise, for transaction response, Merchant System can generate hash value based on Merchant Password and certain response fields and then verify this hash value with the value retrieved from HashValue field of response. If both hash value mismatched, Merchant System can treat the response as invalid and discard it. Merchant System can only accept genuine payment response which had not been tampered with.

2.13.1.1 Payment Request Hashing

Payment request's Hash Value should be generated based on the following fields:

Hash Key = Password + ServiceID + PaymentID + MerchantReturnURL + MerchantApprovalURL + MerchantUnApprovalURL + MerchantCallBackURL + Amount + CurrencyCode + CustIP + PageTimeout + CardNo + Token

NOTE:

a) If Token value is provided in Token field, Token value to be put in the above Hash Key string needs to be in clear format and not in encoded format. For example, if Token value is "2r8j/OsME hxI99PZcHwVg==", then Token value to be put in Hash Key is "2r8j/OsME hxI99PZcHwVg==" instead of "2r8j%2FOsME%20hxI99PZcHwVg%3D%3D".
b) If CardNo / Token / MerchantApprovalURL / MerchantUnApprovalURL / MerchantCallBackURL are not provided in Payment request, then these fields in the above Hash Key should be just left empty.

Hash Key Example
abc123S22PAYTEST123https://www.shop.com/success.asp12.34MYR113.210.6.1509004444333322221111

Hash Value (SHA256)
28010d7207bdbd6e8ae3890fdb56c541c552c10b609b978fd69428fbb7a4fbc2

2.13.1.2 Payment/Query/Reversal/Capture/Refund Response Hashing

Payment/Query/Reversal/Capture/Refund response's Hash Value should be generated based on the following fields:

a) HashValue2 (Highly Recommended)

Hash Key = Password + TxnID + ServiceID + PaymentID + TxnStatus + Amount + CurrencyCode + AuthCode + OrderNumber + Param6 + Param7

Hash Key Example
abc123TESTTXN123S22PAYTEST12312.34MYR12345600079010006677

Hash Value (SHA256)
8795c391a3091585295906a0694d9d13d29c38aa3d4d4521385f222ac19fb773

b) HashValue

Hash Key = Password + TxnID + ServiceID + PaymentID + TxnStatus + Amount + CurrencyCode + AuthCode

Hash Key Example
abc123TESTTXN123S22PAYTEST12312.34MYR123456

Hash Value (SHA256)
67ab68d31bafd14dfe1972301012c41b08e49da314871fdf9cce92fc698300f1

2.13.1.3 Query/Reversal/Capture/Refund Request Hashing

Query/Reversal/Capture/Refund request's Hash Value should be generated based on the following fields:

Hash Key = Password + ServiceID + PaymentID + Amount + CurrencyCode

Hash Key Example
abc123S22PAYTEST12312.34MYR

Hash Value (SHA256)
320b0e212a875228fb80efd7604534af47055e56167aa7e83842cf68788097b5

2.13.1.4 Settle Request Hashing

Settle request's Hash Value should be generated based on the following fields:

Hash Key = Password + ServiceID + SettleTAID + SettleAmount + SettleTxnCount

Hash Key Example
abc123S22101234.56999

Hash Value (SHA256)
2885423326b1396862040c3cf3b658f91fb0e0164485fc6a25acc6e0107ae3a7

2.13.1.5 Settle Response Hashing

Settle response's Hash Value should be generated based on the following fields:

Hash Key = Password + ServiceID + SettleTAID + GatewaySettleAmount + GatewaySettleTxnCount + TxnStatus

Hash Key Example
abc123S22101234.569990

Hash Value (SHA256)
e43ecfbb451f00727d28ca74ce0b9f66297b0748e4b1b6461ae834fc47dfbaea

2.13.1.6 Optimize/SOP Hashing

Optimize/SOP request's Hash Value should be generated based on the following fields:

Hash Key = Password + ServiceID + PaymentID + CustIP

Hash Key Example
abc123S22PAYTEST123127.0.0.1

Hash Value (SHA256)
2f57b8563373ab7e489034e7e21d62f7146e8a7c90e8c4636aa63831ebfd5ca7

2.14 Masterpass Express Checkout (MPE)

Masterpass Express Checkout consists of two process flows:

1) Pairing with checkout is for consumer who first time uses Masterpass wallet from Merchant System and consumer would like to connect their Masterpass wallet to Merchant System.

2) Express checkout is for consumer who has already connected their Masterpass wallet with Merchant System.

In order to complete the Masterpass Express Checkout process flow, Merchant System is required to send two MPE request messages to Payment Gateway.

1st MPE request message will be the same for both process flow but response message will be different, namely Pairing with checkout response or Express checkout response.

2nd MPE request message will be similar to normal payment request (section 2.1) with additional Masterpass fields included. The Masterpass fields to be included are depending on whether the 1st MPE response message belongs to Pairing with checkout or Express checkout. 2nd MPE Response will be the same as normal payment response (section 2.2)

2.14.1 Requirement

Merchant needs to provide Masterpass account details to eGHL which will use the account details to call Masterpass API.

1. Masterpass Checkout ID (Masterpass Merchant Portal->Digital Asset)
2. Consumer Key (Masterpass Merchant Portal->Key Management)
3. Keystore Password (Developer Portal -> Generate Masterpass Key)
4. Keystore p12 Certificate (Developer Portal -> Generate Masterpass Key)

2.14.2 1st MPE Request (Merchant System → Payment Gateway)

1st MPE request message received from Merchant System is to determine the specific consumer from Merchant System should go for Pairing with checkout or Express checkout.

The following fields are the information expected from Merchant System to Payment Gateway in order to perform MPE transaction.

No MerchantReturnURL involved. Merchant System can get response on same session.

NOTE: The protocol used to communicate with Payment Gateway for test environment is not limited but production environment only allows TLS 1.2 protocol for PCI compliance and security concerns. It is recommended to test out the TLS 1.2 connectivity with production environment before going LIVE.

No.	Field	Data Type	Max Length	Req?	Description
1.	ServiceID	AN	3	Y	Merchant Service ID given by eGHL
2.	TokenType	A	3	Y	MPE – Masterpass Express Checkout

3.	Token	AN	50	Y	This field is expecting consumer 's Login ID / Unique Identity to the merchant system
4.	PaymentDesc	AN	100	Y	Order's descriptions
5.	Amount	N	12(2)	Y	Payment Amount submitted in Payment Request
6.	CurrencyCode	A	3	Y	Payment Currency Code submitted in Payment Request

Sample 1st MPE Request Message

ServiceID=SIT&TokenType=MPE&Token=User@abc.com&PaymentDesc=Payment for Order:12345&Amount=10.00&CurrencyCode=MYR

2.14.3 1st MPE Response (Payment Gateway → Merchant System)

Upon 1st MPE process completion, Payment Gateway will return two different responses to Merchant System depending on consumer had been connected or not connected with Merchant System.

These two responses are Pairing with checkout response and Express checkout response:

1) When Pairing with checkout response received, Merchant System is required to launch Masterpass UI (Lightbox) (section 2.14.3) to allow user to login into their Masterpass wallet account and pair with Merchant System. In some circumstances, Payment Gateway will return ReqToken and PairingToken to Merchant System even the consumer had paired with merchant. If these tokens received, Merchant System has to continue to perform Masterpass UI (Lightbox) process. This could be due to the previous token had expired or invalid. Therefore, consumer needs to pair again with merchant.

2) When Express checkout response received, Merchant System is required to display the LastFour digit or other credit card details based on merchant web page requirement whereas CardId from Cards field and PreCheckoutId are to be included in 2nd MPE request message (section 2.14.5).

Response fields for Pairing with checkout as below.

No.	Field	Data Type	Max Length	Req?	Description
1.	ReqToken	AN	60	Y	Request token from Masterpass API
2.	PairingToken	AN	60	Y	Pairing request token from Masterpass API

Response fields for Express checkout as below.

No.	Field	Data Type	Max Length	Req?	Description
1.	PreCheckoutId	AN	60	Y	Consumer's pre-checkout ID from Masterpass API. To be included in 2nd MPE Request Message (section 2.14.5)
2.	Cards	AN	-	Y	Consumer's credit card details from Masterpass API. The Selected CardId from Merchant System is to be included in 2nd MPE Request Message (section 2.14.5)

Sample 1st MPE Response (A single string will be returned to Merchant System on the same session)

Pairing with checkout response
ReqToken=b86f20aa51e71bb0e57486917b58cb6529c8a130&PairingToken=d22fc1784c2d5945ade0e674b05757446174828d

Express checkout response – JSON format
{"PreCheckoutId":"a4a6x55-26qm7r-izo4aj1h-1-izobw8vh-6qf","Cards":[{"LastFour":"0014","CardId":"358f9812-a99a-49d9-9385-f0a7b67e377c","BrandId":"master","CardAlias":null,"ExpiryMonth":12,"SelectedAsDefault":true,"BNBUnverified":null,"CardHolderName":"OM Testing","ExtensionPoint":null,"BillingAddress":{"City":"KL","Country":"US","CountrySubdivision":"WA","Line1":"123, street 123","Line2":null,"Line3":null,"PostalCode":"11222","ExtensionPoint":null},"BrandName":"MasterCard","ExpiryYear":2020},{"LastFour":"0010","CardId":"ddab9f01-7e38-49e7-b1ee-f598bc9357da","BrandId":"visa","CardAlias":null,"ExpiryMonth":12,"SelectedAsDefault":false,"BNBUnverified":null,"CardHolderName":"OM Testing","ExtensionPoint":null,"BillingAddress":{"City":"KL","Country":"US","CountrySubdivision":"WA","Line1":"123, street 123","Line2":null,"Line3":null,"PostalCode":"11222","ExtensionPoint":null},"BrandName":"Visa","ExpiryYear":2020}]}

2.14.4 Masterpass Lightbox Implementation (Merchant System → Masterpass)

Merchant system is required to launch Masterpass UI (Lightbox) upon receiving Pairing with checkout response.

ReqToken and PairingToken from Pairing with checkout response are to be used by Merchant System to invoke Masterpass UI (Lightbox).

Steps as below:-

1. To invoke the Lightbox, Merchant System must include the following scripts to the page on which they are adding the Masterpass Checkout buttons.

   NOTE: Merchants invoking the Lightbox from an iFrame must include both the sandbox and production scripts on the parent (outer) web page and the iFrame source that is invoking Masterpass Lightbox.

   a. Add jQuery. Include this jQuery file from the public jQuery repository:
   https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js

   b. Add the Masterpass Integration Script to your checkout page.
   – Sandbox:
   https://sandbox.static.masterpass.com/dyn/js/switch/integration/MasterPass.client.js
   – Production:
   https://static.masterpass.com/dyn/js/switch/integration/MasterPass.client.js

2. Add the Masterpass Checkout Button to your checkout page.

   NOTE: Be sure to change the image name to the image size that you would like to use. For further details, refer to Masterpass Branding.

3. Launch Masterpass checkout on the click of the button.

4. Handle the callback on completion of checkout.
   On completion of checkout (success, failure or cancellation), the control will be transferred to your system either via the callback url or the callback functions.

   NOTES:
   - If Masterpass is forced to go full screen, then you will not be able to call the JavaScript callback method and must use a merchant callback redirect. As a result, you must always support the callback url redirect.
   - If the successCallback parameter is not provided, wallets will not be able to complete checkout from the standard full-screen display. If the failureCallback parameter is not provided, failed checkouts will not be handled properly.

5. For the full list of Lightbox parameters, see Masterpass Lightbox Parameters.

Lightbox Javascript Example

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
   <title>IPG MasterPass Simulator</title>
</head>
<body>

  <form id="form1" runat="server">
   <div>
          <script type="text/javascript"
src="https://sandbox.static.masterpass.com/dyn/js/switch/integration/MasterPass.client.js"></
script>
          <div id='MPCheckoutBtn' onClick='MasterPassCheckout();' style="cursor:pointer;">
          <img
src="https://www.mastercard.com/mc_us/wallet/img/mcpp_wllt_btn_chk_147x034px.png"
alt="Buy with MasterPass">
          </div>
          <script>function
MasterPassCheckout(){MasterPass.client.checkout({"requestPairing":"true","requestToken":"d7
5fe3d5710781363402842162989a251a7b569a","pairingRequestToken":"18fe6f0395f1488fa134
c8ab886385d8bf3f3d9d","callbackUrl":"https://test2pay.ghl.com/IPGSimulator/RespFrmMP.aspx
","failureCallback":"https://test2pay.ghl.com/IPGSimulator/RespFrmMP.aspx","cancelCallback":
"https://test2pay.ghl.com/IPGSimulator/RespFrmMP.aspx","merchantCheckoutId":"a32d844020
2b408dbdcc3ca8763d4625","requestedDataTypes":["ADDRESS","PROFILE","CARD"],"requestEx
pressCheckout":"true","allowedCardTypes":["master","visa"],"suppressShippingAddressEnable":
"true","version":"v6"});}</script>
              <a
href="javascript:openWindow('https://www.mastercard.com/mc_us/wallet/learnmore/en/MY/',6
00,340,'LearnMore');" style="font-size:12px;">Learn More</a>
   </div>
   </form>

   <script src="jquery/jquery.min.js"></script>
       <script src="jquery/jquery-migrate.min.js"></script>
       <script type="text/javascript">
       function openWindow(szURL, winWidth, winHeight, szWinName) {
                  window.open(szURL, szWinName, config = 'height=' + winHeight + ', width='
+ winWidth + ', toolbar=no, menubar=no, scrollbars=yes, resizable=no,location=no,
directories=no, status=no');
              }
       </script>

</body>
</html>
```

Redirect Response to callback URL or call callback function from Masterpass to Merchant System

After the consumer completes checkout, Masterpass returns data to the Merchant System. By default, Masterpass redirects the browser to the callback URL, returning the data via URL parameters. Optionally, if callback functions were provided, Masterpass returns control to the page that initiated the Lightbox, and the data is passed to the callback function.

Under certain conditions, the Masterpass UI may be forced to run in Full Screen display instead of launching the Lightbox. When this occurs, the browser will redirect to the callback URL when the Masterpass flow is complete, even if callback functions were supplied.

On success, Masterpass returns the following parameters:-

- mpstatus: String that indicates whether the Masterpass flow resulted in success, failure, or cancel.
- checkout_resource_url: The API URL that will be used to retrieve checkout information.
- oauth_verifier: Checkout verifier token.
- oauth_token: Checkout request token. This token has the same value as the checkout request token that is submit in Masterpass UI (Lightbox).
- pairing_verifier: Pairing verifier token.
- pairing_token: Pairing request token. Note: Optional return field

Note: pairing_verifier and pairing_token is an optional. It's ONLY return when consumer enable pairing/connect with Merchant System in Masterpass UI.

Example Callback URL (Success) With Enable Pairing
http://www.somemerchant.com/checkoutcomplete.htm ?mpstatus=success
&checkout_resource_url=https://sandbox.api.mastercard.com/masterpass/v6/checkout/711472310&oauth_verifier=fbe45bcad30299c93765b1fb4b45bab208f84458
&oauth_token=d9382e34e0721a68a9952110cecdf89517e45498
&pairing_verifier=6c50838e31b7441e6eafa2229385452889255b13
&pairing_token=35b2a0cf87f8160fcb5d24996a12edb7cce4c530

Example Callback URL (Success) with Not Enable Pairing
http://www.somemerchant.com/checkoutcomplete.htm ?mpstatus=success
&checkout_resource_url=https://sandbox.api.mastercard.com/masterpass/v6/checkout/711472310&oauth_verifier=fbe45bcad30299c93765b1fb4b45bab208f84458
&oauth_token=d9382e34e0721a68a9952110cecdf89517e45498

Example Callback URL (Cancel)
http://www.somemerchant.com/checkoutcomplete.htm ?mpstatus=cancel

Example Callback URL (Failure)
http://www.somemerchant.com/checkoutcomplete.htm ?mpstatus=failure

If Merchant System received Cancel or Failure from Masterpass UI (Lightbox), it is recommended that the Merchant System can lead consumer to normal payment process via eGHL payment page.

2.14.5 2nd MPE Request (Merchant System → Payment Gateway)

2nd MPE request message will be following the normal payment request message (section 2.1) with additional Masterpass fields as shown below.

TokenType and Token are two of the existing fields in the normal payment request message (section 2.1). These two fields are required fields to continue with Masterpass process flow.

1) Additional request fields for Pairing with checkout.

No.	Field	Data Type	Max Length	Req?	Description
1.	ReqToken	AN	60	Y	Request token (oauth_token) received from Masterpass UI (Lightbox)
2.	ReqVerifier	AN	60	Y	Request verifier (oauth_verifier) received from Masterpass UI (Lightbox)
3.	PairingToken	AN	60	C	Pairing request token (pairing_token) received from Masterpass UI (Lightbox) Note: This value is needed when value return by Masterpass UI (Lightbox)
4.	PairingVerifier	AN	60	C	Pairing request verifier (pairing_verifier) received from Masterpass UI (Lightbox) Note: This value is needed when value return by Masterpass UI (Lightbox)
5.	CheckoutResourceURL	AN	100	Y	Checkout URL (checkout_resource_url) received from Masterpass UI (Lightbox)

Sample HTML Form Post Payment Request

```
<form name="frmPayment" method="post" action="https://<URL to be provided by eGHL>">
<input type="hidden" name="TransactionType" value="SALE">
<input type="hidden" name="PymtMethod" value="ANY">
<input type="hidden" name="ServiceID" value="A07">
..
..
<input type="hidden" name="TokenType" value="MPE">
<input type="hidden" name="Token" value="User@abc.com">
<input type="hidden" name="ReqToken" value="
d9382e34e0721a68a9952110cecdf89517e45498">
<input type="hidden" name="ReqVerifier" value="
fbe45bcad30299c93765b1fb4b45bab208f84458">
<input type="hidden" name="PairingToken"
value="35b2a0cf87f8160fcb5d24996a12edb7cce4c530">
<input type="hidden" name="PairingVerifier"
value="6c50838e31b7441e6eafa2229385452889255b13">
<input type="hidden" name="CheckoutResourceURL" value="
https://sandbox.api.mastercard.com/masterpass/v6/checkout/711472310">
</form>
```

2) Additional fields for Express checkout.

No.	Field	Data Type	Max Length	Req?	Description
1.	CardId	AN	60	Y	Selected CardId by consumer via Merchant System. Re: Section 2.14.3
2.	PreCheckoutId	AN	60	Y	PreCheckoutId value received from Payment Gateway's Express checkout response Re: Section 2.14.3

Sample HTML Form Post Payment Request

```
<form name="frmPayment" method="post" action="https://<URL to be provided by eGHL>">
<input type="hidden" name="TransactionType" value="SALE">
<input type="hidden" name="PymtMethod" value="ANY">
<input type="hidden" name="ServiceID" value="A07">
..
..
<input type="hidden" name="TokenType" value="MPE">
<input type="hidden" name="Token" value="User@abc.com">
<input type="hidden" name="CardId" value="65e86c05-7049-46f7-b2dd-c1685b43e9d2">
<input type="hidden" name="PreCheckoutId" value="a4a6x55-d5am50-ixkjgdiy-1-ixzj5pca-bvor">
</form>
```

2.14.6 2nd MPE Response (Payment Gateway → Merchant System)

2nd MPE response will be same as normal payment response (referring to section 2.2).

2.15 Masterpass Standard Checkout (MSC)

Masterpass Standard Checkout ONLY applicable to merchants who had integration with eGHL Masterpass Express Checkout (MPE) and want to skip eGHL Payment Page.

2.15.1 1st MSC Request Message (Merchant System -> Payment Gateway)

No.	Field	Data Type	Max Length	Req?	Description
1.	ServiceID	AN	3	Y	Merchant Service ID given by eGHL
2.	TokenType	A	3	Y	MSC – Masterpass Standard Checkout
3.	PaymentDesc	AN	100	Y	Order's descriptions
4.	Amount	N	12(2)	Y	Payment Amount submitted in Payment Request
5.	CurrencyCode	A	3	Y	Payment Currency Code submitted in Payment Request

Sample 1st MPE Request Message

ServiceID=SIT&TokenType=MPE&PaymentDesc=Payment for Order: 12345&Amount=10.00&CurrencyCode=MYR

2.15.2 1st MSC Response (Payment Gateway → Merchant System)

Masterpass Standard Checkout response field as below:-

No.	Field	Data Type	Max Length	Req?	Description
1.	ReqToken	AN	60	Y	Request token from MSC flow

Sample 1st MSC Response (A single string will be returned to Merchant System on the same session)

Masterpass Standard Checkout response
ReqToken=b86f20aa51e71bb0e57486917b58cb6529c8a130

2.15.3 Masterpass Lightbox Implementation for Standard Checkout (Merchant System -> Masterpass)

Refer to section 2.14.4 Masterpass Lightbox Implementation (Merchant System -> Masterpass)

Fields that MUST exclude from Lightbox JavaScript as below:-
1. requestPairing=true
2. pairingRequestToken

2.15.4 2nd MSC Request Message (Merchant System -> Payment Gateway)

Refer to section 2.14.5 2nd MPE Request Message (Merchant System -> Payment Gateway)

Changes on field is TokenType='MSC' and Token not a required field.

2.15.5 2nd MSC Response (Payment Gateway → Merchant System)

2nd MSC response will be same as normal payment response (referring to section 2.2).

3 FPX E-Madate

3.1 Enrollment request (Merchant System → Payment Gateway)

The following fields are the Enrollment information expected from Merchant System to Payment Gateway in order to perform an online enrollment transaction:

No.	Field	Data Type	Max Length	Req?	Description
1.	TransactionType	A	7	Y	ENROLL – the Parameter use for gateway side to capture the transaction as FPX e-Mandate(Recurring Transaction)
2.	PymtMethod	A	3	Y	Payment Method DD – Direct Debit
3.	ServiceID	AN	3	Y	Merchant Service ID given by eGHL
4.	PaymentID	AN	20	Y	Unique transaction ID/reference code assigned by merchant for this transaction (No duplicate PaymentID is allowed)
5.	OrderNumber	AN	20	Y	Reference number / Invoice number for this order PaymentID must be unique but OrderNumber can be the same under different PaymentID, indicating multiple payment attempts are made on a particular order If Order Number is not applicable, please provide the same value as PaymentID *For insurance type of transaction, this may be use for the **'Policy Number'**

6.	PaymentDesc	AN	27	Y	The description/reason for debiting the payment.
7.	MerchantReturnURL	AN	255	Y	Merchant system's browser redirect URL which receives payment response from eGHL when transaction is completed (approved/declined/system error/ cancelled by buyer on eGHL Payment Page) If MerchantApprovalURL is provided, when payment is approved, MerchantApprovalURL will be used instead of MerchantReturnURL If MerchantUnApprovalURL is provided, when payment is declined, MerchantUnApprovalURL will be used instead of MerchantReturnURL For server-to-server integration for non-3D payment transaction, please provide value as s2s. Note: Replace "&" with ";" if any. e.g. https://merchantdomain/index.php?field1=value1&field2=value2 to https://merchantdomain/index.php?field1=value1;field2=value2
8.	Amount	N	12(2)	Y	Payment amount in 2 decimal places regardless whether the currency has decimal places or not. Please exclude "," sign. e.g. 1000.00 for IDR Invalid format: 1,000.00 or 1000

9.	CurrencyCode	A	3	Y	3-letter ISO4217 of Payment Currency Code Re: Section 3.6
10.	HashValue	AN	100	Y	Message digest value calculated by Merchant System in hexadecimal string using SHA256 hash algorithm Re: Section 2.7.1.1
11.	CustIP	AN	20	Y	Customer's IP address captured by merchant system
12.	CustName	AN	50	Y	Customer Name
13.	CustEmail	AN	60	Y	Customer's Email Address
14.	CustPhone	AN	25	Y	Customer's Contact Number
19.	MerchantApprovalURL	AN	255	N	URL to link to merchant's website when payment is approved If not provided, MerchantReturnURL will be used Instead. Note: Replace "&" with ";" if any. e.g. https://merchantdomain/index.php?field1=value1&field2=value2 to https://merchantdomain/index.php?field1=value1;field2=value2
20.	MerchantUnApprovalURL	AN	255	N	URL to link to merchant's website when payment is declined If not provided, MerchantReturnURL will be used instead.

					Note: Replace "&" with ";" if any. e.g. https://merchantdomain/index.php?field1=value1&field2=value2 to https://merchantdomain/index.php?field1=value1;field2=value2
21.	MerchantCallBackURL	AN	255	N	Server-to-server URL as an additional link to merchant's website to be informed of the transaction status This is useful when browser redirect URLs (MerchantReturnURL/MerchantApprovalURL/MerchantUnApprovalURL) were not able to receive payment response due to buyer's Internet connectivity problem or buyer closed browser Upon receiving response from Gateway, MerchantCallBackURL is to return an acknowledgement message "OK" to the Gateway or else Gateway will continue to send response to this URL for a maximum of 3 times Note: Replace "&" with ";" if any. e.g. https://merchantdomain/index.php?field1=value1&field2=value2 to https://merchantdomain/index.php?field1=value1;field2=value2
22.	LanguageCode	A	2	N	ISO 639-1 language Code for eGHL Payment Info Collection Page Re: Section 3.7
23.	PageTimeout	N	4	N	This parameter is the timeout period on eGHL Payment Page(for Info Collection) – It is running on time format(seconds)

					Applicable for merchant system which would like to bring forward to Payment Gateway, the time remaining before product/order is released For example, a movie ticket sales page shows time remaining countdown from 15 minutes till 5 minutes. Upon customer's clicking "checkout / proceed / pay" button, merchant system can then pass the value of (5 minutes x 60 seconds=300) seconds in this field to Gateway which will then continue the countdown from 5 minutes. Upon timeout, all entry fields and buttons on the Collection Page will be disabled
42.	Param6	ANS	20	Y	Payer ID Information ID Number - Payer Identification Number ID Type - 1(New IC) - 2(Old IC) - 3(Passport Number) - 4(Business Registration) - 5(Others) Please refer to below example; - 1|123456789012 Not Supported: Slash(/) Ampersand(&) Apostrophe(')
43.	Param7	ANS	50	Y	FPX e-Mandate Information Maximum Frequency - The maximum number of times that customer wish to be deducted on this particular Order/Transaction. Mode of Frequency - The type of frequency customer wish the account to be deducted.

					DL - Daily WK - Weekly MT - Monthly YR - Yearly Effective Date - The effective date for the first payment to be deducted from customer's account.(DDMMYY) Example as per below; - 9|MT|101018

Req? – *indicator on whether the fields/param is mandatory or not*
Y – Yes
N – No
C – Conditional

3.2 Enrollment Response (Payment Gateway → Merchant System)

Upon payment process completion, the following fields will be returned from Payment Gateway to Merchant System's (MerchantReturnURL or MerchantApprovalURL/MerchantUnapprovalURL) in order to complete an end-to-end payment process:

No.	Field	Data Type	Max Length	Req?	Description
1.	TransactionType	A	7	Y	Follows request
2.	PymtMethod	A	3	Y	Payment Method DD – Direct Debit
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 amount for transaction (should customer did not amend the amount in the Online Direct Debit Form)
7.	CurrencyCode	N	3	Y	Follows request
8.	HashValue	AN	100	Y	Message digest value calculated by Payment Gateway in hexadecimal string using SHA256 hash algorithm

					Re: Section 2.13.1.2
9.	HashValue2	AN	100	Y	HIGHLY RECOMMENDED to verify this message digest value calculated by Payment Gateway in hexadecimal string using SHA256 hash algorithm Re: Section 2.13.1.2
10.	TxnID	AN	30	Y	Unique Transaction ID or Reference Code assigned by Payment Gateway for this transaction
11.	BuyerBank	AN	10	N	Bank branch code that Buyer opened
12.	BuyerName	AN	40	N	Buyer Name
12.	TxnStatus	N	4	Y	Re: Section 3.2
13.	AuthCode	AN	12	N	Authorization Code returned by bank
14.	TxnMessage	AN	255	N	Message that briefly explains the response
17.	Param6	ANS	50	C	Follows request
18.	Param7	ANS	50	C	Follows request

4 Direct Email Payment Link

2 API to be called, please find the information below:

4.1 1st API to get authentication token

Test URL: https://test2pay.ghl.com/IPGPortalAPI2/api/User/PostUserLogin

Method: POST

No	Field	Data type	Max length	Req?	Description
1.	ulcversion	Int		Y	Must sent in '3'
2.	ulcuserlogin	Alphanumeric	50	Y	Portal user id
3.	ulcpwd	Alphanumeric	50	Y	Portal password

Sample Request Data:

```
{
   "ulcversion": "3",
   "ulcuserlogin": "test007",
   "ulcpwd": "N6NrF0sT"
}
```

Sample Response Data:

```
{
   "uuserlogin": "test007",
   "ulogintoken": "ffSCDhO5uBpBjWSlIUgsP3o0tKc=",
   "udomainshortname": "FM",
   "ucurrency": "MYR",
   "merchantname": null,
   "merchantlogourl": null,
   "ulastupdated": "23/11/2018 14:41:01",
   "ustatus": 0,
   "uremark": "",
   "ugroupdesc": null,
   "udisplayname": null,
   "uentity": [
      ""
   ],
   "menus": null
}
```

4.2 2nd API to initiate an Email Payment Link

After obtaining the response from 1st API, ustatus = 0 then only proceed with 2nd API calling

Test URL: https://test2pay.ghl.com/IPGPortalAPI2/api/Services/PostSendEmailPaymentLink

Method: POST

No	Field	Data type	Max length	Req?	Description
1.	epltoken	Alphanumeric	50	Y	Login token from first API
2.	eplcustomername	Alphanumeric	99	Y	Alphabets only, special character not allowed
3.	eplcustomercontact	Number	99	Y	
4.	eplcustomeremail	Email	50	Y	Valid email only
5.	eplorderdesc	Alphanumeric	99	Y	special character not allowed
6.	eplordernumber	Alphanumeric	19	Y	special character not allowed
7.	eplcurrency	Alphabet only	3	Y	special character not allowed
8.	eplamount	Decimal	Decimal(18, 2)	Y	Eg: 1.00
9.	eplreminder	Int		optional	0 – NO, 1 – YES (if don't want to set reminder no need to send in)
10.	eplreminderday	Int		optional	1 – 6 only (if eplreminder = 1, this have to send in, else it will set to 3)
11.	eplserviceid	Alphanumeric	3	optional	Merchant ID

Sample Request Data:


```
{
   "epltoken": "ffSCDhO5uBpBjWSlIUgsP3o0tKc=",
   "eplcustomername": "Test 007",
   "eplcustomercontact": "01234567890",
   "eplcustomeremail": "test123@ghl.com",
   "eplorderdesc": "TEST",
   "eplordernumber": "TEST1",
   "eplcurrency": "MYR",
   "eplamount": "1.00"
}
```


Sample Response Data:


```
{
   "AccessControl": {
      "respCode": 0,
      "respMessage": ""
   },
   "queryResult": {
      "paymentlink": "https://test2pay.ghl.com/IPGSG/payment.aspx?TokenType=BNB&CustEmail=foongmei.hon@ghl.com&Token=nh7uGbaQw0Aaa+jgJg4Rvg&CustPhone=01234567890"
,
      "paymentlinkbulk": null,
      "recipients": [
         "test123@ghl.com",
         "test123@ghl.com"
      ],
      "respCode": 0,
      "respMessage": null
   }
}
```

5 Appendix

5.1Transaction Type

TransactionType	Description
SALE	Payment transaction
AUTH	Authorization transaction
CAPTURE	Capture transaction
QUERY	Status inquiry transaction
RSALE	Reversal transaction
REFUND	Refund transaction
SETTLE	Settlement transaction

5.2Payment/Capture Transaction Status

TxnStatus	Description
0	Transaction successful
1	Transaction failed
2	Transaction pending

5.3Query 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

5.4 Reversal/Refund Transaction Status

TxnStatus	Description
0	Reversal/Refund success
1	Reversal/Refund failed, original transaction could be still under processing or failed due to other reasons like rejected by bank; Merchant System can send a Query to get actual payment status. If payment status from Query response is transaction successful, then only determine to whether proceed with sending Reversal/Refund. This will minimize the possibility of Reversal/Refund failed due to original transaction still under processing
2	Reversal/Refund pending, merchant system can retry Reversal/Refund
-1	Original transaction not found based on ServiceID, PaymentID, Amount and CurrencyCode, merchant system can stop Reversal/Refund
-2	Internal system error encountered during Reversal/Refund processing, Merchant System can retry Reversal/Refund

5.5 Settlement Transaction Status

TxnStatus	Description
0	Settlement success
1	Settlement failed
2	Settlement pending
-2	Internal system error encountered during Settlement processing

5.6 Currency Code

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

5.7 Language Code

Language Code	Language
EN	English (default language)
MS	Malay
TH	Thai
ZH	Chinese

5.8 Country Code

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