

INTRODUCTION

Description

Update Log

How to Read this Document

Use Cases for this API

Make Payment

Status Enquiry

Cancel & Refund

Order Confirmation

GETTING STARTED

How to Connect

API Gateway URL

API Authentication

User Identification

Connection Security

Message Security

Sign & Encrypt

Decrypt & Verify

Summary

How to make API request

with Plain Message

with Data Encryption

Data Type Overview

FAQ

SSL Connection

Message Encryption

JOSE Framework

API OPERATIONS

Payments

Payment Page Redirect API

Payment Status Enquiry API

Order Cancellation API

Refund API

Callback Payment Notification API

API SCHEMA

Schema Definitions

commonRespObj

paymentReqModel

pay\_rqt\_bxn\_Obj

pay\_rqt\_system\_Obj

pay\_rqt\_payment\_Obj

pay\_rqt\_merchant\_Obj

pay\_rqt\_customer\_Obj

pay\_rqt\_order\_Obj

descriptionsObj

pay\_rqt\_other\_Obj

udfsObj

paymentRespModel

pay\_rpn\_bxn\_Obj

pay\_rpn\_system\_Obj

enquiryReqModel

enq\_rqt\_bxn\_Obj

enq\_rqt\_merchant\_Obj

enquiryRespModel

enq\_rpn\_sys\_Obj

enq\_rpn\_bxn\_Obj

enq\_rpn\_payment\_Obj

enq\_rpn\_cc\_Obj

enq\_rpn\_hpp\_Obj

enq\_rpn\_other\_Obj

enq\_rpn\_refund\_Obj

cancelReqModel

cancel\_rqt\_bxn\_Obj

cancel\_rqt\_merchant\_Obj

cancelRespModel

cancel\_rpn\_sys\_Obj

cancel\_rpn\_bxn\_Obj

cancel\_rpn\_payment\_Obj

refundReqModel

refund\_rqt\_bxn\_Obj

refund\_rqt\_merchant\_Obj

refundRespModel

refund\_rpn\_sys\_Obj

refund\_rpn\_bxn\_Obj

refund\_rpn\_payment\_Obj

statusRtnReqModel

notif\_rqt\_bxn\_Obj

notif\_rqt\_merchant\_Obj

notif\_rqt\_payment\_Obj

notif\_rqt\_cc\_Obj

notif\_rqt\_hpp\_Obj

notif\_rqt\_other\_Obj

statusRtnRespModel

REFERENCE

Lifecycle of Cryptographic Keys

Key Generation & Exchange

Key Maintenance

Key Renewal

Payment Options

Download Swagger

DISCLAIMER

Disclaimer

API Specification for Malaysia Cards and Alternate Payment Methods

Version: 1.6

Description

This document introduces the **OpenAPI specification** describing the REST APIs of HSBC's ASP Omni Collection for Malaysia Cards and Alternate Payment Methods.

The target audience of this document are Developers, Business Analysts and other Project Team Members.

Update Log

- [Dec 16, 2021] **v1.6** Revised several content sections
- [Sep 2, 2021] **v1.5** Added the support of Installment Payment
- [Oct 20, 2020] **v1.4**
  - Added request field `token` to support Tokenization
  - Added more possible values to field `payment_option` and extend length to 20
- [Jul 14, 2020] **v1.3** Updated content sections [How to Connect](#) and [How to make API request](#)
- [Apr 22, 2020] **v1.2**
  - Moved response field `bank_auth_code` from object `payment` to `cc` at Enquiry API and Callback Notification
  - Removed response field `bank_auth_code` at Cancel and Refund API
  - Removed possible value `2` of field `txnStatus` at `notif_rqt_bxn_Obj`
- [Mar 6, 2020] **v1.1** Added new field `gateway_txn_id` in:
  - the response message of Payment Status Enquiry API;
  - the response message of Order Cancellation API;
  - the response message of Refund API;
  - and the request message of Callback Payment Notification API
- [Dec 6, 2019] **v1.0** Initial Version

How to Read this Document

This document walks through the API listing the key functions by section: [API Usage Flow](#), [API Connectivity](#), and [API Operation](#). There is also a [FAQ](#) and a list of [Schema Definitions](#) used by API operations.

This document has links to subsequent sections. For example, when you visit the section API Operation, it has links to the data model or schemas containing the data and status codes definitions.

Use Cases for this API

The HSBC Omni Collection provides a wide range of online payment solutions - enabling online merchants to process payments including credit / debit card, e-Wallet and Internet Banking (see the table below). The platform supports implementations with either websites or mobile applications.

To access the online secured payment gateway and make payments, the Merchant uses the API (API-generated) or manually creates a URL link in the Merchant Portal (Merchant Portal-generated). See details in the next section.

Using our APIs services, you can build your own eCommerce website and accept payments using the following payment channels:

Make Payment with our supported Payment Channels

These are our supported Payment Channels:

Payment Channel	Brand / Option	
Credit / Debit Card	<ul style="list-style-type: none"><li>Visa</li><li>MasterCard</li><li>Diners Club</li><li>China UnionPay</li><li>American Express</li><li>JCB</li></ul>	
	<ul style="list-style-type: none"><li>Boost</li><li>AiPay</li><li>TouchNGo</li><li>GrabPay</li><li>WeChatPay</li><li>Vcash</li><li>FavePay</li></ul>	
Internet Banking	<ul style="list-style-type: none"><li>Maybank</li><li>CIMB</li><li>PBe</li><li>RHB</li><li>HongLeong</li><li>Alliance Bank</li><li>Bank Islam</li><li>HSBC</li></ul>	<ul style="list-style-type: none"><li>AmBank Group</li><li>RAKYAT</li><li>Bank Muamalat</li><li>UOB</li><li>Standard Chartered</li><li>OCBC</li><li>KFH</li><li>AFFINBank</li></ul>

API Use Case (API-Generated)

INTRODUCTION

- Description
- Update Log
- How to Read this Document
- Use Cases for this API
- Make Payment
- Status Enquiry
- Cancel & Refund
- Order Confirmation

GETTING STARTED

- How to Connect
  - API Gateway URL
  - API Authentication
  - User Identification
  - Connection Security
  - Message Security
    - Sign & Encrypt
    - Decrypt & Verify
  - Summary
- How to make API request
  - with Plain Message
  - with Data Encryption
- Data Type Overview
- FAQ
  - SSL Connection
  - Message Encryption
  - JOSE Framework

API OPERATIONS

- Payments
  - Payment Page Redirect API
  - Payment Status Enquiry API
  - Order Cancellation API
  - Refund API
  - Callback Payment Notification API

API SCHEMA

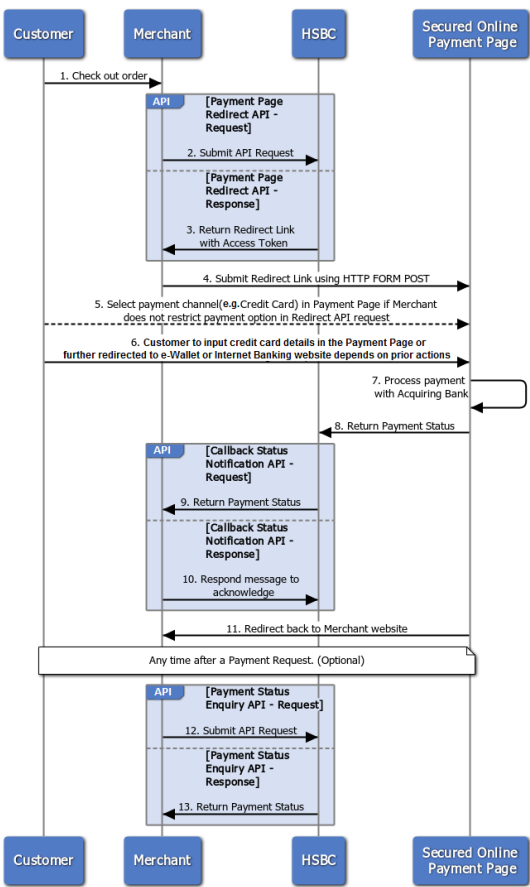
- Schema Definitions
  - commonRespObj
  - paymentReqModel
    - pay\_rqt\_bxn\_Obj
    - pay\_rqt\_system\_Obj
    - pay\_rqt\_payment\_Obj
    - pay\_rqt\_merchant\_Obj
    - pay\_rqt\_customer\_Obj
    - pay\_rqt\_order\_Obj
    - descriptionsObj
    - pay\_rqt\_other\_Obj
    - udfsObj
  - paymentRespModel
    - pay\_rpn\_bxn\_Obj
    - pay\_rpn\_system\_Obj
    - enquiryReqModel
      - enq\_rqt\_bxn\_Obj
      - enq\_rqt\_merchant\_Obj
    - enquiryRespModel
      - enq\_rpn\_sys\_Obj
      - enq\_rpn\_bxn\_Obj
      - enq\_rpn\_payment\_Obj
      - enq\_rpn\_cc\_Obj
      - enq\_rpn\_hpp\_Obj
      - enq\_rpn\_other\_Obj
      - enq\_rpn\_refund\_Obj
    - cancelReqModel
      - cancel\_rqt\_bxn\_Obj
      - cancel\_rqt\_merchant\_Obj
    - cancelRespModel
      - cancel\_rpn\_sys\_Obj
      - cancel\_rpn\_bxn\_Obj
      - cancel\_rpn\_payment\_Obj
    - refundReqModel
      - refund\_rqt\_bxn\_Obj
      - refund\_rqt\_merchant\_Obj
    - refundRespModel
      - refund\_rpn\_sys\_Obj
      - refund\_rpn\_bxn\_Obj
      - refund\_rpn\_payment\_Obj
    - statusRtnReqModel
      - notif\_rqt\_bxn\_Obj
      - notif\_rqt\_merchant\_Obj
      - notif\_rqt\_payment\_Obj
      - notif\_rqt\_cc\_Obj
      - notif\_rqt\_hpp\_Obj
      - notif\_rqt\_other\_Obj
    - statusRtnRespModel

REFERENCE

- Lifecycle of Cryptographic Keys
  - Key Generation & Exchange
  - Key Maintenance
  - Key Renewal
- Payment Options
- Download Swagger

DISCLAIMER

Disclaimer



- The Customer conducts a checkout process on the merchant's website.
- The Merchant submits a **Payment Page Redirect API** request to HSBC.
- HSBC returns a JSON response which embeds the Redirect Link of the Secured Online Payment Page with an access token inside the field `redirectLink`. The redirect link is in a **HTML FORM POST** format. More details are covered in the **Payment Page Redirect API**.
- The Merchant submits the redirect link using a **HTML FORM POST**. It redirects the Merchant website to the Secure Online Payment Page.
- The Customer can select different credit / debit card brands, e-Wallet or Internet Banking (see the table below) in the Payment Gateway - providing the Merchant does not restrict it by passing a value in the API request field `payment_option`.
- | Option              | Scenario   |
|---------------------|--|
| Credit / Debit Card | The Customer inputs Credit Card details in the Payment Page and will be further redirected to the 3D Secure (3DS) Page (Only if 3DS is enabled) for inputting One-Time password. |
| e-Wallet            | The Customer is further redirected to the e-Wallet web page for completing the payment process   |
| Internet Banking    | The Customer is further redirected to the Internet Bank web page for completing the payment process  |
- The payment page securely connects to the bank's backend systems to process the payment.
- HSBC receives the payment status once it is updated from the backend system.
- HSBC triggers a **Callback Payment Notification API** and sends the payment status back to the Merchant.

**NOTICE:**  
This server-to-server Notification is only sent out for a successful payment case. The Merchant can define their URL endpoint in request field `notifyurl` in **Payment Page Redirect API**

- To acknowledge, the Merchant sends a response to the Callback API. Failure to return a correct response triggers a Notification resend mechanism.
- A redirect is sent back to merchant website once the payment process is completed in the Payment Gateway.

**NOTE:**  
The Merchant can define the redirect URL using the request field `redirecturl` in **Payment Page Redirect API**.  
  
This redirection returns payment information (e.g. unique Transaction Reference Number) which enables the Merchant to create a more dynamic order confirmation page. However, unlike other standard OpenAPIs, this information is returned in **FORM POST** data. As a result, the Merchant can choose whether or not to process this data and it does not require a response. Details are covered in the **Order Confirmation** Section.

- The Merchant can optionally submit a **Payment Status Enquiry API** at any time after a payment request is submitted. This is useful when the Merchant finds no acknowledge message returned after a certain period of time.
- HSBC returns the latest payment status according to the transaction reference number the Merchant provided.

API Use Case (Merchant Portal-Generated)

INTRODUCTION

- Description
- Update Log
- How to Read this Document
- Use Cases for this API
  - Make Payment
  - Status Enquiry
  - Cancel & Refund
  - Order Confirmation

GETTING STARTED

- How to Connect
  - API Gateway URL
  - API Authentication
  - User Identification
  - Connection Security
    - Message Security
    - Sign & Encrypt
    - Decrypt & Verify
    - Summary
- How to make API request
  - with Plain Message
  - with Data Encryption
- Data Type Overview
- FAQ
  - SSL Connection
  - Message Encryption
  - JOSE Framework

API OPERATIONS

- Payments
  - Payment Page Redirect API
  - Payment Status Enquiry API
  - Order Cancellation API
  - Refund API
  - Callback Payment Notification API

API SCHEMA

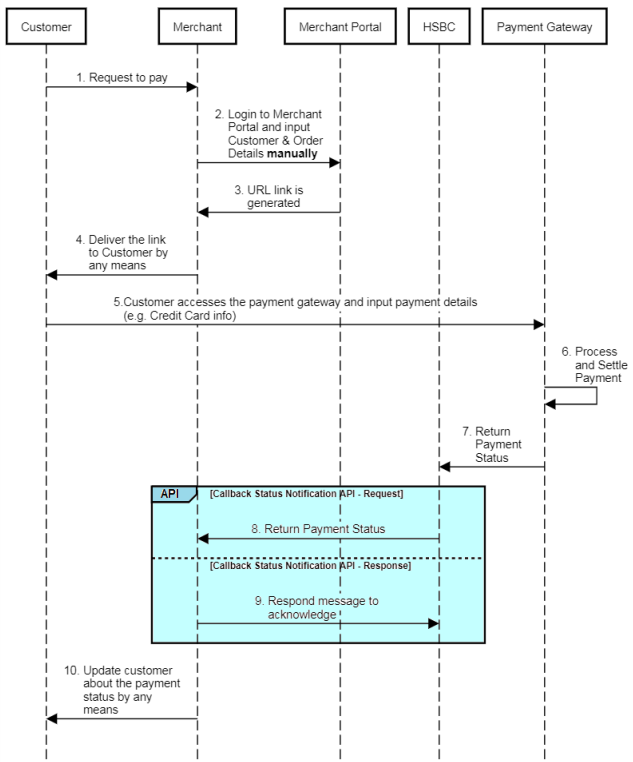
- Schema Definitions
  - commonRespObj
  - paymentReqModel
    - pay\_rqt\_txn\_Obj
    - pay\_rqt\_system\_Obj
    - pay\_rqt\_payment\_Obj
    - pay\_rqt\_merchant\_Obj
    - pay\_rqt\_customer\_Obj
    - pay\_rqt\_order\_Obj
    - descriptionsObj
    - pay\_rqt\_other\_Obj
    - udfsObj
  - paymentRespModel
    - pay\_rpn\_txn\_Obj
    - pay\_rpn\_system\_Obj
  - enquiryReqModel
    - enq\_rqt\_txn\_Obj
    - enq\_rqt\_merchant\_Obj
  - enquiryRespModel
    - enq\_rpn\_sys\_Obj
    - enq\_rpn\_txn\_Obj
    - enq\_rpn\_payment\_Obj
    - enq\_rpn\_cc\_Obj
    - enq\_rpn\_lpp\_Obj
    - enq\_rpn\_other\_Obj
    - enq\_rpn\_refund\_Obj
  - cancelReqModel
    - cancel\_rqt\_txn\_Obj
    - cancel\_rqt\_merchant\_Obj
  - cancelRespModel
    - cancel\_rpn\_sys\_Obj
    - cancel\_rpn\_txn\_Obj
    - cancel\_rpn\_payment\_Obj
  - refundReqModel
    - refund\_rqt\_txn\_Obj
    - refund\_rqt\_merchant\_Obj
  - refundRespModel
    - refund\_rpn\_sys\_Obj
    - refund\_rpn\_txn\_Obj
    - refund\_rpn\_payment\_Obj
  - statusRtnReqModel
    - notif\_rqt\_txn\_Obj
    - notif\_rqt\_merchant\_Obj
    - notif\_rqt\_payment\_Obj
    - notif\_rqt\_cc\_Obj
    - notif\_rqt\_lpp\_Obj
    - notif\_rqt\_other\_Obj
  - statusRtnRespModel

REFERENCE

- Lifecycle of Cryptographic Keys
  - Key Generation & Exchange
  - Key Maintenance
  - Key Renewal
- Payment Options
- Download Swagger

DISCLAIMER

Disclaimer



- The Customer requests to pay.
- The Merchant logs into the Merchant Portal and inputs customer and order details.

**NOTE:**  
Please contact your HSBC support to create a Merchant Portal Account

- The Merchant Portal generates a URL link.
- The Merchant shares this link to the Customer in a secure way.
- Depending on the particular payment option, the Customer accesses the online payment gateway by clicking the link and then inputting payment details.
- The Payment is processed and settled on the backend systems.
- HSBC receives payment status as soon as the payment process is completed.
- HSBC triggers a [Callback Payment Notification API](#) and sends the payment status back to the Merchant.

**NOTE:**  
This server-to-server Notification is only sent out for a successful payment case, the URL endpoint of Merchant is pre-defined in their Merchant Profile in HSBC.

- The Merchant responds to the API with an acknowledge. Failure to return a correct response triggers a Notification resend mechanism.
- The Merchant shares the payment status with the customer in secure way.

## Check Status Feature

The Omni collection provides a feature for the merchant to check the status of each transaction. To implement Check Status, please refer to the [Status Enquiry API](#).

## Cancel & Refund

To cancel an existing order whose payment transaction is still unsettled, the Merchant can request the [Cancel API](#).

To refund a settled transaction (Settled on both issuing and acquiring bank), the Merchant can request the [Refund API](#). HSBC accepts Full Refund and multiple Partial Refund.

## Order Confirmation

In the above API use case flow, the final step is to redirect the Payment Page back to the Merchant website. The Merchant can build a dynamic Order Confirmation Page with payment details, where the details can be retrieved from the asynchronous [Callback Payment Notification API](#).

## How to Connect

API Connectivity refers to all measures and their components that establishes a connection between HSBC - the API Provider, and the Merchant - the API Consumer.

	Definition	Components
API Authentication	HTTP BASIC Authentication	<ul style="list-style-type: none"><li>Username</li><li>Password</li></ul>
User Identification	Locate API Gateway Policy of the corresponding user	<ul style="list-style-type: none"><li>Client ID</li><li>Client Secret</li></ul>
Connection Security	A Merchant Profile	<ul style="list-style-type: none"><li>Merchant ID</li><li>Merchant Profile</li></ul>
Message Security	HTTPS Connection (TLS 1.2) and Network Whitelisting	<ul style="list-style-type: none"><li>SSL Certificate</li><li>Network Whitelist</li></ul>
Message Security	Digital Signing and Data Encryption	<ul style="list-style-type: none"><li>A pair of Private Key &amp; Public Key Certificate (PKI Model)</li><li>JWS Key ID</li><li>JWE Key ID</li></ul>

GETTING STARTED

- How to Connect
- API Gateway URL
- API Authentication
- User Identification
- Connection Security
- Message Security
- Sign & Encrypt
- Decrypt & Verify
- Summary
- How to make API request
- with Plain Message
- with Data Encryption
- Data Type Overview
- FAQ
- SSL Connection
- Message Encryption
- JOSE Framework

API OPERATIONS

- Payments
- Payment Page Redirect API
- Payment Status Enquiry API
- Order Cancellation API
- Refund API
- Callback Payment Notification API

API SCHEMA

- Schema Definitions
- commonRespObj
- paymentReqModel
- pay\_rqt\_txn\_Obj
- pay\_rqt\_system\_Obj
- pay\_rqt\_payment\_Obj
- pay\_rqt\_merchant\_Obj
- pay\_rqt\_customer\_Obj
- pay\_rqt\_order\_Obj
- descriptionsObj
- pay\_rqt\_other\_Obj
- udfsObj
- paymentRespModel
- pay\_rpn\_txn\_Obj
- pay\_rpn\_system\_Obj
- enquiryReqModel
- enq\_rqt\_txn\_Obj
- enq\_rqt\_merchant\_Obj
- enquiryRespModel
- enq\_rpn\_sys\_Obj
- enq\_rpn\_txn\_Obj
- enq\_rpn\_payment\_Obj
- enq\_rpn\_cc\_Obj
- enq\_rpn\_hpp\_Obj
- enq\_rpn\_other\_Obj
- enq\_rpn\_refund\_Obj
- cancelReqModel
- cancel\_rqt\_txn\_Obj
- cancel\_rqt\_merchant\_Obj
- cancelRespModel
- cancel\_rpn\_sys\_Obj
- cancel\_rpn\_txn\_Obj
- cancel\_rpn\_payment\_Obj
- refundReqModel
- refund\_rqt\_txn\_Obj
- refund\_rqt\_merchant\_Obj
- refundRespModel
- refund\_rpn\_sys\_Obj
- refund\_rpn\_txn\_Obj
- refund\_rpn\_payment\_Obj
- statusRtnReqModel
- notif\_rqt\_txn\_Obj
- notif\_rqt\_merchant\_Obj
- notif\_rqt\_payment\_Obj
- notif\_rqt\_cc\_Obj
- notif\_rqt\_hpp\_Obj
- notif\_rqt\_other\_Obj
- statusRtnRespModel

REFERENCE

- Lifecycle of Cryptographic Keys
- Key Generation & Exchange
- Key Maintenance
- Key Renewal
- Payment Options
- Download Swagger

DISCLAIMER

Disclaimer

API Gateway URL

You need to include this before each API endpoint to make API calls.

Production
https://cmb-api.hsbc.com.hk/glc-mobilecoll-mcmy-ea-merchantservices-prod-proxy/v1
Sandbox
https://devclustercmb.api.p2g.netd2.hsbc.com.hk/glc-mobilecoll-mcmy-ea-merchantservices-cert-proxy/v1

API Authentication

Username & Password	
Purpose	All APIs are authorized using <code>Basic Authorization</code>
Components	<ul style="list-style-type: none"><li>Username</li><li>Password</li></ul>
Where to get it?	Delivered by HSBC via secure email during onboarding procedure
Implementation	In HTTP header: <code>Authorization: Basic [Base64-encoded Credential]</code>

Client ID & Client Secret	
Purpose	API Gateway locates the corresponding policy of the specific API consumer
Components	<ul style="list-style-type: none"><li>Client ID</li><li>Client Secret</li></ul>
Where to get it?	Delivered by HSBC via secure email during onboarding procedure
Implementation	In HTTP header: <code>x-hsbc-client-id: [Client ID]</code>  In HTTP header: <code>x-hsbc-client-secret: [Client Secret]</code>

User Identification

Merchant Profile & Merchant ID	
Purpose	<ul style="list-style-type: none"><li>Merchant Profile contains all necessary information from a Merchant in order to enable payment service.</li><li>Merchant ID is used for Merchant identification in each API call.</li></ul>
Components	<ul style="list-style-type: none"><li>Merchant Profile</li><li>Merchant ID</li></ul>
Where to get it?	<ul style="list-style-type: none"><li>Set up by HSBC team after collect information from Merchant</li><li>Delivered by HSBC via secure email during onboarding procedure</li></ul>
Implementation	<i>nil</i>  In HTTP header: <code>x-hsbc-msg-encrypt-id: [Merchant ID]+[JWS ID]+[JWE ID]</code>

Connection Security

SSL Certificate & Network Whitelist	
Purpose	<ul style="list-style-type: none"><li>Request HSBC API over HTTPS connection (TLS 1.2)</li><li>Accept Callback API request over HTTPS connection (TLS 1.2)</li></ul>
Components	<ul style="list-style-type: none"><li>Public SSL Certificate issued by HSBC</li><li>Merchant's web server or domain whose HTTPS connection is enabled</li><li>Network Whitelist on HSBC system</li></ul>
Where to get it?	<ul style="list-style-type: none"><li>Downloaded automatically by Browsers or API Tools, if any problem found, please contact HSBC</li></ul> <i>nil</i> <i>nil</i>
Implementation	<i>nil</i> <i>nil</i> <ul style="list-style-type: none"><li>Merchant's domain URL will be configured in HSBC's network whitelist by HSBC team</li></ul>

Message Security - Data Encryption and Signing

In addition to the Transport Layer Security, HSBC adopts additional security - Data Encryption on the message being passed across the session. This serves as a type of locked briefcase containing the data (the API message) within the HTTPS "tunnel". In other words, the communication has double protection.

! DID YOU KNOW?

JavaScript Object Signing and Encryption (JOSE™), is a framework that secures information transferred between parties. To achieve this, the JOSE framework provides a collection of specifications, including JSON Web Signature (JWS™) and JSON Web Encryption (JWE™).

HSBC uses **JWS** to sign message payloads, and **JWE** to encrypt the signed message. These are created by using the [Private Key & Public Key Certificate \(PKI Model\)](#).

Private Key & Public Key Certificate (PKI Model)	
Purpose	<ul style="list-style-type: none"><li>Digitally sign a API request message</li><li>Decrypt a API response message</li><li>Encrypt the signed API request message</li><li>Verify a signed API response message</li></ul>
Components	<ul style="list-style-type: none"><li>Private Key issued by Merchant</li><li>Public Key Certificate issued by HSBC</li></ul>
Where to get it?	<ul style="list-style-type: none"><li>Created by any Public Key Infrastructure (PKI) toolkits, such as Keytool™ and OpenSSL™. Technical detail is in <a href="#">here</a></li><li>Exchanged with HSBC with the Public Key Certificate issued by Merchant</li></ul>
Implementation	Please see the technical detail in <a href="#">here</a>

! NOTE:

Technically, an X.509 certificate can serve as a SSL Certificate as well as a Public Key Certificate for Data Encryption. However, for segregation of certificate usage, HSBC recommends that the Merchant uses a different X.509 Certificate for Data Encryption. Moreover, the Public Key Certificate does not have to be CA-signed. However, if the Merchant decides to enhance security, a CA-Signed Certificate is acceptable.

GETTING STARTED

- How to Connect
- API Gateway URL
- API Authentication
- User Identification
- Connection Security
- Message Security
- Sign & Encrypt
- Decrypt & Verify
- Summary
- How to make API request
- with Plain Message
- with Data Encryption
- Data Type Overview
- FAQ
- SSL Connection
- Message Encryption
- JOSE Framework

API OPERATIONS

- Payments
- Payment Page Redirect API
- Payment Status Enquiry API
- Order Cancellation API
- Refund API
- Callback Payment Notification API

API SCHEMA

- Schema Definitions
- commonRespObj
- paymentReqModel
- pay\_rqt\_bxn\_Obj
- pay\_rqt\_system\_Obj
- pay\_rqt\_payment\_Obj
- pay\_rqt\_merchant\_Obj
- pay\_rqt\_customer\_Obj
- pay\_rqt\_order\_Obj
- descriptionsObj
- pay\_rqt\_other\_Obj
- udfsObj
- paymentRespModel
- pay\_rpn\_bxn\_Obj
- pay\_rpn\_system\_Obj
- enquiryReqModel
- enq\_rqt\_bxn\_Obj
- enq\_rqt\_merchant\_Obj
- enquiryRespModel
- enq\_rpn\_sys\_Obj
- enq\_rpn\_bxn\_Obj
- enq\_rpn\_payment\_Obj
- enq\_rpn\_cc\_Obj
- enq\_rpn\_hpp\_Obj
- enq\_rpn\_other\_Obj
- enq\_rpn\_refund\_Obj
- cancelReqModel
- cancel\_rqt\_bxn\_Obj
- cancel\_rqt\_merchant\_Obj
- cancelRespModel
- cancel\_rpn\_sys\_Obj
- cancel\_rpn\_bxn\_Obj
- cancel\_rpn\_payment\_Obj
- refundReqModel
- refund\_rqt\_bxn\_Obj
- refund\_rqt\_merchant\_Obj
- refundRespModel
- refund\_rpn\_sys\_Obj
- refund\_rpn\_bxn\_Obj
- refund\_rpn\_payment\_Obj
- statusRtnReqModel
- notif\_rqt\_bxn\_Obj
- notif\_rqt\_merchant\_Obj
- notif\_rqt\_payment\_Obj
- notif\_rqt\_cc\_Obj
- notif\_rqt\_hpp\_Obj
- notif\_rqt\_other\_Obj
- statusRtnRespModel

REFERENCE

- Lifecycle of Cryptographic Keys
- Key Generation & Exchange
- Key Maintenance
- Key Renewal
- Payment Options
- Download Swagger

DISCLAIMER

Disclaimer

keyID of JWS™ & JWE™		
Purpose	• The unique identifier to bind Merchant's Private Key in order to create a JWS object - a signed Message Payload	• The unique identifier to bind HSBC's Public Key Certificate in order to create a JWE object - an encrypted JWS object
Components	• keyID of JWS™	• keyID of JWE™
Where to get it?	• Mutual agreed between Merchant and HSBC	• Mutual agreed between Merchant and HSBC
Implementation	Define in program coding, see demo in <a href="#">here</a>	

! NOTE:

For security purposes, [HSBC's Public Key Certificate](#) and its associated [keyID](#) is renewed every year and a Certificate Renewal process is triggered. More detail is covered in the section [Key Renewal](#)

How to Sign and Encrypt Outgoing Message

Every message sent to HSBC must be signed and encrypted. From the Merchant's perspective, an **Outgoing Message** means:

- the Request Message of a Service API, or
- the Respond Message of a Callback API.

To help you understand how to construct a Signed and Encrypted Message, let's take the Java program below as an example. Don't worry if you are not familiar with Java, the idea is to let you know the steps and the required components:

! NOTE: These Java codes are for demonstration only - it's not *plug and play*.

```
private JWSObject signMessage(String messagePayload, KeyStore ks, String keyAlias, String keyPw)
    throws UnrecoverableKeyException, KeyStoreException, NoSuchAlgorithmException, JOSEException {
#1 Payload payload = new Payload(messagePayload);

#2 JWHeader header = new JWHeader
    .Builder(JWAAlgorithm.RS256)
    .keyID("0001")
    .customParam("iat", Instant.now().getEpochSecond()).build();
#3 JWSObject jwsObject = new JWSObject(header, payload);

#4 PrivateKey privateKey = (PrivateKey) ks.getKey(keyAlias, keyPw.toCharArray());
JWSSigner signer = new RSASSASigner(privateKey);
#5 jwsObject.sign(signer);

    return jwsObject;
}
```

- Prepare your **Message Payload**, that is, the plain `json` request message.
- Create a **JWS Header** where the parameters are as follows:

```
{
  "alg": "RS256",           //Signing Algorithm is RS256
  "kid": "0001",           //Put your own Key ID value, "0001" is just an example
  "iat": "1625587913"      //Issued At - the time this request is sent, in Unix Time format
}
```

- Create a **JWS Object** by combining JWS Header and Message Payload.
- Retrieve your **Private Key** as the signer.
- Create a **Signed JWS Object** by signing it with the Private Key.

Next, **Encrypt** the Signed JWS Object:

```
private JWEObject getEncryptedJWEObject(JWSObject jwsObject, RSAPublicKey key)
    throws JOSEException {
#1 Payload jwepayload = new Payload(jwsObject.serialize());

#2 JWHeader jweheader = new JWHeader.Builder(JWAAlgorithm.RSA_OAEP_256, EncryptionMethod.A128GCM)
#3 JWEObject jweObject = new JWEObject(jweheader, jwepayload);

#4 JWEEncrypter encrypter = new RSAEncrypter(key);
#5 jweObject.encrypt(encrypter);

    return jweObject;
}
```

- Prepare your **JWE Payload**, that is, the **Signed JWS Object**.
- Create the **JWE Header**. The algorithm used to encrypt the message body is `A128GCM` while the algorithm used to encrypt the encryption key is `RSA_OAEP_256`. **JWE keyID** is `0002`.
- Create the **JWE Object** by combining JWE Header and JWE Payload.
- Retrieve the **HSBC's Public Key** as the encrypter.
- Create the **Encrypted JWE Object** by encrypting it with HSBC's Public Key.

You are now ready to put the Encrypted JWE Object in the message body (*you may need to first **serialize** it into String format, depends on your program code design*) of any API call.

How to Decrypt Message and Verify Signature of an Incoming Message

Every message sent from HSBC must be decrypted and verified. From the Merchant's perspective, an **Incoming Message** means:

- the Respond Message of a Service API, or
- the Request Message of a Callback API.

Let's look into the following example to see how to decrypt a response message from HSBC:

```
private String decryptMessage(String respMsgPayload, KeyStoreFactory keyStore)
    throws KeyStoreException, NoSuchAlgorithmException, CertificateException, IOException,
    java.text.ParseException, UnrecoverableKeyException, JOSEException {
#1 JWEObject jweObject = JWEObject.parse(respMsgPayload);

#2 PrivateKey privateKey = (PrivateKey) keyStore.getPrivateKey("merchant_private_key_alias");

    JWDecrypter decrypter = new RSADecrypter(privateKey);
#3 jweObject.decrypt(decrypter);

#4 String signedMessage = jweObject.getPayload().toString();
    return signedMessage;
}
```

- Create an **Encrypted JWE Object** by parsing the encrypted response message payload.
- Retrieve the **Private Key** as the decrypter.
- Decrypt the JWE Object using your Private Key.
- Get the **Signed Message** from the decrypted JWE Object.

You are now able to extract the plain `json` message, but first you **must** verify the signature to guarantee data integrity.

```
private String verifySignature(String signedMessage, KeyStore ks, String keyAlias)
    throws KeyStoreException, JOSEException, ParseException {
#1 JWSObject jwsObject = JWSObject.parse(signedMessage);

    Certificate certificate = ks.getCertificate(keyAlias);
#2 JWSVerifier verifier = new RSASSAVerifier((RSAPublicKey) certificate.getPublicKey());

#3 if (!jwsObject.verify(verifier)) {
    throw new ValidationException("Invalid Signature");
}
#4 return jwsObject.getPayload().toString();
}
```

GETTING STARTED

API OPERATIONS

API SCHEMA

REFERENCE

DISCLAIMER

1. Create a **JWS Object** by parsing the **Signed Message**.
2. Retrieve the **HSBC's Public Key** as the verifier.
3. Verify the signed JWS Object. Invoke error handling if an invalid signature is found (*depends on your code design*).
4. Get the plain **json** message for further actions.

Summary

Components \ Steps	Message Signing	Message Encryption	Message Decryption	Verify Signature
JWS Object	Signing Algorithm: <b>RS256</b>			
JWE Object		JWE Algorithm: <b>RSA_OAEP_256</b>		
		Encryption Method: <b>A128GCM</b>		
KeyID	<b>0002</b>	<b>0002</b>		
Merchant's Private Key	Used as <b>Signer</b>		Used as <b>Decrypter</b>	
HSBC's Public Key		Used as <b>Encrypter</b>		Used as <b>Verifier</b>

How to Make an API Request

An API request can be submitted without Message Encryption, in case you want to:

- learn about the basic API Call;
- test API connectivity before spending substantial development effort on Message Encryption.

Data encryption is a required data security imposed by HSBC standards. The Merchant has to invoke the encryption logic before moving to Production and must be fully tested during the testing phase.

Make Your API Request with Plain Messages

!

**NOTE:**  
In the Sandbox Environment you can skip message encryption. However, this is for testing purpose only.

**Submit an example API request using cURL™**

cURL™ is a simple command-line tool that enables you to make any HTTP request. Merchant can choose any other GUI tool such as Postman™ and SoapUI™.

**Step 1.** Run this command on your platform:

**POST**

```
#1 curl -X POST "https://devclustercmb.ap1.p2g.netd2.hsbc.com.hk/g lcm-mobilecoll-mcmy-ea-merchantserv
#2 -H "message-encrypt: false"
#3 -H "Authorization: Basic ew9ic191c2VybWFrZTpw5b3VyX3Bhc3N3b3Jk"
#4 -H "x-HSBC-client-id: 8b915a4f5b5047f091f210e2232b5ced"
#5 -H "x-HSBC-client-secret: 1bb450a541dc416d8601685f9583c606"
#6 -H "x-HSBC-msg-encrypt-id: 42298549900001+0001+0002"
#7 -H "Content-Type: application/json"
#8 -d '{"txnRef": "\PAY-QJZV956664", "merId": "\42298549900001"}'
```

1. Submit the **POST** request to the API URL endpoint.
2. Set the secret header **message\_encrypt: false** to indicate this API request is without message encryption. This header is only applicable in Sandbox environment.
3. Put the **Basic Authorization** in HTTP header **Authorization**.
4. Put the **Client ID** in HTTP header **x-HSBC-client-id**.
5. Put the **Client Secret** in HTTP header **x-HSBC-client-secret**.
6. Put the **Merchant ID**, the **JWS ID** and the **JWE ID** in HTTP header **x-HSBC-msg-encrypt-id** respectively.
7. Set the **Content-Type** to JSON format.
8. Plain **json** message payload.

**GET**

```
#1 curl -X GET "https://devclustercmb.ap1.p2g.netd2.hsbc.com.hk/g lcm-mobilecoll-mcmy-ea-merchantserv
#2 -H "message-encrypt: false"
#3 -H "Authorization: Basic ew9ic191c2VybWFrZTpw5b3VyX3Bhc3N3b3Jk"
#4 -H "x-HSBC-client-id: 8b915a4f5b5047f091f210e2232b5ced"
#5 -H "x-HSBC-client-secret: 1bb450a541dc416d8601685f9583c606"
#6 -H "x-HSBC-msg-encrypt-id: 42298549900001+0001+0002"
#7 -H "Content-Type: application/json"
```

1. Submit the **GET** request to the API URL endpoint.
2. Set the secret header **message\_encrypt: false** to indicate this API request is without message encryption. This header is only applicable in Sandbox environment.
3. Put the **Basic Authorization** in HTTP header **Authorization**.
4. Put the **Client ID** in HTTP header **x-HSBC-client-id**.
5. Put the **Client Secret** in HTTP header **x-HSBC-client-secret**.
6. Put the **Merchant ID**, the **JWS ID** and the **JWE ID** in HTTP header **x-HSBC-msg-encrypt-id** respectively.
7. Set **Content-Type** to JSON format.

**Step 2.** Receive the response message in plain **json** format.

Making API Request with Message Encryption

**Step 1.** Run this cURL™ command on your platform:

**POST**

```
#1 curl -X POST "https://devclustercmb.ap1.p2g.netd2.hsbc.com.hk/g lcm-mobilecoll-mcmy-ea-merchantserv
#2 -H "Authorization: Basic ew9ic191c2VybWFrZTpw5b3VyX3Bhc3N3b3Jk"
#3 -H "x-HSBC-client-id: 8b915a4f5b5047f091f210e2232b5ced"
#4 -H "x-HSBC-client-secret: 1bb450a541dc416d8601685f9583c606"
#5 -H "x-HSBC-msg-encrypt-id: 42298549900001+0001+0002"
#6 -H "Content-Type: application/json"
#7 -d '{"txnRef": "\PAY-QJZV956664", "merId": "\42298549900001"}'
```

1. Submit the **POST** request to the API URL endpoint. Any **{id}** adhered in the URL must be encrypted.
2. Put the **Basic Authorization** in HTTP header **Authorization**.
3. Put the **Client ID** in HTTP header **x-HSBC-client-id**.
4. Put the **Client Secret** in HTTP header **x-HSBC-client-secret**.
5. Put the **Merchant ID**, the **JWS ID** and the **JWE ID** in HTTP header **x-HSBC-msg-encrypt-id** respectively.
6. Set the **Content-Type** to JSON format.
7. The Encrypted Message Payload.

**GET**

```
#1 curl -X GET "https://devclustercmb.ap1.p2g.netd2.hsbc.com.hk/g lcm-mobilecoll-mcmy-ea-merchantserv
#2 -H "Authorization: Basic ew9ic191c2VybWFrZTpw5b3VyX3Bhc3N3b3Jk"
#3 -H "x-HSBC-client-id: 8b915a4f5b5047f091f210e2232b5ced"
#4 -H "x-HSBC-client-secret: 1bb450a541dc416d8601685f9583c606"
#5 -H "x-HSBC-msg-encrypt-id: 42298549900001+0001+0002"
#6 -H "Content-Type: application/json"
```

INTRODUCTION

Description

Update Log

How to Read this Document

Use Cases for this API

Make Payment

Status Enquiry

Cancel & Refund

Order Confirmation

GETTING STARTED

How to Connect

API Gateway URL

API Authentication

User Identification

Connection Security

Message Security

Sign & Encrypt

Decrypt & Verify

Summary

How to make API request

with Plain Message

with Data Encryption

Data Type Overview

FAQ

SSL Connection

Message Encryption

JOSE Framework

API OPERATIONS

Payments

Payment Page Redirect API

Payment Status Enquiry API

Order Cancellation API

Refund API

Callback Payment Notification API

API SCHEMA

Schema Definitions

commonRespObj

paymentReqModel

pay\_rqt\_bxn\_Obj

pay\_rqt\_system\_Obj

pay\_rqt\_payment\_Obj

pay\_rqt\_merchant\_Obj

pay\_rqt\_customer\_Obj

pay\_rqt\_order\_Obj

descriptionsObj

pay\_rqt\_other\_Obj

udfsObj

paymentRespModel

pay\_rpn\_bxn\_Obj

pay\_rpn\_system\_Obj

enquiryReqModel

enq\_rqt\_bxn\_Obj

enq\_rqt\_merchant\_Obj

enquiryRespModel

enq\_rpn\_sys\_Obj

enq\_rpn\_bxn\_Obj

enq\_rpn\_payment\_Obj

enq\_rpn\_cc\_Obj

enq\_rpn\_lpp\_Obj

enq\_rpn\_other\_Obj

enq\_rpn\_refund\_Obj

cancelReqModel

cancel\_rqt\_bxn\_Obj

cancel\_rqt\_merchant\_Obj

cancelRespModel

cancel\_rpn\_sys\_Obj

cancel\_rpn\_bxn\_Obj

cancel\_rpn\_payment\_Obj

refundReqModel

refund\_rqt\_bxn\_Obj

refund\_rqt\_merchant\_Obj

refundRespModel

refund\_rpn\_sys\_Obj

refund\_rpn\_bxn\_Obj

refund\_rpn\_payment\_Obj

statusRtnReqModel

notif\_rqt\_bxn\_Obj

notif\_rqt\_merchant\_Obj

notif\_rqt\_payment\_Obj

notif\_rqt\_cc\_Obj

notif\_rqt\_lpp\_Obj

notif\_rqt\_other\_Obj

statusRtnRespModel

REFERENCE

Lifecycle of Cryptographic Keys

Key Generation & Exchange

Key Maintenance

Key Renewal

Payment Options

Download Swagger

DISCLAIMER

Disclaimer

1. Submit the `GET` request to the API URL endpoint. Any `{id}` adhered in the URL must be encrypted.
2. Put the `Basic Authorization` in HTTP header `Authorization`.
3. Put the `Client ID` in HTTP header `x-HSBC-client-id`.
4. Put the `Client Secret` in HTTP header `x-HSBC-client-secret`.
5. Put the `Merchant ID`, the `JWS ID` and the `JWE ID` in HTTP header `x-HSBC-msg-encrypt-id` respectively.
6. Set the `Content-Type` to JSON format.

**NOTE:**  
Data Encryption invokes compulsory prerequisites, such as [JOSE library](#) and program coding, please make sure the section on [Message Security](#) has been gone through thoroughly.

**Step 2.** For a successful request (HTTP Status Code 200), an encrypted response message is returned, otherwise, a plain `json` with failure message is returned.

## Data Type Overview

Data Type Control:

Data Type	Allowed Characters	Definition & Important Notice
String (For general field)	AlphaNumeric and Symbols	A General field is a field which is <b>NOT</b> a critical field. HSBC will character check on all string fields received in order to tackle security vulnerability, such as Cross-site Scripting. HSBC recommend you to try use AlphaNumeric only for most cases.
String (For critical field)	<div><div>0-9</div><div>A-Z</div><div>a-z</div><div>.</div></div>	<p>A Critical field is used to be either a key or search criteria in HSBC backend system and hence tight restriction is applied to permitted characters.</p> <p>Moreover, the starting and ending space of the string value will be trimmed before stored in HSBC system. For example, string <code>" example 12 34 "</code> will be trimmed to <code>"example 12 34"</code>.</p> <p><b>List of Critical Fields:</b></p> <div><div>txnRef</div><div>merId</div><div>rfdRef</div></div>
Integer	<div>0-9</div>	Instead of having Max Length check for String, integer range will be checked, e.g. <code>0 ≤ x ≤ 9999</code>

Field Mandatory Control:

Field Mandatory Type	Definition & Important Notice
Mandatory	<p>Annotated with <code>required</code> tag in field definition section.</p> <p>Field &amp; value must be present in the request with valid <code>JSON</code> format.</p>
Optional	<p>Annotated with <code>optional</code> tag in field definition section.</p> <p>If you don't want to pass fields that are optional, your handler should not pass neither empty strings (<code>("example":"")</code>) nor blank value (<code>("example":" ")</code>).</p>
Conditional	<p>Annotated with <code>conditional</code> tag in field definition section.</p> <p>Required under a specific condition whose logic is always provided in the field definition if it is a Conditional Field.</p>

Time Zone Control:

Aspect	Format	Definition & Important Notice
In Request Message	<code>yyyy-MM-dd'T'HH:mm:ssZ</code>	Time zone is expected to be <code>GMT+8</code> (Malaysia local time). Merchant is required to perform any necessary time zone conversion before submit request if needed.
In Response Message	<code>yyyy-MM-dd'T'HH:mm:ss±hh:mm</code>	<p>Timezone returned in <code>apiGateway</code> object is generated from HSBC API Gateway which located in Cloud and hence is calculated in <code>GMT+8</code>.</p> <p>On the other hand, time field in <code>response</code> object will be returned together with timezone information. For more details, please read each field definition carefully.</p>

## FAQ

### SSL Connection Questions

#### Where can I find the HSBC SSL server certificates?

The Merchant developer can export SSL server certificates installed in your browser. To achieve this, visit the domain of the corresponding API endpoint in your browser. For example, to get the SSL certificate of sandbox environment, use the domain name <https://devcluster.api.p2g.netd2.HSBC.com.hk/>

However, in production, we provide a certificate and require TLS 1.2 implementation.

### Message Encryption Questions

#### What certificates do I need to work with Message Encryption in HSBC's sandbox and production environments?

A self-sign certificate is acceptable. However, if the Merchant decides to enhance security, a CA-Signed Certificate is also acceptable.

### Javascript Object Signing and Encryption (JOSE) Framework Questions

#### Where can I get more information about JOSE Framework?

If you want to fully understand the framework, you can read [here](#) for more details.

Please note these urls or websites do not belong to HSBC, use them at your own discretion. By clicking these urls or websites signifies you accept these terms and conditions.

#### Where can I download JOSE libraries for development?

For your reference, you may find the following JOSE libraries of different programming languages.

- [Ruby](#)
- [Python](#)
- [PHP](#)
- [Java](#)
- [Node](#)
- [.NET](#)

Please note these urls or websites do not belong to HSBC, use them at your own discretion. By clicking these urls or websites signifies you accept these terms and conditions.

## Payments

Contains resource collections for conventional digital payments, enquiry, cancel and refund, etc.

### Payment Page Redirect API

POST /payment/pageRedirect

#### DESCRIPTION

This API returns a redirect link of the Secured Online Payment Page that aims to redirect Merchant's browser to the payment page. Customer then input all other necessary information (such as Credit Card details) in that page to complete the payment.

#### How to do Redirection

Merchant is required to use HTTP Form POST to submit the redirect link which is presented in a `HTML Form` format together with an access token. Below is a sample, please be noticed any data modification inside the form is not allowed. Otherwise, the data integrity checking will block the connection from accessing the online payment page.

```
<script language="javascript">window.onload=function(){document.pay_form.submit();}</script>
<form id="pay_form" name="pay_form" action="https://test2pay.ghl.com/IPGSG/Payment.aspx" method="post">
<input name="TransactionType" type="hidden" id="TransactionType" value="SALE" />
<input name="PyntMethod" type="hidden" id="PyntMethod" value="ANY" />
<input name="ServiceID" type="hidden" id="ServiceID" value="HBC" />
<input name="HashValue" type="hidden" id="HashValue" value="dba46976c106a17e8d506a1243b30499c9ae826be" />
/* More Input Fields Here... */
</form>
```

#### REQUEST PARAMETERS

<b>Authorization</b> <div>required in header</div>	BASIC [Base64-encoded Credential]
<b>x-hsbc-client-id</b> <div>required in header</div>	[Client ID]
<b>x-hsbc-client-secret</b> <div>required in header</div>	[Client Secret]
<b>x-hsbc-msg-encrypt-id</b> <div>optional in header</div>	[Merchant ID]+[JWS ID]+[JWE ID]
<b>Content-Type</b> <div>required in header</div>	application/json

#### REQUEST BODY

<b>paymentReqModel</b>	<i>Data Encryption is enforced. API Schema intends to demonstrate the skeleton of the message payload only.</i>
------------------------	---

#### RESPONSES

<b>200 OK</b> <b>paymentRespModel</b>	Successful operation.  <i>Data Encryption is enforced. API Schema intends to demonstrate the skeleton of the message payload only.</i>
<b>400 Bad Request</b> <b>commonRespObj</b>	Missing or invalid Parameters.
<b>403 Forbidden</b>	Authorization credentials are missing or invalid.
<b>404 Not Found</b>	Empty resource/resource not found.
<b>500 Internal Server Error</b>	The request failed due to an internal error.

Request Content-Types: application/json

Request Example

```
{
  "transaction": {
    "txnRef": "ORD-438UL748T6"
  },
  "system": {
    "default_lang": "MS",
    "redirectUrl": "https://www.example.com/redirectBacktoMerchantSite",
    "notifyUrl": "https://www.example.com/receiveNotification"
  },
  "payment": {
    "country": "MYR",
    "currency": "MYR",
    "payment_option": "ANY",
    "amount": 170000,
    "payment_expiry": 780,
    "token": "S0FG12345FGH23456"
  },
  "merchant": {
    "merId": "HBC"
  },
  "customer": {
    "customer_name": "Kamat bin Muda",
    "customer_email": "customer.name@example.com",
    "customer_phone": "60121235678"
  },
  "order": {
    "description": "Proceed check out for your order #ORD-438UL748T6",
    "descriptions": [
      {
        "product_name": "Product Item 1",
        "product_id": "PRO-ASDF-1234",
        "unitAmt": 10000,
        "unit": 2,
        "subAmt": 20000
      },
      {
        "product_name": "Product Item 2",
        "product_id": "PRO-JHGF-9876",
        "unitAmt": 50000,
        "unit": 3,
        "subAmt": 150000
      }
    ]
  },
  "other": {
    "udfs": [
      {
        "definition": "Product Image in Base64 format",
        "value": "iVBORw0KGgoAAAANSUHEU..."
      },
      {
        "definition": "Special Notes from Customer",
        "value": "Customer is a non-smoker"
      }
    ]
  }
}
```

Response Content-Types: application/json

Response Example (200 OK)

```
{
  "api_gw": {
    "messageId": "89817674-da00-4883",
    "returnCode": "200",
    "returnReason": "Successful operation",
    "sentTime": "2016-11-15T10:00:00.000Z",
    "responseTime": "2016-11-15T10:00:00.000Z"
  },
  "response": {
    "transaction": {
      "txnRef": "ORD-438UL748T6"
    },
    "system": {
      "sysCode": "0000000",
      "sysMsg": "Request Successful",
      "sysDateTime": "2016-11-15T10:00:00.000Z",
      "redirectLink": "<Encoded_Redirect_Submit_Form>"
    }
  }
}
```

Response Example (400 Bad Request)

```
{
  "messageId": "89817674-da00-4883",
  "returnCode": "400",
  "returnReason": "Error Message Here",
  "sentTime": "2016-11-15T10:00:00.000Z",
  "responseTime": "2016-11-15T10:00:00.000Z"
}
```

### Payment Status Enquiry API

POST /payment/enquiry

#### DESCRIPTION



INTRODUCTION

Description

Update Log

How to Read this Document

Use Cases for this API

Make Payment

Status Enquiry

Cancel & Refund

Order Confirmation

GETTING STARTED

How to Connect

API Gateway URL

API Authentication

User Identification

Connection Security

Message Security

Sign & Encrypt

Decrypt & Verify

Summary

How to make API request

with Plain Message

with Data Encryption

Data Type Overview

FAQ

SSL Connection

Message Encryption

JOSE Framework

API OPERATIONS

Payments

Payment Page Redirect API

Payment Status Enquiry API

Order Cancellation API

Refund API

Callback Payment Notification API

API SCHEMA

Schema Definitions

commonRespObj

paymentReqModel

pay\_rqt\_txn\_Obj

pay\_rqt\_system\_Obj

pay\_rqt\_payment\_Obj

pay\_rqt\_merchant\_Obj

pay\_rqt\_customer\_Obj

pay\_rqt\_order\_Obj

descriptionsObj

pay\_rqt\_other\_Obj

udfsObj

paymentRespModel

pay\_rpn\_txn\_Obj

pay\_rpn\_system\_Obj

enquiryReqModel

enq\_rqt\_txn\_Obj

enq\_rqt\_merchant\_Obj

enquiryRespModel

enq\_rpn\_sys\_Obj

enq\_rpn\_txn\_Obj

enq\_rpn\_payment\_Obj

enq\_rpn\_cc\_Obj

enq\_rpn\_hpp\_Obj

enq\_rpn\_other\_Obj

enq\_rpn\_refund\_Obj

cancelReqModel

cancel\_rqt\_txn\_Obj

cancel\_rqt\_merchant\_Obj

cancelRespModel

cancel\_rpn\_sys\_Obj

cancel\_rpn\_txn\_Obj

cancel\_rpn\_payment\_Obj

refundReqModel

refund\_rqt\_txn\_Obj

refund\_rqt\_merchant\_Obj

refundRespModel

refund\_rpn\_sys\_Obj

refund\_rpn\_txn\_Obj

refund\_rpn\_payment\_Obj

statusRtnReqModel

notif\_rqt\_txn\_Obj

notif\_rqt\_merchant\_Obj

notif\_rqt\_payment\_Obj

notif\_rqt\_cc\_Obj

notif\_rqt\_hpp\_Obj

notif\_rqt\_other\_Obj

statusRtnRespModel

REFERENCE

Lifecycle of Cryptographic Keys

Key Generation & Exchange

Key Maintenance

Key Renewal

Payment Options

Download Swagger

DISCLAIMER

Disclaimer

Merchant can optionally initiate payment status enquiry at any time after a payment request is submitted. This is used when Merchant wants to check payment status any time after a payment request or find no acknowledge message returned after a certain period of time. HSBC Mobile Collection will return the latest transaction status according to the transaction reference number Merchant provides.

REQUEST PARAMETERS

<b>Authorization</b> <small>required</small> in header	BASIC [Base64-encoded Credential]
<b>x-hsbc-client-id</b> <small>required</small> in header	[Client ID]
<b>x-hsbc-client-secret</b> <small>required</small> in header	[Client Secret]
<b>x-hsbc-msg-encrypt-id</b> <small>optional</small> in header	[Merchant ID]+[JWS ID]+[JWE ID]
<b>Content-Type</b> <small>required</small> in header	application/json

REQUEST BODY

<a href="#">enquiryReqModel</a>	<i>Data Encryption is enforced. API Schema intends to demonstrate the skeleton of the message payload only.</i>
---------------------------------	---

RESPONSES

<b>200 OK</b> <a href="#">enquiryRespModel</a>	Successful operation.  <i>Data Encryption is enforced. API Schema intends to demonstrate the skeleton of the message payload only.</i>
<b>400 Bad Request</b> <a href="#">commonRespObj</a>	Missing or invalid Parameters.
<b>403 Forbidden</b>	Authorization credentials are missing or invalid.
<b>404 Not Found</b>	Empty resource/resource not found.
<b>500 Internal Server Error</b>	The request failed due to an internal error.

Request Content-Types: application/json

Request Example

```
{  "transaction": {    "txnRef": "ORD-438UL748T6"  },  "merchant": {    "merId": "HBC"  }}
```

Response Content-Types: application/json

Response Example (200 OK)

```
{  "api_gw": {    "messageId": "89817674-da00-4883",    "returnCode": "200",    "returnReason": "Successful operation",    "sentTime": "2016-11-15T10:00:00.000Z",    "responseTime": "2016-11-15T10:00:00.000Z"  },  "response": {    "system": {      "sysCode": "000000",      "sysMsg": "Request Successful"    },    "transaction": {      "txnRef": "ORD-438UL748T6",      "txnStatus": "0",      "txnMessage": "Request was processed successfully.",      "gateway_txn_id": "HBC0000GHLPAYNUTR23563"    },    "payment": {      "amount": 170000,      "currency": "MYR",      "payment_datetime": "2019-12-12T14:10:25+08:00",      "payment_option": "CC",      "issuing_bank": "ITS BANK",      "acquirer": "GPay"    },    "cc": {      "brand": "VISA",      "mcn": "409587XXXXX4977",      "bank_auth_code": "HBC000"    },    "hpp": {      "hppPeriod": 12,      "hppFrequency": "MONTHLY"    },    "other": {      "udfs": [        {          "definition": "Product Image in Base64 format",          "value": "iVBORw0KGgoAAAANSUHEU..."        },        {          "definition": "Special Notes from Customer",          "value": "Customer is a non-smoker"        }      ]    },    "refund": {      "rfdTotalAmount": 100000    }  } }
```

Response Example (400 Bad Request)

```
{  "messageId": "89817674-da00-4883",  "returnCode": "400",  "returnReason": "Error Message Here",  "sentTime": "2016-11-15T10:00:00.000Z",  "responseTime": "2016-11-15T10:00:00.000Z"}
```

Request Content-Types: application/json

Request Example

```
{  "transaction": {    "txnRef": "ORD-438UL748T6"  },  "merchant": {    "merId": "HBC"  } }
```

Response Content-Types: application/json

Response Example (200 OK)

```
{  "api_gw": {
```

Order Cancellation API

**POST** /payment/cancel

DESCRIPTION

This API is used to send a cancellation (a.k.a Sale Reversal) request for an unsettled transaction.

REQUEST PARAMETERS

<b>Authorization</b> <small>required</small> in header	BASIC [Base64-encoded Credential]
<b>x-hsbc-client-id</b> <small>required</small> in header	[Client ID]
<b>x-hsbc-client-secret</b> <small>required</small> in header	[Client Secret]
<b>x-hsbc-msg-encrypt-id</b> <small>optional</small> in header	[Merchant ID]+[JWS ID]+[JWE ID]
<b>Content-Type</b> <small>required</small> in header	application/json

REQUEST BODY

<a href="#">cancelReqModel</a>	<i>Data Encryption is enforced. API Schema intends to demonstrate the skeleton of the message payload only.</i>
--------------------------------	---

RESPONSES

<b>200 OK</b> <a href="#">cancelRespModel</a>	Successful operation.  <i>Data Encryption is enforced. API Schema intends to demonstrate the skeleton of the message payload only.</i>
--	--

Payments

<b>400 Bad Request</b> commonRespObj	Missing or invalid Parameters.
<b>403 Forbidden</b>	Authorization credentials are missing or invalid.
<b>404 Not Found</b>	Empty resource/resource not found.
<b>500 Internal Server Error</b>	The request failed due to an internal error.

<b>200 OK</b> refundRespModel	Successful operation.
<b>400 Bad Request</b> commonRespObj	Missing or invalid Parameters.
<b>403 Forbidden</b>	Authorization credentials are missing or invalid.
<b>404 Not Found</b>	Empty resource/resource not found.
<b>500 Internal Server Error</b>	The request failed due to an internal error.

## Refund API

POST

/payment/refund

### DESCRIPTION

This API is used to send a refund request for a previously settled transaction. It supports both full and multiple partial refund. Before requesting a new partial refund, any prior partial refund request must have been settled.

### REQUEST PARAMETERS

<b>Authorization</b> required in header	BASIC [Base64-encoded Credential]
<b>x-hsbc-client-id</b> required in header	[Client ID]
<b>x-hsbc-client-secret</b> required in header	[Client Secret]
<b>x-hsbc-msg-encrypt-id</b> optional in header	[Merchant ID]+[JWS ID]+[JWE ID]
<b>Content-Type</b> required in header	application/json

### REQUEST BODY

refundReqModel	Data Encryption is enforced. API Schema intends to demonstrate the skeleton of the message payload only.
----------------	--

### RESPONSES

<b>200 OK</b> refundRespModel	Successful operation.
<b>400 Bad Request</b> commonRespObj	Missing or invalid Parameters.
<b>403 Forbidden</b>	Authorization credentials are missing or invalid.
<b>404 Not Found</b>	Empty resource/resource not found.
<b>500 Internal Server Error</b>	The request failed due to an internal error.

```
{
  "messageId": "89817674-da00-4883",
  "returnCode": "200",
  "returnReason": "Successful operation",
  "sentTime": "2016-11-15T10:00:00.000Z",
  "responseTime": "2016-11-15T10:00:00.000Z"
},
{
  "system": {
    "sysCode": "000000",
    "sysMsg": "Request Successful"
  },
  "transaction": {
    "txnRef": "ORD-438UL748T6",
    "txnStatus": "0",
    "txnMessage": "RSALE Success",
    "gateway_txn_id": "HBC0000GHLPAYCUI916784"
  },
  "payment": {
    "amount": 500000,
    "currency": "MYR",
    "payment_option": "CC",
    "issuing_bank": "ITS BANK",
    "acquirer": "GPay"
  }
}
```

#### Response Example (400 Bad Request)

```
{
  "messageId": "89817674-da00-4883",
  "returnCode": "400",
  "returnReason": "Error Message Here",
  "sentTime": "2016-11-15T10:00:00.000Z",
  "responseTime": "2016-11-15T10:00:00.000Z"
}
```

Request Content-Types: application/json

#### Request Example

```
{
  "transaction": {
    "txnRef": "ORD-438UL748T6",
    "rfdRef": "RFD-438UL748T6",
    "rfdAmount": 100000,
    "currency": "MYR"
  },
  "merchant": {
    "merId": "HBC"
  }
}
```

Response Content-Types: application/json

#### Response Example (200 OK)

```
{
  "apl_gw": {
    "messageId": "89817674-da00-4883",
    "returnCode": "200",
    "returnReason": "Successful operation",
    "sentTime": "2016-11-15T10:00:00.000Z",
    "responseTime": "2016-11-15T10:00:00.000Z"
  },
  "response": {
    "system": {
      "sysCode": "000000",
      "sysMsg": "Request Successful"
    },
    "transaction": {
      "txnRef": "ORD-438UL748T6",
      "rfdRef": "RFD-438UL748T6",
      "txnStatus": "0",
      "txnMessage": "REFUND Success",
      "gateway_txn_id": "HBC0000GHLPAYOUNI364019"
    },
    "payment": {
      "rfdAmount": 100000,
      "currency": "MYR",
      "payment_option": "CC",
      "issuing_bank": "ITS BANK",
      "acquirer": "GPay"
    }
  }
}
```

#### Response Example (400 Bad Request)

```
{
  "messageId": "89817674-da00-4883",
  "returnCode": "400",
  "returnReason": "Error Message Here",
  "sentTime": "2016-11-15T10:00:00.000Z",
  "responseTime": "2016-11-15T10:00:00.000Z"
}
```

## Payment Status Notification API

POST

/<Callback URL predefined by Merchant>

### DESCRIPTION

Payment status will be pushed back to Merchant by asynchronous callback once Mobile Collection completes reconciliation with bank and receives a payment result.

!

**Implementation**  
This is a Callback API. HSBC will trigger this API call and defines the interface with OpenAPI standard. Merchant is required to provide implementation.

!

**Retry Mechanism**  
If no success response is received, up to 4 retries will be triggered in every 2 minutes. Maximum 5 calls including

×

INTRODUCTION

Description

Update Log

How to Read this Document

Use Cases for this API

Make Payment

Status Enquiry

Cancel & Refund

Order Confirmation

GETTING STARTED

How to Connect

API Gateway URL

API Authentication

User Identification

Connection Security

Message Security

Sign & Encrypt

Decrypt & Verify

Summary

How to make API request

with Plain Message

with Data Encryption

Data Type Overview

FAQ

SSL Connection

Message Encryption

JOSE Framework

API OPERATIONS

Payments

Payment Page Redirect API

Payment Status Enquiry API

Order Cancellation API

Refund API

Callback Payment Notification API

API SCHEMA

Schema Definitions

commonRespObj

paymentReqModel

pay\_rqt\_bxn\_Obj

pay\_rqt\_system\_Obj

pay\_rqt\_payment\_Obj

pay\_rqt\_merchant\_Obj

pay\_rqt\_customer\_Obj

pay\_rqt\_order\_Obj

descriptionsObj

pay\_rqt\_other\_Obj

udfsObj

paymentRespModel

pay\_rpn\_bxn\_Obj

pay\_rpn\_system\_Obj

enquiryReqModel

enq\_rqt\_bxn\_Obj

enq\_rqt\_merchant\_Obj

enquiryRespModel

enq\_rpn\_sys\_Obj

enq\_rpn\_bxn\_Obj

enq\_rpn\_payment\_Obj

enq\_rpn\_cc\_Obj

enq\_rpn\_ipp\_Obj

enq\_rpn\_other\_Obj

enq\_rpn\_refund\_Obj

cancelReqModel

cancel\_rqt\_bxn\_Obj

cancel\_rqt\_merchant\_Obj

cancelRespModel

cancel\_rpn\_sys\_Obj

cancel\_rpn\_bxn\_Obj

cancel\_rpn\_payment\_Obj

refundReqModel

refund\_rqt\_bxn\_Obj

refund\_rqt\_merchant\_Obj

refundRespModel

refund\_rpn\_sys\_Obj

refund\_rpn\_bxn\_Obj

refund\_rpn\_payment\_Obj

statusRtnReqModel

notif\_rqt\_bxn\_Obj

notif\_rqt\_merchant\_Obj

notif\_rqt\_payment\_Obj

notif\_rqt\_cc\_Obj

notif\_rqt\_ipp\_Obj

notif\_rqt\_other\_Obj

statusRtnRespModel

REFERENCE

Lifecycle of Cryptographic Keys

Key Generation & Exchange

Key Maintenance

Key Renewal

Payment Options

Download Swagger

DISCLAIMER

Disclaimer

the 1st attempt.

Endpoint Definition  
Field `notificationURL` from [Payment Page Redirect API](#) will be used as URL endpoint of the corresponding transaction.

Exception Handling  
Only success case will be returned. Merchant can submit a [Payment Status Enquiry API](#) request if found no acknowledge message returned after a certain period of time.

REQUEST PARAMETERS

Content-Type	text/plain
<div><div>required</div></div> in header	

REQUEST BODY

<code>statusRtnReqModel</code>	<i>Data Encryption is enforced. API Schema intends to demonstrate the skeleton of the message payload only.</i>
--------------------------------	---

RESPONSES

<code>200 OK</code>	Successful operation.
<code>statusRtnRespModel</code>	<i>Data Encryption is enforced. API Schema intends to demonstrate the skeleton of the message payload only.</i>

Schema Definitions

commonRespObj: object

PROPERTIES

**messageId**: string range: (up to 36 chars) 

required

  
System generated unique message ID only for HSBC internal reference use

**returnCode**: string range: (up to 3 chars) 

required

  
System Return Code.

- This checking is on API Operational level, in other words, it checks upon Authorization, Connectivity and JSON Message Structure.

Possible Value	Definition
200	Successful operation
400	Bad Request (With detail message in field <code>returnReason</code> )
	Internal Error.
500	Important Notices: If any tier comes before the API Cloud Foundry is unavailable, such as the API Gateway, there will be no json respond message returned.  Furthermore, the respond message of 500 will be ignored by some common HTTP libraries, in such case, the respond message body can be considered as a hint for troubleshooting during development and testing phase.

**returnReason**: string range: (up to 200 chars) 

required

  
Corresponding Text message of returnCode

Corr. Return Code	Return Message Sample	Definition
		A successful API operation in terms of Authorization, Connectivity and valid JSON Message Structure.
200	Successful operation	Any checking failure on Business Logic level will be still considered a successful API operation yet the Business Logic checking result will be returned in <code>response</code> object.
400	Client ID - Merchant ID mapping is not correct/updated!	The binding of Client ID, Merchant ID and Merchant Public Certificate is incorrect or not up-to-date.
400	object has missing required properties <code>field name</code>	Fail to pass JSON Field Mandatory Check.
400	instance type <code>data type</code> does not match any allowed primitive type	Fail to pass JSON Field Type Check.
400	string <code>field value</code> is too long	Fail to pass JSON Field Max Length Check
400	instance failed to match at least one required schema among <code>no. of conditional field</code>	Fail to pass JSON Conditional Field Check.
500	java.net.ConnectException: Connection refused: connect	<b>Notices:</b> Message can be varied depended on the downstream systems which return this message. Yet, all reasons can be concluded into Internal Error or System Unavailable.

**sentTime**: string range: (up to 27 chars) 

required

  
Time of request received by HSBC system from client, only for HSBC internal reference use

**responseTime**: string range: (up to 27 chars) 

required

  
Time of HSBC system provides response to client, only for HSBC internal reference use

Request Content-Types: text/plain

Request Example

```
{
  "transaction": {
    "txnRef": "ORD-43BUL74BT6",
    "txnStatus": "0",
    "txnMessage": "Transaction Successful",
    "gateway_txn_id": "HBC0000GHLPAYDUG187609"
  },
  "merchant": {
    "merId": "HBC"
  },
  "payment": {
    "amount": 500000,
    "currency": "MYR",
    "payment_datetime": "2019-12-12T14:10:25+08:00",
    "payment_option": "cc",
    "issuing_bank": "ITS BANK",
    "acquirer": "GPay",
    "token": "SDFG12345FGH23456"
  },
  "cc": {
    "brand": "VISA",
    "mcn": "403587XXXXX4977",
    "bank_auth_code": "HBC000"
  },
  "ipp": {
    "ippPeriod": 12,
    "ippFrequency": "MONTHLY"
  },
  "other": {
    "udfs": [
      {
        "definition": "Product Image in Base64 format",
        "value": "IV8DRwKGG0AAAANSuHEU..."
      },
      {
        "definition": "Special Notes from Customer",
        "value": "Customer is a non-smoker"
      }
    ]
  }
}
```

Response Content-Types: application/json

Response Example (200 OK)

```
{
  "status": "SUCCESS"
}
```

Example

```
{
  "messageId": "B9817674-da00-4883",
  "returnCode": "200",
  "returnReason": "Successful operation",
  "sentTime": "2016-11-15T10:00:00.000Z",
  "responseTime": "2016-11-15T10:00:00.000Z"
}
```

## paymentReqModel: object

### PROPERTIES

**transaction:** `pay_rqt_txn_Obj` required

**system:** `pay_rqt_system_Obj` required

**payment:** `pay_rqt_payment_Obj` required

**merchant:** `pay_rqt_merchant_Obj` required

**customer:** `pay_rqt_customer_Obj` required

**order:** `pay_rqt_order_Obj` required

**other:** `pay_rqt_other_Obj` optional

## pay\_rqt\_txn\_Obj: object

### PROPERTIES

**txnRef:** `string` (Critical Field) range: (up to 20 chars) required

Unique transaction ID assigned by merchant

- No duplicate Transaction Reference is allowed

## pay\_rqt\_system\_Obj: object

### PROPERTIES

**default\_lang:** `string` enum: [ EN, MS, TH, ZH ] range: (up to 2 chars) optional

ISO 639-1 Language Code appeared in the payment gateway

Possible Value	Definition
EN	English (Default Language)
MS	Malay
TH	Thai
ZH	Chinese

**redirectUri:** `string` range: (up to 255 chars) required

Define URL endpoint for redirecting customer back to merchant site after completing the payment

**notifyUri:** `string` range: (up to 255 chars) required

Define URL endpoint for receiving server-to-server payment result notification from HSBC after payment completed

## pay\_rqt\_payment\_Obj: object

### PROPERTIES

**country:** `string` enum: [ MY ] range: (up to 2 chars) required

Country Code (Format: `ISO alpha-2`)

Possible Value	Definition
MY	Malaysia

**currency:** `string` enum: [ MYR, SGD, THB, CNY, PHP ] range: (up to 3 chars) required

Payment Currency (Format: `ISO 4217 Alpha`)

Possible Value	Definition
MYR	Malaysia Ringgit
SGD	Singapore Dollar
THB	Thai Baht
CNY	China Yuan
PHP	Philippine Peso

**payment\_option:** `string` range: (up to 20 chars) required

To restrict customer payment methods shown in the payment gateway. Please see all possible values [here](#).

**amount:** `integer` range: 1 ≤ x ≤ 999999999999999 required

Payment Amount in 2 decimal places regardless whether the currency has decimal places or not

**payment\_expiry:** `integer` range: 1 ≤ x ≤ 9999 required

### Example

```
{
  "transaction": {
    "txnRef": "ORD-438UL748T6"
  },
  "system": {
    "default_lang": "MS",
    "redirectUrl": "https://www.example.com/redirectBacktoMerchantSite",
    "notifyUrl": "https://www.example.com/receiveNotification"
  },
  "payment": {
    "country": "MY",
    "currency": "MYR",
    "payment_option": "AMY",
    "amount": 170000,
    "payment_expiry": 780,
    "token": "SDFG12345FGH23456"
  },
  "merchant": {
    "merId": "HBC"
  },
  "customer": {
    "customer_name": "Kamat bin Muda",
    "customer_email": "customer.name@example.com",
    "customer_phone": "60121235678"
  },
  "order": {
    "description": "Proceed check out for your order #ORD-438UL748T6",
    "descriptions": [
      {
        "product_name": "Product Item 1",
        "product_id": "PRO-ASDF-1234",
        "unitAmt": 10000,
        "unit": 2,
        "subAmt": 20000
      },
      {
        "product_name": "Product Item 2",
        "product_id": "PRO-JHGF-9876",
        "unitAmt": 50000,
        "unit": 3,
        "subAmt": 150000
      }
    ]
  },
  "other": {
    "udfs": [
      {
        "definition": "Product Image in Base64 format",
        "value": "iVBORw0KGgoAAAANSUHEU..."
      },
      {
        "definition": "Special Notes from Customer",
        "value": "Customer is a non-smoker"
      }
    ]
  }
}
```

### Example

```
{
  "txnRef": "ORD-438UL748T6"
}
```

### Example

```
{
  "default_lang": "MS",
  "redirectUrl": "https://www.example.com/redirectBacktoMerchantSite",
  "notifyUrl": "https://www.example.com/receiveNotification"
}
```

### Example

```
{
  "country": "MY",
  "currency": "MYR",
  "payment_option": "AMY",
  "amount": 170000,
  "payment_expiry": 780,
  "token": "SDFG12345FGH23456"
}
```



## INTRODUCTION

Description

Update Log

How to Read this Document

Use Cases for this API

Make Payment

Status Enquiry

Cancel & Refund

Order Confirmation

## GETTING STARTED

How to Connect

API Gateway URL

API Authentication

User Identification

Connection Security

Message Security

Sign & Encrypt

Decrypt & Verify

Summary

How to make API request

with Plain Message

with Data Encryption

Data Type Overview

FAQ

SSL Connection

Message Encryption

JOSE Framework

## API OPERATIONS

Payments

Payment Page Redirect API

Payment Status Enquiry API

Order Cancellation API

Refund API

Callback Payment Notification API

## API SCHEMA

Schema Definitions

commonRespObj

paymentReqModel

pay\_rqt\_bxn\_Obj

pay\_rqt\_system\_Obj

pay\_rqt\_payment\_Obj

pay\_rqt\_merchant\_Obj

pay\_rqt\_customer\_Obj

pay\_rqt\_order\_Obj

descriptionsObj

pay\_rqt\_other\_Obj

udfsObj

paymentRespModel

pay\_rpn\_bxn\_Obj

pay\_rpn\_system\_Obj

enquiryReqModel

enq\_rqt\_bxn\_Obj

enq\_rqt\_merchant\_Obj

enquiryRespModel

enq\_rpn\_sys\_Obj

enq\_rpn\_bxn\_Obj

enq\_rpn\_payment\_Obj

enq\_rpn\_cc\_Obj

enq\_rpn\_hpp\_Obj

enq\_rpn\_other\_Obj

enq\_rpn\_refund\_Obj

cancelReqModel

cancel\_rqt\_bxn\_Obj

cancel\_rqt\_merchant\_Obj

cancelRespModel

cancel\_rpn\_sys\_Obj

cancel\_rpn\_bxn\_Obj

cancel\_rpn\_payment\_Obj

refundReqModel

refund\_rqt\_bxn\_Obj

refund\_rqt\_merchant\_Obj

refundRespModel

refund\_rpn\_sys\_Obj

refund\_rpn\_bxn\_Obj

refund\_rpn\_payment\_Obj

statusRtnReqModel

notif\_rqt\_bxn\_Obj

notif\_rqt\_merchant\_Obj

notif\_rqt\_payment\_Obj

notif\_rqt\_cc\_Obj

notif\_rqt\_hpp\_Obj

notif\_rqt\_other\_Obj

statusRtnRespModel

## REFERENCE

Lifecycle of Cryptographic Keys

Key Generation & Exchange

Key Maintenance

Key Renewal

Payment Options

Download Swagger

## DISCLAIMER

Disclaimer

## paymentRespModel: object

### PROPERTIES

**api\_gw:** [commonRespObj](#) required

**response:** object required

### PROPERTIES

**transaction:** [pay\\_rpn\\_bxn\\_Obj](#) required

**system:** [pay\\_rpn\\_system\\_Obj](#) required

## pay\_rpn\_txn\_Obj: object

### PROPERTIES

**txnRef:** string ([Critical Field](#)) range: (up to 20 chars) required

Returning back Transaction ID/Reference

## pay\_rpn\_system\_Obj: object

### PROPERTIES

**sysCode:** string range: (up to 6 chars) required

Return Code of System Checking

Possible Value	Definition
000000	Request Successful of creating the Payment Page Submit Form
800010	Unsupported Payment Option Found
900030	Duplicate Transaction Reference

**sysMsg:** string range: (up to 128 chars) required

Corresponding Message of the System Return Code

**sysDatetime:** string range: (up to 25 chars) required

Time of sending out this request / response

- Server system time. A [GMT+8](#) timezone information is appended to the end of the timestamp to indicate this time is a Malaysia local time. Format: `|yyyy-MM-dd'T'HH:mm:ss±hh:mm|`

**redirectLink:** string range: (up to 5120 chars) conditional

Encoded Redirect Link with all form submit parameters

- Return only if this is a successful request

## enquiryReqModel: object

### PROPERTIES

**transaction:** [enq\\_rqt\\_bxn\\_Obj](#) required

**merchant:** [enq\\_rqt\\_merchant\\_Obj](#) required

## enq\_rqt\_txn\_Obj: object

### PROPERTIES

**txnRef:** string ([Critical Field](#)) range: (up to 20 chars) required

Pass Transaction Reference that refers to one specific transaction

## enq\_rqt\_merchant\_Obj: object

### PROPERTIES

**merId:** string ([Critical Field](#)) range: (up to 3 chars) required

Merchant ID

## enquiryRespModel: object

### PROPERTIES

**api\_gw:** [commonRespObj](#) required

**response:** object required

### PROPERTIES

**system:** [enq\\_rpn\\_sys\\_Obj](#) required

**transaction:** [enq\\_rpn\\_bxn\\_Obj](#) required

**payment:** [enq\\_rpn\\_payment\\_Obj](#) required

### Example

```
{
  "api_gw": {
    "messageId": "89817674-da00-4883",
    "returnCode": "200",
    "returnReason": "Successful operation",
    "sentTime": "2016-11-15T10:00:00.000Z",
    "responseTime": "2016-11-15T10:00:00.000Z"
  },
  "response": {
    "transaction": {
      "txnRef": "ORD-438UL748T6"
    },
    "system": {
      "sysCode": "000000",
      "sysMsg": "Request Successful",
      "sysDatetime": "2020-01-01T13:00:00+08:00",
      "redirectLink": "<Encoded_Redirect_Submit_Form>"
    }
  }
}
```

### Example

```
{
  "txnRef": "ORD-438UL748T6"
}
```

### Example

```
{
  "sysCode": "000000",
  "sysMsg": "Request Successful",
  "sysDatetime": "2020-01-01T13:00:00+08:00",
  "redirectLink": "<Encoded_Redirect_Submit_Form>"
}
```

### Example

```
{
  "transaction": {
    "txnRef": "ORD-438UL748T6"
  },
  "merchant": {
    "merId": "HBC"
  }
}
```

### Example

```
{
  "txnRef": "ORD-438UL748T6"
}
```

### Example

```
{
  "merId": "HBC"
}
```

### Example

```
{
  "api_gw": {
    "messageId": "89817674-da00-4883",
    "returnCode": "200",
    "returnReason": "Successful operation",
    "sentTime": "2016-11-15T10:00:00.000Z",
    "responseTime": "2016-11-15T10:00:00.000Z"
  },
  "response": {
    "system": {
      "sysCode": "000000",
```

**cc:** [enq\\_rpn\\_cc\\_Obj](#) conditional

Return only if this is a Credit Card payment

**ipp:** [enq\\_rpn\\_hpp\\_Obj](#) conditional

Return only if this is an Installment payment

**other:** [enq\\_rpn\\_other\\_Obj](#) optional

**refund:** [enq\\_rpn\\_refund\\_Obj](#) conditional

Return only if there is any prior refund request

## enq\_rpn\_sys\_Obj: object

### PROPERTIES

**sysCode:** string range: (up to 6 chars) required

Return Code of System Checking where the checking is on API operation level but not transaction level. e.g. Request Successful means the API operation is successful.

Possible Value	Definition
000000	Request Successful
900010	Transaction Record Not Found
999999	Request Fail

**sysMsg:** string range: (up to 128 chars) required

The corresponding message of System Return Code.

## enq\_rpn\_txn\_Obj: object

### PROPERTIES

**txnRef:** string (Critical Field) range: (up to 20 chars) required

Returning Unique Transaction Reference

**txnStatus:** string range: (up to 4 chars) required

Transaction Status

Possible Value	Definition
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 existed / not found
-2	Internal system error

**txnMessage:** string range: (up to 255 chars) required

Additional text message that explains the response

**gateway\_txn\_id:** string range: (up to 30 chars) required

Returning Unique Transaction ID or Reference Code assigned by Payment Gateway

## enq\_rpn\_payment\_Obj: object

### PROPERTIES

**amount:** integer range:  $1 \leq x \leq 9999999999999999$  required

Payment Amount in 2 decimal places regardless whether the currency has decimal places or not

!

NOTE: Do not have sign. For example, value `10000` means `100.00`

**currency:** string range: (up to 3 chars) required

Return Payment Currency (Format: `ISO 4217 Alpha`)

**payment\_datetime:** string range: (up to 25 chars) required

Returning Transaction time for the inward credit payment

- Bank system local time. A `GMT+8` timezone information is appended to the end of the timestamp to indicate this time is a Malaysia local time. Format: `yyyy-MM-dd'T'HH:mm:ss±hh:mm`

**payment\_option:** string range: (up to 20 chars) required

Return Payment Option

Possible Value	Definition
CC	Credit Card (Online 3D/Non3D)
DD	Direct Debit
WA	e-Wallet

```
    "sysMsg": "Request Successful"
  },
  "transaction": {
    "txnRef": "ORD-438UL748T6",
    "txnStatus": "0",
    "txnMessage": "Request was processed successfully.",
    "gateway_txn_id": "HBC0000GHLPAYNUTR023563"
  },
  "payment": {
    "amount": 170000,
    "currency": "MYR",
    "payment_datetime": "2019-12-12T14:10:25+08:00",
    "payment_option": "CC",
    "issuing_bank": "ITS BANK",
    "acquirer": "GPay"
  },
  "cc": {
    "brand": "VISA",
    "acn": "403587XXXXX4977",
    "bank_auth_code": "HBC000"
  },
  "ipp": {
    "ippPeriod": 12,
    "ippFrequency": "MONTHLY"
  },
  "other": {
    "udfs": [
      {
        "definition": "Product Image in Base64 format",
        "value": "iVBORw0KGgoAAAANSUHEU..."
      },
      {
        "definition": "Special Notes from Customer",
        "value": "Customer is a non-smoker"
      }
    ]
  },
  "refund": {
    "rfdTotalAmount": 100000
  }
}
```

### Example

```
{
  "sysCode": "000000",
  "sysMsg": "Request Successful"
}
```

### Example

```
{
  "txnRef": "ORD-438UL748T6",
  "txnStatus": "0",
  "txnMessage": "Request was processed successfully.",
  "gateway_txn_id": "HBC0000GHLPAYNUTR023563"
}
```

### Example

```
{
  "amount": 170000,
  "currency": "MYR",
  "payment_datetime": "2019-12-12T14:10:25+08:00",
  "payment_option": "CC",
  "issuing_bank": "ITS BANK",
  "acquirer": "GPay"
}
```

×

INTRODUCTION

Description

Update Log

How to Read this Document

Use Cases for this API

Make Payment

Status Enquiry

Cancel & Refund

Order Confirmation

GETTING STARTED

How to Connect

API Gateway URL

API Authentication

User Identification

Connection Security

Message Security

Sign & Encrypt

Decrypt & Verify

Summary

How to make API request

with Plain Message

with Data Encryption

Data Type Overview

FAQ

SSL Connection

Message Encryption

JOSE Framework

API OPERATIONS

Payments

Payment Page Redirect API

Payment Status Enquiry API

Order Cancellation API

Refund API

Callback Payment Notification API

API SCHEMA

Schema Definitions

commonRespObj

paymentReqModel

pay\_rqt\_txn\_Obj

pay\_rqt\_system\_Obj

pay\_rqt\_payment\_Obj

pay\_rqt\_merchant\_Obj

pay\_rqt\_customer\_Obj

pay\_rqt\_order\_Obj

descriptionsObj

pay\_rqt\_other\_Obj

udfsObj

paymentRespModel

pay\_rpn\_txn\_Obj

pay\_rpn\_system\_Obj

enquiryReqModel

enq\_rqt\_txn\_Obj

enq\_rqt\_merchant\_Obj

enquiryRespModel

enq\_rpn\_sys\_Obj

enq\_rpn\_txn\_Obj

enq\_rpn\_payment\_Obj

enq\_rpn\_cc\_Obj

enq\_rpn\_ipp\_Obj

enq\_rpn\_other\_Obj

enq\_rpn\_refund\_Obj

cancelReqModel

cancel\_rqt\_txn\_Obj

cancel\_rqt\_merchant\_Obj

cancelRespModel

cancel\_rpn\_sys\_Obj

cancel\_rpn\_txn\_Obj

cancel\_rpn\_payment\_Obj

refundReqModel

refund\_rqt\_txn\_Obj

refund\_rqt\_merchant\_Obj

refundRespModel

refund\_rpn\_sys\_Obj

refund\_rpn\_txn\_Obj

refund\_rpn\_payment\_Obj

statusRtnReqModel

notif\_rqt\_txn\_Obj

notif\_rqt\_merchant\_Obj

notif\_rqt\_payment\_Obj

notif\_rqt\_cc\_Obj

notif\_rqt\_ipp\_Obj

notif\_rqt\_other\_Obj

statusRtnRespModel

REFERENCE

Lifecycle of Cryptographic Keys

Key Generation & Exchange

Key Maintenance

Key Renewal

Payment Options

Download Swagger

DISCLAIMER

Disclaimer

**issuing\_bank:** string range: (up to 50 chars) optional

Issuing Bank (available as per request)

**acquirer:** string range: (up to 30 chars) optional

Acquirer (available as per request)

## enq\_rpn\_cc\_Obj: object

### PROPERTIES

**brand:** string range: (up to 20 chars) required

Brand Name

**mcn:** string range: (up to 19 chars) required

Masked Credit Card Number

- First 6 and last 4 digits of credit card number

**bank\_auth\_code:** string range: (up to 30 chars) optional

Authorization Code issued by bank (available as per request)

## enq\_rpn\_ipp\_Obj: object

### PROPERTIES

**ippPeriod:** integer range:  $1 \leq x \leq 99$  required

IPP tenor

**ippFrequency:** string enum: [ WEEKLY, BIMONTHLY, MONTHLY ] range: (up to 20 chars) required

Frequency of the installment plan

## enq\_rpn\_other\_Obj: object

### PROPERTIES

**udfs:** Array< udfsObj > range: (up to 20 objects) optional

Array of User Defined Fields.

## enq\_rpn\_refund\_Obj: object

### PROPERTIES

**rfdTotalAmount:** integer range:  $1 \leq x \leq 999999999999999$  required

Total Successfully Refunded Amount (Applicable to Credit Card Refund only)

**NOTE:** Do not have sign. For example, value | 10000 | means | 100.00 |

## cancelReqModel: object

### PROPERTIES

**transaction:** cancel\_rqt\_txn\_Obj required

**merchant:** cancel\_rqt\_merchant\_Obj required

## cancel\_rqt\_txn\_Obj: object

### PROPERTIES

**txnRef:** string (Critical Field) range: (up to 20 chars) required

Pass Transaction Reference that refers to one specific transaction

## cancel\_rqt\_merchant\_Obj: object

### PROPERTIES

**merId:** string (Critical Field) range: (up to 3 chars) required

Merchant ID

## cancelRespModel: object

### PROPERTIES

**api\_gw:** commonRespObj required

**response:** object required

### PROPERTIES

### Example

```
{  "brand": "VISA",  "mcn": "483587XXXXX48777",  "bank_auth_code": "HBC688"}
```

### Example

```
{  "ippPeriod": 12,  "ippFrequency": "MONTHLY"}
```

### Example

```
{  "udfs": [    {      "definition": "Product Image in Base64 format",      "value": "1VB0Rw0KGg0AAAANSUHEU..."    },    {      "definition": "Special Notes from Customer",      "value": "Customer is a non-smoker"    }  ]}
```

### Example

```
{  "rfdTotalAmount": 100000}
```

### Example

```
{  "transaction": {    "txnRef": "ORD-438UL748T6"  },  "merchant": {    "merId": "HBC"  } }
```

### Example

```
{  "txnRef": "ORD-438UL748T6"}
```

### Example

```
{  "merId": "HBC"}
```

### Example

```
{  "api_gw": {    "messageId": "89817674-da00-4883",    "returnCode": "200",    "returnReason": "Successful operation",  }
```



system: cancel\_rpn\_sys\_Obj required

transaction: cancel\_rpn\_txn\_Obj required

payment: cancel\_rpn\_payment\_Obj required

## cancel\_rpn\_sys\_Obj: object

### PROPERTIES

sysCode: string range: (up to 6 chars) required

Return Code of System Checking where the checking is on API operation level but not transaction level. e.g. Request Successful means the API operation is successful.

Possible Value	Definition
000000	Request Successful
900010	Transaction Record Not Found
999999	Request Fail

sysMsg: string range: (up to 128 chars) required

The corresponding message of System Return Code.

## cancel\_rpn\_txn\_Obj: object

### PROPERTIES

txnRef: string (Critical Field) range: (up to 20 chars) required

Returning Unique Transaction Reference

txnStatus: string range: (up to 4 chars) required

Cancellation (a.k.a Sale Reversal) Status

Possible Value	Definition
0	Cancel success
1	Cancel failed, original transaction could be still under processing or failed due to other reasons like rejected by bank
2	Cancel is pending, merchant system can retry Cancel
-1	Original transaction not found
-2	Internal System Error

txnMessage: string range: (up to 255 chars) required

Additional text message that explains the response

gateway\_txn\_id: string range: (up to 30 chars) required


Returning Unique Transaction ID or Reference Code assigned by Payment Gateway

## cancel\_rpn\_payment\_Obj: object

### PROPERTIES

amount: integer range:  $1 \leq x \leq 99999999999999$  required

Payment Amount in 2 decimal places regardless whether the currency has decimal places or not

 NOTE: Do not use sign. For example, input `10000` instead of `100.00`

currency: string range: (up to 3 chars) required

Return Payment Currency (Format: `ISO 4217 Alpha`)

payment\_option: string range: (up to 20 chars) required

Return Payment Option

Possible Value	Definition
CC	Credit Card
DD	Direct Debit
WA	e-Wallet

issuing\_bank: string range: (up to 50 chars) optional

Issuing Bank (available as per request)

acquirer: string range: (up to 30 chars) optional

Acquirer (available as per request)

## refundReqModel: object

### PROPERTIES

transaction: refund\_rqt\_txn\_Obj required

merchant: refund\_rqt\_merchant\_Obj required

```
{
  "sentTime": "2016-11-15T10:00:00.000Z",
  "responseTime": "2016-11-15T10:00:00.000Z"
},
{
  "response": {
    "system": {
      "sysCode": "000000",
      "sysMsg": "Request Successful"
    },
    "transaction": {
      "txnRef": "ORD-438UL748T6",
      "txnStatus": "0",
      "txnMessage": "RSALE Success",
      "gateway_txn_id": "HBC0000GHLPAYKCU1916784"
    },
    "payment": {
      "amount": 500000,
      "currency": "MYR",
      "payment_option": "CC",
      "issuing_bank": "ITS BANK",
      "acquirer": "GPay"
    }
  }
}
```

### Example

```
{
  "sysCode": "000000",
  "sysMsg": "Request Successful"
}
```

### Example

```
{
  "txnRef": "ORD-438UL748T6",
  "txnStatus": "0",
  "txnMessage": "RSALE Success",
  "gateway_txn_id": "HBC0000GHLPAYKCU1916784"
}
```

### Example

```
{
  "amount": 500000,
  "currency": "MYR",
  "payment_option": "CC",
  "issuing_bank": "ITS BANK",
  "acquirer": "GPay"
}
```

### Example

```
{
  "transaction": {
    "txnRef": "ORD-438UL748T6",
    "rfdRef": "RFD-438UL748T6",
    "rfdAmount": 1000000,
    "currency": "MYR"
  },
  "merchant": {
    "merId": "HBC"
  }
}
```

## INTRODUCTION

### Description

#### Update Log

How to Read this Document

Use Cases for this API

Make Payment

Status Enquiry

Cancel & Refund

Order Confirmation

## GETTING STARTED

### How to Connect

API Gateway URL

API Authentication

User Identification

Connection Security

Message Security

Sign & Encrypt

Decrypt & Verify

Summary

### How to make API request

with Plain Message

with Data Encryption

### Data Type Overview

#### FAQ

SSL Connection

Message Encryption

JOSE Framework

## API OPERATIONS

### Payments

Payment Page Redirect API

Payment Status Enquiry API

Order Cancellation API

Refund API

Callback Payment Notification API

## API SCHEMA

### Schema Definitions

commonRespObj

paymentReqModel

pay\_rqt\_txn\_Obj

pay\_rqt\_system\_Obj

pay\_rqt\_payment\_Obj

pay\_rqt\_merchant\_Obj

pay\_rqt\_customer\_Obj

pay\_rqt\_order\_Obj

descriptionsObj

pay\_rqt\_other\_Obj

udfsObj

paymentRespModel

pay\_rpn\_txn\_Obj

pay\_rpn\_system\_Obj

enquiryReqModel

enq\_rqt\_txn\_Obj

enq\_rqt\_merchant\_Obj

enquiryRespModel

enq\_rpn\_sys\_Obj

enq\_rpn\_txn\_Obj

enq\_rpn\_payment\_Obj

enq\_rpn\_cc\_Obj

enq\_rpn\_hpp\_Obj

enq\_rpn\_other\_Obj

enq\_rpn\_refund\_Obj

cancelReqModel

cancel\_rqt\_txn\_Obj

cancel\_rqt\_merchant\_Obj

cancelRespModel

cancel\_rpn\_sys\_Obj

cancel\_rpn\_txn\_Obj

cancel\_rpn\_payment\_Obj

refundReqModel

refund\_rqt\_txn\_Obj

refund\_rqt\_merchant\_Obj

refundRespModel

refund\_rpn\_sys\_Obj

refund\_rpn\_txn\_Obj

refund\_rpn\_payment\_Obj

statusRtnReqModel

notif\_rqt\_txn\_Obj

notif\_rqt\_merchant\_Obj

notif\_rqt\_payment\_Obj

notif\_rqt\_cc\_Obj

notif\_rqt\_hpp\_Obj

notif\_rqt\_other\_Obj

statusRtnRespModel

## REFERENCE

### Lifecycle of Cryptographic Keys

Key Generation & Exchange

Key Maintenance

Key Renewal

### Payment Options

Download Swagger

## DISCLAIMER

Disclaimer

## refund\_rqt\_txn\_Obj: object

### PROPERTIES

**txnRef**: string (Critical Field) range: (up to 20 chars) required

Pass Transaction Reference that refers to one specific transaction

**rfdRef**: string (Critical Field) range: (up to 20 chars) required

Pass Unique Refund Reference for each Refund Request

- Duplicate Refund Reference will be rejected
- Refund Reference must be different from Transaction Reference

**rfdAmount**: integer range:  $1 \leq x \leq 99999999999999$  required

Merchant provides requested Refund Amount

- Refund Amount must not exceed original Payment Amount

! NOTE: Do not use comma or dot. For example: Input  instead of

**currency**: string enum: [ MYR, SGD, THB, CNY, PHP ] range: (up to 3 chars) required

Refund Currency (Format:

Possible Value	Definition
MYR	Malaysia Ringgit
SGD	Singapore Dollar
THB	Thai Baht
CNY	China Yuan
PHP	Philippine Peso

## refund\_rqt\_merchant\_Obj: object

### PROPERTIES

**merId**: string (Critical Field) range: (up to 3 chars) required

Merchant ID

## refundRespModel: object

### PROPERTIES

**api\_gw**: commonRespObj required

**response**: object required

#### PROPERTIES

**system**: refund\_rpn\_sys\_Obj required

**transaction**: refund\_rpn\_txn\_Obj required

**payment**: refund\_rpn\_payment\_Obj required

## refund\_rpn\_sys\_Obj: object

### PROPERTIES

**sysCode**: string range: (up to 6 chars) required

Return Code of System Checking where the checking is on API operation level but not transaction level. e.g. Request

Successful means the API operation is successful.

Possible Value	Definition
000000	Request Successful
800010	Refund Request Not Allowed: Prior refund not settled
900010	Transaction Record Not Found
900030	Duplicate Refund Transaction Reference
999999	Request Fail

**sysMsg**: string range: (up to 128 chars) required

The corresponding message of System Return Code.

## refund\_rpn\_txn\_Obj: object

### PROPERTIES

**txnRef**: string (Critical Field) range: (up to 20 chars) required

Returning Unique Transaction Reference

**rfdRef**: string (Critical Field) range: (up to 20 chars) required

Returning Refund Reference

**txnStatus**: string range: (up to 4 chars) required

Refund Status

Possible Value	Definition
0	refund success

### Example

```
{
  "txnRef": "ORD-438UL748T6",
  "rfdRef": "RFD-438UL748T6",
  "rfdAmount": 100000,
  "currency": "MYR"
}
```

### Example

```
{
  "merId": "HBC"
}
```

### Example

```
{
  "api_gw": {
    "messageId": "89017674-da00-4683",
    "returnCode": "200",
    "returnReason": "Successful operation",
    "sentTime": "2016-11-15T10:00:00.000Z",
    "responseTime": "2016-11-15T10:00:00.000Z"
  },
  "response": {
    "system": {
      "sysCode": "000000",
      "sysMsg": "Request Successful"
    },
    "transaction": {
      "txnRef": "ORD-438UL748T6",
      "rfdRef": "RFD-438UL748T6",
      "txnStatus": "0",
      "txnMessage": "REFUND Success",
      "gateway_txn_id": "HBC0000GHLPAYOUNI364819"
    },
    "payment": {
      "rfdAmount": 100000,
      "currency": "MYR",
      "payment_option": "CC",
      "issuing_bank": "ITS BANK",
      "acquirer": "GPay"
    }
  }
}
```

### Example

```
{
  "sysCode": "000000",
  "sysMsg": "Request Successful"
}
```

### Example

```
{
  "txnRef": "ORD-438UL748T6",
  "rfdRef": "RFD-438UL748T6",
  "txnStatus": "0",
  "txnMessage": "REFUND Success",
  "gateway_txn_id": "HBC0000GHLPAYOUNI364819"
}
```

## INTRODUCTION

### Description

### Update Log

### How to Read this Document

### Use Cases for this API

### Make Payment

### Status Enquiry

### Cancel & Refund

### Order Confirmation

## GETTING STARTED

### How to Connect

### API Gateway URL

### API Authentication

### User Identification

### Connection Security

### Message Security

### Sign & Encrypt

### Decrypt & Verify

### Summary

### How to make API request

### with Plain Message

### with Data Encryption

### Data Type Overview

### FAQ

### SSL Connection

### Message Encryption

### JOSE Framework

## API OPERATIONS

### Payments

### Payment Page Redirect API

### Payment Status Enquiry API

### Order Cancellation API

### Refund API

### Callback Payment Notification API

## API SCHEMA

### Schema Definitions

### commonRespObj

### paymentReqtModel

### pay\_rqt\_txn\_Obj

### pay\_rqt\_system\_Obj

### pay\_rqt\_payment\_Obj

### pay\_rqt\_merchant\_Obj

### pay\_rqt\_customer\_Obj

### pay\_rqt\_order\_Obj

### descriptionsObj

### pay\_rqt\_other\_Obj

### udfsObj

### paymentRespModel

### pay\_rpn\_txn\_Obj

### pay\_rpn\_system\_Obj

### enquiryReqModel

### enq\_rqt\_txn\_Obj

### enq\_rqt\_merchant\_Obj

### enquiryRespModel

### enq\_rpn\_sys\_Obj

### enq\_rpn\_txn\_Obj

### enq\_rpn\_payment\_Obj

### enq\_rpn\_cc\_Obj

### enq\_rpn\_ipp\_Obj

### enq\_rpn\_other\_Obj

### enq\_rpn\_refund\_Obj

### cancelReqModel

### cancel\_rqt\_txn\_Obj

### cancel\_rqt\_merchant\_Obj

### cancelRespModel

### cancel\_rpn\_sys\_Obj

### cancel\_rpn\_txn\_Obj

### cancel\_rpn\_payment\_Obj

### refundReqModel

### refund\_rqt\_txn\_Obj

### refund\_rqt\_merchant\_Obj

### refundRespModel

### refund\_rpn\_sys\_Obj

### refund\_rpn\_txn\_Obj

### refund\_rpn\_payment\_Obj

### statusRtnReqModel

### notif\_rqt\_txn\_Obj

### notif\_rqt\_merchant\_Obj

### notif\_rqt\_payment\_Obj

### notif\_rqt\_cc\_Obj

### notif\_rqt\_ipp\_Obj

### notif\_rqt\_other\_Obj

### statusRtnRespModel

## REFERENCE

### Lifecycle of Cryptographic Keys

### Key Generation & Exchange

### Key Maintenance

### Key Renewal

### Payment Options

### Download Swagger

## DISCLAIMER

### Disclaimer

Possible Value	Definition
1	refund failed, original transaction could be still under processing or failed due to other reasons like rejected by bank
2	refund is pending, merchant system can retry refund
-1	Original transaction not found
-2	Internal System Error

**txnMessage:** string range: (up to 255 chars) required

Additional text message that explains the response

**gateway\_txn\_id:** string range: (up to 30 chars) required

Returning Unique Transaction ID or Reference Code assigned by Payment Gateway

## refund\_rpn\_payment\_Obj: object

### PROPERTIES

**rfdAmount:** integer range:  $1 \leq x \leq 9999999999999999$  required

Return Refund Amount in 2 decimal places regardless whether the currency has decimal places or not

!

NOTE: Do not have sign. For example, value |10000| means |100.00|

**currency:** string range: (up to 3 chars) required

Return Payment Currency (Format: ISO 4217 Alpha)

**payment\_option:** string range: (up to 20 chars) required

Return Payment Option

Possible Value	Definition
CC	Credit Card
DD	Direct Debit
WA	e-Wallet

**issuing\_bank:** string range: (up to 50 chars) optional

Issuing Bank (available as per request)

**acquirer:** string range: (up to 30 chars) optional

Acquirer (available as per request)

## statusRtnReqModel: object

### PROPERTIES

**transaction:** notif\_rqt\_txn\_Obj required

**merchant:** notif\_rqt\_merchant\_Obj required

**payment:** notif\_rqt\_payment\_Obj required

**cc:** notif\_rqt\_cc\_Obj conditional

Return only if this is a Credit Card payment

**ipp:** notif\_rqt\_ipp\_Obj conditional

Return only if this is an Installment payment

**other:** notif\_rqt\_other\_Obj optional

## notif\_rqt\_txn\_Obj: object

### PROPERTIES

**txnRef:** string (Critical Field) range: (up to 20 chars) required

Returning Unique Transaction Reference

**txnStatus:** string range: (up to 4 chars) required

Transaction Status

Possible Value	Definition
0	transaction successful
1	transaction failed

**txnMessage:** string range: (up to 255 chars) required

Additional text message that explains the response

**gateway\_txn\_id:** string range: (up to 30 chars) required

Returning Unique Transaction ID or Reference Code assigned by Payment Gateway

## notif\_rqt\_merchant\_Obj: object

### PROPERTIES

**merId:** string (Critical Field) range: (up to 3 chars) required

Merchant ID

### Example

```
{  "rfdAmount": 100000,  "currency": "MYR",  "payment_option": "CC",  "issuing_bank": "ITS BANK",  "acquirer": "GPay"}
```

### Example

```
{  "transaction": {    "txnRef": "ORD-438UL748T6",    "txnStatus": "0",    "txnMessage": "Transaction Successful",    "gateway_txn_id": "HBC0000GHLPAYPDUG187660"  },  "merchant": {    "merId": "HBC"  },  "payment": {    "amount": 500000,    "currency": "MYR",    "payment_datetime": "2019-12-12T14:10:25+08:00",    "payment_option": "CC",    "issuing_bank": "ITS BANK",    "acquirer": "GPay",    "token": "SDFG12345FGH23456"  },  "cc": {    "brand": "VISA",    "mcn": "403587XXXXXX4977",    "bank_auth_code": "HBC000"  },  "ipp": {    "ippPeriod": 12,    "ippFrequency": "MONTHLY"  },  "other": {    "udfs": [      {        "definition": "Product Image in Base64 format",        "value": "iVBORw0KGgoAAAANSUHEU..."      },      {        "definition": "Special Notes from Customer",        "value": "Customer is a non-smoker"      }    ]  } }
```

### Example

```
{  "txnRef": "ORD-438UL748T6",  "txnStatus": "0",  "txnMessage": "Transaction Successful",  "gateway_txn_id": "HBC0000GHLPAYPDUG187660"}
```

### Example

```
{  "merId": "HBC"}
```

## INTRODUCTION

### Description

### Update Log

### How to Read this Document

### Use Cases for this API

### Make Payment

### Status Enquiry

### Cancel & Refund

### Order Confirmation

## GETTING STARTED

### How to Connect

### API Gateway URL

### API Authentication

### User Identification

### Connection Security

### Message Security

### Sign & Encrypt

### Decrypt & Verify

### Summary

### How to make API request

### with Plain Message

### with Data Encryption

### Data Type Overview

### FAQ

### SSL Connection

### Message Encryption

### JOSE Framework

## API OPERATIONS

### Payments

### Payment Page Redirect API

### Payment Status Enquiry API

### Order Cancellation API

### Refund API

### Callback Payment Notification API

## API SCHEMA

### Schema Definitions

### commonRespObj

### paymentReqModel

### pay\_rqt\_bxn\_Obj

### pay\_rqt\_system\_Obj

### pay\_rqt\_payment\_Obj

### pay\_rqt\_merchant\_Obj

### pay\_rqt\_customer\_Obj

### pay\_rqt\_order\_Obj

### descriptionsObj

### pay\_rqt\_other\_Obj

### udfsObj

### paymentRespModel

### pay\_rpn\_bxn\_Obj

### pay\_rpn\_system\_Obj

### enquiryReqModel

### enq\_rqt\_bxn\_Obj

### enq\_rqt\_merchant\_Obj

### enquiryRespModel

### enq\_rpn\_sys\_Obj

### enq\_rpn\_bxn\_Obj

### enq\_rpn\_payment\_Obj

### enq\_rpn\_cc\_Obj

### enq\_rpn\_ipp\_Obj

### enq\_rpn\_other\_Obj

### enq\_rpn\_refund\_Obj

### cancelReqModel

### cancel\_rqt\_bxn\_Obj

### cancel\_rqt\_merchant\_Obj

### cancelRespModel

### cancel\_rpn\_sys\_Obj

### cancel\_rpn\_bxn\_Obj

### cancel\_rpn\_payment\_Obj

### refundReqModel

### refund\_rqt\_bxn\_Obj

### refund\_rqt\_merchant\_Obj

### refundRespModel

### refund\_rpn\_sys\_Obj

### refund\_rpn\_bxn\_Obj

### refund\_rpn\_payment\_Obj

### statusRtnReqModel

### notif\_rqt\_bxn\_Obj

### notif\_rqt\_merchant\_Obj

### notif\_rqt\_payment\_Obj

### notif\_rqt\_cc\_Obj

### notif\_rqt\_ipp\_Obj

### notif\_rqt\_other\_Obj

### statusRtnRespModel

## REFERENCE

### Lifecycle of Cryptographic Keys

### Key Generation & Exchange

### Key Maintenance

### Key Renewal

### Payment Options

### Download Swagger

## DISCLAIMER

### Disclaimer

## notif\_rqt\_payment\_Obj: object

### PROPERTIES

**amount:** integer range: 1 ≤ x ≤ 9999999999999999 **required**

Payment Amount in 2 decimal places regardless whether the currency has decimal places or not

**NOTE:** Do not have sign. For example, value `10000` means `100.00`

**currency:** string range: (up to 3 chars) **required**

Return Payment Currency (Format: `ISO 4217 Alpha`)

**payment\_datetime:** string range: (up to 25 chars) **required**

Returning Transaction time for the inward credit payment

- Bank system local time. A `GMT+8` timezone information is appended to the end of the timestamp to indicate this time is a Malaysia local time. Format: `yyyy-MM-dd'T'HH:mm:ss±hh:mm`

**payment\_option:** string range: (up to 20 chars) **required**

Return Payment Option

Possible Value	Definition
CC	Credit Card (Online 3D/Non3D)
DD	Direct Debit
WA	e-Wallet

**issuing\_bank:** string range: (up to 50 chars) **optional**

Issuing Bank (available as per request)

**acquirer:** string range: (up to 30 chars) **optional**

Acquirer (available as per request)

**token:** string range: (up to 50 chars) **optional**

This token will be returned in payment notification if customer selects to securely save their credit card information during the payment process in the online payment page.

The credit card information will then be securely stored in the system and binds with this `token`. Starting from the 2nd Page Redirect submission, pass the same `token` in request will retrieve the corresponding credit card information in the online payment page.

## notif\_rqt\_cc\_Obj: object

### PROPERTIES

**brand:** string range: (up to 20 chars) **required**

Brand Name

**mcn:** string range: (up to 19 chars) **required**

Masked Credit Card Number

- First 6 and last 4 digits of credit card number

**bank\_auth\_code:** string range: (up to 30 chars) **optional**

Authorization Code issued by bank (available as per request)

## notif\_rqt\_ipp\_Obj: object

### PROPERTIES

**ippPeriod:** integer range: 1 ≤ x ≤ 99 **optional**

IPP tenor

**ippFrequency:** string enum: [ WEEKLY, BIMONTHLY, MONTHLY ] range: (up to 20 chars) **required**

Frequency of the installment plan

## notif\_rqt\_other\_Obj: object

### PROPERTIES

**udfs:** Array< udfsObj > **optional**

Array of User Defined Fields

## statusRtnRespModel: object

### PROPERTIES

**status:** string range: (up to 30 chars) **required**

Return Message

## Lifecycle of Cryptographic Keys

This section highlights the Lifecycle of cryptographic keys in the following stages:

- Generate keys pair (Private Key and Public Key Certificate)
- Optional:** Export CSR (Certificate Signing Request) and sign using a CA (Certificate Authority)

### DID YOU KNOW?

In public key infrastructure (PKI) systems, a certificate signing request is a message sent from an applicant to a certificate authority in order to apply for a digital identity certificate. It usually contains the public key for which the certificate should be issued.

- Exchange Certificate with HSBC
- Certificate and Keys Maintenance
- Certificate and Keys Renewal Process

### Example

```
{  "amount": 500000,  "currency": "MYR",  "payment_datetime": "2019-12-12T14:10:25+00:00",  "payment_option": "CC",  "issuing_bank": "ITS BANK",  "acquirer": "GPay",  "token": "SDFG12345FGH23456"}
```

### Example

```
{  "brand": "VISA",  "mcn": "483587XXXXX4977",  "bank_auth_code": "HBC000"}
```

### Example

```
{  "ippPeriod": 4,  "ippFrequency": "MONTHLY"}
```

### Example

```
{  "udfs": [    {      "definition": "Product Image in Base64 format",      "value": "iVBORw0KGgoAAAANSUHE..."    },    {      "definition": "Special Notes from Customer",      "value": "Customer is a non-smoker"    }  ]}
```

### Example

```
{  "status": "SUCCESS"}
```

## INTRODUCTION

Description

Update Log

How to Read this Document

Use Cases for this API

Make Payment

Status Enquiry

Cancel & Refund

Order Confirmation

## GETTING STARTED

How to Connect

API Gateway URL

API Authentication

User Identification

Connection Security

Message Security

Sign & Encrypt

Decrypt & Verify

Summary

How to make API request

with Plain Message

with Data Encryption

Data Type Overview

FAQ

SSL Connection

Message Encryption

JOSE Framework

## API OPERATIONS

Payments

Payment Page Redirect API

Payment Status Enquiry API

Order Cancellation API

Refund API

Callback Payment Notification API

## API SCHEMA

Schema Definitions

commonRespObj

paymentReqModel

pay\_rqt\_bxn\_Obj

pay\_rqt\_system\_Obj

pay\_rqt\_payment\_Obj

pay\_rqt\_merchant\_Obj

pay\_rqt\_customer\_Obj

pay\_rqt\_order\_Obj

descriptionsObj

pay\_rqt\_other\_Obj

udfsObj

paymentRespModel

pay\_rpn\_bxn\_Obj

pay\_rpn\_system\_Obj

enquiryReqModel

enq\_rqt\_bxn\_Obj

enq\_rqt\_merchant\_Obj

enquiryRespModel

enq\_rpn\_sys\_Obj

enq\_rpn\_bxn\_Obj

enq\_rpn\_payment\_Obj

enq\_rpn\_cc\_Obj

enq\_rpn\_hpp\_Obj

enq\_rpn\_other\_Obj

enq\_rpn\_refund\_Obj

cancelReqModel

cancel\_rqt\_bxn\_Obj

cancel\_rqt\_merchant\_Obj

cancelRespModel

cancel\_rpn\_sys\_Obj

cancel\_rpn\_bxn\_Obj

cancel\_rpn\_payment\_Obj

refundReqModel

refund\_rqt\_bxn\_Obj

refund\_rqt\_merchant\_Obj

refundRespModel

refund\_rpn\_sys\_Obj

refund\_rpn\_bxn\_Obj

refund\_rpn\_payment\_Obj

statusRtnReqModel

notif\_rqt\_bxn\_Obj

notif\_rqt\_merchant\_Obj

notif\_rqt\_payment\_Obj

notif\_rqt\_cc\_Obj

notif\_rqt\_hpp\_Obj

notif\_rqt\_other\_Obj

statusRtnRespModel

## REFERENCE

Lifecycle of Cryptographic Keys

Key Generation & Exchange

Key Maintenance

Key Renewal

Payment Options

Download Swagger

## DISCLAIMER

Disclaimer

The Key Renewal Process Command line tool **Java Keytool™** is used in the demonstration. The tool can generate public key / private key pairs and store them into a Java KeyStore. The Keytool executable is distributed with the **Java SDK (or JRE)™**, so if you have an SDK installed you will also have the Keytool executable. The Merchant is free to choose any other tool to generate and manage keys, such as **OpenSSL™**.

## Key Generation and Certificate Exchange with HSBC

1. Create a new keys pair (Private Key and Public Key Certificate) with a new or existing Keystore.

```
keytool -genkey
        -alias merchant_key_pair
        -keyalg RSA
        -keystore merchant_keystore.jks
        -keysize 2048
        -validity 3650
        -storepass <your keystore password>
```

- **-genkey** - command to generate keys pair.
- **-alias** - define the alias name (or unique identifier) of the keys pair stored inside the keystore.
- **-keyalg** - key algorithm, it must be **[RSA]** regarding to HSBC standard. If **[RSA]** is taken, the default hashing algorithm will be **[SHA-256]**.
- **-keystore** - file name of the keystore. If the file already exists in your system location, the key will be created inside your existing keystore, otherwise, a new keystore with the defined name will be created.

**! DID YOU KNOW?**  
Keystore is a password-protected repository of keys and certificates. A file with extension **[jks]** means it is a Java Keystore which is originally supported and executable with Java™.

There are several keystore formats in the industry like **[PKCS12]** with file extension **[p12]** which is executable with Microsoft Windows™, merchant can always pick the one most fit their application.

- **-keysize** - key size, it must be **[2048]** regarding to HSBC standard.
- **-validity** - the validity period of the private key and its associated certificate. The unit is **[day]**, 3650 means 10 years.
- **-storepass** - password of the keystore.

- 1.1. Provide the **[Distinguished Name]** information after running the command:

```
Information required for CSR generation
-----
What is your first and last name?
[Unknown]:  MERCHANT INFO
What is the name of your organizational unit?
[Unknown]:  MERCHANT INFO
What is the name of your organization?
[Unknown]:  MERCHANT INFO
What is the name of your City or Locality?
[Unknown]:  HK
What is the name of your State or Province?
[Unknown]:  HK
What is the two-letter country code for this unit?
[Unknown]:  HK
Is CN=XXX, OU=XXX, O=XXX, L=HK, ST=HK, C=HK correct? (type "yes" or "no")
[no]:  yes

Enter key password for <merchant_key_pair>
(RETURN if same as keystore password):
Re-enter new password:
```

**! NOTE:**  
The Private Key password and Keystore password can be identical, however to be more secure, the Merchant should set them differently.

2. **Optional:** Export CSR and get signed with CA. This step can be skipped if the Merchant decides to work with a Self-Signed Certificate.

```
keytool -certreq
        -alias merchant_key_pair
        -keyalg RSA
        -file merchant_csr.csr
        -keystore merchant_keystore.jks
```

- **-certreq** - command to generate and export CSR.
- **-alias** - the name of the associated keys pair.
- **-keyalg** - key algorithm, it must be **[RSA]** regarding to HSBC standard.
- **-file** - file name of the CSR. This will be generated at the location where the command is run.
- **-keystore** - specify the keystore which you are working on.

- 2.1. Select and purchase a plan at Certificate Authority and then submit the CSR accordingly. After a signed Certificate is issued by CA, import the Certificate back to the Merchant's keystore.

```
keytool -import
        -alias merchant_signed_cert_0001
        -trustcacerts -file CA_signed_cert.p7b
        -keystore merchant_keystore.jks
```

- **-import** - command to import object into a specific keystore.
- **-alias** - define the alias name (or unique identifier) of the signed Certificate.
- **-trustcacerts -file** - specify the file name of the signed Certificate in Merchant's local file system.

**! NOTE:**  
**[PKCS#7]** is one of the common formats that contains certificates and has a file extension of **[.p7b]** or **[.p7c]**. The certificate format may be varied depending on the policy of the issuing CA.

- **-keystore** - specify the keystore which you are working on.

3. Export the Certificate and send it to HSBC for key exchange.

**! DID YOU KNOW:**  
A Certificate or Public Key Certificate is an electronic document that contains a public key and additional information that prove the ownership and maintains integrity of the public key. It is essential for the sender to ensure the key is not altered by any chance during delivery.

```
keytool -export
        -alias merchant_key_pair
        -file merchant_cert_0001.cer
        -keystore merchant_keystore.jks
```

- **-export** - command to export object from a specific keystore.
- **-alias** - the name of the associated keys pair.

**! NOTE:**  
If the Merchant associates the original keys pair **[merchant\_key\_pair]**, the exported Certificate is without CA-signed, and hence, Self-Signed. However, if the Merchant associates the imported Certificate **[merchant\_signed\_cert\_0001]** mentioned in step #2, the exported Certificate is CA-signed.

- **-file** - specify the file name of the Certificate where the file will be exported to Merchant's local file system.

**! NOTE:**  
The default Certificate file encoding is binary. HSBC accepts both binary and base64 encoding. To export a printable base64 encoding file, please attach an extra parameter **[-rfc]** in the command.  
e.g. **-file merchant\_cert\_0001.crt -rfc**.

- **-keystore** - specify the keystore which you are working on.

GETTING STARTED

- How to Connect
- API Gateway URL
- API Authentication
- User Identification
- Connection Security
- Message Security
- Sign & Encrypt
- Decrypt & Verify
- Summary
- How to make API request
- with Plain Message
- with Data Encryption
- Data Type Overview
- FAQ
- SSL Connection
- Message Encryption
- JOSE Framework

API OPERATIONS

- Payments
- Payment Page Redirect API
- Payment Status Enquiry API
- Order Cancellation API
- Refund API
- Callback Payment Notification API

API SCHEMA

- Schema Definitions
- commonRespObj
- paymentReqModel
- pay\_rqt\_bxn\_Obj
- pay\_rqt\_system\_Obj
- pay\_rqt\_payment\_Obj
- pay\_rqt\_merchant\_Obj
- pay\_rqt\_customer\_Obj
- pay\_rqt\_order\_Obj
- descriptionsObj
- pay\_rqt\_other\_Obj
- udfsObj
- paymentRespModel
- pay\_rpn\_bxn\_Obj
- pay\_rpn\_system\_Obj
- enquiryReqModel
- enq\_rqt\_bxn\_Obj
- enq\_rqt\_merchant\_Obj
- enquiryRespModel
- enq\_rpn\_sys\_Obj
- enq\_rpn\_bxn\_Obj
- enq\_rpn\_payment\_Obj
- enq\_rpn\_cc\_Obj
- enq\_rpn\_hpp\_Obj
- enq\_rpn\_other\_Obj
- enq\_rpn\_refund\_Obj
- cancelReqModel
- cancel\_rqt\_bxn\_Obj
- cancel\_rqt\_merchant\_Obj
- cancelRespModel
- cancel\_rpn\_sys\_Obj
- cancel\_rpn\_bxn\_Obj
- cancel\_rpn\_payment\_Obj
- refundReqModel
- refund\_rqt\_bxn\_Obj
- refund\_rqt\_merchant\_Obj
- refundRespModel
- refund\_rpn\_sys\_Obj
- refund\_rpn\_bxn\_Obj
- refund\_rpn\_payment\_Obj
- statusRtnReqModel
- notif\_rqt\_bxn\_Obj
- notif\_rqt\_merchant\_Obj
- notif\_rqt\_payment\_Obj
- notif\_rqt\_cc\_Obj
- notif\_rqt\_hpp\_Obj
- notif\_rqt\_other\_Obj
- statusRtnRespModel

REFERENCE

- Lifecycle of Cryptographic Keys
- Key Generation & Exchange
- Key Maintenance
- Key Renewal
- Payment Options
- Download Swagger

DISCLAIMER

Disclaimer

4. Import HSBC's Certificate into the merchant's Keystore.

```
keytool -import
        -alias hsbc.cert.0002
        -file hsbc.cert.0002.cer
        -keystore merchant.keystore.jks
```

- **-import** - command to import object into a specific keystore.
- **-alias** - define the alias name of HSBC's Certificate in your keystore.
- **-file** - specify the file name of HSBC's Certificate in Merchant's local file system.
- **-keystore** - specify the keystore which you are working on.

5. **Optional:** List keystore objects. Merchant is suggested to verify that all required objects are properly maintained. 2 - 3 entries should be found in your Java Keystore: *(Entries may be varied if other key repository format is used)*

Alias name	Corresponding Object	Remark
merchant_key_pair	<ul style="list-style-type: none"><li>• Merchant's Private Key</li><li>• Merchant's Public Certificate (Self-Signed)</li></ul>	These two objects appear to be one entry in a JAVA Keystore. Merchant can still export them separately into two objects (files) on your local file system depending on your application design.
merchant_signed_cert_0001	<ul style="list-style-type: none"><li>• Merchant's Public Certificate (CA-Signed)</li></ul>	Not exist if Merchant skips step #2
hsbc_cert_0002	<ul style="list-style-type: none"><li>• HSBC's Public Certificate</li></ul>	

```
keytool -list -v -keystore merchant.keystore.jks

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 3 entries

Alias name: merchant_key_pair
Creation date: Jan 1, 2020
Entry type: PrivateKeyEntry

<Other Information>
.....

Alias name: merchant_signed_cert_0001
Creation date: Jan 1, 2020
Entry type: trustedCertEntry

<Other Information>
.....

Alias name: hsbc.cert.0002
Creation date: Jan 1, 2020
Entry type: trustedCertEntry

<Other Information>
.....
```

Certificates and Keys Maintenance

Here are some recommendations to Merchant of how to properly maintain certificates and keys:

Component	Storage	Validity
Merchant's Private Key	<p>Private Key should be maintained and handled with the most secure approach that a Merchant can apply. The most common and yet secure enough approach is:</p> <ul style="list-style-type: none"><li>• <b>key password</b> - Do not save the password in plain text or hard-coded in application. Recommend to encrypt it by any Password Encryption Tools</li><li>• <b>key storage</b> - Store inside password-protected key repository, such as <code>JKS</code> or <code>PKCS12</code> keystore. Keystore password should also be encrypted.</li></ul>	No restriction on the Validity Period. However, if Merchant suspects there is any chance that the key is leaked or for any other security reason, a new Private Key and its associated Public Key Certificate should be generated.
Merchant's Public Key Certificate	<p>Since Public Key Certificate is publicly distributed, a comparative moderate secure storage approach is acceptable. Merchant can store the physical file in any system's file system or store all keys and certificates in one single key repository for a centralised key management.</p>	<p>For a self-signed Certificate, the same condition has been mentioned as above.</p> <p>However, the validity period of a CA-signed Certificate is depended on the purchase plan of the issuing CA. The most common standard is 1 to 2 years.</p>
HSBC's Public Key Certificate	Same as the above	<p>1 Year</p> <p><b>NOTE:</b> Technically, the validity period is usually 1 Year plus 1 to 2 months more. The spare period is a buffer for a merchant to switch a "to-be-expired" Certificate to the new one during the Certificate Renewal Process. More technical detail will be covered in later section.</p>

Certificates and Keys Renewal

Every Public Key Certificate has an expiration date. When either the Merchant's or HSBC's Certificate is about to expire, a key renewal process takes place. Please see the Key Renewal Process Flow below:

- ! SOME RULES YOU SHOULD KNOW:
- **Keys Repository:** This is a mock-up for demonstration purpose only.
  - **Keys Name:** Using a `|Key Name|KeyID` naming convention makes for a simpler demonstration. The suggested identifier of one key should be the alias name inside a key repository.
  - **KeyID Value:** HSBC uses the naming convention `|0001|0002|0003|...|n+1|`, each time the HSBC certificate is renewed, the `|KeyID|` value is `|n+1|`.
  - **KeyID Binding:** The binding between the `|KeyID|` and the corresponding `|Keys Pair|` in the merchant's system can make use of any key/value logic, such as a Database table. In our example below, `KeyID 000X` binds to `Private Key v.000X` and `Public Certificate v.000X`, etc.
  - **Validity Date:** All dates are made-up for demonstration purposes only.

INTRODUCTION

Description

Update Log

How to Read this Document

Use Cases for this API

Make Payment

Status Enquiry

Cancel & Refund

Order Confirmation

GETTING STARTED

How to Connect

API Gateway URL

API Authentication

User Identification

Connection Security

Message Security

Sign & Encrypt

Decrypt & Verify

Summary

How to make API request

with Plain Message

with Data Encryption

Data Type Overview

FAQ

SSL Connection

Message Encryption

JOSE Framework

API OPERATIONS

Payments

Payment Page Redirect API

Payment Status Enquiry API

Order Cancellation API

Refund API

Callback Payment Notification API

API SCHEMA

Schema Definitions

commonRespObj

paymentReqModel

pay\_rqt\_bxn\_Obj

pay\_rqt\_system\_Obj

pay\_rqt\_payment\_Obj

pay\_rqt\_merchant\_Obj

pay\_rqt\_customer\_Obj

pay\_rqt\_order\_Obj

descriptionsObj

pay\_rqt\_other\_Obj

udfsObj

paymentRespModel

pay\_rpn\_bxn\_Obj

pay\_rpn\_system\_Obj

enquiryReqModel

enq\_rqt\_bxn\_Obj

enq\_rqt\_merchant\_Obj

enquiryRespModel

enq\_rpn\_sys\_Obj

enq\_rpn\_bxn\_Obj

enq\_rpn\_payment\_Obj

enq\_rpn\_cc\_Obj

enq\_rpn\_hpp\_Obj

enq\_rpn\_other\_Obj

enq\_rpn\_refund\_Obj

cancelReqModel

cancel\_rqt\_bxn\_Obj

cancel\_rqt\_merchant\_Obj

cancelRespModel

cancel\_rpn\_sys\_Obj

cancel\_rpn\_bxn\_Obj

cancel\_rpn\_payment\_Obj

refundReqModel

refund\_rqt\_bxn\_Obj

refund\_rqt\_merchant\_Obj

refundRespModel

refund\_rpn\_sys\_Obj

refund\_rpn\_bxn\_Obj

refund\_rpn\_payment\_Obj

statusRtnReqModel

notif\_rqt\_bxn\_Obj

notif\_rqt\_merchant\_Obj

notif\_rqt\_payment\_Obj

notif\_rqt\_cc\_Obj

notif\_rqt\_hpp\_Obj

notif\_rqt\_other\_Obj

statusRtnRespModel

REFERENCE

Lifecycle of Cryptographic Keys

Key Generation & Exchange

Key Maintenance

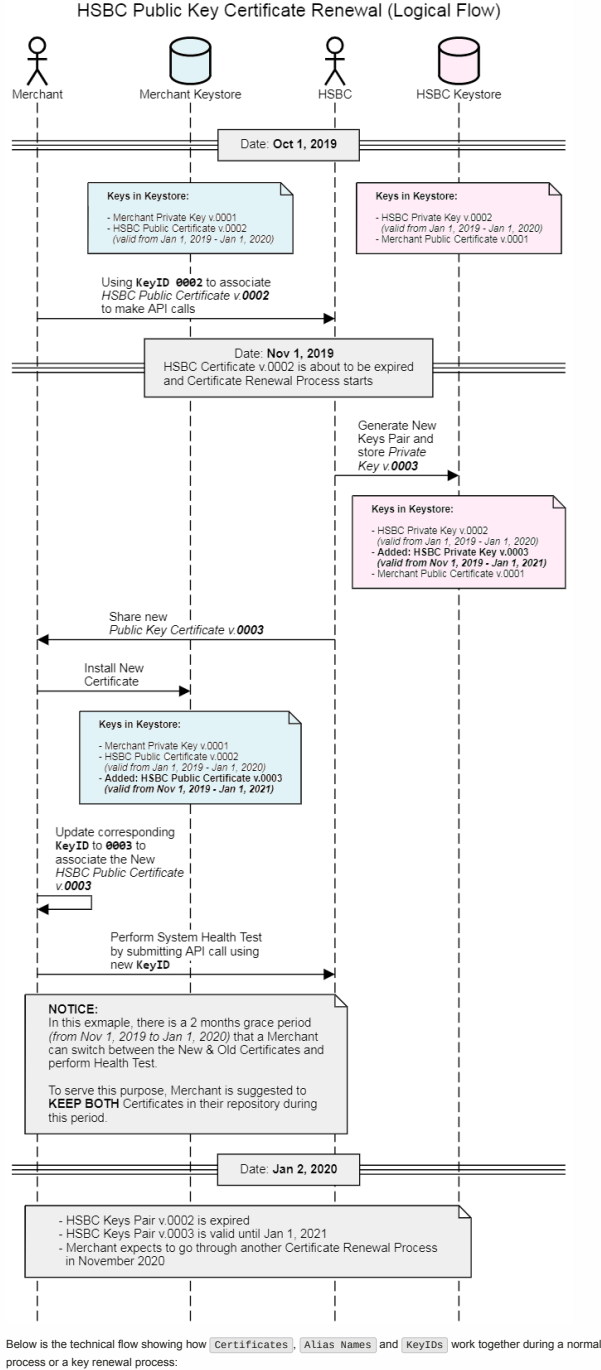
Key Renewal

Payment Options

Download Swagger

DISCLAIMER

Disclaimer



Below is the technical flow showing how Certificates, Alias Names and KeyIDs work together during a normal process or a key renewal process:

INTRODUCTION

- Description
- Update Log
- How to Read this Document
- Use Cases for this API
  - Make Payment
  - Status Enquiry
  - Cancel & Refund
  - Order Confirmation

GETTING STARTED

- How to Connect
  - API Gateway URL
  - API Authentication
  - User Identification
  - Connection Security
  - Message Security
    - Sign & Encrypt
    - Decrypt & Verify
  - Summary
- How to make API request
  - with Plain Message
  - with Data Encryption
- Data Type Overview
- FAQ
  - SSL Connection
  - Message Encryption
  - JOSE Framework

API OPERATIONS

- Payments
  - Payment Page Redirect API
  - Payment Status Enquiry API
  - Order Cancellation API
  - Refund API
  - Callback Payment Notification API

API SCHEMA

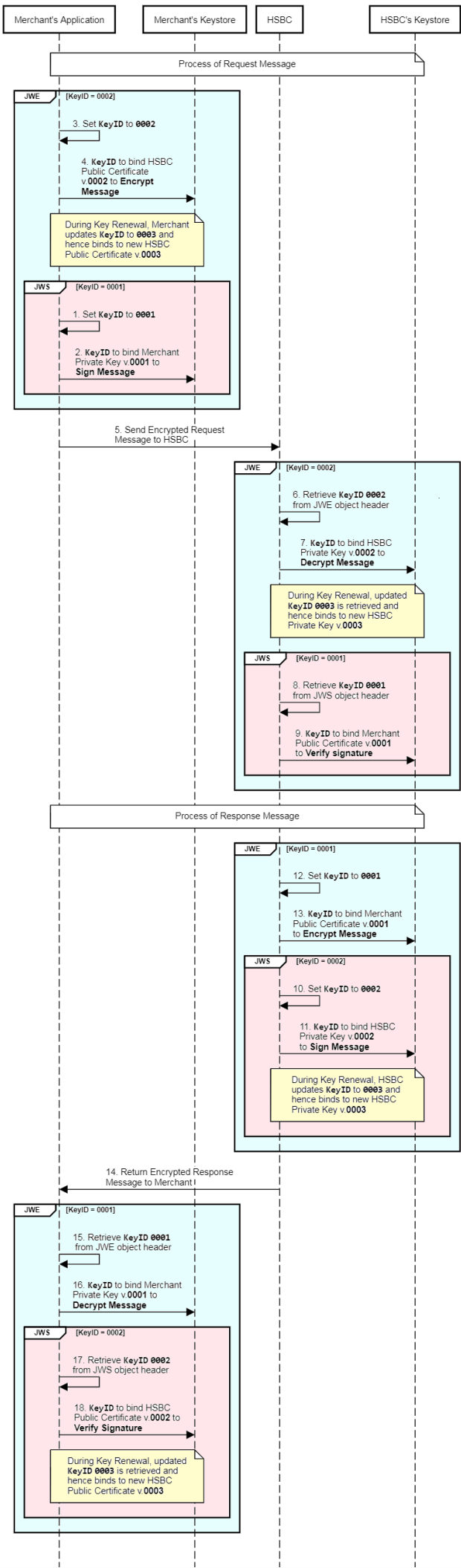
- Schema Definitions
  - commonRespObj
  - paymentReqModel
    - pay\_rqt\_bxn\_Obj
    - pay\_rqt\_system\_Obj
    - pay\_rqt\_payment\_Obj
    - pay\_rqt\_merchant\_Obj
    - pay\_rqt\_customer\_Obj
    - pay\_rqt\_order\_Obj
  - descriptionsObj
  - pay\_rqt\_other\_Obj
  - udfsObj
  - paymentRespModel
    - pay\_rpn\_bxn\_Obj
    - pay\_rpn\_system\_Obj
  - enquiryReqModel
    - enq\_rqt\_bxn\_Obj
    - enq\_rqt\_merchant\_Obj
  - enquiryRespModel
    - enq\_rpn\_sys\_Obj
    - enq\_rpn\_bxn\_Obj
    - enq\_rpn\_payment\_Obj
    - enq\_rpn\_cc\_Obj
    - enq\_rpn\_lpp\_Obj
    - enq\_rpn\_other\_Obj
    - enq\_rpn\_refund\_Obj
  - cancelReqModel
    - cancel\_rqt\_bxn\_Obj
    - cancel\_rqt\_merchant\_Obj
  - cancelRespModel
    - cancel\_rpn\_sys\_Obj
    - cancel\_rpn\_bxn\_Obj
    - cancel\_rpn\_payment\_Obj
  - refundReqModel
    - refund\_rqt\_bxn\_Obj
    - refund\_rqt\_merchant\_Obj
  - refundRespModel
    - refund\_rpn\_sys\_Obj
    - refund\_rpn\_bxn\_Obj
    - refund\_rpn\_payment\_Obj
  - statusRtnReqModel
    - notif\_rqt\_bxn\_Obj
    - notif\_rqt\_merchant\_Obj
    - notif\_rqt\_payment\_Obj
    - notif\_rqt\_cc\_Obj
    - notif\_rqt\_lpp\_Obj
    - notif\_rqt\_other\_Obj
  - statusRtnRespModel

REFERENCE

- Lifecycle of Cryptographic Keys
  - Key Generation & Exchange
  - Key Maintenance
  - Key Renewal
- Payment Options
- Download Swagger

DISCLAIMER

Disclaimer



**NOTE:**  
All examples above concern the HSBC Certificate Renewal. Whenever the Merchant needs to renew their Certificate, they need to switch role and steps to follow those of HSBC's.



INTRODUCTION

- Description
- Update Log
- How to Read this Document
- Use Cases for this API
- Make Payment
- Status Enquiry
- Cancel & Refund
- Order Confirmation

GETTING STARTED

- How to Connect
  - API Gateway URL
  - API Authentication
  - User Identification
  - Connection Security
  - Message Security
    - Sign & Encrypt
    - Decrypt & Verify
  - Summary
- How to make API request
  - with Plain Message
  - with Data Encryption
- Data Type Overview
- FAQ
  - SSL Connection
  - Message Encryption
  - JOSE Framework

API OPERATIONS

- Payments
  - Payment Page Redirect API
  - Payment Status Enquiry API
  - Order Cancellation API
  - Refund API
  - Callback Payment Notification API

API SCHEMA

- Schema Definitions
  - commonRespObj
  - paymentReqModel
    - pay\_rqt\_txn\_Obj
    - pay\_rqt\_system\_Obj
    - pay\_rqt\_payment\_Obj
    - pay\_rqt\_merchant\_Obj
    - pay\_rqt\_customer\_Obj
    - pay\_rqt\_order\_Obj
  - descriptionsObj
  - pay\_rqt\_other\_Obj
  - udfsObj
  - paymentRespModel
    - pay\_rpn\_txn\_Obj
    - pay\_rpn\_system\_Obj
  - enquiryReqModel
    - enq\_rqt\_txn\_Obj
    - enq\_rqt\_merchant\_Obj
  - enquiryRespModel
    - enq\_rpn\_sys\_Obj
    - enq\_rpn\_txn\_Obj
    - enq\_rpn\_payment\_Obj
    - enq\_rpn\_cc\_Obj
    - enq\_rpn\_lpp\_Obj
    - enq\_rpn\_other\_Obj
    - enq\_rpn\_refund\_Obj
  - cancelReqModel
    - cancel\_rqt\_txn\_Obj
    - cancel\_rqt\_merchant\_Obj
  - cancelRespModel
    - cancel\_rpn\_sys\_Obj
    - cancel\_rpn\_txn\_Obj
    - cancel\_rpn\_payment\_Obj
  - refundReqModel
    - refund\_rqt\_txn\_Obj
    - refund\_rqt\_merchant\_Obj
  - refundRespModel
    - refund\_rpn\_sys\_Obj
    - refund\_rpn\_txn\_Obj
    - refund\_rpn\_payment\_Obj
  - statusRtnReqModel
    - notif\_rqt\_txn\_Obj
    - notif\_rqt\_merchant\_Obj
    - notif\_rqt\_payment\_Obj
    - notif\_rqt\_cc\_Obj
    - notif\_rqt\_lpp\_Obj
    - notif\_rqt\_other\_Obj
  - statusRtnRespModel

REFERENCE

- Lifecycle of Cryptographic Keys
  - Key Generation & Exchange
  - Key Maintenance
  - Key Renewal
- Payment Options
- Download Swagger

DISCLAIMER

Disclaimer

## Payment Options

!

**NOTE:**  
By choosing Integrated Options, customer will be landed to an integrated payment selection page hosted by the default payment gateway.  
  
By choosing any other Individual Options, the integrated payment selection page is skipped while customer will be redirected to the payment gateway of the corresponding payment provider.

### Integrated Options

Possible Value	Definition
ANY	All available option(s) registered
CC	Credit Card (Online 3D/Non3D)
DD	All Direct Debit Options
WA	All e-Wallets Options

### Individual e-Wallet Options

Possible Value	Definition
TouchNGo	TouchNGo
GrabPay	GrabPay
Boost	Boost
Mcash	Mcash
QRPay	MayBank QRPay
Paypal2	PayPal

### Individual Direct Debit Options

Possible Value	Definition
ALIPAY	Alipay (Direct Debit)
ALIPAYUSD	Alipay USD (Direct Debit)
PBBUPOP2	PBB UnionPay (Direct Debit)

### Individual FPX B2C Options

Possible Value	Definition
FPXD_ABB0233	Affin Bank Berhad
FPXD_ABMB0212	Alliance Bank Malaysia Berhad
FPXD_AMBB0209	AmBank Malaysia Berhad
FPXD_BIMB0340	Bank Islam Malaysia Berhad
FPXD_BMMB0341	Bank Muamalat Malaysia Berhad
FPXD_BKRM0602	Bank Kerjasama Rakyat Malaysia Berhad
FPXD_BSN0601	Bank Simpanan Nasional
FPXD_BCBB0235	CIMB Bank Berhad
FPXD_CIT0219	CITIBANK BHD
FPXD_HLB0224	Hong Leong Bank Berhad
FPXD_HSBC0223	HSBC Bank Malaysia Berhad
FPXD_KFH0346	Kuwait Finance House (Malaysia) Berhad
FPXD_MBB0228	Malayan Banking Berhad (MZE)
FPXD_MB2U0227	Malayan Banking Berhad (MZU)
FPXD_OCBC0229	OCBC Bank Malaysia Berhad
FPXD_PBB0233	Public Bank Berhad
FPXD_RHB0218	RHB Bank Berhad
FPXD_SCB0216	Standard Chartered Bank
FPXD_UOB0226	United Overseas Bank

### Individual FPX B2B Options

Possible Value	Definition
FPXD62B_ABB0232	Affin Bank Berhad
FPXD62B_ABB0235	Affin Bank Berhad (Max)
FPXD62B_ABMB0213	Alliance Bank Malaysia Berhad
FPXD62B_AMBB0208	AmBank Malaysia Berhad
FPXD62B_BIMB0340	Bank Islam Malaysia Berhad
FPXD62B_BMMB0342	Bank Muamalat Malaysia Berhad
FPXD62B_BKRM0602	Bank Kerjasama Rakyat Malaysia Berhad
FPXD62B_BNP003	BNP Paribas Malaysia Berhad
FPXD62B_BCBB0235	CIMB Bank Berhad
FPXD62B_CIT0218	CITIBANK BHD
FPXD62B_DBB0199	Deutsche Bank Berhad
FPXD62B_HLB0224	Hong Leong Bank Berhad
FPXD62B_HSBC0223	HSBC Bank Malaysia Berhad
FPXD62B_KFH0346	Kuwait Finance House (Malaysia) Berhad
FPXD62B_MBB0228	Malayan Banking Berhad (MZE)
FPXD62B_OCBC0229	OCBC Bank Malaysia Berhad
FPXD62B_PBB0233	Public Bank Berhad
FPXD62B_PBB0234	Public Bank Enterprise
FPXD62B_RHB0218	RHB Bank Berhad
FPXD62B_SCB0215	Standard Chartered Bank
FPXD62B_UOB0228	United Overseas Bank B2B Regional

×

INTRODUCTION

Description

Update Log

How to Read this Document

Use Cases for this API

Make Payment

Status Enquiry

Cancel & Refund

Order Confirmation

GETTING STARTED

How to Connect

API Gateway URL

API Authentication

User Identification

Connection Security

Message Security

Sign & Encrypt

Decrypt & Verify

Summary

How to make API request

with Plain Message

with Data Encryption

Data Type Overview

FAQ

SSL Connection

Message Encryption

JOSE Framework

API OPERATIONS

Payments

Payment Page Redirect API

Payment Status Enquiry API

Order Cancellation API

Refund API

Callback Payment Notification API

API SCHEMA

Schema Definitions

commonRespObj

paymentReqModel

pay\_rqt\_txn\_Obj

pay\_rqt\_system\_Obj

pay\_rqt\_payment\_Obj

pay\_rqt\_merchant\_Obj

pay\_rqt\_customer\_Obj

pay\_rqt\_order\_Obj

descriptionsObj

pay\_rqt\_other\_Obj

udfsObj

paymentRespModel

pay\_rpn\_txn\_Obj

pay\_rpn\_system\_Obj

enquiryReqModel

enq\_rqt\_txn\_Obj

enq\_rqt\_merchant\_Obj

enquiryRespModel

enq\_rpn\_sys\_Obj

enq\_rpn\_txn\_Obj

enq\_rpn\_payment\_Obj

enq\_rpn\_cc\_Obj

enq\_rpn\_hpp\_Obj

enq\_rpn\_other\_Obj

enq\_rpn\_refund\_Obj

cancelReqModel

cancel\_rqt\_txn\_Obj

cancel\_rqt\_merchant\_Obj

cancelRespModel

cancel\_rpn\_sys\_Obj

cancel\_rpn\_txn\_Obj

cancel\_rpn\_payment\_Obj

refundReqModel

refund\_rqt\_txn\_Obj

refund\_rqt\_merchant\_Obj

refundRespModel

refund\_rpn\_sys\_Obj

refund\_rpn\_txn\_Obj

refund\_rpn\_payment\_Obj

statusRtnReqModel

notif\_rqt\_txn\_Obj

notif\_rqt\_merchant\_Obj

notif\_rqt\_payment\_Obj

notif\_rqt\_cc\_Obj

notif\_rqt\_hpp\_Obj

notif\_rqt\_other\_Obj

statusRtnRespModel

REFERENCE

Lifecycle of Cryptographic Keys

Key Generation & Exchange

Key Maintenance

Key Renewal

Payment Options

Download Swagger

DISCLAIMER

Disclaimer

## Download Swagger

Click [here](#) to download Swagger 2.0 file in YAML format.

## Disclaimer

### IMPORTANT NOTICE

This document is issued by The Hongkong and Shanghai Banking Corporation Limited, Hong Kong ("HSBC"). HSBC does not warrant that the contents of this document are accurate, sufficient or relevant for the recipient's purposes and HSBC gives no undertaking and is under no obligation to provide the recipient with access to any additional information or to update all or any part of the contents of this document or to correct any inaccuracies in it which may become apparent. Receipt of this document in whole or in part shall not constitute an offer, invitation or inducement to contract. The recipient is solely responsible for making its own independent appraisal of the products, services and other content referred to in this document. This document should be read in its entirety and should not be photocopied, reproduced, distributed or disclosed in whole or in part to any other person without the prior written consent of the relevant HSBC group member. Copyright: HSBC Group 2019. ALL RIGHTS RESERVED.