

INTRODUCTION

- Description
- Update Log
- How to Read this Document
- Use Cases for this API
- Credit Card
- Instalment & Recurring Payment
- Status Enquiry
- Void & 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
- Payment Status Enquiry API v2
- Void API
- Refund API
- Callback Payment Notification API

Plans

- Create Plan
- Fetch All Plans
- Fetch Plan by ID

Subscriptions

- Fetch Subscription by ID

API SCHEMA

- Schema Definitions
- commonRespObj
- paymentReqModel
- pay_rqt_txn_Obj
- pay_rqt_system_Obj
- pay_rqt_payment_Obj
- pay_rqt_merchant_Obj
- pay_rqt_order_Obj
- descriptionObj
- pay_rqt_other_Obj
- udfObj
- paymentRespModel
- pay_rpn_bn_Obj
- pay_rpn_system_Obj
- enquiryReqModel
- enq_rqt_txn_Obj
- enq_rqt_merchant_Obj
- enquiryRespModel
- v2enquiryReqModel
- enq_rpn_sya_Obj
- enq_rpn_txn_Obj
- v2_enq_rpn_txn_Obj
- enq_rpn_payment_Obj
- v2_enq_rpn_payment_Obj
- enq_rpn_creditcard_Obj
- enq_rpn_refund_Obj
- voidReqModel
- void_rqt_txn_Obj
- void_rqt_merchant_Obj
- voidRespModel
- void_rpn_sya_Obj
- void_rpn_bn_Obj
- refundReqModel
- refund_rqt_txn_Obj
- refund_rqt_merchant_Obj
- refundRespModel
- refund_rpn_sya_Obj
- refund_rpn_txn_Obj
- statusRtnReqModel
- notif_rqt_bnn_Obj
- notif_rqt_merchant_Obj
- notif_rqt_creditcard_Obj
- statusRtnRespModel
- createPlanReqModel
- createPlanRespModel
- systemPostObj
- systemGetObj
- hatLinkObj
- planObj
- subscriptionObj
- getPlanRespModel
- retrieveSubRespModel

REFERENCE

- Lifecycle of Cryptographic Keys
- Key Generation & Exchange
- Key Maintenance
- Key Renewal

iFrame-Enabled Payment Gateway

Download Swagger

DISCLAIMER

Disclaimer

API Specification for Australia and Singapore Cards and Alternate Payment Methods

Version: 3.2

Description

This document introduces the [OpenAPI specification](#) which describes the REST APIs of HSBC's ASP Omni Collection for Retail Payments.

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

Update Log

- [Dec 3, 2021] v3.2 Revised several content sections
- [Dec 3, 2020] v3.1
 - Added possible value to field `payment_option`
 - Removed enumerated type of field `currency`
- [Nov 5, 2020] v3.0
 - Added new country support - Singapore
 - Added Credit Card Tokenization Support
 - Added Instalment Payment Plan Support
- [Jul 8, 2020] v2.0
 - Added new Payment Capability - Recurring Payment
 - Added new APIs [Enquiry API v2](#), new API entities [Plans](#) and [Subscriptions](#)
- [May 19, 2020] v1.4 Changed JSON Message Object name from `credit_card` to `creditcard`
- [May 5, 2020] v1.3 Added possible value `SETTLED` in field `txStatus` of `enq_rpn_txn_Obj` and field `rrdstatus` of `enq_rpn_refund_Obj`
- [Mar 25, 2020] v1.2
 - Added new request field `iframeHostUrl` in [Payment Page Redirect API](#)
 - Added new content [Present Payment Gateway in an iFrame](#)
- [Nov 8, 2019] v1.1 Updated [API Base URL](#) including both Sandbox and Production
- [Sep 20, 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

HSBC's Omni Collection provides a wide range of online payment solutions - enabling online merchants to process credit/debit card payments. The payment platform supports implementations with websites or mobile applications.

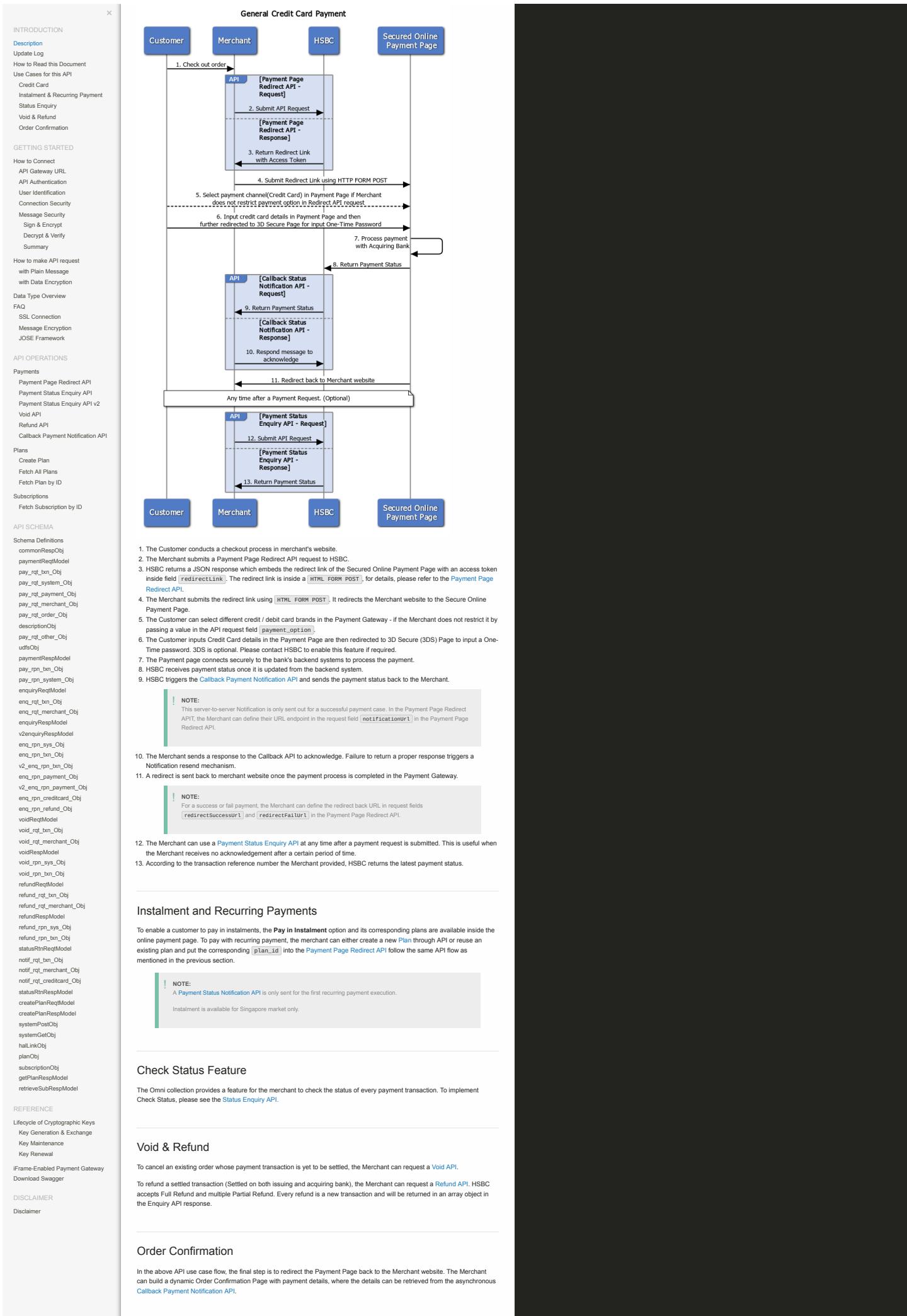
Using our APIs services, you can build your own eCommerce website and accept payments using the following payment channels:

Card Brands / Companies	Australia Market	Singapore Market
Visa	✓	✓
MasterCard	✓	✓
Diners	✓	✗
China UnionPay	✓	✓
American Express	✓	✓
GrabPay	✗	✗

Credit Card / Debit Card Payments

For Credit card transactions in Australia and Singapore, the online Merchant is advised to implement additional security from the issuer Bank, called 3D Secure. This process asks the credit card holder to authenticate by entering an Internet PIN or One Time PIN(OTP).

API Use Case



INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Enquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udfsObj
paymentRespModel
pay_rpn_bn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_sya_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_sya_Obj
void_rpn_txn_Obj
refundReqModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRespModel
refund_rpn_sya_Obj
refund_rpn_txn_Obj
statusRtrReqModel
notif_rqt_bn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtrRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal
iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

How to Connect

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

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

API Gateway URL

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

Production
https://cmb-api.hsbc.com.hk/glcmb-mobilecoll-mcau-ea-merchantservices-prod-proxy/v1

Sandbox
https://devclustercmb.api.p2g.netd2.hsbc.com.hk/glcmb-mobilecoll-mcau-ea-merchantservices-cert-proxy/v1

API Authentication

Username & Password	
Purpose	All APIs are authorized using Basic Authorization
Components	<ul style="list-style-type: none"> Username Password
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"> Client ID Client Secret
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"> Merchant Profile contains all necessary information from a Merchant in order to enable payment service. Merchant ID is used for Merchant identification in each API call.
Components	<ul style="list-style-type: none"> Merchant Profile Merchant ID
Where to get it?	<ul style="list-style-type: none"> Set up by HSBC team after collect information from Merchant Delivered by HSBC via secure email during onboarding procedure
Implementation	nil 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"> Request HSBC API over HTTPS connection (TLS 1.2) Accept Callback API request over HTTPS connection (TLS 1.2)
Components	<ul style="list-style-type: none"> Public SSL Certificate issued by HSBC Merchant's web server or domain whose HTTPS connection is enabled Network Whitelist on HSBC system
Where to get it?	<ul style="list-style-type: none"> Downloaded automatically by Browsers or API Tools, if any problem found, please contact HSBC nil nil
Implementation	nil nil <ul style="list-style-type: none"> Merchant's domain URL will be configured in HSBC's network whitelist by HSBC team

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™).

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Enquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID
Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udsObj
paymentRespModel
pay_rpn_bn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_systObj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_systObj
void_rpn_txn_Obj
refundRtnReqModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRtnReqModel
refund_rpn_systObj
refund_rpn_txn_Obj
statusRtnReqModel
notif_rqt_txn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtnRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

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

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.

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 here		

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

#2 JWSSigner header = new JWSSigner.Builder(JWSAlgorithm.RS256)
.keyID("0001")
.customParam("iat", Instant.now().getEpochSecond()).build();

#3 JWSSignature jwsObject = new JWSSignature(header, payload);

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

return jwsObject;
}
```

1. Prepare your [Message Payload](#), that is, the plain [JSON](#) request message.

2. 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
    "customParam": {
        "iat": "1625587913" // Issued At - the time this request is sent, in Unix Time format
    }
}
```

3. Create a [JWS Object](#) by combining JWS Header and Message Payload.

4. Retrieve your [Private Key](#) as the signer.

5. Create a [Signed JWS Object](#) by signing it with the Private Key.

Next, [Encrypt](#) the Signed JWS Object:

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

#2 JWEHeader jweheader = new JWEHeader.Builder(JWEAlgorithm.RSA_OAEP_256, EncryptionMethod.A128GCM)
.keyID("0002")
.jweObject(jwsObject);
#3 JWEEncrypter encrypter = new RSAEncrypter(key);
#4 jweObject.encrypt(encrypter);
#5 return jweObject;
}
```

1. Prepare your [JWE Payload](#), that is, the [Signed JWS Object](#).

2. 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](#).

3. Create the [JWE Object](#) by combining JWE Header and JWE Payload.

4. Retrieve the [HSBC's Public Key](#) as the encrypter.

5. 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");

#3 JWEDecrypter decrypter = new RSAEncrypter(privateKey);
#4 jweObject.decrypt(decrypter);
```

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Inquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udfsObj
paymentRespModel
pay_rpn_bnn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_systy_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpt_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
statusRtrReqModel
notif_rqt_bnn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtrRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

```

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

```

1. Create an **Encrypted JWE Object** by parsing the encrypted response message payload.
2. Retrieve the **Private Key** as the decrypter.
3. Decrypt the JWE Object using your Private Key.
4. 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();
}

```

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: <code>RS256</code>			
JWE Object		JWE Algorithm: <code>RSA_OAEP_256</code>		
KeyID	<code>6002</code>	<code>6002</code>		
Merchant's Private Key	Used as <code>Signer</code>	Used as <code>Decrypter</code>		
HSBC's Public Key		Used as <code>Encrypter</code>		Used as <code>Verifier</code>

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      GET

#1 curl -X POST "https://devclusteremb.api.p2g.netd2.hsbc.com.hk/gclm-mobilecoll-mcau-ea-merchant
#2 -H "message_encrypt: false"
#3 -H "Authorization: Basic ew91l9ic2ybmftTp5bVyy3Nhcb3jk"
#4 -H "x-HSBC-client-id: 8b015a4f5b5047f091f210e223295ced"
#5 -H "x-HSBC-client-secret: 1bb45ea541dc416d8601685f9583c66"
#6 -H "x-HSBC-msg-encrypt-id: 4229854990001-0001-0002"
#7 -H "Content-Type: application/json"
#8 -d "{ \"txRef\": \"PAY-Q12V956664\", \"merId\": \"4229854990001\"}"

```

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.

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      GET

#1 curl -X POST "https://devclusteremb.api.p2g.netd2.hsbc.com.hk/gclm-mobilecoll-mcau-ea-merchant
#2 -H "Authorization: Basic ew91l9ic2ybmftTp5bVyy3Nhcb3jk"
#3 -H "x-HSBC-client-id: 8b015a4f5b5047f091f210e223295ced"
#4 -H "x-HSBC-client-secret: 1bb45ea541dc416d8601685f9583c66"
#5 -H "x-HSBC-msg-encrypt-id: 4229854990001-0001-0002"
#6 -H "Content-Type: application/json"
#7 -d "eyJraWQiOiIwMDAxIiwZWIjoiQTEyEdOTSisImSzYI61lJTOSIPQUQLT11N1J9.W4nobitovXUMOXGMSI-WO

```

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.

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.

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Enquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udfsObj
paymentRespModel
pay_rpn_bn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_sya_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_sya_Obj
void_rpn_txn_Obj
statusRtnReqModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRespModel
refund_rpn_sya_Obj
refund_rpn_txn_Obj
statusRtnRespModel
notif_rqt_txn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtnRespModel
createPlainReqModel
createPlainRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

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	General field means field which is NOT a critical field. HSBC system will execute characters checking upon all string fields we received in order to tackle security vulnerability, such as Cross-site Scripting. Yet, we recommend you to try use Alphanumeric only for most cases.
String (For critical field)	0-9 A-Z a-z - . [List of Critical Fields: txRef merId product_id rfdRef]	Critical field is used to be either a key or search criteria in HSBC backend system and hence tight restriction is applied to the allowed characters. Moreover, the starting and ending space of the string value will be trimmed before stored in HSBC system. For example, string " example 12 34 " will be trimmed to "example 12 34".
Integer	0-9	Instead of having Max Length check for String, integer range will be checked, e.g. 0 ≤ x ≤ 9999

Field Mandatory Control:

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

Time Zone Control:

Aspect	Format	Definition & Important Notice
In Request Message	yyyy-MM-dd'T'HH:mm:ssZ	Time zone is expected to be <code>GMT+10</code> (Australia local time) or <code>GMT+8</code> (Singapore local time). Merchant is required to perform any necessary time zone conversion before submit request if needed.
In Response Message	yyyy-MM-dd'T'HH:mm:ssZ	Timezone returned in <code>api_l_gw</code> object is generated from HSBC API Gateway which located in Cloud and hence is calculated in <code>GMT+9</code> . 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.

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.net2.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 payments, enquiry, notification, etc.

Payments

POST	/payment/pageRedirect

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Enquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udfsObj
paymentRespModel
pay_rpn_txn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryReqModel
enq_rpn_sya_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rtn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_sya_Obj
void_rpn_txn_Obj
refundRqtReqModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRespModel
refund_rpn_sya_Obj
refund_rpn_txn_Obj
statusRtnReqModel
notif_rqt_txn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtnRespModel
createPlanReqModel
createPlanReqModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

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://test.lpg-online.com/connect/gateway/processing">
<input name="oid" type="hidden" id="oid" value="PAY-QJZV956664" />
<input name="txRef" type="hidden" id="txRef" value="QJZV956664" />
<input name="timezone" type="hidden" id="timezone" value="Asia/Hong_Kong" />
<input name="txndatetime" type="hidden" id="txndatetime" value="2020-01-18:15:21" />
<input name="hash_algorithm" type="hidden" id="hash_algorithm" value="SHA256" />
<input name="hashExtended" type="hidden" id="hashExtended" value="614fbf549912b89a4f88a042e461134e2" />
/* More Input Fields Here... */
</form>
```

REQUEST PARAMETERS

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

REQUEST BODY

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

RESPONSES

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

Request Content-Types: application/json

Request Example

```
{
  "transaction": {
    "txRef": "PAY-QJZV956664",
    "plan_id": "PLN-MJF6114625"
  },
  "system": {
    "redirectSuccessUrl": "https://www.example.com/successPayment",
    "redirectFailUrl": "https://www.example.com/failPayment",
    "notificationUrl": "https://www.example.com/notification",
    "frameHostUrl": "https://www.example.com"
  },
  "payment": {
    "country": "AU",
    "currency": "AUD",
    "payment_option": "all",
    "amount": 1600,
    "token": "TKN-D46HJKB9"
  },
  "merchant": {
    "merId": "42298549900001"
  },
  "order": {
    "description": "Proceed check out order #PAY-QJZV956664",
    "descriptions": [
      {
        "product_name": "Product Item 1",
        "product_id": "PRO-ASDF-1234",
        "unitAmt": 98,
        "unit": 1,
        "vat": 16,
        "subAmt": 100
      },
      {
        "product_name": "Product Item 2",
        "product_id": "PRO-JHGF-9876",
        "unitAmt": 280,
        "unit": 5,
        "vat": 160,
        "subAmt": 1500
      }
    ],
    "other": {
      "udfs": [
        {
          "definition": "Product Image in Base64 format",
          "value": "1VBORw0KGgoAAAANSUEU.."
        },
        {
          "definition": "Special Notes from Customer",
          "value": "Customer is a non-smoker"
        }
      ]
    }
  }
}
```

Response Content-Types: application/json

Response Example (200 OK)

```
{
  "api_id": {
    "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": {
      "txRef": "PAY-QJZV956664"
    },
    "system": {
      "sysCode": "000000",
      "sysMsg": "Request Successful",
      "sysDateTime": "2020-01-01T13:00:00+10:00",
      "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"
}
```

Payments

Payment Status Enquiry API

POST /payment/enquiry

DESCRIPTION
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

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

Request Content-Types: application/json

Request Example

```
{
  "transaction": {
    "txRef": "PAY-QJZV956664"
  },
  "merchant": {
    "merId": "42298549900001"
  }
}
```

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Enquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udfsObj
paymentRespModel
pay_rpn_bn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_sya_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_sya_Obj
void_rpn_bn_Obj
refundReqModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRespModel
refund_rpn_sya_Obj
refund_rpn_txn_Obj
statusRtnReqModel
notif_rqt_txn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtnRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

REQUEST BODY

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

RESPONSES

200 OK *Successful operation.*
enquiryRespModel *Data Encryption is enforced. API Schema intends to demonstrate the skeleton of the message payload only.*

400 Bad Request *Missing or invalid Parameters.*
commonRespObj

403 Forbidden *Authorization credentials are missing or invalid.*

404 Not Found *Empty resource/resource not found.*

500 Internal Server Error *The request failed due to an internal error.*

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": {
      "txRef": "PAY-QIZV956664",
      "txnStatus": "SETTLED",
      "txnApprovalCode": "Y:541641:4525266279:PPXX:25303"
    },
    "payment": {
      "amount": 1600,
      "currency": "AUD",
      "payment_datetime": "2020-01-01T13:02:00+10:00"
    },
    "creditcard": {
      "brand": "VISA",
      "mcn": "463587...4977"
    },
    "refund": [
      {
        "rfdRef": "RFD-KJ05775511",
        "rfdApprovalCode": "Y:898171:4525267235:PPXX:25313",
        "rfdStatus": "DECLINED",
        "rfdAmount": 1600,
        "rfdDatetime": "2020-01-02T13:00:00+10:00"
      }
    ]
  }
}
```

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 v2

GET /payment/transaction/{txRef}

DESCRIPTION

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.

NOTICE:
This version 2 supports both conventional and recurring payments, for any new API consumer, please consider using version 2 instead of 1.

REQUEST PARAMETERS

Authorization *Basic [Base64-encoded Credential]*
required in header

x-hsbc-client-id *[Client ID]*
required in header

x-hsbc-client-secret *[Client Secret]*
required in header

x-hsbc-msg-encrypt-id *[Merchant ID]+[JWS ID]+[JWE ID]*
required in header

Content-Type *application/json*
required in header

txRef *string* *Data Encryption is enforced. API Schema intends to demonstrate the skeleton of the message payload only.*
required in path

RESPONSES

200 OK *Successful operation.*
v2enquiryRespModel *Data Encryption is enforced. API Schema intends to demonstrate the skeleton of the message payload only.*

400 Bad Request *Missing or invalid Parameters.*
commonRespObj

403 Forbidden *Authorization credentials are missing or invalid.*

404 Not Found *Empty resource/resource not found.*

500 Internal Server Error *The request failed due to an internal error.*

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": {
      "txRef": "PAY-QIZV956664",
      "plan_id": "PLN-MFG114625",
      "subscription_id": "SUB-GRVY522694"
    },
    "payments": [
      {
        "id": "4634114049",
        "txnStatus": "SETTLED",
        "txApprovalCode": "Y:541641:4525266279:PPXX:25303",
        "amount": 1600,
        "currency": "AUD",
        "payment_datetime": "2020-01-01T13:02:00+10:00",
        "brand": "VISA",
        "mcn": "463587...4977"
      }
    ],
    "refunds": [
      {
        "rfdRef": "RFD-DFCV112233",
        "rfdApprovalCode": "Y:898171:4525267235:PPXX:25313",
        "rfdStatus": "SETTLED",
        "rfdAmount": 1600,
        "rfdDatetime": "2020-01-02T13:00:00+10:00"
      }
    ]
  }
}
```

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Enquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udfsObj
paymentRespModel
pay_rpn_bn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_sya_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_sya_Obj
void_rpn_txn_Obj
void_rpn_bn_Obj
refundReqModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRespModel
refund_rpn_sya_Obj
refund_rpn_txn_Obj
statusRtnReqModel
notif_rqt_txn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtnRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

```

    "rfdamount": 3000,
    "rfdbatetime": "2020-01-02T13:00:00+10:00"
  },
  {
    "rfdref": "RFD-K3DST75511",
    "rfdapprovalCode": "N-10601:Total amount passed is more than the Return/Void amount.",
    "rfdstatus": "DECLINED",
    "rfdamount": 15000,
    "rfdbatetime": "2020-01-02T13:00:00+10:00"
  }
],
"subscription": {
  "id": "SUB-GKVY522694",
  "status": "installed",
  "period": "month",
  "interval": 1,
  "total_count": 6,
  "paid_count": 6,
  "create_date": "2020-01-01T13:02:00+10:00",
  "start_date": "2020-01-01T00:00:00+10:00",
  "next_charge_date": "2020-02-01T00:00:00+10:00"
},
"udrs": [
  {
    "definition": "Product Image in base64 format",
    "value": "iVBORw0KGgoAAAANSUhEU.."
  },
  {
    "definition": "Special Notes from Customer",
    "value": "Customer is a non-smoker"
  }
],
"links": [
  {
    "href": "/plan/{id}",
    "id": "PLN-KJGF114625",
    "rel": "plan",
    "method": "GET"
  },
  {
    "href": "/subscription/{id}",
    "id": "SUB-GKVY522694",
    "rel": "subscription",
    "method": "GET"
  }
]
}
}

```

Response Example (400 Bad Request)

```

{
  "messageId": "89817674-da00-4883",
  "returnCode": "209",
  "returnReason": "Error Message Here",
  "sentTime": "2016-11-15T10:00:00.000Z",
  "responseTime": "2016-11-15T10:00:00.000Z"
}

```

Void API

Payments

POST /payment/void

DESCRIPTION

This API is used to send a void request for an unsettled transaction.

REQUEST PARAMETERS

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

REQUEST BODY

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

RESPONSES

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

Request Content-Types: application/json

Request Example

```

{
  "transaction": {
    "txRef": "PAY-QJZV958664"
  },
  "merchant": {
    "merId": "42298549900001"
  }
}

```

Response Content-Types: application/json

Response Example (200 OK)

```

{
  "api_ow": {
    "messageId": "89817674-da00-4883",
    "returnCode": "209",
    "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": {
      "txRef": "PAY-QJZV958664",
      "txnStatus": "APPROVED",
      "apprcode": "Y:000000:4525266717:PPXX:25305"
    }
  }
}

```

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"
}

```

Refund API

Payments

POST /payment/refund

DESCRIPTION

This API is used to send a refund request for a previously settled transaction. It supports both full and partial refund.

REQUEST PARAMETERS

Authorization BASIC [Base64-encoded Credential]

Request Content-Types: application/json

Request Example

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Enquiry
Void 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udfsObj
paymentRespModel
pay_rpn_bn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_sya_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_sya_Obj
void_rpn_bn_Obj
refundReqModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRespModel
refund_rpn_sya_Obj
refund_rpn_txn_Obj
statusRtnReqModel
notif_rqt_txn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtnRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

required in header

x-hsbc-client-id [Client ID]
required in header

x-hsbc-client-secret [Client Secret]
required in header

x-hsbc-msg-encrypt-id [Merchant ID]+[JWS ID]+[JWE ID]
optional in header

Content-Type application/json
required in header

REQUEST BODY

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

RESPONSES

200 OK refundRespModel Successful operation.
Data Encryption is enforced. API Schema intends to demonstrate the skeleton of the message payload only.

400 Bad Request commonRespObj Missing or invalid Parameters.

403 Forbidden Authorization credentials are missing or invalid.

404 Not Found Empty resource/resource not found.

500 Internal Server Error The request failed due to an internal error.

```
{
  "transaction": {
    "txRef": "PAY-QJZV956664",
    "fdRef": "RFD-DFCV112233",
    "fdAmount": 1000,
    "currency": "AUD"
  },
  "merchant": {
    "merId": "4229854990001"
  }
}
```

Response Content-Types: application/json

Response Example (200 OK)

```
{
  "api_out": {
    "messageId": "89817674-da00-4883",
    "returnCode": "200",
    "returnReason": "Successful operation",
    "sentTime": "2016-11-15T10:00:00Z",
    "responseTime": "2016-11-15T10:00:00Z"
  },
  "response": {
    "system": {
      "sysCode": "000000",
      "sysMsg": "Request Successful"
    },
    "transaction": {
      "txRef": "PAY-QJZV956664",
      "fdRef": "RFD-DFCV112233",
      "txnStatus": "APPROVED",
      "approvalCode": "Y-098171:4525267235:PPXX:25313"
    }
  }
}
```

Response Example (400 Bad Request)

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

Callback Payment Notification API

POST /<Callback URL predefined by Merchant>

DESCRIPTION

Payment status will be returned to Merchant by asynchronous callback once Mobile Collection receives a payment request. After Mobile Collection payment platform completes reconciliation with bank and receives payment result, Mobile Collection will push the result back to Merchant by calling this API.

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 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 acknowledgement message returned after a certain period of time.

REQUEST PARAMETERS

Content-Type: string text/plain
required in header

REQUEST BODY

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

REQUEST PARAMETERS

Content-Type: string text/plain
required in header

REQUEST BODY

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

RESPONSES

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

Request Content-Types: text/plain

Request Example

```
{
  "transaction": {
    "txRef": "PAY-QJZV956664",
    "approval_code": "Y-541641:4525266279:PPXX:25383",
    "txnStatus": "APPROVED",
    "txndateTime": "2020-01-01T13:02:00+10:00"
  },
  "merchant": {
    "merId": "4229854990001"
  },
  "creditcard": {
    "issuing_bank": "542606",
    "cardholder_country": "DEU",
    "brand": "VISA",
    "resp3DST": "4",
    "token": "TKN-D46HJKB9"
  },
  "subscription": {
    "id": "SUB-GKVS22694",
    "status": "Installed",
    "period": "month",
    "interval": "1",
    "total_count": 6,
    "paid_count": 6,
    "create_date": "2020-01-01T13:02:00+10:00",
    "start_date": "2020-01-01T00:00:00+10:00"
  },
  "other": {
    "udfs": [
      {
        "definition": "Product Image in Base64 format",
        "value": "1VBORw0KGgoAAAANSUhEUQAA"
      },
      {
        "definition": "Special Notes from Customer",
        "value": "Customer is a non-smoker"
      }
    ]
  }
}
```

Response Content-Types: application/json

Response Example (200 OK)

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

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Enquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udsObj
paymentRespModel
pay_rpn_bnn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqn_txn_Obj
enq_rqn_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_systa_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_systa_Obj
void_rpn_txn_Obj
refundReqModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRespModel
refund_rpn_systa_Obj
refund_rpn_txn_Obj
statusRtrnReqModel
notif_rqn_txn_Obj
notif_rqn_merchant_Obj
notif_rqn_creditcard_Obj
statusRtrnRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

Plans

Create a Recurring Payment through a Plan. It acts as a reusable template and contains details of the billing cycle. Depending upon your business, you can create multiple plans with different billing cycles.

You can either reuse an existing plan you previously created or create any new that fits your business needs.

Once a plan is created, submit the [Plan ID](#) in [Payment Page Redirect API](#).

Create Plan

POST /plan

DESCRIPTION
Create a recurring payment plan.

REQUEST PARAMETERS

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

REQUEST BODY

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

RESPONSES

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

Request Content-Types: application/json

Request Example

```
{
  "type": "R",
  "description": "Monthly Recurring Plan #1",
  "period": "month",
  "interval": 1,
  "totalCount": 6
}
```

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": {
    "sysType": "2",
    "sysExt": "00000000",
    "sysMsg": "Request Successful"
  },
  "plan": {
    "id": "PLN-W3FG114625",
    "type": "R",
    "description": "Monthly Recurring Plan #1",
    "period": "month",
    "interval": 1,
    "totalCount": 6,
    "createDate": "2020-01-01T13:02:00+10:00"
  },
  "links": [
    {
      "href": "/plan/{id}",
      "id": "PLN-W3FG114625",
      "rel": "self",
      "method": "GET"
    }
  ]
}
```

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"
}
```

Retrieve All Plans

GET /plan

DESCRIPTION
Use this endpoint to fetch all plans.

REQUEST PARAMETERS

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

RESPONSES

200 OK getPlanRespModel	Successful operation. Data Encryption is enforced. API Schema intends to demonstrate the skeleton of the message payload only.
400 Bad Request commonRespObj	Missing or invalid Parameters.

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": {
    "plans": [
      {
        "id": "PLN-W3FG114625",
        "type": "R",
        "description": "Monthly Recurring Plan #1",
        "period": "month",
        "interval": 1,
        "totalCount": 6,
        "createDate": "2020-01-01T13:02:00+10:00"
      }
    ]
  }
}
```

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Instalment & Recurring Payment
Status Enquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udfsObj
paymentRespModel
pay_rpn_bnn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_systa_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_systa_Obj
void_rpn_txn_Obj
refundRqtReqModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRespModel
refund_rpn_systa_Obj
refund_rpn_txn_Obj
statusRtnReqModel
notif_rqt_bnn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtnRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

403 Forbidden Authorization credentials are missing or invalid.

404 Not Found Empty resource/resource not found.

500 Internal Server Error The request failed due to an internal error.

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"
}
```

Plans

Retrieve Plan by Plan ID

GET /plan/{plan_id}

DESCRIPTION
Use this endpoint to fetch details of a plan by its ID.

REQUEST PARAMETERS

Authorization <small>required in header</small>	BASIC [Base64-encoded Credential]
x-hsbc-client-id <small>required in header</small>	[Client ID]
x-hsbc-client-secret <small>required in header</small>	[Client Secret]
x-hsbc-msg-encrypt-id <small>required in header</small>	[Merchant ID]+[JWS ID]+[JWE ID]
Content-Type <small>required in header</small>	application/json
plan_id: string <small>required in path</small>	Data Encryption is enforced.

RESPONSES

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

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",
      "no_of_record": 99,
      "no_of_page": 1
    },
    "plans": [
      {
        "id": "PLN-MJFG114625",
        "type": "R",
        "description": "Monthly Recurring Plan #1",
        "period": "month",
        "interval": 1,
        "total_count": 6,
        "create_date": "2020-01-01T13:02:00+10:00"
      }
    ]
  }
}
```

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"
}
```

Subscriptions

A subscription ties an order and is created right after you have submitted with a recurring payment. It reflects the status of an active recurring payment such as the current status and next charge date, etc.

Subscriptions

Retrieve Subscription by Subscription ID

GET /subscription/{subscription_id}

DESCRIPTION
Use this endpoint to fetch details of a subscription by its ID.

REQUEST PARAMETERS

Authorization <small>required in header</small>	BASIC [Base64-encoded Credential]
x-hsbc-client-id <small>required in header</small>	[Client ID]
x-hsbc-client-secret <small>required in header</small>	[Client Secret]

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Enquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udsObj
paymentRespModel
pay_rpn_bnn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_syst_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_syst_Obj
void_rpn_txn_Obj
refundReqModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRespModel
refund_rpn_syst_Obj
refund_rpn_txn_Obj
statusRtnReqModel
notif_rqt_bnn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtnRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal
iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

in header

x-hsbc-msg-encrypt-id [Merchant ID]+[JWS ID]+[JWE ID]
Content-Type application/json
subscription_id: string Data Encryption is enforced.

RESPONSES

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

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 Err.
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

Cor.	Return	Return Message Sample	Definition
Return	Code	Sample	Definition
200	Successful operation		A successful API operation in terms of Authorization, Connectivity and valid JSON Message Structure.
400	Client ID - Merchant ID mapping is not correct/updated!		Any checking failure on Business Logic level will be still considered a successful API operation but the Business Logic checking result will be returned in <code>response</code> object.
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		Notices: 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

paymentReqModel: object

PROPERTIES

transaction: pay_rqt_txn_Obj **required**
system: pay_rqt_system_Obj **required**
payment: pay_rqt_payment_Obj **required**

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": {
      "statusCode": "000000",
      "sysMsg": "Request Successful",
      "no_of_record": 99,
      "no_of_page": 1
    }
  },
  "subscriptions": [
    {
      "id": "SUB-GKVK522004",
      "status": "Installed",
      "period": "month",
      "interval": 1,
      "total_count": 6,
      "paid_count": 6,
      "create_date": "2020-01-01T13:02:00+10:00",
      "start_date": "2020-01-01T13:02:00+10:00"
    }
  ]
}
```

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"
}
```

Example

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

Example

```
{
  "transaction": {
    "txRef": "PAY-QJZV958664",
    "plan_id": "PLN-MJFG114625"
  },
  "response": {
    "system": {
      "statusCode": "000000",
      "sysMsg": "Request Successful",
      "no_of_record": 99,
      "no_of_page": 1
    }
  },
  "subscriptions": [
    {
      "id": "SUB-GKVK522004",
      "status": "Installed",
      "period": "month",
      "interval": 1,
      "total_count": 6,
      "paid_count": 6,
      "create_date": "2020-01-01T13:02:00+10:00",
      "start_date": "2020-01-01T13:02:00+10:00"
    }
  ]
}
```

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Enquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonReqObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udfsObj
paymentRespModel
pay_rpn_bnn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_systa_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidReqModel
void_rpn_systa_Obj
void_rpn_txn_Obj
refundRqtModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRespModel
refund_rpn_systa_Obj
refund_rpn_txn_Obj
statusRtnReqModel
notif_rqt_bnn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtnRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

merchant: pay_rqt_merchant_Obj required
order: pay_rqt_order_Obj required
other: pay_rqt_other_Obj optional

```

"system": {
    "redirectSuccessUrl": "https://www.example.com/successPayment",
    "redirectFailUrl": "https://www.example.com/failPayment",
    "notificationUrl": "https://www.example.com/notification",
    "iFrameHostUrl": "https://www.example.com"
},
"payment": {
    "country": "AU",
    "currency": "AUD",
    "payment_option": "all",
    "amount": 1600,
    "token": "TKN-D46HJK89"
},
"merchant": {
    "merId": "42298549900001"
},
"order": {
    "description": "Proceed check out order #PAY-QJZV956664",
    "descriptions": [
        {
            "product_name": "Product Item 1",
            "product_id": "PRO-ASDF-1234",
            "unitAmt": 98,
            "unit": 1,
            "vat": 10,
            "subAmt": 100
        },
        {
            "product_name": "Product Item 2",
            "product_id": "PRO-JHGF-9876",
            "unitAmt": 280,
            "unit": 5,
            "vat": 100,
            "subAmt": 1500
        }
    ],
    "other": {
        "udfs": [
            {
                "definition": "Product Image in Base64 format",
                "value": "iVBORw0KGgoAAAANSUHEU.."
            },
            {
                "definition": "Special Notes from Customer",
                "value": "Customer is a non-smoker"
            }
        ]
    }
}

```

pay_rqt_txn_Obj: object

PROPERTIES

txRef: string range: (up to 100 chars) required
Unique ID referred to a specific transaction.

- Merchant is required to generate a unique ID for each transaction in alphanumeric format, duplicated ID will be rejected.

plan_id: string range: (up to 100 chars) optional
The entity ID of a Plan. Having this ID means the corresponding payment request is a recurring payment.

NOTICE:
Merchant can either create a new Plan or reuse an existing plan.

Example

```
{
  "txRef": "PAY-QJZV956664",
  "plan_id": "PLN-KJF6114625"
}
```

pay_rqt_system_Obj: object

PROPERTIES

redirectSuccessUrl: string range: (up to 256 chars) required
Define URL endpoint for redirecting customer back from Payment Gateway to Merchant website after completing a successful payment.

redirectFailUrl: string range: (up to 256 chars) required
Define URL endpoint for redirecting customer back from Payment Gateway to Merchant website after any fail scenario is taken place.

notificationUrl: string range: (up to 256 chars) required
Define URL endpoint for receiving payment result notification (server-to-server) from HSBC after payment completed.

iFrameHostUrl: string range: (up to 256 chars) optional
Define the Host URL of Merchant's Website in order to switch the Payment Gateway into an iFrame-enabled mode.

NOTICE: The backend behaviour of a typical HTTP-Redirected Payment Gateway is different from the iFrame-enabled mode. Please submit this field only when Merchant decides to present the Payment Gateway inside an iFrame. The Host URL of the Merchant's Website is usually the host of the Parent Frame. Please read [HERE](#) for details.

Example

```
{
  "redirectSuccessUrl": "https://www.example.com/successPayment",
  "redirectFailUrl": "https://www.example.com/failPayment",
  "notificationUrl": "https://www.example.com/notification",
  "iFrameHostUrl": "https://www.example.com"
}
```

pay_rqt_payment_Obj: object

PROPERTIES

country: string enum: [AU, SG] range: (up to 2 chars) required
Country Code (Format: ISO alpha-2)

Possible Value	Definition
AU	Australia
SG	Singapore

currency: string range: (up to 3 chars) required
Payment Currency (Format: ISO 4217 Alpha)

IMPORTANT NOTICE:
Singapore market supports multi-currencies, however, Australia market currently supports **AUD** only.

payment_option: string enum: [all, V, M, A, CUP, G] range: (up to 10 chars) required
To restrict customer payment methods shown in the secured online Payment Gateway

Possible Value	Definition	Australia Market	Singapore Market
all	All available options	✓	✓
V	VISA	✓	✓
M	MasterCard	✓	✓
A	American Express	✓	✓
C	Diners	✓	✗
CUP	UnionPay	✓	✓
G	GrabPay	✗	✗

amount: integer range: 1 ≤ x ≤ 999999999999 required
Payment Amount

IMPORTANT NOTICE:

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Enquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_bnx_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udfsObj
paymentRespModel
pay_rpn_bnx_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_bnx_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryReqModel
enq_rpn_sya_Obj
enq_rpn_tnx_Obj
v2_enq_rpn_tnx_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_bnx_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_sya_Obj
void_rpn_tnx_Obj
refundReqModel
refund_rqt_tnx_Obj
refund_rqt_merchant_Obj
refundRespModel
refund_rpn_sya_Obj
refund_rpn_tnx_Obj
statusRtnReqModel
notif_rqt_bnx_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtnRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

Amount value must include 2 decimal places due to the system default setting for all currencies. Furthermore, do not use any comma or dot. For instance, value **150000** means **1,500.00**.

token: string range: (up to 50 chars) **optional**
Tokenization of Credit Card details. The credit card information will be securely stored in payment gateway and bound with the **[token]** during the first payment submission. Starting from the 2nd submission, passing the same **[token]** will retrieve the corresponding credit card information and Customer only needs to input CVV value.

NOTICE:
[token] can be either assigned by Merchant or System. To be assigned by system, pass value ***** ("asterisk") and retrieve the system-assigned token in [Callback Notification](#)

pay_rqt_merchant_Obj: object

PROPERTIES

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

Merchant ID

- Distributed by HSBC for identifying each merchant's identity

Example

```
{
  "merId": "42298549900001"
}
```

pay_rqt_order_Obj: object

PROPERTIES

description: string range: (up to 100 chars) **optional**

A brief Order Description to be displayed in the settlement report

descriptions: Array<[descriptionObj](#)> range: (up to 20 objects) **required**

Example

```
{
  "description": "Proceed check out order #PAY-QJZV956664",
  "descriptions": [
    {
      "product_name": "Product Item 1",
      "product_id": "PRO-ASDF-1234",
      "unitAmt": 90,
      "unit": 1,
      "vat": 10,
      "subAmt": 100
    },
    {
      "product_name": "Product Item 2",
      "product_id": "PRO-JHGF-9876",
      "unitAmt": 200,
      "unit": 5,
      "vat": 100,
      "subAmt": 1500
    }
  ]
}
```

descriptionObj: object

PROPERTIES

product_name: string range: (up to 200 chars) **required**

Product Item Name / Description

product_id: string range: (up to 50 chars) **required**

Product Number / ID

unitAmt: integer range: 1 ≤ x ≤ 999999999999 **required**

Unit Amount of each item

Example

```
{
  "product_name": "Product Item 1",
  "product_id": "A",
  "unitAmt": 90,
  "unit": 1,
  "vat": 10,
  "subAmt": 100
}
```

pay_rqt_other_Obj: object

PROPERTIES

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

Array of User Defined Fields

Example

```
{
  "udfs": [
    {
      "definition": "Product Image in Base64 format",
      "value": "iVBORw0KGgoAAAANSUhEU.."
    },
    {
      "definition": "Special Notes from Customer",
      "value": "Customer is a non-smoker"
    }
  ]
}
```

udfsObj: object

PROPERTIES

definition: string range: (up to 1024 chars) **optional**

Merchant Defined Definition

value: string range: (up to 2048 chars) **optional**

Merchant Defined Value

Example

```
{
  "definition": "Special Notes from Customer",
  "value": "Customer is a non-smoker"
}
```

paymentRespModel: object

Example

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Inquiry
Void & 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
- Payment Status Enquiry API v2
- Void API
- Refund API
- Callback Payment Notification API

Plans

- Create Plan
- Fetch All Plans
- Fetch Plan by ID

Subscriptions

- Fetch Subscription by ID

API SCHEMA

Schema Definitions

- commonRespObj
- paymentReqModel
- pay_rqt_txn_Obj
- pay_rqt_system_Obj
- pay_rqt_payment_Obj
- pay_rqt_merchant_Obj
- pay_rqt_order_Obj
- descriptionObj
- pay_rqt_other_Obj
- udfsObj
- paymentRespModel
- pay_rpn.bn_Obj
- pay_rpn_system_Obj
- enquiryReqModel
- enq_rqt_txn_Obj
- enq_rqt_merchant_Obj
- enquiryRespModel
- v2enquiryReqModel
- enq_rpn_sya_Obj
- enq_rpn_txn_Obj
- v2_enq_rpn_txn_Obj
- enq_rpn_payment_Obj
- v2_enq_rpn_payment_Obj
- enq_rpn_creditcard_Obj
- enq_rpn_refund_Obj
- voidReqModel
- void_rqt_txn_Obj
- void_rqt_merchant_Obj
- voidRespModel
- void_rpn_sya_Obj
- void_rpn_txn_Obj
- statusRtnReqModel
- notif_rqt_bn_Obj
- notif_rqt_merchant_Obj
- notif_rqt_creditcard_Obj
- statusRtnRespModel
- createPlanReqModel
- createPlanRespModel
- systemPostObj
- systemGetObj
- hatLinkObj
- planObj
- subscriptionObj
- getPlanRespModel
- retrieveSubRespModel

Lifecycle of Cryptographic Keys

- Key Generation & Exchange
- Key Maintenance
- Key Renewal

iFrame-Enabled Payment Gateway

Download Swagger

DISCLAIMER

Disclaimer

REFERENCE

Notice: Optional object(s) will be returning only in a successful request.

PROPERTIES

api_gw: commonRespObj **required**
response: object **required**

PROPERTIES

transaction: pay_rpn_txn_Obj **required**
system: pay_rpn_system_Obj **required**

```
{
  "api_gw": {
    "messageId": "88817674-da00-4883",
    "returnCode": "200",
    "returnReason": "Successful operation",
    "sentTime": "2016-11-15T10:00:00Z",
    "responseTime": "2016-11-15T10:00:00Z"
  },
  "response": {
    "transaction": {
      "txRef": "PAY-QJZV956664"
    },
    "system": {
      "sysCode": "000000",
      "sysMsg": "Request Successful",
      "sysDatetime": "2020-01-01T13:00:00+10:00",
      "redirectLink": "<Encoded_Redirect_Submit_Form>"
    }
  }
}
```

pay_rpn_txn_Obj: object

PROPERTIES

txRef: string range: (up to 100 chars) **required**
Returning back Transaction Reference

Example

```
{
  "txRef": "PAY-QJZV956664"
}
```

pay_rpn_system_Obj: object

PROPERTIES

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

Possible Value	Definition
000000	Request Successful
800110	Invalid Calculation Found in Product Sub-Amount
800120	Invalid Calculation Found in Order Total Amount
900030	Duplicate Transaction Reference

Example

```
{
  "sysCode": "000000",
  "sysMsg": "Request Successful",
  "sysDatetime": "2020-01-01T13:00:00+10:00",
  "redirectLink": "<Encoded_Redirect_Submit_Form>"
}
```

enquiryReqModel: object

PROPERTIES

transaction: enq_rqt_txn_Obj **required**
merchant: enq_rqt_merchant_Obj **required**

Example

```
{
  "transaction": {
    "txRef": "PAY-QJZV956664"
  },
  "merchant": {
    "merId": "42298549900001"
  }
}
```

enq_rqt_txn_Obj: object

PROPERTIES

txRef: string range: (up to 100 chars) **required**
Merchant to pass Transaction Reference that refers to one specific transaction

Example

```
{
  "txRef": "PAY-QJZV956664"
}
```

enq_rqt_merchant_Obj: object

PROPERTIES

merId: string range: (up to 20 chars) **required**
Merchant ID

* Distributed by HSBC to merchant for identifying each merchant's identity

Example

```
{
  "merId": "42298549900001"
}
```

enquiryRespModel: object

PROPERTIES

api_gw: commonRespObj **required**
response: object **required**

PROPERTIES

system: enq_rpn_sya_Obj **required**

Example

```
{
  "api_gw": {
    "messageId": "88817674-da00-4883",
    "returnCode": "200",
    "returnReason": "Successful operation",
    "sentTime": "2016-11-15T10:00:00Z",
    "responseTime": "2016-11-15T10:00:00Z"
  },
  "response": {
    "system": {
      "sysCode": "000000",
      "sysMsg": "Request Successful"
    },
    "transaction": {
      "txRef": "PAY-QJZV956664",
      "txStatus": "SETTLED",
      "txApprovalCode": "Y:541641:4525266279:PPXX:25303"
    },
    "payment": {
      "amount": 1600,
      "currency": "AUD",
      "payment_datetime": "2020-01-01T13:02:00+10:00"
    }
  }
}
```

NOTICE: Optional object(s) will be returning only in a successful request.

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Enquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udfsObj
paymentRespModel
pay_rpn_txn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_systObj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_systObj
void_rpn_txn_Obj
refundReqModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRespModel
refund_rpn_systObj
refund_rpn_txn_Obj
statusRtnReqModel
notif_rqt_txn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtnRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER
Disclaimer

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

v2enquiryRespModel: object

PROPERTIES

```
api_gw: commonRespObj required
response: object required
  PROPERTIES
    system: enq_rpn_systObj required
    transaction: v2_enq_rpn_txn_Obj required
    payments: Array< v2_enq_rpn_payment_Obj > optional
      Return if it is a successful payment
    refunds: Array< enq_rpn_refund_Obj > optional
      Return if any previous refund is requested
    subscription: subscriptionObj optional
      Return if it is a recurring payment
    udfs: Array< udfsObj > range: (up to 20 objects) optional
      Array of User Defined Fields
    links: Array< hatLinkObj > optional
      Collection of related resources
```

Example

```
{
  "api_gw": {
    "messageId": "89817674-da00-4883",
    "returnCode": "200",
    "returnReason": "Successful operation",
    "sentTime": "2016-11-15T10:00:00Z",
    "responseTime": "2016-11-15T10:00:00Z"
  },
  "response": {
    "system": {
      "sysCode": "0000000",
      "sysMsg": "Request Successful"
    },
    "transaction": {
      "txnrId": "QJZV956664",
      "txnid": "PLN-M9F6114625",
      "subscription_id": "SUB-GKVV522694"
    },
    "payments": [
      {
        "id": "645341134483",
        "txnStatus": "SETTLED",
        "txnApprovalCode": "Y:541641:4525266279:PPXX:25303",
        "amount": 1600,
        "currency": "AUD",
        "lastUpdatedTime": "2020-01-01T13:02:00+10:00",
        "brand": "VISA",
        "mon": "463587...4977"
      }
    ],
    "refunds": [
      {
        "rfdRef": "RFD-DFCV112233",
        "rfdApprovalCode": "Y:988171:4525267235:PPXX:25313",
        "rfdStatus": "SETTLED",
        "rfdAmount": 1600,
        "rfdDatetime": "2020-01-02T13:00:00+10:00"
      }
    ],
    "subscription": {
      "id": "SUB-GKVV522694",
      "status": "Installed",
      "period": "month",
      "interval": 1,
      "total_count": 6,
      "paid_count": 6,
      "create_date": "2020-01-01T13:02:00+10:00",
      "start_date": "2020-01-01T00:00:00+10:00",
      "next_charge_date": "2020-02-01T00:00:00+10:00"
    },
    "udfs": [
      {
        "definition": "Product Image in Base64 format",
        "value": "IVBORwKGp0AAAUNREU.."
      },
      {
        "definition": "Special Notes from Customer",
        "value": "Customer is a non-smoker"
      }
    ],
    "links": [
      {
        "href": "/plan/{id}",
        "id": "PLN-M9F6114625",
        "rel": "plan",
        "method": "GET"
      },
      {
        "href": "subscription/{id}",
        "id": "SUB-GKVV522694",
        "rel": "subscription",
        "method": "GET"
      }
    ]
  }
}
```

enq_rpn_systObj: object

PROPERTIES

```
sysCode: string range: (up to 6 chars) required
System Return Code
```

Possible Value	Definition
000000	Request Successful
100010	Transaction is Pending
900010	Transaction Record Not Found
999999	System Error

sysMsg: string range: (up to 128 chars) **required**
System Return Status. This is the corresponding message of System Return Code.

Example

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

enq_rpn_txnObj: object

PROPERTIES

```
txRef: string range: (up to 100 chars) required
Returning Transaction Reference
```

```
txnStatus: string range: (up to 20 chars) required
Transaction Response Status
```

Possible Value	Definition
SETTLED	The transaction is successfully settled
CAPTURED	The transaction is successfully captured

Example

```
{
  "txRef": "PAY-QJZV956664",
  "txnStatus": "SETTLED",
  "txApprovalCode": "Y:541641:4525266279:PPXX:25303"
}
```

INTRODUCTION	
Description	How to Read this Document
Update Log	Use Cases for this API
Credit Card	Installment & Recurring Payment
Status Enquiry	Void & Refund
Void & Refund	Order Confirmation
GETTING STARTED	
How to Connect	API Gateway URL
API Authentication	User Identification
User Identification	Connection Security
Connection Security	Message Security
Message Security	Sign & Encrypt
Sign & Encrypt	Decrypt & Verify
Decrypt & Verify	Summary
How to make API request	with Plain Message
with Plain Message	with Data Encryption
Data Type Overview	FAQ
FAQ	SSL Connection
SSL Connection	Message Encryption
Message Encryption	JOSE Framework
API OPERATIONS	
Payments	Payment Page Redirect API
Payment Status Enquiry API	Payment Status Enquiry API v2
Payment Status Enquiry API v2	Void API
Void API	Refund API
Refund API	Callback Payment Notification API
Plans	Create Plan
Create Plan	Fetch All Plans
Fetch All Plans	Fetch Plan by ID
Fetch Plan by ID	Subscriptions
Subscriptions	Fetch Subscription by ID
API SCHEMA	
Schema Definitions	
commonRespObj	
paymentReqModel	
pay_rqt_txn_Obj	
pay_rqt_system_Obj	
pay_rqt_payment_Obj	
pay_rqt_merchant_Obj	
pay_rqt_order_Obj	
descriptionObj	
pay_rqt_other_Obj	
udsObj	
paymentRespModel	
pay_rpn.bn_Obj	
pay_rpn_system_Obj	
enquiryReqModel	
enq_rqt_txn_Obj	
enq_rqt_merchant_Obj	
enquiryRespModel	
v2enquiryRespModel	
enq_rpn_sya_Obj	
enq_rpn_txn_Obj	
v2_enq_rpn_txn_Obj	
enq_rpn_payment_Obj	
v2_enq_rpn_payment_Obj	
enq_rpn_creditcard_Obj	
enq_rpn_refund_Obj	
voidReqModel	
void_rqt_txn_Obj	
void_rqt_merchant_Obj	
voidRspModel	
void_rpn_sya_Obj	
void_rpn.bn_Obj	
refundReqModel	
refund_rqt_txn_Obj	
refund_rqt_merchant_Obj	
refundRspModel	
refund_rpn_sya_Obj	
refund_rpn_txn_Obj	
statusRtnReqModel	
notif_rqt.bn_Obj	
notif_rqt_merchant_Obj	
notif_rqt_creditcard_Obj	
statusRtnRespModel	
createPlanReqModel	
createPlanRespModel	
systemPostObj	
systemGetObj	
hatLinkObj	
planObj	
subscriptionObj	
getPlanRespModel	
retrieveSubRespModel	
REFERENCE	
Lifecycle of Cryptographic Keys	
Key Generation & Exchange	
Key Maintenance	
Key Renewal	
iFrame-Enabled Payment Gateway	
Download Swagger	
DISCLAIMER	
Disclaimer	

Possible Value	Definition
VOIDED	The transaction is voided
DECLINED	The transaction is cancelled by Customer or declined by Payment Gateway

txApprovalCode: string range: (up to 256 chars) required
 Returning Transaction Approval Code. This code is created by the Payment Gateway. Have this number ready in case you detect any problems with your transaction and you want to contact support.

- Code started with Y means this transaction has no error found.
- Code started with N means this transaction has problem found.

v2_enq_rpn_txn_Obj: object

PROPERTIES

txRef: string range: (up to 100 chars) required
 Returning Transaction Reference

plan_id: string range: (up to 100 chars) optional
 Returning Plan ID

subscription_id: string range: (up to 100 chars) optional
 Returning subscription ID

enq_rpn_payment_Obj: object

PROPERTIES

amount: integer range: 1 ≤ x ≤ 9999999999999 required
 Payment Amount

! NOTE: NO comma or dot. For instance, value 150000 means 1,500.00.

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+10 or GMT+8 timezone information is appended to the end of the timestamp to indicate this time is a Australia or Singapore local time. Format: yyyy-MM-dd'T'HH:mm:ss:hh:mm

v2_enq_rpn_payment_Obj: object

PROPERTIES

id: string range: (up to 100 chars) required
 Payment ID

txStatus: string range: (up to 20 chars) required
 Transaction Response Status

Possible Value	Definition
SETTLED	The transaction is successfully settled
CAPTURED	The transaction is successfully captured
VOIDED	The transaction is voided
DECLINED	The transaction is cancelled by Customer or declined by Payment Gateway

txApprovalCode: string range: (up to 256 chars) required

Returning Transaction Approval Code. This code is created by the Payment Gateway. Have this number ready in case you detect any problems with your transaction and you want to contact support.

- Code started with Y means this transaction has no error found.
- Code started with N means this transaction has problem found.

amount: integer range: 1 ≤ x ≤ 9999999999999 required
 Payment Amount

! IMPORTANT NOTICE:

Amount value must include 2 decimal places due to the system default setting for all currencies. Furthermore, do not use any comma or dot. For instance, value 150000 means 1,500.00.

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+10 or GMT+8 timezone information is appended to the end of the timestamp to indicate this time is a Australia or Singapore local time. Format: yyyy-MM-dd'T'HH:mm:ss:hh:mm

brand: string range: (up to 20 chars) optional
 Brand Name

mcc: string range: (up to 16 chars) optional
 Masked Credit Card Number

- First 6 and last 4 digits of credit card number

enq_rpn_creditcard_Obj: object

PROPERTIES

brand: string range: (up to 20 chars) required
 Brand Name

mcc: string range: (up to 16 chars) required
 Masked Credit Card Number

- First 6 and last 4 digits of credit card number

enq_rpn_refund_Obj: object

PROPERTIES

rfdRef: string range: (up to 100 chars) required
 Unique Refund reference number defined by Merchant

rfdApprovalCode: string range: (up to 256 chars) required

Example
{ "txRef": "PAY-QJV956664", "plan_id": "PLN-KJL614825", "subscription_id": "SUB-GKVY522694" }

Example
{ "amount": 1500, "currency": "AUD", "payment_datetime": "2020-01-01T13:02:00+10:00" }

Example
{ "id": "8E54415469", "txStatus": "SETTLED", "txApprovalCode": "Y:541641:4525266279:PPX:25303", "amount": 1500, "currency": "AUD", "payment_datetime": "2020-01-01T13:02:00+10:00", "brand": "VISA", "mcc": "493587...4977" }

Example
{ "brand": "VISA", "mcc": "493587...4977" }

Example
{ "rfdRef": "RFD-PAY-QJV956664", "rfdApprovalCode": "Y:010075:4524874228:PPX:0000026642", "rfdStatus": "SETTLED", "rfdTimestamp": "2020-01-01T13:02:00+10:00" }

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Enquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udfsObj
paymentRespModel
pay_rpn_bnn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_syst_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRspModel
void_rpn_sys_Obj
void_rpn_txn_Obj
refundRqtModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRspModel
refund_rpn_syst_Obj
refund_rpn_txn_Obj
statusRtnRqtModel
notif_rqt_bnn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtnRspModel
createPlanReqModel
createPlanRspModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRspModel
retrieveSubRspModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

Returning Refund transaction Approval Code. This code is created by the Payment Gateway. Have this number ready in case you detect any problems with your Refund transaction and you want to contact support.

- Code started with **[Y]** means this Refund transaction has no error found.
- Code started with **[N]** means this Refund transaction has problem found.

rfdStatus: string range: (up to 20 chars) **required**
Refund status of the refund transaction

Possible Value	Definition
SETTLED	The refund transaction is successfully settled
CAPTURED	The refund transaction is successfully captured
DECLINED	The refund transaction is declined

rfdAmount: integer range: 1 ≤ \$ ≤ 999999999999 **required**
Returning Refund Amount

- Refund Amount should not exceed the value of total transaction amount
- Support multiple partial refund

NOTE: NO comma or dot. For example: Input **100000** instead of **100.00**

rfdDatetime: string range: (up to 25 chars) **required**
Time of sending out this request

- Server system time. A **GMT+10** or **GMT+8** timezone information is appended to the end of the timestamp to indicate this time is a Australia or Singapore local time. Format: **yyyy-MM-dd'T'HH:mm:sszH:mm**

voidReqModel: object

PROPERTIES

transaction: void_rqt_txn_Obj **required**
merchant: void_rqt_merchant_Obj **required**

Example

```
{
  "transaction": {
    "txRef": "PAY-QJZV956664"
  },
  "merchant": {
    "merId": "42298549900001"
  }
}
```

void_rqt_txn_Obj: object

PROPERTIES

txnRef: string range: (up to 100 chars) **required**
Merchant to pass Transaction Reference that refers to one specific transaction

Example

```
{
  "txRef": "PAY-QJZV956664"
}
```

void_rqt_merchant_Obj: object

PROPERTIES

merId: string range: (up to 20 chars) **required**
Merchant ID

Example

```
{
  "merId": "42298549900001"
}
```

voidRspModel: object

PROPERTIES

api_gw: commonRespObj **required**
response: object **required**

PROPERTIES

system: void_rpn_sys_Obj **required**
transaction: void_rpn_txn_Obj **optional**

NOTICE: Optional object(s) will be returning only in a successful request.

Example

```
{
  "api_gw": {
    "messageId": "88017674-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": {
      "txRef": "PAY-QJZV956664",
      "txnStatus": "APPROVED",
      "approvalCode": "Y:00000014525266717:PPXX:25305"
    }
  }
}
```

void_rpn_sys_Obj: object

PROPERTIES

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

Possible Value	Definition
000000	Request Successful
800010	Void Request Not Allowed: Prior refund found
900010	Transaction Record Not Found
999999	System Error

sysMsg: string range: (up to 128 chars) **required**
System Return Message of the corresponding Code

Example

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

void_rpn_txn_Obj: object

PROPERTIES

Example

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Inquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udfsObj
paymentRespModel
pay_rpn.bn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_sya_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_sya_Obj
void_rpn_txn_Obj
refundRqtModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRspModel
refund_rpn_sya_Obj
refund_rpn_txn_Obj
statusRtnReqModel
notif_rqt_bn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtnRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRspModel
retrieveSubRspModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

Properties

txnRef: string range: (up to 100 chars) required
Return Transaction Reference

txStatus: string range: (up to 20 chars) required
Return Result of the corresponding Void Request

approvalCode: string range: (up to 256 chars) required
Returning Void Request Approval Code. This code is created by the Payment Gateway. Have this number ready in case you detect any problems with your Void Request and you want to contact support.

- Code started with Y means this Void Request has no error found.
- Code started with N means this Void Request has problem found.

refundReqModel: object

Properties

transaction: refund_rqt_txn_Obj required
Merchant to pass the original Transaction Reference

merchant: refund_rqt_merchant_Obj required
Merchant to pass a unique Refund Transaction Reference for each refund request

Properties

txRef: string range: (up to 100 chars) required
Merchant to pass the original Transaction Reference

rfdRef: string range: (up to 100 chars) required
Merchant to pass a unique Refund Transaction Reference for each refund request

rfdAmount: integer range: 1 ≤ \$ 999999999999 required
Merchant provides requested Refund Amount

IMPORTANT NOTICE:
Refund Amount must not exceed original Payment Amount

Amount value must include 2 decimal places due to the system default setting for all currencies. Furthermore, do not use any comma or dot. For instance, value 150000 means 1,500.00.

Properties

currency: string range: (up to 3 chars) required
Payment Currency (Format: ISO 4217 Alpha)

IMPORTANT NOTICE:
Singapore market supports multi-currencies, however, Australia market currently supports AUD only.

Properties

merId: string range: (up to 20 chars) required
Merchant ID

Properties

api_gw: commonRespObj required
response: object required

Properties

system: refund_rpn_sys_Obj required
transaction: refund_rpn_txn_Obj optional

NOTICE: Optional object(s) will be returning only in a successful request.

Properties

sysCode: string range: (up to 6 chars) required
System Return Code

Possible Value	Definition
000000	Request Successful
900010	Transaction Record Not Found
900030	Duplicate Refund Transaction Reference
999999	System Error

sysMsg: string range: (up to 128 chars) required
System Return Status

Properties

txRef: string range: (up to 100 chars) required
Return Transaction Reference

rfdRef: string range: (up to 100 chars) required
Return Transaction Reference

rfdAmount: integer range: 1 ≤ \$ 999999999999 required
Merchant provides requested Refund Amount

```
{
  "txRef": "PAY-QJZV956664",
  "rfdRef": "RFD-DFCV112233",
  "rfdAmount": 1000,
  "currency": "AUD"
}
```

Example

```
{
  "transaction": {
    "txRef": "PAY-QJZV956664",
    "rfdRef": "RFD-DFCV112233",
    "rfdAmount": 1000,
    "currency": "AUD"
  },
  "merchant": {
    "merId": "42298549900001"
  }
}
```

Properties

txRef: string range: (up to 100 chars) required
Return Transaction Reference

rfdRef: string range: (up to 100 chars) required
Return Transaction Reference

rfdAmount: integer range: 1 ≤ \$ 999999999999 required
Merchant provides requested Refund Amount

```
{
  "txRef": "PAY-QJZV956664",
  "rfdRef": "RFD-DFCV112233",
  "rfdAmount": 1000,
  "currency": "AUD"
}
```

Example

```
{
  "txRef": "PAY-QJZV956664",
  "rfdRef": "RFD-DFCV112233",
  "rfdAmount": 1000,
  "currency": "AUD"
}
```

Properties

merId: string range: (up to 20 chars) required
Merchant ID

Properties

api_gw: commonRespObj required
response: object required

Properties

system: refund_rpn_sys_Obj required
transaction: refund_rpn_txn_Obj optional

NOTICE: Optional object(s) will be returning only in a successful request.

Properties

sysCode: string range: (up to 6 chars) required
System Return Code

Possible Value	Definition
000000	Request Successful
900010	Transaction Record Not Found
900030	Duplicate Refund Transaction Reference
999999	System Error

sysMsg: string range: (up to 128 chars) required
System Return Status

Properties

txRef: string range: (up to 100 chars) required
Return Transaction Reference

rfdRef: string range: (up to 100 chars) required
Return Transaction Reference

rfdAmount: integer range: 1 ≤ \$ 999999999999 required
Merchant provides requested Refund Amount

```
{
  "api_gw": {
    "messageId": "89817674-da00-4983",
    "resultCode": "200",
    "returnReason": "Successful operation",
    "sentTime": "2016-11-15T08:00:00Z",
    "responseTime": "2016-11-15T08:00:00Z"
  },
  "response": {
    "system": {
      "statusCode": "000000",
      "sysMsg": "Request Successful"
    },
    "transaction": {
      "txRef": "PAY-QJZV956664",
      "rfdRef": "RFD-DFCV112233",
      "txStatus": "APPROVED",
      "approvalCode": "Y-098171:4525267235:PPXX:25313"
    }
  }
}
```

Example

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

Properties

txRef: string range: (up to 100 chars) required
Return Transaction Reference

rfdRef: string range: (up to 100 chars) required
Return Transaction Reference

rfdAmount: integer range: 1 ≤ \$ 999999999999 required
Merchant provides requested Refund Amount

Properties

txRef: string range: (up to 100 chars) required
Return Transaction Reference

rfdRef: string range: (up to 100 chars) required
Return Transaction Reference

rfdAmount: integer range: 1 ≤ \$ 999999999999 required
Merchant provides requested Refund Amount

Properties

txRef: string range: (up to 100 chars) required
Return Transaction Reference

rfdRef: string range: (up to 100 chars) required
Return Transaction Reference

rfdAmount: integer range: 1 ≤ \$ 999999999999 required
Merchant provides requested Refund Amount

Properties

txRef: string range: (up to 100 chars) required
Return Transaction Reference

rfdRef: string range: (up to 100 chars) required
Return Transaction Reference

rfdAmount: integer range: 1 ≤ \$ 999999999999 required
Merchant provides requested Refund Amount

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Enquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udsObj
paymentRespModel
pay_rpn_bnn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_syst_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_syst_Obj
void_rpn_txn_Obj
refundReqModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRespModel
refund_rpn_syst_Obj
refund_rpn_txn_Obj
statusRtnReqModel
notif_rqt_txn_Obj
notif_rqt_merchant_Obj
statusRtnRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

txRef: string range: (up to 100 chars) required
Return Transaction Reference

rfdRef: string range: (up to 100 chars) required
Return Refund Transaction Reference

txnStatus: string range: (up to 256 chars) required
Return Result of the corresponding Refund Request

approvalCode: string range: (up to 256 chars) required
Returning Refund Request Approval Code. This code is created by the Payment Gateway. Have this number ready in case you detect any problems with your Refund Request and you want to contact support.

- Code started with Y means this Refund Request has no error found.
- Code started with N means this Refund Request has problem found.

```
{
  "txRef": "PAY-01ZV956664",
  "rfdRef": "RFD-DFCV112233",
  "txStatus": "APPROVED",
  "approvalCode": "Y:098171:4525267235:PPXX:25313"
}
```

statusRtnReqModel: object

PROPERTIES

transaction: notif_rqt_txn_Obj required
merchant: notif_rqt_merchant_Obj required
creditcard: notif_rqt_creditcard_Obj optional
subscription: subscriptionObj optional

Return if it is the initial payment of a recurring payment

other: pay_rqt_other_Obj optional

Example

```
{
  "transaction": {
    "txRef": "PAY-QJZV956664",
    "approval_code": "Y:541641:4525266279:PPXX:25303",
    "txStatus": "APPROVED",
    "txDate": "2020-01-01T13:02:00+10:00"
  },
  "merchant": {
    "merId": "42298549900001"
  },
  "creditcard": {
    "issuing_bank": "542606",
    "cardholder_country": "DEU",
    "brand": "VISA",
    "resp3DS": "4",
    "token": "TKN-D46HJK89"
  },
  "subscription": {
    "id": "SUB-GKVS22694",
    "status": "Installed",
    "period": "month",
    "amount": 10,
    "total_count": 6,
    "paid_count": 6,
    "create_date": "2020-01-01T13:02:00+10:00",
    "start_date": "2020-01-01T00:00:00+10:00"
  },
  "other": {
    "uds": [
      {
        "definition": "Product Image in Base64 format",
        "value": "IVBORwKGgAAAQABDQHkUE.."
      },
      {
        "definition": "Special Notes from Customer",
        "value": "Customer is a non-smoker"
      }
    ]
  }
}
```

notif_rqt_txn_Obj: object

PROPERTIES

txRef: string range: (up to 100 chars) required
Returning Transaction Reference

approval_code: string range: (up to 256 chars) required
Returning transaction Approval Code. This code is created by the Payment Gateway. Have this number ready in case you detect any problems with your transaction and you want to contact support.

- Code started with Y means this transaction has no error found.
- Code started with N means this transaction has problem found.

txStatus: string range: (up to 256 chars) required
Returning Transaction status

txnDateTime: string range: (up to 25 chars) required

Time of transaction processing

- Server system time. A GMT+10 or GMT+8 timezone information is appended to the end of the timestamp to indicate this time is a Australia or Singapore local time. Format: yyyy-MM-dd'T'HH:mm:sszhh:mm

Example

```
{
  "txRef": "PAY-QJZV956664",
  "approval_code": "Y:541641:4525266279:PPXX:25303",
  "txStatus": "APPROVED",
  "txDate": "2020-01-01T13:02:00+10:00"
}
```

notif_rqt_merchant_Obj: object

PROPERTIES

merId: string range: (up to 10 chars) required
Returning Merchant ID

Example

```
{
  "merId": "42298549900001"
}
```

notif_rqt_creditcard_Obj: object

PROPERTIES

issuing_bank: string range: (up to 12 chars) required
6 digit identifier of the card issuing bank

cardholder_country: string range: (up to 3 chars) required
3 letter alphanumeric ISO code of the cardholder's country (e.g. USA, DEU, ITA, etc.) Filled with "N/A" if the cardholder's country cannot be determined or the payment type is not credit card

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

Brand of the credit or debit card

Example

```
{
  "issuing_bank": "542606",
  "cardholder_country": "DEU",
  "brand": "VISA",
  "resp3DS": "4",
  "token": "TKN-D46HJK89"
}
```

resp3DS: string range: (up to 2 chars) required

Return code indicating the classification of the transaction:

Possible Value	Definition
MASTERCARD	MasterCard
VISA	VISA
AMEX	American Express
DINERSCLUB	Diners Club
JCB	JCB
CHINA_UNION_PAY	UnionPay

Possible Value	Definition
1	Successful authentication (VISA ECI 05, MasterCard ECI 02)
2	Successful authentication without AVV (VISA ECI 05, MasterCard ECI 02)
3	Authentication failed / incorrect password (transaction declined)

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Inquiry
Void & 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 Inquiry API
Payment Status Inquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udfsObj
paymentRespModel
pay_rpn_bnn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_systa_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_systa_Obj
void_rpn_txn_Obj
refundReqModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRespModel
refund_rpn_systa_Obj
refund_rpn_txn_Obj
statusRtnReqModel
notif_rqt_txn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtnRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
halLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

Possible Value	Definition
4	Authentication attempt (VISA ECI 06, MasterCard ECI 01)
5	Unable to authenticate / Directory Server not responding (VISA ECI 07)
6	Unable to authenticate / Access Control Server not responding (VISA ECI 07)
7	Cardholder not enrolled for 3D Secure (VISA ECI 06)
8	Invalid 3D Secure values received, most likely by the credit card issuing bank's Access Control Server (ACS)

token: string range: (up to 50 chars) optional
Return Token

statusRtnRespModel: object

PROPERTIES

status: string range: (up to 30 chars) required
Return Message to acknowledge

Example

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

createPlanReqModel: object

PROPERTIES

type: string enum: [R] range: (up to 1 chars) optional
Type of Plan

Possible Value	Definition	Remark
R	Recurring	Set to be default value if this field is not provided.

description: string range: (up to 100 chars) required
Description

period: string enum: [day, week, month, year] range: (up to 25 chars) required
Period

interval: integer range: 1 ≤ x ≤ 99 required
Recurring Interval

total_count: integer range: 1 ≤ x ≤ 999 required
Recurring Total Count

Example

```
{
  "type": "P",
  "description": "Monthly Recurring Plan #1",
  "period": "month",
  "interval": 1,
  "totalCount": 6
}
```

createPlanRespModel: object

PROPERTIES

api_gw: commonRespObj required
Response Object

response: object required
Properties

system: systemPostObj required
Plan Obj

links: Array< halLinkObj > optional
Collection of related resources

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",
      "sysMsg": "Request Successful"
    },
    "plan": {
      "id": "P14-WJFG114625",
      "name": "P14-WJFG114625",
      "description": "Monthly Recurring Plan #1",
      "period": "month",
      "interval": 1,
      "totalCount": 6,
      "createDate": "2020-01-01T13:02:00+10:00"
    },
    "links": [
      {
        "href": "/plan/14",
        "rel": "self",
        "method": "GET"
      }
    ]
  }
}
```

systemPostObj: object

PROPERTIES

sysCode: string range: (up to 6 chars) required
System Return Code

Possible Value	Definition
000000	Request Successful
999999	System Error

sysMsg: string range: (up to 128 chars) required
Corresponding Text Message of System Return Code

Example

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

systemGetObj: object

PROPERTIES

sysCode: string range: (up to 6 chars) required
System Return Code

Possible Value	Definition
000000	Request Successful
900010	Record Not Found
999999	System Error

sysMsg: string range: (up to 128 chars) required
Corresponding Text Message of System Return Code

no_of_record: integer range: 1 ≤ x ≤ 999 required
Total No. of Record(s)

Example

```
{
  "sysCode": "000000",
  "sysMsg": "Request Successful",
  "no_of_record": 99,
  "no_of_page": 1
}
```

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Instalment & Recurring Payment
Status Enquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udsObj
paymentRespModel
pay_rpn_bnn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_sya_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_sya_Obj
void_rpn_txn_Obj
refundRqtModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRqtModel
refund_rpn_sya_Obj
refund_rpn_txn_Obj
statusRtnReqModel
notif_rqt_bnn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtnRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
halLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

no_of_page: integer range: 1 ≤ x ≤ 999 **required**
Total No. of Page(s)

halLinkObj: object

PROPERTIES

href: string range: (up to 100 chars) **required**
URL Endpoint of the related resource

id: string range: (up to 100 chars) **optional**
Entity ID of the related resource where it replaces the `@{id}` in the URI.

rel: string range: (up to 100 chars) **required**
Related entity name

method: string range: (up to 100 chars) **required**
HTTP Method of the related resource

INFORMATION:
This object and fields make use of Hypertext Application Language (HAL) standard

Example

```
{
  "href": "/plan/{id}",
  "id": "{Entity_ID}",
  "rel": "{Related_Entity_Name}",
  "method": "GET"
}
```

planObj: object

PROPERTIES

id: string range: (up to 100 chars) **required**
Plan ID

type: string enum: [R] range: (up to 1 chars) **optional**
Type of Plan

Possible Value	Definition	Remark
R	Recurring	Set to be default value if this field is not provided.

description: string range: (up to 100 chars) **required**
Description

period: string enum: [day, week, month, year] range: (up to 25 chars) **required**
Period of occurrence

interval: integer range: 1 ≤ x ≤ 99 **required**
Interval of occurrence within a period

total_count: integer range: 1 ≤ x ≤ 999 **required**
Total Count of occurrence

create_date: string range: (up to 25 chars) **required**
The time this entity is created

- Server system time. A `GMT+10` or `GMT+8` timezone information is appended to the end of the timestamp to indicate this time is a Australia or Singapore local time. Format: `yyyy-MM-dd'T'HH:mm:sszhh:mm`

subscriptionObj: object

PROPERTIES

id: string range: (up to 100 chars) **required**
Subscription ID

status: string range: (up to 100 chars) **required**
Corresponding transaction reference

period: string enum: [day, week, month, year] range: (up to 25 chars) **required**
Period

interval: integer range: 1 ≤ x ≤ 99 **required**
Recurring Interval

total_count: integer range: 1 ≤ x ≤ 999 **required**
Recurring Total Count

paid_count: integer range: 1 ≤ x ≤ 999 **required**
Recurring Paid Count

create_date: string range: (up to 25 chars) **required**
The time this entity is created

- Server system time. A `GMT+10` or `GMT+8` timezone information is appended to the end of the timestamp to indicate this time is a Australia or Singapore local time. Format: `yyyy-MM-dd'T'HH:mm:sszhh:mm`

start_date: string range: (up to 25 chars) **required**
The time the first recurring payment starts to be executed.

- Server system date. No time is specified and always returned `00:00:00`. A `GMT+10` or `GMT+8` timezone information is appended to the end of the timestamp to indicate this time is a Australia local time. Format: `yyyy-MM-dd'T'HH:mm:sszhh:mm`

NOTICE:
It is always the same time when customer first registers on the online payment page.

next_charge_date: string range: (up to 25 chars) **optional**
The time the next recurring payment is executed. This field will not be returned in Callback notification.

- Server system date. No time is specified and always returned `00:00:00`. A `GMT+10` or `GMT+8` timezone information is appended to the end of the timestamp to indicate this time is a Australia or Singapore local time. Format: `yyyy-MM-dd'T'HH:mm:sszhh:mm`

getPlanRespModel: object

PROPERTIES

api_gw: commonRespObj **required**

response: object **required**

PROPERTIES

system: systemGetObj **required**

plans: Array< planObj > **optional**
Return if any plan is successfully created

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

Example

```
{
  "api_gw": {
    "messageId": "89817674-da00-4983",
    "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",
      "no_of_record": 99,
      "no_of_page": 1
    },
    "plans": [
      {
        "id": "PLN-MJFG114625",
        "type": "R",
        "description": "Monthly Recurring Plan #1",
        "period": "month",
        "interval": 1,
        "total_count": 6,
        "create_date": "2020-01-01T13:02:00+10:00"
      }
    ]
}
```

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Enquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udfsObj
paymentRespModel
pay_rpn_bnn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_systObj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_systObj
void_rpn_txn_Obj
refundRqtModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRespModel
refund_rpn_systObj
refund_rpn_txn_Obj
statusRtrReqModel
notif_rqt_txn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtrRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

Lifecycle of Cryptographic Keys

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

1. Generate keys pair (Private Key and Public Key Certificate)
2. **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.

3. Exchange Certificate with HSBC
4. Certificate and Keys Maintenance
5. Certificate and Keys Renewal Process

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, 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.

Example

```
{
  "api_gw": {
    "messageId": "89817674-da00-4883",
    "returnCode": "200",
    "returnReason": "Successful operation",
    "requestTime": "2016-11-15T10:00:00Z",
    "responseTime": "2016-11-15T10:00:00Z"
  },
  "response": {
    "system": {
      "sysId": "8000009",
      "sysMsg": "Request Successful",
      "no_of_record": 99,
      "no_of_page": 1
    },
    "subscriptions": [
      {
        "id": "SUB-GKVY522694",
        "status": "Installed",
        "period": "month",
        "interval": 1,
        "totalCount": 6,
        "paidCount": 6,
        "createDate": "2020-01-01T13:02:00+10:00",
        "startDate": "2020-01-01T00:00:00+10:00"
      }
    ]
  }
}
```

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Inquiry
Void & 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 Inquiry API
Payment Status Inquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonReqObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udfsObj
paymentRespModel
pay_rpn_bn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_sya_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_sya_Obj
void_rpn_txn_Obj
refundRqtReqModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRqtReqModel
refund_rpn_sya_Obj
refund_rpn_txn_Obj
statusRtnReqModel
notif_rqt_txn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtnRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

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.cer -rfc`

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

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"> Merchant's Private Key Merchant's Public Certificate (Self-Signed) 	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"> Merchant's Public Certificate (CA-Signed) 	Not exist if Merchant skips step #2
hsbc_cert_0002	<ul style="list-style-type: none"> HSBC's Public Certificate 	

```
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	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: <ul style="list-style-type: none"> key password - Do not save the password in plain text or hard-coded in application. Recommend to encrypt it by any Password Encryption Tools key storage - Store inside password-protected key repository, such as <code>JKS</code> or <code>PKCS12</code> keystore. Keystore password should also be encrypted. 	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	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.	For a self-signed Certificate, the same condition has been mentioned as above. However, the validity period of a CA-signed Certificate is dependent on the purchase plan of the issuing CA. The most common standard is 1 to 2 years.

INTRODUCTION

Description
Update Log
How to Read this Document
Use Cases for this API
Credit Card
Installment & Recurring Payment
Status Enquiry
Void & 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
Payment Status Enquiry API v2
Void API
Refund API
Callback Payment Notification API

Plans
Create Plan
Fetch All Plans
Fetch Plan by ID

Subscriptions
Fetch Subscription by ID

API SCHEMA

Schema Definitions
commonRespObj
paymentReqModel
pay_rqt_txn_Obj
pay_rqt_system_Obj
pay_rqt_payment_Obj
pay_rqt_merchant_Obj
pay_rqt_order_Obj
descriptionObj
pay_rqt_other_Obj
udfObj
paymentRespModel
pay_rpn_bnn_Obj
pay_rpn_system_Obj
enquiryReqModel
enq_rqt_txn_Obj
enq_rqt_merchant_Obj
enquiryRespModel
v2enquiryRespModel
enq_rpn_syst_Obj
enq_rpn_txn_Obj
v2_enq_rpn_txn_Obj
enq_rpn_payment_Obj
v2_enq_rpn_payment_Obj
enq_rpn_creditcard_Obj
enq_rpn_refund_Obj
voidReqModel
void_rqt_txn_Obj
void_rqt_merchant_Obj
voidRespModel
void_rpn_syst_Obj
void_rpn_bnn_Obj
refundReqModel
refund_rqt_txn_Obj
refund_rqt_merchant_Obj
refundRespModel
refund_rpn_syst_Obj
refund_rpn_txn_Obj
statusRtrReqModel
notif_rqt_bnn_Obj
notif_rqt_merchant_Obj
notif_rqt_creditcard_Obj
statusRtrRespModel
createPlanReqModel
createPlanRespModel
systemPostObj
systemGetObj
hatLinkObj
planObj
subscriptionObj
getPlanRespModel
retrieveSubRespModel

REFERENCE

Lifecycle of Cryptographic Keys
Key Generation & Exchange
Key Maintenance
Key Renewal

iFrame-Enabled Payment Gateway
Download Swagger

DISCLAIMER

Disclaimer

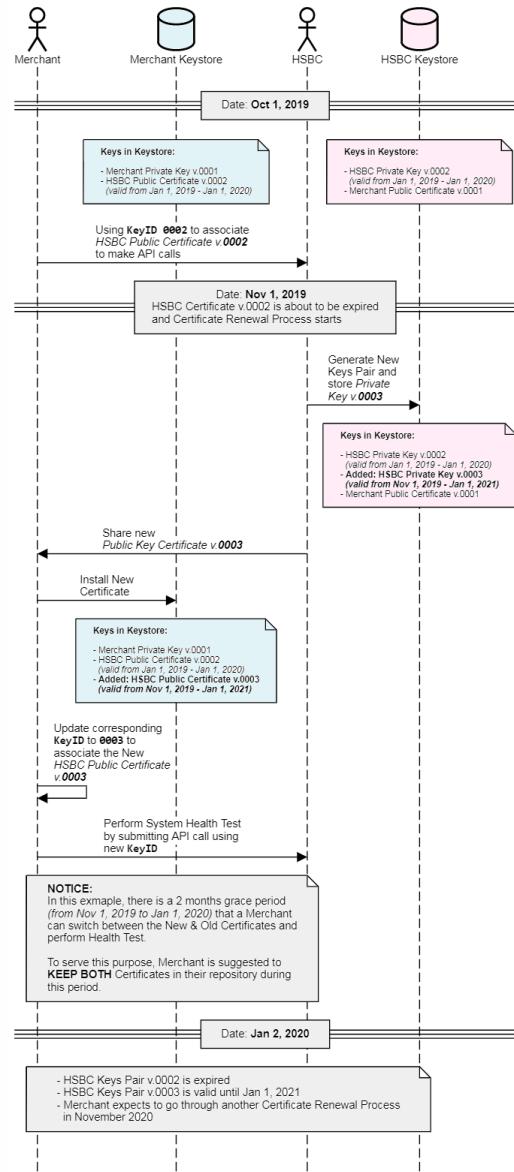
Component	Storage	Validity
HSBC's Public Key Certificate	Same as the above	NOTE: Technically, the validity period is usually 1 Year plus 1 to 2 months more. The spare period is a buffer for a merchant to switch a "to-be-expired" Certificate to the new one during the Certificate Renewal Process. More technical detail will be covered in later section.

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 `v.0001`, `v.0002`, `v.0003`...`v.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 keyvalue 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.

HSBC Public Key Certificate Renewal (Logical Flow)



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

