



Payment API Document

Version 1.66 - 30 May 2023

Table of Contents

Table of Contents	1
1. Introduction	4
1.1 About this Document	4
1.2 Intended Audience	4
1.3 Version Control	4
2. Definitions	5
2.1 Payment Methods	5
2.2 Term Definitions	6
3. System Requirements	7
3.1 Prerequisite	7
3.2 API Rules	8
3.2.1 Protocol Rules	8
3.2.2 Parameter Specifications	9
3.2.2.1 Request Header	9
3.2.2.2 Response Body	9
3.2.2.3 Transaction Amount	9
3.2.3 Security Specification	11
3.2.3.1 Signature	11
3.2.3.2 Signature Verification	12
3.2.4 Access Nodes	13
3.2.5 Backward Compatible Changes	14
4. Payment Method API	15
4.1 Merchant Presented Mode (MPM)	15
4.1.1 User Payment Flow	15
4.1.2 Merchant Process	17
4.1.3 API Body	19
4.1.3.1 Create Dynamic QR API	19
4.1.3.2 Notify Transaction Status	23
4.1.3.3 QR Invalidate API	23
4.1.3.4 Check Transaction Status API	24
4.1.3.5 Create Refund API [New]	24
4.2 Customer Presented Mode (CPM)	25
4.2.1 User Payment Flow	25
4.2.2 Merchant Process	27
4.2.3 API Body	29
4.2.3.1 Create CPM Payment	29

4.2.3.2 Notify Transaction Status	33
4.2.3.3 Check Transaction Status API	33
4.2.3.4 Create Refund API [New]	33
4.3 App-to-app Redirection (Jump App)	34
4.3.1 User Payment Flow	34
4.3.2 Merchant Process	37
4.3.3 API Body	39
4.3.3.1 Create Order	39
4.3.3.2 Notify Transaction Status	43
4.3.3.3 Order Invalidate API	44
4.3.3.4 Check Transaction Status API	45
4.3.3.5 Create Refund API [New]	45
4.4 Account Linking (Tokenized Payment)	46
4.4.1 User Link and Unlink Account Flow	46
4.4.2 User Direct Payment Flow	49
4.4.3 User Auth & Capture Flow	50
4.4.4 Merchant Link and Unlink Account Flow	52
4.4.5 Merchant Direct Payment Flow	54
4.4.6 Merchant Auth Capture Flow	56
4.4.7 API Body	58
4.4.7.1 Account Link	58
4.4.7.2 Get Access Token	61
4.4.7.3 Account Unlink	64
4.4.7.4 Account Link Notify Status	66
4.4.7.5 Get User Info	68
4.4.7.6 Get Redeemable Coin Value	71
4.4.7.7 Direct Payment	73
4.4.7.8 Create Auth	77
4.4.7.9 Reverse Auth	81
4.4.7.10 Capture	84
4.4.7.11 Notify Transaction Status	87
4.4.7.12 Create Refund API [New]	87
4.4.7.13 Check Transaction Status API	87
4.5. Ticket Linking	88
4.5.1 User Ticket Linking and Payment Flow	88
4.5.2 Merchant Process	91
4.5.3 API Body	93
4.5.3.1 Ticket Charge API	93
Sample Request	96

Sample Response	96
4.5.3.2 Check Transaction Status API	96
4.5.3.3 Create Refund API [New]	96
5. Common APIs	97
5.1 Notify Transaction Status	97
5.2 Check Transaction Status API	101
5.3 Create Refund API [New]	105
5.4 Get Transactions API	109
5.5 Set Merchant API	112
5.6 Set Store API	116
Appendix	122
Appendix 1A - Transaction Types	122
Appendix 1B - Payment Status	123
Appendix 1C - Transaction Status	124
Appendix 1D - Promotion Types	125
Appendix 1E - Payment Channels	126
Appendix 2 - Point of Initiation	127
Appendix 3 - Merchant Category Codes (MCC)	128
Appendix 4 - National Identity Types	129
Appendix 5 - Error Codes	131
Version Control	134

1. Introduction

1.1 About this Document

This document is the official manual for integration with ShopeePay Payment Application Programming Interfaces (APIs).

It describes the APIs between ShopeePay and any client system which intends to offer ShopeePay as a payment option in their system, such as merchant aggregator systems, point-of-sale (POS) systems, Independent Software Vendor (ISV) system and online commerce platforms.

This document is designed to be read in order.

1.2 Intended Audience

The intended audience for this document are technical personnels and programmers with experience in payment integrations.

1.3 Version Control

Please refer to the end of this document for detailed changes between versions.

2. Definitions

2.1 Payment Methods

1) Merchant Presented Mode (MPM)

The client generates transaction QR according to ShopeePay payment protocol, containing transaction amount. User will scan the QR code to make payment. Clients will be notified via callback about the payment status of the transaction.

2) Customer Presented Mode (CPM)

The user will show his user QR code to the cashier at the merchant store. Cashier will enter the payment amount and scan the user's QR code in order to make payment directly.

3) App-to-app Redirect (Jump App)

The user will select ShopeePay as a payment method from an external app. External app will redirect the user to the ShopeePay/Shopee app to complete payment. Upon payment success, user can jump back to the external app to view their order status.

4) Account Linking Mode (Tokenized Payment)

The user will perform a one-time authentication and authorization to successfully link ShopeePay on the external platform. An access token will be issued by ShopeePay and used by the external platform for a fast and seamless payment experience in the future.

5) Ticket Linking Mode

The client generates a physical ticket that will be passed to the user, which contains information of the service provided by the client. The user will perform a one-time authorization to link their ShopeePay account to the ticket via scanning the ticket. Clients will be able to make a one-time ticket payment charge to the user's ShopeePay account for the linked ticket within a certain period after ticket linking.

Note: Ticket Linking Mode is currently only available in Indonesia.

2.2 Term Definitions

Term	Definition
Client	Refers to the party that integrate with ShopeePay APIs.
Merchant	A business entity that can have one or more stores operating under it.
Store	A shop at a physical location offering ShopeePay as a payment method.
Terminal ID	If terminal ID information is contained in QR code scanned by the user, the payment transaction will contain this identifier of the merchant's terminal that the payment was made at.
User ID Hash	For on-us payments, the user_id_hash is an alphanumeric string of the hashed ShopeePay userid making payment. For off-us payments, the user_id_hash is a string of the user making payment.
Signature	Signature is created based on shared secret key and algorithm, used by ShopeePay and client to verify each other's identity in every API request. The signature algorithm is created and provided by ShopeePay. The common signature modes used are HMAC, SHA-256, Base64.
Secret Key	A secret key that is shared between ShopeePay Server and the Client. This key is used to generate the Signature to authenticate the client. [Note] Never hard code the secret key in a frontend application. All API calls should only be made from the Client's servers.

3. System Requirements

3.1 Prerequisite

Once commercial agreement between ShopeePay and client is finalised, each client will be assigned the following for integration testing purposes.

Name	Description
Client ID	Unique identifier to identify the caller of each request, to be added to the request header of each API call.
Secret Key	Only ShopeePay and the Client should know the shared secret. Secret Key will be shared offline.

3.2 API Rules

3.2.1 Protocol Rules

This specifies the rules for calling APIs in this document.

Component	Format/Method
Transfer Mode	HTTPS
Submit Mode	POST Method
Date Format	UNIX
Char Encoding	UTF-8
Signature	HMAC, SHA-256, Base64

3.2.2 Parameter Specifications

3.2.2.1 Request Header

All requests will require this 2 parameters in the header

Name	Description
Client ID	Unique identifier for caller, issued by ShopeePay
Signature	Generated using the shared secret that is given by Shopee to authenticate the API caller.

3.2.2.2 Response Body

The ShopeePay API response body is in JSON format.

- Parameters can be returned in random sequence.
- Additional parameters may be returned in future without advance notice from ShopeePay.

Client system's logic should not assume that the order of arrangement of response parameters or the total number of parameters returned will remain constant throughout time.

3.2.2.3 Transaction Amount

Transaction amount will be:

- Integer format.
- Inflated by a factor of 100.

Example:

- Smallest amount will be 1, which represents 0.01.
- For currencies like IDR and VND, that do not have denomination in cents, the smallest amount will be 100, which represents 1 IDR (Rp 1) and 1 VND (đ 1).

There will not be a default currency. The current supported currency will be:

- IDR: Indonesian Rupiah
- MYR: Malaysian Ringgit

- PHP: Philippine Peso
- SGD: Singapore Dollar
- THB: Thai Baht
- VND: Vietnamese Dong

3.2.3 Security Specification

3.2.3.1 Signature

A signature should be generated from the request body to authenticate the request. A secret key is needed to sign the request body. Please request from our technical team if you do not have one.

The signature is a hash-based message authentication code (HMAC) using the SHA256 cryptographic function and the aforementioned secret key, operated on the request JSON. Code implementation using some common programming languages is listed below for your reference. You could check the implementation in more programming languages at this public [docs](#).

Python	<pre>import hmac import hashlib def generate_signature(request_json): return hmac.new(<secret_key>, request_json, hashlib.sha256) .digest() .encode('base64') .rstrip("\n")</pre>
Golang	<pre>import ("crypto/hmac" "crypto/sha256" "encoding/base64") // Sign - sign the `payload` with `key` func Sign(payload, key []byte) string { mac := hmac.New(sha256.New, key) // Cannot return error if _, err := mac.Write(payload); err != nil { return "" } return base64.StdEncoding.EncodeToString(mac.Sum(nil)) }</pre>
Javascript	<pre>let hash = CryptoJS.HmacSHA256(body, secret) let sig = CryptoJS.enc.Base64.stringify(hash).replace(/\n+\$/, "")</pre>

Include the signature in the request

The signature should be added in the HTTP header in the following format:

```
X-Airpay-ClientId: <clientid>
X-Airpay-Req-H: <request_signature>
```

Signature Example

Perform the following steps to sign the message.

Secret: pz148x0gXyPCLHxnLhEydNLg55jni91i

1. Extract the request body to be signed. For example:

```
{"request_id": "", "store_ext_id": "externalstore", "merchant_ext_id": "externalmerchant", "amount": 1000, "terminal_id": "terminal", "convenience_fee_percentage": 0, "convenience_fee_fixed": 0, "convenience_fee_indicator": "", "additional_info": "", "currency": "IDR", "qr_validity_period": 1000, "payment_reference_id": "testreference"}
```

2. Hash the JSON content by using the SHA256 algorithm.

5f5532bf23018674788770f43743522dff1af5782243fab06b1e8677a7391d4c

3. Encode to Base64

X1UyvyMBhnR4h3D0N0NSLf8a9XgiQ/qwax6Gd6c5HUw=

3.2.3.2 Signature Verification

Upon receiving the API response, the Client should verify the signature of the response in the following way:

1. Use the shared secret to sign the response body to obtain the response signature.
2. Compare the generated signature to the signature in the header of the API response.
3. Signatures from steps #1 and #2 must be identical. If they are different, the response should not be trusted.

3.2.4 Access Nodes

ShopeePay access nodes are country specific, each country will have their own API domains to be called.

Region Domain Suffix

Region	Abbreviation	Code
Indonesia	ID	co.id
Malaysia	MY	com.my
Philippines	PH	com.ph
Singapore	SG	sg
Thailand	TH	co.th
Vietnam	VN	vn

Append the region code to the end of the domain.

Environment	Domain
sandbox	api.uat.wallet.airpay.<region>
live	api.wallet.airpay.<region>

3.2.5 Backward Compatible Changes

Backward compatible changes refer to changes ShopeePay may make on their sole discretion without informing existing Clients in advance, as such changes are deemed to not break any integration implemented by the Client. During integration, Clients need to ensure their service does not get impacted by the following changes from ShopeePay:

- Add new API resource or new types of server callback to a new endpoint.
- Add new or remove optional field to existing API request parameters.
- Add new fields to existing API response/server callback parameters.
- Change the order of fields in existing API responses or server callback parameters.
- Increase the max length limit of request fields of existing API request parameters.

4. Payment Method API

Payment Method APIs refers to APIs that are used specifically for certain payment methods. Currently there are five payment methods: Merchant Presented Mode (MPM), Customer Presented Mode (CPM), App-to-app Redirection (Jump App), Account Linking Mode (Tokenized Payment), and Ticket Linking Mode.

4.1 Merchant Presented Mode (MPM)

4.1.1 User Payment Flow

- 1) The cashier generates dynamic QR code containing payment amount correspondingly, as shown in figure 4.1.1.
- 2) User clicks on “Scan” icon on ShopeePay homepage and scan the QR code, as shown in figure 4.1.2.
- 3) User confirms the payment amounts, as shown in figure 4.1.3.
- 4) User choose payment method and apply voucher if applicable, as shown in figure 4.1.4.
- 5) The user enter PIN to proceed with the payment, as shown in figure 4.1.5.
- 6) User is prompted of successful payment after completing the payment, as shown in figure 4.1.6. Merchant delivers the products to user after receiving notification of successful payment from ShopeePay system.

See screenshots on the following page for the front-end user flow on ShopeePay.

Figure 4.1.1 Dynamic QR generated by the cashier using Partner App or API



Figure 4.1.2 User scans QR code on Shopee app



Figure 4.1.3 User inputs or checks payment amount

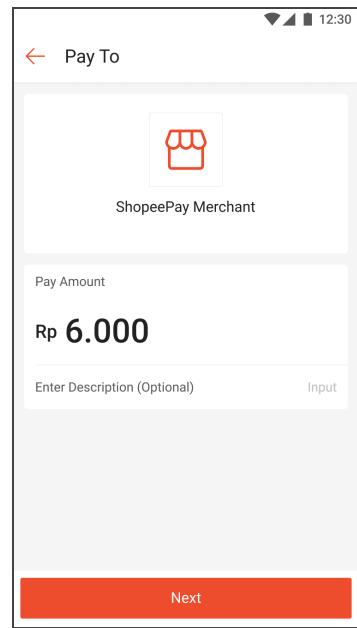


Figure 4.1.4 User confirms payment method and promotion

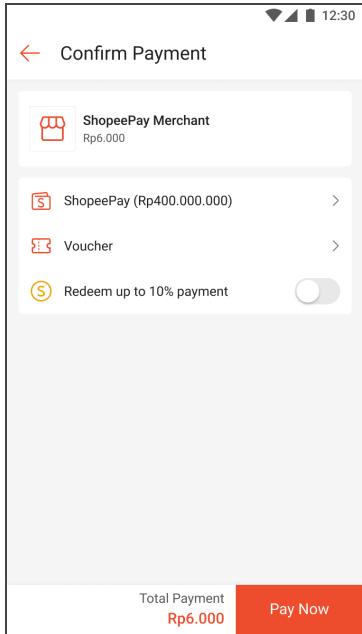


Figure 4.1.5 User authorise payment by using PIN or biometrics

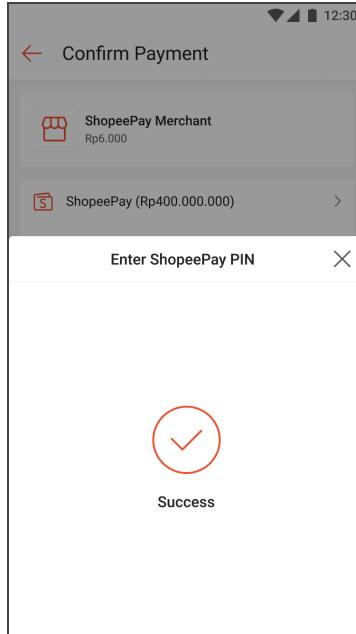
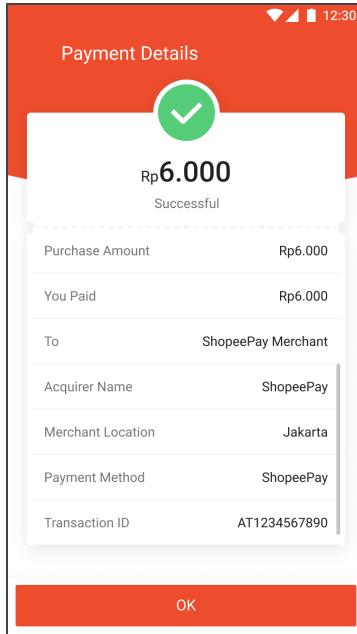
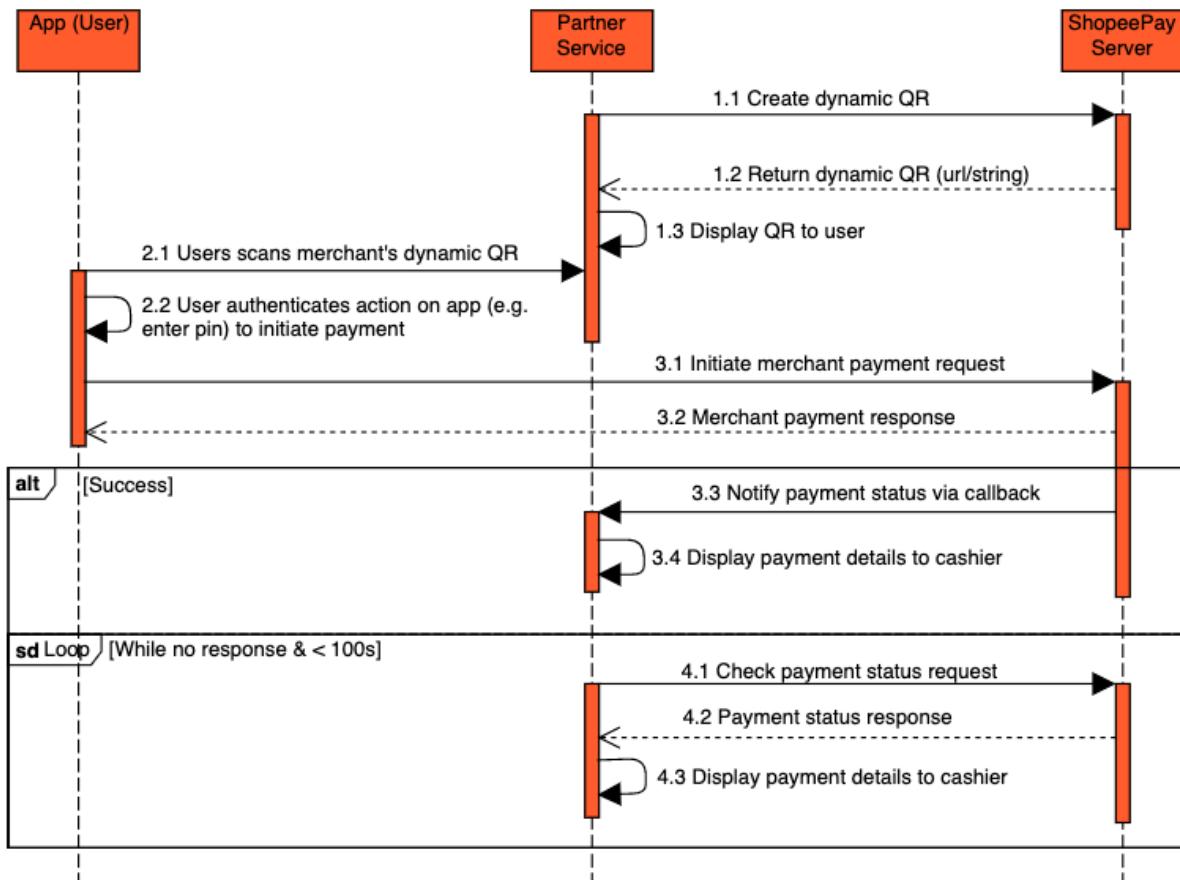


Figure 4.1.6 User redirected to transaction success page



4.1.2 Merchant Process



- 1) The client backend generates a QR code by calling ShopeePay system ([Create Dynamic QR API](#)) and displays it to the user to be scanned.
- 2) The user scan the QR code, checks payment amount and input PIN to proceed with the payment.
- 3) ShopeePay system returns the transaction result to user and the user is redirected to payment successful page.
- 4) ShopeePay also sends asynchronous message via callback URL ([Notify Transaction Status](#)) to inform client backend regarding payment result.
- 5) If no response from ShopeePay Server, Client can check transaction status by calling ShopeePay [Check Transaction Status API](#).
- 6) If Check Transaction Status API returns:

- a) Success - Payment process stops and user walks out of the store with purchase
 - b) Not found - Retry the payment process from the start
 - c) No response - continue retrying
 - i) Retry should be done at an incremental time range, at our recommended retry timing at the following time marks:
 - (1) 5 seconds
 - (2) 10 seconds
 - (3) 15 seconds
 - (4) 20 seconds
 - (5) 25 seconds
 - .
 - .
 - .
 - (6) 100 seconds
- 7) After 100 seconds of retry with no response from ShopeePay, client can consider payment transaction to have timed out.
 - 8) If buyer can show evidence of payment success on their payment app, merchant can choose to consider payment as successful and wait for reconciliation process on T+1 to settle discrepancies if any.

4.1.3 API Body

4.1.3.1 Create Dynamic QR API

Use this API to generate a dynamic QR code containing the payment amount and unique payment reference ID.

- URL: "/v3/merchant-host/qr/create"

Request Parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for this request.
amount	int64	Y	Amount used for the payment, refer to Parameters Specifications for specifics.
currency	string	Y	Refers to the currency abbreviation for the transaction. <ul style="list-style-type: none"> • MYR • IDR • PHP • SGD • THB • VND
merchant_ext_id	string	Y	Unique identifier of merchant in client system.
store_ext_id	string	Y	Unique identifier of store in client system.
payment_reference_id	string	Y	Unique identifier of transaction generated by client. Accepts up to <ul style="list-style-type: none"> • 25 characters in ID, MY, PH, SG • 64 characters in TH, VN
qr_validity_period	uint32	N	If this field is filled, the QR code containing the payment_reference_id in this request will expire after the specified time period (in seconds) and payment to this QR code will fail.

			<p>If this field is not filled, default shopeepay expiry time of 1200 seconds (20min) will apply, from the time the request was received.</p> <p>Accepts up to</p> <ul style="list-style-type: none"> • 3600 seconds (60 min) in MY, PH, SG • 172800 seconds (2 days) for VN • 432000 seconds (5 days) for ID, TH
expiry_time	uint32	N	<p>Unix timestamp.</p> <p>If this field is filled, the QR code containing the payment_reference_id in this request will expire and payment that QR code will fail after the specified timestamp.</p> <p>If this field is not filled, default shopeepay expiry time of 1200 seconds (20min) will apply, from the time the request was received.</p> <p>Accepts up to</p> <ul style="list-style-type: none"> • 3600 seconds (60 min) in MY, PH, SG • 172800 seconds (2 days) for VN • 432000 seconds (5 days) for ID, TH
terminal_id	string	N	<p>Unique identifier of store terminal.</p> <p>Accepts up to</p> <ul style="list-style-type: none"> • 25 characters in ID, MY, PH, SG • 20 characters in TH, VN
convenience_fee_indicator	string	N	<p>Convenience fee can be fixed, or percentage of the payment amount.</p> <ul style="list-style-type: none"> • 01 - User inputs free amount • 02 - Convenience fee is a fixed value, to be indicated in convenience_fee_fixed • 03 - Convenience fee is based on percentage of payment, to be

			<p>indicated in convenience_fee_percentage</p> <p>Note: This field is temporarily unavailable in Thailand and Vietnam. Reserved for future usage.</p>
convenience_fee_fixed	int64	N	<p>Required if convenience fee is a fixed fee as indicated by convenience fee indicator</p> <p>Note: This field is temporarily unavailable in Thailand and Vietnam. Reserved for future usage.</p>
convenience_fee_percentage	uint32	N	<p>Required if convenience fee is percentage as indicated by convenience fee indicator.</p> <p>Note: This field is temporarily unavailable in Thailand and Vietnam. Reserved for future usage.</p>
promo_ids	string	N	<p>Comma separated eligible promo_ids of 100 characters or less (up to 20), if any.</p> <p>These promo_ids will be checked against promotional campaigns configured in ShopeePay to determine the valid reward amount the user can receive for this transaction.</p>
additional_info	string	N	<p>Additional information will be in json format.</p> <p>This field supports up to 3 fields and needs to be sent in this format <code>{"field1\":\"{Data1}\",\"field2\":\"{Data2}\",\"field3\":\"{Data3}\")</code></p> <p>Note: The field's format is only applicable in ID, TH, VN.</p>

Response Parameters

Field	Type	Description
request_id	string	Unique identifier for request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.
store_name	string	Name of the store onboarded to Shopeepay.
qr_content	string	QR string in plain text.
qr_url	string	URL to download QR image, URL is valid for 5 minutes.

Sample Request

```

1  {
2    "request_id": "unique-request-id-here",
3    "payment_reference_id": "ref-must-be-unique",
4    "merchant_ext_id": "externalmerchant",
5    "store_ext_id": "externalstore",
6    "amount": 10000,
7    "terminal_id": "T2903",
8    "currency": "IDR",
9    "qr_validity_period": 1600,
10   "convenience_fee_percentage": 0,
11   "convenience_fee_fixed": 0,
12   "convenienccce_fee_indicator": "01",
13   "promo_ids": "1234567890,2345678901",
14   "additional_info": "{\"field1\":\"Name\", \"field2\":\"Customer ID\", \"field3\":\"Phone Number\"}"
15 }
```

Sample Response

```

1  {
2    "request_id": "unique-request-id-here",
3    "errcode": 0,
4    "debug_msg": "success",
5    "qr_content": "00020101021226580016ID.CO.SHOPPEE.
WWW011893600918002024585002082024585003005204123453033605406100.
005502015802ID59116007Bandung61054016362280518ref-must-be-unique0702A1630483F6",
6    "qr_url": "https://api.uat.wallet/airpay.co.id/v3/merchant-host/qr/download?
qr=HNMAQjc46NMNEUekDbU5AXX7qA8EI8lghCSLFriKb9",
7    "store_name": "externalstore"
8 }
```

4.1.3.2 Notify Transaction Status

Please refer to [Notify Transaction Status](#)

4.1.3.3 QR Invalidate API

Use this API to invalidate the payment reference in a QR code if the order in Client system is closed/cancelled. Once QR is invalidated successfully, the QR code containing the payment reference ID can no longer be used to make payment.

- URL: "/v3/merchant-host/qr/invalidate"

Request Parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for request.
merchant_ext_id	string	Y	Unique identifier of merchant in client system.
store_ext_id	string	Y	Unique identifier of store in client system.
payment_reference_id	string	Y	Payment reference ID that corresponds to the QR being invalidated.

Response Parameters

Field	Type	Description
request_id	string	Unique identifier for request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.

Sample Request

```

1  {
2    "request_id": "unique-request-identifier-here",
3    "payment_reference_id": "payment-ref-in-qr",
4    "merchant_ext_id": "externalmerchant",
5    "store_ext_id": "externalstore"
6  }

```

Sample Response

```
1  {
2    ... "request_id": "unique-request-identifier-here",
3    ... "errcode": 0,
4    ... "debug_msg": "success"
5 }
```

4.1.3.4 Check Transaction Status API

Please refer to [Check Transaction Status API](#)

4.1.3.5 Create Refund API [New]

Please refer to [Create Refund API \[New\]](#)

4.2 Customer Presented Mode (CPM)

4.2.1 User Payment Flow

- 1) User have to click on “Pay” button as the entry point for CPM option in QR Offline Payments, as shown in figure 4.2.1.
- 2) User is prompted to enter ShopeePay pin or biometrics before showing the QR/barcode, as shown in figure 4.2.2.
- 3) User QR is shown on ShopeePay, as shown in figure 4.2.3. Merchant scans user QR and submits payment requests to ShopeePay system. User can also choose to pay using barcode, as shown in figure 4.2.4.
- 4) If payment is successful, User is shown the successful payment screen, as shown in figure 4.2.5.

See screenshots on the following page for the front-end user flow on ShopeePay.

Figure 4.2.1 User click on "Pay" from Home Page

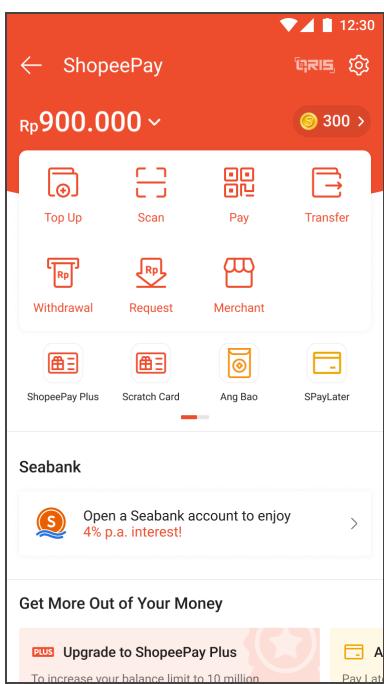


Figure 4.2.2 User is prompted to enter ShopeePay pin

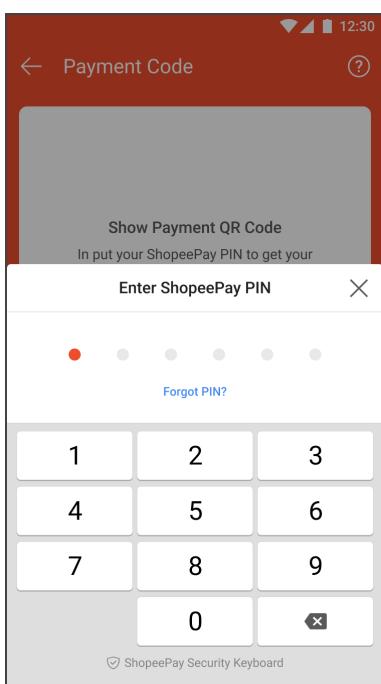


Figure 4.2.3 User CPM QR/barcode is ready to be scanned by merchant

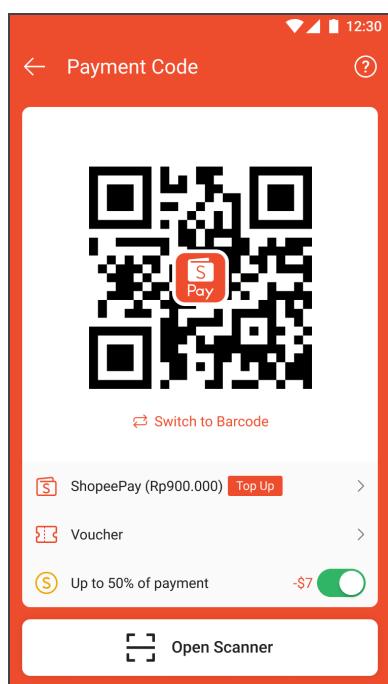


Figure 4.2.4 Switch to barcode payment method

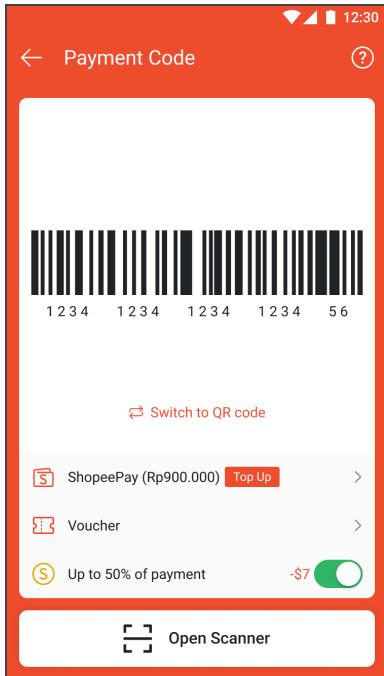
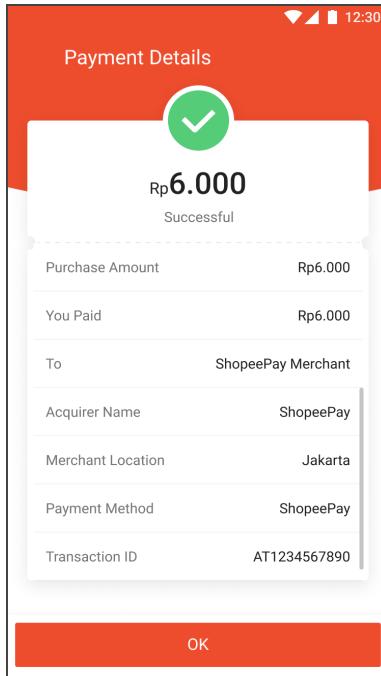
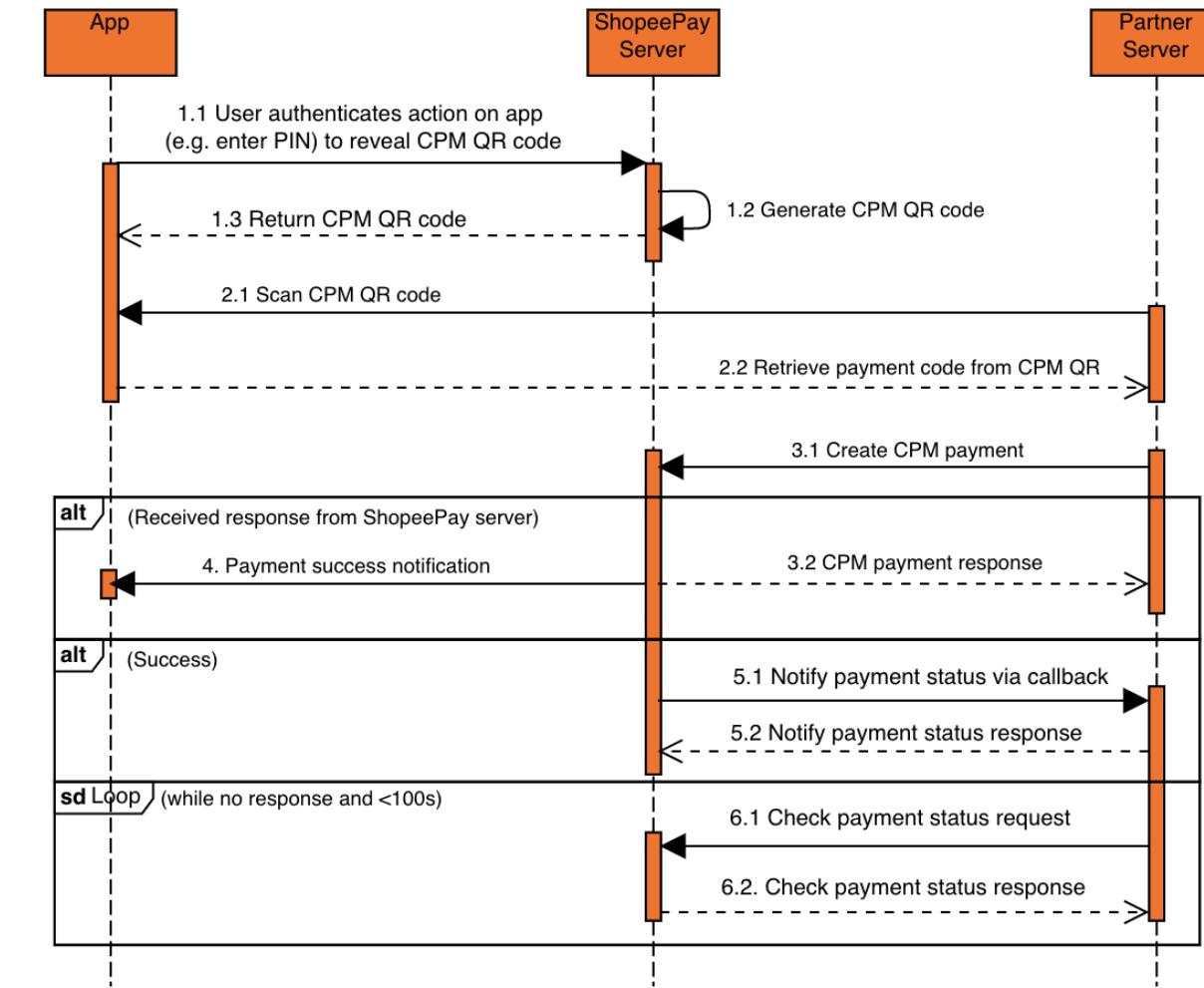


Figure 4.2.5 User redirected to payment success page



4.2.2 Merchant Process



- 1) User clicks on the “Pay” button on ShopeePay homepage to generate QR/barcode.
- 2) Cashier scans the user’s QR/barcode.
- 3) Client calls [Create CPM Payment](#) API on ShopeePay Server with the transaction value.
- 4) ShopeePay system returns the transaction result to the client, and informs the user of the successful payment on ShopeePay App.
- 5) ShopeePay also sends an asynchronous message via the callback URL ([Notify Transaction Status](#)) to inform the client backend of the payment result.
- 6) If there’s no response from ShopeePay Server before request timeout (time out duration is determined by Client), Client can check transaction status by calling [Check Transaction Status API](#).

7) If [Check Transaction Status API](#) returns:

- a) Success - Payment process stops and user walks out of the store with purchase
 - b) Not found - Retry the payment process from the start
 - c) No response - continue retrying
 - i) Retry should be done at an incremental time range, at our recommended retry timing at the following time mark:
 - 5 seconds
 - 10 seconds
 - 15 seconds
 - 20 seconds
 - .
 - .
 - .
 - 100 seconds
- 8) After 100 seconds, payment transaction will timeout (ShopeePay will issue a timeout on that request) and the merchant should refund the original payment transaction before attempting payment creation again.
- 9) Calling [Create Refund API](#) will return 2 status types:
- a) Success - transaction successfully refunded, user can try paying again.
 - b) Fail - transaction can be found but it does not meet the requirements for refund, merchant can try again at a later time.
 - c) If the payment transaction is not found, or payment is still processing, calling [Create Refund API](#) will return an error code (see [Appendix 5 - Error Codes](#)) to indicate failure.
- 10) If the user making the payment can show evidence of payment success on their payment app, merchant can choose to consider payment as successful and wait for reconciliation process on T+1 to settle discrepancies if any.

4.2.3 API Body

4.2.3.1 Create CPM Payment

Use this API to initiate a payment request after scanning a user's CPM QR.

- URL: "/v3/merchant-host/transaction/payment/create"

Request Parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for request.
amount	uint64	Y	Amount used for the payment, refer to Parameters Specifications for specifics.
merchant_ext_id	string	Y	Unique identifier of merchant in client system.
store_ext_id	string	Y	Unique identifier of store in client system.
payment_code	string	Y	Payment code is the QR/barcode content from the user's CPM QR code. If there's no national QR code standard defined on a national level, this code starts with "918" and contains 20 digits.
payment_reference_id	string	Y	Unique identifier of transaction generated by client.
validity_period	uint32	N	If this field is filled, the order with the corresponding payment_reference_id in this request will expire after the specified validity period (in seconds). The minimum validity period is 60s.
expiry_time	uint32	N	Unix timestamp. If this field is filled, the order with the corresponding payment_reference_id

			in this request will fail after the specified timestamp. The minimum expiry time will be 60s from the request time.
terminal_id	string	N	Terminal ID refers to the Point of Sale terminal in a store.
currency	string	Y	Refers to the currency abbreviation for the transaction. <ul style="list-style-type: none"> • MYR • IDR • PHP • SGD • THB • VND
promo_ids	string	N	Comma separated eligible promo_ids of 100 characters or less (up to 20), if any. These promo_ids will be checked against promotional campaigns configured in ShopeePay to determine the valid reward amount the user can receive for this transaction.
additional_info	string	N	Additional information will be in json format. This field supports up to 3 fields and needs to be sent in this format <pre>{"field1\":\"{Data1}\",\"field2\":\"{Data2}\",\"field3\":\"{Data3}\")}</pre> Note: The field's format is only applicable in ID, TH, VN.

Response Parameters

Field	Type	Description
request_id	string	Unique identifier for request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.
transaction_list*	object	List of transactions. Will return Nil if transaction fails.
• reference_id	string	Unique identifier of payment transaction generated by client.
• amount	int64	Payment amount - inflated by a factor of 100.
• create_time	uint32	Create time of the individual transaction.
• update_time	uint32	Update time of the individual transaction.
• transaction_sn	string	Transaction identifier in ShopeePay system.
• status	uint32	Refer to Appendix 1C - Transaction Status for specific transaction statuses.
• transaction_type	uint32	13 (Refer to Appendix 1A - Transaction Types for specific transaction types).
• merchant_ext_id	string	Unique identifier of merchant in client system.
• terminal_id	string	Terminal ID refers to the Point of Sale terminal in a store.
• user_id_hash	string	Used to identify the unique user.
• store_ext_id	string	Unique identifier of store in client system.
• payment_channel	uint32	Indicates the source of fund used. Refer to Appendix 1E - Payment Channels for the respective source of fund.
co_funding	object	Will return empty if no co-funding involved. Note: this field is temporarily not available in ID, MY, PH, SG.

<ul style="list-style-type: none"> ● promotion_ext_name 	string	Name of the promotion for client's reference.
<ul style="list-style-type: none"> ● redeemed_promotion_amount 	int64	<p>Amount of the promotion benefit obtained by user through co-funding.</p> <p>Note: for refund transaction types, this will be the amount of promotion benefit ShopeePay clawbacks from the user in that refund transaction.</p> <p>Refer to Parameters Specifications for specifics.</p>
<ul style="list-style-type: none"> ● redeemed_promotion_type 	uint32	Refer to Appendix 1D - Promotion Types for specific transaction types.
<ul style="list-style-type: none"> ● co_funding_cost_amount 	int64	<p>Amount of cost born by merchant for the promotion benefit.</p> <p>Note: for refund transaction types, this will be the amount of co-funding amount that the client will no longer be charged for due to that refund transaction.</p> <p>Refer to Parameters Specifications for specifics.</p>

*[Note]: in Thailand and Vietnam, the name of object "transaction_list" is returned as "transaction".

Sample Request

```

1  {
2    ...
3    "request_id": "unique-request-id-here",
4    ...
5    "amount": 10000,
6    ...
7    "merchant_ext_id": "externalmerchant",
8    ...
9    "store_ext_id": "externalstore",
10   ...
11   "payment_code": "hQVDUFYwMWF1TwegAAAGAiAgUAdxcm1zY3BtWgqTYAkYMAkBV0BOxyASUGVuZ2d1bmEgU2hvcGVlUGF5nyUCdARj05900FghWEsvTmhuMgdWMHp11zI2U29IMUJ2WENqZkpSZGVza0h5bVIwVUJWRnpZb3p0ZUFBQUFBQT09",
12   ...
13   "payment_reference_id": "external-payment-reference-1",
14   ...
15   "expiry_time": 1664258483,
16   ...
17   "terminal_id": "T2903",
18   ...
19   "currency": "IDR",
20   ...
21   "promo_ids": "1234567890,2345678901",
22   ...
23   "additional_info": "{\"field1\": \"Name\", \"field2\": \"Customer ID\", \"field3\": \"Phone Number\"}"
24 }
```

Sample Response

```
1  {
2    "request_id": "unique-request-id-here",
3    "errcode": 0,
4    "debug_msg": "success",
5    "transaction_list": [
6      {
7        "reference_id": "external-payment-reference-id",
8        "amount": 10000,
9        "create_time": 1681473904,
10       "update_time": 1681473905,
11       "transaction_sn": "168301613660829325",
12       "status": 3,
13       "transaction_type": 13,
14       "merchant_ext_id": "externalmerchant",
15       "terminal_id": "T2903",
16       "user_id_hash": "15e455125fba426a4a2bb9145b3af2906813a7f222f6eab93b026ead62dc8d76",
17       "store_ext_id": "externalstore",
18       "payment_channel": 1
19     }
20   ]
21 }
```

4.2.3.2 Notify Transaction Status

Please refer to [Notify Transaction Status](#).

4.2.3.3 Check Transaction Status API

Please refer to [Check Transaction Status API](#).

4.2.3.4 Create Refund API [New]

Please refer to [Create Refund API \[New\]](#).

4.3 App-to-app Redirection (Jump App)

4.3.1 User Payment Flow

- 1) User selects ShopeePay as the payment method during checkout/purchase on the Client's application, as shown in Figure 4.3.1.
- 2) User clicks on the payment confirmation button in the Client's application to confirm they are paying with ShopeePay, as shown in Figure 4.3.2.
- 3) External application checks whether Shopee App is installed on user's device.
 - a) If not, notify or redirect the user to install Shopee App first before completing payment.
 - b) If yes, redirect the user to the ShopeePay payment amount page on Shopee app, as shown in Figure 4.3.3.
- 4) User choose payment method and apply voucher if applicable, as shown in figure 4.3.4
- 5) Upon selecting payment options/vouchers in ShopeePay (if any), User enters their ShopeePay PIN to proceed with the payment, as shown in Figure 4.3.5.
- 6) If the payment is successful, User sees the successful payment screen, as shown in Figure 4.3.6.
- 7) User jump backs to the original Client's application and view their order status, as shown in Figure 4.3.7. External application can then release the order to the user after receiving notification of a successful payment from ShopeePay system.

See screenshots on the following page for the front-end user flow on ShopeePay.

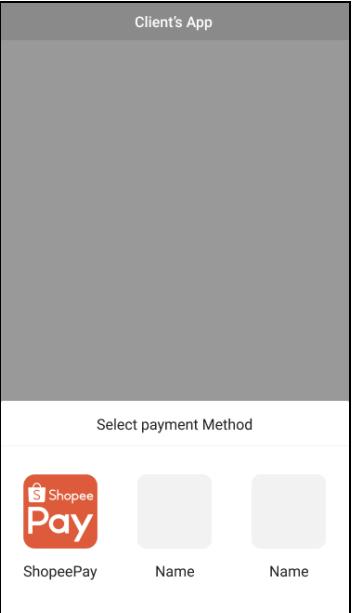
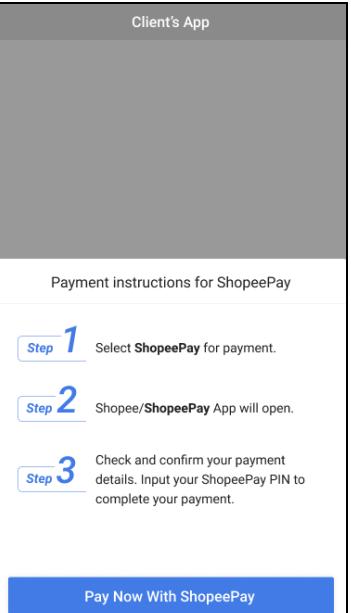
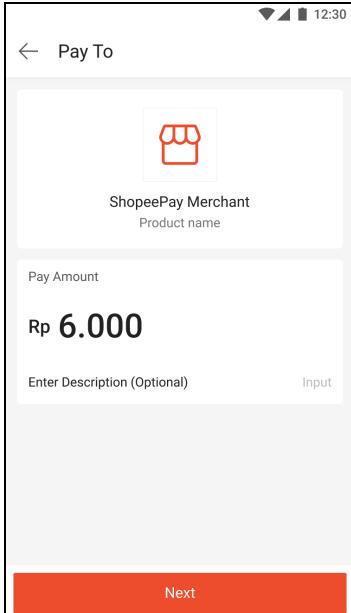
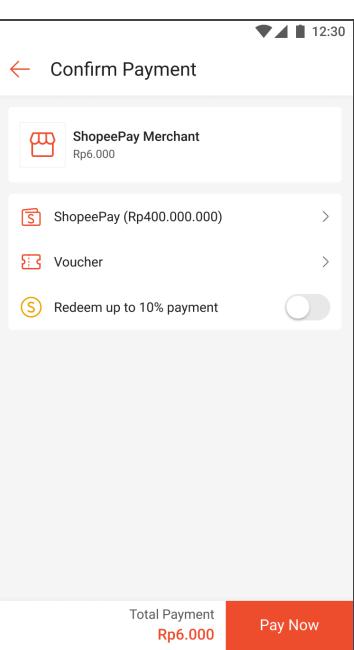
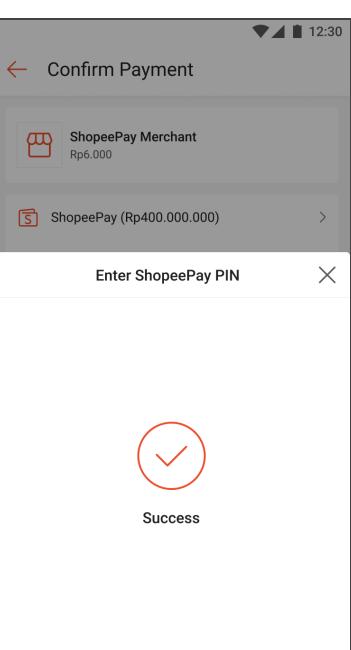
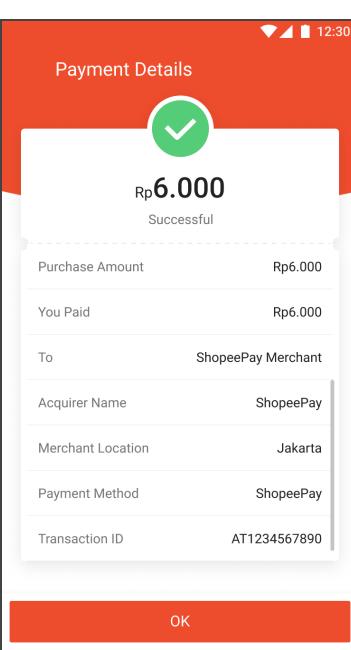
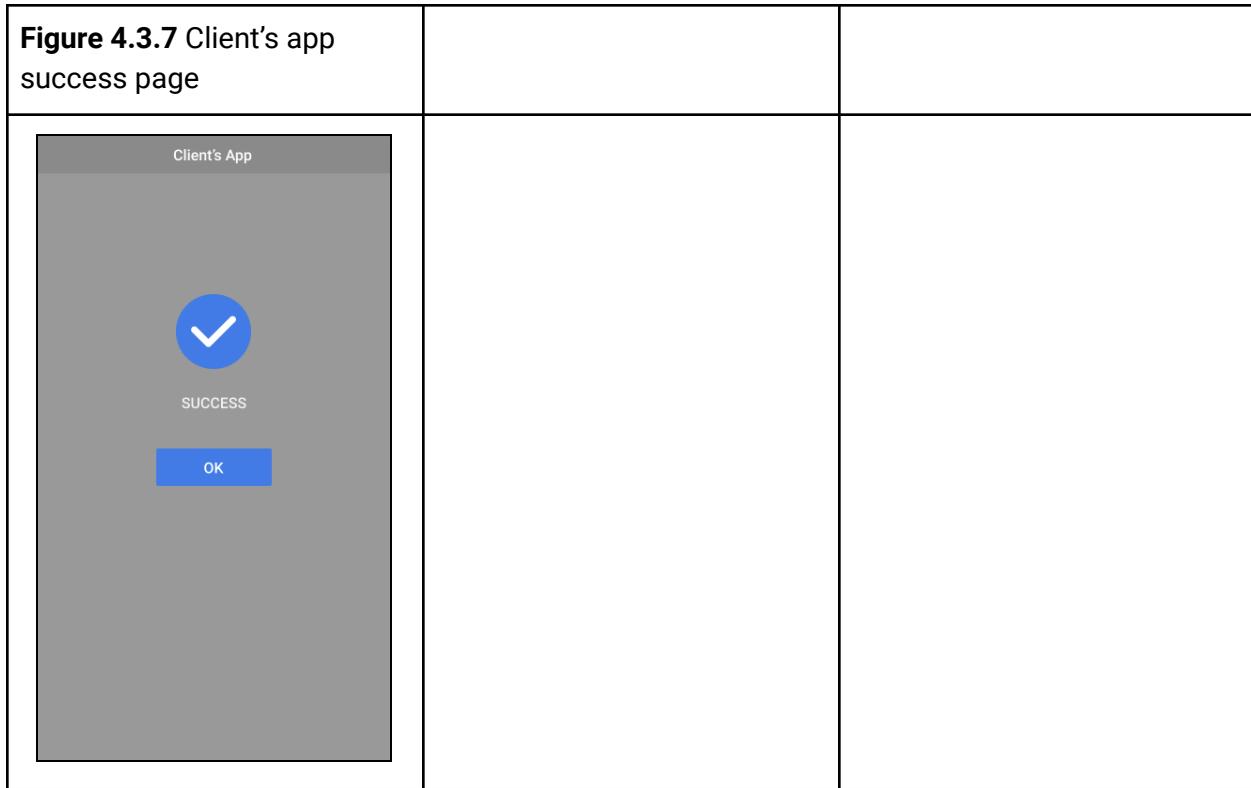
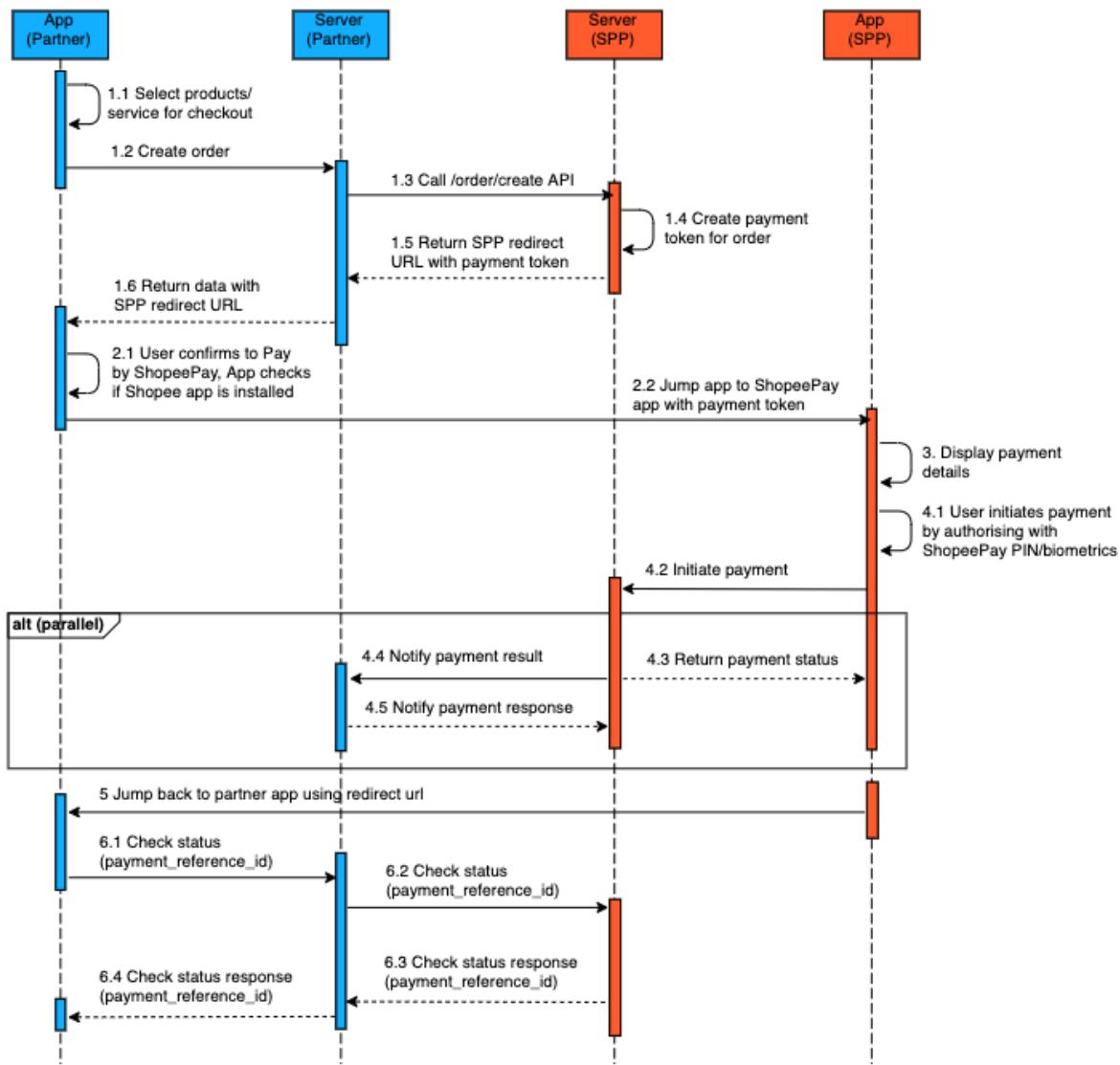
<p>Figure 4.3.1 User selects ShopeePay as payment method</p>	<p>Figure 4.3.2 User confirms they are paying with ShopeePay</p>	<p>Figure 4.3.3 Payment Amount page</p>
		
<p>Figure 4.3.4 User confirms payment method and promotion</p>	<p>Figure 4.3.5 User authorize payment by using PIN or biometrics</p>	<p>Figure 4.3.6 User redirected to payment success page</p>
		

Figure 4.3.7 Client's app success page



4.3.2 Merchant Process



- 1) After User creates an order on the Client's app and confirms to pay by ShopeePay, Client's app calls [Create Order API](#) to get the redirect URLs to ShopeePay.
- 2) User clicks on the redirect URL to jump to ShopeePay payment screen. **Before the user is redirected to ShopeePay, the Client's app should check if the user's device has Shopee App installed.**
 - a) If the Shopee App is not installed, it is recommended that the Client app redirects the user to install the shopee app using the `redirect_url_http` link returned in [Create Order API](#), for an optimal user experience.

- b) If Shopee App is installed, the Client's app can direct the user to the ShopeePay payment page.
- 3) ShopeePay app displays order details using payment identifier in redirect URL.
- 4) User checks the payment amount and inputs Shopeepay PIN to complete the payment.
- 5) ShopeePay system returns the transaction result and the user is redirected to payment successful page. From the payment success page in ShopeePay, user can jump back to the Client's app using the *return_url* to view their order status.
[Note] Never use the redirect to *return_url* as an indication of payment success. Client should only rely on server request/response to check the final status of payment.
- 6) ShopeePay sends the transaction status update via callback URL ([Notify Transaction Status](#)) to inform the Client about the payment result.
- 7) Client can also call [Check Transaction Status API](#) to check the payment status.
- 8) If Check Transaction API returns:
 - a) Success - Payment process stops and user walks out of the store with purchase
 - b) Not found - Retry the payment process from the start
 - c) No response - continue retrying
 - i) Retry should be done at an incremental time range, at our recommended retry timing at the following time marks:
 - (1) 5 seconds
 - (2) 10 seconds
 - .
 - .
 - .
 - (5) 30 seconds
- 9) After 30 seconds of retry with no response from ShopeePay, client can consider payment transaction to have timed out. Differences can be settled during the T+1 reconciliation process with ShopeePay.

4.3.3 API Body

4.3.3.1 Create Order

Use this API to create a pending order to be paid by ShopeePay.

- URL: "/v3/merchant-host/order/create"

Request Parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for request.
payment_reference_id	string	Y	Unique identifier of transaction generated by client.
merchant_ext_id	string	Y	Unique identifier of merchant in client system.
store_ext_id	string	Y	Unique identifier of store in client system.
amount	int64	Y	Amount used for the payment, refer to Parameters Specifications for specifics.
currency	string	Y	Refers to the currency abbreviation for the transaction. <ul style="list-style-type: none"> • MYR • IDR • PHP • SGD • THB • VND
return_url	string	Y	Indicates the URL of the Client's app to redirect back to once payment has been completed in ShopeePay. Shopeepay accepts HTTP links and URL scheme formats.
platform_type	string	Y	This indicates the type of the Client's platform that the user triggered the jump app flow from.

			<p>Accepted values:</p> <ul style="list-style-type: none"> • "app" • "pc" • "mweb" (mobile web)
validity_period	uint32	N	<p>If this field is filled, the order with the corresponding payment_reference_id in this request will expire after the specified time period (in seconds) and payment attempts to this payment_reference_id will fail.</p> <p>If this field is not filled, default shopeepay expiry time of 1200 seconds (20min) will apply, from the time the request was received.</p> <p>Accepts up to</p> <ul style="list-style-type: none"> • 3600 seconds (60 min) in MY, PH, SG • 172800 seconds (2 days) for VN • 432000 seconds (5 days) for ID, TH
expiry_time	uint32	N	<p>Unix timestamp.</p> <p>If this field is filled, the order with the corresponding payment_reference_id in this request will expire after the specified timestamp and payment attempts to this payment_reference_id will fail.</p> <p>If this field is not filled, default shopeepay expiry time of 1200 seconds (20min) will apply, from the time the request was received.</p> <p>Accepts up to</p> <ul style="list-style-type: none"> • 3600 seconds (60 min) in MY, PH, SG • 172800 seconds (2 days) for VN • 432000 seconds (5 days) for ID, TH

promo_ids	string	N	<p>Comma separated eligible promo_ids of 100 characters or less (up to 20), if any.</p> <p>These promo_ids will be checked against promotional campaigns configured in ShopeePay to determine the valid reward amount the user can receive for this transaction.</p>
additional_info	string	N	<p>Additional information will be in json format.</p> <p>This field supports up to 3 fields and needs to be sent in this format <code>{"field1":"{Data1}","field2":"{Data2}" , "field3":"{Data3}"}</code></p> <p>Note: The field's format is only applicable in ID, TH, VN.</p>

Response Parameters

Field	Type	Description
request_id	string	Unique identifier for request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.
redirect_url_app	string	<p>Returns an URL scheme to ShopeePay payment page.</p> <p>Sample link: <code>shopeeid://main?aprl=%2Frn%2FTRANSFER_PAGE%3Fnavigate_url%3Dhttps%253A%252F%252Fwallet.airplay.co.id%252Fwallet%252Fpay%253Ftoken%253D{paymentToken}</code></p>
redirect_url_http	string	<p>Returns a universal link to ShopeePay payment page. This link is recommended when the Client is unable to implement a check for whether Shopee app is installed on the user's device before redirection.</p> <p>Sample link:</p>

		https://wsa.uat.wallet.airpay.co.id/universal-link/wallet/pay?deep_and_deferred=1&token={paymentToken}
--	--	---

Sample Request

```

1  {
2    "request_id": "unique-request-id-here",
3    "payment_reference_id": "ref-must-be-unique",
4    "merchant_ext_id": "externalmerchant",
5    "store_ext_id": "externalstore",
6    "amount": 1000,
7    "currency": "IDR",
8    "return_url": "https://shopee.co.id",
9    "platform_type": "app",
10   "validity_period": 1000,
11   "promo_ids": "1234567890,2345678901",
12   "additional_info": "{\"field1\":\"Name\",\"field2\":\"Customer ID\",\"field3\":\"Phone Number\"}"
13 }
```

Sample Response

```

1  {
2    "request_id": "unique-request-id-here",
3    "errcode": 0,
4    "debug_msg": "success",
5    "redirect_url_app": "shopeeid://main?
  app1=%2Fr%2FTRANSFER_PAGE%3Fnavigate_url%3Dhttps%253A%252F%252Fwsa.uat.wallet.airpay.co.
  id%252Fwallet%252Fpay%253Ftoken%253DdFhkbmR1bTBIamhW6EeWLj-y6biZ0ao1VyS10GEPxzPR7vdkZtbVewqst8Z_0
  g",
6    "redirect_url_http": "https://wsa.uat.wallet.airpay.co.id/universal-link/wallet/pay?
  deep_and_deferred=1&token=dFhkbmR1bTBIamhW0FEdcx036uGGiu9LRzq4gSVbDoP61GwpU_d00jUFJ4B0pQ"
7 }
```

[Note] In Thailand and Vietnam, when ShopeePay redirects user back to the return_url provided in the Client's API request, the following parameters may be appended in the return_url. Client may utilize such information to display some preliminary results to users on front end.

Example:

https://www.google.com/?amount=2000&client_id=11000193&reference_id=Test_returnURL_03&result_code=100&signature=hw8FvuV1oVLTNll7Xafawfqv1BGZsceeTpzo%3D&transaction_sn=1007011238

Field	Type	Mandatory	Description
amount	string	Y	Amount used for the payment, refer to Parameters Specifications for specifics.
client_id	string	Y	Unique identifier to identify the caller; issued by ShopeePay.

reference_id	string	Y	<p>Unique identifier of payment transaction generated by client.</p> <p>Same value as the "payment_reference_id" in the request.</p>
result_code	string	Y	<p>Preliminary status of the payment order for Client's front-end display only.</p> <p>[Note] Never take this as an indication of actual payment status. Client should only rely on server request/response to check the final status of payment.</p> <p>Enumerations:</p> <ul style="list-style-type: none"> - 100: payment success - 201: payment cancelled but the redirection_url is still usable - 202: payment failed - 203: payment still in processing - 204: payment expired
transaction_sn	string	N	<p>Transaction identifier in ShopeePay system. Only returned when a transaction is created in ShopeePay system.</p>
signature	string	Y	<p>Encrypted signature of the above parameters using the Secret Key. Client should verify the signature first before consuming any parameters above.</p> <p>Base64.URLEncoding applies here, using the Secret issued by ShopeePay.</p>

4.3.3.2 Notify Transaction Status

Please refer to [Notify Transaction Status](#).

4.3.3.3 Order Invalidate API

Use this API to invalidate the existing order payment reference id for the pending order in the Client system if it is closed/cancelled. Once the order payment reference id is invalidated successfully, the corresponding “redirect_url_app” and “redirect_url_http” can no longer be used to make payment.

- URL: "/v3/merchant-host/order/invalidate"

Request Parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for request
merchant_ext_id	string	Y	Unique identifier of merchant in client system
store_ext_id	string	Y	Unique identifier of store in client system
payment_reference_id	string	Y	Payment reference ID that corresponds to the order being invalidated

Response Parameters

Field	Type	Description
request_id	string	Unique identifier for request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.

Sample Request

```
1  {
2    ... "request_id": "unique-request-id-here",
3    ... "merchant_ext_id": "externalmerchant",
4    ... "store_ext_id": "externalstore",
5    ... "payment_reference_id": "original-jump-app-payment-ref-to-be-invalidate"
6 }
```

Sample Response

```
1  {
2    ... "request_id": "unique-request-id-here",
3    ... "errcode": 0,
4    ... "debug_msg": "success"
5 }
```

4.3.3.4 Check Transaction Status API

Please refer to [Check Transaction Status API](#).

4.3.3.5 Create Refund API [New]

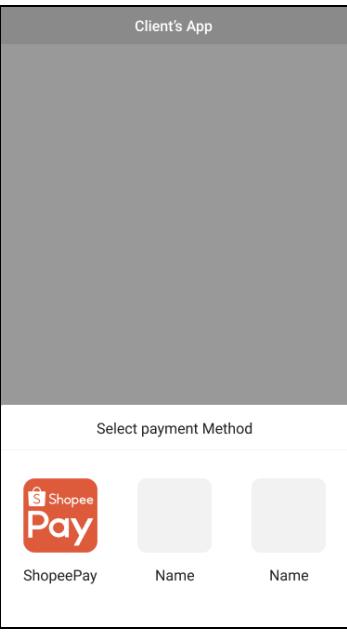
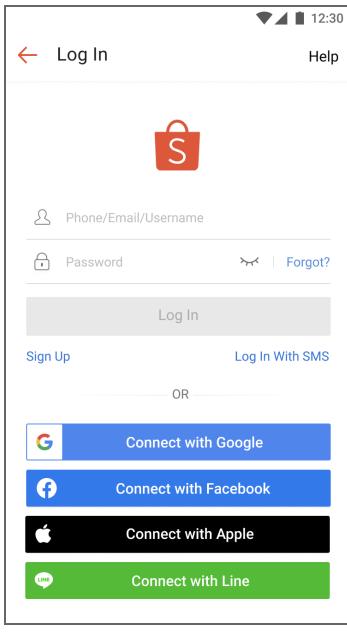
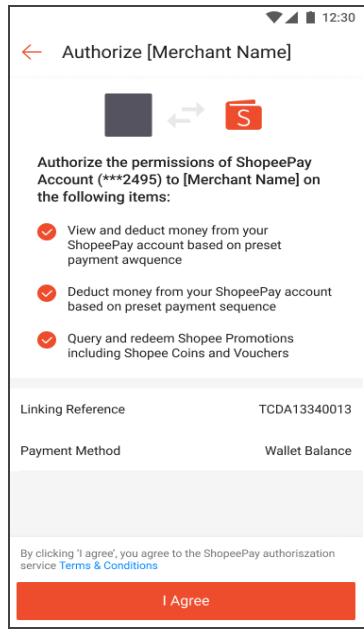
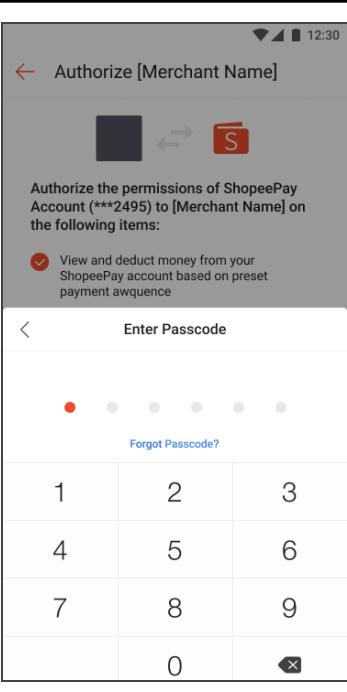
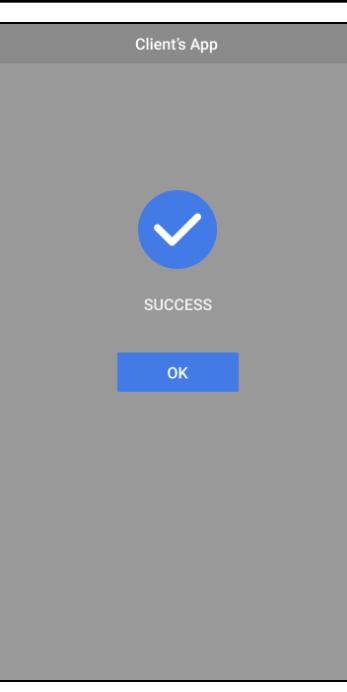
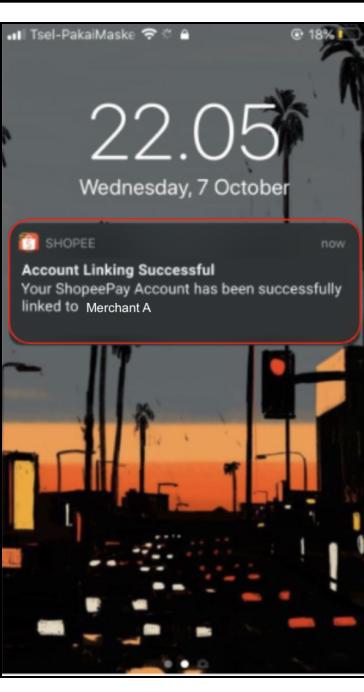
Please refer to [Create Refund API \[New\]](#).

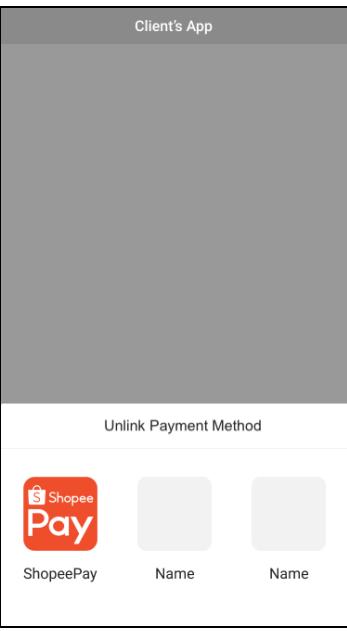
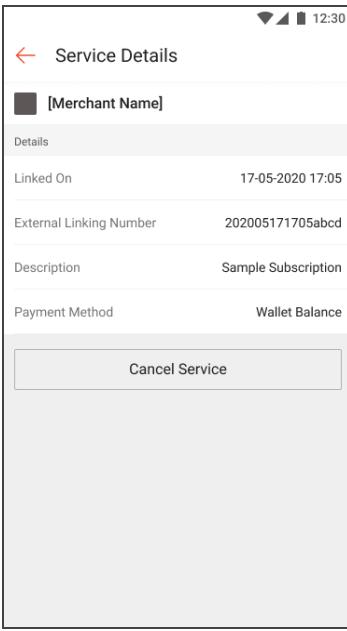
4.4 Account Linking (Tokenized Payment)

4.4.1 User Link and Unlink Account Flow

- 1) User links their ShopeePay account to the Client's application during checkout/purchase on the Client's application, as shown in Figure 4.4.1.
- 2) If User has not logged in yet, User will be redirected to login page, as shown in Figure 4.4.2. (Note: Login page UI in Thailand and Vietnam are slightly different).
- 3) After login, User is redirected to the ShopeePay Agreement Page to authorise the linking, as shown in Figure 4.4.3.
- 4) User enters their ShopeePay PIN, as shown in Figure 4.4.4.
- 5) User sees the result of the account linking on the ShopeePay application and/or Client's application, as shown in Figure 4.4.5.
- 6) Upon successful account linking, User receives a push notification from ShopeePay/Shopee App, as shown in Figure 4.4.6.
- 7) User can unlink ShopeePay via the Client's platform or via the ShopeePay/Shopee App, as shown in Figure 4.4.7 and Figure 4.4.8 respectively.
- 8) Once the ShopeePay account is unlinked, User is not able to access ShopeePay from the Client's platform anymore.

See screenshots on the following page for the front-end user flow.

<p>Figure 4.4.1 User chooses to link ShopeePay</p>	<p>Figure 4.4.2 User logs in first if not logged in the web yet</p>	<p>Figure 4.4.3 User gives consent on Agreement page</p>
		
<p>Figure 4.4.4 User enters PIN to verify</p>	<p>Figure 4.4.5 User sees the result of the account linking</p>	<p>Figure 4.4.6 User receives a push notification upon successful linking</p>
		

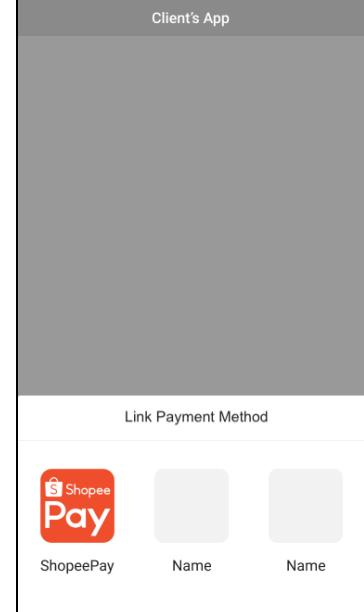
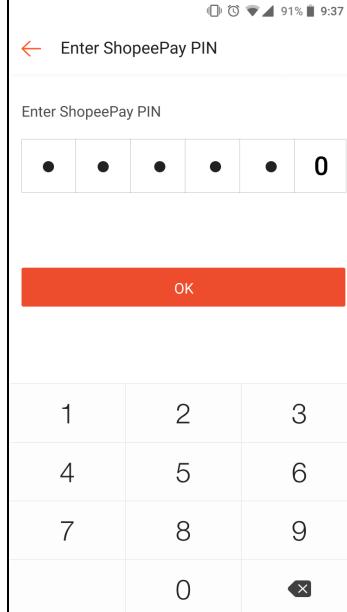
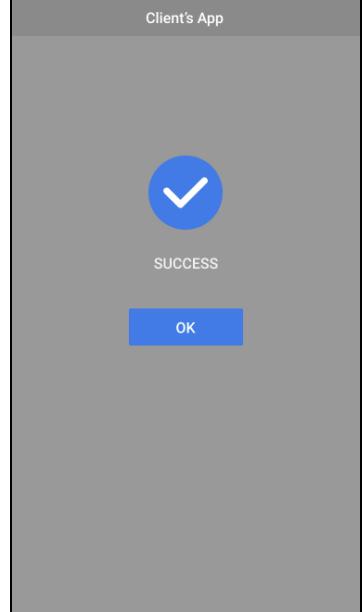
<p>Figure 4.4.7 User can unlink ShopeePay account via Client's App</p>	<p>Figure 4.4.8 User can unlink from the Client's platform via ShopeePay/Shopee App</p>											
 <p>The screenshot shows a dark grey header labeled "Client's App". Below it is a white section titled "Unlink Payment Method". It displays a ShopeePay logo icon followed by two placeholder boxes labeled "Name".</p>	 <p>The screenshot shows a mobile device screen with a top bar indicating signal strength, battery level, and time (12:30). Below this is a navigation bar with a back arrow and the text "Service Details". The main content area displays the following information:</p> <table border="1"> <thead> <tr> <th colspan="2">Details</th> </tr> </thead> <tbody> <tr> <td>Linked On</td> <td>17-05-2020 17:05</td> </tr> <tr> <td>External Linking Number</td> <td>202005171705abcd</td> </tr> <tr> <td>Description</td> <td>Sample Subscription</td> </tr> <tr> <td>Payment Method</td> <td>Wallet Balance</td> </tr> </tbody> </table> <p>At the bottom is a large button labeled "Cancel Service".</p>	Details		Linked On	17-05-2020 17:05	External Linking Number	202005171705abcd	Description	Sample Subscription	Payment Method	Wallet Balance	
Details												
Linked On	17-05-2020 17:05											
External Linking Number	202005171705abcd											
Description	Sample Subscription											
Payment Method	Wallet Balance											

4.4.2 User Direct Payment Flow

- 1) User chooses to pay with the linked ShopeePay account on the Client's platform, as shown in Figure 4.4.9.
- 2) User may be prompted with ShopeePay PIN verification, as shown in Figure 4.4.10.
- 3) If the payment is successful, User will see a payment success page on the Client's platform, as shown in Figure 4.4.11.

Note: the PIN verification process during Direct Payment is currently not applicable to Thailand and Vietnam.

See screenshots on the following page for the front-end user flow.

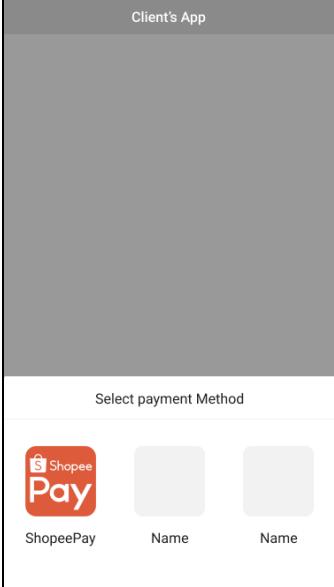
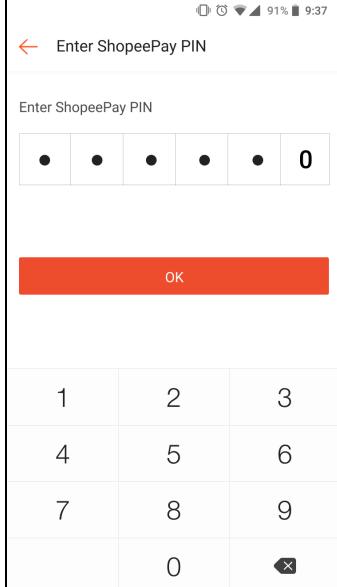
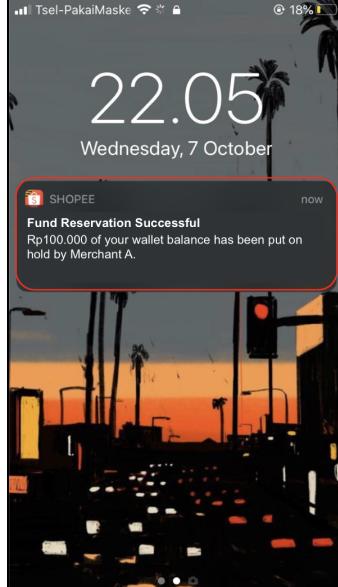
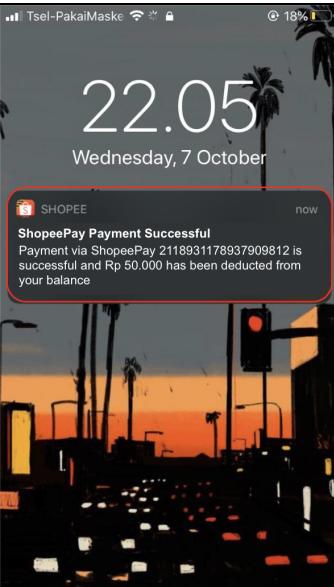
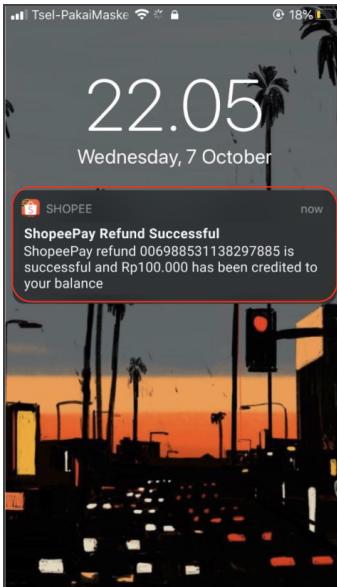
Figure 4.4.9 User pays with linked ShopeePay account	Figure 4.4.10 User may be prompted with ShopeePay PIN re-authentication	Figure 4.4.11 Client's platform payment success page																								
 <p>Client's App</p> <p>Link Payment Method</p> <p>ShopeePay Name Name</p>	 <p>Enter ShopeePay PIN</p> <p>Enter ShopeePay PIN</p> <table border="1"> <tr><td>•</td><td>•</td><td>•</td><td>•</td><td>•</td><td>0</td></tr> <tr><td colspan="6">OK</td></tr> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>4</td><td>5</td><td>6</td></tr> <tr><td>7</td><td>8</td><td>9</td></tr> <tr><td>0</td><td></td><td>✖</td></tr> </table>	•	•	•	•	•	0	OK						1	2	3	4	5	6	7	8	9	0		✖	 <p>Client's App</p> <p>SUCCESS</p> <p>OK</p>
•	•	•	•	•	0																					
OK																										
1	2	3																								
4	5	6																								
7	8	9																								
0		✖																								

4.4.3 User Auth & Capture Flow

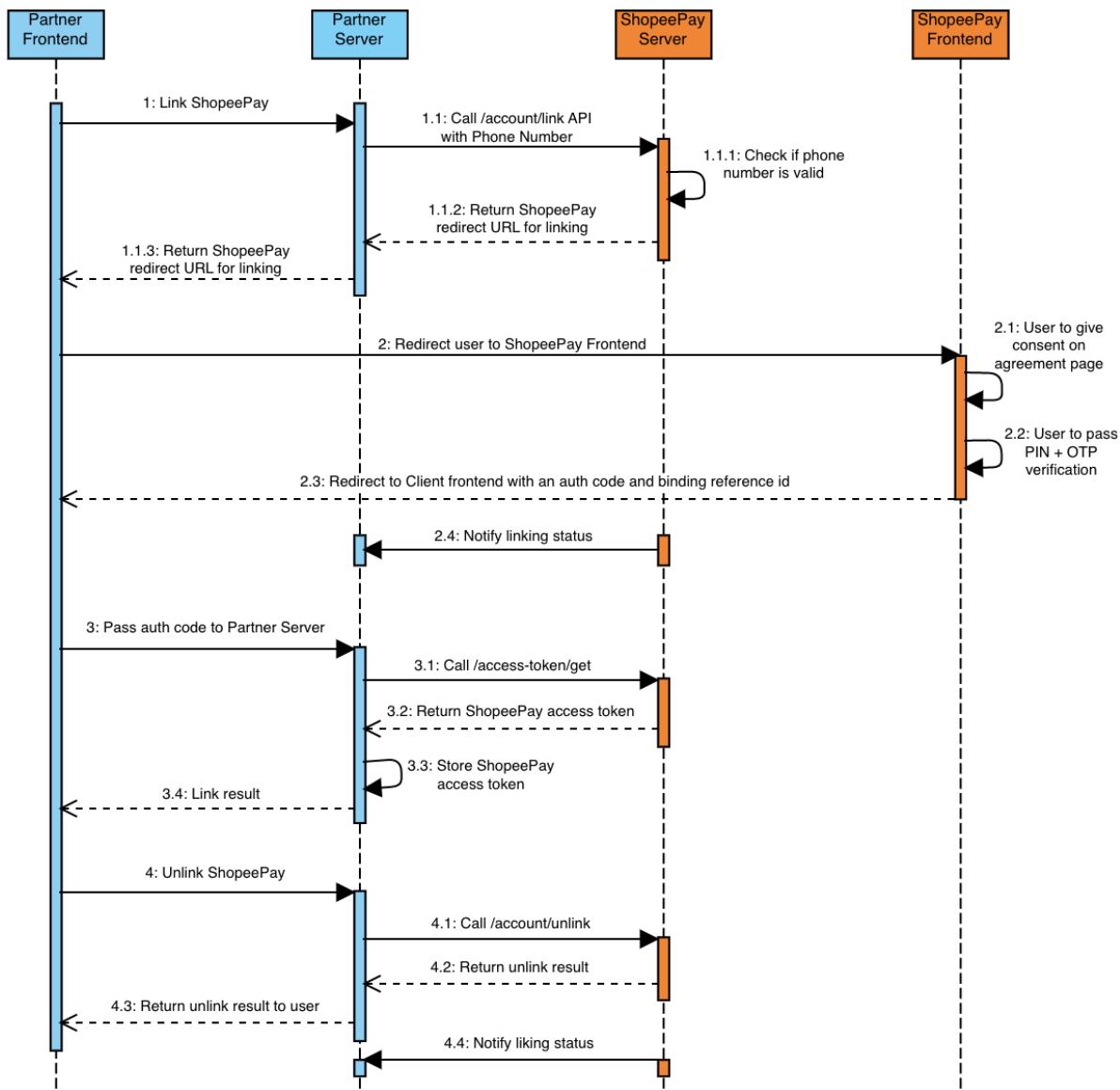
- 1) User pays with the linked ShopeePay account on the Client's platform, as shown in Figure 4.4.12.
- 2) User may be prompted with ShopeePay PIN verification at the auth (fund reservation) stage, as shown in Figure 4.4.13.
- 3) If auth (fund reservation) is successful, User will see a push notification in ShopeePay/Shopee App, as shown in Figure 4.4.14.
- 4) If capture (actual payment) is successful after auth, User will receive another push notification in ShopeePay/Shopee App, as shown in Figure 4.4.15.
- 5) If a refund is initiated and processed, User will receive a push notification in ShopeePay/Shopee App, as shown in Figure 4.4.16.

Note: Auth and Capture flow is temporarily not available in Thailand and Vietnam.

See screenshots on the following page for the front-end user flow.

<p>Figure 4.4.12 User pays with linked ShopeePay</p> 	<p>Figure 4.4.13 User may be prompted with PIN re-authentication</p> 	<p>Figure 4.4.14 Push notification for successful fund reservation</p> 
<p>Figure 4.4.15 Push notification for the final successful payment</p> 	<p>Figure 4.4.16 Push notification for successful refund</p> 	

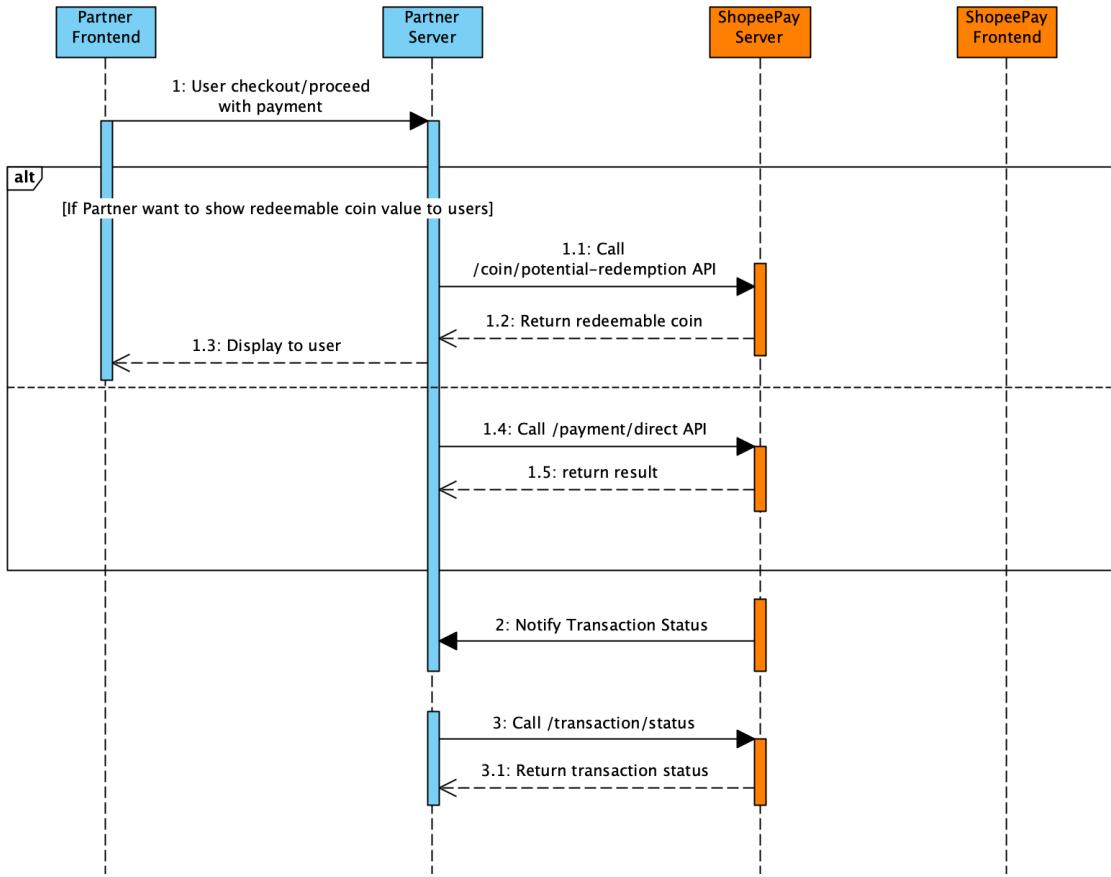
4.4.4 Merchant Link and Unlink Account Flow



- 1) After User chooses to link their ShopeePay account, Client calls the [Account Link API](#) to kick start the linking process.
 - a) If the user's phone number is passed in the request, ShopeePay checks internally if the phone number belongs to a valid ShopeePay user and returns a web URL for Client to redirect the user to.

- 2) Client redirects User to the ShopeePay agreement page. User reads and consents to the permissions required by the Client, and completes both ShopeePay PIN and OTP verification. After successful verification, User is redirected to the Client's page as indicated in the 'return_url' in the Account Link API request along with an auth code.
- 3) Client's frontend passes the auth code to the Client's server and exchanges for a ShopeePay access token by calling the [Get Access Token API](#).
- 4) If User chooses to unlink ShopeePay, Client calls the [Account Unlink API](#) to invalidate the ShopeePay access token.
 - a) If the access token is invalidated successfully, User is not able to access ShopeePay from the Client's platform anymore.
- 5) ShopeePay sends the linking status update via callback URL ([Notify Linking Status](#)) to inform the Client about the linking result.

4.4.5 Merchant Direct Payment Flow



1) During the checkout/payment process on Client's frontend

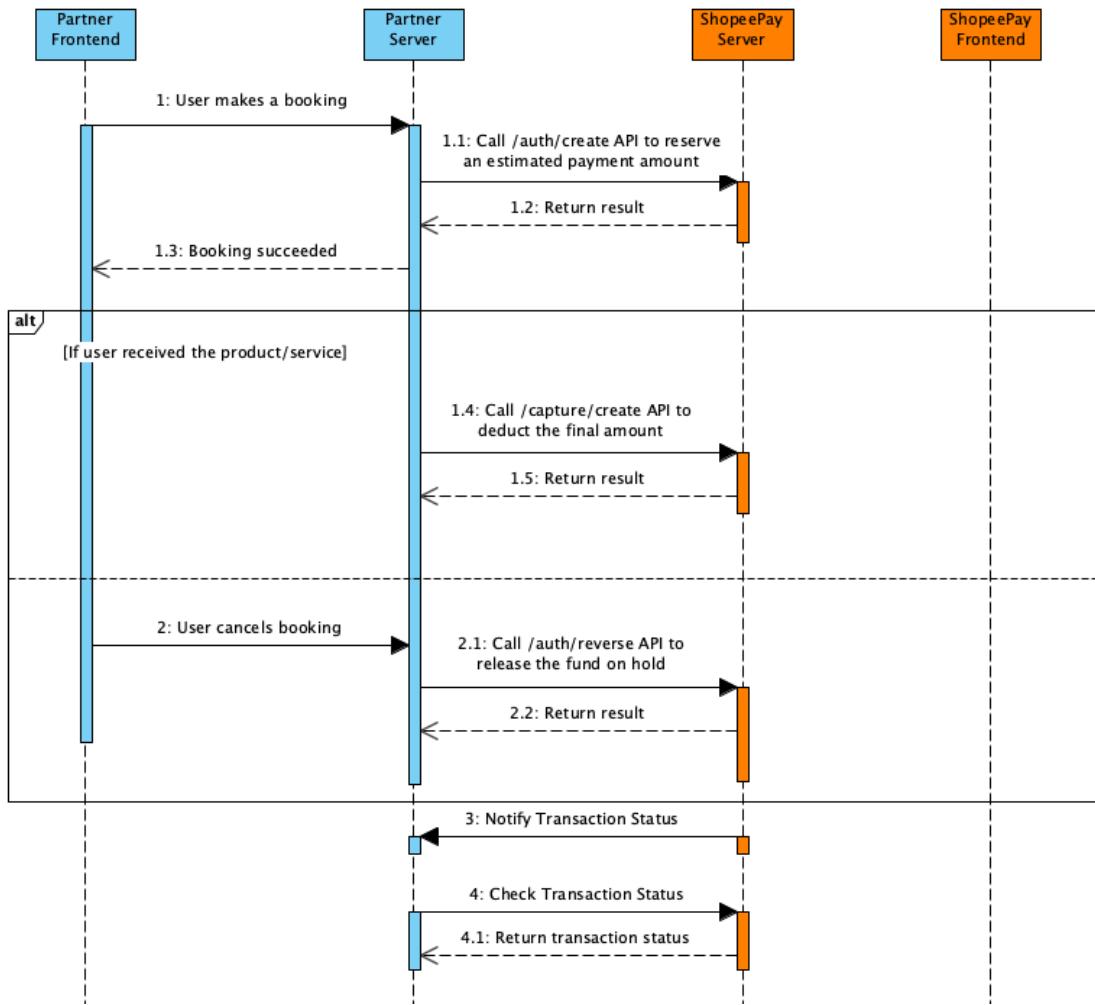
- a) If Client wants to display the redeemable Shopee coins value to the User:
 - i) Client can call [Get Redeemable Coin Value API](#) to get a maximum coin redemption percentage allowed; OR
 - ii) Client can pass “payment_amount” in the request of the Maximum Redeemable Coin API call. The response will indicate the exact number of coins User can spend in this payment, and the exact remaining amount to be paid using ShopeePay.
- b) If Client does not need to display the redeemable Shopee coins value to the User:
 - i) Client can call [Direct Payment API](#) to initiate a direct charge. A redirect url to the ShopeePay PIN verification page may be returned if this verification

is required. (Note: this PIN verification process currently is not applicable to Thailand and Vietnam)

- 2) ShopeePay sends the transaction status update via callback URL ([Notify Transaction Status](#)) to inform the Client about the payment result.
- 3) Client can also call [Check Transaction Status API](#) to check the payment status.

4.4.6 Merchant Auth Capture Flow

Auth capture flow is useful for merchants (e.g. Ride hailing apps) that want to reserve a user's fund first and only deduct either a full or partial amount later.



- 1) User expresses intent to make a purchase (e.g. attempts to book a taxi via a ride hailing app)
 - a) Client calls [Create Auth API](#) to reserve the estimated amount to be charged. After successful fund reservation, Client can inform the user that they can successfully proceed to the next step (e.g. client app informs the user that their booking for the taxi is successful).
 - b) Once the user has received the product/service (e.g. after a taxi ride is complete), Client calls [Capture API](#) to collect the final payment amount (e.g. deduct the final taxi trip fare). Any amount reserved during Auth but not collected in the final

payment capture will be returned to the user after capture is successfully processed.

- 2) If user exits the purchase process (e.g. cancels the taxi booking before the ride begins), Client can call [Reverse Auth API](#) to release the funds reserved during Auth.
- 3) Similar to the direct payment flow, ShopeePay sends the transaction status update via callback URL ([Notify Transaction Status](#)) to inform the client about the payment result.
- 4) Client can also call Check [Transaction Status API](#) to check the payment status.

Note: Auth and Capture flow is temporarily not available in Thailand and Vietnam.

4.4.7 API Body

4.4.7.1 Account Link

Use this API to kick start the account linking process of a ShopeePay account to your platform.

- URL: "/v3/merchant-host/account/link"

Request Parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for each account linking request.
return_url	string	Y	The URL to redirect to after User completes verification on ShopeePay. For a normal web URL which starts with 'http' or 'https', ShopeePay client will open it with the default browser. For others (for example, some_app://xxxxx), ShopeePay client will try to open the app registered with the schema.
linking_reference_id	string	N	Unique identifier in Client's system to identify this linking. Client is strongly encouraged to use linking_reference_id for better tracking and identification purposes.
merchant_ext_id	string	Y	Unique identifier of merchant in Client's system.
service_name	string	N	Name in Client's system to identify this linking service if required.
service_id	string	N	Identifier in Client's system to identify this service associated with this specific linking. Example values are phone number for phone bills payment, or meter number for utility bills, or membership number for recurring membership payment.

linking_description	string	N	General description about this linking, e.g. Payment deduction frequency, item information or user service information etc.
phone	string	N	<p>Registered phone number of the user on Client's platform.</p> <p>This field is required if Client wants to restrict the user to link a ShopeePay account with the same registered phone number.</p> <p>Note: please input a phone number with the country code, e.g., for a VN phone number "0981234567", it shall be passed in as '84981234567' in this field</p>
platform_type	string	N	<p>This indicates the type of the Client's platform that the user triggered the account linking flow from.</p> <p>Accepted values:</p> <ul style="list-style-type: none"> • "app" • "pc" • "mweb" (mobile web) • "qr" (for scan QR linking) <p>Note: "qr" enumeration is currently only available in TH VN.</p>

Response Parameters

Field	Type	Description
request_id	string	The same value as the `request_id` in the request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.
redirect_url_web	string	Returns a URL to ShopeePay's linking agreement page.

		If Client wants to restrict the user to link a ShopeePay account with the same registered phone number, Client must use this redirect URL to reduce cases of account linking failure due to mismatched phone number.
redirect_url_qr	string	Returns a URL to download a QR image for users to use ShopeePay/Shopee App to scan and perform linking on App Note: this field is currently only available in TH, VN
redirect_url_app	string	Returns an URL scheme to ShopeePay's linking agreement page. Sample link: shopeeid://main?apprl=%2Frnl%2FTRANSFER_PAGE%3Fnavigate_url%3Dhttps%253A%252F%2525%252522%25257D%2526ticket%253DASSBSC5AdwNUT

Sample Request

```

1  {
2    ...
3    "request_id": "account_request_2",
4    "return_url": "https://www.isv.com",
5    "linking_reference_id": "account_linking_1",
6    "merchant_ext_id": "externalmerchant",
7    "linking_description": "testing",
8    "phone": "6281234567890",
9    "platform_type": "app"
}

```

Sample Response

```

1  {
2    ...
3    "request_id": "account_request_2",
4    "errcode": 0,
5    "debug_msg": "success",
6    "redirect_url_web": "https://uat.shopee.co.id/payment/account-linking/agreement?client_id=1&return_url=https%3A%2F%2Fwww.
isv.com&state=%7B%22purpose%22%3A%22fCTnHxHxWK%22%7D&ticket=118592511596776763",
7    "redirect_url_app": "shopeeid://main?apprl=%2Frnl%2FTRANSFER_PAGE%3Fnavigate_url%3Dhttps%253A%252F%252Fwsa.uat.wallet.
airpay.co.id%252FWallet%252Faccount_linking%253Fclient_id%253D1%2526return_url%253Dhttps%25253A%25252F%25252Fwww.isv.
com%2526state%253D%25257B%252522purpose%252522%25253A%252522fCTnHxHxWK%252522%25257D%2526ticket%253D118592511596776763
"
}

```

4.4.7.2 Get Access Token

Use this API to get a ShopeePay access token using auth code returned in the account linking step.

- URL: "/v3/merchant-host/access-token/get"

Request Parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for request.
linking_reference_id	string	C	<p>Unique identifier in Client's system to identify this linking, same as what is used in the Account Link API.</p> <p>Either linking_reference_id or auth_code should be provided in the request, but Client is strongly encouraged to use linking_reference_id for better tracking and identification purposes.</p>
auth_code	string	C	<p>The auth code is returned by ShopeePay when the user is redirected to the 'return_url' specified by Client in Account Link API.</p> <p>Client should call Get Access Token as soon as auth code is received. Auth code will expire after 30 mins.</p> <p>Either linking_reference_id or auth_code should be provided in the request.</p> <p>Sample Link: www.isv-integrator.com/?LinkingReferenceld=merasd3mjac&authCode=ATHQ4cJdgpx3Xy&result=100</p> <p>some_app://?LinkingReferencenceld=merasd3mjac&authCode=ATHQ4cJdgpx3Xy&result=100</p> <p>Result codes 100 - User authentication success 202 - Account linking failed</p>

Response Parameters

Field	Type	Description
request_id	string	The same value as the `request_id` in the request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.
access_token	string	The access token used to identify the ShopeePay account to be linked.
user_id_hash	string	Unique identifier for a ShopeePay user.
linking_reference_id	string	Unique identifier in Client's system to identify this linking.
merchant_ext_id	string	Unique identifier of merchant in client system.
linking_status	int32	Status of this linking 1 - Active 2 - Inactive (could be reverted back to Active) 3 - Invalid
create_time	string	Create time of this individual linking.
update_time	string	Update time of this individual linking.

Sample Request with auth_code

```
1  {
2    "request_id": "account_request_2",
3    "auth_code": "40992ac9-1d3a-49f1-8f9d-7db395c0a4de"
4 }
```

Sample Request with linking_reference_id

```
1  {
2    "request_id": "account_request_2",
3    "linking_reference_id": "account_linking_1"
4 }
```

Sample Response

```
1  {
2    "request_id": "account_request_2",
3    "errcode": 0,
4    "debug_msg": "success",
5    "access_token": "GK3FqPzbEuKRmlaH04ay/ifutAQW5aBg",
6    "user_id_hash": "f44eaf4c0ac265f20788ca5e970deea220f73a51612375a3e22056ec9bbb63c",
7    "linking_reference_id": "account_linking_1",
8    "merchant_ext_id": "externalmerchant",
9    "linking_status": 1,
10   "create_time": 1664533075,
11   "update_time": 1664533075
12 }
```

4.4.7.3 Account Unlink

Use this API to unlink a ShopeePay account from a User's account on your platform.

- URL: "/v3/merchant-host/account/unlink"

Request Parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for request.
access_token	string	C	The access token used to identify a ShopeePay user's account.
linking_reference_id	string	C	<p>Unique identifier in Client's system to identify this linking, same as what is used in the Account Link API.</p> <p>Either linking_reference_id or access_token should be provided in the request, but Client is strongly encouraged to use linking_reference_id for better tracking and identification purposes.</p>

Response Parameters

Field	Type	Description
request_id	string	The same value as the `request_id` in the request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.
merchant_ext_id	string	Unique identifier of merchant in client system.
linking_status	int32	<p>Status of this linking</p> <p>1 - Active</p> <p>2 - Inactive (could be reverted back to Active)</p> <p>3 - Invalid</p>
create_time	int32	Create time of this individual unlinking.

update_time	int32	Update time of this individual linking.
access_token	string	The access token used to identify a ShopeePay user's account.
user_id_hash	string	Identifier for user that made the payment.
linking_reference_id	string	Unique identifier in Client's system to identify this linking.

Sample Request with access_token

```

1  {
2    ...
3      "request_id": "the-second-request",
4      "access_token": "GK3FqPzbEuKRmlaH04ay/ifutAQW5aBg"
5  }

```

Sample Request with linking_reference_id

```

1  {
2    ...
3      "request_id": "the-second-request",
4      "linking_reference_id": "account_linking_1"
5  }

```

Sample Response

```

1  {
2    ...
3      "request_id": "the-second-request",
4      "errcode": 0,
5      "debug_msg": "success",
6      "merchant_ext_id": "externalmerchant",
7      "linking_status": 3,
8      "create_time": 1664533075,
9      "update_time": 1664533075,
10     "access_token": "GK3FqPzbEuKRmlaH04ay/ifutAQW5aBg",
11     "user_id_hash": "f44eaf4c0ac265f20788cabef970deea220f73a51612375a3e22056ec9bbb63c",
12     "linking_reference_id": "account_linking_1"
13  }

```

4.4.7.4 Account Link Notify Status

This request will send details of the specific linking lifecycle to the Client server's callback URL. This URL is configured in ShopeePay system at time of onboarding, or sent in API requests.

Each callback request will contain a signature in the request header. This signature is generated using the ShopeePay-issued secret key, assigned to the Client receiving the callback request. The Client should use this signature to verify the authenticity of the callback content.

Multiple repeated notifications might be sent to the Client and Client shall be able to handle the repeated notifications with same contents properly.

[Note] It is recommended for the Client using this Callback to validate that the *linking_reference_id* matches the original account/link request's *linking_reference_id* before considering the user's action to be completed.

Field	Type	Description
request_id	string	The same value as the `request_id` in the request.
linking_reference_id	string	Unique identifier in client's system to identify this linking, same as what is used in the Account Link API. If client did not use linking_reference_id in the Account Link API, this parameter will be auto-generated by ShopeePay system
merchant_ext_id	string	Unique identifier of merchant in client system.
update_type	int64	Update_type of this user's lifecycle: Status of this linking 1 - User successfully authorised the linking 2 - User account is linked successfully 3 - User token status is inactive 4 - User successfully unlinked with Client

Sample Callback for User successfully authorized

```
{  
    "linking_reference_id": "HUA563Qb0rUXSdAMNgEKQntZSk05p3Vq",  
    "merchant_ext_id": "Merchant2812",  
    "request_id": "fcd5944cecf30fdfaf68d4341466e400",  
    "update_type": 1  
}
```

Sample Callback for User successfully unlinked

```
{  
    "linking_reference_id": "HUA563Qb0rUXSdAMNgEKQntZSk05p3Vq",  
    "merchant_ext_id": "Merchant2812",  
    "request_id": "31981d6fecf315421b45eb0e4f5d1100",  
    "update_type": 4  
}
```

4.4.7.5 Get User Info

Use this API to get a user's ShopeePay account information for display purposes.

- URL: "/v3/merchant-host/user-info/get"

Request Parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for request.
access_token	string	C	The access token used to identify the user account. Either linking_reference_id or access_token should be provided in the request.
linking_reference_id	string	C	Unique identifier in Client's system to identify this linking, same as what is used in the Account Link API. Either linking_reference_id or access_token should be provided in the request, but Client is strongly encouraged to use linking_reference_id for better tracking and identification purposes.

Response Parameters

Field	Type	Description
request_id	string	The same value as the `request_id` in the request.
errcode	int32	Error code to specify the error returned. <ul style="list-style-type: none"> User_info will not be returned if user's account status is deleted or banned. User_info will be returned per normal if user's account status is frozen.
debug_msg	string	Debug message to provide more information.

linking_info	object	Linking information. Will only be returned if there is an linking found with the linking_reference_id or access_token provided.
• linking_reference_id	string	Unique identifier in Client's system to identify this linking.
• linking_status	int32	Status of this linking 1 - Active 2 - Inactive (could be reverted back to Active) 3 - Invalid
• create_time	string	Create time of this individual linking.
• update_time	string	Update time of this individual linking.
• merchant_ext_id	string	Unique identifier of merchant in client system.
• user_id_hash	string	Identifier for the user that made the payment.
user_info	object	Will return empty if the ShopeePay account is banned or deleted.
• phone_number	string	Masked phone number of the user's linked ShopeePay account. Example: *****1234 This value is returned if permission is allowed via client's contract with ShopeePay.
• wallet_balance	int64	Value inflated by 100. This value is returned if permission is allowed via client's contract with ShopeePay.
• coin_balance	int64	Value inflated by 100. Only available if user allow coin redemption for their account.
• kyc_passed	boolean	True - User passed KYC False - User has not passed KYC yet This value is returned if permission is allowed via client's contract with ShopeePay.

Sample Request with access_token

```
1  {
2    ... "request_id": "user-info-get-request",
3    ... "access_token": "GK3FqPzbEuKRmlaH04ay/ifutAQW5aBg"
4 }
```

Sample Request with linking_reference_id

```
1  {
2    ... "request_id": "user-info-get-request",
3    ... "access_token": "account_linking_1"
4 }
```

Sample Response

```
1  {
2    ... "request_id": "user-info-get-request",
3    ... "errcode": 0,
4    ... "debug_msg": "success",
5    ... "user_info": {
6      ... "phone_number": "6281xxxxxxxxx",
7      ... "wallet_balance": 94027400,
8      ... "coin_balance": 28937400,
9      ... "kyc_passed": true
10     },
11    ... "linking_info": {
12      ... "linking_status": 1,
13      ... "linking_reference_id": "account_linking_1",
14      ... "create_time": 1661781445,
15      ... "update_time": 1661781445,
16      ... "merchant_ext_id": "externalmerchant",
17      ... "user_id_hash": "f44eaf4c0ac265f2707088ca5970deea220f73a51612375a3e22056ec9bbb63c"
18    }
19 }
```

4.4.7.6 Get Redeemable Coin Value

Use this API to get the user's maximum redeemable coin amount in percentage or in absolute amount.

- URL: "/v3/merchant-host/coin/potential-redemption"

Note: This API is temporarily unavailable for use in Thailand and Vietnam.

Request Parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for request.
access_token	string	Y	The access token used to identify the user account.
payment_amount	int64	N	Payment amount value inflated by 100. Pass this field if Client wants to display the exact amount of redeemable coins for a specific payment.

Response Parameters

Field	Type	Description
request_id	string	The same value as the `request_id` in the request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.
max_coin_percentage	int64	Maximum percentage of the payment can be paid for using coins.
potential_payment_amount	int64	Estimated payment amount to be paid with ShopeePay after coin redemption. Value inflated by 100. This value is returned if "payment_amount" is sent in the request.
potential_coin_amount	int64	Maximum numbers of coins can be used in the payment. Value inflated by 100. This value is returned if "payment_amount" is sent in the request.

Sample Request

```
1  {
2    "request_id": "coin-get-request",
3    "access_token": "GK3FqPzbEuKRmlaH04ay/ifutAQW5aBg",
4    "payment_amount": 10000
5 }
```

Sample Response

```
1  {
2    "request_id": "coin-get-request",
3    "errcode": 0,
4    "debug_msg": "success",
5    "max_coin_percentage": 10,
6    "potential_payment_amount": 9000,
7    "potential_coin_amount": 1000
8 }
```

4.4.7.7 Direct Payment

Use this API to initiate a direct payment with an access token.

- URL: "/v3/merchant-host/transaction/payment/direct"

Request Parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for request.
payment_reference_id	string	Y	Unique identifier of payment transaction generated by client.
merchant_ext_id	string	Y	Unique identifier of merchant in client system.
store_ext_id	string	Y	Unique identifier of store in client system.
access_token	string	Y	The access token used to identify the user's ShopeePay account.
amount	int64	Y	Amount used for the payment, refer to Parameters Specifications for specifics.
currency	string	Y	Refers to the currency abbreviation for the transaction. <ul style="list-style-type: none"> • MYR • IDR • PHP • SGD • THB • VND
return_url	string	C	Indicates the URL of the Client's platform to redirect back to, once payment verification on ShopeePay (if applicable) is done.
use_coin	boolean	N	Users may indicate their preference to use or not to use coins to offset their payment on the Client's platform. Use this field to pass the user's preference.

			<ul style="list-style-type: none"> • True: attempt to use coins • False: do not use coins <p>Note: This field is temporarily unavailable in Thailand and Vietnam. Reserved for future usage.</p>
promo_ids	string	N	<p>Comma separated eligible promo_ids of 100 characters or less (up to 20), if any.</p> <p>These promo_ids will be checked against promotional campaigns configured in ShopeePay to determine the valid reward amount the user can receive for this transaction.</p>
additional_info	string	N	<p>Additional information will be in json format.</p> <p>This field supports up to 3 fields and needs to be sent in this format {"field1": "Data1"}, {"field2": "Data2"}, {"field3": "Data3"} Note: The field's format only applicable in ID, TH, VN.</p>

Response Parameters

Field	Type	Description
request_id	string	The same value as the `request_id` in the request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.
redirect_url	string	The URL for the Client's front-end to redirect to for PIN verification. Only applicable if PIN verification is needed.
transaction	object	Will return Nil if transaction failed.
• reference_id	string	Unique identifier of payment transaction generated by Client.

		Same value as the “payment_reference_id” in the request.
• amount	int64	Amount used for the payment, refer to Parameters Specifications for specifics.
• status	uint32	Refer to Appendix 1C - Transaction Status for specific transaction statuses.
• transaction_type	uint32	13 (Refer to Appendix 1A - Transaction Types for specific transaction types).
• transaction_sn	string	Transaction identifier in ShopeePay system.
• create_time	uint32	Create time of the individual transaction.
• update_time	uint32	Update time of the individual transaction.
• user_id_hash	string	Identifier for user that made the payment.
• merchant_ext_id	string	Unique identifier of merchant in Client system.
• store_ext_id	string	Unique identifier of store in Client system.
• promo_id_applied	string	promo_id applied to the payment transaction. This will be filled only if the user received a reward from a valid promotion with the promo_id specified.
• payment_channel	uint32	Indicates the source of fund used. Refer to Appendix 1E - Payment Channels for the respective source of fund.

Sample Request

```

1  {
2    "request_id": "request_direct_payment_1",
3    "payment_reference_id": "direct_payment_testing1",
4    "merchant_ext_id": "externalmerchant",
5    "store_ext_id": "externalstore",
6    "access_token": "GK3FqPzbEuKRmlaH04ay/ifutAQWSaBg",
7    "amount": 100,
8    "currency": "IDR",
9    "return_url": "https://shopee.co.id",
10   "use_coin": true,
11   "promo_ids": "1,2,3",
12   "additional_info": {"field1": "Name", "field2": "Customer ID", "field3": "Phone Number"}
13 }

```

Sample Response 1 (when ShopeePay PIN verification is not required)

```

1  {
2    "request_id": "request_direct_payment_1",
3    "errcode": 0,
4    "debug_msg": "success",
5    "transaction": {
6      "reference_id": "direct_payment_testing_1",
7      "amount": 100,
8      "create_time": 1661778527,
9      "update_time": 1661778527,
10     "transaction_sn": "150565663100918374",
11     "status": 3,
12     "transaction_type": 13,
13     "merchant_ext_id": "externalmerchant",
14     "user_id_hash": "e1030777a3868f152526ebc19bd899d0a68aabb087a38a759b1bb9577d04483b",
15     "store_ext_id": "externalstore",
16     "promo_id_applied": "1,2,3",
17     "payment_channel": 1
18   }
19 }

```

Sample Response 2 (when ShopeePay PIN verification is required)

```

1  {
2    "request_id": "request_direct_payment_1",
3    "errcode": 0,
4    "debug_msg": "success",
5    "redirect_url": "https://uat.shopee.co.id/s/browser/payment/auth/passcode-verify?mode=fullscreen&next=https%3A%2F%2Fuat.shopee.co.id%2Fbrowser%2Fpayment%2Fauth%2FtokenizedPaymentResult%3Freturn_url%3Dhttps%253A%252F%252Fwww.isv.com%252Fpayment%26ticket%3DiStkwEbZpAzLsHgnZeD7pa6kfmrqzqqb&return_url=https%3A%2F%2Fuat.shopee.co.id%2Fbrowser%2Fpayment%2Fauth%2FtokenizedPaymentResult%3Fresult%3D201%26return_url%3Dhttps%253A%252F%252Fwww.isv.com%252Fpayment%26ticket%3DiStkwEbZpAzLsHgnZeD7pa6kfmrqzqqb&scenario=7cdccb66-cd7a-448f-a01f-b2ddd8970e53&source=TokenizedPayment&ticket=iStkwEbZpAzLsHgnZeD7pa6kfmrqzqqb"
6  }

```

4.4.7.8 Create Auth

Use this API to create an auth transaction to reserve funds for subsequent capture.

- URL: "/v3/merchant-host/transaction/auth/create"

Note: This API is temporarily unavailable for use in Thailand and Vietnam.

Request Parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for request.
auth_reference_id	string	Y	Unique identifier of authorisation transaction generated by client. Accepts up to 64 characters.
merchant_ext_id	string	Y	Unique identifier of merchant in client system.
store_ext_id	string	Y	Unique identifier of store in client system.
access_token	string	Y	The access token used to identify the user account.
amount	int64	Y	Amount to be reserved during auth, refer to Parameters Specifications for specifics.
currency	string	Y	Refers to the currency abbreviation for the transaction <ul style="list-style-type: none"> • MYR • IDR • PHP • SGD • THB • VND
return_url	string	C	Indicates the URL of the Client's platform to redirect back to once payment verification on ShopeePay (if applicable) is done.
auth_expiry_time	uint32	N	Unix timestamp. The value represents when this auth transaction will expire.

			<p>Maximum expiry time is 14 days from the time the request is received by ShopeePay.</p> <p>If no value is passed, the expiry time of an auth is by default 24 hours from the time the request is received by ShopeePay.</p>
--	--	--	---

Response Parameters

Field	Type	Description
request_id	string	The same value as the `request_id` in the request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.
redirect_url	string	The URL for the Client's frontend to redirect to for PIN verification, valid for 30 mins. Only applicable if PIN verification is needed.
transaction	object	Will return empty if no such transaction.
• reference_id	string	Unique identifier of auth transaction generated by client. Same value as the "auth_reference_id" in the request.
• amount	int64	Amount used for the payment, refer to Parameters Specifications for specifics.
• status	uint32	Refer to Appendix 1C - Transaction Status for specific transaction statuses.
• transaction_type	uint32	1000 (Refer to Appendix 1A - Transaction Types for specific transaction types).
• transaction_sn	string	Transaction identifier in ShopeePay system.
• create_time	uint32	Create time of the individual transaction.
• update_time	uint32	Update time of the individual transaction.

● user_id_hash	string	Identifier for user that made the payment.
● merchant_ext_id	string	Unique identifier of merchant in Client system.
● store_ext_id	string	Unique identifier of store in Client system.
● payment_channel	uint32	Indicates the source of fund used. Refer to Appendix 1E - Payment Channels for the respective source of fund.

Sample Request

```

1  {
2    "request_id": "request_test_auth_1",
3    "auth_reference_id": "isv_auth_reference_id_1",
4    "merchant_ext_id": "externalmerchant",
5    "store_ext_id": "externalstore",
6    "access_token": "GK3FqPzbEuKRmlaH04ay/i futAQWSaBg",
7    "amount": 10000,
8    "currency": "IDR",
9    "return_url": "https://shopee.co.id",
10   "auth_expiry_time": 1681453891
11 }
```

Sample Response 1 (when ShopeePay PIN verification is not required)

```

1  {
2    "request_id": "request_test_auth_1",
3    "errcode": 0,
4    "debug_msg": "success",
5    "redirect_url": "",
6    "transaction": {
7      "reference_id": "isv_auth_reference_id_1",
8      "amount": 10000,
9      "create_time": 1664693752,
10     "update_time": 1664693752,
11     "transaction_sn": "123991230867261786",
12     "status": 3,
13     "transaction_type": 1000,
14     "merchant_ext_id": "externalmerchant",
15     "user_id_hash": "e1030777a3868f152526ebc19bd899d0a68aabb087a38a759b1bb9577d04483b",
16     "store_ext_id": "externalstore",
17     "payment_channel": 1
18   }
19 }
```

Sample Response 2 (when ShopeePay PIN verification is required)

```
1  {
2    "request_id": "request_test_auth_1",
3    "errcode": 0,
4    "debug_msg": "success",
5    "redirect_url": "https://uat.shopee.co.id/s/browser/payment/auth/passcode-verify?mode=fullscreen&
next=https%3A%2F%2Fuat.shopee.co.
id%2Fs%2Fbrowser%2Fpayment%2Fauth%2FtokenizedPaymentResult%3Freturn_url%3Dhttps%253A%252F%252Fwww
.isv.com%252Fpayment%26ticket%3DiStkwEbZpAzLsHgnZeD7pa6kfmrqzqqb&return_url=https%3A%2F%2Fuat.
shopee.co.
id%2Fs%2Fbrowser%2Fpayment%2Fauth%2FtokenizedPaymentResult%3Fresult%3D201%26return_url%3Dhttps%25
3A%252F%252Fwww.isv.com%252Fpayment%26ticket%3DiStkwEbZpAzLsHgnZeD7pa6kfmrqzqqb&
scenario=7cdccb66-cd7a-448f-a01f-b2ddd8970e53&source=TokenizedPayment&
ticket=iStkwEbZpAzLsHgnZeD7pa6kfmrqzqqb"
6  }
```

4.4.7.9 Reverse Auth

Use this API to reverse funds reserved previously using an auth request.

- URL: "/v3/merchant-host/transaction/auth/reverse"

Note: This API is temporarily unavailable for use in Thailand and Vietnam.

Request Parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for request.
reverse_auth_reference_id	string	Y	Unique identifier of authorisation reversal transaction generated by client. Accepts up to 64 characters.
merchant_ext_id	string	Y	Unique identifier of merchant in client system.
store_ext_id	string	Y	Unique identifier of store in client system.
auth_reference_id	string	Y	The 'reference_id' of the auth transaction Client wants to reverse.

Response Parameters

Field	Type	Description
request_id	string	The same value as the `request_id` in the request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.
transaction	object	Will return empty if no such transaction
• reference_id	string	Unique identifier of auth reversal transaction generated by client. Same value as the "reverse_auth_reference_id" in the request.

● amount	int64	Amount used for the payment, refer to Parameters Specifications for specifics.
● status	uint32	Refer to Appendix 1C - Transaction Status for specific transaction statuses.
● transaction_type	uint32	1002 (Refer to Appendix 1A - Transaction Types for specific transaction types).
● transaction_sn	string	Transaction identifier in ShopeePay system.
● create_time	uint32	Create time of the individual transaction.
● update_time	uint32	Update time of the individual transaction.
● user_id_hash	string	Identifier for user that made the payment.
● merchant_ext_id	string	Unique identifier of merchant in Client system.
● store_ext_id	string	Unique identifier of store in Client system.
● payment_channel	uint32	Indicates the source of fund used. Refer to Appendix 1E - Payment Channels for the respective source of fund.

Sample Request

```
1  {
2    "request_id": "request_test_reverse_auth_1",
3    "reverse_auth_reference_id": "isv_reverse_auth_reference_id_1",
4    "merchant_ext_id": "externalmerchant",
5    "store_ext_id": "externalstore",
6    "auth_reference_id": "isv_auth_reference_id_1"
7 }
```

Sample Response

```
1  {
2    "request_id": "request_test_reverse_auth_1",
3    "errcode": 0,
4    "debug_msg": "success",
5    "transaction": {
6      "reference_id": "isv_reverse_auth_reference_id_1",
7      "amount": 10000,
8      "create_time": 1664693755,
9      "update_time": 1664693755,
10     "transaction_sn": "123991230867261788",
11     "status": 3,
12     "transaction_type": 1002,
13     "merchant_ext_id": "externalmerchant",
14     "user_id_hash": "e103077a3868f152526ebc19bd899d0a68aabb087a38a759b1bb9577d04483b",
15     "store_ext_id": "externalstore",
16     "payment_channel": 1
17   }
18 }
```

4.4.7.10 Capture

Use this API to create a full or partial capture transaction using funds reserved from the corresponding auth transaction. Capture must be requested before the auth transaction expires. Only one partial capture will be accepted for each auth transaction.

- URL: "/v3/merchant-host/transaction/auth/capture"

Note: This API is temporarily unavailable for use in Thailand and Vietnam.

Request Parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for request.
capture_reference_id	string	Y	Unique identifier of capture transaction generated by client. Accepts up to 64 characters.
merchant_ext_id	string	Y	Unique identifier of merchant in client system.
store_ext_id	string	Y	Unique identifier of store in client system.
amount	int64	Y	The amount to capture. Capture amount should be no greater than the original auth amount.
auth_reference_id	string	Y	The 'auth_reference_id' of the auth transaction Client wants to capture.
use_coin	boolean	N	Users may indicate their preference to use or not to use coins to offset their payment on the Client's platform. Use this field to pass the user's preference. <ul style="list-style-type: none"> • True: attempt to use coins • False: do not use coins
promo_ids	string	N	Comma separated eligible promo_ids of 100 characters or less (up to 20), if any. These promo_ids will be checked against promotional campaigns

			configured in ShopeePay to determine the valid reward amount the user can receive for this transaction.
additional_info	string	N	<p>Additional information will be in json format.</p> <p>This field supports up to 3 fields and needs to be sent in this format {"field1":"{Data1}","field2":"{Data2}" , "field3":"{Data3}"}</p> <p>Note: The field's format only applicable in ID, TH, & VN.</p>

Response Parameters

Field	Type	Description
request_id	string	The same value as the `request_id` in the request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.
transaction	object	Will return empty if no such transaction.
• reference_id	string	<p>Unique identifier of capture transaction generated by client.</p> <p>Same value as the `capture_reference_id` in the request.</p>
• amount	int64	Amount used for the payment, refer to Parameters Specifications for specifics.
• status	uint32	Refer to Appendix 1C - Transaction Status for specific transaction statuses.
• transaction_type	uint32	1001 (Refer to Appendix 1A - Transaction Types for specific transaction types).
• transaction_sn	string	Transaction identifier in ShopeePay system.
• create_time	uint32	Create time of the individual transaction.

• update_time	uint32	Update time of the individual transaction.
• user_id_hash	string	Identifier for user that made the payment.
• merchant_ext_id	string	Unique identifier of merchant in Client system.
• store_ext_id	string	Unique identifier of store in Client system.
• promo_id_applied	string	promo_id applied to the payment transaction. This will be filled only if the user received a reward from a valid promotion with the promo_id specified.
• payment_channel	uint32	Indicates the source of fund used. Refer to Appendix 1E - Payment Channels for the respective source of fund.

Sample Request

```

1  {
2    "request_id": "request_test_capture_auth_1",
3    "capture_reference_id": "isv_capture_auth_reference_id_1",
4    "merchant_ext_id": "externalmerchant",
5    "store_ext_id": "externalstore",
6    "amount": 10000,
7    "auth_reference_id": "isv_auth_reference_id_1",
8    "use_coin": false,
9    "promo_ids": "1,2,3",
10   "additional_info": "{\"field1\": \"Name\", \"field2\": \"Customer ID\", \"field3\": \"Phone Number\"}"
11 }
```

Sample Response

```
1  {
2      "request_id": "request_test_capture_auth_1",
3      "errcode": 0,
4      "debug_msg": "success",
5      "transaction": {
6          "reference_id": "isv_capture_auth_reference_id_1",
7          "amount": 10000,
8          "create_time": 1664693755,
9          "update_time": 1664693755,
10         "transaction_sn": "123991230867261788",
11         "status": 3,
12         "transaction_type": 1001,
13         "merchant_ext_id": "externalmerchant",
14         "user_id_hash": "e103077a3868f152526ebc19bd899d0a68aabb087a38a759b1bb9577d04483b",
15         "store_ext_id": "externalstore",
16         "promo_id_applied": "1,2,3",
17         "payment_channel": 1
18     }
19 }
```

4.4.7.11 Notify Transaction Status

Please refer to [Notify Transaction Status](#)

4.4.7.12 Create Refund API [New]

Please refer to [Create Refund API \[New\]](#)

4.4.7.13 Check Transaction Status API

Please refer to [Check Transaction Status API](#)

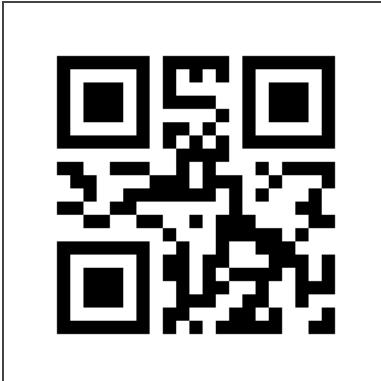
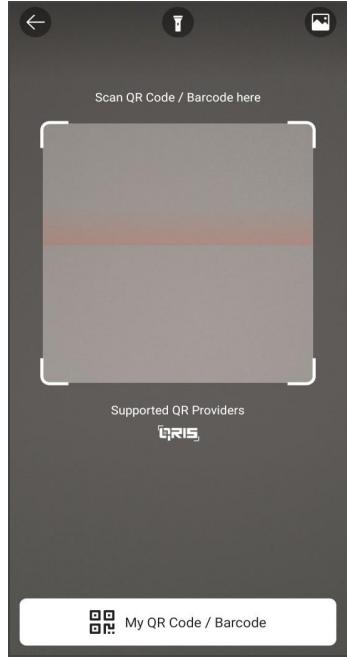
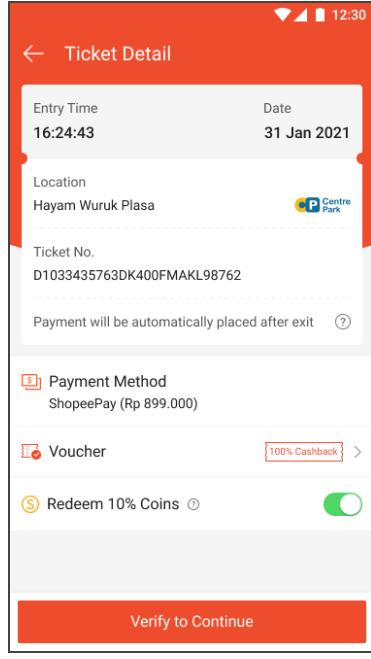
4.5. Ticket Linking

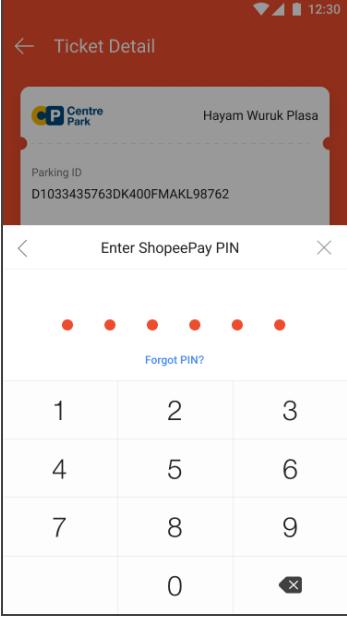
4.5.1 User Ticket Linking and Payment Flow

Ticket Linking has a similar concept to CPM flow.

- 1) User obtains the physical ticket (containing QR or Barcode) from the client.
- 2) User clicks on “Scan” icon on Shopee/ShopeePay homepage and scan the QR code, as shown in figure 4.5.2.
- 3) If the ticket is valid, User can verify the ticket details and select vouchers and coins redemption (if any), as shown in figure 4.5.3.
- 4) User inputs ShopeePay PIN to confirm the ticket linking, as shown in figure 4.5.4.
- 5) If the linking is successful, User will see the successful linking result, as shown in figure 4.5.5.
- 6) User passes the ticket to the cashier when requested, and the ticket will be charged by the cashier to User’s ShopeePay account.
- 7) If payment is successful, User will be able to check their transaction history and see the successful payment screen, as shown in figure 4.5.6.

See screenshots below for the front-end user flow on ShopeePay

<p>Figure 4.5.1 User receives the Partner generated ticket</p>	<p>Figure 4.5.2 User proceeds to scan the ticket</p>	<p>Figure 4.5.3 User will see the ticket detail page</p>
		
<p>Figure 4.5.4 PIN Verification for the User after confirming ticket details.</p>	<p>Figure 4.5.5 Successful linking page</p>	<p>Figure 4.5.6 Successful deduction of ticket payment.</p>



Ticket Detail

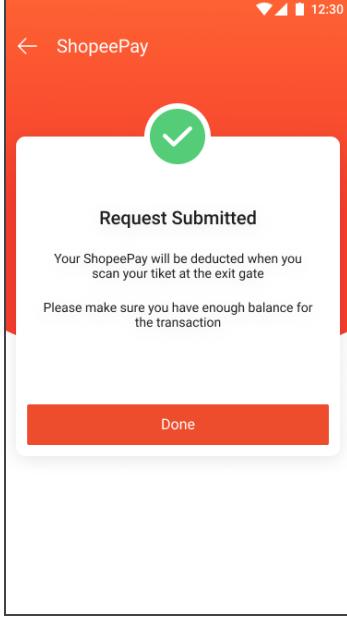
Centre Park Hayam Wuruk Plaza

Parking ID
D1033435763DK400FMAKL98762

Enter ShopeePay PIN

Forgot PIN?

1	2	3
4	5	6
7	8	9
0		✖



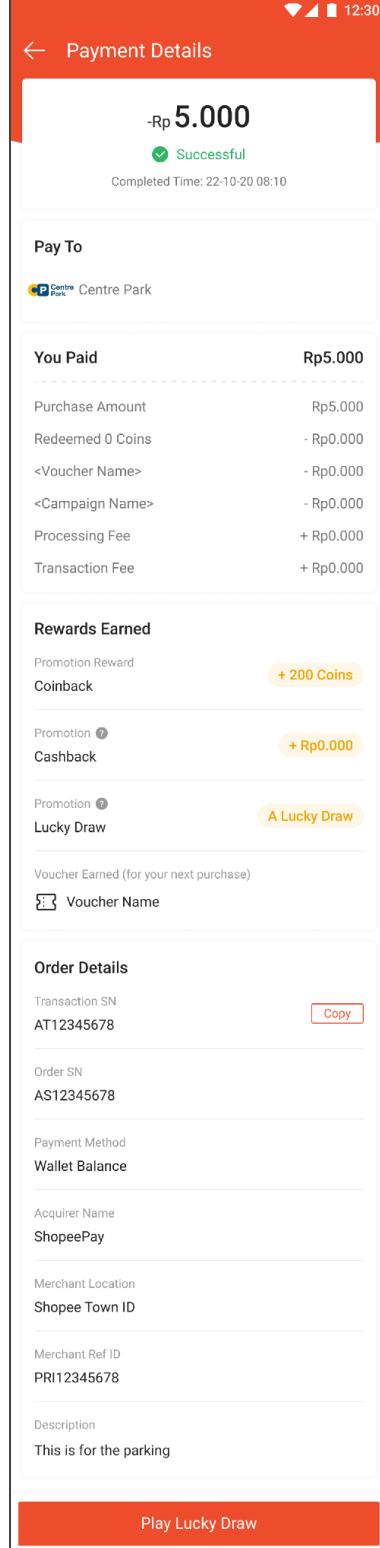
ShopeePay

Request Submitted

Your ShopeePay will be deducted when you scan your ticket at the exit gate

Please make sure you have enough balance for the transaction

Done



Payment Details

-Rp 5.000
Successful
Completed Time: 22-10-20 08:10

Pay To
Centre Park

You Paid	Rp5.000
Purchase Amount	Rp5.000
Redeemed 0 Coins	- Rp.000
<Voucher Name>	- Rp.000
<Campaign Name>	- Rp.000
Processing Fee	+ Rp.000
Transaction Fee	+ Rp.000

Rewards Earned

- Promotion Reward
Coinback + 200 Coins
- Promotion Cashback + Rp.000
- Promotion Lucky Draw A Lucky Draw

Voucher Earned (for your next purchase)
Voucher Name

Order Details

Transaction SN
AT12345678 Copy

Order SN
AS12345678

Payment Method
Wallet Balance

Acquirer Name
ShopeePay

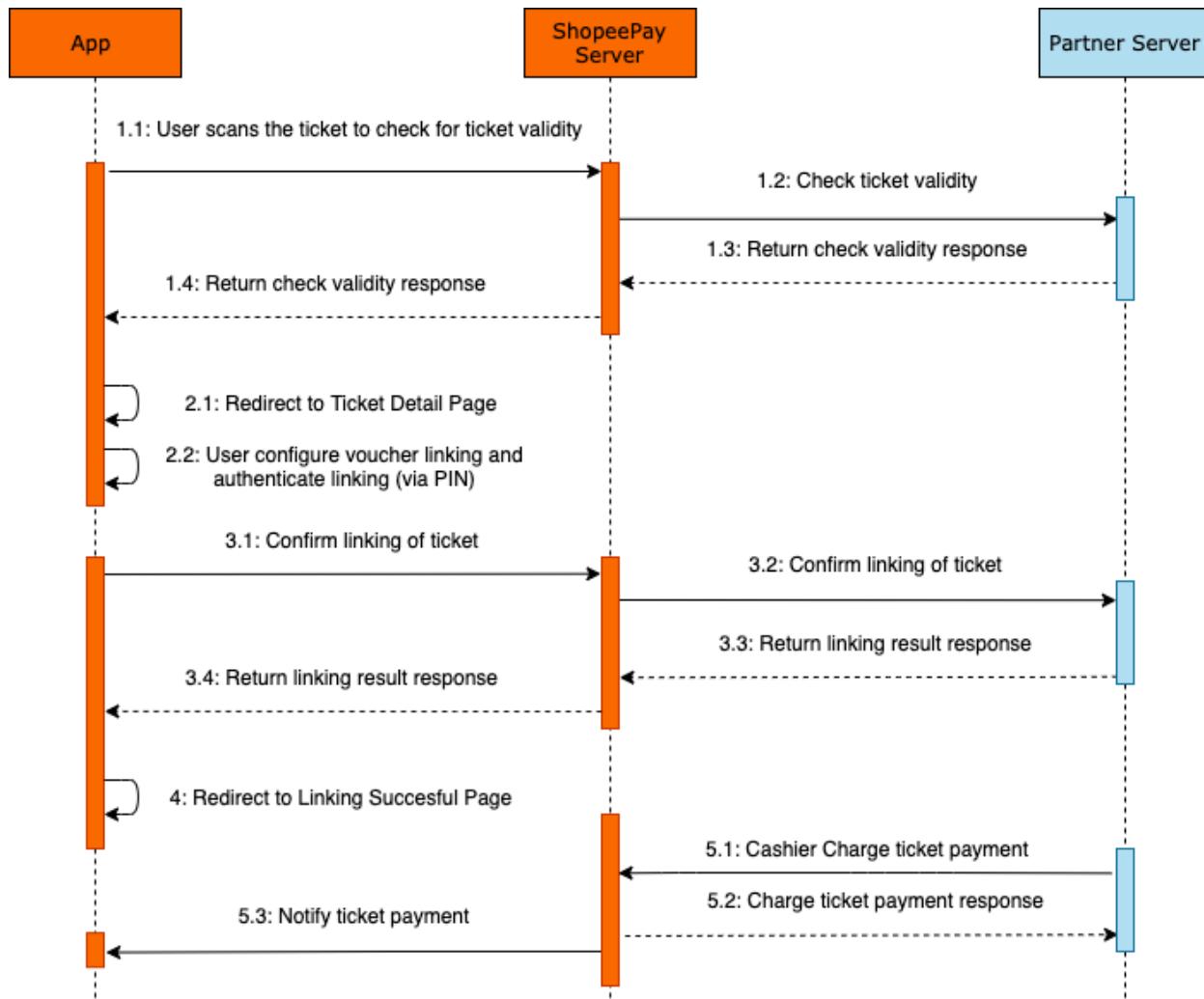
Merchant Location
Shopee Town ID

Merchant Ref ID
PRI12345678

Description
This is for the parking

Play Lucky Draw

4.5.2 Merchant Process



- 1) Client generates the ticket based on pre-aligned protocols with ShopeePay.
- 2) User will scan the ticket generated by Client, and ShopeePay will call Client's endpoint to validate the Ticket.
- 3) User will be redirected to the ticket detail page and configure the ticket linking.
- 4) After User confirms their configurations, ShopeePay will require User to verify ShopeePay PIN first and call Client to confirm this linking action.
- 5) After Client responds 'Success' to ShopeePay, Shopeepay will redirect to the linking successful page.

- 6) The cashier will request for the physical ticket from the User to process the payment, and the Client will call the ticket charge API when the cashier receives the ticket to do payment deduction from user's ShopeePay account.
- 7) If there's no response from ShopeePay Server before request timeout (time out duration is determined by Client), Client can check transaction status by calling [Check Transaction Status API](#).
- 8) If [Check Transaction Status API](#) returns:
 - a) Success - Payment has succeeded
 - b) Not found - Retry the payment process from the start
 - c) No response - continue retrying
 - i) Retry should be done at an incremental time range, at our recommended retry timing at the following time mark:
 - 5 seconds
 - 10 seconds
 - 15 seconds
 - 20 seconds
 -
 -
 -
 - 100 seconds
- 9) After 100 seconds, payment transaction will timeout (ShopeePay will issue a timeout on that request) and the merchant should refund the original payment transaction before attempting payment creation again.
- 10) Calling [Create Refund API](#) will return 2 status types:
 - a) Success - transaction successfully refunded, user can try paying again.
 - b) Fail - transaction can be found but it does not meet the requirements for refund, merchant can try again at a later time.
 - c) If the payment transaction is not found, or payment is still processing, calling [Create Refund API](#) will return an error code (see [Appendix 5 - Error Codes](#)) to indicate failure.

4.5.3 API Body

4.5.3.1 Ticket Charge API

Use this API to initiate a payment request after scanning the User's ticket.

- URL: "/v3/merchant-host/transaction/ticket/charge"

Note: this API is currently only available in Indonesia.

Request Parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for request.
amount	int64	Y	Amount used for the payment, refer to Parameters Specifications for specifics.
merchant_ext_id	string	Y	Unique identifier of merchant in client system.
store_ext_id	string	Y	Unique identifier of store in client system.
payment_code	string	Y	Unique identifier of the linking for the specific user and ticket. (refNumber).
ticket	string	Y	The ticket code generated by the client.
payment_reference_id	string	Y	Unique identifier of payment transaction generated by client.
expiry_time	uint32	N	<p>Unix timestamp.</p> <p>If this field is filled, the order with the corresponding payment_reference_id in this request will fail after the specified timestamp.</p> <p>If this field is not filled, default ShopeePay expiry time of 120 seconds (2 min) will apply for this</p>

			payment method, from the time the request was received.
currency	string	Y	<p>Refers to the currency abbreviation for the transaction</p> <ul style="list-style-type: none"> • MYR • IDR • PHP • SGD • THB • VND
promo_ids	string	N	<p>Comma separated eligible promo_ids of 100 characters or less (up to 20), if any.</p> <p>These promo_ids will be checked against promotional campaigns configured in ShopeePay to determine the valid reward amount the user can receive for this transaction.</p>
additional_info	string	N	<p>Additional information will be in json format.</p> <p>This field supports up to 3 fields and needs to be sent in this format {"field1": "{Data1}" , "field2": "{Data2}" , "field3": "{Data3}"}</p>

Response Parameters

Field	Type	Description
request_id	string	Unique identifier for request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.
transaction	object	Will return Nil if transaction fails.
• reference_id	string	Unique identifier of payment transaction generated by client.

• amount	int64	Payment amount - inflated by a factor of 100.
• create_time	uint32	Create time of the individual transaction.
• update_time	uint32	Update time of the individual transaction.
• transaction_sn	string	Transaction identifier in ShopeePay system.
• status	uint32	Refer to Appendix 1C - Transaction Status for specific transaction statuses.
• transaction_type	uint32	13 (Refer to Appendix 1A - Transaction Types for specific transaction types).
• merchant_ext_id	string	Unique identifier of merchant in client system.
• terminal_id	string	Terminal ID refers to the Point of Sale terminal in a store.
• user_id_hash	string	Used to identify the unique user.
• store_ext_id	string	Unique identifier of store in client system.
• payment_channel	uint32	Indicates the source of fund used. Refer to Appendix 1E - Payment Channels for the respective source of fund.

Sample Request

```
1  {
2    ...
3    "request_id": "request_ticket_charge_1",
4    ...
5    "amount": 10000,
6    ...
7    "merchant_ext_id": "externalmerchant",
8    ...
9    "store_ext_id": "externalstore",
10   ...
11   "payment_code": "3acd50ba-5ec5-11ea-bc55-0242ac13003",
12   ...
13   "ticket": "EF1231",
14   ...
15   "payment_reference_id": "isv_ticket_charge_1",
16   ...
17   "expiry_time": 1664693788,
18   ...
19   "currency": "IDR",
20   ...
21   "promo_ids": "1,2,3",
22   ...
23   "additional_info": "{\"field1\":\"Name\", \"field2\":\"Customer ID\", \"field3\":\"Phone Number\"}"
```

Sample Response

```
1  {
2    ...
3    "request_id": "request_ticket_charge_1",
4    ...
5    "errcode": 0,
6    ...
7    "debug_msg": "success",
8    ...
9    "transaction": {
10      ...
11      "reference_id": "isv_ticket_charge_1",
12      ...
13      "amount": 10000,
14      ...
15      "create_time": 1664693778,
16      ...
17      "update_time": 1664693778,
18      ...
19      "transaction_sn": "123991230868361799",
20      ...
21      "status": 3,
22      ...
23      "transaction_type": 13,
24      ...
25      "merchant_ext_id": "externalmerchant",
26      ...
27      "terminal_id": "T2903",
28      ...
29      "user_id_hash": "e1030777a3868f152526ebc19bd899d0a68aabb087a38a759b1bb9577d04483b",
30      ...
31      "store_ext_id": "externalstore",
32      ...
33      "payment_channel": 1
34    }
35  }
```

4.5.3.2 Check Transaction Status API

Please refer to [Check Transaction Status API](#)

4.5.3.3 Create Refund API [New]

Please refer to [Create Refund \[New\] API](#)

5. Common APIs

The Common API serves as the trivial API interface to be called for additional functions. Clients can choose to integrate these functions based on their needs. The Common API includes Query status, create refund, set merchant, set store, get transactions

5.1 Notify Transaction Status

This request will send details of a transaction result to the Client server's callback URL. This URL is configured in ShopeePay system at time of onboarding, or sent in API requests.

Each callback request will contain a signature in the request header. This signature is generated using the ShopeePay-issued secret key, assigned to the Client receiving the callback request. The Client should use this signature to verify the authenticity of the callback content.

Multiple repeated notifications might be sent to the Client and Client shall be able to handle the repeated notifications with same contents properly.

[Note] It is essential for the Client to validate that payment *amount* and *payment_reference_id* matches the original /qr/create request before considering the payment as successful.

Common Fields

Field	Type	Description
amount	int64	Amount of the transaction, refer to Parameters Specifications for specifics.
transaction_sn	string	Transaction identifier in ShopeePay system.
payment_method	uint32	Indicates the payment method used. Refer to Appendix 2 - Point of Initiation for the respective payment methods.
user_id_hash	string	Unique identifier of the user making the payment.
merchant_ext_id	string	Unique identifier of the merchant in client's system.
store_ext_id	string	Unique identifier of the store in client's system.

terminal_id	string	Terminal id in merchant's MPM/CPM API request. This field will be empty if the payment flow is not MPM or CPM.
payment_channel	uint32	Indicates the source of fund used. Refer to Appendix 1E - Payment Channels for the respective source of fund.
co_funding	object	Will return empty if no co-funding involved . Note: this object is temporarily unavailable in ID, MY, PH, SG.
• promotion_ext_name	string	Name of the promotion for client's reference.
• redeemed_promotion_amount	int64	Amount of the promotion benefit obtained by user. Note: for refund transaction types, this will be the amount of promotion benefit ShopeePay clawbacks from the user in that refund transaction. Refer to Parameters Specifications for specifics.
• redeemed_promotion_type	uint32	Refer to Appendix 1D - Promotion Types for specific transaction types.
• co_funding_cost_amount	int64	Amount of cost born by merchant for the promotion benefit. Note: for refund transaction types, this will be the amount of co-funding amount that the client will no longer be charged for due to that refund transaction. Refer to Parameters Specifications for specifics.

For ID, MY, PH, SG: Additional Fields For Transactions that are not part of [Account Linking APIs](#)

Field	Type	Description
payment_reference_id	string	Unique identifier of payment transaction generated by client.
payment_status	uint32	Successful payment: payment_status = 1

		Refer to Appendix 1B - Payment Status for the respective payment status.
--	--	--

For ID, MY, PH, SG: Additional Fields For Transactions that are part of [Account Linking APIs](#)

Callbacks for transactions related to Account Linking (i.e. Payment Authorization, Payment Capture, Authorization Reversal, Direct Payment) will contain the following fields.

Field	Type	Description
reference_id	string	A new field introduced to generalise the notify callback for all types of transactions. This could be either payment_reference_id or capture_reference_id.
transaction_type	uint32	A new field introduced to generalise the notify callback for all types of transactions. Refer to Appendix 1A - Transaction Types for specific transaction types.
transaction_status	uint32	A new field introduced to generalise the notify callback for all types of transactions. Refer to Appendix 1C - Transaction Status for specific transaction statuses.

For TH, VN: Additional Fields for All Payment Transactions

Field	Type	Description
reference_id	string	A new field introduced to generalise the notify callback for all types of transactions. This could be either payment_reference_id or capture_reference_id.
transaction_type	uint32	A new field introduced to generalise the notify callback for all types of transactions. Refer to Appendix 1A - Transaction Types for specific transaction types.
transaction_status	uint32	A new field introduced to generalise the notify callback for all types of transactions. Refer to Appendix 1C - Transaction Status for specific transaction statuses.

Response Parameters

Note: in Thailand and Vietnam, if Client does not respond with code = 0, ShopeePay will resend notifications up to 2 times.

Field	Type	Mandatory	Description
errcode	int32	Y	Error code to specify the error returned - 0: success
debug_msg	string	N	Debug message to provide more information.

Sample Callback For Transactions that are not part of [Account Linking APIs](#)

```
{
  "amount": 100,
  "transaction_sn": "162950422138187054",
  "payment_method": 2,
  "user_id_hash": "15e455125fba426a4a2bb9145b3af2906813a7f222f6eab93b026ead62dc8d76",
  "merchant_ext_id": "externalmerchant",
  "store_ext_id": "externalstore",
  "terminal_id": "T2903",
  "payment_reference_id": "ref-must-be-unique",
  "payment_status": 1,
  "payment_channel": 1
}
```

Sample Callback For Transactions that are part of [Account Linking APIs](#)

```
{
  "amount": 100,
  "transaction_sn": "125049480044539088",
  "payment_method": 128,
  "user_id_hash": "0669713efca2e64614dd00ebc766b50b1f74e5916f54ebdf14b35ebde73ebf3",
  "merchant_ext_id": "externalmerchant",
  "store_ext_id": "externalstore",
  "reference_id": "ref-must-be-unique2",
  "transaction_type": 13,
  "transaction_status": 3,
  "payment_channel": 1
}
```

5.2 Check Transaction Status API

Use this API to query the status of a transaction across all supported transaction types.

- URL: "/v3/merchant-host/transaction/check"
- Check Transaction Status API can not be used for transactions (payment/refund) which are not created using ShopeePay Payment API.

If Check Transaction Status API returns:

- 1) Success - payment process stops and Client can mark this transaction as success.
- 2) Not found - retry the payment process from the start.
- 3) No response - continue retrying.
 - a) Retry should be done at an incremental time range, at our recommended retry timing at the following time marks:
 - i) 5 seconds
 - ii) 10 seconds
 - .
 - .
 - .
 - vi) 30 seconds
- 4) After 30 seconds of retry with no response from ShopeePay, Client can consider payment transaction to have timed out. Differences can be settled during the T+1 reconciliation process with ShopeePay.

[Note] This API is an upgraded version of previous Check Payment Status API to support transaction status inquiry for all transaction types including Auth, Capture and Auth Reversal.

Request Parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier of the request.
reference_id	string	Y	Unique identifier of the transaction client would like to query, which could be payment_reference_id,

			capture_reference_id or refund_reference_id.
transaction_type	uint32	Y	Transaction type of the transaction client would like to query. Refer to Appendix 1A - Transaction Types for specific transaction types.
merchant_ext_id	string	Y	Unique identifier of merchant in client system.
store_ext_id	string	Y	Unique identifier of store in client system.
amount	int64	Y	This amount must match the original transaction amount.

Response Parameters

Field	Type	Description
request_id	string	Unique identifier for request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.
payment_method	uint32	Indicates the payment method used. Refer to Appendix 2 - Point of Initiation for the respective payment methods.
transaction	object	Will return empty if no such transaction.
• reference_id	string	Unique identifier of transaction generated by client. This value should match the reference_id sent in the request, which could be payment_reference_id, capture_reference_id or refund_reference_id.
• amount	int64	Amount of the transaction, inflated by a factor of 100. Refer to Parameters Specifications for specific inflation method
• transaction_sn	string	Transaction serial number of the transaction in ShopeePay system.

<ul style="list-style-type: none"> • status 	uint32	Refer to Appendix 1C - Transaction Status for specific transaction statuses.
<ul style="list-style-type: none"> • transaction_type 	uint32	Refer to Appendix 1A - Transaction Types for specific transaction types.
<ul style="list-style-type: none"> • create_time 	uint32	Create time of the transaction.
<ul style="list-style-type: none"> • update_time 	uint32	Update time of the transaction.
<ul style="list-style-type: none"> • user_id_hash 	string	Identifier for user that made the payment.
<ul style="list-style-type: none"> • merchant_ext_id 	string	Unique identifier of merchant in client system.
<ul style="list-style-type: none"> • store_ext_id 	string	Unique identifier of store in client system.
<ul style="list-style-type: none"> • terminal_id 	string	If terminal ID info exists in transaction.
<ul style="list-style-type: none"> • promo_id_applied 	string	<p>promo_id applied to the payment transaction. This will be filled only if the user received a reward from a valid promotion with the promo_id specified.</p> <p>Note: this field is temporarily unavailable in TH, VN.</p>
<ul style="list-style-type: none"> • payment_channel 	uint32	Indicates the source of fund used. Refer to Appendix 1E - Payment Channels for the respective source of fund.
co_funding	object	<p>Will return empty if no co-funding involved.</p> <p>Note: this object is temporarily unavailable in ID, MY, PH, SG.</p>
<ul style="list-style-type: none"> • promotion_ext_name 	string	Name of the promotion for client's reference.
<ul style="list-style-type: none"> • redeemed_promotion_amount 	int64	<p>Amount of the promotion benefit obtained by user.</p> <p>Note: for refund transaction types, this will be the amount of promotion benefit ShopeePay clawbacks from the user in that refund transaction.</p> <p>Refer to Parameters Specifications for specifics.</p>
<ul style="list-style-type: none"> • redeemed_promotion_type 	uint32	Refer to Appendix 1D - Promotion Types for specific transaction types.

<ul style="list-style-type: none"> co_funding_cost_amount 	int64	<p>Amount of cost borne by merchant for the promotion benefit.</p> <p>Note: for refund transaction types, this will be the amount of co-funding amount that the client will no longer be charged for due to that refund transaction.</p> <p>Refer to Parameters Specifications for specifics.</p>
--	-------	---

Sample Request

```

1 {
2   ...
3   "request_id": "status_request",
4   "reference_id": "payment_abc12345",
5   "transaction_type": 13,
6   "merchant_ext_id": "externalmerchant",
7   "store_ext_id": "externalstore",
8   "amount": 10000
9 }
```

Sample Response

```

1 {
2   ...
3   "request_id": "status_request",
4   "errcode": 0,
5   "debug_msg": "success",
6   "payment_method": 16,
7   "transaction": {
8     ...
9     "reference_id": "payment_abc12345",
10    "amount": 10000,
11    "create_time": 1681465256,
12    "update_time": 1681465281,
13    "transaction_sn": "157438766250444662",
14    "status": 3,
15    "transaction_type": 13,
16    "merchant_ext_id": "externalmerchant",
17    "terminal_id": "A88",
18    "user_id_hash": "15e455125fba426a4a2bb9145b3af2906813a7f222f6eab93b026ead62dc8d76",
19    "store_ext_id": "externalstore",
20    "promo_id_applied": "1,2,3",
21    "payment_channel": 1
22 }
```

5.3 Create Refund API [New]

Use this upgraded create refund API to request a refund of a successful transaction, subject to refund conditions.

- URL: "/v3/merchant-host/transaction/refund/create-new"
- Valid Refund Period: 1-365 days
- Create Refund API can not be used for transactions which are not created using ShopeePay Payment API

[Note] Create Refund API supports idempotent response. Multiple API requests using the same payload will only trigger the action once. It can prevent the risk of creating duplicate refunds. ShopeePay system will return the latest state of refund transaction which was created on the first API request.

Request Parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for request.
reference_id	string	Y	Unique identifier of transaction generated by client. This can be either "payment_reference_id" or "capture_reference_id" passed to ShopeePay when payment/capture was created.
transaction_type	uint32	Y	Refer to Appendix 1A - Transaction Types for specific transaction types. Eligible transaction types for refund are: <ul style="list-style-type: none"> • Payment • Payment captured • Direct payment
refund_reference_id	string	Y	Unique identifier of refund transaction generated by client. Accepts up to 64 characters.
merchant_ext_id	string	Y	Unique identifier of merchant in client system.

store_ext_id	string	Y	Unique identifier of store in client system.
amount	int64	Y	Amount of the refund transaction. Note: Refer to Parameters Specifications for specifics.

Response Parameters

Field	Type	Description
request_id	string	Unique identifier for request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.
transaction	object	Will return empty list if no such transaction.
• reference_id	string	Unique identifier of refund transaction generated by client.
• amount	int64	Amount of the refund transaction, refer to Parameters Specifications for specifics.
• transaction_sn	string	Transaction serial number of the refund transaction in ShopeePay system.
• status	uint32	Refer to Appendix 1C - Transaction Status for specific transaction statuses.
• transaction_type	uint32	(Refer to Appendix 1A - Transaction Types for specific transaction types).
• create_time	uint32	Create time of the individual transaction.
• update_time	uint32	Update time of the individual transaction.
• user_id_hash	string	Identifier for user that made the payment.
• merchant_ext_id	string	Unique identifier of merchant in client system.
• store_ext_id	string	Unique identifier of store in client system.

• terminal_id	string	If terminal ID info exists in transaction.
• payment_channel	uint32	Indicates the source of fund used. Refer to Appendix 1E - Payment Channels for the respective source of fund.
co_funding	object	Will return empty if no co-funding involved. Note: this object is temporarily unavailable in ID, MY, PH, SG.
• promotion_ext_name	string	Name of the promotion for client's reference.
• redeemed_promotion_amount	int64	Amount of the promotion benefit obtained by user. Note: for refund transaction types, this will be the amount of promotion benefit ShopeePay clawbacks from the user in that refund transaction. Refer to Parameters Specifications for specifics.
• redeemed_promotion_type	uint32	Refer to Appendix 1D - Promotion Types for specific transaction types.
• co_funding_cost_amount	int64	Amount of cost born by merchant for the promotion benefit. Note: for refund transaction types, this will be the amount of co-funding amount that the client will no longer be charged for due to that refund transaction. Refer to Parameters Specifications for specifics.

Sample Request

```
1  {
2    "request_id": "refund_request",
3    "reference_id": "payment_abc12345",
4    "transaction_type": 13,
5    "refund_reference_id": "refund_payment_abc12345",
6    "merchant_ext_id": "externalmerchant",
7    "store_ext_id": "externalstore",
8    "amount": 10000
9  }
```

Sample Response

```
1  {
2    "request_id": "refund_request",
3    "errcode": 0,
4    "debug_msg": "success",
5    "transaction": {
6      "reference_id": "refund_payment_abc12345",
7      "amount": 10000,
8      "create_time": 1681474164,
9      "update_time": 1681474164,
10     "transaction_sn": "164881881748164612",
11     "status": 3,
12     "transaction_type": 15,
13     "merchant_ext_id": "externalmerchant",
14     "terminal_id": "A88",
15     "user_id_hash": "0669713efca2e64614dd00ebc766b50b1f74e5916f54ebdfd14b35ebde73ebf3",
16     "store_ext_id": "externalstore",
17     "payment_channel": 1
18   }
19 }
```

5.4 Get Transactions API

Use this API to retrieve the list of transactions for a certain time range, for purposes such as reconciliation.

The maximum number of transactions returned for each API request is 5000. API requests should be made multiple times in succession until the response body is empty, indicating that all transactions within the time range have been returned.

For every new API request made within the specified time range, the request should indicate a *last_reference_id* value, which is the *reference_id* of the last transaction returned in the response of a previous API request.

The transactions will be returned in *Descending* order by update time (i.e. the latest transaction will be returned first).

- URL: “/v3/merchant-host/transaction/list”

Note: This API is temporarily unavailable for use in Thailand and Vietnam.

Request parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for request
created_since	uint32	Y	Filter transactions from the start of this time
created_until	uint32	Y	Filter transactions to the end of this time
last_reference_id	string	N	The last transaction that was returned in the previous API response
last_transaction_type	uint32	N	The last transaction type (Appendix 1A - Transaction Types) that was returned in the previous API response
transaction_type_list	[] uint32	N	Refer to Appendix 1A - Transaction Types for specific transaction types
limit	uint32	N	The maximum number of transactions that will be returned in each request. This value should be within [1, 5000].

			If this value is not filled, the latest 100 records will be returned by default.
--	--	--	--

Response Parameters

Field	Type	Description
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.
transaction_list	object	List of transactions.
• reference_id	string	Reference ID is the unique identifier of the transaction generated by the client.
• amount	int64	Amount used for the payment, inflated by a factor of 100.
• create_time	uint32	Create time of the individual transaction.
• update_time	uint32	Update time of the individual transaction.
• transaction_sn	string	Transaction identifier in ShopeePay system.
• status	uint32	Refer to Appendix 1C - Transaction Status for specific transaction statuses.
• transaction_type	uint32	Refer to Appendix 1A - Transaction Types for specific transaction types.
• merchant_ext_id	string	Unique identifier of merchant in client system.
• terminal_id	string	Terminal ID refers to the Point of Sale terminal in a store.
• user_id_hash	string	Used to identify the unique user.
• store_ext_id	string	Unique identifier of store in client system.
• promo_id_applied	string	promo_id applied to the payment transaction. This will be filled only if the user received a reward from a valid promotion with the promo_id specified.

• payment_channel	uint32	Indicates the source of fund used. Refer to <u>Appendix 1E - Payment Channels</u> for the respective source of fund.
-------------------	--------	--

5.5 Set Merchant API

Use this API to create a Merchant in the ShopeePay system.

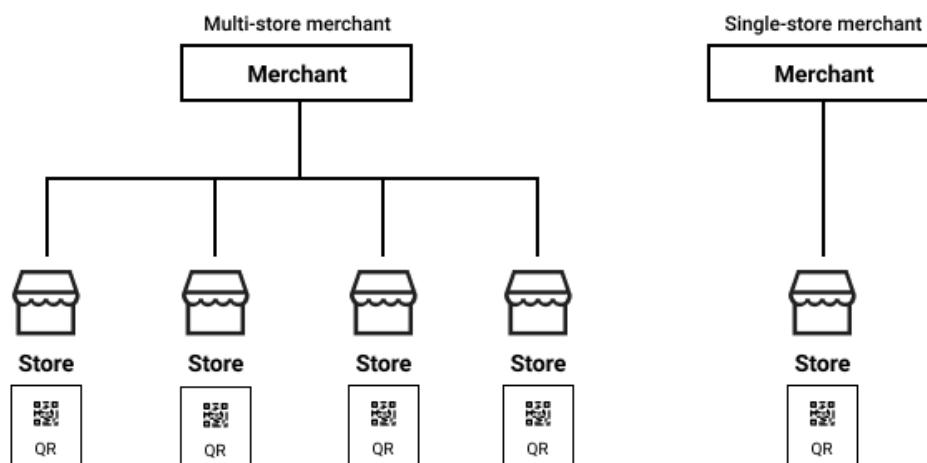
- URL: "/v3/merchant-host/merchant/set"

Note: This API is only available in Philippine and Malaysia based on ShopeePay operational requirements.

Structure: 1 Merchant - has many Stores (minimum 1 store).

Periodic Update: It is highly recommended for Clients to sync the latest Store information in their system to ShopeePay, to ensure all users can enjoy a smooth payment experience at newly added/updated Stores.

The following diagram represents the structure of Merchants and Stores in ShopeePay that integration partners should adhere to during onboarding and sending payment requests. Payments are only accepted on the Store level.



Request parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for request.
merchant_name	string	Y	Name of the Merchant e.g. Starbucks Maximum character length: 50 characters, including blank spaces.

merchant_ext_id	string	Y	Unique identifier of merchant in client system.
phone	string	N	Merchant's main phone number.
email	string	N	Merchant's main email.
logo	string	N	Merchant's logo, encoded in base64.
postal_code	string	Y	Merchant's main postal code.
city	string	Y	Main city that Merchant operates in.
state	string	N	Main state that Merchant operates in.
district	string	N	Main district that Merchant operates in.
ward	string	N	Main ward that Merchant operates in.
address	string	N	Merchant's main address.
business_tax_id	string	N	Business Tax ID used in that country.
national_id_type	uint32	N	Type of identification. Refer to Appendix 4 - National Identity Type for values to use.
national_id	string	N	National identity number.
additional_info	string	N	Additional information stored in JSON string.
mcc	string	Y	Refers to Merchant Category Code, refer to Appendix 3 - Merchant Category Code for suggestions.
point_of_initiation	uint32	Y	Method of QR Payment, Refer to appendix Appendix 2 - Point of initiation for complete list of values to use.
settlement_emails	string	N	Emails to receive daily settlement report of all Stores under the Merchant. For multiple emails, separate by commas.
withdrawal_option	uint32	Y	Withdraw to individual merchant's bank account = 1. Withdraw to Integration Partner's bank account = 3.

status	uint32	N	Status of Merchant OK = 0 Banned = 4 Client can only move status from 0 to 4 and vice versa.
--------	--------	---	--

Response Parameters

Field	Type	Description
request_id	string	Unique identifier for request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.
merchant	object	Merchant Entity.
<ul style="list-style-type: none"> ● merchant_name ● merchant_host_id ● merchant_ext_id 	string uint32 string	Name of the Merchant e.g. Starbucks. Aggregator ID given by ShopeePay Server. Unique identifier of merchant in client system.
<ul style="list-style-type: none"> ● status 	uint32	Status of Merchant Merchant OK = 0 Banned by ShopeePay = 1 Frozen by ShopeePay = 2 Deleted on ShopeePay = 3 Banned by Client via API = 4
<ul style="list-style-type: none"> ● mcc ● point_of_initiation ● phone ● email ● postal_code ● city ● state 	string uint32 string string string string string	Merchant category code. Binary value of available POI values. Merchant's main phone number. Merchant's main email. Merchant's main postal code. Merchant's main city. Merchant's main state.

• district	string	Merchant's main district.
• ward	string	Merchant's main ward.
extinfo	object	
• address	string	Merchant's main address.
• settlement_emails	string	Emails to receive daily settlement report.
• withdrawal_option	uint32	Withdrawal option of Merchant.
• national_id_type	uint32	National identity type.
• national_id	string	National identity number.
• business_tax_id	string	Tax identity number.
• additional_info	string	Additional information relating to Store.

Update Merchants' Information

Clients can update information of previously added merchants via the [Set Merchant API](#), using "merchant_ext_id" as the identifier for each merchant.

Mandatory Field: merchant_ext_id

The only mandatory field is "merchant_ext_id".

Fields that are not filled in in the [Set Merchant API](#) during updating will not be updated.

Fields that are mandatory during merchant onboarding cannot be updated to empty values.

5.6 Set Store API

Use this API to create Stores for a Merchant in the ShopeePay system. A store has to be created under a merchant before payment APIs can be successfully called by that store.

- URL: "/v3/merchant-host/store/set"

Note: This API is only available in Philippine and Malaysia based on ShopeePay operational requirements.

Request parameters

Field	Type	Mandatory	Description
request_id	string	Y	Unique identifier for request.
store_name	string	Y	Name of this store (max. 25 characters).
store_full_name	String	N	Full name of the store which will be used on static QR display (max. 50 characters). Note: this field is only applicable in SG, MY, PH.
merchant_ext_id	string	Y	Unique identifier of merchant in client system (max .64 characters).
store_ext_id	string	Y	Unique identifier of store in client system (max .64 characters).
status	uint32	N	Status of Store OK = 0 Banned = 4 Client can only move status from 0 to 4 and vice versa.
point_of_initiation	uint32	Y	Method of QR Payment, Refer to appendix Appendix 2 - Point of initiation for complete list of values to use.
phone	string	N	Phone number for this store.
email	string	N	Store's email.

logo	string	N	Store's logo, encoded in base64.
address	string	N	Address of the store.
postal_code	string	Y	The postal code that this store is operating in.
city	string	Y	The city that this store operates in.
state	string	N	The state that this store operates in.
district	string	N	The district that this store operates in.
ward	string	N	The district that this store operates in, if applicable.
gps_longitude	uint32	N	Longitude coordinates of the store - inflated by 100000.
gps_latitude	uint32	N	Latitude coordinates of the store - inflated by 100000.
terminal_ids	[] string	N	Point-of-Sale ID to identify every terminal that a store has, comma separated.
mcc	string	Y	Refers to Merchant Category Code, refer to Appendix3 - Merchant Category Code for suggestions.
nmid	string	N	National merchant ID, applicable to countries with QR interoperability (e.g. ID).
merchant_criteria	string	N	<p>Merchant Criteria</p> <p>“UMI”: Micro merchant “UKE”: Small merchant “UME”: Medium merchant “UBE”: Big merchant</p> <p>Compulsory for ID.</p>
settlement_emails	string	N	Emails to receive daily settlement report. For multiple emails, separate by commas.
additional_info	string	N	Additional information stored in JSON string.

Response parameters

Field	Type	Description
request_id	string	Unique identifier for request.
errcode	int32	Error code to specify the error returned.
debug_msg	string	Debug message to provide more information.
store	object	Store Entity.
• store_name	string	Store name.
• merchant_host_id	uint32	Aggregator ID given by ShopeePay Server.
• merchant_ext_id	string	Merchant External Identifier Merchant ID given by aggregator.
• store_ext_id	string	Store External Identifier Aggregator issued store ID.
• status	uint32	Status of Store Merchant OK = 0 Banned by ShopeePay = 1 Frozen by ShopeePay = 2 Deleted on ShopeePay = 3 Banned by Client via API = 4
• point_of_initiation	uint32	Method of QR Payment, Refer to appendix Point of initiation (POI) values for complete list of values to use.
• phone	string	Store's phone number.
• email	string	Store's email.
• postal_code	string	Store's postal code.
• city	string	Store's city.
• state	string	Store's state.
• district	string	Store's district.
• ward	string	Store's ward, if applicable.

• mcc	string	Refers to Merchant Category Code, refer to Appendix3 - Merchant Category Code for suggestions.
• nmid	string	National merchant ID, applicable to countries with QR interoperability (e.g. ID).
• merchant_criteria	string	Merchant criteria.
• create_time	uint32	Timestamp that the store was created.
• update_time	uint32	Timestamp that the store was updated.
• gps_longitude	uint32	Longitude coordinates of the store - inflated by 100000.
• gps_latitude	uint32	Latitude coordinates of the store - inflated by 100000.
• store_id	string	Unique identifier of store by ShopeePay system.
extinfo	object	
• address	string	Address of the store.
• settlement_emails	string	Emails to receive daily settlement report.
• withdrawal_option	uint32	Set to option 4 (follows merchant withdrawal option) by default.
• merchant_criteria	string	Compulsory for QRIS in ID.
• additional_info	string	Additional information relating to Store.
• terminal_ids	[] string	List of terminal IDs under Store.

Update Stores' Information

Clients can update information of previously added merchants via the [Set Store API](#), using "merchant_ext_id" and "store_ext_id" as the identifier for each merchant.

Mandatory Field: merchant_ext_id, store_ext_id

The only mandatory fields are "merchant_ext_id" and "store_ext_id". No other mandatory fields are required for **Update Stores' Information**. Fields that are not filled in in the [Set Store API](#) during updating will not be updated.

Fields that are mandatory during store creation cannot be updated to empty values.

Appendix

Appendix 1A - Transaction Types

Transaction type	Value	Can be queried in Get Transactions API
Payment	13	Y
Refund	15	Y
Adjustment (Add)	32	Y
Adjustment (Deduct)	33	Y
Payment Authorised	1000	N
Payment Captured	1001	Y
Authorisation Reversed	1002	N
Direct Payment	13	Y
Capture Refund	15	Y
Direct Payment Refund	15	Y

Appendix 1B - Payment Status

Payment Status	Value
Payment successful	1
Payment not found	2
Payment refunded	3
Payment cancelled	4
Payment processing	5
Payment failed	6

Appendix 1C - Transaction Status

Transaction status	Value
Transaction initial	1
Transaction processing	2
Transaction successful	3
Transaction failed	4
Transaction expired (*)	6
Transaction invalidated (*)	7
Transaction refunded (*)	8

(*) for future use

Appendix 1D - Promotion Types

Promotion types	Value
Coins	1
Cashback	2
Discount	3

Appendix 1E - Payment Channels

Payment Channel	Value
ShopeePay wallet balance	1
Credit card / debit card	2
Linked bank account	3
SPayLater - 1 month instalment	4
SPayLater - 2 months instalment	5
SPayLater - 3 months instalment	6
SPayLater - 6 months instalment	7
SPayLater - 12 months instalment	8

Appendix 2 - Point of Initiation

POI methods offered	Binary value	Decimal value to indicate during onboarding
MPM Static only	00000001	1
MPM Dynamic only	00000010	2
CPM only	00000100	4
Deals only	00001000	8
Jump App only	00010000	16
Tokenised Payment (Direct Payment & Auth and Capture)	10000000	128
Cross Border only	100000000	256
Disbursement only	1000000000	512
Ticket Linking only	10000000000	1024
Account Linking Only	100000000000	2048

Sample decimal value calculations

POI methods offered	Binary value	Decimal value to indicate during onboarding
MPM Static, MPM Dynamic	0011	3
MPM Static, CPM	0101	5

Appendix 3 - Merchant Category Codes (MCC)

The following table contains some suggested MCC codes that can be assigned to Merchants and Stores depending on the type of business.

Type of merchant	MCC
Miscellaneous general merchandise stores	5399
Department stores	5411
Grocery stores, supermarkets	5411
Bakeries	5462
Miscellaneous food stores, convenience stores, markets, specialty stores, and vending machines	5499
Automotive parts, accessories stores	5533
Men's and Boy's clothing and accessories stores	5611
Men's and women's clothing stores	5691
Miscellaneous accessory and apparel stores	5699
Equipment, furniture and home furnishing stores (excluding appliances)	5712
Eating places, restaurants	5812
Fast food chains	5814
Drug stores and pharmacies	5912
Health and beauty shops	7298
Education services	8299
Charities, social services and government	8398

For other merchant category codes, clients may choose to take reference from the [merchant standards manual by Visa](#).

Appendix 4 - National Identity Types

Country	Value	National identity type
ID	1	KTP
ID	2	KITAS
MY	1	IC
MY	2	Passport
SG	1	NRIC
SG	2	FIN
PH	1	Passport
PH	2	Driver's License
PH	3	National Bureau of Investigation (NBI) Clearance
PH	4	Alien Certificate of Registration
PH	5	Immigrant Certificate of Registration
PH	6	Unified Multi-Purpose ID (UMID)
PH	7	Social Security System (SSS) with Date of Birth
PH	8	TIN ID
PH	9	Professional Regulation Commission (PRC) ID
PH	10	PhilHealth ID with Date of Birth
PH	11	Postal ID (Issued 2015 onwards)
PH	12	Integrated Bar of the Philippines ID
PH	13	Philippine Postal ID (Issued November 2016 onwards)
PH	14	Police Clearance

PH	15	LTO Student Permit
PH	16	Senior Citizen ID
PH	17	Commission on Elections (COMELEC) Voter's ID
PH	18	TIN Card
PH	19	OFW ID Card
PH	20	PWD ID
PH	21	PhilSys ID

Appendix 5 - Error Codes

Value	Description
-2	A server dropped the connection
-1	A server error occurred
0	Success
1	Request parameters error, such as missing mandatory parameters, for example, "Invalid token" or "Invalid currency" etc.
2	Permission denied
4	Merchant/store/user/transaction not found
5	Transaction processing, use Notify or Check Transaction Status API for updated payment status
6	The user making the payment has not completed wallet activation
7	Expired
9	User's account is banned
11	Duplicate request/transaction
14	User's account is deleted
15	Payment amount exceeded limit
24	User's account is frozen
42	Insufficient balance
50	Payload sent is not a valid json
101	User wallet limit exceeded
102	User wallet limit exceeded
103	User exceeded daily payment limit Limit will reset the next day
104	User wallet limit exceeded
105	Invalid auth code

121	Client attempts to update completed transaction
126	User transaction limit reached
140	User wallet limit exceeded
150	Active linking count threshold reached
152	Fail to unlink due to ongoing auth
301	Invalid payment code or QR content
303	Merchant is trying to make payment to their own user account
304	Refund cannot be processed due to payment exceeding validity period
305	Merchant invalid
308	Refund cannot be processed due to payment was made by using non-refunded voucher
309	MDR fee don't have enough balance where merchant doesn't have enough balance to refund a transaction
320	Refund cannot be processed due to another ongoing refund request for the transaction
321	Refund cannot be processed due to reach maximum refund limit
401	Refund cannot be processed due to refund amount mismatch with transaction amount
601	<p>Request to refund a payment transaction does not meet rules such as refund operation is not allowed for the chosen merchant/store, transaction has already been refunded before, and refund to a payment made by a different merchant/store.</p> <p>This error is also returned during refund block periods, set at 12.00am to 5am by default.</p> <p><i>This timing is subject to changing during campaign periods or system maintenance</i></p>
602	Request to refund an unsuccessful payment transaction
603	Can not use the selected payment channel
604	Can not use this voucher

629	Publisher Id not matched
630	SKU ID not matched
702	Client is not allowed to use coins
705	User does not allow coin redemption
1001	User is not allowed to make the transaction
1002	This service is currently under maintenance
1100	ShopeePay internal payment module timeout
1101	ShopeePay internal payment module general error
1102	User's SPayLater credit limit is reached
1103	User is required to link their bank account first before making payment
1104	Unable to use selected payment channel to complete the transaction
1105	Transaction failed due to unsuccessful multi factor authentication
1302	Insufficient balance for refund
1901	QR code can not be invalidated
1902	Jump App order can not be invalidated
1903	Unable to create capture request because auth payment is still in processing
1904	Unable to create capture request because auth payment is already expired
1907	No eligible payment channel available for tokenized payment transaction
1908	ShopeePay is having system maintenance

Version Control

Version	Date	Editor	Remarks
V1.0	11.12.2019	G.Z. & J.L.	Initial draft
V1.1	17.12.2019	G.Z.	1 - Add user_id_hash as unique identifier for user paying to merchant 2 - Add currency indicator for amount 3 - Add amount inflation of 100
v1.2	3.1.2020	H.Y., Y.S.	1 - Remove "QRIS" 2 - Update set merchant/store API
V1.3	6.1.2020	G.Z.	1 - Update definition of user_id_hash 2 - Update definition and usage of terminal ID 3 - Add "district" and "ward" for onboarding merchant and stores
V1.4	13.1.2020	Y.S., G.Z., H.Y., J.L.	1 - Combine MPM and CPM integration documents 2 - Upgrade and rename all CPM and MPM endpoints to v3 3 - Add refund transaction details in refund response 4 - Add parameters in set merchant & set store 5 - Add qr_validity_period to /QR/create API 6 - Add support for QR Invalidate API
V1.45	10.2.2020	G.Z.	1 - Add new fields in /merchant/set and /store/set/ APIs 2 - Add sample request and responses
V1.46	11.2.2020	H.Y.	1 - Update definition of withdrawal_option values
V1.47	28.2.2020	G.Z.	1- Specify values of transaction status in appendix 1C 2- Specify link to appendix 1C for "status" response parameter
V1.48	16.3.2020	G.Z.	1 - Add signature generation sample 2 - Add App-to-app redirection flow and API specs 3 - Add logo parameter to both merchant and store

V1.49	26.5.2020	G.Z.	1 - Add details for essential fields to validate in Check Status API request and Notify Transaction Status callback 2 - Add time ranges where refund/void requests are blocked for settlement processing
V1.50	27.5.2020	J.L.	1 - Added Signature Verification instructions
V1.51	22.6.2020	G.Z.	1 - Add "amount" parameter in check status API
V1.52	14.7.2020	G.Z.	1 - Update guidelines for the usage of Get Transactions API
V1.53	05.8.2020	S.F., G.Z.	1 - Update error code mappings 2 - Add payment_method value in Notify Transaction Status API
V1.54	12.10.2020	G.Z.	1 - Add payment_method value in Check Payment Status API response
V1.55	22.10.2020	S.F.	1 - Add promo_ids field in Create Dynamic QR API, Create CPM Payment API and Create Order API requests 2 - Add promo_id_applied field in Check Payment Status, Get Transactions API responses, and Notify Transaction Status callback
V1.55.1	13.11.2020	S.F.	1 - Remove promo_id_applied field from Notify Transaction Status callback 2 - Add error code mapping for invalid promo_ids
V1.56	10.11.2020	Y.X.	1 - Add Account Linking flows and APIs 2 - Update Notify Transaction Status to generalise the notification to all kinds of transactions 3 - Add Check Transaction Status API 4 - Add new Create Refund API 5 - Reorganise Set Merchant API response
V1.56.1	17.11.2020	G.K.	1- Added Cash In (Top Up) flows and APIs 2- Added New Transaction Type 3- Added New Point of Initiations
V1.56.2	08.1.2021	W.T, Y.C	1- Added new error code mapping - 5, for payment processing, check callback and check status API for updated status

			2- Added new error code 1001 and 1002, to handle access control and system maintenance use case.
V1.56.3	14.1.2021	Y.X.	1- Updated Appendix 1A - Transaction Types table 2- Updated Account Linking APIs and sample request/response.
V1.56.4	26.1.2021	S.F.	1- Updated default withdrawal option for Set Store API
V1.56.5	26.2.2021	G.Z.	1 - Updated valid void period from 365 days to within the create date of the original payment
V1.57	08.3.2021	G.S.	1 - Added Order Invalidate API 2 - Added new Error Code 452 for Order Invalidate API
V1.58	13.4.2021	Y.X.	1 - Remove access scope field in Get User Info API response 2 - Remove Appendix 6 - Access Scope 3 - Update `payment_reference_id`, `auth_reference_id`, `reverse_auth_reference_id` and `capture_reference_id` description in Account Linking Direct Payment, Create Auth, Reverse Auth, Capture APIs
V1.59	19.4.2021	G.Z.	1 - Deprecate Void API. Refund API provides the same functionality to fulfil the refund needs of ISVs and merchants
V1.60	26.4.2021	G.Z.	1 - Update supported currencies and API domains for Thailand and Vietnam 2 - Indicate temporarily unavailable APIs in Thailand and Vietnam: Create Auth, Reverse Auth, Get User Info, Get Redeemable Coin Value, Create Refund, Cash In (Topup) APIs
V1.61	19.5.2021	P.L.	1 - Indicate a temporarily unavailable field (promo_ids) in Thailand and Vietnam for the following APIs: Create Dynamic QR, Create CPM Payment, Create Order, Direct Payment. 2 - Indicate a temporarily unavailable field (use_coin) in Thailand and Vietnam for the following APIs: Direct Payment. 3 - Update Thailand's domain from .in.th to .co.th

V1.62	24.5.2021	G.Z., P.L.	<p>1 - Add kyc_passed value in Get User Info API</p> <p>2 - Indicate the partial refund capability of Create Refund API [New] in Thailand and Vietnam</p> <p>3 - Indicate a temporarily unavailable API in Thailand and Vietnam: Get Transactions API</p> <p>4 - Add co_funding in the responses of following APIs: Create Refund API [New], Check Transaction Status API [New], Create CPM Payment and in the request of Notify Transaction Status API.</p> <p>Note: this field will only be available for Thailand and Vietnam.</p>
V1.63	19.7.2021	P.L.	<p>1 - Add new error code 150 (active linking count threshold reached) for Account Link API</p> <p>2 - Indicate a temporarily unavailable field (user_id_hash) in Thailand and Vietnam</p> <p>3 - Add "platform_type" field in request parameters for Create Order API in Thailand and Vietnam and indicate that "point_of_initiation" field' in Create Order API is for ID, MY, PH, SG only</p> <p>4 - Indicate non-applicable result codes in Account Link API in Thailand and Vietnam</p> <p>5 - Indicate a temporarily unavailable response field (redirect_url_app) in Thailand and Vietnam for Create Order API</p> <p>6 - Add Transaction Type 1004 for Capture Refund and 1005 for Direct Payment Refund</p> <p>7 - Indicate a non-applicable flow: Cash in and non-applicable APIs: Generate Cash In QR Code API and Cash In QR Invalidate API in Thailand and Vietnam</p> <p>8 - Add expiry_time field in the following APIs: Create Dynamic QR, Create CPM Payment, and Create Order. This field is only applicable in Thailand and Vietnam.</p> <p>9 - Update Check Payment Status API to be Check Transaction Status API, and Check Refund API to be Check Refund API [New] in all payment methods.</p>
V1.64	23.7.2021	G.Z., P.L., M.E.	<p>1 - Add eligible National ID Type value = 21, for PH PhilSys ID</p> <p>2 - Add error code 152 for failure to unlink account linking due to ongoing auth transactions</p> <p>3 - Add transaction_sn in the response parameters for Direct Payment and Capture API</p>

			<p>4 - Remove Check Payment Status and Create Refund API from API docs as new merchants shall always call Check Transaction Status and Create Refund [New] API instead</p> <p>5 - Add user_id_hash in the response of Get Access Token API</p> <p>6 - Add Ticket Linking in Payment method Definition</p> <p>7 - Add new Ticket Linking Flow</p> <p>8 - Add new Point of Initiation for Cross Border, Disbursement and Ticket Linking</p> <p>9 - Indicate that the PIN verification process in Direct Payment Flow is currently not applicable to Thailand and Vietnam</p> <p>10 - Add response parameters for Notify Transaction Status and retrying logic</p> <p>11 - Indicate error 50 for wrong request format</p> <p>12 - Indicate the availability of user_id_hash in all API response for Thailand and Vietnam</p> <p>13 - Add additional_info in the request parameters of Direct Payment</p> <p>14 - Add parameters appending mechanisms when visiting return_url in App-to-app redirection payment flow in Thailand and Vietnam</p> <p>15 - Remove mcc and terminal_ip request parameters from CPM Create Payment API</p> <p>16 - Remove point_of_initiation request parameters from Create Order API</p>
V1.65	12.10.2021	P.L., G.Z., M.E., K.V.	<p>1 - Remove the 'Full Refund Only' remark for Indonesia, Malaysia, Singapore, and Philippines</p> <p>2 - Indicate Refund API can not be used for static QR transactions</p> <p>3 - Indicate max value for 'qr_validity_period' request parameter in Create Dynamic QR API and 'validity_period' request parameter in Create Order API</p> <p>4 - Indicate the country code requirement for 'phone' request parameter in Account Link API</p> <p>5 - Change the Transaction Type for Direct Payment as 13</p> <p>6 - Change the Transaction Type for Direct Payment Refund and Capture Refund as 15</p>

			<p>7 - Indicate that promo_id_applied parameter is not available in Check Transaction Status response in Thailand and Vietnam</p> <p>8 - Indicate that promo_ids parameter is applicable for usage in Thailand and Vietnam for Create Dynamic QR API, Create CPM Payment API and Create Order API</p> <p>9 - Indicate the max length as 20 characters for terminal_id parameter in Create Dynamic QR API request for Thailand and Vietnam</p> <p>10 - Indicate max character length for "refund_reference_id" in Refund API, "reverse_auth_reference_id" in Reverse Auth API, and "capture_reference_id" in Capture API</p> <p>11 - Indicate format and max character length for "additional_info" in Create Dynamic QR API, Create CPM Payment API, Create Order API, and Direct Payment API for Indonesia, Thailand and Vietnam</p> <p>12 - Add store_full_name parameter in Set Store API</p> <p>13 - Add "qr" enumeration for platform_type parameter in Account Link API request for Thailand and Vietnam</p> <p>14 - Add a new conditional linking_reference_id parameter and modify auth_code's mandatory type to be conditional in Get Access Token API, Account Unlink API, and Get User Info API for all countries</p> <p>15 - Add service_name, service_id, linking_description parameters in Account Linking API request</p> <p>16 - Add redirect_url_app parameter in Account Link API response</p> <p>17 - Add redirect_url_qr parameter in Account Link API response for Thailand and Vietnam</p> <p>18 - Add service_name, service_id, and linking_description parameters in Account Link API request</p> <p>19 - Remove web page limitation for return_url parameter in Account Link API request</p> <p>20 - Add redirect_url_qr and redirect_url_app parameters in Account Link API response</p> <p>21 - Add linking_reference_id, merchant_ext_id, linking_status, create_time, and update_time</p>
--	--	--	--

			<p>parameters for all countries in Get Access Token API response</p> <p>22 - Add merchant_ext_id, linking_status, create_time, update_time, access_token, user_id_hash, and linking_reference_id parameters in Account Unlink API response</p> <p>23 - Add linking_info object in Get User Info API response</p> <p>24 - Indicate return_url parameter as Conditional "C" in Direct Payment API request and Create Auth API request</p> <p>25 - Add promo_id_applied parameter in Direct Payment API response</p> <p>26 - Remove use_coin and promo_ids parameters in Create Auth API request</p> <p>27 - Add transaction_sn parameter in Create Auth API response</p> <p>28 - Add transaction_sn parameter in Reverse Auth API response</p> <p>29 - Remove order_reference_id parameter in Create Auth and Direct Payment API request</p> <p>30 - Add use_coin, promo_ids, and additional_info parameters in Capture API request</p> <p>31 - Add promo_id_applied parameter in Capture API response</p> <p>32 - Add additional_info parameter on Ticket Charge API request</p> <p>33 - Indicate new error codes i.e. 14, 308, 320, 321, 401, 603, 702, 705, 1302, 1901, 1902, 1903, 1904</p> <p>34 - Remove existing error codes i.e. 452, 627, 628</p> <p>35 - Add more detail and update the UI of user payment flow on Merchant Presented Mode (MPM), Customer Presented Method, App to App, and Ticket Linking</p> <p>36 - Move Cash In APIs to Disbursement API documents</p> <p>37 - Indicate Notify Transaction Status API for Customer Presented Mode (CPM) and Account Linking (Tokenized Payment) transaction</p> <p>38 - Add account link notify status callback for Account Linking (Tokenized Payment) transaction</p> <p>39 - Indicate minimum value of expiry_time parameter and remove the unavailability remark</p>
--	--	--	---

			<p>for Indonesia, Malaysia, Singapore, and Philippines in Create CPM Payment API request</p> <p>40 - Remove url-encoded remark for auth_code parameter in Get Access Token API request</p> <p>41 - Add backward compatible changes section</p> <p>42 - Change refund block time period from '11.55 pm to 3.00am' to '12.00am to 5.00am'</p> <p>43 - Remove the remark for client who requires phone number match in section 4.4.1 User Link and Unlink Account Flow</p>
V1.66	14.03.2023	K.V.	<p>1 - Update the description and remove 'only available in ID, MY, PH, SG' remark for user_id_hash parameter</p> <p>2 - Indicate max character length for payment_reference_id parameter in Create Dynamic QR API request</p> <p>3 - Remove 'only available in ID, MY, PH, SG' remark and indicate max value in TH & VN for qr_validity_period parameter in Create Dynamic QR API request</p> <p>4 - Remove 'only available in TH, VN' remark and indicate max value for expiry_time parameter in Create Dynamic QR API and Create Order API request</p> <p>5 - Add validity_period parameter in Create CPM Payment API request</p> <p>6 - Remove 'only available in ID, MY, PH, SG' remark and indicate max value in TH & VN for validity_period parameter in Create Order API request</p> <p>7 - Remove 'not available in TH, VN' remark for redirect_url_app parameter in Create Order API response</p> <p>8 - Remove result code 201 and remove 'result code 202 not applicable in TH, VN' remark for auth_code parameter in Get Access Token API</p> <p>9 - Remove '5 years expiry time' remark for access_token parameter in Get Access Token API</p> <p>10 - Add value 2 & 3 for update_type parameter in Account Link Notify Status</p> <p>11 - Remove 'not available in TH, VN' remark for linking_info and user_info parameter in Get User Info API response</p>

			<p>12 - Remove 'not available in TH, VN' remark for promo_id_applied parameter in Direct Payment API response</p> <p>13 - Add payment_channel parameter in Create CPM Payment API, Direct Payment API, Create Auth API, Reverse Auth API, Capture API, Ticket Charge API, Check Transaction Status API, Create Refund API, Get Transaction API, and Notify Transaction Status</p> <p>14 - Indicate that Create Refund API and Check Transaction Status API can't be used for transactions which are not made using ShopeePay Payment API</p> <p>15 - Add remark for idempotent response behaviour in Create Refund API</p> <p>16 - Indicate Payment Captured, Direct Payment, Capture Refund, and Direct Payment Refund can be queried in Get Transaction API on Appendix 1A - Transaction Types</p> <p>17 - Indicate 'reserve for future use' for Transaction Status 6, 7, and 8 in Appendix 1C - Transaction Status</p> <p>18 - Add Appendix 1E - Payment Channel</p> <p>19 - Add new error codes i.e. 1100, 1101, 1102, 1103, 1104, 1105, 1907, 1908</p>
--	--	--	---