

General Payout Service Integration Guide

API Documentation

Table of Contents

Table of Contents	2
1. Introduction	4
1.1. About This Document	4
1.2. Intended Audience	4
1.3. Version Control	4
2. Definitions	5
2.1. Term Definitions	5
3. Overview	7
3.1. General Payout Sub-Modules	7
3.1.1. SPBA	7
3.1.2. Safeguard	7
3.2. Use Cases	8
3.3. Payout Flow	8
3.3.1. Platform Agnostic Payout Solution (Pure Payout Solution)	8
3.3.2. Payout With SPBA Solution	9
4. Getting Started	10
4.1. Onboarding & Setting Up	10
5. SPM Payout APIs	12
5.1. Create Payout	12
5.1.1. Request Parameters	13
5.1.1.1. Beneficiary Object	15
5.1.2. Response Parameters	16
5.2. Query Payout	18
5.1.1. Request Parameters	18
5.1.2. Response Parameters	19
5.3. Payout Callbacks	21
5.3.1. Setup	21
5.3.2. Delivery Attempts & Retries	22
5.3.2.1. Payout Confirmation Callback	22
5.3.2.1. Payout Status Change Callback	22
5.3.3. Payout Confirmation Callback API Details	23
5.3.3.1. Request Parameters	23
5.3.4. Payout Status Change Callback	25
5.3.4.1. Request Parameters	26

6. SPBA APIs	28
6.1. Bank Account Name Validation Service	28
6.2. Bank Account Name Validation Results Update	29
7. Reports	30
7.1. Payout Channel Reports	30
7.2. Reports Retrieval	30
7.3. Reports Information	30
7.2. SSG Reports	31
7.3. Bank Account Name Validation Status Reports	32
8. Errors	34
8.1 Error code	34
8.2 Failure Code	34
9. Payout Testing Integration	38
Appendix	40
Appendix 1 - General Payout State Machine	40
Appendix 2 - Supported Countries & Destination Accounts	41
Indonesia	41
Singapore	46
Malaysia	47
Thailand	48
Philippines	49
Taiwan	51
Appendix 3 - Available Payout Channels	53
Appendix 4 - Required Client Business Information	54
Appendix 5 - Limit Information	55
Q&A	56
Version Control	57

1. Introduction

1.1. About This Document

This document is the official manual for integration with Shopee Payment Module (SPM) General Payout Service APIs, intended **only for internal use within the Shopee business ecosystem**.

This document describes the APIs between SPM and any client system which intends to send money to bank accounts, ShopeePay, and AirPay Wallet in SPM supported countries.

1.2. Intended Audience

The intended audience for this document are technical personnels, product managers, and programmers **within the Shopee business ecosystem** 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. Term Definitions

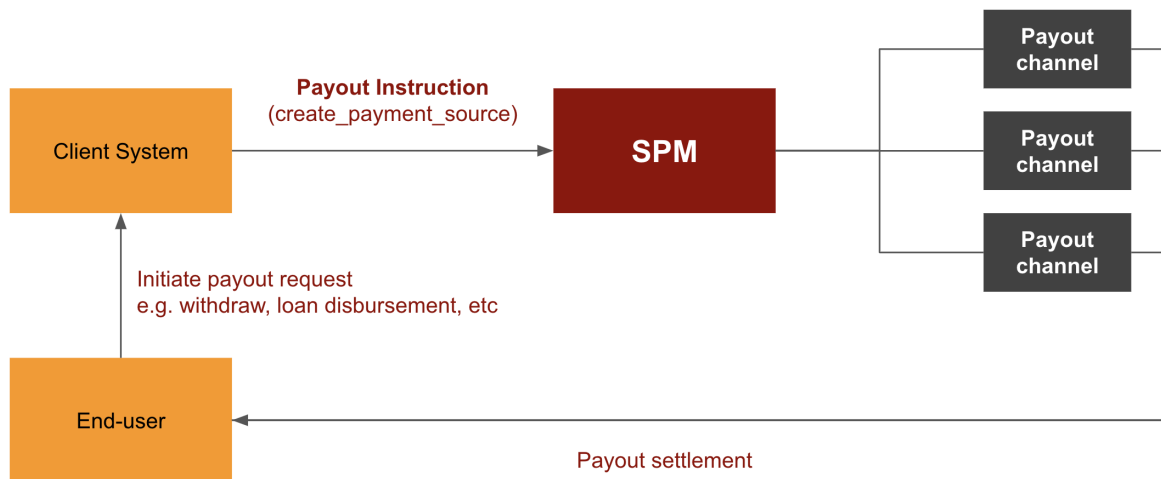
Term	Definition
SPM	Shopee Payment Module, the internal payment gateway that facilitates payment in and out of Shopee businesses.
SPBA	<p>Shopee Bank Account system, the bank account system that stores bank account details of a beneficiary.</p> <p>To utilise SPBA, clients need to use the Shopee User Account system (either toB or toC account).</p>
SSG	Safeguard (SSG) is a module within SPM that helps to minimise the risk of double payout.
Payout	The act of sending money to end-users of any destination account that is supported by SPM.
Client	Refers to the party making API calls.
Payout Channel	Refers to the party that is processing the payout instruction i.e. Banks/ShopeePay.
Beneficiary	The recipient of the payout.
Originating Account	The bank account that is used as the source of funds for the payout. This usually belongs to a Shopee legal entity.
Destination Account	<p>The destination of the payout request. SPM supports multiple destination accounts including Bank accounts, AirPay wallet & ShopeePay wallet.</p> <p>Refer to Appendix 2 for a full list of SPM supported countries & destination accounts.</p>
Source Type	<p>A source type indicates a unique payout use-case that clients wish to use.</p> <p>One client may have more than one source type to help differentiate various use-cases a client may have.</p> <p>For example, a Wallet system could have two different source types: (1) Wallet_Withdrawal</p>

	(2) Merchant_Withdrawal
Inhouse	Fund transfer where the destination account is hosted in the same Bank as the originating account.
Interbank	Fund transfer where the destination account is hosted in a different Bank from the originating account, within the same operating country.
SPM Supported Countries	The countries where SPM operate and offer General Payout service. For a full list please refer to Appendix 2 - Supported Countries .
Shopee Account System	Shopee user account system, which records all the buyers and sellers.

3. Overview

General Payout Service allows clients to send money to supported destination accounts in SPM supported countries. You can send us payout instructions via API.

General Payout Service is designed to help clients within the Shopee ecosystem to easily integrate and enable auto payout for their businesses.



SPM currently supports payout to bank accounts (in SPM Supported Countries), Thailand AirPay Wallet, ShopeePay Wallet (Indonesia, Singapore, Malaysia and Philippines) .

3.1. General Payout Sub-Modules

In processing payout requests from clients, generally there are a few sub-modules that are involved in the process.

3.1.1. SPBA

Shopee Bank Account system (SPBA), is a system that stores bank account details of a beneficiary. As a prerequisite to utilise SPBA, client needs to use the Shopee User Account system (either toB or toC account).

The SPBA APIs enable clients to add, edit, retrieve, and delete bank account details.

3.1.2. Safeguard

Safeguard (SSG), is a system that helps to run some checks & validations for each of the payout requests that SPM receives from clients.

There are some general validations that SSG will run including:

SSG Check	Description	Configurable?
Duplicate reference_ID	SSG will block incoming payout requests if the reference_ID has been received by SPM previously.	SPM, by default, will activate this checker for all clients.
Daily transaction count threshold	If the payout count in a day exceeds the maximum threshold set in SSG, SSG will block further incoming payout requests.	Configurable by source type.
Daily amount threshold	If the payout amount in a day exceeds the maximum threshold set in SSG, SSG will block further incoming payout requests.	Configurable by source type.
Multiple payouts to the same user_id	If SSG detects multiple payout with the same amount being made to the same user_id, SSG will block the payout.	Universal check

3.2. Use Cases

Some of the use-cases that SPM has supported so far include:

- Marketplace/e-commerce seller withdrawal
- Loan disbursements
- Wallet withdrawals

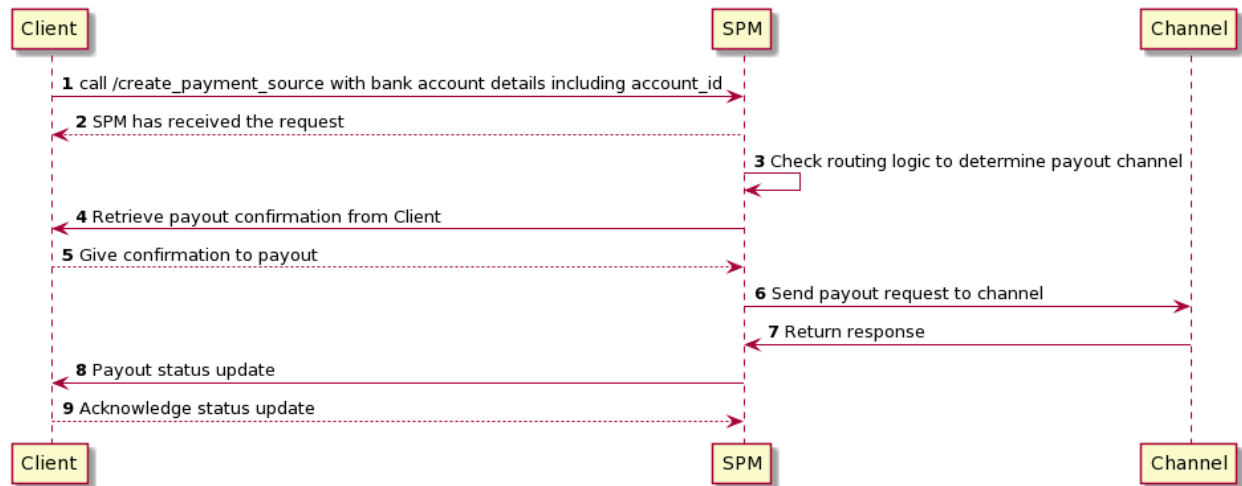
3.3. Payout Flow

In general, there are two payout flows that SPM support: (1) **Platform Agnostic Payout Solution** and (2) **Payout with SPBA Solution**.

3.3.1. Platform Agnostic Payout Solution (Pure Payout Solution)

This solution is suitable for clients that are not operating under the Shopee user account system. In this option, SPM will not store the beneficiary's destination account details. Client will need to send the destination account details to SPM when making the payout instruction.

Note: step 4&5 are optional for clients



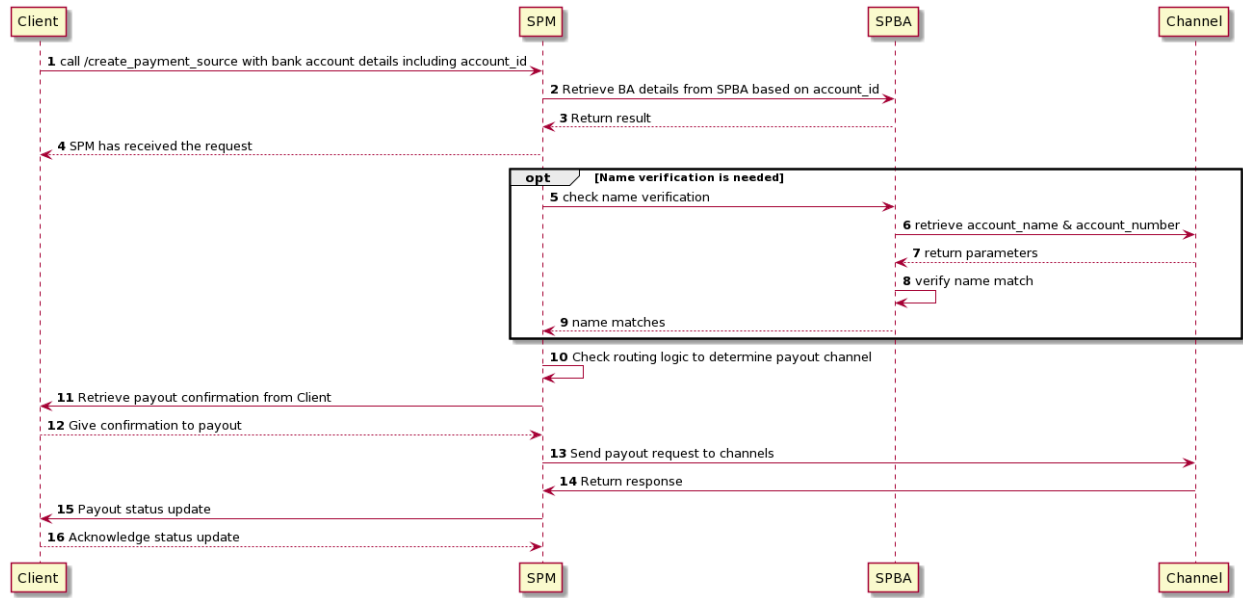
3.3.2. Payout With SPBA Solution

Pre-requisite: Client needs to be integrated with Shopee User Account System, either toC or toB account will work for this Payout with SPBA Solution.

If client chooses to use this payout flow, client may opt to use the Bank Account name validation service as part of the payout flow. Please refer to [6.1. Bank Account Name Validation Service](#) for more details.

This option allows client to store the beneficiary's bank account details in a module called Shopee Bank Account (SPBA) System. This information will be used to help SPM process the payout.

Note: step 11&12 are optional for clients



4. Getting Started

4.1. Onboarding & Setting Up

To start the integration process with SPM General Payout Service, client will need to provide the following information:

Client Information	Description
Business Name	Client's business name. Will be used to generate API endpoints for clients.
Client ID (assigned by SPM)	For new clients, SPM will assign a client_id. For existing clients, SPM may re-use the existing client_id, but client may request for a new client_id if necessary.
Client Keys	SPM will create a set of keys for client for all environments including Test, UAT, and Live.
Source Type	A source type indicates a unique payout use-case that client wish to use. One client may have more than one source type to help differentiate various payout use-cases a client may have.

	<p>SPM will assign an integer value for each of the source types.</p> <p>For example, a Wallet system could have two different source types:</p> <ul style="list-style-type: none"> (1) Wallet_Withdrawal (2) Merchant_Withdrawal
Desired Payout Channels	<p>Refers to the payout channels that clients want to use for the auto-payout. A channel may or may not support an interbank transaction.</p> <p>Please refer to the Appendix 3 for the full list of payout channels.</p>
Bank Accounts	<p>Client's bank account information.</p> <p>Should include:</p> <ul style="list-style-type: none"> 1. Bank Name 2. Bank Account Number 3. Bank Account Owner 4. Bank Branch <p>Example: Bank Name: BCA Bank Account Number: 1234567890 Bank Account Owner: PT Shopee International Indonesia Bank Branch: Grand Indonesia, Jl. M.H. Thamrin.</p> <p>Note: For payout to bank accounts, clients should own at least one bank account in each of the Payout Channels that clients choose to activate.</p> <p>Clients generally have two options:</p> <ul style="list-style-type: none"> (1) Maintain its own bank account (2) Use the existing bank account which SPM already has the proper permission to deduct from <p>For Option (1), additional configuration on the bank side may be required to grant SPM the access to deduct from the said bank account.</p> <p>For Option (2), Client will need to discuss with Finance with regards to using the existing account as the source of funds for client's transactions.</p> <p>Please make sure that you have obtained confirmation from finance and related stakeholders before providing us with the bank account information.</p>
IP Addresses	<p>IP addresses from where clients will make the API call to SPM.</p> <p>SPM will whitelist the IP addresses to secure the API connection between clients & SPM.</p>

Report Recipients	List of email addresses that will receive the daily payout channel reports which will be generated by SPM.
Callback URL	<p>Clients need to define callback URLs that SPM will use to send payout confirmation requests and payout status changes.</p> <p>Refer to 5.3. Payout Callbacks for more details.</p>

5. SPM Payout APIs

SPM has a set of APIs that allow clients to send money to supported destination accounts in SPM supported countries. SPM Payout API includes Create Payout, Payout Query.

As part of the Payout process, SPM uses callback for two purposes: (1) to ask for client's confirmation prior to sending payout to a third-party, (2) notify client of the payout status change.

5.1. Create Payout

Clients can create a payout instruction to SPM by calling the **create payout API**.

This API is designed to be idempotent, which means SPM will return the existing result of the payout request if the client calls this API more than once.

[Note]

When SPM returns "Success" upon the client calling this API, it only guarantees that SPM has successfully received the payout request. The payout may not be processed immediately, as SPM payout processing time is dependent on each Bank's operating hours.

SPM will notify (callback) clients of the payout status asynchronously whenever the payout status changes.

Method	POST
URL	Test: <a href="https://spm.i.test.shopee.co.id/api/<biz_name>/create_payment_source/">https://spm.i.test.shopee.co.id/api/<biz_name>/create_payment_source/ UAT: <a href="https://spm.i.uat.shopee.co.id/api/<biz_name>/create_payment_source/">https://spm.i.uat.shopee.co.id/api/<biz_name>/create_payment_source Live: <a href="https://spm.i.shopee.co.id/api/<biz_name>/create_payment_source/">https://spm.i.shopee.co.id/api/<biz_name>/create_payment_source
Body	JSON
Authorization	Clients need to use the API credential to sign the request and pass the signature through HTTP header HTTP_AUTHORIZATION. The signature is generated using HMAC_SHA256 algorithm. SPM will generate the key for each client in each environment.
Signature demo	Golang type RequestParams struct { TS int `json:"_ts"` SourceType int `json:"source_type"`

```

        Amount      int      `json:"amount"`
        Referenceld  int      `json:"reference_id"`
        ClientId     int      `json:"client_id"`
        Beneficiary   *Beneficiary `json:"beneficiary"`
        BizCommon     string    `json:"biz_common,omitempty"`
        ClientPaymentData string  `json:"client_payment_data,omitempty"`
    }

    func Signature() string {
        //clientKey get from client config
        var (
            url =
            "https://spm.i.test.shopee.vn/api/seller_withdrawal/create_payment_source/"
            clientKey =
            []byte("eb2b2877e9d7b997dc632eb59d7705039cbcdf890b47b0aba824ea5867c32777")
        )
        params := &RequestParams{
            TS: 1624592316,
            Beneficiary: &Beneficiary{
                UserId: 1200070653,
                BankAccount: &BankAccount{
                    AccountId: 738,
                    BankNameId: 9,
                    AccountName: "Fsadfds",
                    AccountNumber: "2342232442",
                },
            },
            SourceType: 6,
            Amount: 800000000,
            Referenceld: 3390706002,
            ClientId: 50002,
        }
        paramsJson, _ := json.Marshal(params)
        baseString := fmt.Sprintf("%s|%s", url, string(paramsJson))
        h := hmac.New(sha256.New, clientKey)
        h.Write([]byte(baseString))
        return hex.EncodeToString(h.Sum(nil))
    }

```



5.1.1. Request Parameters

Request Body:

For the beneficiary, either **[Bank Account]** **[ID - ShopeePay]** **[TH - AirPay]** **[CB-Payee]** is supported.

JSON Key	Type	Description
reference_id mandatory	uint64	<p>Unique ID of the payout request, generated by the client's system.</p> <p>Used to reconcile payouts between SPM & client.</p> <p>Note:</p> <ol style="list-style-type: none">1. Client side to make sure this ID is unique for each payout request sent to SPM2. Client side to make sure this ID is unique for each source_type3. SPM will use reference_id and Source_type as a unique key for idempotency. The same reference_id in the same source_type was considered as the same request.
Source_type mandatory	int	<p>The transaction type of this payout request.</p> <p>The client can use this to differentiate the payout category, for example:</p> <p>1: Merchant_Withdrawal</p> <p>2: Buyer_Withdrawal</p>
Amount mandatory	int64	<p>Payout amount.</p> <p>100,000 times of the real amount.</p> <p>For example, sending 5199000 means \$51.99.</p>

Currency optional	string	<p>The payout currency used in that country.</p> <p>Local payout</p> <p>ID: 'IDR' TH: 'THB'</p> <p>MY: 'MYR' TW: 'TWD'</p> <p>PH: 'PHP'</p> <p>SG: 'SGD'</p> <p>CB payout</p> <p>'USD' : [MY, TW, TH, PH, BR, CO, CL, MX, VN]</p> <p>SGD: SG</p> <p>IDR: ID</p> <p>VND: VN</p>
Client_id mandatory	int	<p>The client_id assigned to the client.</p>
Channel_id optional	int	<p>Only mandatory for CB payout.</p> <p>For local payout, it's optional</p>
Beneficiary mandatory	JSON	<p>The destination account type of the payout.</p> <p>Can be either:</p> <p>[Bank Account] [ID - ShopeePay] [TH - AirPay]</p> <p>[CB-Payout] [Proxy] [QR Code]</p> <ul style="list-style-type: none"> • bank_account • shopeepay_wallet • th_airpay_wallet • cross_broder • proxy • qr_code <p>Please refer to 5.1.1.1 Beneficiary Object for the schema of each beneficiary type.</p>

beneficiary/user_id Mandatory	int	Shopee's user_id. Mandatory, regardless of destination account type.  Note: for clients who are not using Shopee's shared account system, SPM need to disable this user_id field.
Biz_common optional	JSON object	Transparent to SPM. Client can store any JSON object in this dict. SPM will return the same content as-is in callbacks.
Callback_url optional	String	The URL to receive the callback for payout status update.
Callback_mask optional	string array	To instruct SPM not to notify certain payout status changes to client. Mask emun value CREATED SPM_PROCESSING BANK_PROCESSING SUCCESSFUL FAILED
Client_payment_data optional	Object	
client_payment_data/client_remark optional	String (18) Alphanumerics	[Bank Account] Client's remark for the payout. Will be passed on to the Payout Channels. Can be utilised for reconciliation purposes against Bank's records. May contain whitespace.

		Supported channels: ALL
_ts mandatory	int	Current timestamp in second
Extra_data optional	Object	Only use when is spm and country is sg and source_type = 200001014

5.1.1.1. Beneficiary Object

Bank Account Object - **bank_account**

JSON Key	Type	Description
account_id conditional	int	Bank account identifier in SPBA. Required for payout to Bank Account using SPBA flow .
bank_name_id mandatory	int	The destination bank name ID. Required for payout to Bank Account. Full list of Bank_name_ID can be found in the Appendix 2 .
account_number mandatory	string	Destination bank account number. Required for payout to Bank Account
account_name mandatory	string	Bank account owner's name. Required for payout to Bank Account.
branch_id mandatory(TW)	int	The destination bank branch ID.
branch_name	string	The destination bank branch name

optional(VN)		
ic_or_company_id conditional(TW)	string	The ID of an individual or enterprise beneficiary.
full_branch_code mandatory(TW)	string	The compulsory component indicating bank code and branch code in Taiwan. The reason why need upstream client to transmit this info is that regarding some business scenario, SPM cannot get bank code/branch code from SPBA in terms of a specific bank account.
account_info_type conditional(BR)	int	1 - checking account 2 - saving account Required for payout to Bank Account using Pure flow .
tax_id conditional(BR)	string	CPF Id or CNPJ Id Required if account_info_type is not empty
tax_id_type conditional(BR)	int	1 - CPF 2 - CNPJ Required if account_info_type is not empty
branch_info_code conditional(BR)	string	Required if account_info_type is not empty

ID+MY+SG+PH+VN ShopeePay - **shopeepay_wallet**

JSON Key	Type	Description
user mandatory	JSON	
user/id mandatory	int	User_id of the user's Shopee Account
merchant_name mandatory		Mandatory when payout destination is ShopeePay, regardless of source of fund. Will be used to display in ShopeePay FE for Payout From field. To indicate the sender. Should be a String variable, to accept alphabets, numbers, and whitespace.
merchant_id		This parameter is an optional field.

optional		To indicate the merchant_id of the source of fund. Additional note: In the payout received endpoint, merchant_id field will be mandatory when source of fund is Merchant Wallet.
mobile mandatory for vn	string	Phone number format example: TBA

MY+PH+ID `seller_wallet. category: SHOPEE_WALLET_PAYOUT (73)`

JSON Key	Type	Description
<code>seller_wallet</code> mandatory	JSON	
<code>seller_wallet/adjustment_type</code> mandatory	int	TransactionType

TH AirPay - `th_airpay_wallet`

JSON Key	Type	Description
<code>merchant_type</code> mandatory	int	To identify the AirPay's wallet type. 7 -- AirPay buyer wallet 8 -- AirPay seller wallet

Cross border payout - `cross_border`

JSON Key	Type	Description
<code>payee_id</code> mandatory	int	Beneficiary payee id
<code>account_id</code> optional	int	Account id in 3rd party

third_party_id optional	int	Lianlian, Payoneer, Pingpong (different region has different id)
account_number optional	int	Mandatory for withdrawal to BA
account_name optional	String	Payee name

5.1.1.2 client_payment_data object

VN

JSON Key	Type	Description
withdrawl_verified_time mandatory	int	timestamp
user_paid_transaction_fee mandatory	int	User paid transaction fee

5.1.2. Response Parameters

JSON Key	Type	Description
source_id	String	SPM's unique ID for the payout
error_code	int	0 - success 7 - Error_Param (bad request)
error_msg	String	Will only show if error_code is anything but 0
error_retry	int	This field will tell clients how to handle the error. 0 - client should not retry on this error (this field will be 0 if zaiererror_code=0) 1 - client should retry on this error
source_status	int	The payment source status.

		0 - READY -- means the payout has not been processed yet 1 - PENDING -- payout is being processed 2 - DONE -- means the payout has been processed by SPM
--	--	--

Sample Request

POST https://spm-i.test.shopee.co.id/api/shopee_credit/create_payment_source/

```
{
  "reference_id": 1352057712341234,
  "source_type": 3,
  "currency": "IDR",
  "amount": 1200000,
  "_ts": 1675882000,
  "client_id": 80007,
  "callback_url": "https://www.google.com",
  "callback_mask": ["SPM_Processing"],
  "beneficiary": {
    "bank_account": {
      "account_id": 1748,
      "bank_name_id": 1,
      "account_number": "160013344556677",
      "account_name": "JOANITA JOSEPH"
    },
    "biz_common": {
      "anything": "you want"
    }
  },
  "user_id": 1234
}
```

Sample Response

```
{
  "source_id": "12342345",
  "source_status": 0,
  "error_code": 0,
  "error_msg": ""
}
```

5.2. Query Payout

Use this API to query the status of a payout request.

Method	POST
URL	Test: <a href="https://spm.i.test.shopee.co.id/api/<biz_name>/get_payout_details_of_payment_source">https://spm.i.test.shopee.co.id/api/<biz_name>/get_payout_details_of_payment_source UAT: <a href="https://spm.i.uat.shopee.co.id/api/<biz_name>/get_payout_details_of_payment_source">https://spm.i.uat.shopee.co.id/api/<biz_name>/get_payout_details_of_payment_source Live: <a href="https://spm.i.shopee.co.id/api/<biz_name>/get_payout_details_of_payment_source">https://spm.i.shopee.co.id/api/<biz_name>/get_payout_details_of_payment_source
Body	JSON
Authorization	Clients need to use the API credential to sign the request and pass the signature through HTTP header HTTP_AUTHORIZATION. The signature is generated using HMAC_SHA256 algorithm. SPM will generate the key for each client in each environment.

5.1.1. Request Parameters


Request Body:

JSON Key	Type	Description
Reference_id mandatory	uint64	The unique ID of the payout request, generated by the client's system. Same as the payout request.
Source_type	int	The transaction type of this payout request.

mandatory	(0 - 99)	The same as the payout request.
Client_id mandatory	int	The client_id assigned to client
_ts mandatory	int	Current timestamp in second

5.1.2. Response Parameters

JSON Key	Type	Description
payment_source	Object	The payout that is received and processed by SPM.
source_id	string	SPM's unique ID for the payout.
reference_id	uint64	Unique ID of the payout request, generated by the client's system. Will be the same as what client sent during Create Payout.
source_type	Int	The transaction type of this payout request. Will be the same as what client sent during Create Payout.
amount	Int	Amount of the payout. Will be the same as what client sent during Create Payout.
_ts	Int	Current timestamp in second
beneficiary	Object	The destination account type for the payout. Will contain the beneficiary object of the payout destination i.e. Bank account, ShopeePay, or AirPay.
account_id	Int	The bank account ID of this payout.
bank_name_id	Int	The destination bank ID for this payout. Will be the same as what client sent during Create Payout.

account_number	String	The destination bank account number. Will be the same as what client sent during Create Payout.
account_name	String	The destination bank account owner's name. Will be the same as what client sent during Create Payout.
biz_common	Object	Will be the same as what client sent during Create Payout.
payment_detail	Object	Payment detail object.
status	String	The payout status. For details please refer to Appendix 1 . Possible status: <ul style="list-style-type: none"> • CREATED • SPM_PROCESSING • Bank_PROCESSING • SUCCESSFUL • FAILED
Status_change_ts	Int	Timestamp for the latest status change.
channel	String	The payout channel used for this payout request.
arrived_ts	Int	Timestamp that signifies when the fund arrived in the beneficiary's account.  Note: Due to Bank's limitation, the timestamp for interbank channels is not representative of the true time when the fund is reflected.
failure_code	String	Failure code. Only given when payment status is Failed. Please refer to 8.2 Failure Code for a full list of failure_code.
failure_reason	String	Failure reason (message).

Sample Response

```
{
  "payment_source": {
    "source_id": 13433,
```

```

    "reference_id": 13510572,
    "source_type": 3,
    "currency": "MYR",
    "amount": 1200000,
    "_ts": 1575882000,
    "client_id": 20008,
    "beneficiary": {
      "bank_account": {
        "bank_account_id": 15331,
        "bank_name_id": 7,
        "account_number": "160013344556677",
        "account_name": "JOANITA JOSEPH"
      }
    },
    "biz_common": {
      "anything": "you want"
    }
  },
  "payment_detail": {
    "status": "SUCCESSFUL",
    "status_change_ts": 1579882000,
    "channel": "Mandiri-Interbank",
    "arrived_ts": 1579882000,
    "failure_code": "BAC000001",
    "failure_reason": "BANK_INVALID_ACCOUNT"
  }
}

```

5.3. Payout Callbacks

SPM General Payout Service uses callback to notify client's application for two general purposes:

- (1) **Payout confirmation;** This is a step that SPM will do prior to sending the payout request to our Payout Channel (Banks or Third-Party Payment). SPM will only proceed if SPM receives a payout confirmation from the client.
- (2) **Informing payout status change;** General Payout Service will send a callback to client's application whenever payout status changes from one state to another.

5.3.1. Setup

(1) Payout Confirmation (not mandatory)

Clients need to provide an endpoint in your system to receive a callback from SPM for payout confirmation purposes. The callback notification will be sent over HTTP POST request to the callback URL. Client needs to provide this URL to SPM as part of the onboarding requirement.

For detailed information of the definition of this POST request, please refer to [5.3.4. Payout Confirmation](#).

(2) Receiving Payout Status Change


Clients need to provide an endpoint in your system to receive payout status updates from SPM. The callback notification will be sent over HTTP POST request to the callback URL. Clients need to send this callback URL when creating a payout request.

For detailed information of the definition of this POST request, please refer to [5.3.5. Receiving Payout Status Update](#).

5.3.2. Delivery Attempts & Retries

5.3.2.1. Payout Confirmation Callback

For every payout request, SPM will send a Payout Confirmation callback. SPM will only proceed to send this payout request to the payout channel if SPM receives a confirmation from the client.

 Note that client should take this confirmation step to double check the important parameters such as amount, user's status, etc. It is entirely up to client to determine whether to proceed or to fail the payout request based on client's check.

Please refer to 5.3.3. for details of the payload.

Possible outcome from this payout confirmation is as follows:

Callback Response from Client	SPM Handling
Success (confirmed)	SPM to send the payout request to payout channel
Rejected or Fail to Retrieve Confirmation	SPM to fail the payout request

5.3.2.1. Payout Status Change Callback


SPM will send a callback whenever a payout status changes.

If the callback is not acknowledged by the client's system, SPM will retry the callback with exponential delay until SPM receives an acknowledgement from client's side.

Retry Number	Interval (Relative to original callback attempt)
1	5 minutes
2	10 minutes
3	30 minutes
4	60 minutes
5	120 minutes
...	Every 24 hours until callback is acknowledged

5.3.3. Payout Confirmation Callback API Details

SPM will send a POST request to the client, expecting a confirmation on whether to proceed with the payout request.

 **Client needs to provide the endpoint to which SPM will send the callback to as part of the onboarding process.**

The API definition is as follows (note that this API is provided by client, not SPM).

Method	POST
URL	https://client.hostname/any/url/
Body	JSON
Description	<p>Client needs to provide this URL to SPM as part of the onboarding requirement.</p> <p>SPM will send this confirmation callback before the payout is sent to the payout channel. SPM will only proceed with the payout when a confirmation is received from the client.</p>

5.3.3.1. Request Parameters

JSON Key	Type	Description
header	JSON	The response header
payment_source	JSON	The payout that is received and processed by SPM.

header

JSON Key	Type	Description
platform_user_id	String	The shopee uid passed by client when calling create payout beneficiary/user_id of create_payment_source API

payment_source

JSON Key	Type	Description
source_id	Int	SPM's unique ID for the payout.
reference_id	uint64	Unique ID of the payout request, generated by the client's system. reference_id of create_payment_source API
source_type	Int	The transaction type of this payout request. Will be the same as what client sent during Create Payout. source_type of create_payment_source API
currency	String	The currency of the payout currency of create_payment_source API
amount	Int	Amount of the payout. Will be the same as what client sent during Create Payout.
_ts	Int	Current timestamp in second _ts of create_payment_source API
client_id	int	The client_id assigned to client client_id of create_payment_source API
beneficiary	JSON	The destination account type for the payout.

		Will contain the beneficiary object of the payout destination i.e. Bank account, ShopeePay, or AirPay. beneficiary of create_payment_source API
user_id	Int	The shopee uid passed by client when calling create payout beneficiary/user_id of create_payment_source API
biz_common	JSON	Will be the same as what client sent during Create Payout. biz_common of create_payment_source API

Sample Request

```
{
  "header": {
    "platform_user_id": "21234"
  },
  "payment_source": {
    "user_id": 21234,
    "currency": "IDR",
    "_ts": 1604995856,
    "biz_common": {
      "biz_type": 2,
      "biz_country": "id"
    },
    "beneficiary": {
      "bank_account": {
        "account_id": 234123,
        "account_number": 8022141232,
        "account_name": Bambang Pamungkas,
        "bank_name_id": 4
      }
    },
    "source_type": 1,
    "amount": 58800000000,
    "reference_id": 1111111111,
    "client_id": 800222
  }
}
```

Sample Response:

We added a retry mechanism here for some clients which need to retry confirmation. If any clients want to retry Payout Confirmation API, please contact PM.

No need retry:

Expected Response from Client - confirm payout


```
{
  "need_disburse": true,
  "amount": 58800000000
}
```

Expected response from Client - cancel payout

```
{
  "need_disburse": false,
  "amount": 58800000000
}
```

Need to retry:

For the retry mechanism, we will retry three times every 2s, after three times, if still returns to retry, we will set the payment to fail

Expected Response from Client - retry

```
{
  "need_disburse": false,
  "need_retry": true,
  "amount": 58800000000
}
```

Expected Response from Client - confirm payout

```
{
  "need_disburse": true,
  "need_retry": false,
  "amount": 58800000000
}
```

Expected Response from Client - cancel payout

```
{
  "need_disburse": false,
  "need_retry": false,
  "amount": 58800000000
}
```

5.3.4. Payout Status Change Callback

Whenever a payout status changes, SPM will send a POST request to client using the `callback_url` defined when client sends Payout Request API to SPM.


Note that if the client has instructed SPM not to send a callback on specific status via `callback_mask` parameter upon Creating Payout, SPM will not send the callback when the payout changes to that particular status.

Method	POST
URL	https://client.hostname/any/url/?any=1&parameters=1&defined=1&by=1&client=1
Body	JSON
Description	<p>Client should send this callback URL on <code>create_payment_source</code> request, SPM will call this API to update payment status to the client.</p> <p>Client side can include any parameters as needed, SPM will not parse or modify the parameters of this URL.</p>
Header	<p>The callback request will carry the authentication signature in the Header: 'Authorization': \${signature}, Clients(STS) should use the signature to auth the request is legal. The signature uses '{url} {body}' as the plain_text and client_key as the key, generated by the HMAC_SHA256 algorithm</p>

5.3.4.1. Request Parameters

Request Body

JSON Key		Type	Description
payment_source		Object	<p>Contains the same information that client passed during Create Payout step.</p> <p>SPM, however, will exclude the following parameters from payment_source object:</p> <ol style="list-style-type: none">1. callback_url2. callback_mask
payment_detail		Object	Payment details. Contains the status and error reason (if any).
payment_detail/status		String	<p>Status of the payout request.</p> <p>For a complete list of status, please refer to Appendix 1 - General Payout State Machine.</p> <ul style="list-style-type: none">• CREATED• SPM_PROCESSING• BANK_PROCESSING• FAILED• SUCCESS <p>Only when the status is failed and the failure_code starts with BAC is the real transaction failure final state</p>
payment_detail/status_change_ts		Int	The timestamp of the status change.
payment_detail/channel		String	The payout channel that was used for the payout.
payment_detail/arrived_ts		Int	The time when the fund is reflected in the destination account. Only available if the payout is Successful.

			 Note: Due to Bank's limitation, the timestamp for interbank channels is not representative of the true time when the fund is reflected.
payment_detail/failure_code		String	Only available when the payout has failed. For a full list of failure code, please refer to 8.2. Failure Code .
payment_detail/failure_reason		String	Only available when the payout has failed.

Sample Request

```
{
  "payment_source": {
    "source_id": 123423,
    "reference_id": 13510572,
    "source_type": 3,
    "currency": "MYR",
    "amount": 1200000,
    "_ts": 1575882000,
    "client_id": 20008,
    "beneficiary": {
      "bank_account": {
        "bank_account_id": 15331,
        "bank_name_id": 7,
        "account_number": "160013344556677",
        "account_name": "JOANITA JOSEPH"
      }
    }
  },
  "biz_common": {
    "anything": "you want"
  }
},
{
  "payment_detail": {
    "status": "SUCCESSFUL",
    "status_change_ts": 1579882000,
    "channel": "Mandiri-Interbank",
    "arrived_ts": 1579882000,
    "failure_code": "BAC000001",
    "failure_reason": "BANK_INVALID_ACCOUNT"
  }
}
}
```

Expected Response from Client

Client needs to acknowledge upon receiving the Payout Status Change callback from SPM. Otherwise, SPM will retry.

```
{  
  "status": "success"  
}
```

5.4. Exchange Rate API

Get_exchange_rate API is only used for CB payout

5.4.1 Use cases

Basically there are 2 steps for CB seller payout: funding and payout.

1. Funding is for local team to deposit cash with supported currency by 3pp into 3PP account
2. Payout step is triggered by STS and then SPM tells 3pp to transfer from Shopee main wallet to seller wallet.

So both step 1 and 2 have currency exchange flow, step 1 CB finance needs to ask cooperative banks to provide a FX rate to change local currency to i,e USD currency and FX rate is A, step 2 should refer to A to make payout to sellers with USD currency.

5.4.2 Request Parameters

Method	POST
URL	Test: https://spm.i.test.shopee.co.id/ UAT: https://spm.i.uat.shopee.co.id/ Live: https://spm.i.shopee.co.id/
Path	v2/exchange_rate/get/
Body	JSON
Authorization	Clients need to use the API credential to sign the request and pass the signature through HTTP header HTTP_AUTHORIZATION. The signature is generated using HMAC_SHA256 algorithm.

SPM will generate the key for each client in each environment.

Request Body

JSON Key	Type	Description
from_currency	String	from_currency is the currency of the source nation's currency.
to_currency	String	to_currency is the currency of the target nation's currency.
rate_time	Int	The timestamp at currency conversion
exchange_type	Int	exchange_type determines which scenario will be used for the conversion, currently includes the following channels: <ul style="list-style-type: none">1. GENERAL=02. ESUN=13. WECHATPAY=24. STS_CB=35. WECHATPAY_V2=4

Request Sample

```
{
  "from_currency": "MYR",
  "to_currency": "USD",
  "rate_time": 1631006370,
  "exchange_type": 1
}
```

Response Sample

```
{
  "rate": 0.23547
}
```

5.6. Create Refund Payout

Core can create a refund payout instruction to SPM by calling the **create refund payout API**.

This API is designed to be idempotent, which means SPM will return the existing result of the payout request if the client calls this API more than once.

[Note]

When SPM returns "Success" upon the client calling this API, it only guarantees that SPM has successfully received the payout request. The payout may not be processed immediately, as SPM payout processing time is dependent on each Bank's operating hours.

SPM will notify (callback) clients of the payout status asynchronously whenever the payout status changes.

Method	POST
URL	Test: https://spm2.i.test.shopee.co.id/api/create_refund_payment_source/ UAT: https://spm2.i.uat.shopee.co.id/api/create_refund_payment_source Live: https://spm2.i.shopee.co.id/api/create_refund_payment_source
Body	JSON
Authorization	Core need to use the API credential to sign the request and pass the signature through HTTP header HTTP_AUTHORIZATION. The signature is generated using HMAC_SHA256 algorithm. The actual client_id with the secret key is the core written hard in code.
Signature demo	<pre>Golang type RequestParams struct { TS int `json:"_ts"` SourceType int `json:"source_type"` Amount int `json:"amount"` Referenceld int `json:"reference_id"` ClientId int `json:"client_id"` Beneficiary *Beneficiary `json:"beneficiary"` BizCommon string `json:"biz_common,omitempty"` ClientPaymentData string `json:"client_payment_data,omitempty"` } func Signature() string { //clientKey get from client config var (url = "https://spm.i.test.shopee.vn/api/create_refund_payment_source/"</pre>

```


        clientKey =
[]byte("eb2b2877e9d7b997dc632eb59d7705039cbcdf890b47b0aba824ea586
7c32777")
    )
    params := &RequestParams{
        TS: 1624592316,
        Beneficiary: &Beneficiary{
            UserId: 1200070653,
            BankAccount: &BankAccount{
                AccountId: 738,
                BankNameId: 9,
                AccountName: "Fsadfds",
                AccountNumber: "2342232442",
            },
        },
        SourceType: 6,
        Amount: 800000000,
        ReferenceId: 3390706002,
        ClientId: 50002,
    }
    paramsJson, _ := json.Marshal(params)
    baseString := fmt.Sprintf("%s|%s", url, string(paramsJson))
    h := hmac.New(sha256.New, clientKey)
    h.Write([]byte(baseString))
    return hex.EncodeToString(h.Sum(nil))
}

```


5.6.1. Request Parameters

Request Body:

JSON Key	Type	Description
reference_id mandatory	uint64	Unique ID of the payout request, generated by the client's system. Used to reconcile payouts between SPM & client. Note: <ul style="list-style-type: none">4. Client side to make sure this ID is unique for each payout request sent to SPM5. Client side to make sure this ID is unique for each source_type
Source_type mandatory	int	The transaction type of this payout request. Client can use this to differentiate payout category, for example: 1: Merchant_Withdrawal 2: Buyer_Withdrawal
Amount mandatory	int64	Payout amount. 100,000 times of the real amount. For example, sending 5199000 means \$51.99.
Currency optionalmandatory	string	The payout currency used in that country. Local payout ID: 'IDR' TH: 'THB' MY: 'MYR' TW: 'TWD' PH: 'PHP' SG: 'SGD' CB payout 'USD' : [MY, TW, TH, PH, BR, CO, CL, MX, VN]

		<p>SGD: SG</p> <p>IDR: ID</p> <p>VND: VN</p>
Client_id mandatory	int	The client_id assigned to the client.
Beneficiary mandatory	JSONObject	<p>The destination account type of the payout.</p> <p>Can be either:</p> <p>[Bank Account] [ID - ShopeePay]</p> <ul style="list-style-type: none"> • bank_account • shopeepay_wallet <p>Please refer to 5.1.1.1 Beneficiary Object for the schema of each beneficiary type.</p>
beneficiary/user_id Mandatory	int	<p>Shopee's user_id.</p> <p>Mandatory, regardless of destination account type.</p> <p> Note: for clients who are not using Shopee's shared account system, SPM need to disable this user_id field.</p>
Callback_url mandatory	String	The URL to receive the callback for payout status update.
Callback_mask optional	string array	<p>To instruct SPM not to notify certain payout status changes to client.</p> <p>Mask emun value</p> <p>CREATED</p> <p>SPM_PROCESSING</p> <p>BANK_PROCESSING</p> <p>SUCCESSFUL</p>

		FAILED
_ts mandatory	int	Current timestamp in second
Extra_data mandatory	Object	Core's payment_id is stored in (target_payment_id)

5.6.1.1. Beneficiary Object

Bank Account Object - **bank_account**

JSON Key	Type	Description
account_id conditional	int	Bank account identifier in SPBA. Required for payout to Bank Account using SPBA flow .
bank_name_id mandatory	int	The destination bank name ID. Required for payout to Bank Account. Full list of Bank_name_ID can be found in the Appendix 2 .
account_number mandatory	string	Destination bank account number. Required for payout to Bank Account
account_name mandatory	string	Bank account owner's name. Required for payout to Bank Account.
branch_id mandatory(TW VN)	int	The destination bank branch ID.
branch_name	string	The destination bank branch name

optionalmandatory(VN)		
ic_or_company_id conditional(TW)	string	The ID of an individual or enterprise beneficiary.
full_branch_code mandatory(TW)	string	The compulsory component indicating bank code and branch code in Taiwan. The reason why need upstream client to transmit this info is that regarding some business scenario, SPM cannot get bank code/branch code from SPBA in terms of a specific bank account.
account_info_type conditional(BR)	int	1 - checking account 2 - saving account Required for payout to Bank Account using Pure flow .
tax_id conditional(BR)	string	CPF Id or CNPJ Id Required if account_info_type is not empty
tax_id_type conditional(BR)	int	1 - CPF 2 - CNPJ Required if account_info_type is not empty
branch_info_code conditional(BR)	string	Required if account_info_type is not empty

ID+MY+SG+PH+VN ShopeePay - **shopeepay_wallet**

JSON Key	Type	Description
user mandatory	JSON Object	
user/id mandatory	int	User_id of the user's ShopeePay account
merchant_name mandatory		Mandatory when payout destination is ShopeePay, regardless of source of fund. Will be used to display in ShopeePay FE for Payout From field. To indicate the sender. Should be a String variable, to accept alphabets, numbers, and whitespace.

merchant_id optional		<p>This parameter is an optional field. To indicate the merchant_id of the source of fund.</p> <p>Additional note:</p> <p>In the payout received endpoint, merchant_id field will be mandatory when source of fund is Merchant Wallet.</p>
mobile mandatory for vn	string	VN shopeepay wallet need mobile to do precheck
airpay_uid mandatory for vn	int	

5.6.2. Response Parameters

JSON Key	Type	Description
source_id	String	SPM's unique ID for the payout
error_code	int	<p>0 - success</p> <p>7 - Error_Param (bad request)</p>
error_msg	String	Will only show if error_code is anything but 0
error_retry	int	<p>This field will tell clients how to handle the error.</p> <p>0 - client should not retry on this error (this field will be 0 if error_code=0)</p> <p>1 - client should retry on this error</p>
source_status	int	<p>The payment source status.</p> <p>0 - READY -- means the payout has not been processed yet</p> <p>1 - PENDING -- payout is being processed</p> <p>2 - DONE -- means the payout has been processed by SPM</p>

Error code that can be set to fail

Error Code Description	Value
------------------------	-------

ERROR_CREATE_PAYMENT_SOURCE	7
-----------------------------	---

Other error codes need to initiate a retry

Sample Request

POST https://spm2.i.test.shopee.co.id/api/shopee_credit/create_payment_source/

```
{
  "reference_id": 1003940109,
  "source_type": 200009999,
  "currency": "IDR",
  "amount": 1200000,
  "_ts": 1675882000,
  "client_id": 80007,
  "callback_url": "https://www.google.com",
  "callback_mask": ["SPM_PROCESSINGrocessing"],
  "beneficiary": {
    "user_id": 1218320093,
    "bank_account": {
      "account_id": 1748,
      "bank_name_id": 1,
      "account_number": "160013344556677",
      "account_name": "JOANITA JOSEPH"
    },
    "extra_data": {
      "target_payment_id": 10102342352353,
    }
  }
}
```

Sample Response

```
{
  "source_id": "12342345",
  "source_status": 0,
  "error_code": 0,
  "error_msg": ""
}
```

5.7. Query Refund Payout

Use this API to query the status of a payout request.

Method	POST
URL	Test: https://spm2.i.test.shopee.co.id/api/get_refund_payout_details_of_payment_source UAT: https://spm.i.uat.shopee.co.id/api/get_refund_payout_details_of_payment_source Live: https://spm.i.shopee.co.id/api/get_refund_payout_details_of_payment_source
Body	JSON
Authorization	<p>Core need to use the API credential to sign the request and pass the signature through HTTP header HTTP_AUTHORIZATION.</p> <p>The signature is generated using HMAC_SHA256 algorithm.</p> <p>The actual client_id with the secret key is the core written hard in code.</p>

5.21.1. Request Parameters

Request Body:

JSON Key	Type	Description
Reference_id mandatory	uint64	The unique ID of the payout request, generated by the client's system. Same as the payout request.
Source_type mandatory	int (0 - 99)	The transaction type of this payout request. The same as the payout request.
Client_id mandatory	int	The client_id assigned to client
_ts mandatory	int	Current timestamp in second

5.21.2. Response Parameters

JSON Key	Type	Description
header	JSON	The response header
payment_source	JSON	The payout that is received and processed by SPM.
payment_detail	JSONObject	Payment detail object.

header


JSON Key	Type	Description
platform_user_id	String	The shopee uid passed by client when calling create payout

payment_source

JSON Key	Type	Description
source_id	String	SPM's unique ID for the payout.
reference_id	uint64	Unique ID of the payout request, generated by the client's system. Will be the same as what client sent during Create Payout.
source_type	Int	The transaction type of this payout request. Will be the same as what client sent during Create Payout.
currency	String	The currency of the payout
amount	Int	Amount of the payout. Will be the same as what client sent during Create Payout.
_ts	Int	Current timestamp in second
client_id	int	The client_id assigned to client
beneficiary	JSON	The destination account type for the payout.

		Will contain the beneficiary object of the payout destination i.e. Bank account, ShopeePay, or AirPay.
user_id	Int	The shopee uid passed by client when calling create payout
biz_common	JSON	Will be the same as what client sent during Create Payout.

payment_detail

JSON Key	Type	Description
status	String	<p>The payout status. For details please refer to Appendix 1.</p> <p>Possible status:</p> <ul style="list-style-type: none"> • CREATED • SPM_PROCESSING • Bank_PROCESSING • SUCCESSFUL • FAILED
status_change_ts	Int	Timestamp for the latest status change.
channel	String	The payout channel used for this payout request.
arrived_ts	Int	<p>Timestamp that signifies when the fund arrived in the beneficiary's account.</p> <p> Note: Due to Bank's limitation, the timestamp for interbank channels is not representative of the true time when the fund is reflected.</p>
failure_code	String	<p>Failure code. Only given when payment status is Failed.</p> <p>Please refer to 8.2 Failure Code for a full list of failure_code.</p>
failure_reason	String	Failure reason (message).

beneficiary/bank_account

JSON Key	Type	Description
bank_account_id	Int	The bank account ID of this payout.
bank_name_id	Int	<p>The destination bank ID of this payout.</p> <p>Will be the same as what client sent during Create Payout.</p>

bank_name	String	The destination bank name of this payout.
account_number	String	The destination bank account number. Will be the same as what client sent during Create Payout.
account_name	String	The destination bank account owner's name. Will be the same as what client sent during Create Payout.

Sample Response

```
{
  "header": {
    "source_id": "1213874827"
  },
  "payment_source": {
    "source_id": 13433,
    "reference_id": 13510572,
    "source_type": 3,
    "user_id": 1213874827,
    "currency": "MYR",
    "amount": 1200000,
    "_ts": 1575882000,
    "client_id": 20008,
    "beneficiary": {
      "bank_account": {
        "bank_account_id": 15331,
        "bank_name_id": 8907,
        "bank_name": "MariBank",
        "account_number": "160013344556677",
        "account_name": "JOANITA JOSEPH"
      }
    }
  },
  "biz_common": {
    "anything": "you want"
  }
},
  "payment_detail": {
    "status": "SUCCESSFUL",
    "status_change_ts": 1579882000,
    "channel": "Mandiri-Interbank",
    "arrived_ts": 1579882000,
    "failure_code": "BAC000001",
    "failure_reason": "BANK_INVALID_ACCOUNT"
  }
}
```

```
}  
}
```

6. SPBA APIs

Pre-requisite: Client need to register themselves to SPEX API SPACE platform to be able to consume the SPBA SPEX API.

Shopee Bank Account system (SPBA), is a system that stores bank account details of a beneficiary. As a prerequisite to utilise SPBA, client needs to use the Shopee User Account system (either toB or toC account).

The SPBA APIs enable client to:

- Add new bank account details by userID
- Get bank account or list of bank accounts by userID
- Set default bank account on SPBA
- Update bank account with rejected status
- Delete bank account by userID

For the full list of services and most updated SPBA API documentation, please refer to [SPBA SPEX API](#) (you will need access to Shopee Space Platform to view).

6.1. Bank Account Name Validation Service

Prerequisite: Payout Flow using SPBA

Bank Account Name Validation Services enables you to check against the Bank's database to validate information before making the actual payout, for example, checking that you are sending funds to the right person by checking the name of the account owner.

This service adds an additional layer of validation on top of SPBA bank account validation.

Validation Flow

If a client requires a fund to only be paid out to the actual account owner, SPM will validate the bank account details (name & account number) in SPBA against the Bank's database. This validation takes place upon SPM receiving the payout request from the client, and SPM will only proceed with the payout request if the Bank Account Name Validation returns a positive result (i.e. account name is verified).

Output of The Validation Process

The 3 possible outputs are:

1. Success via Auto-Validation logic that SPM hosts

If there is a match between the bank account name returned from Mandiri API and Shopee's bank account page, the output will be VERIFIED.

2. Success via Manual-Validation process that FinOps performs

Mismatched names from auto-validation will return PENDING, and a [report](#) will be generated in SPM. These reports will be manually verified by the FinOps team, by calling the Xendit API.

**Note: Client will have to arrange this with the FinOps assigned for client's use-case.*

3. Fail to validate

Bank accounts which fail to be manually verified by FinOps team will return REJECTED. The payout will be failed by SPM as well.

6.2. Bank Account Name Validation Results Update

For the bank accounts that have to go through a manual validation process by the FinOps team, FinOps personnel will need to gain access to SPM Admin portal and have the access privilege to Operation Flow to update bank account name validation results.

FinOps will need to perform the operation flow on a daily basis, to move any bank account status with name validation status as PENDING to either VERIFIED or REJECTED.

7. Reports

7.1. Payout Channel Reports

To help clients with reconciliation, SPM will generate Payout Channel reports that contain all the payouts that SPM processed on behalf of clients.

The payout channel reports are generally generated on a daily basis (D+1), and will contain the list of payouts that have reached final statuses (Success/Failed).

Payout channel reports are generated per payout channel. Note that the payout channel will consist of other clients' transactions as well. You can filter your transactions by client_id which SPM has assigned for your system.

See [Appendix 3](#) for list of payout channels.

7.2. Reports Retrieval

To retrieve the payout channel reports, your users will need to apply for access to SPM Admin portal for respective countries. You can guide your users to apply for SOUP permissions to SPM Admin.

Once access is granted, your users will be able to login to SPM Admin and go to **Reporting > Financial Report** and find for the relevant Payout Channel reports.

Payout channel reports filename: {PayoutChannel}_Payout_Financial_Report_YYYYMMDD

Example: BCA_Payout_Financial_Report_20200317.csv

7.3. Reports Information

The following information will be provided in the payout channel reports:

Column	Description	Example
referral_no	Common key between SPM and channels (e.g. banks). Your operations team can use this column for reconciliation against bank statements.	1611981895635240310

provision_type	Provision Type (SPM internal)	OUTBOUND_GENERAL
user_id	user_id of the beneficiary (if client is using Shopee Account System)	152411867
entity_id	SPM's unique ID for the payout. Same as payment_source_id.	31486165
payee_account_number	Destination account number	848251462
amount	Payout amount	611925.0 IDR
payment_status	Status of payout	SUCCESSFUL <i>See Appendix 1 for full list of status</i>
payout_datetime	Datetime payout was created	2021-01-25 20:29:10
payment_source_type	The payout type Unique to client	SELLER_WITHDRAWAL MERCHANT_WITHDRAWAL
client_id	Client ID. Each client is assigned a unique ID by SPM.	80002
payment_source_id	SPM's unique ID for the payout	31486165
payment_source_reference_id	Client's unique ID for the payout. Should be mapped to client's payout request e.g. withdrawal, disbursement, refund, etc. Reconciliation key between SPM & Client's system.	173991792

7.2. SSG Reports

When there is a payout that is blocked by SSG, Safeguard will send out an email to the recipient list that's configured via SPM admin.

Column Name	Description	Example
SourceID	SSG's source ID. Unique running number from SSG's side.	34534124

SourceRefID	SPM's payment source reference ID. Essentially the reference ID that is passed by client upon making a payout request.	2605167591030031100
UserID	The userID of the beneficiary.	8618235
Source Type	The source type of this payout.	Refund Withdrawal etc
Source Status	The SSG source status.	RESET BLOCKED
Source Reason Code	Explains the reason why payout is blocked by SSG.	Duplicate_ref_id
Block Type	Explains whether a blocked payout is definitive or unconfirmed (need Ops to verify).	wrong_block unconfirmed_block
cTime	Source created time.	2020-11-30 11:06:22
mTime	Source last modified time.	2020-11-30 14:25:28
Extra Data	Contain extra data objects.	{"reset_remark":"ok ","from_status":2,"reset_operator":"system"}

7.3. Bank Account Name Validation Status Reports

Column Name	Description	Example
bank_account_id	Bank account ID (stored in SPBA) of payout destination.	814897
bank_account_name	Bank account holder's name (from SPBA). This will be used to run the auto-validation logic in SPBA.	Peter Tan

bank_account_name_verified_status	Status of verification as returned by SPBA/ FinOps.	<ul style="list-style-type: none"> ● VERIFIED: Successfully verified ● PENDING: Unsuccessful verification from SPBA, pending verification result from FinOps ● REJECTED: Unsuccessful manual verification as determined by FinOps
bank_account_number	Bank account number (from SPBA) of payout destination.	50533361
bank_account_status	Bank account status stored in SPBA. Only those which are "CHECKED" OR "VERIFIED" will be go through bank account name validations and if successful, be eligible for payouts	<ul style="list-style-type: none"> ● CHECKED: Bank account length is valid, but no payout has been made ● VERIFIED: BA is valid and a payout has been made before ● REJECTED: Previous payout failed, BA is ineligible for payout ● BANNED: Potential fraud case, ineligible for payout ● DELETED: deleted BA, ineligible for payout
bank_name	Name of destination bank account for payout.	BCA Refer to Appendix 2 for full list of bank names
source_ctime	Time at which payout request was received by SPM.	2021-01-11 09:08:03+07:00
source_ref_id	ID of payout request (from client-side).	1348622157828530176

8. Errors

8.1 Error code

Error codes will be shown when there is error in the API call (e.g. bad request, invalid param).

Error	Code	Description	Suggested Client Handling
ERROR_NOT_FOUND	2	Payment source not found. Requesting client_id not found. You may encounter this when querying or create the payout.	Ensure the data entered e.g. payment_source_id is correct. Ensure that you (client) have the sufficient permission to access this API. Client keeps querying or manual checking. 1. Dev suggests Clients also can map to failure if still not found over 2 days.
ERROR_CREATE_PAYMENT_SOURCE	7	Bad request. Failed to create payment source on the SPM side.	Please review & input a valid value in the parameter(s) that returned error. Clients can map to failure.

8.2 Failure Code

Failure codes will only be available when a payout has failed.

Failure Code Category	Description	Client Handling Logic
SYSXXXXXX	SPM internal server error.	Please contact the Development team to handle this error.
SSGXXXXXX	Payout is blocked by safeguard because of the high risk of double payout.	Payout that is blocked by SSG will not be processed by SPM anymore. Client should not retry the payout upon receiving this error code.

		<p>FinOps will be informed of any SSG_Blocked cases via report to their email. Subsequently, FinOps can help to reset & retry the payout if they deem the payout to be wrongly blocked.</p> <p>Note that there is an inherent risk in doing so, and clients should proceed with caution.</p>
BACXXXXXX	<p>The payout transaction is rejected by the payout channel.</p> <p>This can be caused by invalid account information, invalid payout channel, etc.</p>	<p>A payout with this failure code should be deemed as a failed payout on the client's side.</p> <p>Clients can create a new payout request by using a new payout reference_ID and new destination account details (make sure the destination account is valid).</p>

[Failure reason describes the detailed failure reason.](#)

Below only describe part of the code.

Failure Code	Description	Client Handling Logic
SYS000001	Internal server error	Contact the SPM development team to handle this error.
SYS000002	Insufficient balance	<p>There is insufficient balance in the client's bank account to make the disbursement in the desired amount.</p> <p>Clients may request another payout after ensuring that they have sufficient balance in their account.</p>

SYS000003	Bank information (account_number, account_name) does not match with SPBA for the particular bank_account_id.	Clients should create a new payment_source and ensure the bank account details are correct.
SYS000004	Error when sending confirmation request to client side (because of either client side error or network error)	<p>Payout will be FAILED status. SPM will not send the payout request to channel.</p> <p>Client may create another request.</p>
SYS000005	Payment retry is not allowed	<p>Due to payment request issues, retrying is not allowed.</p> <p>Client can create a new payout request if it's deemed safe to do so (after checking with FinOps/Bank statement).</p>
SYS000006	Invalid client remark	<p>Ensure client remark value is valid.</p> <p>Notice the length limit & special character limitations.</p> <p>Review the remarks before retrying the payout.</p>
SSG000002	Get error when validating payment from SSG	<p>Clients should not retry in this case.</p> <p>SPM proposes map the status to be pending</p>
BAC000001	Invalid bank account	<p>The bank account provided by the user is invalid.</p> <p>A payout with this failure code should be deemed as a failed payout on the client's side.</p>

		Clients can create a new payout request by using a new payout reference_ID and new bank account details.
BAC000002	Client using invalid payout channel	Client is not allowed to use this particular payout channel.
BAC000004 AirPay Wallet	Invalid wallet	<p>The wallet provided by the user is invalid.</p> <p>Clients can create new payout requests by using a new payout reference_ID and valid wallet details.</p>
BAC000005	Failed due to the client telling SPM not to proceed with the payout during the Payout Confirmation stage.	<p>Payout failed.</p> <p>Clients may create new payout requests using valid details.</p>
BAC000006	Rejected by the payout provider (bank side or payout service)	<p>Bank-side error.</p> <p>Clients may create new payout requests by passing a new reference_id to SPM.</p> <p>Clients are advised to use new destination bank information to increase the chance of successful transactions.</p>
BAC000007 ShopeePay	Invalid beneficiary account status	<p>Payout failed.</p> <p>Clients may create new payout requests using valid details.</p>
BAC000008 ShopeePay	Insufficient beneficiary limit	<p>Payout failed.</p> <p>Clients may create new payout requests using valid details.</p>
BAC000100	Kyc balance limit	Payout failed.

		Clients may create new payout requests using valid details.
BAC000101	No kyc balance limit	<p>Payout failed.</p> <p>Clients may create new payout requests using valid details.</p>
BAC000102	Wallet inflow monthly limit	<p>Payout failed.</p> <p>Clients may create new payout requests using valid details.</p>
BAC000110	Invalid wallet account	<p>Payout failed.</p> <p>Clients may create new payout requests using valid details.</p>
BAC000200	Payee account was closed	<p>Payout failed.</p> <p>Clients may create new payout requests using valid details.</p>
BAC000201	Invalid bank account name	<p>Payout failed.</p> <p>Clients may create new payout requests using valid details.</p>
BAC000202	Invalid bank account number	<p>Payout failed.</p> <p>Clients may create new payout requests using valid details.</p>
BAC000203	Account info mismatched	<p>Payout failed.</p> <p>Clients may create new payout requests using valid details.</p>
BAC000300	Rejected by 3PP due to incorrect transaction info	<p>Payout failed.</p> <p>Clients may create new payout requests using valid details.</p>

BAC000301	Rejected by 3PP due to transaction not found	Payout failed. Clients may create new payout requests using valid details.
BAC000302	Rejected by 3PP due to wrong currency	Payout failed. Clients may create new payout requests using valid details.
BAC000303	Rejected by 3PP due to wrong bank info	Payout failed. Clients may create new payout requests using valid details.
BAC000400	Rejected by 3PP due to insufficient balance	Payout failed. Clients may create new payout requests using valid details.
BAC000500	Rejected by 3PP due to inactive account	Payout failed. Clients may create new payout requests using valid details.
BAC000502	Rejected by 3PP due to no available service	Payout failed. Clients may create new payout requests using valid details.
BAC000503	Rejected by 3PP due to bank system maintenance	Payout failed. Clients may create new payout requests using valid details.
BAC000600	Amount exceed the maximum amount	Payout failed. Clients may create new payout requests using valid details.
BAC000601	Amount is lower than the minimum amount	Payout failed.

		Clients may create new payout requests using valid details.
BAC000602	Amount mismatch	<p>Payout failed.</p> <p>Clients may create new payout requests using valid details.</p>
BAC000700	Payout rejected by client	<p>Payout failed.</p> <p>Clients may create new payout requests using valid details.</p>
BAC000701	Request canceled by SPM	<p>Payout failed.</p> <p>Clients may create new payout requests using valid details.</p>
BAC000702	Request canceled by bank	<p>Payout failed.</p> <p>Clients may create new payout requests using valid details.</p>

9. Payout Testing Integration

Before taking your payout integration live, clients should test their integration thoroughly in the test/UAT environment. Integration testing should not only include positive scenarios, but should also include negative scenarios to ensure proper error handling is in place.

Scenarios for Specific Responses & Errors

Indonesia

Scenarios	Parameter(s)	Value
Successful payout	bank_name_id amount	2 (BCA) 5000000000 (IDR 50,000)
Invalid bank account BAC000001	bank_name_id amount	2 (BCA) 5000100000 (IDR 50,001)
Bank-side error BAC000006	bank_name_id amount	3 (BNI) 5000200000 (IDR 50,002)
Payout confirmation failed BAC000005	need_disburse	False
Insufficient balance SYS000002	bank_name_id amount	2 (BCA) 5000800000 (IDR 50,008)
Unable to retrieve payout confirmation SYS000004	Client to return unexpected response upon payout confirmation	{'error': 'xxx'}
Error_Param	client_remark	Length > 18
Duplicate reference_id	reference_id	Send a reference_id that has been received by SPM previously

Note on Integration testing

Note that this list of scenarios provided by SPM is not exhaustive and clients should determine the level of testing comprehensiveness that clients deems appropriate when integrating with SPM General Payout Flow.

Beyond the provided scenarios, clients should perform the proper due diligence, test edge cases, incomplete data, invalid data, and duplicate data (e.g. retry the exact same request).

If there are uncovered scenarios that clients would like to cover in their test, please contact the developer/PM for assistance to simulate the scenario in the test/UAT environment.

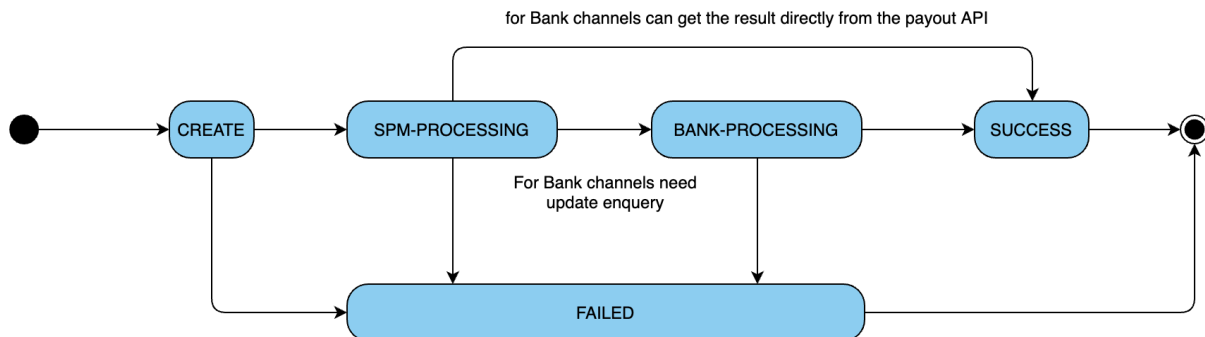
10. Go-Live Checklist

After conducting all your testing, clients should perform the following steps before going live.

- Simulate negative test scenarios, and ensure error handling measure are in place:
 - Duplicate reference_id
 - Invalid bank account
 - Payout confirmation failed
- Ensure code is properly written to handle various error types.
- Ensure that you have received the API keys for Live environment.
- Ensure that you have correctly set up the Callback URL in the Live environment.
- Ensure that your originating bank account details are correct.
 - If a client is using the client's own bank account, please ensure that the Banks are informed and SPM has the access to use the client's originating bank account as the source of funds for the client's payout.
 - If a client intends to use SPM's existing account, please ensure that finance & relevant stakeholders are informed of this plan.
- Have a log for all requests and corresponding status.

Appendix

Appendix 1 - General Payout State Machine



Status	Value	Description
CREATE	0	SPM has received the payout request from client.
SPM_PROCESSING	1	SPM has started processing the payment.
BANK_PROCESSING	2	The payment request has been sent to the Payout Channel for processing.
SUCCESS	3	The beneficiary bank account has received the payment. (Note that the beneficiary bank account may receive the payment before the status changes to SUCCESS. This is because of the different designs of banking systems).
FAILED	4	The payout has failed.

Appendix 2 - Supported Countries & Destination Accounts

Indonesia

Bank Name	SPM Bank Name ID	Valid Length
Bank Mandiri	1	[10,11,12,13,14,15,16]
BCA	2	[10,17]
BNI	3	[7,8,9,10,16]
BRI	4	[14, 15]
CIMB Niaga	6	[12,13,16]
CITIBANK	7	[10, 15, 16, 17, 18]
Danamon	8	[8, 10, 11, 12, 15, 16, 17, 18]
HSBC	9	[9, 10, 11, 12, 13, 14, 15, 16, 17, 18]
BTN	10	[10,13,15,16,17,18]
Bukopin	11	[10, 13, 15, 16, 17, 18]
Mega	12	[15, 16, 17, 18]
Bangkok Bank Public CO	13	[13, 15, 16, 17, 18]
Bank Hana	14	[11,15,16]
BJB Syariah	15	[13, 15, 16, 17, 18]
Bank Jatim Unit Usaha Syariah	16	[10, 15, 16, 17, 18]
Bank Of America	17	[15, 16, 17, 18]
Bank Of China Limited	18	[13, 15, 16, 17, 18]
BPD Jatim	19	[10, 15, 16, 17, 18]
BPD Kalsel	20	[10, 13, 15, 16, 17, 18]
BPD Kalsel UUS	21	[13, 15, 16, 17, 18]

BPD Kaltim UUS	22	[10, 15, 16, 17, 18]
BPD Lampung	23	[13, 15, 16, 17, 18]
BPD NTT	24	[14, 15, 16, 17, 18]
BPD Sumbar UUS	26	[14, 15, 16, 17, 18]
BPD Sumsel Dan Babel	27	[10, 11, 15, 16, 17, 18]
BPD DIY	28	[12, 15, 16, 17, 18]
Deutsche Bank AG	29	[10, 15, 16, 17, 18]
JPMorgan Chase Bank NA	32	[10, 15, 16, 17, 18]
Bank Raya Indonesia	33	[15, 16, 17, 18]
ANZ Panin Bank	34	[15, 16, 17, 18]
Bank Agris	36	[10, 15, 16, 17, 18]
Bank OKE Indonesia	37	[10, 15, 16, 17, 18]
Bank Antar Daerah	38	[10, 15, 16, 17, 18]
Bank Artha Graha INT	39	[10, 15, 16, 17, 18]
Bank Jago	40	[12, 15, 16, 17, 18]
BCA Syariah	41	[10]
Bank Bisnis International	42	[15, 16, 17, 18]
BPD Aceh UUS	44	[14, 15, 16, 17, 18]
Bank Bumi Arta	45	[10, 15, 16, 17, 18]
Bank Capital Indonesia	46	[12, 13, 15, 16, 17, 18]
Bank China Trust Indonesia	47	[15, 16, 17, 18]
Cimb Niaga UUS	48	[12,13]
Bank Commonwealth	49	[10, 15, 16, 17, 18]

Bank Danamon IND. UU Syariah	50	[12, 15, 16, 17, 18]
Bank DBS Indonesia	51	[9, 10, 15, 16, 17, 18]
Bank Sahabat Sampoerna	52	[10, 15, 16, 17, 18]
Bank DKI	53	[10, 11, 12, 15, 16, 17, 18]
Bank DKI UUS	54	[10, 11, 12, 15, 16, 17, 18]
Bank HSBC Indonesia	55	[12, 15, 16, 17, 18]
Bank Fama International	56	[10, 15, 16, 17, 18]
Bank Ganesha	57	[11, 15, 16, 17, 18]
Allo Bank	58	[10, 15, 16, 17, 18]
Bank ICBC Indonesia	60	[15, 16, 17, 18, 19]
Bank INA Perdana	61	[10, 11, 12, 15, 16, 17, 18]
Bank Index Selindo	62	[10, 15, 16, 17, 18]
BJB	63	[13, 15, 16, 17, 18]
Bank Jasa Jakarta	64	[12, 15, 16, 17, 18]
Bank Kalbar UUS	65	[12, 15, 16, 17, 18]
Bank KEB Hana Indonesia	66	[11, 15, 16, 17, 18]
Bank Maspion Indonesia	70	[10, 15, 16, 17, 18]
Bank Mayapada International T	71	[11, 15, 16, 17, 18]
Bank Aladin Syariah	72	[10, 11, 15, 16, 17, 18]
Bank Mayora Indonesia	73	[10, 15, 16, 17, 18]
Bank Mestika Dharma	74	[11, 15, 16, 17, 18]
Bank Shinhan Indonesia	75	[15, 16, 17, 18]
Bank Mitraniaga	76	[10, 15, 16, 17, 18]

Bank Mizuho Indonesia	77	[10, 15, 16, 17, 18]
Bank Muamalat Indonesia	78	[10, 15, 16, 17, 18]
Bank Multiarta Sentosa	79	[10, 12, 15, 16, 17, 18]
Bank Jtrust Indonesia	80	[10, 15, 16, 17, 18]
Bank National NOBU	81	[11, 15, 16, 17, 18]
Bank Nusantara Parahyangan	83	[11, 15, 16, 17, 18]
Bank OCBC NISP	84	[12, 15, 16, 17, 18]
Bank OCBC NISP Syariah	85	[12, 15, 16, 17, 18]
BPD Jambi	88	[9, 10, 15, 16, 17, 18]
BPD Papua	89	[13, 15, 16, 17, 18]
Bank Permata	90	[6, 7, 8, 9, 10, 15, 16, 17, 18]
Bank Permata Syariah	91	[10,15,16,17,18]
Bank Pembangunan Daerah Banten (BPDB)	92	[10, 15, 16, 17, 18]
RABOBANK International IND	93	[10, 15, 16, 17, 18]
Bank Resona Perdania	94	[11, 15, 16, 17, 18]
BCA Digital	95	[11, 12]
Bank SBI Indonesia	97	[15, 16, 17, 18]
Bank Mandiri Taspen	98	[13, 15, 16, 17, 18]
Bank Sinarmas	99	[10, 15, 16, 17, 18]
Bank Sinarmas Syariah	100	[10, 15, 16, 17, 18]
Bank Sulselbar	101	[15, 16, 17, 18]
Bank Sulselbar Syariah	102	[15, 16, 17, 18]

Bank Sulut	103	[14, 15, 16, 17, 18]
Bank Sumitomo Mitsui Indonesia	104	[15, 16, 17, 18]
BPD Sumut	105	[14, 15, 16, 17, 18]
Bank Swadesi	106	[10, 15, 16, 17, 18]
Bank Syariah Bukopin	108	[10, 15, 16, 17, 18]
Bank Syariah Mega Indonesia	110	[10, 15, 16, 17, 18]
BTPN	111	[11, 12, 15, 16, 17, 18]
BTPN Syariah	112	[10, 15, 16, 17, 18]
Bank UOB Indonesia	113	[10, 15, 16, 17, 18]
Bank Victoria International	114	[10, 15, 16, 17, 18]
Bank Victoria Syariah	115	[10, 15, 16, 17, 18]
Bank Windu Kentjana	116	[10, 15, 16, 17, 18]
Bank Woori Indonesia	117	[12, 15, 16, 17, 18]
Bank Neo Commerce	118	[10, 11, 12, 13, 14, 15, 16, 17, 18]
BNP PARIBAS Indonesia	121	[15, 16, 17, 18]
BPD Aceh	122	[14, 15, 16, 17, 18]
BPD Bali	123	[13, 15, 16, 17, 18]
BPD Bengkulu	124	[13, 15, 16, 17, 18]
BPD DIY Syariah	125	[12, 15, 16, 17, 18]
BPD Jateng	126	[10, 15, 16, 17, 18]
BPD Jateng UUS	127	[10, 15, 16, 17, 18]
BPD Kalbar	128	[10, 15, 16, 17, 18]
BPD Kaltim	129	[10, 15, 16, 17, 18]

BPD Kalteng	130	[13, 15, 16, 17, 18]
BPD Maluku	131	[10, 15, 16, 17, 18]
BPD NTB	132	[13, 15, 16, 17, 18]
BPD Riau	133	[10, 15, 16, 17, 18]
BPD Sulteng	135	[13, 15, 16, 17, 18]
BPD Sultra	136	[14, 15, 16, 17, 18]
BPD Sumsel dan BABEL Syariah	137	[10, 15, 16, 17, 18]
BPD Sumut Syariah	138	[14, 15, 16, 17, 18]
BTN UUS	139	[10, 15, 16, 17, 18]
Centratama Nasional Bank	140	[15, 16, 17, 18]
Prima Master Bank	142	[15, 16, 17, 18]
Standard Chartered Bank	143	[11, 15, 16, 17, 18]
MUFG Bank	144	[15, 16, 17, 18]
Maybank	147	[10,16]
Bank Nagari	148	[14, 15, 16, 17, 18]
Bank Raya Indonesia	149	[15]
ANZ Panin	150	[15, 16, 17, 18]
Bank QNB Indonesia	151	[13, 15, 16, 17, 18]
Bank Mayapada International	152	[11, 15, 16, 17, 18]
Bank Panin	153	[10, 15, 16, 17, 18]
Bank Panin Syariah	154	[10, 15, 16, 17, 18]
Bank MNC Internasional	155	[15, 16, 17, 18]
SeaBank	156	[12, 15, 16, 17, 18]

Bank Syariah Indonesia	157	[10, 15, 16, 17, 18]
------------------------	-----	----------------------

Singapore

Bank Name	SPM Bank Name ID	Valid Length
UNITED OVERSEAS BANK	1	[10]
MAYBANK	3	[11]
DBS BANK / POSB BANK	4	[9, 10, 17]
OVERSEA-CHINESE BANKING CORP.	5	[10, 12]
STANDARD CHARTERED BANK	6	[10]
CITIBANK SINGAPORE LIMITED	7	[10]
CIMB BANK	9	[10]
HONGKONG AND SHANGHAI BANKING CORP.	10	[12]
HSBC BANK (SINGAPORE) LTD	11	[12]
RHB Bank	12	[11]
Deutsche Bank AG	13	[10]
CITIBANK N.A. SINGAPORE	156	[10]
BANK OF CHINA LIMITED	157	[14,15]
BANK OF TOKYO-MITSUBISHI UFJ, LTD	158	[6]
HONG LEONG BANK BERHAD	159	[11,13]
JPMORGAN CHASE BANK, N.A.	160	[10]
SUMITOMO MITSUI BANKING CORPORATION	161	[8]
AUSTRALIA AND NEW ZEALAND BANKING GROUP LTD	162	[16]
ICICI BANK LIMITED	163	[12]
BANGKOK BANK PUBLIC COMPANY LIMITED	164	[11]
BANK OF AMERICA, N.A.	165	[8,12]

Malaysia

Bank Name	SPM Bank Name ID	Valid Length
AMBank BHD	1	[13]

BANK SIMPANAN NASIONAL	2	[16]
CITIBANK BHD	3	[10]
CIMB BANK BHD	4	[10,14]
HONG LEONG BANK BHD	5	[11]
HSBC BANK MALAYSIA BHD	6	[12]
MALAYAN BANKING BHD (MAYBANK)	7	[12]
OCBC BANK MALAYSIA BHD	8	[10]
PUBLIC BANK BHD	9	[10]
RHB BANK BHD	10	[14]
STANDARD CHARTERED BANK BHD	11	[12]
UNITED OVERSEAS BANK MALAYSIA BHD	12	[10,11]
BANK ISLAM MALAYSIA BERHAD	13	[14]
ALLIANCE BANK MALAYSIA BERHAD	14	[15]
AFFIN BANK	15	[12]
AL_RAJHI BANK	16	[15]
BANK OF CHINA (MALAYSIA) BERHAD	17	[13,15]
DEUTSCHE BANK	18	[10,11,12,13,14,15,16,17]
INDUSTRIAL AND COMMERCIAL BANK OF CHINA	19	[17]
J.P. MORGAN CHASE BANK BERHAD	20	[10,11,12,13,14,15,16,17]
BANK MUAMALAT	21	[14]
KUWAIT FINANCE HOUSE (MALAYSIA) BHD	22	[12]
BANK PERTANIAN MALAYSIA BERHAD (AGROBANK)	23	[16]
BANK KERJASAMA RAKYAT MALAYSIA	24	[10,12]
BANK OF AMERICA	25	[5] to [17]

Thailand

Bank Name	SPM Bank Name ID	Valid Length
ไทยพาณิชย์ (SCB)	1	[10]
กสิกรไทย (Kbank)	2	[10]
กรุงไทย (KTb)	3	[10]

กรุงเทพ (BBL)	4	[10]
กรุงศรี (BAY)	5	[10]
ธนชาติ (Thanachart)	6	[10]
ทหารไทย (TMB)	7	[10]
เกียรตินาคิน (Kiatnakin)	8	[10]
เสถียรชาร์เตอร์ด (Standard Chartered)	9	[10,11]
ยูโอบี (UOB)	10	[10]
ทีสโก้ (TISCO)	11	[14]
ซีไอเอ็มบี (CIMB)	12	[10]
ไอซีบีซี (ICBC)	13	[10]
ธนาคารเพื่อการเกษตร (BAAC)	14	[12]
ธนาคารออมสิน (GSB)	15	[12]
ธนาคารอาคารสงเคราะห์ (GHB)	16	[12]
ซิตีแบงก์ (CITIBANK)	17	[10]
Mega ICB	18	[10]
BNP Paribas	19	[16]
Deutsche Bank	20	[10]
HSBC Bank	21	[12]
Sumitomo Mitsui Bank	22	[10]
Mizuho Corporate Bank	23	[12]
Bank of America	24	[8] [9] [10] [11]
The Bank Of Tokyo-Mitsubishi UFJ	25	[10]
The Royal Bank of Scotland N.V.	26	[6]
Islamic Bank of Thailand	27	[10] [12]
JPMorgan Chase	28	[10]
Thai Credit Retail Bank	29	[10]
Land and Houses Bank	30	[10]
ANZ Thai	31	[10]
Oversea-Chinese Banking Corporation	32	[10]
RHB Bank Berhad	33	[15]
Export-Import Bank of Thailand	34	[10]

Small and Medium Enterprise Development Bank of Thailand	35	[10]
--	----	------

Philippines

Bank Name	SPM Bank Name ID	Valid Length
Asia United Bank (AUB)	1	[12]
Banco de Oro Bank (BDO)	2	[10, 11, 12, 13]
Bank of Commerce	3	[10, 11, 12]
Bank of the Philippine Islands (BPI)	4	[10]
China Banking Corporation (Chinabank)	6	[10,12]
Citibank	7	[10, 16]
CTBC Bank	8	[10]
Deutsche Bank	9	[10]
Development Bank of the Philippines (DBP)	10	[11, 12, 13]
East West Bank	11	[11, 12]
Hongkong and Shanghai Banking Corporation (HSBC)	12	[10,12]
Land Bank of the Philippines (Landbank)	13	[10]
Maybank Philippines	14	[10, 11, 12]
Metropolitan Bank and Trust Company(Metrobank)	15	[13]
Philippine Bank of Communications (PBCom)	16	[12]
Philippine National Bank(PNB)	17	[12, 16]
Philippine Veterans Bank (Veterans Bank)	18	[11, 12, 13]
Philippine Trust Company (Philtrust Bank)	19	[10, 11, 12]
Rizal Commercial Banking Corporation	20	[10]

(RCBC)		
Robinsons Bank Corporation (Robinsons Bank)	21	[12,15]
Security Bank Corporation (Security Bank)	22	[12, 13]
Standard Chartered Bank	23	[13]
Union Bank of the Philippines (Unionbank)	24	[12]
United Coconut Planters Bank (UCPB)	25	[12]
Philippine Savings Bank (PSBank)	26	[12]
Chinabank Savings	27	[12]
Sterling Bank	28	[12,13,14,15,16]
Bank of America	29	[8, 9, 10, 11, 12]
ICBC Manila	53	[15]
Seabank Philippines, Inc.	82	[11]

Taiwan

Bank Name	SPM Bank Name ID	Valid Length
臺灣銀行	1	[12]
臺灣土地銀行	3	[12]
合作金庫商業銀行	4	[13]
第一商業銀行	5	[11,13]
華南商業銀行	114	[12]
彰化商業銀行	115	[14]
上海商業儲蓄銀行	116	[14]
台北富邦商業銀行	117	[12,14]
國泰世華商業銀行	118	[11,12,14]
高雄銀行	119	[12]
兆豐國際商業銀行	120	[11]
全國農業金庫	121	[14]
花旗(台灣)商業銀行	122	[10, 14]
澳盛(台灣)商業銀行	123	[10,11,12,13,14]
中華開發工業銀行	124	[12]
王道商業銀行	125	[12, 14]
臺灣中小企業銀行	126	[11]
渣打國際商業銀行	127	[11, 12, 14]
台中商業銀行	128	[12]
京城商業銀行	129	[12, 13]
匯豐(台灣)商業銀行	130	[12, 14]
瑞興商業銀行	131	[13]
華泰商業銀行	132	[13]
臺灣新光商業銀行	133	[12, 13]
陽信商業銀行	134	[11, 12]
板信商業銀行	135	[14]
三信商業銀行	136	[10]
中華郵政股份有限公司	137	[14]
聯邦商業銀行	138	[12]
遠東國際商業銀行	139	[14]
元大商業銀行	140	[13, 14]

永豐商業銀行	141	[14]
玉山商業銀行	142	[13,14]
凱基商業銀行	143	[12,14]
星展(台灣)商業銀行	144	[10,11,12]
台新國際商業銀行	145	[14]
大眾商業銀行	146	[12]
日盛國際商業銀行	147	[14]
安泰商業銀行	148	[14]
中國信託商業銀行	149	[12, 13,14]
日商瑞穗銀行	150	[14]

Vietnam

Appendix 3 - Available Payout Channels

Country	Payout Channels	Support Interbank?
Indonesia	Mandiri Interbank	Yes
	Mandiri Inhouse	No
	BNI	No
	BRI	No
	BCA	No
	ShopeePay	No
Malaysia	UOB	Yes
Singapore	UOB	Yes
Thailand	Airpay (including AirPay wallet)	Yes
	AirPay Bank Account Payout (aka. IS)	

Philippines	DragonPay	Yes (14 banks)
	CitiBank	Yes (including non-DragonPay supported banks)
Taiwan	CTBC	Yes
Vietnam		

Appendix 4 - Required Client Business Information

In preparation for integration with SPM, the client will be required to provide the following information:

Info Required	Description	From Client (Please fill in this column)
Business Name	The name of the current business, used to generate the API endpoint.	
Bank account information	<p>Client bank account information.</p> <p>Including bank name, account number, account name (legal entity), branch name & city.</p> <p>Need to match the payout channel client wants to activate.</p>	
Channel List	<p>The channel (bank) client wants to use.</p> <p>Please refer to Appendix 3 for the complete list of available channels.</p>	
Recipients List	<p>List of email addresses to receive the payout reports.</p> <p>Payout reports will contain the payout that SPM processed on behalf of the client.</p>	
Whitelist IP for Calling API	SPM needs this information to whitelist the IP that will call SPM's payout API.	
Callback URL	<p>Client needs to define the callback URL that SPM will use to send payout confirmation requests.</p> <p>Need to support the payload that SPM has defined as per section <u>5.3.4 Payout Confirmation Callback</u></p>	

Appendix 5 - Limit Information

Note that this is the limit imposed by the Bank side on the account level.
Different bank accounts may have different limits.

Country	Payout Channel	Minimum Transaction Limit	Maximum Transaction Limit	Daily Limit	Per Account Limit
ID	Mandiri Inhouse	IDR 1	No limit	No limit	No limit
ID	Mandiri Interbank	IDR 1	No limit	No limit	No limit
ID	BNI Inhouse*	IDR 1	No limit	No limit	No limit
ID	BCA Inhouse	IDR 1	No limit	IDR 200 billion	No limit
ID	BRI Inhouse	IDR 1	IDR 200 billion	IDR 200 billion	IDR 200 billion

*) Note that for BNI, some destination account type require a minimum transaction amount to be IDR 10,000, and will fail any incoming transactions lower than that threshold.

It is advisable for clients to set a reasonable minimum transaction amount when creating payout use-cases.

Appendix 6 - SPM Admin Use Guide

https://docs.google.com/presentation/d/1RVbYjsB_C5AH_Z_wTieLTur6bSed37H1f66Adm1Vjcl/edit#slide=id.g106d08c8668_1_72

Q&A

1. What information is required when integrating with SPM outbound service?

Answer: [SPM outbound client checklist](#)

<https://docs.google.com/spreadsheets/d/1zg2sPGbYHEJ1-vSaOkQnO1Ws2x9Z5zJZwnsKtf37Oo8/edit#gid=1231501646>

2. How to build a new debit bank account for payout, and how to ask bank grant permission for spm to deduct?

Owner to build a new debit account for client business: [treasury finance team](#)

Who help grant the permission: [ask for relationship manager from the bank](#)

Region	Treasury Finance team	Relevant Finops	Remarks
--------	-----------------------	-----------------	---------

ID	Sea group: armita@shopee.com , Insurance: monica.monica@shopee.com	SPP+MP: andrianus.kurniawan@shopee.com Credit: ines.devina@shopee.com SPX: gabriela.tanasaleh@shopee.com Food: helma.devina@shopee.com	
MY		lavinrush.naidu@shopee.com	
PH	lorraine.marquez@shopee.com shaun.ochia@shopee.com	lorraine.marquez@shopee.com shaun.ochia@shopee.com	
SG		lavinrush.naidu@shopee.com	
TH	--	--	Right now payout service is provided by ShopeePay, so only support ShopeePay BA in TH&VN
VN	--	--	
TW	Katrina Chien 簡秀伶 chienk@seagroup.com Beryl Hou 侯佩伶 houb@seagroup.com Jocelin Wu 吳雅茹 FN boss wujo@seagroup.com	--	申请账户可以发信给到Fin team. 并抄送 Cory Tu <cory.tu@shopee.com>Local Payment team Sean Wang <sean.wang@shopee.com>Local Payment team Johnny Chao <johnny.chao@shopee.com>Local PM team Jenny Wang [TW] <jenny.wang@shopee.com>Local PM team
BR		eugenia.papa@shopee.com fabio.chu@shopee.com payment PM	

3. Daily count or txn accumulation amount limit from gateway side

Please see

<https://docs.google.com/spreadsheets/d/1JnipGVDs7fHK85pgnnTY2WlxYexOHUPyEtCnjTw8MV4/edit#gid=0>

4. QPS for SPM API

API endpoint	Description	QPS (req/s)
/create_payment_source	General payout api	64
/get_payout_details_of_payment_source	Query payout status	52

5. How to make reconciliation among payout services?

Client <> SPM: SPM recon system, which is called SPR will help provide documents to explain how to integrate with SPR.

SPM <> 3PP: SPR would handle channel reconciliation. So for clients, just need to care about the reconciliation between client and SPM.

5. How to integrate ShopeePay wallet as beneficiary account?

There are 2 types of fund movement when clients integrate with SPP wallet.

invoice:

- 不用提前充值
- 结算的时候会计算总共disburse了多少来创invoice单, 然后线下付清invoice金额后, ops会update为已付清
- 结算周期目前只支持每天一次, 以后可能会有其他选项

prefunding:

- ops可以往商户钱包充钱
- disburse的金额只能用充了多少钱, 用完就不能发了

Version Control

Version	Date	Editor	Remarks
v1.0	19.02.2021	BP, YT	First published version
v1.1	02.03.2021	BP, YT	Added Go-Live checklist
v1.2	25.08.2021	Fangbo	Add CB payee payout
v1.3	01.11.2021	Fangbo	Add Q&A