

Slip2Go API Documentation

Version 1.2 (latest update at 07-05-2025)

Overview

Slip2Go provides various API to serve our client about slip verification. We've provided a full set of data that the client can use to check the condition by themself. In the other hand the client has the option to just send information such as bank account or transfer amount and the API will check the condition for you.

Disclaimer

- We're a service provider that provides omni channels for the clients to verify their slips.
- We're not associated with any banks transfer protocol or payment gateway. So we can't guarantee that the transaction was made or not. It's just a slip verification service that verifies with "Official Bank API".

Slip2GO Base API Url

connect.slip2go.com/api

List of contents

- Authentication
 - API Secret
 - IP Whitelist
- API Endpoint
 - Get slip information by QR-Code
 - Get slip information by QR Image
 - Get slip information by refId (checked slip)
 - Get account and package information
- Request Example
- Response Example
- Data Description

Authentication

You need to get through the authentication step to use the Slip2Go API more safely.

IP Whitelist

1. Slip2Go provide IP Whitelist feature that each shop can specify which IP that allowed to use “API Secret”
2. IP Whitelist was set as “*” by default, Which means any IP Addresses was allowed.
3. You can add or remove IP Address at any time, But the cap is at 10 IPs per shop.

API Secret

1. To use Slip2GO API you need to attach an API secret in the header of every request.
 - 1.1. Header Key = “Authorization”
 - 1.2. Header Value = “Bearer {apiSecret}”
2. The “API Secret” can be found in the “API Connect” menu after you logged in to the system.
3. Each shop has only one API Secret at a time. And there is an option to regenerate new API Secret for any purposes.
4. You have to aware that when the “New API Secret” was regenerated, The previous API Secret will no longer be usable anymore.

Description

Method	GET / POST *Depends on your endpoint you want to use.
Url	Depends on your endpoint you want to use.
Header	Authorization: Bearer {apiSecret}
Body	Depends on your endpoint you want to use.

Example (JavaScript)

```
JavaScript
const axios = require('axios')
let config = {
  method: 'GET',
  url: 'http://{apiUrl}/api/account/info',
  headers: {
    'Authorization': 'Bearer {secretKey}',
    'Content-Type': 'application/json'
  }
};
axios.request(config).then((response) => {
  console.log(JSON.stringify(response.data))
}).catch((error) => {
  console.log(error)
})
)
```

API Endpoint

All API endpoints required authorization header. More information can be found at the “Authentication” section.

Verify slip by QR-Code

Endpoint	/verify-slip/qr-code/info
Method	POST
Authorization	*required
Request Data	JSON Body
Description	This API needs QR-Code and condition as a request object. The API will check QR Code is valid or not then check any condition as you requested and send the final result as a response.

Request Information

Key	Type	Description	Example	Required
payload	Object	Payload of data	{ payload: { ... } }	✓
qrCode	String	Code that decoded from QR Code	0123456789xxx	✓
checkCondition	Object	Condition information *Set to check condition	{ ... }	✗
checkDuplicate	Boolean	Flag to check duplicate *Set to check duplicate slip	true false	✗
checkReceiver	Array	Array for check receiver account *Can set more than 1 condition *matched only 1 condition the result will be valid	[{ receiver1 }, { receiver2 }, ...]	✗
accountType	String	Account type to check	“01004” = ธนาคาร กสิกร	✗
accountNameTH	String	Receiver name in Thai to check *Partial check	“สมชาย สลิปทูโก” *Don't set any prefix	✗

		accountNameEN	String	Receiver name in English to check *Partial check	"Somchay Slip2go" *Don't set any prefix	x
		accountNumber	String	Account number to check including merchantId, phoneNumber and citizenID in all others account type. *Partial check	"Xxxxxx1234 *Don't set any space or special characters	x
		checkAmount	Object	Check amount condition *Set to check transfer amount	{ ... }	x
		type	String	Condition type to check	"lte" = lesser than or equal "eq" = only equal "gte" = greater than or equal *default is "eq"	x
		amount	String	Transfer amount	"10000" *don't set 0 and ,	x
		checkDate	Object	Check date information *Set to check transfer date	{ ... }	x
		type	String	Condition type to check	"lte" = lesser than or equal "eq" = only equal "gte" = greater than or equal *default is "eq"	x
		date	DateISO	Transfer date in DateISO format	"2025-10-05T14:48:00.000Z" *GMT time only	x

Example Request (JSON)

Without condition

```
Unset
{
    "payload": {
        "qrCode": "xxx-xxxx-xxxxxx"
    }
}
```

With condition

```
Unset
{
    "payload": {
        "qrCode": "xxx-xxxx-xxxxxx",
        "checkCondition": [
            {
                "checkDuplicate": true, // check duplicate *optional
                "checkReceiver": [
                    {
                        // check all info about receiver bank account
                        "accountType": "01004",
                        "accountNameTH": "สมชาย สลิปทูโก",
                        "accountNameEN": "Somchay Slip2go",
                        "accountNumber": "xxxxxx1234"
                    },
                    {
                        // check only bank number
                        "accountNumber": "xxxxxx1234"
                    },
                    {
                        // check only bank owner name
                        "accountNameTH": "สมชาย สลิปทูโก",
                        "accountNameEN": "Somchay Slip2go",
                    }
                ],
                "checkAmount": { // check transfer amount *optional
                    "type": "eq", // eq, gte, lte
                    "amount": 10000
                }
            }
        ]
    }
}
```

```
        },
        "checkDate": { // check transfer date *optional
            "type": "eq" // eq, gte, lte
            "date": "2025-10-05T14:48:00.000Z" // GMT time
        }
    }
}
```

Response Information

Key		Type	Description	Example
code		String	The code represents the result.	"200000"
message		String	The message represents the result.	"Slip found"
data		Object	Slip information	{ ... }
	referenceId	UUID	Slip2Go reference ID of slip	"92887bd5-60d3-4744-9a98-b8574eaxxxxx-xx"
	decode	String	Decoded qr code from slip	123456789xxx
	transRef	String	Transfer reference from Bank.	"015073xxx5xxx12245"
	dateTime	String	Transfer date	"2025-10-05T14:48:00.000Z"
	amount	Number	Transfer amount	100
	ref1	String	Reference data 1 from Bank.	"xxxxx" null
	ref2	String	Reference data 2 from Bank.	"xxxxx" null
	ref3	String	Reference data 3 from Bank.	"xxxxx" null
	receiver	Object	Receiver information	{ ... }
	account	Object	Receiver account information	{ ... }
	name	String	Receiver account name <i>*partial information</i>	
	bank	Object	Receiver bank account information	{ ... }
	account	String	Receiver bank account number <i>*partial information</i>	"Somchay Slip2go"
	proxy	Object	Proxy information such as PromptPay.	{ ... }
	type	String	Proxy account type including NATID - CitizenID MSISDN - Phone Number EWALLTID - E Wallet EMAIL - Email BILLERID - Bill from biller such as mobile banking ORFT - Wallet from GSB (ເປົາດັ່ງ)	"NATID"
	account	String	Proxy account number	"12850X500"
	bank	Object	Receiver bank information	{ ... }

			id	String	Receiver bank id	“004”
			name	String	Receiver bank name	“ธนาคารกสิกรไทย”
	sender		Object	Sender information	{ ... }	
	account		Object	Sender account information	{ ... }	
	name		String	Sender account name <i>*partial information</i>	“xxx2Go xxVerifyxx”	
	bank		Object	Sender bank account information	{ ... }	
	account		String	Sender bank account number <i>*partial information</i>	“xxx-x-45xx-5”	
	bank		Object	Sender bank information	{ ... }	
	id		String	Sender bank id	“025”	
	name		String	Sender bank name	“ธนาคารกรุงศรีอยุธยา”	

Example Response (JSON)

```

Unset
{
  "code": "200000",
  "message": "Slip found",
  "data": {
    "referenceId": "92887bd5-60d3-4744-9a98-b8574eaxxxxxx-xx",
    "decode": "20014242082547BPM049885102TH9104xxxx",
    "transRef": "015073144041ATF00999",
    "dateTime": "2025-10-05T14:48:00.000Z",
    "amount": 1,
    "ref1": null,
    "ref2": null,
    "ref3": null,
    "receiver": {
      "account": {
        "name": "บริษัท สลิปทูโก จำกัด",
        "bank": {
          "account": "xxx-x-x5366-x"
        },
      }
    }
  }
}

```

```
        "proxy": {
            "type": "NATID",
            "account": "xxx-x-x5366-x"
        },
        "bank": {
            "id": "004",
            "name": "ธนาคารกสิกรไทย"
        }
    },
    "sender": {
        "account": {
            "name": "บริษัท สลิปทูโก จำกัด",
            "bank": {
                "account": "xxx-x-x9866-x"
            }
        },
        "bank": {
            "id": "004",
            "name": "ธนาคารกสิกรไทย"
        }
    }
}
```

Example Usage

JavaScript

```
JavaScript
const axios = require('axios')
let config = {
  method: 'POST',
  maxBodyLength: Infinity,
  url: '{ apiUrl }/api/verify-slip/qr-code/info',
  headers: {
    'Authorization': 'Bearer {secretKey}'
  },
  data: {
    "payload": {
      "qrCode": "0123456789xxx" // decode from QR
    }
  }
}
axios.request(config).then((response) => {
  console.log(JSON.stringify(response.data))
}).catch((error) => {
  console.log(error)
})
```

Curl

```
Unset
curl --location '{ apiUrl }/api/verify-slip/qr-code/info' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer {secretKey}' \
--data '{
  "payload":{
    "qrCode":"xxxxxxxxxxxxxxxxxxxxxx"
  }
}'
```

Verify slip by QR Image

Endpoint	/verify-slip/qr-image/info
Method	POST
Authorization	*required
Request Data	Multipart/Form-Data
Description	This API needs QR-Image as a file and condition object in form data. The API will check the QR Code from the image that is valid or not, then check any condition as you requested and send the final result as a response.

Request Information

Key	Type	Description	Example	Required
file	File	Slip image file that want to verify.	xxx.png	✓
payload	JSON	Condition information to check slip.	{ ...info below... }	✗

Payload (Set only when want to check condition for the slip)

Key	Type	Description	Example	Required
checkDuplicate	Boolean	Flag to check duplicate *Set to check duplicate slip	true false	✗
checkReceiver	Array	Array for check receiver account *Can set more than 1 condition *matched only 1 condition the result will be valid	[{ receiver1 }, { receiver2 }, ...]	✗
accountType	String	Account type to check	"01004" = ธนาคาร กสิกร	✗
accountNameTH	String	Receiver name in Thai to check *Partial check	"สมชาย สลิปทูゴ"	✗
accountNameEN	String	Receiver name in English	"Somchay Slip2go"	✗

				to check *Partial check	*Don't set any prefix	
		accountNumber	String	Account number to check including merchantId, phoneNumber and citizenID in all others account type. *Partial check	"XXXXXX1234 *Don't set any space or special characters	x
		checkAmount	Object	Check amount condition *Set to check transfer amount	{ ... }	x
		type	String	Condition type to check	"lte" = lesser than or equal "eq" = only equal "gte" = greater than or equal *default is "eq"	x
		amount	String	Transfer amount	"10000" *dont set 0 and ,	x
		checkDate	Object	Check date information *Set to check transfer date	{ ... }	x
		type	String	Condition type to check	"lte" = lesser than or equal "eq" = only equal "gte" = greater than or equal *default is "eq"	x
		date	DateISO	Transfer date in DateISO format	"2025-10-05T14:48:00.000Z" *GMT time only	x

Example Payload

```
Unset
{
  "checkDuplicate": true, // set to check duplicate
  "checkReceiver": [{ // set to check receiver account
    "accountType": "01004",
    "accountNameTH": "สมชาย สลิปทูโก",
    "accountNameEN": "Somchay Slip2go",
    "accountNumber": "xxxxxx1234"
  }],
  "checkAmount": { // set to check transfer amount
    "type": "eq", // eq, gte, lte
    "amount": 10000
  }
}
```

Response Information

Key		Type	Description	Example
code		String	The code represents the result.	"200000"
message		String	The message represents the result.	"Slip found"
data		Object	Slip information	{ ... }
	referenceId	UUID	Slip2Go reference ID of slip	"92887bd5-60d3-4744-9a98-b8574eaxxxxx-xx"
	decode	String	Decoded qr code from slip	123456789xxx
	transRef	String	Transfer reference from Bank.	"015073xxx5xxx12245"
	dateTime	String	Transfer date	"2025-10-05T14:48:00.000Z"
	amount	Number	Transfer amount	100
	ref1	String	Reference data 1 from Bank.	"xxxxx" null
	ref2	String	Reference data 2 from Bank.	"xxxxx" null
	ref3	String	Reference data 3 from Bank.	"xxxxx" null
	receiver	Object	Receiver information	{ ... }
	account	Object	Receiver account information	{ ... }
	name	String	Receiver account name <i>*partial information</i>	
	bank	Object	Receiver bank account information	{ ... }
	account	String	Receiver bank account number <i>*partial information</i>	"Somchay Slip2go"
	proxy	Object	Proxy information such as PromptPay.	{ ... }
	type	String	Proxy account type including NATID - CitizenID MSISDN - Phone Number EWALLTID - E Wallet EMAIL - Email BILLERID - Bill from biller such as mobile banking ORFT - Wallet from GSB (ເປົາດັງ)	"NATID"
	account	String	Proxy account number	"12850X500"
	bank	Object	Receiver bank information	{ ... }

			id	String	Receiver bank id	“004”
			name	String	Receiver bank name	“ธนาคารกสิกรไทย”
	sender		Object	Sender information	{ ... }	
	account		Object	Sender account information	{ ... }	
	name		String	Sender account name <i>*partial information</i>	“xxx2Go xxVerifyxx”	
	bank		Object	Sender bank account information	{ ... }	
	account		String	Sender bank account number <i>*partial information</i>	“xxx-x-45xx-5”	
	bank		Object	Sender bank information	{ ... }	
	id		String	Sender bank id	“025”	
	name		String	Sender bank name	“ธนาคารกรุงศรีอยุธยา”	

Example Response (JSON)

```
Unset
{
  "code": "200000",
  "message": "Slip found",
  "data": {
    "referenceId": "92887bd5-60d3-4744-9a98-b8574eaxxxxxx-xx",
    "decode": "20014242082547BPM049885102TH9104xxxx",
    "transRef": "015073144041ATF00999",
    "dateTime": "2025-10-05T14:48:00.000Z",
    "amount": 1,
    "ref1": null,
    "ref2": null,
    "ref3": null,
    "receiver": {
      "account": {
        "name": "บริษัท สลิปทูโก จำกัด",
        "bank": {
          "account": "xxx-x-x5366-x"
        }
      }
    }
  }
}
```

```
        },
        "proxy": {
            "type": "NATID",
            "account": "xxx-x-x5366-x"
        }
    },
    "bank": {
        "id": "004",
        "name": "ธนาคารกสิกรไทย"
    }
},
"sender": {
    "account": {
        "name": "บริษัท สลิปทูโก จำกัด",
        "bank": {
            "account": "xxx-x-x9866-x"
        }
    },
    "bank": {
        "id": "004",
        "name": "ธนาคารกสิกรไทย"
    }
}
}
```

Example Usage

JavaScript

```
JavaScript

const axios = require('axios');
const data = new FormData();
const payload = {
    checkDuplicate: true,
    checkReceiver: [
        accountType: '01004',
        accountNameTH: 'สมชาย ส่ายใหม',
        accountNameEN 'Somchay Saimai',
        accountNumber: 'xxxxxxxx1234'
    ],
    checkAmount: {
        type: 'eq',
        amount: 200.31
    }
}

data.append('file',
fs.createReadStream('~/Users/Desktop/slip.jpg'));
data.append('payload', JSON.stringify(payload))

let config = {
    method: 'post',
    maxBodyLength: Infinity,
    url: '{ apiUrl }/api/verify-slip/qr-image/info',
    headers: {
        'Authorization': 'Bearer { apiSecret}'
    },
    data : data
};

axios.request(config).then((response) => {
    console.log(JSON.stringify(response.data));
})
.catch((error) => {
    console.log(error);
});
```

Curl

```
Unset
curl --location '{apiUrl}/api/verify-slip/qr-image/info' \
--header 'Authorization: Bearer {secretKey}' \
--form 'file=@"/Users/Chay/Desktop/slip.jpg"' \
--form 'payload='{
    checkDuplicate: true,
    checkReceiver: [
        {
            accountType: '01004',
            accountNameTH: 'สมชาย สายไหม',
            accountNameEN: 'Somchay Saimai',
            accountNumber: 'xxxxxxxx1234'
        }],
    checkAmount: {
        type: 'eq',
        amount: 200.31
    }
}'"
```

Get slip by referenceId (Get verified slip)

Endpoint	/verify-slip/{referenceId}
Method	GET
Authorization	*required
Request Data	Request Params
Description	This API is to get the information of a verified slip without paying any quota. Just send a reference ID and you will get the whole slip information.

Request Information

Key	Type	Description	Example	Required
referenceId	UUID	Slip reference ID from slip that already scanned.	"92887bd5-60d3-4744-9a98-b8574eaxx xxx-xx"	✓

Response Information

Key	Type	Description		Example
code	String	The code represents the result.		"200000"
message	String	The message represents the result.		"Slip found"
data	Object	Slip information		{ ... }
referenceId	UUID	Slip2Go reference ID of slip		"92887bd5-60d3-4744-9a98-b8574eaxxxxx-xx"
decode	String	Decoded qr code from slip		123456789xxx
transRef	String	Transfer reference from Bank.		"015073xxx5xxx12245"
dateTime	String	Transfer date		"2025-10-05T14:48:00.000Z"
verifyDate	String	First verify date time of the slip.		"2025-10-05T14:48:00.000Z"
amount	Number	Transfer amount		100
ref1	String	Reference data 1 from Bank.		"xxxxx" null

	ref2	String	Reference data 2 from Bank.	"xxxxx" null
	ref3	String	Reference data 3 from Bank.	"xxxxx" null
	receiver	Object	Receiver information	{ ... }
	account	Object	Receiver account information	{ ... }
	name	String	Receiver account name <i>*partial information</i>	
	bank	Object	Receiver bank account information	{ ... }
	account	String	Receiver bank account number <i>*partial information</i>	"Somchay Slip2go"
	proxy	Object	Proxy information such as PromptPay.	{ ... }
	type	String	Proxy account type including NATID - CitizenID MSISDN - Phone Number EWALLTID - E Wallet EMAIL - Email BILLERID - Bill from biller such as mobile banking ORFT - Wallet from GSB (ເປົາດັ່ງ)	"NATID"
	account	String	Proxy account number	"12850X500"
	bank	Object	Receiver bank information	{ ... }
	id	String	Receiver bank id	"004"
	name	String	Receiver bank name	"ธนาคารกสิกรไทย"
	sender	Object	Sender information	{ ... }
	account	Object	Sender account information	{ ... }
	name	String	Sender account name <i>*partial information</i>	"xxx2Go xxVerifyxx"
	bank	Object	Sender bank account information	{ ... }
	account	String	Sender bank account number <i>*partial information</i>	"xxx-x-45xx-5"
	bank	Object	Sender bank information	{ ... }
	id	String	Sender bank id	"025"
	name	String	Sender bank name	"ธนาคารกรุงศรีอยุธยา"

Example Response (JSON)

```
Unset
{
  "code": "200000",
  "message": "Slip found",
  "data": {
    "referenceId": "92887bd5-60d3-4744-9a98-b8574eaxxxxxx-xx",
    "decode": "20014242082547BPM049885102TH9104xxxx",
    "transRef": "015073144041ATF00999",
    "dateTime": "2025-10-05T14:48:00.000Z",
    "verifyDate": "2025-10-05T14:48:00.000Z",
    "amount": 1,
    "ref1": null,
    "ref2": null,
    "ref3": null,
    "receiver": {
      "account": {
        "name": "บริษัท สลิปทูโก จำกัด",
        "bank": {
          "account": "xxx-x-x5366-x"
        },
        "proxy": {
          "type": "NATID",
          "account": "xxx-x-x5366-x"
        }
      },
      "bank": {
        "id": "004",
        "name": "ธนาคารกสิกรไทย"
      }
    },
    "sender": {
      "account": {
        "name": "บริษัท สลิปทูโก จำกัด",
        "bank": {
          "account": "xxx-x-x9866-x"
        }
      }
    }
  }
}
```

```
        }
    },
    "bank": {
        "id": "004",
        "name": "ธนาคารกสิกรไทย"
    }
}
}
```

Example Usage

JavaScript

```
JavaScript
const axios = require('axios')
let config = {
    method: 'GET',
    url: '{apiUrl}/api/verify-slip/{referenceId}',
    headers: {
        'Authorization': 'Bearer {secretKey}',
    },
}
axios.request(config).then((response) => {
    console.log(JSON.stringify(response.data))
}).catch((error) => {
    console.log(error)
})
```

Curl

```
Unset
curl --location '{apiUrl}/api/verify-slip/{referenceId}' \
--header "Authorization: Bearer {secretKey}"
```

Get account information

Endpoint	/account/info
Method	GET
Authorization	*required
Request Data	No
Description	This API is for anyone want the information of a checked slip without paying any quota. Just send a referenceId and you will get the whole same slip information as last time.

Request Information

- Only auth is required for this endpoint.

Response Information

Key	Type	Description	Example
code	String	The code represents the result.	“200001”
message	String	The message represents the result.	“Get Info Success”
data	Object	Account information	{ ... }
shopName	String	Shop name	“Slip2Go Shop”
package	String	Current package name	“BASIC-1”
packageExpiredDate	DateISO	Current package expire date	“2025-10-05T14:48:00.000Z”
quotaLimit	Number	Maximum quota of current package	400
quotaRemaining	Number	Quota available	100
creditRemaining	Number	Credit available	100
autoRenewalPackage	Boolean	Package renewal setting	true false
checkSlipByCredit	Boolean	Scan by credit setting	true false

Example Response Object (JSON)

```
Unset
{
  "code": "200001",
  "message": "Get Info Success",
  "data": {
    "shopName": "Slip2Go Shop",
    "package": "BASIC-1",
    "packageExpiredDate": "2025-04-23T08:32:45.123Z",
    "quotaLimit": 400,
    "quotaRemaining": 100,
    "creditRemaining": 0,
    "autoRenewalPackage": false,
    "checkSlipByCredit": false
  }
}
```

Example Usage

JavaScript

```
JavaScript
const axios = require('axios')
let config = {
  method: 'GET',
  url: '{ apiUrl }/api/account/info',
  headers: {
    'Authorization': 'Bearer {secretKey}',
  },
}
axios.request(config).then((response) => {
  console.log(JSON.stringify(response.data))
}).catch((error) => {
  console.log(error)
})
```

Curl

```
Unset
curl --location '{ apiUrl }/api/account/info' \
  --header "Authorization: Bearer {secretKey}"
```

Request Example

Verify without condition

```
Unset
{
    "payload": {
        "qrCode": "xxx-xxxx-xxxxxx"
    }
}
```

Check slip duplicated

```
Unset
{
    "payload": {
        "qrCode": "xxx-xxxx-xxxxxx"
    },
    "checkCondition": {
        "checkDuplicate": true
    }
}
```

Check receiver account number

```
Unset
{
  "payload": {
    "qrCode": "xxx-xxxx-xxxxxx",
  },
  "checkCondition": {
    "checkReceiver": [
      {
        "accountNumber": "4201853666"
      }
    ]
  }
}
```

Check receiver account name

```
Unset
{
  "payload": {
    "qrCode": "xxx-xxxx-xxxxxx",
  },
  "checkCondition": {
    "checkReceiver": [
      {
        "accountNameTH": "สลิป2go",
        "accountNameEN": "slip2go",
      }
    ]
  }
}
```

Check transfer amount (greater than or equal 1000 THB)

```
Unset
{
  "payload": {
    "qrCode": "xxx-xxxx-xxxxxx",
  },
  "checkCondition": {
    "checkAmount": {
      "type": "gte",
      "amount": 1000,
    }
  }
}
```

Check transfer date (greater than or equal 05-10-2025 14.48 PM)

```
Unset
{
  "payload": {
    "qrCode": "xxx-xxxx-xxxxxx",
  },
  "checkCondition": {
    "checkDate": {
      "type": "gte",
      "date": "2025-10-05T14:48:00.000Z",
    }
  }
}
```

Check receiver account type and account number (is KBank)

```
Unset
{
  "payload": {
    "qrCode": "xxx-xxxx-xxxxxx",
  },
  "checkCondition": {
    "checkReceiver": [
      {
        "accountType": "01004",
        "accountNumber": "4201853666"
      }
    ]
  }
}
```

Check receiver account type and account number (is PromptPay with PhoneNumber)

```
Unset
{
  "payload": {
    "qrCode": "xxx-xxxx-xxxxxx",
  },
  "checkCondition": {
    "checkReceiver": [
      {
        "accountType": "02001",
        "accountNumber": "0902369994"
      }
    ]
  }
}
```

Check receiver account type and account number (is Merchant such as KShop)

```
Unset
{
  "payload": {
    "qrCode": "xxx-xxxx-xxxxxx",
  },
  "checkCondition": {
    "checkReceiver": [
      {
        "accountType": "03000",
        "accountNumber": "075413264"
      }
    ]
  }
}
```

Check receiver account type and account number (is True Money)

```
Unset
{
  "payload": {
    "qrCode": "xxx-xxxx-xxxxxx",
  },
  "checkCondition": {
    "checkReceiver": [
      {
        "accountType": "04000",
        "accountNumber": "0902369994"
      }
    ]
  }
}
```

Check more than 1 condition (receiver, transfer amount, transfer date)

```
Unset
{
  "payload": {
    "qrCode": "xxx-xxxx-xxxxxx",
  },
  "checkCondition": {
    "checkReceiver": [
      {
        "accountType": "01004",
        "accountNumber": "4201853666"
      }
    ],
    "checkAmount": {
      "type": "gte",
      "amount": 1000,
    },
    "checkDate": {
      "type": "gte",
      "date": "2025-10-05T14:48:00.000Z",
    }
  }
}
```

Check multi receiver condition or multi account

```
Unset
{
  "payload": {
    "qrCode": "xxx-xxxx-xxxxxx",
  },
  "checkCondition": {
    "checkReceiver": [
      {
        "accountNumber": "4201853666"
      },
      {
        "accountType": "01004",
        "accountNumber": "7192488543"
      },
      {
        "accountNameTH": "สลิป2โก ตรวจสอบลิป",
        "accountNameEN": "slip2go checkslip"
      },
    ]
  }
}
```

Response Example

Success Code

ลำดับ	สถานะ	รายละเอียด	HTTP Status	Response Code	คิดโควต้า
1	Slip found	พบข้อมูลสลิปในระบบ ธนาคาร	200	200000	✓
2	Get Info Success	ข้อมูลสำเร็จ	200	200001	✗
3	Slip is Valid	ข้อมูลสลิปถูกต้อง	200	200200	✓
4	Recipient Account Not Match	บัญชีผู้รับไม่ถูกต้อง	200	200401	✓
5	Transfer Amount Not Match	ยอดโอนเงินไม่ตรง เงื่อนไข	200	200402	✓
6	Transfer Date Not Match	วันที่โอนไม่ตรงเงื่อนไข	200	200403	✓
7	Slip Not Found	ไม่พบสลิปในระบบ ธนาคาร	200	200404	✓
8	Slip is Duplicated	สลิปซ้ำในระบบ	200	200501	✗

Success Responses

Slip Found

```
Unset
{
    "code": "200000",
    "message": "Slip found",
    "data": {
        "referenceId": "92887bd5-60d3-4744-9a98-b8574eaxxxxx-xx",
        "decode": "0041000600000101030040220014242082547BPM049885102TH9104xxxx",
    }
}
```

```
"transRef": "184440173749C0T08999",
"dateTime": "2024-05-29T05:37:00.000Z",
"amount": 1000,
"ref1": "",
"ref2": "",
"ref3": "",
"receiver": {
    "account": {
        "name": "บริษัท สลิปทูโก จำกัด",
        "bank": {
            "account": "xxx-x-x1234-x"
        },
        "proxy": {
            "type": "BILLERID",
            "account": "XXXXXXXXXXXX1234"
        }
    },
    "bank": {
        "id": "002",
        "name": "ธนาคารกรุงเทพ"
    }
},
"sender": {
    "account": {
        "name": "นาย สมชาย อ้อดยาคั้ยดี",
        "bank": {
            "account": "xxx-x-x1234-x"
        }
    }
}
```

```
        "bank": {  
            "account": "xxx-x-x1234-x"  
        },  
        "bank": {  
            "id": "004",  
            "name": "ธนาคารกสิกรไทย"  
        }  
    }  
}
```

Get Info Success (account info)

```
Unset  
{  
    "code": "200001",  
    "message": "Get Info Success",  
    "data": {  
        "shopName": "myShop",  
        "package": "BASIC-1",  
        "packageExpiredDate": "2024-05-29T05:37:00.000Z",  
        "quotaLimit": 400,  
        "quotaRemaining": 100,  
        "creditRemaining": 0,  
    }  
}
```

```
        "autoRenewalPackage": false,  
        "checkSlipByCredit": false  
    }  
}
```

Slip is Valid

```
Unset  
{  
    "code": "200200",  
    "message": "Slip is Valid",  
    "data": { ... Same as Slip Found ... }  
}
```

Recipient Account Not Match

```
Unset  
{  
    "code": "200401",  
    "message": "Recipient Account Not Match",  
    "data": { ... Same as Slip Found ... }  
}
```

Transfer Amount Not Match

```
Unset
{
  "code": "200402",
  "message": "Transfer Amount Not Match",
  "data": { ... Same as Slip Found ... }
}
```

Transfer Date Not Match

```
Unset
{
  "code": "200403",
  "message": "Transfer Date Not Match",
  "data": { ... Same as Slip Found ... }
}
```

Slip is Duplicated

```
Unset
{
  "code": "200501",
  "message": "Slip is Duplicated",
  "data": { ... Same as Slip Found ... }
}
```

Slip not found

```
Unset
{
  "code": "200404",
  "message": "Slip not found",
  "data": {
    "referenceId": "92887bd5-60d3-4744-9a98-b8574eaxxxxx-xx",
  }
}
```

Error Code

ลำดับ	สถานะ	รายละเอียด	HTTP Status	Response Code	คิดคำตัว
1	QR Code is Incorrect	QR Code does not match Bank criteria.	400	400001	×
2	File is Incorrect	The file is in the wrong format.	400	400002	×
3	Reference Id is Invalid	Reference id not exist.	400	400003	×
4	Request object is invalid	Request object not correct.	400	400400	×
5	Token Mismatch	Token is not exist.	401	401001	×
6	Token Mismatch	Shop not found.	401	401002	×
7	Token Mismatch	Account suspended.	401	401003	×
8	Package Expired	Package is already expired.	401	401004	×
9	Insufficient Quota	Insufficient quota available..	401	401005	×
10	Insufficient Credit	Insufficient credit available.	401	401006	×
11	IP Address not allowed	Ip address is not in Whitelist.	401	401007	×
12	Internal Server Error	Slip2go system error.	500	500500	×

Error Responses

QR Code is incorrect

```
Unset
{
    "code": "400001",
    "message": "QR Code is incorrect"
}
```

File is Incorrect

```
Unset
{
  "code": "400002",
  "message": "File is Incorrect"
}
```

Reference id is Invalid

```
Unset
{
  "code": "400003",
  "message": "Reference id is Invalid"
}
```

Request object is Invalid

```
Unset
{
  "code": "400400",
  "message": "Request object is Invalid"
}
```

Token Mismatch

```
Unset
{
  "code": "401001",
  "message": "Token Mismatch"
}
```

Token Mismatch

```
Unset
{
  "code": "401002",
  "message": "Token Mismatch"
}
```

Token Mismatch

```
Unset
{
  "code": "401003",
  "message": "Token Mismatch"
}
```

Package Expired

```
Unset
{
  "code": "401004",
  "message": "Package Expired"
}
```

Insufficient Quota

```
Unset
{
  "code": "401005",
  "message": "Insufficient Quota"
}
```

Insufficient Credit

```
Unset
{
  "code": "401006",
  "message": "Insufficient Credit"
}
```

IP Address not allowed

```
Unset
{
  "code": "401007",
  "message": "IP Address not allowed"
}
```

Internal Server Error

```
Unset
{
  "code": "500500",
  "message": "Internal Server Error"
}
```

Data Description

Account Type List (for condition check)

Account Type	Code
ธนาคารกรุงเทพ - Bangkok Bank	“01002”
ธนาคารกสิกรไทย - Kasikorn Bank	“01004”
ธนาคารกรุงไทย - Krung Thai Bank	“01006”
ธนาคารทหารไทยธนชาต - TMB Thanachart Bank	“01011”
ธนาคารไทยพาณิชย์ - SCB Siam Commercial Bank	“01014”
ธนาคารกรุงศรีอยุธยา - Krungsri Bank	“01025”
ธนาคารเกียรตินาคินภัทร - Kiatnakin Bank	“01069”
ธนาคารซีไออีมบีไทย - CIMB Thai Bank	“01022”
ธนาคารทิสโก้ - TISCO Bank	“01067”
ธนาคารยูโอบี - UOB United Overseas Bank	“01024”
ธนาคารไทยเครดิต - Thai Credit Bank	“01071”
ธนาคารแลนด์ แอนด์ เฮ้าส์ - LH Bank	“01073”
ธนาคารไอซีบีซี (ไทย) - ICBC Thai	“01070”
ธนาคารพัฒนาวิสาหกิจขนาดกลางและขนาดย่อมแห่งประเทศไทย - SME Bank	“01098”
ธนาคารเพื่อการเกษตรและสหกรณ์การเกษตร - BAAC	“01034”
ธนาคารเพื่อการส่งออกและนำเข้าแห่งประเทศไทย - EXIM Bank	“01035”
ธนาคารออมสิน - GSB Government Saving Bank	“01030”
ธนาคารอาคารสงเคราะห์ - GHB	“01033”
ธนาคารอิสลามแห่งประเทศไทย - Islamic Bank	“01066”
PromptPay - Phone Number	“02001”
PromptPay - Bank Account Number	“02002”
PromptPay - CitizenID	“02003”
PromptPay - E-Wallet *Rabbit LINE Pay, True Money Wallet	“02004”

PromptPay - G-wallet *เป้าตัง (GSB E-Wallet)	“02005”
K+ Shop (KBANK), แม่�ณี (SCB), Be Merchant NextGen (BBL), TTB Smart Shop (TTB)	“03000”
True Money Wallet	“04000”

Bank Code (from Response)

Bank Name	Code
ธนาคารกรุงเทพ - Bangkok Bank	“002”
ธนาคารกสิกรไทย - Kasikorn Bank	“004”
ธนาคารกรุงไทย - Krung Thai Bank	“006”
ธนาคารทหารไทยธนชาต - TMB Thanachart Bank	“011”
ธนาคารไทยพาณิชย์ - SCB Siam Commercial Bank	“014”
ธนาคารกรุงศรีอยุธยา - Krungsri Bank	“025”
ธนาคารเกียรตินาคินภัทร - Kiatnakin Bank	“069”
ธนาคารซีไอเอ็มบีไทย - CIMB Thai Bank	“022”
ธนาคารทิสโก้ - TISCO Bank	“067”
ธนาคารยูโอบี - UOB United Overseas Bank	“024”
ธนาคารไทยเครดิต - Thai Credit Bank	“071”
ธนาคารแลนด์ แอนด์ เฮ้าส์ - LH Bank	“073”
ธนาคารไอซีบีซี (ไทย) - ICBC Thai	“070”
ธนาคารพัฒนาวิสาหกิจขนาดกลางและขนาดย่อมแห่งประเทศไทย - SME Bank	“098”
ธนาคารเพื่อการเกษตรและสหกรณ์การเกษตร - BAAC	“034”
ธนาคารเพื่อการส่งออกและนำเข้าแห่งประเทศไทย - EXIM Bank	“035”
ธนาคารออมสิน - GSB Government Saving Bank	“030”
ธนาคารอาคารสงเคราะห์ - GHB	“033”
ธนาคารอิสลามแห่งประเทศไทย - Islamic Bank	“066”