

# SlipOK API Guide

Version 1.12 (อัปเดทล่าสุด 8 สิงหาคม 2025)

[EN Ver.](#)

## Check Slip Request

Method	POST
Url	<a href="https://api.slipok.com/api/line/apikey/&lt;YOUR_BRANCH_ID&gt;">https://api.slipok.com/api/line/apikey/&lt;YOUR_BRANCH_ID&gt;</a>
Header	<ol style="list-style-type: none"><li>1. { x-authorization : &lt;YOUR_API_KEY&gt; }</li><li>2. กรณีที่เป็น files รูป { content-type: "multipart/form-data" }</li><li>3. กรณีอื่นสามารถใช้<ul style="list-style-type: none"><li>o { content-type: "application/json" }</li><li>o { content-type: "application/x-www-form-urlencoded" }</li></ul></li></ol>
Body (Required)	<ol style="list-style-type: none"><li>1. ข้อมูล Slip<ul style="list-style-type: none"><li>{ data: string } = ค่าที่อ่านได้จาก qr code ข่าวล่างของสลิป</li><li>{ files: File / base64 } = รูปภาพสลิป รองรับนามสกุล JPG, JPEG, PNG, JFIF หรือ WEBP</li><li>{ url: string } = url ของรูปภาพสลิป (ถ้าหากเป็น link google drive / signed url จะจะไม่สามารถใช้ได้ เพราะต้องเช็คขนาดไฟล์)</li></ul></li><li>** ส่งໄດ้แค่อย่างใดอย่างหนึ่งจากสามรูปแบบข้างต้น **</li><li>2. เช็คสลิปให้ตรงกับธนาคารที่ผูกไว้ใน Line LIFF และตรวจสอบสลิปช้า<ul style="list-style-type: none"><li>{ log: boolean } = ระบุเป็น true ถ้าต้องการเช็คธนาคารรับเงินที่ผูกไว้กับสาขา API และเก็บยอดไว้ดูใน Line LIFF ของร้านค้าเพื่อตรวจสอบสลิปช้าได้ ถ้าหากระบุเป็น false หรือไม่ส่งจะเป็นการตรวจสอบสลิปโดย ๆ ไม่มีการเก็บค่า และไม่ตรวจสอบสลิปช้า</li></ul></li></ol>
Body (Optional)	{ amount: number } = ระบุยอดเงินไว้สำหรับเช็คกับยอดเงินในสลิป

## Check Slip Response

Field	Type	Description	Example	Mandatory
success	boolean	Request success	true	Y
data	object			Y
success	boolean	Valid QR code	true	Y
message	string(50)	Verification Message	✓	Y
language	string(2)	Language (EN, TH)	TH	
receivingBank	string(3)	Receiving bank code	006	Y
sendingBank	string(3)	Sender bank code	004	Y

	transRef	string(25)	Transaction reference number contains reference number of the transaction required to verify the payslip	010092101507665143	Y
	transDate	string(8)	Transaction date text. This field contains the date on the payslip in 'yyyyMMdd' format	20200401	Y
	transTime	string(8)	Transaction time text. This field contains time on the payslip in 'HH:mm:ss' format	10:15:07	Y
	transTimestamp	timestamp	Transaction timestamp text. This field contains the timestamp on the payslip in 'YYYY-MM-DDTHH:mm:ss.sssZ' format which is ISO 8601	2020-04-01T03:15:07.000Z	Y
	sender	object			
	displayName	string(50)	The name to display in Thai or English depends on the issuer bank payslip	นาย กลิกร ร ** The data might not be complete need partial match	Y
	name	string(50)	The name to display in English	Mr. Kasikorn R ** The data might not be complete need partial match	Y
	proxy	object			Y
		type	string(12)	From proxy type (NATID, MSISDN, EWALLETID, EMAIL, BILLERID)	MSISDN
		value	string(128)	From proxy value **The value is always masked. Masking format is depend on bank's slip.	086xxx7894
	account	object			Y

		type	string(12)	From account type (BANKAC, TOKEN, DUMMY)	BANKAC	Y
		value	string(40)	From account value **The value is always masked. Masking format is depend on bank's slip.	xxx-x-x0209-x	Y
		receiver	object			Y
	displayName	string(50)	Receiver name to display in Thai or English depends on issuer bank payslip	ธนาธร ร <b>** The data might not be complete need partial match</b>		Y
	name	string(50)	Receiver name in English.	THANATORN R <b>** The data might not be complete need partial match</b>		Y
	proxy	object				Y
	type	string(12)	To proxy type (NATID, MSISDN, EWALLETID, EMAIL, BILLERID)	MSISDN		Y
	value	string(128)	To proxy value **The value is always masked. Masking format is depend on bank's slip.	086xxx0000		Y
	account	object				Y
	type	string(12)	To account type (BANKAC, TOKEN, DUMMY)	BANKAC		Y
	value	string(40)	To account value **The value is always masked. Masking format is depend on bank's slip.	xxx-x-x3109-x		Y
	amount	decimal(12)	This field contains amount on the payslip	50		Y
	paidLocalAmount	decimal(12)	Paid local amount	1		
	paidLocalCurrency	string(3)	Paid local currency	764		

	countryCode	string(2)	Country code	TH	Y
	transFeeAmount	string(12)	Transaction fee amount	0	
	ref1	string(20)	1st Reference number on the payslip		
	ref2	string(20)	2nd Reference number on the payslip		
	ref3	string(20)	3rd Reference number on the payslip		
	toMerchantId	string(15)	To merchant ID		

## Check Slip Quota Request

Method	GET
Url	<a href="https://api.slipok.com/api/line/apikey/&lt;YOUR_BRANCH_ID&gt;/quota">https://api.slipok.com/api/line/apikey/&lt;YOUR_BRANCH_ID&gt;/quota</a>
Header	{ x-authorization : <YOUR_API_KEY> }

## Check Slip Quota Response

Field	Type	Description	Example	Mandatory
success	boolean	Request success	true	Y
data	object			Y
quota	number	จำนวนโควต้าคงเหลือ * กรณีไม่ได้ส่ง log: true จะคิดโควต้าก็ต่อเมื่อสิลิปถูก ต้อง แต่การส่งข้าก็คิดโควต้า เช่นเดียวกัน ** กรณีส่ง log: true จะคิดโควต้าก็ต่อเมื่อสิลิปถูก ต้องและตรงกับบัญชีผู้รับที่ดัง ไว้ใน Line LIFF การส่งข้าจะ ไม่คิดโควต้า แต่ถ้าหากบัญชีผู้ รับที่ดังไว้ไม่ตรงก็จะคิดโควต้า เช่นเดียวกัน	100	Y
overQuota	number	จำนวนโควต้าที่ใช้เกิน	0	Y

## Quota Calculation

- กรณีไม่ได้ส่ง log: true จะคิดโควต้าก็ต่อเมื่อสลิปถูกต้อง แต่การส่งสลิปถูกต้องเดิมข้าก็คิดโควต้าเช่นเดียวกัน
- กรณีส่ง log: true จะคิดโควต้าก็ต่อเมื่อสลิปถูกต้องและตรงกับบัญชีผู้รับที่ตั้งไว้ใน Line LIFF ซึ่งจะทำให้การส่งข้าไม่คิดโควต้า แต่ถ้าหากบัญชีผู้รับที่ตั้งไว้ไม่ตรงก็จะคิดโควต้าเช่นเดียวกัน

## Remark

ในการต่อ API ทางระบบ SlipOK จะรับข้อมูลสลิปไป ตรวจสอบกับฐานข้อมูลส่วนกลาง และส่งข้อมูลนั้นสลิปทั้งหมดกลับไปผ่านAPI

กรณีถ้าทางผู้พัฒนาระบนมีการส่ง log: true มา ทางระบบ SlipOK จะทำการเก็บข้อมูลสลิปให้ และมีการเช็คสลิปข้าให้ด้วย พร้อมเช็คบัญชีรับเงินที่ได้ผูกไว้กับระบบด้วย

กรณีถ้าผู้พัฒนาระบบไม่ได้ส่ง body log: true มา ทางผู้พัฒนาจะต้องเป็นคนเก็บข้อมูลสลิปเองเพื่อยืนยันข้อมูลบนสลิปอีกรอบหนึ่ง ทางระบบ SlipOK จะไม่ได้เก็บข้อมูลเอาไว้ให้ รวมไปถึงกรณีสลิปข้า ทางผู้พัฒนาระบบจะต้องนำ response ข้อมูลต่างๆมาเช็คกับฐานข้อมูลของทางร้านเช่น จำนวนเงินและบัญชีของทางร้านเองทางระบบ SlipOK จะไม่ได้ทำการตรวจสอบให้ โดยข้อนะในการท่าระบบหลังบ้านมีดังนี้

- ข้อมูลเรียกเก็บเป็นจำนวนเงินเท่าไหร (เพื่อป้องกันการตัดต่อเปลี่ยนแปลงยอด)
- บันทึกบัญชีปลายทางของร้าน (เพื่อป้องกันการตัดต่อรูป เช่นการโอนให้บัญชีตัวเอง)
- เก็บบันทึกข้อมูลสลิปในฐานข้อมูล (เพื่อป้องกันสลิปข้า)

## Webhook Forwarding

กรณีที่ผู้พัฒนาระบบมีระบบ Webhook ของตัวเองอยู่แล้ว และยังอยากรับข้อมูลจาก SlipOK แต่อยากได้ Flex Message ใน การตอบกลับโดยไม่ต้องการใช้ผ่าน API เพราะต้องทำการออกแบบ Flex Message ด้วยตัวเอง สามารถทำได้หลายแบบดังนี้

### 1. Webhook Gateway

สามารถสร้าง service ขึ้นมาใหม่ที่ทำการ Forward Webhook Event ที่ได้จาก LINE ส่งต่อไปยัง webhook อันเดิมของลูกค้า และ Webhook ของ SlipOK ได้ โดยอาจจะมีตัวคัดกรองว่า event ประเภท image event ค่อยส่งให้ SlipOK ส่วนประเภทที่เหลือก็ส่งให้ Webhook เดิมของลูกค้าตามปกติ

### 2. Webhook Forwarding

โดยการใช้ Webhook ของลูกค้าตัวเดิม โดยเลือกเฉพาะ image event และทำการส่งต่อมายัง SlipOK

ทั้งสองวิธีการตามหลักแล้วสิ่งที่จำเป็นสำหรับการส่งต่อคือ

- request.headers[ "X-Line-Signature" ]** เพื่อเป็นการยืนยันว่าต้นทางมาจากของ LINE จริง ๆ และปลายทางเป็นผู้รับที่ถูกต้องจริง ๆ
- request.body** เพื่อเป็นการส่งตัว body ของ webhook ทั้งหมดต่อมา

## Compare Receiver Account

กรณีที่ผู้พัฒนาระบบไม่ได้ส่ง **body log: true** มาแล้วต้องการทำการตรวจสอบว่าข้อมูลผู้รับเงินในสิปตรงกับบัญชีที่รับเงินจริง ๆ หรือไม่ หาก SlipOK มีค่าแนะนำสำหรับการตรวจสอบโดยอ้างอิงจากข้อมูลสองอย่างดังนี้

### 1. เลขบัญชี โดยตรวจสอบจาก API field ดังนี้

- receiver.account.value** กรณีผู้รับเป็นบัญชีธนาคาร
- receiver.proxy.value** กรณีผู้รับไม่ใช่บัญชีธนาคาร เช่น Promptpay
- ref1** กรณีรับชำระผ่านผู้รับชำระอื่น ๆ

หมายเหตุ

- เนื่องจากมีการ masking เลขบัญชี โดยการใช้ตัว X หรือ X ดังนั้นจึงต้องมีการทำการ normalize ข้อมูลก่อนโดยการตัดทุกอย่างที่ไม่ใช่ตัวเลขยกเว้นตัว X กับ X ออกก่อนเพื่อนำมาเบริรย์เปลี่ยนกับบัญชีปลายทางอีกทีหนึ่ง ยกตัวอย่างเช่นถ้าหากเลขบัญชีเป็น “99999991234” จะได้รับข้อมูลผ่าน API มาเป็น “XXX-X-XX123-4” ต้องทำการแปลงเป็น “XXXXXX1234” และค่อยดูเฉพาะตัวที่เป็นตัวเลขแต่ละตัวๆกันต่อและดำเนินการตรวจสอบ หรือไม่
- อาจจะมีข้อมูลแค่ 1 field จาก 2 field เช่นผู้รับเป็นบัญชีธนาคารก็จะมีใน “receiver.account.value” แต่ถ้าไม่ใช่จะมีใน “receiver.proxy.value” ดังนั้นถ้าบัญชีปลายทางมีหลายแบบก็ควรเช็คทั้ง 2 field
- กรณีใช้เครื่อง EDC หรือทำการใช้ Biller ID อื่น ๆ ในการรับชำระแทน โดยปกติ เลข account จะเป็นเลขเดียวกันในทุก ๆ ร้านค้า แต่ผู้รับชำระนั้น ๆ ดังนั้นจะต้องมีการใช้ “ref1” ในการตรวจสอบแทนและร้านเจ้า ( เพราะว่าผู้รับชำระจะใช้เลขบัญชีเดียวกัน แต่ ref1 ต่างกันในการแยกร้านค้าแต่ละร้าน เช่น KPlus Shop หรือ SCB แม่นยำ )

### 2. ชื่อบัญชี โดยตรวจสอบจาก API field ดังนี้

- receiver.displayName** ชื่อผู้รับภาษาอังกฤษ
- receiver.name** ชื่อผู้รับภาษาไทย

หมายเหตุ

- ทั้ง 2 fields นี้สามารถเป็นไปได้ทั้งภาษาไทย และ ภาษาอังกฤษ ขึ้นอยู่กับธนาคารว่าส่งอะไรกลับมา
- อาจจะมีข้อมูลแค่ 1 field จาก 2 field เช่นมี displayName แต่ไม่มี name หรือมี name แต่ไม่มี displayName
- มีการตัดทอนบางส่วนออกโดยต้องใช้วิธี substring check แทน เช่น “นาย สวัสดิ์ ครับ” อาจจะได้รับมาเป็น “นาย สวัสดิ์ ค” ก็ต้องเช็คเฉพาะบางส่วน แทนที่จะเช็คว่าตรงกับ “นาย สวัสดิ์ ครับ”

## Bank Code

รหัสธนาคาร	ชื่อย่อ	ชื่อเต็ม
002	BBL	ธนาคารกรุงเทพ
004	KBANK	ธนาคารกสิกรไทย
006	KTB	ธนาคารกรุงไทย
011	TTB	ธนาคารทหารไทยธนชาต
014	SCB	ธนาคารไทยพาณิชย์
025	BAY	ธนาคารกรุงศรีอยุธยา
069	KKP	ธนาคารเกียรตินาคินภัทร
022	CIMBT	ธนาคารซีไอเอ็มบีไทย
067	TISCO	ธนาคารทิสโก้
024	UOBT	ธนาคารยูโอบี
071	TCD	ธนาคารไทยเครดิตเพื่อรายย่อย
073	LHFG	ธนาคารแลนด์ แอนด์ เฮ้าส์
070	ICBCT	ธนาคารไอซีบีซี (ไทย)
098	SME	ธนาคารพัฒนาวิสาหกิจขนาดกลางและขนาดย่อมแห่งประเทศไทย
034	BAAC	ธนาคารเพื่อการเกษตรและสหกรณ์การเกษตร
035	EXIM	ธนาคารเพื่อการส่งออกและนำเข้าแห่งประเทศไทย
030	GSB	ธนาคารออมสิน
033	GHB	ธนาคารอาคารสงเคราะห์

## Error Status Code

Error Code	รายละเอียด
1000	กรุณาใส่ข้อมูล QR Code ให้ครบใน field data, files หรือ url
1001	ไม่พบข้อมูลสาขา กรุณาตรวจสอบไอเดียสาขา
1002	Authorization Header ไม่ถูกต้อง
1003	Package ของคุณหมดอายุแล้ว
1004	Package ของคุณใช้เกินគอต้ามานาแล้ว 400 บาท กรุณารอสมาชิกแพ็คเกจ
1005	ไฟล์ไม่ใช่ไฟล์ภาพ กรุณาอัปโหลดไฟล์เฉพาะนามสกุล .jpg .jpeg .png .jfif หรือ .webp
1006	รูปภาพไม่ถูกต้อง
1007	รูปภาพไม่มี QR Code
1008	QR ตั้งกล่าวไม่ใช่ QR สำหรับการตรวจสอบการชำระเงิน
1009	ขออภัยในความไม่สะดวก ขณะนี้ข้อมูลธนาคารเกิดขัดข้องชั่วคราว โปรดตรวจสอบใหม่อีกครั้งใน 15 นาทีถัดไป (ไม่เสียគอต้าสลิป)
1010	เนื่องจากเป็นสลิปจากธนาคาร{{ชื่อบанک}} กรุณารอการตรวจสอบสลิปหลังการโอนประมาณ {จำนวนนาที} นาที
1011	QR Code หมดอายุ หรือ ไม่มีรายการอยู่จริง
1012	สลิปเข้า สลิปนี้เคยส่งเข้ามาในระบบเมื่อ {timestamp}
1013	ยอดที่ส่งมาไม่ตรงกับยอดสลิป
1014	บัญชีผู้รับไม่ตรงกับบัญชีหักของร้าน

## Request / Response Example

✓ Correct QR Verification / ✓ Valid Amount

```
Header: { x-authorization: <YOUR_API_KEY> }

Request Body
{
  "data": "0041000600000101030040220013071152533APM077365102TH91048134",
  "log": true,
}
```

HTTP Status: 200

Response Body

```
{
  "success": true,
  "data": {
    "success": true,
    "message": "✓",
    "rqUID": "783_20191108_v4UIS1K2Mobile",
    "language": "TH",
    "transRef": "010092101507665143",
    "sendingBank": "004",
    "receivingBank": "004",
    "transDate": "20200401",
    "transTime": "10:15:07",
    "sender": {
      "displayName": "นาย ธนาкар ก",
      "name": "Mr. Thanakarn K",
      "proxy": {
        "type": null,
        "value": null
      },
      "account": {
        "type": "BANKAC",
        "value": "xxx-x-x0209-x"
      }
    },
    "log": true
  }
}
```

```
"receiver": {  
    "displayName": "กสิกอร ร",  
    "name": "KASIKORN R",  
    "proxy": {  
        "type": "",  
        "value": ""  
    },  
    "account": {  
        "type": "BANKAC",  
        "value": "xxx-x-x3109-x"  
    },  
    "amount": 1,  
    "paidLocalAmount": 1,  
    "paidLocalCurrency": "764",  
    "countryCode": "TH",  
    "transFeeAmount": 0,  
    "ref1": "",  
    "ref2": "",  
    "ref3": "",  
    "toMerchantId": ""  
},  
}
```

## Invalid QR Verification #1

```
Header: { x-authorization: <YOUR_API_KEY> }
```

```
Request Body
```

```
{  
  "data": "ABCD",  
  "log": true,  
}
```

```
HTTP Status: 400
```

```
Response Body
```

```
{  
  "code": 1008,  
  "message": "QR Code ไม่ใช่ QR สำหรับตรวจสอบการชำระเงิน"  
}
```

## Invalid QR Verification #2

```
Header: { x-authorization: <YOUR_API_KEY> }

Request Body
{
  "data": "0041000600000101030040220013071152533APM077365102TH91048134",
  "log": true,
}
```

```
HTTP Status: 400

Response Body
{
  "code": 1007,
  "message": "QR Code หมดอายุ หรือ \"ไม่มีรายการอยู่จริง\""
}
```

### Invalid QR Verification #3

```
HTTP Status: 400
Response Body
{
  "code": 1007,
  "message": "รูปภาพไม่มี Qr code"
}
```

### Invalid Image File

```
HTTP Status: 400
Response Body
{
  "code": 1006,
  "message": "รูปภาพไม่ถูกต้อง",
}
```

### Invalid API Key or Request with No API Key in Header

```
HTTP Status: 401
Response Body
{
  "code": 1002,
  "message": "Authorization Header ไม่ถูกต้อง",
}
```

## Empty or Invalid Request Body

```
HTTP Status: 400
Response Body
{
  "code": 1000,
  "message": "กรุณาใส่ข้อมูล QR Code ให้ครบใน field data หรือ files",
}
```

## Invalid Amount (If user attach amount with the request)

```
Header: { x-authorization: <YOUR_API_KEY> }
Request Body
{
  "data": "0041000600000101030040220123456789ABCDEFGHJK5102TH91044C2D",
  "amount": 100,
  "log": true,
}
```

```
HTTP Status: 400
Response Body
{
  "code": 1013,
  "message": "ยอดที่ส่งมาไม่ตรงกับยอดลิบ",
  "data": {
    "success": true,
    "message": "✓",
    "rqUID": "783_20191108_v4UIS1K2Mobile",
    "language": "TH",
    "transRef": "010092101507665143",
    "sendingBank": "004",
    "receivingBank": "004",
    "transDate": "20200401",
    "transTime": "10:15:07",
    "sender": {
      "displayName": "นาย ธนาคาร ก",
      "name": "Mr. Thanakarn K",
    }
  }
}
```

```
"proxy": {
    "type": null,
    "value": null
},
"account": {
    "type": "BANKAC",
    "value": "xxx-x-x0209-x"
}
},
"receiver": {
    "displayName": "ກສີກອນ ວ",
    "name": "KASIKORN R",
    "proxy": {
        "type": "",
        "value": ""
    },
    "account": {
        "type": "BANKAC",
        "value": "xxx-x-x3109-x"
    }
},
"amount": 1,
"paidLocalAmount": 1,
"paidLocalCurrency": "764",
"countryCode": "TH",
"transFeeAmount": 0,
"ref1": "",
"ref2": "",
"ref3": "",
"toMerchantId": ""
}
}
```

## Invalid Destination Bank (If user attach log with the request)

```
Header: { x-authorization: <YOUR_API_KEY> }

Request Body
{
  "data": "0041000600000101030040220123456789ABCDEFGHIJK5102TH91044C2D",
  "log": true,
}

HTTP Status: 400

Response Body
{
  "code": 1014,
  "message": "บัญชีผู้รับไม่ตรงกับบัญชีหลักของร้าน",
  "data": {
    "success": true,
    "message": "✓",
    "rqUID": "783_20191108_v4UIS1K2Mobile",
    "language": "TH",
    "transRef": "010092101507665143",
    "sendingBank": "004",
    "receivingBank": "004",
    "transDate": "20200401",
    "transTime": "10:15:07",
    "sender": {
      "displayName": "นาย ธนาภาณ ก",
      "name": "Mr. Thanakarn K",
      "proxy": {
        "type": null,
        "value": null
      },
      "account": {
        "type": "BANKAC",
        "value": "xxx-x-x0209-x"
      }
    },
    "receiver": {
      "displayName": "กสิกอร ร",
      "name": "KASIKORN R",
    }
  }
}
```

```
"proxy": {  
    "type": "",  
    "value": ""  
},  
"account": {  
    "type": "BANKAC",  
    "value": "xxx-x-x3109-x"  
}  
,  
"amount": 1,  
"paidLocalAmount": 1,  
"paidLocalCurrency": "764",  
"countryCode": "TH",  
"transFeeAmount": 0,  
"ref1": "",  
"ref2": "",  
"ref3": "",  
"toMerchantId": ""  
}  
}
```

Repeat Slip (If user attach log with the request and the slip is already sent - slip data will also be provided)

```
Header: { x-authorization: <YOUR_API_KEY> }  
Request Body  
{  
    "data": "0041000600000101030040220123456789ABCDEFGHIJK5102TH91044C2D",  
    "log": true,  
}  
  
HTTP Status: 400  
Response Body  
{  
    "code": 1012,  
    "message": "สลิปช้า สลิปนี้เคยส่งเข้ามาในระบบเมื่อ 2023-09-09 09:09:09",  
    "data": {  
        "success": true,  
    }  
}
```

```
"message": "✓",
"rqUID": "783_20191108_v4UIS1K2Mobile",
"language": "TH",
"transRef": "010092101507665143",
"sendingBank": "004",
"receivingBank": "004",
"transDate": "20200401",
"transTime": "10:15:07",
"sender": {
    "displayName": "นาย ธนาภาณ ก",
    "name": "Mr. Thanakarn K",
    "proxy": {
        "type": null,
        "value": null
    },
    "account": {
        "type": "BANKAC",
        "value": "xxx-x-x0209-x"
    }
},
"receiver": {
    "displayName": "กสิกอร ร",
    "name": "KASIKORN R",
    "proxy": {
        "type": "",
        "value": ""
    },
    "account": {
        "type": "BANKAC",
        "value": "xxx-x-x3109-x"
    }
},
"amount": 1,
"paidLocalAmount": 1,
"paidLocalCurrency": "764",
"countryCode": "TH",
"transFeeAmount": 0,
"ref1": "",
"ref2": ""
```

```
    "ref3": "",  
    "toMerchantId": ""  
}  
}  
}
```

## Bank Delay Slip (For BBL & SCB)

**\*\* You need to recheck this slip again to validate the result based on delay (mins) \*\***

Header: { x-authorization: <YOUR\_API\_KEY> }

Request Body

```
{  
  "data": "0041000600000101030040220123456789ABCDEFGHIJK5102TH91044C2D",  
  "log": true,  
}
```

HTTP Status: 400

Response Body

```
{  
  "code": 1010,  
  "message": "เนื่องจากเป็นสลิปจากธนาคารกรุงเทพ กรุณารอการตรวจสอบสลิปหลังการโอนประมาณ 8  
  นาที",  
  "data": {  
    "qrcodeData":  
      "0041000600000101030040220123456789ABCDEFGHIJK5102TH91044C2D,  
      "bankCode": "002",  
      "bankName": "กรุงเทพ",  
      "delay": 8,  
    }  
}
```

## NODE.js Example

### File Example

```
const axios = require("axios");
const fs = require("fs");

const testFile = async () => {
  try {
    const branchId = "<YOUR_BRANCH_ID>";
    const apiKey = "<YOUR_API_KEY>";
    const path = "<YOUR_IMAGE_FILE_PATH>";

    const buffer = fs.readFileSync(path);

    const res = await axios.post(
      `https://api.slipok.com/api/line/apikey/${branchId}`,
      {
        files: buffer,
        log: true,
        // amount: number, // Add this to check with amount of the slip
      },
      {
        headers: {
          "x-authorization": apiKey,
          "Content-Type": "multipart/form-data",
        },
      }
    );
    // Handle success slip
    const slipData = res.data.data;
    console.log(slipData);
  } catch (err) {
    // Handle invalid slip
    if (axios.isAxiosError(err)) {
      const errorData = err.response.data;
      console.log(errorData.code); // Check error code
      console.log(errorData.message); // Check error message
      return;
    }
    console.log(err);
  }
};

testFile();
```

## Data Example

```
const axios = require("axios");

const testData = async () => {
  try {
    const branchId = "<YOUR_BRANCH_ID>";
    const apiKey = "<YOUR_API_KEY>";
    const qrString = "<YOUR_QR_STRING>";

    const res = await axios.post(
      `https://api.slipok.com/api/line/apikey/${branchId}`,
      {
        data: qrString,
        log: true,
        // amount: number, // Add this to check with amount of the slip
      },
      {
        headers: {
          "x-authorization": apiKey,
        },
      }
    );
    // Handle success slip
    const slipData = res.data.data;
    console.log(slipData);
  } catch (err) {
    // Handle invalid slip
    if (axios.isAxiosError(err)) {
      const errorData = err.response.data;
      console.log(errorData.code); // Check error code
      console.log(errorData.message); // Check error message
      return;
    }
    console.log(err);
  }
};

testData();
```

## Url Example

```
const axios = require("axios");

const testUrl = async () => {
  try {
    const branchId = "<YOUR_BRANCH_ID>";
    const apiKey = "<YOUR_API_KEY>";
    const url = "<YOUR_IMAGE_URL>";

    const res = await axios.post(
      `https://api.slipok.com/api/line/apikey/${branchId}`,
      {
        url,
        log: true,
        // amount: number, // Add this to check with amount of the slip
      },
      {
        headers: {
          "x-authorization": apiKey,
        },
      }
    );
    // Handle success slip
    const slipData = res.data.data;
    console.log(slipData);
  } catch (err) {
    // Handle invalid slip
    if (axios.isAxiosError(err)) {
      const errorData = err.response.data;
      console.log(errorData.code); // Check error code
      console.log(errorData.message); // Check error message
      return;
    }
    console.log(err);
  }
};

testUrl();
```

## cURL Example

### File Example

```
curl --location --request  
'http://api.slipok.com/api/line/apikey/<YOUR_BRANCH_ID>' \  
--header 'x-authorization: <YOUR_KEY>' \  
--form 'files=@<YOUR_FILE_LOCATION>'' \  
--form 'log=true'
```

### Data Example

```
curl --location --request  
'http://api.slipok.com/api/line/apikey/<YOUR_BRANCH_ID>' \  
--header 'x-authorization: <YOUR_KEY>' \  
--header 'Content-Type: application/json' \  
--data '{  
    "data": "<YOUR_QR_DATA>",  
    "log": true  
}'
```

### Url Example

```
curl --location --request  
'http://api.slipok.com/api/line/apikey/<YOUR_BRANCH_ID>' \  
--header 'x-authorization: <YOUR_KEY>' \  
--header 'Content-Type: application/json' \  
--data '{  
    "url": "<YOUR_IMAGE_URL>",  
    "log": true  
}'
```

## PHP Example

### File Example

```
<?php

$branchId = <YOUR_BRANCH_ID>;
$apiKey = '<YOUR_API_KEY>';
$file = '<YOUR_IMAGE_FILE>';

$url = 'https://api.slipok.com/api/line/apikey/' . $branchId;

$headers = [
    'Content-Type: multipart/form-data',
    'x-authorization: ' . $apiKey
];

$fields = [
    'files' => new CURLFile($file),
    'log' => true,
    // 'amount' => amount, // Add this to check with amount of the slip
];
curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => $url,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_FOLLOWLOCATION => true,
    CURLOPT_POST => 1,
    CURLOPT_POSTFIELDS => $fields,
    CURLOPT_HTTPHEADER => $headers,
));

$response = curl_exec($curl);

curl_close($curl);

if($response){
    $json_response = json_decode($response);
    $http_code = curl_getinfo($curl, CURLINFO_HTTP_CODE);

    if($http_code === 200 && $json_response->{'success'} === true) {
        // Handle correct slip
        echo(json_encode($json_response->{'data'})); // Slip data
    } else {
        // Handle incorrect slip
        echo 'code: '.$json_response->{'code'}; // Error code
        echo '<br/>';
        echo 'message: '.$json_response->{'message'}; // Error message
    }
} else {
    echo curl_errno($curl); // Curl error no
    echo curl_error($curl); // Curl error message
}
```

## Data Example

```
<?php

$branchId = <YOUR_BRANCH_ID>;
$apiKey = '<YOUR_API_KEY>';
$data = '<YOUR_QR_DATA>';

$url = 'https://api.slipok.com/api/line/apikey/' . $branchId;

$headers = [
    'Content-Type: multipart/form-data',
    'x-authorization: ' . $apiKey
];

$fields = [
    'data' => $data,
    'log' => true,
    // 'amount' => amount, // Add this to check with amount of the slip
];

$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => $url,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_FOLLOWLOCATION => true,
    CURLOPT_POST => 1,
    CURLOPT_POSTFIELDS => $fields,
    CURLOPT_HTTPHEADER => $headers,
));

$response = curl_exec($curl);

curl_close($curl);

if($response){
    $json_response = json_decode($response);
    $http_code = curl_getinfo($curl, CURLINFO_HTTP_CODE);

    if($http_code === 200 && $json_response->{'success'} === true) {
        // Handle correct slip
        echo(json_encode($json_response->{'data'})); // Slip data
    } else {
        // Handle incorrect slip
        echo 'code: '.$json_response->{'code'}; // Error code
        echo '<br/>';
        echo 'message: '.$json_response->{'message'}; // Error message
    }
} else {
    echo curl_errno($curl); // Curl error no
    echo curl_error($curl); // Curl error message
}
```

## Url Example

```
<?php

$branchId = <YOUR_BRANCH_ID>;
$apiKey = '<YOUR_API_KEY>';
$imageUrl = '<YOUR_IMAGE_URL>';

$url = 'https://api.slipok.com/api/line/apikey/' . $branchId;

$headers = [
    'Content-Type: multipart/form-data',
    'x-authorization: ' . $apiKey
];

$fields = [
    'url' => $imageUrl,
    'log' => true,
    // 'amount' => amount, // Add this to check with amount of the slip
];

$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => $url,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_FOLLOWLOCATION => true,
    CURLOPT_POST => 1,
    CURLOPT_POSTFIELDS => $fields,
    CURLOPT_HTTPHEADER => $headers,
));

$response = curl_exec($curl);

curl_close($curl);

if($response){
    $json_response = json_decode($response);
    $http_code = curl_getinfo($curl, CURLINFO_HTTP_CODE);

    if($http_code === 200 && $json_response->{'success'} === true) {
        // Handle correct slip
        echo(json_encode($json_response->{'data'})); // Slip data
    } else {
        // Handle incorrect slip
        echo 'code: '.$json_response->{'code'}; // Error code
        echo '<br/>';
        echo 'message: '.$json_response->{'message'}; // Error message
    }
} else {
    echo curl_errno($curl); // Curl error no
    echo curl_error($curl); // Curl error message
}
```

## Apps Script Example

### Google Drive File Example

```
function testGoogleDriveFile() {
  try {
    const branchId = "<YOUR_BRANCH_ID>";
    const apiKey = "<YOUR_API_KEY>";
    const drive fileId = "<YOUR_GOOGLE_DRIVE_FILE_ID>";

    const file = DriveApp.getFileById(drive fileId);
    const imageBlob = file.getBlob();
    const imageBytes = imageBlob.getBytes();
    const base64 = Utilities.base64Encode(imageBytes);

    const payload = {
      files: base64,
      log: true,
      // amount: number, // Add this to check with amount of the slip
    };

    const options = {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
        "x-authorization": apiKey,
      },
      payload: JSON.stringify(payload),
    };

    const response = UrlFetchApp.fetch(
      `https://api.slipok.com/api/line/apikey/${branchId}`,
      options
    );

    // Check if the request was successful
    if (response.getResponseCode() === 200) { // Handle success slip
      const responseData = JSON.parse(response.getContentText());
      const slipData = responseData.data;
      console.log(slipData);
      return slipData;
    } else { // Handle HTTP error responses
      const errorData = JSON.parse(response.getContentText());
      console.log("Error Code:", errorData.code);
      console.log("Error Message:", errorData.message);
      return null;
    }
  } catch (err) { // Handle invalid slip or other errors
    console.log("Error occurred:", err.toString());
    return null;
  }
}
```

## Url Example

```
function testUrl() {
  try {
    const branchId = "<YOUR_BRANCH_ID>";
    const apiKey = "<YOUR_API_KEY>";
    const url = "<YOUR_IMAGE_URL>";

    const payload = {
      url,
      log: true,
      // amount: number, // Add this to check with amount of the slip
    };

    const options = {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
        "x-authorization": apiKey,
      },
      payload: JSON.stringify(payload),
    };

    const response = UrlFetchApp.fetch(
      `https://api.slipok.com/api/line/apikey/${branchId}`,
      options
    );

    // Check if the request was successful
    if (response.getResponseCode() === 200) {
      // Handle success slip
      const responseData = JSON.parse(response.getContentText());
      const slipData = responseData.data;
      console.log(slipData);
      return slipData;
    } else {
      // Handle HTTP error responses
      const errorData = JSON.parse(response.getContentText());
      console.log("Error Code:", errorData.code);
      console.log("Error Message:", errorData.message);
      return null;
    }
  } catch (err) {
    // Handle invalid slip or other errors
    console.log("Error occurred:", err.toString());
    return null;
  }
}
```

## **Change Logs**

### **Version 1.12 (8 August 2025)**

- Add Google Drive file Apps Script example

### **Version 1.11 (19 June 2025)**

- Add Webhook forwarding requirement and Apps Script example

### **Version 1.10 (28 May 2025)**

- Add base64 support & header details

### **Version 1.9 (19 April 2025)**

- Add compare receiver account details

### **Version 1.8 (30 July 2024)**

- Add .jfif and .webp image support

### **Version 1.7 (26 March 2024)**

- Update Node & PHP Example

### **Version 1.6 (17 March 2024)**

- Update PHP Example

### **Version 1.5 (15 March 2024)**

- Add full English version

### **Version 1.4 (14 February 2024)**

- Add slip data for error code 1012 repeated slip.
- Add bankCode, bankName and delay for error code 1010 delay slip.

### **Version 1.3 (9 October 2023)**

- Add check quota route to check the remaining quota.

### **Version 1.2 (14 September 2023)**

- Add log parameter to log slip data and analyze in LINE LIFF dashboard.
  - Be able to check the receiver associated with the bank account registered in Line LIFF. Wrong account will trigger error code 1014.
  - Repeated slip sent will trigger error code 1012.

- Add amount parameter to cross-check with slip amount. Wrong amount will trigger error code 1013
- New error code
  - 1012: Repeated Slip
  - 1013: Wrong Slip Amount
  - 1014: Wrong Receiver Slip

### **Version 1.1 (16 August 2023)**

- Add error code for better integration.

### **Version 1.0 (29 March 2023)**

- Initialize documents.
- Add Request & Response for check slip route
- Add examples for NODE.js / CURL / PHP.