

SMSTH Public API v1

เชื่อมระบบของคุณ (เว็บไซต์ ร้านค้าออนไลน์ POS CRM แอป) เข้ากับ SMSTH เพื่อส่ง SMS แจ้งเตือน ส่ง OTP ยืนยันตัวตน และยิงแคมเปญ โดยใช้เครดิตเดียวกับบัญชีบนเว็บ **1 เครดิต = 1 ข้อความ**

Base URL: `https://likesms.me/v1` · รูปแบบข้อมูล: JSON (UTF-8) · HTTPS เท่านั้น

เริ่มต้น 5 นาที

1. เข้าสู่ระบบ likesms.me แล้วไปที่เมนู **API สำหรับนักพัฒนา**
2. กด **สร้าง API key** แล้วคัดลอกเก็บไว้ทันที (ระบบแสดงครั้งเดียว)
3. ตรวจสอบชื่อผู้ส่ง (Sender) ที่บัญชีใช้ได้ด้วย `GET /v1/senders`
4. ส่งข้อความแรก:

```
curl -X POST https://likesms.me/v1/sms/send \  
  -H "Authorization: Bearer YOUR_API_KEY" \  
  -H "Content-Type: application/json" \  
  -d '{"to":"0812345678","sender":"MYBRAND","message":"คำสั่งซื้อ #1024 จัดส่งแล้ว","ref":"order-1024"}'
```

5. เก็บ `id` ที่ได้กลับมา ใช้เช็คสถานะด้วย `GET /v1/messages/{id}` หรือรอ webhook แจ้งกลับ

เก็บ API key ไว้ฝั่งเซิร์ฟเวอร์เท่านั้น ห้ามใส่ในโค้ดหน้าเว็บ (JavaScript) หรือแอปมือถือ ถ้าหลุดให้กด **สร้าง key ใหม่** ทันที key เดิมจะใช้ไม่ได้

การยืนยันตัวตน

ส่ง API key ใน header ทุก request

```
Authorization: Bearer sk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

ถ้าระบบของคุณตั้ง header `Authorization` ไม่ได้ ใช้ `X-API-Key: sk_live_...` แทนได้

- 1 บัญชีมี 1 key · สร้างใหม่เมื่อไรก็ได้ (key เดิมหยุดทำงานทันที)

- ตั้ง **IP allow-list** ในหน้าเดียวกันได้ ถ้าตั้งไว้ จะรับเฉพาะ request จาก IP ในรายการ

รูปแบบ Response

สำเร็จ (HTTP 200):

```
{ "success": true, "data": { "...": "..." } }
```

ผิดพลาด (HTTP 4xx/5xx):

```
{ "success": false, "error": { "code": "INSUFFICIENT_CREDITS", "message": "เครดิตไม่เพียงพอ  
ต้องการ 50 เครดิต แต่มีเพียง 8 เครดิต", "details": { "required": 50, "available": 8 } } }
```

ให้ตรวจสอบ `error.code` ในโค้ด (ค่าคงที่) ส่วน `error.message` เป็นภาษาไทยสำหรับแสดงผล

Rate limit

- สูงสุด **60 request ต่อนาที** ต่อบัญชี
- ทุก response มี header `X-RateLimit-Limit` และ `X-RateLimit-Remaining`
- เกินกำหนดได้ HTTP 429 `RATE_LIMITED` พร้อม header `Retry-After` (วินาที)
- ส่งหลายเบอร์ให้ใช้ `to` แบบ array (สูงสุด 100 เบอร์ต่อ request) หรือ `POST /v1/sms/bulk` (สูงสุด 1,000 เบอร์)

Error codes

code	HTTP	ความหมาย / ควรทำอะไร
UNAUTHORIZED	401	ไม่มี key, key ผิด หรือบัญชีถูกระงับ
HTTPS_REQUIRED	403	เรียกผ่าน http:// — เปลี่ยนเป็น https://
IP_NOT_ALLOWED	403	IP ไม่อยู่ใน allow-list ของบัญชี
SENDER_NOT_ALLOWED	403	บัญชีไม่มีสิทธิใช้ sender นี้ — ดู <code>GET /v1/senders</code>
NOT_FOUND	404	ไม่พบ endpoint หรือข้อมูล (หรือเป็นของบัญชีอื่น)
METHOD_NOT_ALLOWED	405	ใช้ GET/POST ไม่ตรงกับ endpoint
VALIDATION_ERROR	422	ข้อมูลไม่ครบ/ผิดรูปแบบ ดู <code>message</code>
OTP_INVALID	422	รหัส OTP ผิดหรือหมดอายุ
INSUFFICIENT_CREDITS	402	เครดิตไม่พอ — เติมเครดิตที่เว็บ ระบบไม่หักเครดิตในกรณีนี้
RATE_LIMITED	429	เรียกถี่เกินไป รอตาม <code>Retry-After</code>
PROVIDER_ERROR	502	เครือข่ายปลายทางปฏิเสธ เช่น พบคำต้องห้าม (เครดิตถูกคืนอัตโนมัติ)
INTERNAL_ERROR	500	ระบบขัดข้องชั่วคราว ลองใหม่ภายหลัง

Endpoints

Method	Path	ใช้ทำอะไร
POST	/v1/sms/send	ส่ง SMS 1–100 เบอร์ ทันที
POST	/v1/sms/bulk	ส่งแคมเปญ 1–1,000 เบอร์ ทันทีหรือตั้งเวลา
POST	/v1/otp/send	ส่งรหัส OTP
POST	/v1/otp/verify	ตรวจรหัส OTP
GET	/v1/messages/{id}	สถานะข้อความ
GET	/v1/campaigns/{id}	สถานะแคมเปญ
GET	/v1/account/balance	เครดิตคงเหลือ
GET	/v1/senders	ชื่อผู้ส่งที่ใช้ได้

POST /v1/sms/send

ส่ง SMS ทันที หักเครดิตตามจำนวนเบอร์ เบอร์ที่ส่งไม่สำเร็จได้เครดิตคืนอัตโนมัติ

ฟิลด์	ชนิด	จำเป็น	รายละเอียด
to	string หรือ array	ใช่	เบอร์ไทย เช่น 0812345678 , 081-234-5678 , +66812345678 สูงสุด 100 เบอร์ (string คั่นด้วย , ได้)
sender	string	ใช่	ชื่อผู้ส่งที่บัญชีมีสิทธิ์
message	string	ใช่	ข้อความ สูงสุด 1,000 ตัวอักษร (ภาษาไทย 70 ตัวอักษรต่อ 1 SMS)
ref	string	ไม่	รหัสอ้างอิงของคุณ (≤64) จะส่งกลับใน webhook และ GET /v1/messages/{id}

```
{
  "to": ["0812345678", "12"],
  "sender": "MYBRAND",
  "message": "รหัสสมาชิกของคุณคือ A1024",
  "ref": "member-1024"
}
```

Response:

```
{
  "success": true,
  "data": {
    "messages": [
      { "id": 5012, "to": "0812345678", "status": "sending" },
      { "to": "12", "status": "failed", "error": "เบอร์โทรศัพท์ไม่ถูกต้อง: 12", "error_type":
"invalid_number" }
    ],
    "accepted": 1,
    "failed": 1,
    "credits_used": 1,
    "credits_remaining": 842
  }
}
```

ถ้าส่งไม่สำเร็จเลยสักเบอร์ จะได้ error (`VALIDATION_ERROR` เมื่อทุกเบอร์ผิดรูปแบบ หรือ `PROVIDER_ERROR`)
พร้อม `details.messages`

POST /v1/sms/bulk

สร้างแคมเปญ (แสดงในหน้า **แคมเปญ** บนเว็บด้วย) ส่งข้อความเดียวกันถึงทุกเบอร์ เบอร์ซ้ำถูกตัดอัตโนมัติ หัก
เครดิตตอนสร้าง ถ้าส่งไม่สำเร็จทั้งแคมเปญจะคืนเครดิตทั้งหมด

ฟิลด์	ชนิด	จำเป็น	รายละเอียด
<code>to</code>	array หรือ string	ใช่	1–1,000 เบอร์ทุกเบอร์ต้องถูกรูปแบบ
<code>sender</code>	string	ใช่	ชื่อผู้ส่ง
<code>message</code>	string	ใช่	≤1,000 ตัวอักษร
<code>name</code>	string	ไม่	ชื่อแคมเปญ (ไม่ใช่ = <code>API วันที่ เวลา</code>)
<code>schedule_at</code>	string	ไม่	ISO 8601 เช่น <code>2026-10-01T09:00:00+07:00</code> ต้องเป็นอนาคต ไม่เกิน 30 วัน ไม่ระบุ timezone = เวลาไทย
<code>ref</code>	string	ไม่	รหัสอ้างอิงของคุณ

```
{
  "to": ["0812345678", "0898765432"],
  "sender": "MYBRAND",
  "message": "โปรวันนี้ ลด 20% ทุกรายการ ถึง 22:00",
  "name": "โปรเย็นวันศุกร์",
  "schedule_at": "2026-10-02T17:00:00+07:00"
}
```

```
{
  "success": true,
  "data": {
    "campaign_id": 311,
    "name": "โปรเย็นวันศุกร์",
    "total": 2,
    "status": "scheduled",
    "scheduled_at": "2026-10-02T17:00:00+07:00",
    "credits_used": 2,
    "credits_remaining": 840
  }
}
```

`status` เป็น `sending` เมื่อส่งทันที หรือ `scheduled` เมื่อตั้งเวลา

POST /v1/otp/send

ส่งรหัส OTP (ระบบสร้างรหัสให้) 1 เบอร์ต่อครั้ง ใช้ 1 เครดิต

ฟิลด์	ชนิด	จำเป็น	รายละเอียด
<code>to</code>	string	ใช่	เบอร์ผู้รับ 1 เบอร์
<code>sender</code>	string	ใช่	ชื่อผู้ส่ง
<code>ref</code>	string	ไม่	รหัสอ้างอิงของคุณ

```
{
  "success": true,
  "data": {
    "message_id": 5013,
    "to": "0812345678",
    "token": "b7f1c2e0-...",
    "ref_code": "ABCD",
    "credits_used": 1,
    "credits_remaining": 839
  }
}
```

เก็บ `token` ไว้ฝั่งเซิร์ฟเวอร์ แสดง `ref_code` ให้ผู้ใช้เทียบกับ SMS ที่ได้รับ

POST /v1/otp/verify

ฟิลด์	ชนิด	จำเป็น	รายละเอียด
<code>token</code>	string	ใช่	token จาก <code>/v1/otp/send</code> (ต้องเป็นของบัญชีนี้)
<code>code</code>	string	ใช่	รหัสที่ผู้ใช้กรอก

```
{ "success": true, "data": { "verified": true, "message_id": 5013 } }
```

รหัสผิดหรือหมดอายุ: HTTP 422 `OTP_INVALID`

GET /v1/messages/{id}

```
{
  "success": true,
  "data": {
    "id": 5012,
    "to": "0812345678",
    "sender": "MYBRAND",
    "type": "plain",
    "status": "delivered",
    "ref": "member-1024",
    "created_at": "2026-09-25T14:03:05+07:00",
    "updated_at": "2026-09-25T14:03:11+07:00"
  }
}
```

status	ความหมาย
queued	รอคิวส่ง
sending	ส่งแล้ว รอรายงานการส่งถึง
delivered	ถึงเครื่องผู้รับแล้ว
failed	ส่งไม่สำเร็จ

GET /v1/campaigns/{id}

```
{
  "success": true,
  "data": {
    "id": 311, "name": "โปรเหีนวันศุกร์", "status": "sending", "ref": null,
    "total": 2, "delivered": 1, "failed": 0, "pending": 1,
    "scheduled_at": "2026-10-02T17:00:00+07:00", "created_at": "2026-09-25T14:10:00+07:00"
  }
}
```

status ของแคมเปญ: queued (รอเวลา), processing, sending, delivered (จบ), failed, cancelled

GET /v1/account/balance

```
{ "success": true, "data": { "credits": 839, "unit": "credit", "note": "1 เครดิต = 1 ข้อความ" } }
```

GET /v1/senders

```
{ "success": true, "data": { "senders": ["MYBRAND", "MYSHOP"], "default": "MYBRAND" } }
```

ต้องการซื้อผู้ส่งใหม่ ติดต่อทีมงานทาง LINE เพื่อยืนยันตัวตน

Webhook แจ้งสถานะ

ตั้ง URL ในหน้า **API สำหรับนักพัฒนา** (ต้องเป็น https) ระบบจะส่ง **POST** เมื่อข้อความที่ส่งผ่าน API มีสถานะสุดท้าย (**delivered** / **failed**) และเมื่อแคมเปญจาก API จบ

```
POST /webhooks/smsth HTTP/1.1
Content-Type: application/json
X-SMSTH-Event: message.status
X-SMSTH-Delivery: 9812
X-SMSTH-Signature: sha256=5d41402abc4b2a76b9719d911017c592...
```

```
{
  "event": "message.status",
  "message_id": 5012,
  "to": "0812345678",
  "status": "delivered",
  "ref": "member-1024",
  "sender": "MYBRAND",
  "timestamp": "2026-09-25T14:03:11+07:00"
}
```

แคมเปญจบ (**X-SMSTH-Event: campaign.status**):

```
{
  "event": "campaign.status", "campaign_id": 311, "name": "โปรเจกต์วันศุกร์", "status":
  "delivered", "ref": null,
  "total": 2, "delivered": 2, "failed": 0, "pending": 0, "timestamp": "2026-10-
  02T17:04:40+07:00"
}
```

กติกา:

- ตอบ HTTP 2xx ภายใน 10 วินาที = รับแล้ว (ระบบไม่ตาม redirect)
- ไม่ตอบ 2xx จะส่งซ้ำอีก 3 ครั้ง ห่าง 1 นาที, 5 นาที, 30 นาที แล้วหยุด
- ใช้ X-SMSTH-Delivery กันประมวลผลซ้ำ (ครั้งที่ส่งซ้ำใช้ค่าเดิม)
- ปุ่ม **ส่งทดสอบ** ในหน้าเว็บจะส่ง {"event":"test"} มาให้ลองรับก่อน

ตรวจสอบลายเซ็น

X-SMSTH-Signature = sha256= + HMAC-SHA256 ของ **body ดิบ** ด้วย webhook secret ของคุณ
 ตรวจสอบทุกครั้งก่อนเชื่อข้อมูล

```
<?php
$secret = getenv('SMSTH_WEBHOOK_SECRET');
$body = file_get_contents('php://input');
$expected = 'sha256=' . hash_hmac('sha256', $body, $secret);
if (!hash_equals($expected, $_SERVER['HTTP_X_SMSTH_SIGNATURE'] ?? '')) {
    http_response_code(401);
    exit;
}
$event = json_decode($body, true);
// บันทึกสถานะ $event['message_id'] / $event['ref'] ในระบบของคุณ
http_response_code(200);
```

```
// Node.js + Express
const crypto = require('crypto');
app.post('/webhooks/smith', express.raw({ type: 'application/json' })), (req, res) => {
  const expected = 'sha256=' + crypto.createHmac('sha256',
process.env.SMSTH_WEBHOOK_SECRET).update(req.body).digest('hex');
  const got = req.get('X-SMSTH-Signature') || '';
  if (got.length !== expected.length || !crypto.timingSafeEqual(Buffer.from(got),
Buffer.from(expected))) return res.sendStatus(401);
  const event = JSON.parse(req.body);
  res.sendStatus(200);
});
```

```
# Python + Flask
import hmac, hashlib, os
from flask import Flask, request, abort
app = Flask(__name__)

@app.post("/webhooks/smith")
def smith_webhook():
  body = request.get_data()
  expected = "sha256=" + hmac.new(os.environ["SMSTH_WEBHOOK_SECRET"].encode(), body,
hashlib.sha256).hexdigest()
  if not hmac.compare_digest(expected, request.headers.get("X-SMSTH-Signature", "")):
    abort(401)
  event = request.get_json()
  return "", 200
```

ตัวอย่างโค้ดส่ง SMS

```
<?php
$ch = curl_init('https://likesms.me/v1/sms/send');
curl_setopt_array($ch, [
    CURLOPT_POST => true,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTPHEADER => [
        'Authorization: Bearer ' . getenv('SMSTH_API_KEY'),
        'Content-Type: application/json',
    ],
    CURLOPT_POSTFIELDS => json_encode([
        'to' => '0812345678',
        'sender' => 'MYBRAND',
        'message' => 'คำสั่งซื้อ #1024 จัดส่งแล้ว',
        'ref' => 'order-1024',
    ], JSON_UNESCAPED_UNICODE),
]);
$res = json_decode(curl_exec($ch), true);
if (!$res['success']) {
    error_log('SMSTH ' . $res['error']['code'] . ': ' . $res['error']['message']);
}
```

```
// Node.js 18+
const res = await fetch('https://likesms.me/v1/sms/send', {
    method: 'POST',
    headers: { Authorization: `Bearer ${process.env.SMSTH_API_KEY}`, 'Content-Type':
    'application/json' },
    body: JSON.stringify({ to: '0812345678', sender: 'MYBRAND', message: 'คำสั่งซื้อ #1024 จัดส่ง
    แล้ว', ref: 'order-1024' }),
});
const json = await res.json();
if (!json.success) console.error(json.error.code, json.error.message);
```

```
# Python 3 + requests
import os, requests
r = requests.post(
    "https://likesms.me/v1/sms/send",
    headers={"Authorization": f"Bearer {os.environ['SMSTH_API_KEY']}"},
    json={"to": "0812345678", "sender": "MYBRAND", "message": "คำสั่งซื้อ #1024 จัดส่งแล้ว",
    "ref": "order-1024"},
    timeout=30,
)
data = r.json()
if not data["success"]:
    print(data["error"]["code"], data["error"]["message"])
```

ตัวอย่าง OTP ครบวงจร:

```
# 1) ส่ง OTP
curl -X POST https://likesms.me/v1/otp/send -H "Authorization: Bearer $SMSTH_API_KEY" \
-H "Content-Type: application/json" -d '{"to":"0812345678","sender":"MYBRAND"}'

# 2) ผู้ใช้กรอกรหัส → ตรวจ
curl -X POST https://likesms.me/v1/otp/verify -H "Authorization: Bearer $SMSTH_API_KEY" \
-H "Content-Type: application/json" -d '{"token":"TOKEN_จากข้อ1","code":"123456"}'
```

แนวปฏิบัติที่แนะนำ

- ตั้ง timeout ผังคุณ 30 วินาที และ retry เฉพาะ `RATE_LIMITED`, `PROVIDER_ERROR`, `INTERNAL_ERROR` แบบเว้นระยะ
- อย่า retry `VALIDATION_ERROR`, `SENDER_NOT_ALLOWED`, `INSUFFICIENT_CREDITS` — แก้ข้อมูลก่อน
- ใส่ `ref` เป็นรหัสคำสั่งซื้อ/สมาชิกของคุณ เพื่อจับคู่สถานะจาก webhook ได้ง่าย
- เช็ค `credits_remaining` ใน response แล้วแจ้งเตือนทีมเมื่อต่ำ
- หลีกเลี่ยงคำต้องห้าม/ลิงก์จากบริการภายนอก เครือข่ายอาจปฏิเสธ (`PROVIDER_ERROR` + คืนเครดิต)

คำถามพบบ่อย

ส่งแล้วหักเครดิตเท่าไร? 1 เครดิตต่อเบอร์ต่อครั้ง เบอร์ที่ส่งไม่สำเร็จคืนอัตโนมัติ

ข้อความยาวคิดอย่างไร? ระบบคิด 1 เครดิตต่อเบอร์ต่อคำขอ ข้อความยาวอาจถูกเครือข่ายแบ่งเป็นหลาย SMS ที่ปลายทาง แนะนำภาษาไทยไม่เกิน 70 ตัวอักษร

มี sandbox ใหม่? v1 ยังไม่มี ทดสอบด้วยเบอร์ของคุณเองและดูประวัติในเมนู **ประวัติ SMS** (ข้อความจาก API จะแสดงรวมกัน)

ข้อความที่ส่งผ่าน API เห็นบนเว็บไหม? เห็น — ใน **ประวัติ SMS** และแคมเปญจาก `/v1/sms/bulk` อยู่ในหน้า **แคมเปญ**

ต้องการ sender ใหม่ / วอลุ่มสูง / ใบเสนอราคา? ติดต่อทีมงานทาง LINE จากหน้าแรก likesms.me

Changelog

- **v1 (2026-09-25)** เปิดตัว: sms/send, sms/bulk (ตั้งเวลาได้), otp/send, otp/verify, messages, campaigns, account/balance, senders, webhook พร้อมลายเซ็น HMAC

อัปเดตล่าสุด 25/09/2026 · © 2026 SMSTH