

คู่มือการตั้งค่าและทดสอบระบบแจ้งเตือน

LINE & Telegram สำหรับ GNSS Monitoring System

หน้า “SETTING AND TESTER” ปัจจุบัน

SETTING AND TESTER การแจ้งเตือนอัตโนมัติ LINE & TELEGRAM

ตั้งค่าช่องทางแจ้งเตือนการทำงานของระบบ (LINE / Telegram) และทดสอบการเชื่อมต่อ — ค่า credential เก็บฝั่ง server เท่านั้น

การแจ้งเตือน (LINE & Telegram)

LINE Notification ☒ เปิดใช้งาน

Telegram Notification ☒ เปิดใช้งาน

บันทึกการเปิดใช้งานการแจ้งเตือน ☐ คลิกเลือกช่องทางที่ต้องการเปิดใช้งานจริงแล้วกดบันทึก — ใช้แจ้งเตือนจริงทันที (ไม่ต้องอัปเดต credential)

สถานะแยก Enabled (เปิดใช้) กับ Configured (มี credential) — เปิดใช้งาน ปิดใช้งาน

สถานะระบบปัจจุบัน (อัปเดตทุก refresh cycle)

DEVICE	STATUS	LAST SEEN	AGE	FIX	SATS w/u
GNSS01	 ONLINE	15 09 2026 16:20:00+07	113 s	VALID	15/18

☒ ระบบ GNSS01 อยู่ในสถานะ ONLINE (ปกติ) — แจ้งเตือนสถานะ ONLINE ด้วย

LINE	Telegram	LINE ฟื้นฟู config	Telegram ฟื้นฟู config	อัปเดตล่าสุด
● RZra_U = U2b1.22	● 8714_s4 = 8951.70			15 09 2026 16:03:11+07

กรอกค่า LINE / Telegram ที่ใช้งานจริง (อัปเดตสู่ server)

กรอกครบ 4 ช่องแล้วกด อัปเดต credentials + บันทึกการตั้งค่า — ระบบเก็บค่าล่าสุดไว้ใน server (DB) เพื่อใช้แบบอัตโนมัติตาม; ช่องที่เว้นว่าง = เก็บค่าเดิมไว้; ปุ่ม ล้าง = คืนค่าไปใช้ config/notifications.php — ค่าเป็น secret ฝั่ง server ไม่แสดงค่าเดิมกลับมา

LINE_CHANNEL_ACCESS_TOKEN LINE channel access token LINE_TARGET_ID LINE target user/group ID TELEGRAM_BOT_TOKEN Telegram bot token TELEGRAM_CHAT_ID Telegram chat ID

LINE ON · Telegram ON · การตั้งค่าเปลี่ยนล่าสุด 15 09 2026 16:07:20+07 · Auto-notify (web autopilot) = ON ตลอดที่เปิดหน้าเว็บนี้

ภาพอ้างอิงหน้า Setting and Tester ที่ใช้เป็นฐานในการปรับคู่มือฉบับนี้

ขอบเขตของคู่มือฉบับนี้

ใช้สำหรับระบบที่สร้าง LINE Official Account / Messaging API และ Telegram Bot ไว้แล้ว

โดยผู้ดูแลจัดการการเปิดใช้งาน, อัปเดต credentials, ทดสอบการเชื่อมต่อ และตรวจสอบสถานะจากหน้าเว็บนี้เป็นหลัก

ไม่จำเป็นต้องสร้างไฟล์ PHP ทดสอบแยกสำหรับงานปกติ

หมายเหตุด้านความปลอดภัย: คู่มือฉบับปรับปรุงจะไม่แสดง Channel Access Token, Bot Token หรือ Credential แบบเต็ม เนื่องจากหน้าเว็บระบุชุดว่าค่า secret เก็บฝั่ง server เท่านั้น

1. ภาพรวมการทำงานของหน้า Setting and Tester

หน้าเดียวสำหรับเปิด/ปิดช่องทาง, ตรวจสอบสถานะ GNSS, จัดการ credentials และทดสอบ LINE / Telegram

หน้าเว็บปัจจุบันแบ่งการทำงานออกเป็น 4 ส่วนหลัก ได้แก่ การเปิดใช้งานช่องทางแจ้งเตือน, สถานะ GNSS ปัจจุบัน, สถานะ credential ของแต่ละช่องทาง และชุดคำสั่งอัปโหลด/ล้าง/ทดสอบ

ส่วนบนหน้าเว็บ	หน้าที่
Enable / Disable	เลือก LINE Notification และ/หรือ Telegram Notification แล้วกดบันทึก เพื่อให้การตั้งค่ามีผลทันที
Current GNSS Status	แสดง Device, STATUS, LAST SEEN, AGE, FIX และ SATS v/u โดยอัปเดตตาม refresh cycle
Configured status	แสดงว่าช่องทางมี credential ใช้งานได้หรือไม่ พร้อมค่าที่ mask ไว้และแหล่งที่มาของค่า เช่น config
Credential upload	รับ 4 ค่า ได้แก่ LINE_CHANNEL_ACCESS_TOKEN, LINE_TARGET_ID, TELEGRAM_BOT_TOKEN และ TELEGRAM_CHAT_ID แล้วเก็บค่าล่าสุดไว้ฝั่ง server
Tester actions	ทดสอบ LINE, ทดสอบ Telegram, ทดสอบทุกช่องทางที่เปิด และส่งแจ้งเตือนสถานะปัจจุบัน

หลักสำคัญ: Enabled ไม่เท่ากับ Configured

Enabled หมายถึงผู้ดูแล “อนุญาตให้ช่องทางนั้นส่งแจ้งเตือน” ส่วน Configured หมายถึง “ระบบมี credential สำหรับช่องทางนั้นแล้ว” ดังนั้นควรตรวจสอบทั้งสองสถานะก่อนใช้งานจริง

จากภาพตัวอย่าง LINE และ Telegram ถูกเปิดใช้งานทั้งคู่ และระบบแสดง credential แบบย่อ (masked)

โดยไม่แสดงค่าฉบับเต็มกลับมาที่หน้า browser

2. ส่วนการเปิดใช้งานและสถานะระบบ



ส่วนบนของหน้าเว็บ: เปิด/ปิดแจ้งเตือน + สถานะ GNSS + สถานะ credential

2.1 เปิดหรือปิด LINE / Telegram

1. ดึงช่อง LINE Notification เมื่อต้องการให้ระบบส่งแจ้งเตือนทาง LINE
2. ดึงช่อง Telegram Notification เมื่อต้องการให้ระบบส่งแจ้งเตือนทาง Telegram
3. กด “บันทึกการเปิดใช้งานการแจ้งเตือน” เพื่อยืนยันการเลือก
4. หลังบันทึก ให้ดูสถานะด้านขวาของแต่ละช่องทางว่าขึ้น “เปิดใช้งาน” หรือ “ปิดใช้งาน” ตามที่ต้องการ

ไม่ต้องอัปโหลด credential ใหม่ทุกครั้งที่เปิด/ปิด

ถ้าระบบมี credential อยู่แล้ว การเปลี่ยน Enabled สามารถบันทึกและใช้งานได้ทันที การอัปโหลด credential จำเป็นเฉพาะเมื่อจะเพิ่มหรือเปลี่ยนค่าที่ใช้เชื่อมต่อ

3. การอ่านสถานะ GNSS และสถานะช่องทาง

ช่องสถานะ	ความหมาย
DEVICE	รหัสอุปกรณ์ที่กำลังติดตาม เช่น GNSS01
STATUS	สถานะ Online / Offline ของอุปกรณ์ ตามข้อมูลล่าสุดที่ระบบได้รับ
LAST SEEN	เวลาที่ระบบได้รับข้อมูลล่าสุดจากอุปกรณ์
AGE	ระยะเวลาที่ผ่านไปนับจากข้อมูลล่าสุด
FIX	สถานะ GNSS fix เช่น VALID
SATS v/u	ค่าจำนวนดาวเทียมตามที่ backend ส่งมาให้หน้าเว็บแสดง

ในภาพอ้างอิง อุปกรณ์ GNSS01 อยู่ในสถานะ ONLINE, FIX = VALID และมีข้อความสีเขียวแจ้งว่า “ระบบ GNSS01 อยู่ในสถานะ ONLINE (ปกติ)”

3.1 การ์ด LINE / Telegram

- แสดงค่า credential แบบย่อ เช่นต้นและท้ายของ Token / ID เพื่อใช้ตรวจสอบว่ามีค่าถูกโหลดอยู่ โดยไม่เปิดเผยค่าเต็ม
- แสดง “ที่มา” ของค่าที่กำลังใช้งาน เช่น config
- แสดงเวลา “อัปเดตล่าสุด” เพื่อช่วยยืนยันว่าการอัปเดต credential ครั้งล่าสุดถูกบันทึกแล้ว

หมายเหตุจากภาพอ้างอิง

ช่อง STATUS ในภาพมีข้อความ HTML เช่น `<span class="badge on">` แสดงออกมาเป็นตัวอักษรบางส่วน ซึ่งเป็นประเด็นด้านการ render หน้าเว็บ ไม่ใช่ค่าผิดพลาดจาก GNSS โดยสถานะที่ต้องอ่านคือ ONLINE / OFFLINE ที่ระบบรายงาน

4. การอัปโหลด Credentials ไปยัง Server

กรอกรหัส LINE / Telegram ที่ใช้งานจริง (อัปโหลดสู่ server)

กรอกรหัส 4 ช่องแล้วกด อัปโหลด credentials + บันทึกการตั้งค่า — ระบบเก็บค่าล่าสุดไว้ใน server (DB) เพื่อใช้แบบอัตโนมัติตามวาระ, ช่องที่เว้นว่าง = เก็บค่าเดิมไว้, ปุ่ม ล้าง = คืนค่าไปใช้ config/notifications.php — ค่าเป็น secret ผิด server ไม่แสดงค่าเก็บกลับ

LINE_CHANNEL_ACCESS_TOKEN LINE channel access token LINE_TARGET_ID LINE target user/group ID TELEGRAM_BOT_TOKEN Telegram bot token TELEGRAM_CHAT_ID Telegram chat ID

อัปโหลด credentials + บันทึกการตั้งค่า ล้างค่า LINE (ใช้ config เดิม) ล้างค่า Telegram (ใช้ config เดิม) กดลบ LINE กดลบ Telegram กดลบช่องทางที่เปิด แจ้งเตือนตามนาฬิกา

LINE ON · Telegram ON · การตั้งค่าเปลี่ยนล่าสุด 15 09 2026 16:07:20+07 · Auto-notify (web autopilot) = ON กดคลิกเพื่อดูหน้าเว็บนี้

ส่วนล่างของหน้าเว็บ: ช่องกรอก 4 ค่าและปุ่มจัดการ / ทดสอบ

หน้าเว็บปัจจุบันออกแบบให้ผู้ดูแลกรอกรหัส credential แล้วอัปโหลดไปเก็บฝั่ง server โดยไม่ต้องเปิดไฟล์ config เพื่อแก้ค่าทุกครั้ง

ค่าที่กรอก	การใช้งาน
LINE_CHANNEL_ACCESS_TOKEN	Channel Access Token ของ LINE Messaging API
LINE_TARGET_ID	User ID หรือ Group ID ปลายทางที่ LINE จะส่งข้อความไปหา
TELEGRAM_BOT_TOKEN	Token ของ Telegram Bot
TELEGRAM_CHAT_ID	Chat ID ปลายทางของ Telegram

4.1 วิธีอัปโหลดค่า

1. สำหรับการตั้งค่าครั้งแรก ให้กรอกทั้ง 4 ช่องให้ครบและตรวจสอบค่าก่อนกดบันทึก
2. กด “อัปโหลด credentials + บันทึกการตั้งค่า”
3. ตรวจสอบเวลา “อัปโหลดล่าสุด” และการแจ้งเตือน LINE / Telegram ว่าระบบเห็น credential แล้ว
4. กดทดสอบ LINE และ Telegram แยกกันก่อน จากนั้นจึงทดสอบช่องทางที่เปิดทั้งหมด

การแก้เพียงบางค่า

ตามข้อความบนหน้าเว็บ ช่องที่เว้นว่างจะเก็บค่าเดิมไว้ ดังนั้นหลังตั้งค่าครบครั้งแรก หากต้องการเปลี่ยนเฉพาะ Token หรือ ID บางตัว สามารถกรอกเฉพาะช่องที่เปลี่ยนและปล่อยช่องอื่นว่างได้

5. ปุ่ม “ล้างค่า” และแหล่ง Credential

ปุ่มล้างค่าไม่ได้หมายถึงการปิดช่องทางแจ้งเตือน แต่เป็นการลบค่าที่อัปโหลดไว้บน server สำหรับช่องทางนั้น เพื่อให้ระบบกลับไปใช้ค่าจาก config/notifications.php ตามข้อความกำกับบนปุ่ม

ปุ่ม	ผลที่คาดหวัง
ล้างค่า LINE (ใช้ config เดิม)	ล้างค่า override ของ LINE ที่เก็บบน server แล้วกลับไปใช้ค่า LINE จาก config/notifications.php
ล้างค่า Telegram (ใช้ config เดิม)	ล้างค่า override ของ Telegram ที่เก็บบน server แล้วกลับไปใช้ค่า Telegram จาก config/notifications.php
ก่อนกดล้างค่า ตรวจสอบให้แน่ใจว่า config/notifications.php ยังมีค่าที่ใช้งานได้ หาก config ไม่มีค่าหรือค่าไม่ถูกต้อง ช่องทางนั้นอาจยัง Enabled แต่ไม่สามารถส่งข้อความได้	

5.1 ลำดับตรวจหลังล้างค่า

1. กดล้างเฉพาะช่องทางที่ต้องการ
2. ตรวจสอบการ์ด “LINE ที่มา” หรือ “Telegram ที่มา” ว่ากลับเป็น config ตามที่หน้าเว็บระบุ
3. กด “ทดสอบ LINE” หรือ “ทดสอบ Telegram” เพื่อยืนยันว่า fallback config ใช้งานได้

6. การทดสอบการเชื่อมต่อและการแจ้งเตือน

ปุ่มทดสอบ	หน้าที่
ทดสอบ LINE	ส่งข้อความทดสอบเฉพาะ LINE ใช้แยกตรวจ Token / Target ID และการเชื่อมต่อ LINE
ทดสอบ Telegram	ส่งข้อความทดสอบเฉพาะ Telegram ใช้แยกตรวจ Bot Token / Chat ID
ทดสอบช่องทางที่เปิด	ทดสอบเฉพาะช่องทางที่ตั้ง Enabled ไว้ เหมาะสำหรับตรวจการตั้งค่าหลังบันทึก
แจ้งเตือนสถานะปัจจุบัน	สั่งให้ระบบส่งข้อความที่อ้างอิงสถานะ GNSS ปัจจุบัน เพื่อทดสอบรูปแบบการแจ้งเตือนจริง

6.1 ลำดับทดสอบที่แนะนำ

1. ตรวจสอบว่า GNSS มีสถานะล่าสุดและไม่ใช้ข้อมูลเก่าผิดปกติ
2. ทดสอบ LINE แยก
3. ทดสอบ Telegram แยก
4. ทดสอบ “ช่องทางที่เปิด”
5. กด “แจ้งเตือนสถานะปัจจุบัน” เพื่อยืนยันข้อความในงานจริง

เหตุผลที่ควรทดสอบแยกก่อน

หาก LINE ผ่านแต่ Telegram ไม่ผ่าน (หรือกลับกัน) จะระบุปัญหาได้ทันทีว่าอยู่ที่ credential / ปลายทาง / API ของช่องทางใด โดยไม่ต้องแกะโค้ดทดสอบหลายไฟล์

7. Auto-notify และการแจ้งเตือนสถานะ

แถบสถานะด้านล่างของภาพแสดง “LINE ON · Telegram ON” และ “Auto-notify (web autopilot) = ON ตลอดที่เปิดหน้าเว็บ” ดังนั้นคู่มือนี้ให้ถือว่า web autopilot เป็นกลไกที่ทำงานในบริบทของหน้าเว็บตามข้อความสถานะที่ระบบแสดง

7.1 สิ่งที่ต้องตรวจระหว่างใช้งาน

- LINE / Telegram ที่ต้องการใช้งานต้องเป็น ON
- ช่องทางต้องมี credential พร้อมใช้งาน (Configured)
- สถานะอุปกรณ์ต้องอัปเดตตาม refresh cycle และ LAST SEEN / AGE ต้องสมเหตุสมผล
- หากระบบแจ้ง ONLINE ด้วย ให้ตรวจว่าข้อความที่ได้รับตรงกับสถานะในหน้าเว็บ
- เวลา “การตั้งค่าเปลี่ยนล่าสุด” และ “อัปเดตล่าสุด” ใช้ช่วยยืนยันการเปลี่ยนแปลงที่เพิ่งทำ

ขอบเขตของ Auto-notify

ข้อความบน UI ระบุว่า web autopilot = ON ตลอดที่เปิดหน้าเว็บ คู่มือนี้จึงไม่ถือว่า web autopilot เพียงอย่างเดียวรับประกันการทำงานเมื่อเปิด browser หากระบบมี cron หรือ backend worker แยกให้ตรวจกลไกนั้นในเอกสารของระบบส่วนดังกล่าว

8. Troubleshooting แบบย่อ

อาการ	แนวทางตรวจ
Enabled แต่ส่งไม่ได้	ตรวจ Configured, การ์ด credential แบบ masked, แหล่งที่มาของค่า และทดสอบช่องทางนั้นแยก
อัปเดตแล้วเหมือนไม่เปลี่ยน	ตรวจเวลา “อัปเดตล่าสุด”; Refresh หน้าเว็บ แล้วทดสอบ LINE / Telegram ใหม่
LINE ไม่เข้า แต่ Telegram เข้า	ตรวจ LINE Channel Access Token, Target ID และสถานะ Official Account / Messaging API
Telegram ไม่เข้า แต่ LINE เข้า	ตรวจ Bot Token, Chat ID และว่าปลายทางเคย Start / เปิดบทสนทนากับ Bot แล้ว
GNSS แสดง Offline / LAST SEEN เก่า	แก้ปัญหาการรับข้อมูล GNSS / network ก่อน เพราะเป็นคนละส่วนกับ credential แจ้งเตือน
STATUS มี HTML โหล่	เป็นปัญหาการแสดงผล template/escaping ของหน้าเว็บ ควรแก้ renderer แต่ไม่ควรตีความว่า GNSS status ผิด

8.1 สิ่งที่ไม่ต้องทำใน workflow ปกติ

- ไม่ต้องสร้าง Provider / LINE Official Account / Telegram Bot ใหม่ หากของเดิมยังใช้งานอยู่
- ไม่ต้องสร้าง line-test.php, telegram-test.php หรือ notifications-test.php แยกเพื่อทดสอบทั่วไป เพราะหน้า Setting and Tester มีปุ่มทดสอบแล้ว
- ไม่ควรพิมพ์ Token เต็มลงรายงานหรือภาพหน้าจอที่แชร์ให้บุคคลอื่น

9. Checklist ก่อนใช้งานจริง

ลำดับ	รายการตรวจ	สถานะ
1	LINE Notification / Telegram Notification เลือกตามช่องทางที่ต้องการ	☐
2	กดบันทึกการเปิดใช้งานแล้วสถานะตรงกับที่เลือก	☐
3	ระบบแสดง credential เป็น Configured / มีค่าที่ mask แล้ว	☐
4	ถ้าเพิ่งเปลี่ยน credential เวลา “อัปเดตล่าสุด” อัปเดตแล้ว	☐
5	ทดสอบ LINE ผ่านและมีข้อความเข้าปลายทาง	☐
6	ทดสอบ Telegram ผ่านและมีข้อความเข้าปลายทาง	☐
7	ทดสอบช่องทางที่เปิดผ่าน	☐
8	แจ้งเตือนสถานะปัจจุบันได้ และข้อความตรงกับ GNSS status	☐
9	LAST SEEN / AGE / FIX แสดงข้อมูลล่าสุดที่สมเหตุสมผล	☐
10	ไม่มี Token / Secret แบบเต็มอยู่ในเอกสารหรือ screenshot ที่จะแชร์	☐

สรุปการปรับจากคู่มือเดิม

ตัดขั้นตอนที่ไม่จำเป็นต่อการใช้งานหน้าปัจจุบันออก ได้แก่การสร้าง Provider / Bot ตั้งแต่ต้น, การเปิด Webhook สำหรับการ Push อย่างเดียว, ตัวอย่างโค้ด PHP สำหรับ test endpoint และการพิมพ์ credentials จริงไว้ในเอกสาร แล้วแทนด้วย workflow จากหน้า Setting and Tester ตามภาพอ้างอิง

เอกสารฉบับนี้ตั้งใจให้ผู้ดูแลระบบใช้เป็นคู่มือปฏิบัติงานจากหน้าเว็บปัจจุบันโดยตรง และลดการจัดการ secret ผ่านไฟล์หรือเอกสารให้เหลือน้อยที่สุด

ค่าหลักที่ระบบต้องใช้

ตัวแปร	ได้จาก	รูปแบบ	ใช้ทำอะไร
LINE_CHANNEL_ACCESS_TOKEN	LINE Developers → Messaging API	ข้อความยาว	Authorization สำหรับ LINE Push API
LINE_TARGET_ID	Basic settings → Your user ID	Uxxxxxxx...	ผู้รับ LINE
TELEGRAM_BOT_TOKEN	@BotFather	123456:AA...	Authorization สำหรับ Telegram Bot API
TELEGRAM_CHAT_ID	getUpdates → message.chat.id	เลขจำนวนเต็ม	ปลายทาง Telegram

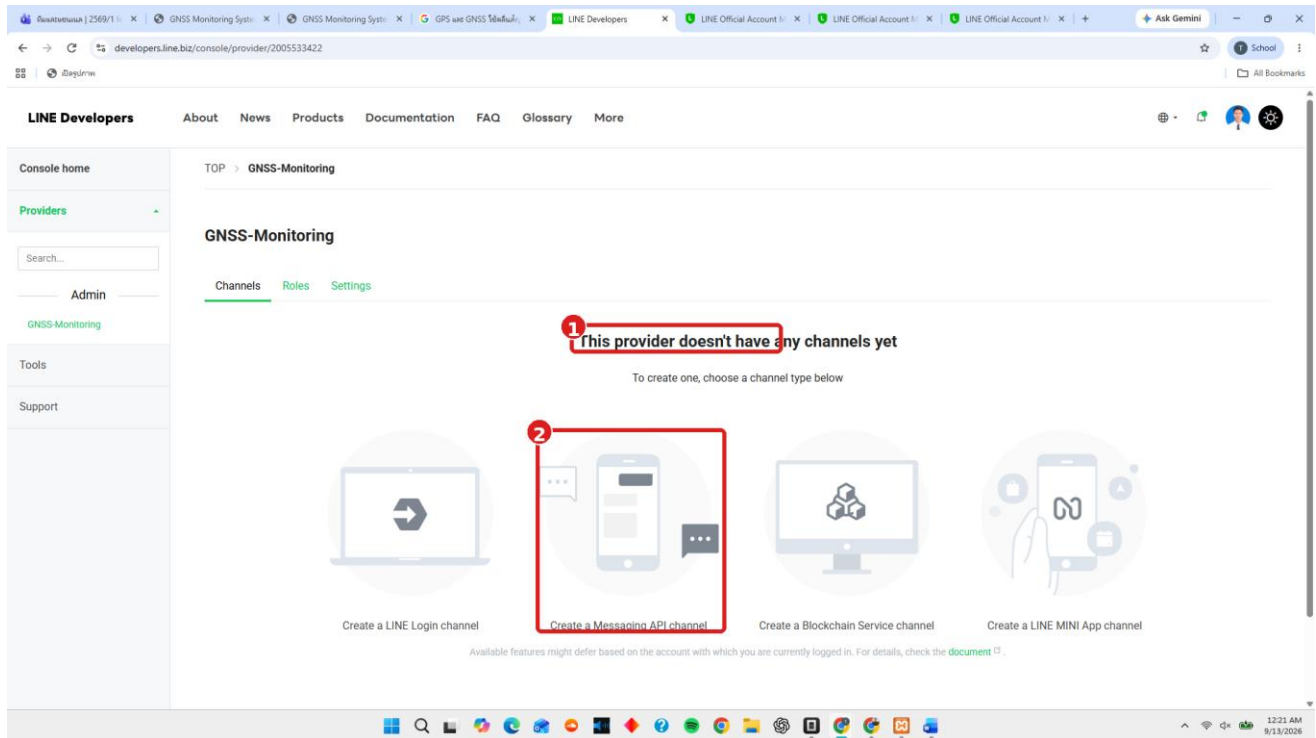
ส่วน A — ตั้งค่า LINE Messaging API

i ข้อควรรู้สำหรับ LINE เวอร์ชันปัจจุบัน

ปัจจุบัน Messaging API Channel ต้องสร้างโดยเปิด Messaging API ให้ LINE Official Account ใน LINE Official Account Manager แล้วเลือก Provider ที่ต้องการ ไม่ใช่สร้าง Messaging API Channel โดยตรงจาก LINE Developers Console แบบคู่มือเก่า

A1. ตรวจสอบ / สร้าง Provider

1. เข้า <https://developers.line.biz> แล้ว Log in ด้วยบัญชี LINE ที่ใช้ดูแลระบบ
2. เข้า Console → Providers และสร้าง Provider ชื่อ GNSS-Monitoring หากยังไม่มี
3. เมื่อเข้า Provider แล้ว หากยังไม่ได้เปิด Messaging API จะเห็นข้อความว่า “This provider doesn’t have any channels yet”

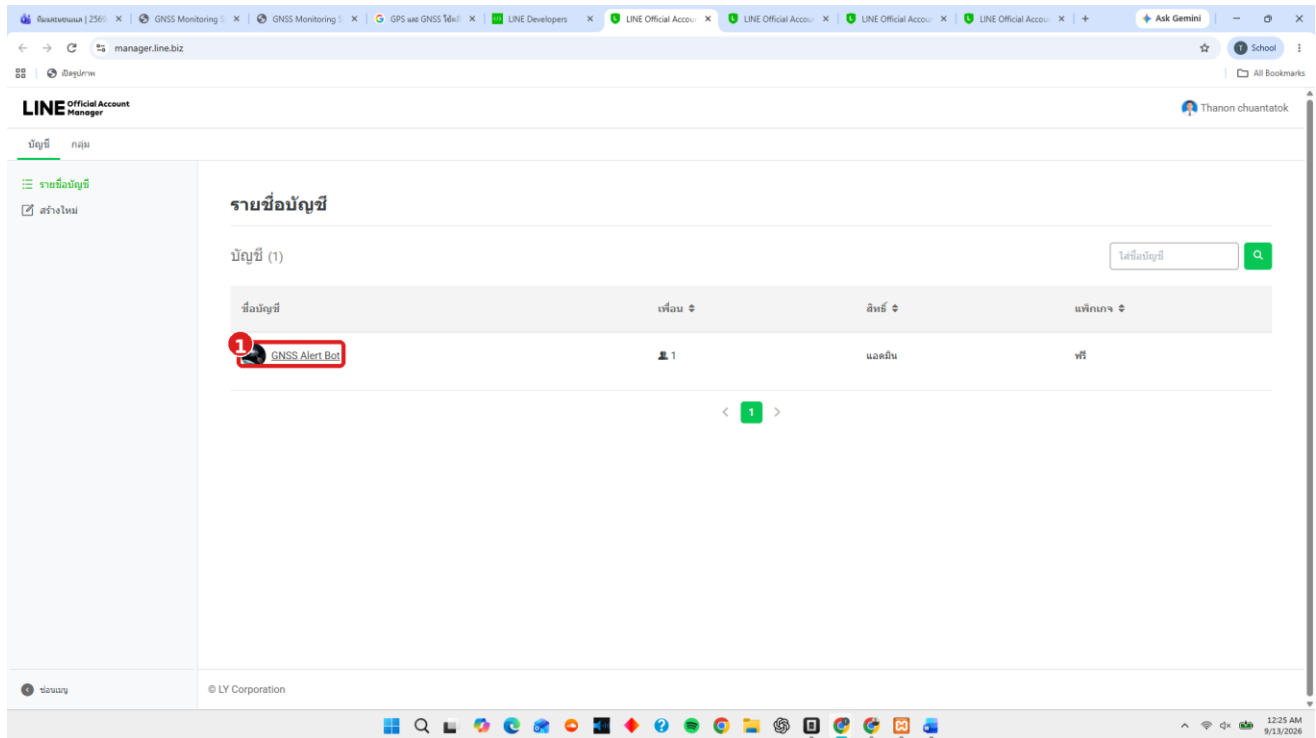


รูป A1 — Provider GNSS-Monitoring มีอยู่แล้ว แต่ยังไม่มี Channel; ไม่ต้องสร้าง Messaging API
จากการกดด้านล่างโดยตรง

Provider name = ตั้งชื่ออะไรก็ได้ เช่น GNSS-Monitoring

A2. เลือก LINE Official Account

4. เปิด LINE Official Account Manager: <https://manager.line.biz>
5. เลือกบัญชี GNSS Alert Bot จากรายการบัญชี

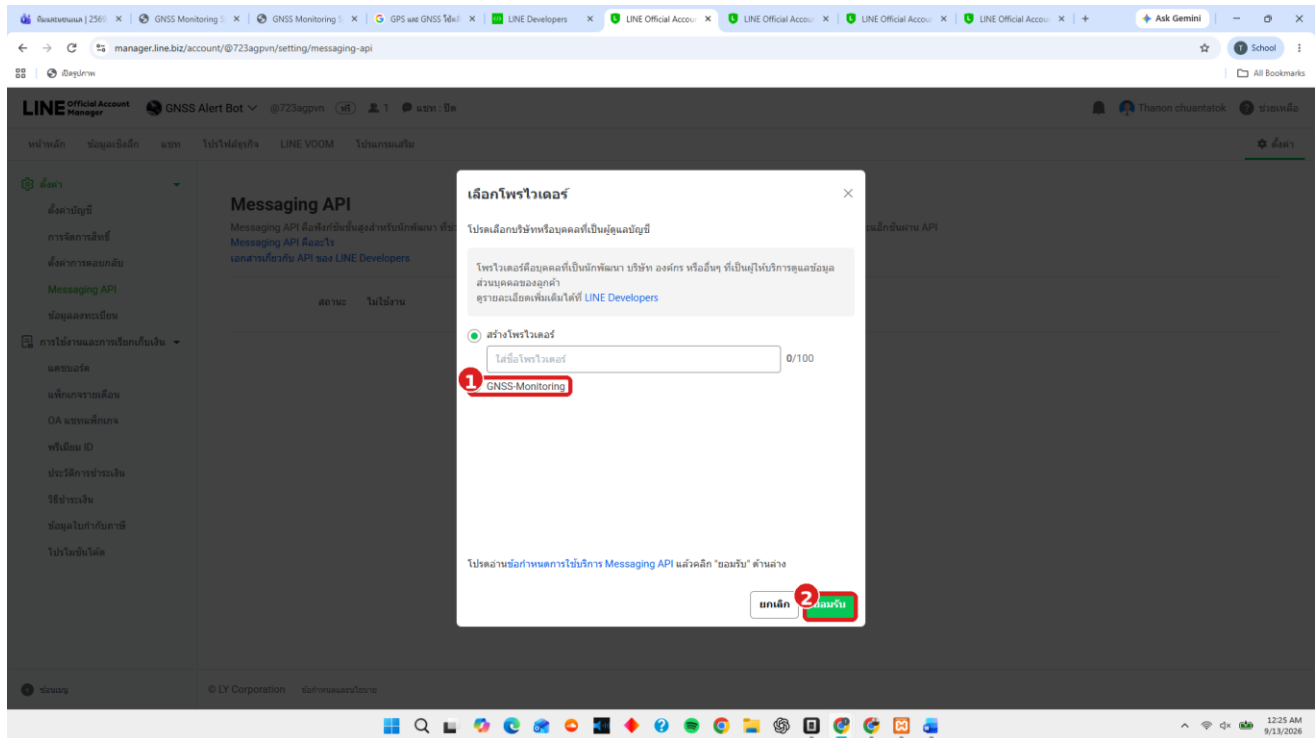


รูป A2 — เลือก LINE Official Account ชื่อ GNSS Alert Bot

LINE Official Account = ตั้งชื่ออะไรก็ได้ เช่น GNSS Alert Bot

A3. เปิด Messaging API และผูกกับ Provider

6. ใน GNSS Alert Bot ไปที่ Settings / ตั้งค่า → Messaging API
7. กดเปิดใช้งาน Messaging API
8. ในหน้าต่างเลือก Provider ให้เลือก GNSS-Monitoring ที่มีอยู่แล้ว ห้ามเลือก “สร้างโปรไวเดอร์” ขึ้น
9. กด “ยอมรับ” เพื่อผูก Official Account กับ Provider



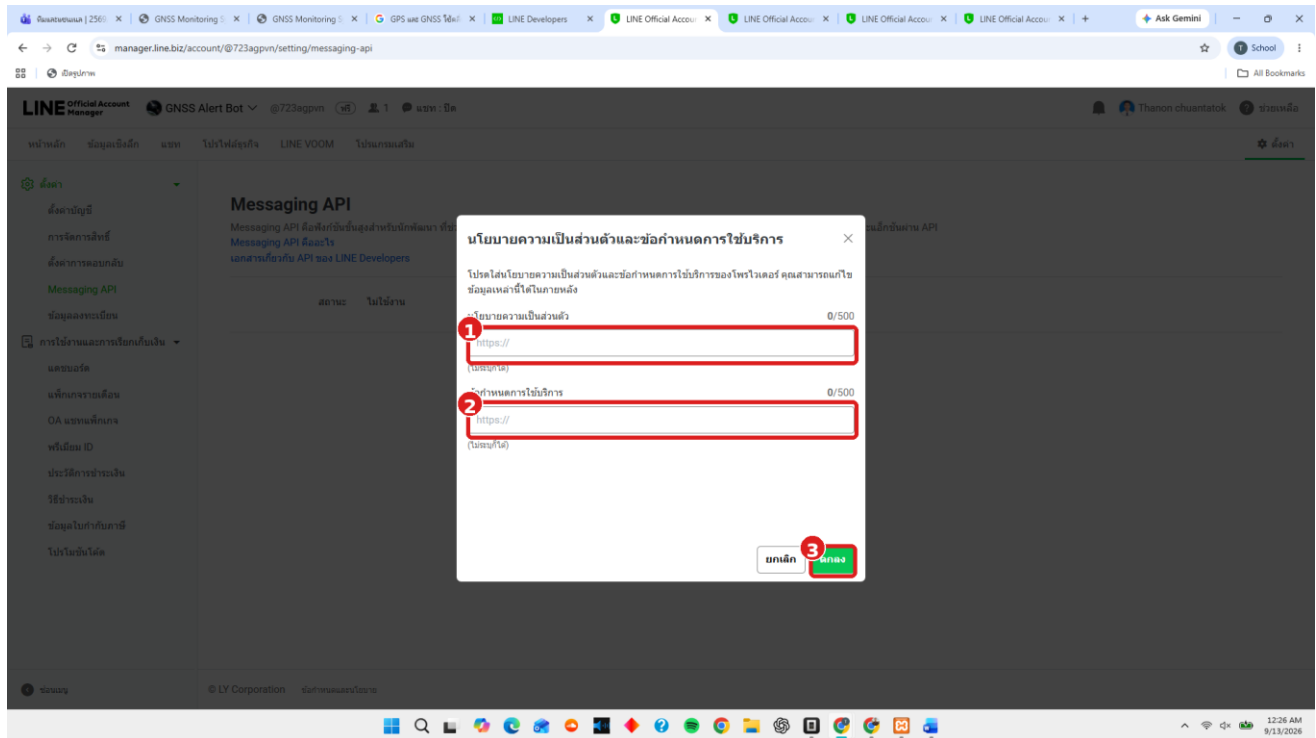
รูป A3-1 — เลือก GNSS-Monitoring แล้วกด “ยอมรับ”

! เหตุผลที่ต้องเลือก Provider ให้ถูกต้อง

เมื่อผูก LINE Official Account กับ Provider แล้ว จะไม่สามารถย้าย Provider ได้โดยง่าย และ User ID ที่ LINE ออกให้มีความสัมพันธ์กับ Provider จึงต้องใช้ชื่อที่ตั้งขึ้นมา เช่น GNSS-Monitoring

A3.1 Privacy Policy / Terms of Use

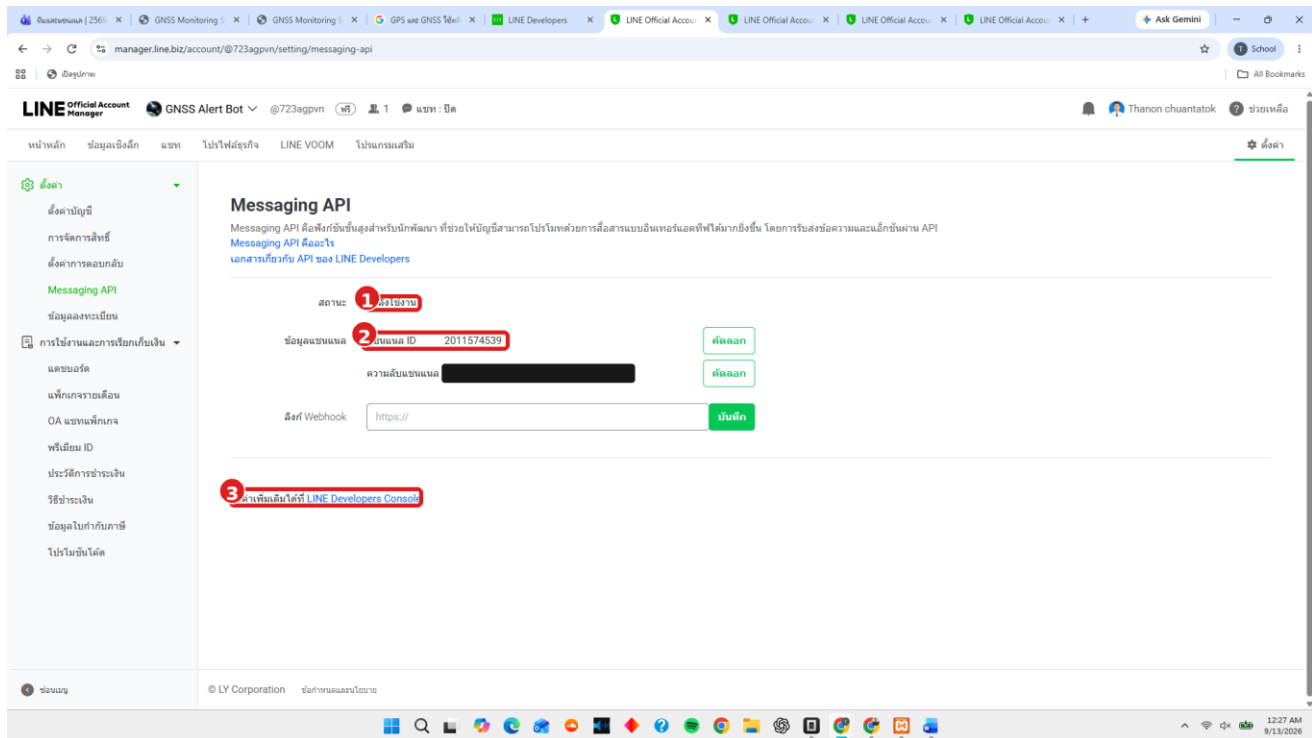
LINE จะแสดงช่อง URL สำหรับนโยบายความเป็นส่วนตัวและข้อกำหนดการให้บริการ หากหน้าจอร์บุว่า “ไม่ระบุก็ได้” สามารถเว้นว่างในขั้นตอนี้ แล้วกด “ตกลง” ได้



รูป A3-2 — ช่อง Privacy Policy / Terms สามารถเว้นว่างได้ตามที่หน้าจอระบุ

A4. ตรวจสอบว่า Messaging API เปิดใช้งานสำเร็จ

หลังยืนยันแล้ว หน้าจอ Messaging API ใน LINE Official Account Manager ต้องแสดงสถานะ “กำลังใช้งาน” และมี Channel ID ปรากฏ



รูป A4 — Messaging API เปิดใช้งานแล้ว; Channel Secret ในภาพถูกปิดบังเพราะไม่จำเป็นต่อการ Push ข้อความในคู่มือนี้

✓ สถานะที่ถูกต้อง

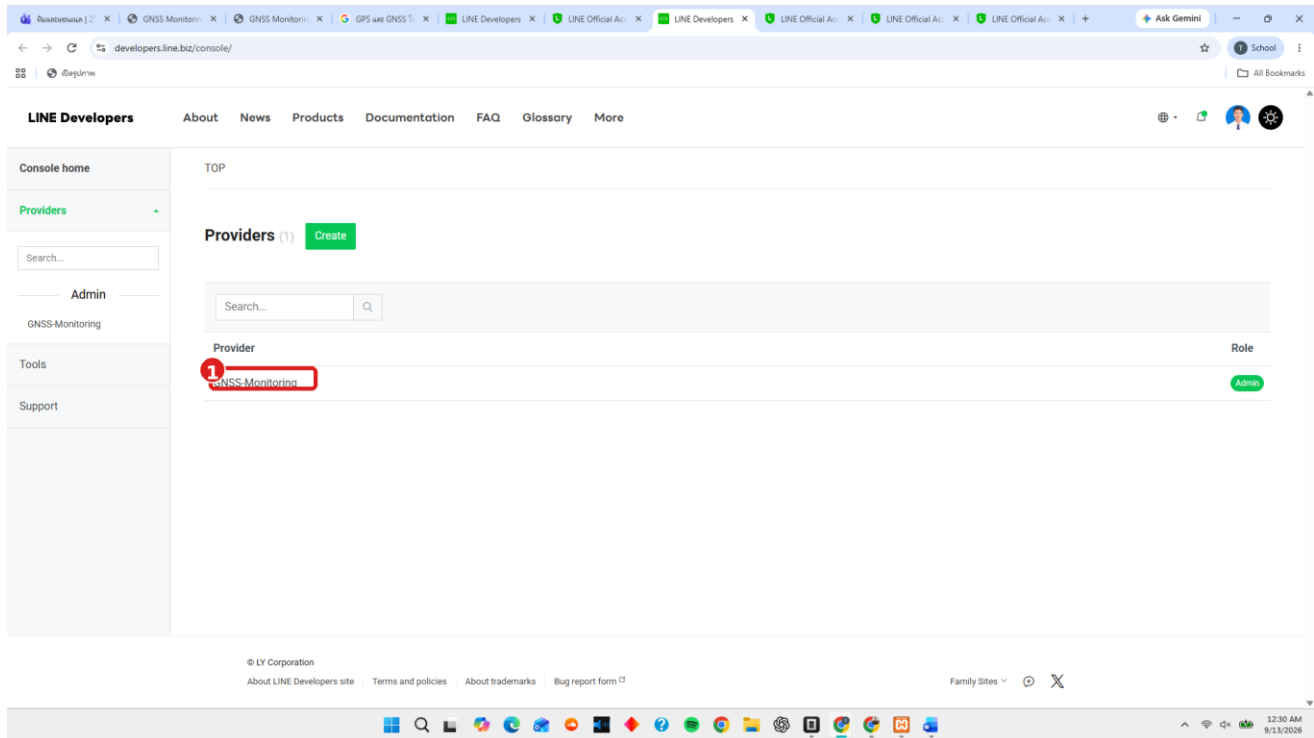
Status = กำลังใช้งาน | Channel ID มีค่าแล้ว | Webhook URL ยังเว้นว่างได้สำหรับงาน Push Notification อย่างเดียว

i Webhook จำเป็นเมื่อใด

ถ้าต้องการรับข้อความ/event จากผู้ใช้หรือกลุ่มเข้าระบบ จึงค่อยตั้ง Webhook. สำหรับการส่งแจ้งเตือนออกไปหา LINE ของผู้ใช้เพียงอย่างเดียว ไม่ต้องเปิด Webhook

A5. เข้า LINE Developers Console และ Channel

10. คลิก LINE Developers Console จากหน้าจอ Messaging API หรือเข้า developers.line.biz/console
11. เลือก Provider GNSS-Monitoring

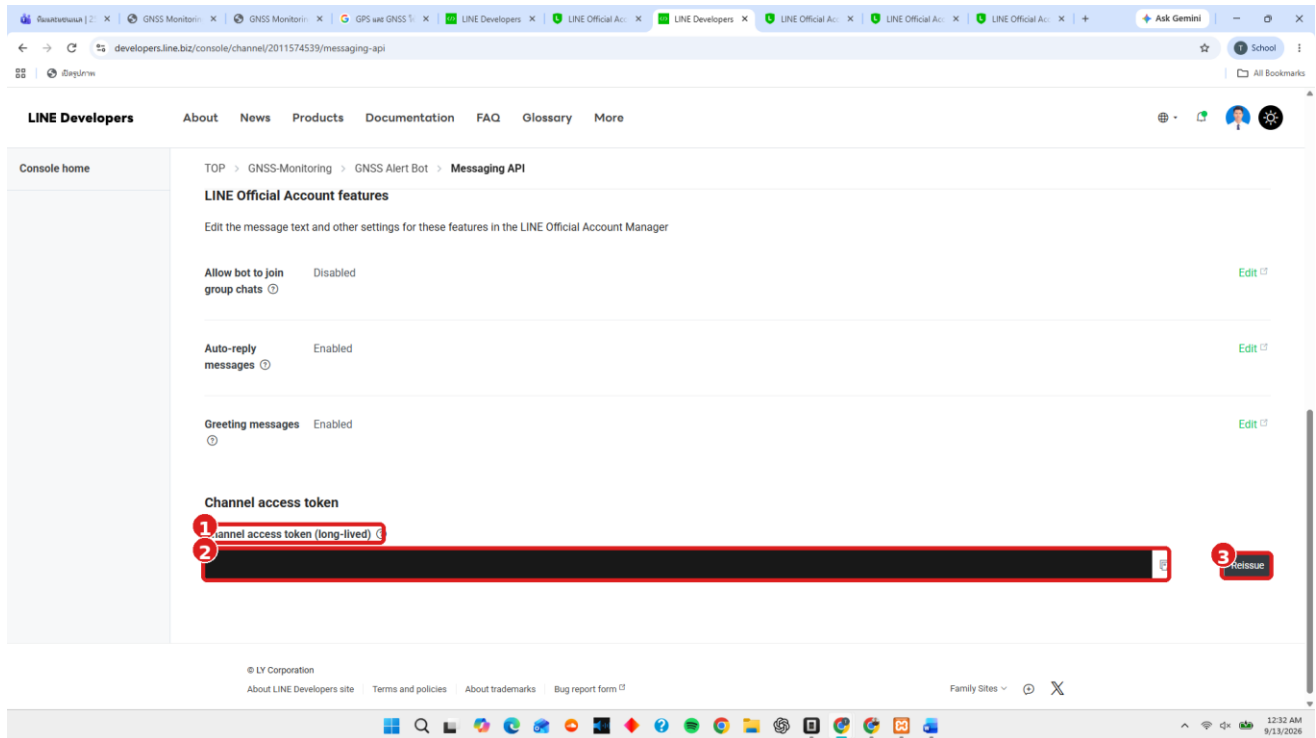


รูป A5 — เลือก Provider GNSS-Monitoring

12. ภายใน Provider จะมี Messaging API Channel ของ GNSS Alert Bot ที่ถูกสร้างจาก Official Account Manager โดยอัตโนมัติ

A6. ออก Long-lived Channel Access Token

13. เข้า GNSS Alert Bot → แท็บ Messaging API
14. เลื่อนลงถึงหัวข้อ Channel access token → Channel access token (long-lived)
15. ถ้ายังไม่มี Token ให้กด Issue; ถ้ามีแล้วปุ่มจะเป็น Reissue
16. Copy Token ไปเก็บในไฟล์ config/notifications.php



รูป A6 — ตำแหน่ง Long-lived Channel Access Token (ค่าในภาพถูกปิดบัง; ค่าจริงอยู่ในตาราง Credential ของเอกสารนี้)

i คุณสมบัติ Long-lived Token

มีอายุแบบ Indefinite และมี Long-lived Token ที่ active ได้ 1 ตัวต่อ Messaging API Channel. การ Reissue จะทำให้ Token เดิมถูกยกเลิกตามนโยบาย LINE

A7. หา LINE_TARGET_ID (Your user ID)

17. ใน LINE Developers → GNSS Alert Bot → Basic settings

18. เลื่อนหา “Your user ID”

19. Copy ค่าที่ขึ้นต้นด้วย U แล้วใช้เป็น LINE_TARGET_ID

ค่าตัวอย่าง

LINE_TARGET_ID = U2b1445a0b7671a4c7fe422b4n126a64a45

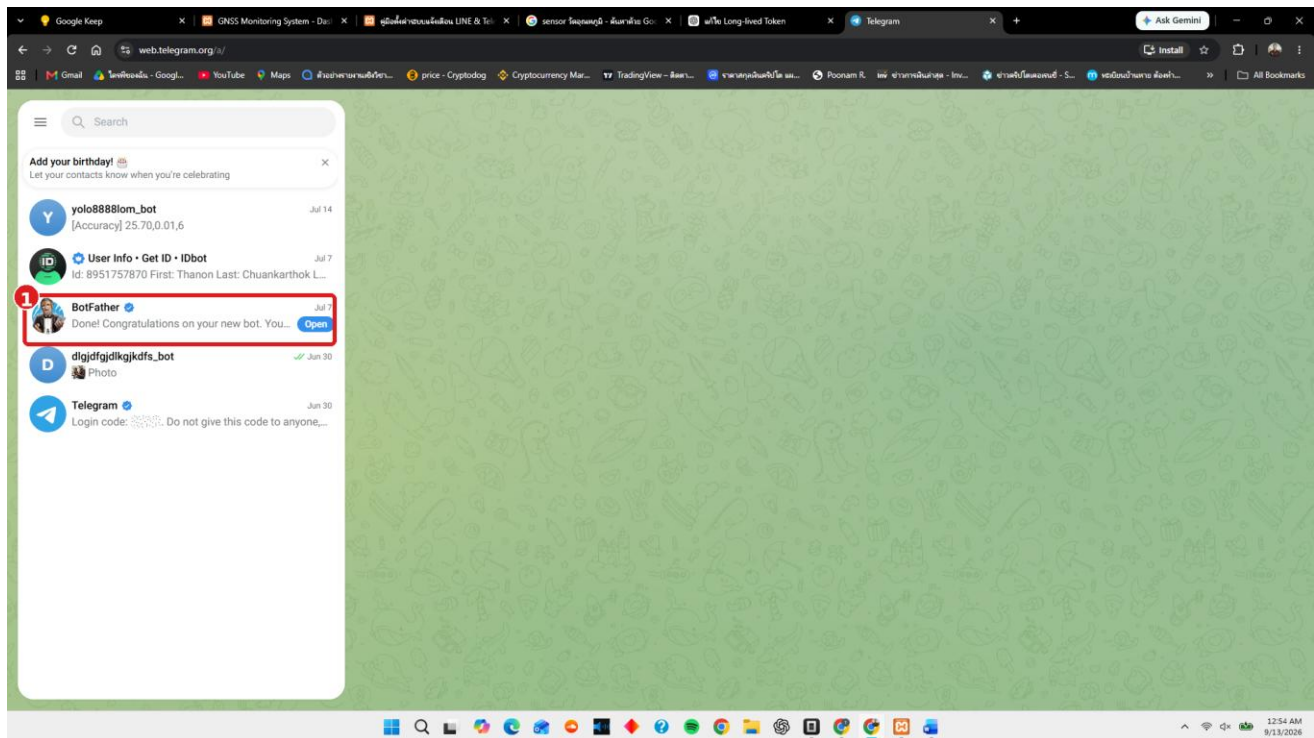
กรณีส่งเข้ากลุ่ม LINE จะต้องใช้ groupId ซึ่งได้จาก Webhook event และค่ามักขึ้นต้นด้วย C;
 แต่คู่มือนี้ตั้งค่าการส่งตรงถึงบัญชีผู้ใช้ จึงใช้ Your user ID และไม่ต้องเปิด Webhook

ส่วน B — ตั้งค่า Telegram Bot

B1. เปิด @BotFather

20. เปิด Telegram แล้วค้นหา @BotFather ซึ่งเป็น Bot ทางการสำหรับสร้างและจัดการ Bot

21. กด Start / Open



รูป B1 — เลือก @BotFather ที่มีเครื่องหมายยืนยัน

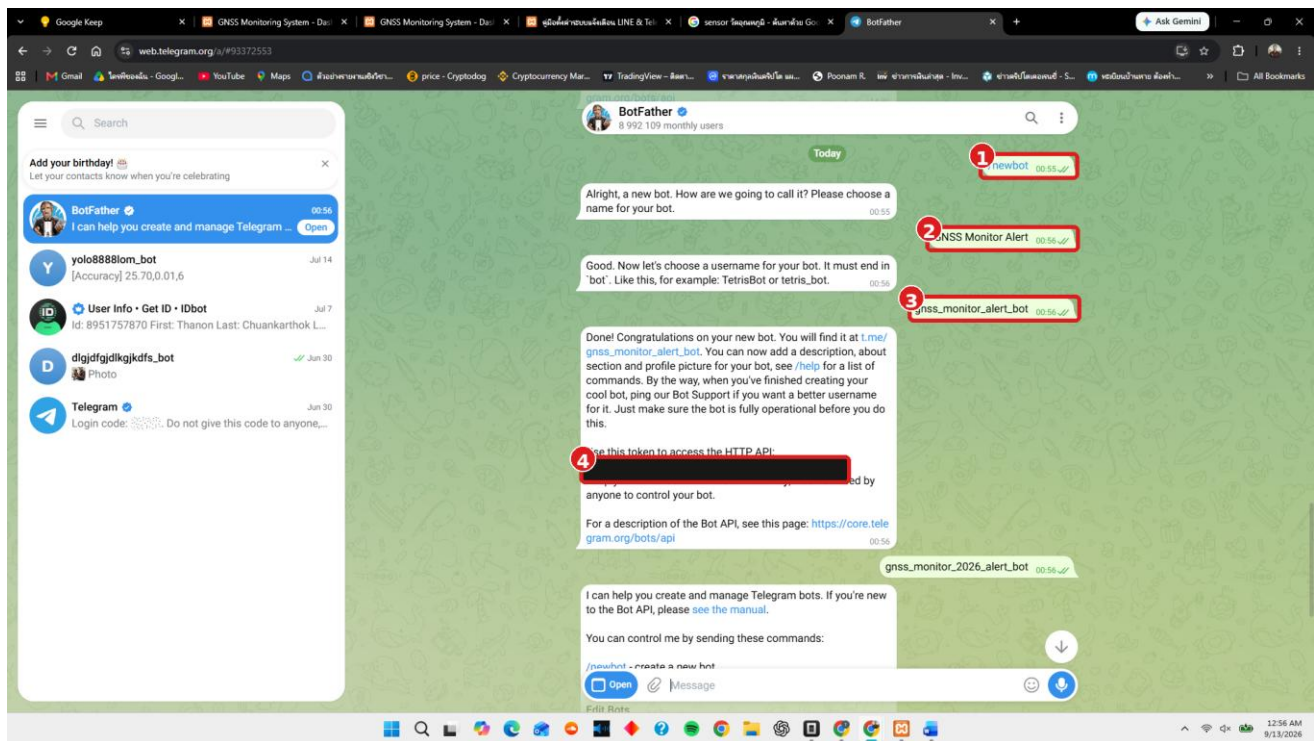
B2. สร้าง GNSS Monitor Alert Bot

22. ส่งคำสั่ง /newbot

23. เมื่อ BotFather ถามชื่อ ให้ใส่ GNSS Monitor Alert

24. เมื่อถาม username ให้ใส่ gnss_monitor_alert_bot (ต้องลงท้ายด้วย bot และต้องไม่ซ้ำ)

25. เมื่อสร้างสำเร็จ BotFather จะส่ง Bot Token ให้ Copy เก็บไว้



รูป B2 — สร้าง Bot สำเร็จ; Bot Token ในภาพถูกปิดบัง

Bot name = ตั้งชื่ออะไรก็ได้ เช่น GNSS Monitor Alert

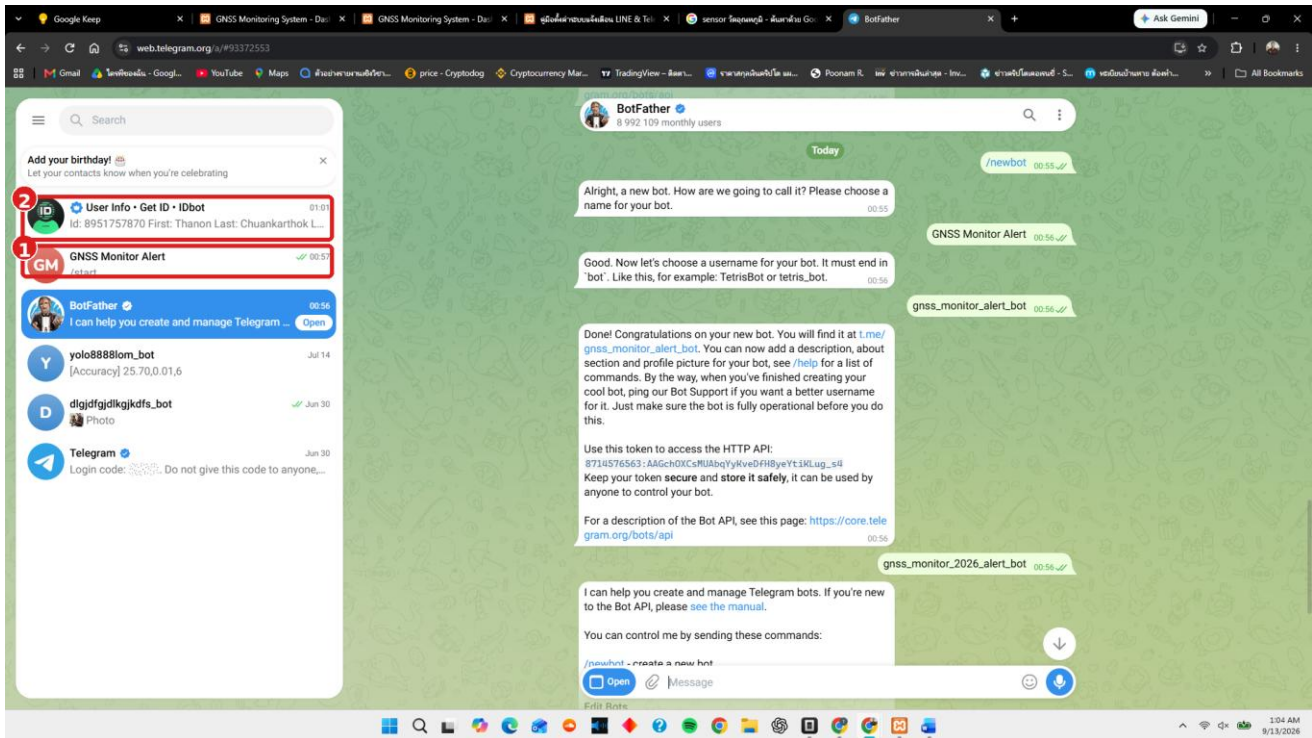
Bot username = ตั้งชื่ออะไรก็ได้ เช่น @gnss_monitor_alert_bot

B3. Start Bot ก่อนหา Chat ID

26. คลิกลิงก์ t.me/gnss_monitor_alert_bot หรือค้นหา @gnss_monitor_alert_bot

27. กด Start

28. ส่งข้อความทดสอบ เช่น test อย่างน้อย 1 ข้อความ เพื่อให้ Telegram สร้าง Update ที่ Bot API อ่านได้



รูป B3 — GNSS Monitor Alert พร้อมใช้งาน; IDBot เป็นเพียงทางเลือกตรวจ User ID ไม่จำเป็นต้องหาใช้ getUpdates

B4. หา TELEGRAM_CHAT_ID ด้วย getUpdates

เปิด URL ต่อไปนี้โดยต้องแทน <TOKEN> ด้วย Bot Token จริงทั้งหมด รวมถึงเอาเครื่องหมาย < > ออก

รูปแบบ URL

```
https://api.telegram.org/bot<TOKEN>/getUpdates
```

ตัวอย่างโครงสร้างผลลัพธ์ที่ต้องการ:

ตัวอย่าง JSON จาก getUpdates

```
{
  "ok": true,
```

```

"result": [
  {
    "message": {
      "chat": {
        "id": 8951752014,
        "type": "private"
      }
    }
  }
]
}

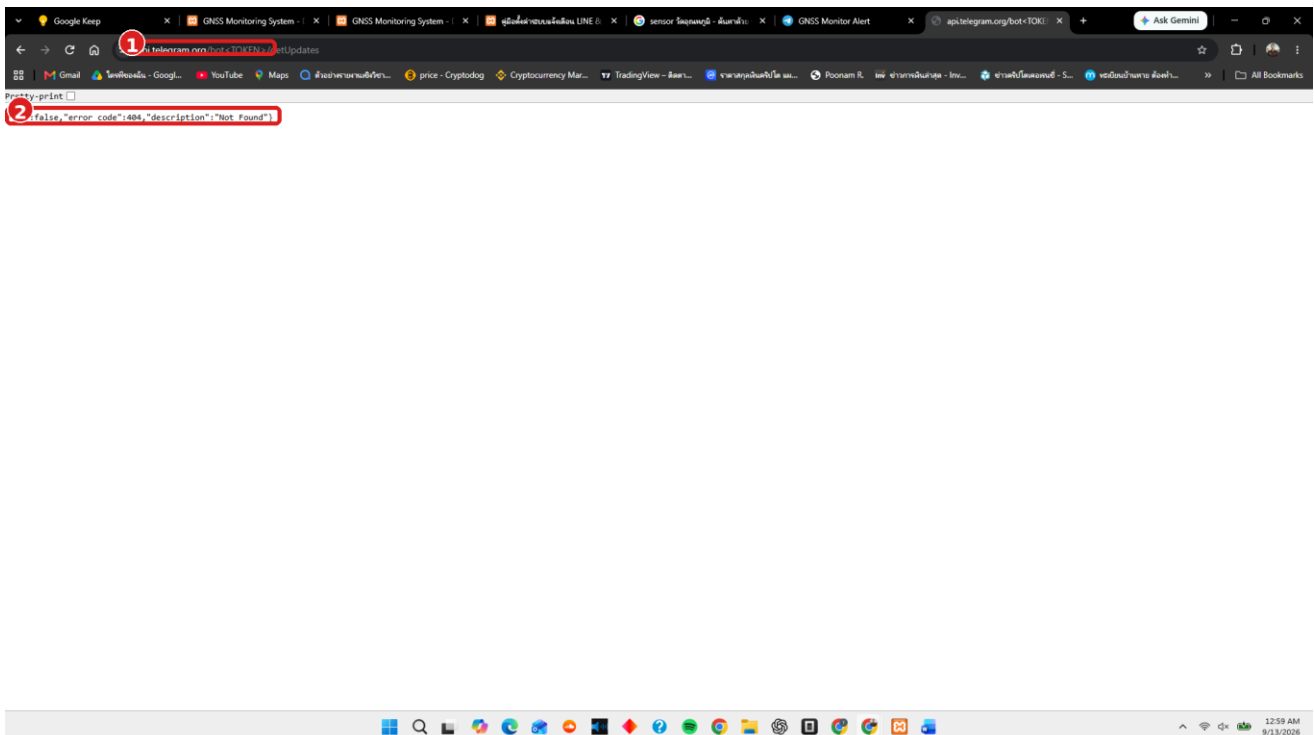
```

ค่าตัวอย่าง

TELEGRAM_CHAT_ID = 8951752014

B4.1 ถ้าเจอ 404 Not Found

ถ้า URL ยังมีข้อความ <TOKEN> อยู่ Telegram จะไม่รู้ว่า เป็น Bot ตัวใด จึงตอบ 404 Not Found



รูป B4 — ตัวอย่าง 404 เพราะ URL ยังใช้ <TOKEN> แทน Token จริง

แก้ วิธีแก้

กลับไป Copy Token จาก BotFather แล้วสร้าง URL เป็น

[https://api.telegram.org/bot\[BotTokenจริง\]/getUpdates](https://api.telegram.org/bot[BotTokenจริง]/getUpdates) จากนั้น Refresh

C1. ตาราง Credential ที่ได้มาต้องนำไปกรอกค่าในหน้า Dashboard

ตัวแปร	ค่าตัวอย่าง
LINE_CHANNEL_ACCESS_TOKEN	RZraz6y6bZhUoXQa/4Sfq42LB9kKN/rngxEPE82JAL7xd2Xmt5cwmyyxD1oVx3LT0KQM7FZta mVZyzZWddO4X2h3jEg8ecYuL8/yOViNTDXEJoRcqqVTjYEGSTdEwrXeWLJmYO9A78QT9cQfL nQmaAdB04t89/1O/w1cjhgjgkjl54lFU=
LINE_TARGET_ID	U2b1445a0b7fgfhj4c7fe422b4a6a64a22
TELEGRAM_BOT_TOKEN	8714576563:AAGch0XCsMUAbdhgfdYykveDfH8yeYt455Lug_s4
TELEGRAM_CHAT_ID	8951752014