

Week 3: LINE Messaging API & External API

Building LINE Bot with n8n

วิชา n8n Workflow Automation

ภาคการศึกษาที่ 2/2568

สารบัญ

- 1 Introduction & Recap
- 2 LINE Messaging API
- 3 Webhook & ngrok
- 4 HTTP Request Node
- 5 Thai Gold API
- 6 LINE Reply API
- 7 Code Node
- 8 Complete Workflow
- 9 Summary

Recap: สิ่งที่เราเรียนมาแล้ว

Lab 01: n8n Basic Workflow

- Workflow, Node, Connection
- Webhook Node รับข้อมูล
- Code Node ประมวลผล
- JSON Data Flow

Lab 02: (Previous content)

- HTTP Request Node
- External API Integration

วันนี้จะเรียนอะไร?

สร้าง LINE Bot ราคาทอง

หัวข้อที่จะเรียน:

- 1 LINE Messaging API Overview
- 2 Webhook และ ngrok
- 3 HTTP Request Node (GET & POST)
- 4 Thai Gold API
- 5 LINE Reply API
- 6 Hands-on: สร้าง Gold Price Bot

LINE Messaging API คืออะไร?

LINE Messaging API = API สำหรับสร้าง LINE Bot

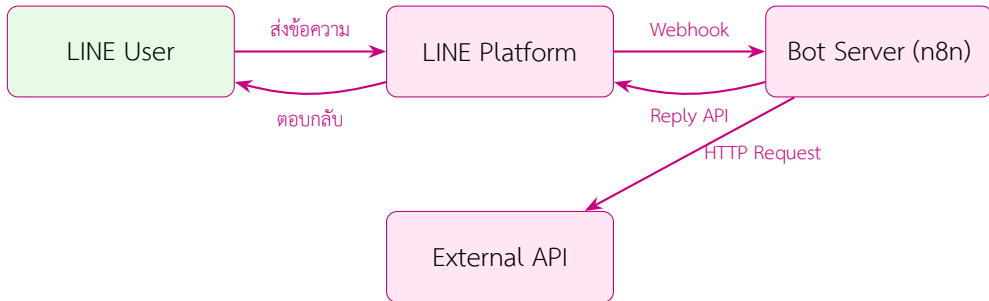
สิ่งที่ทำได้:

- รับข้อความจากผู้ใช้
- ส่งข้อความตอบกลับ
- ส่ง Push Message
- Rich Menu, Flex Message
- Group/Room Management

ตัวอย่างการใช้งาน:

- Customer Service Bot
- Notification Bot
- E-commerce Bot
- Information Bot

LINE Messaging API Architecture



LINE Channel Types

Messaging API Channel:

- สร้าง Bot
- รับ-ส่งข้อความ
- Webhook Integration
- ใช้ใน Lab นี้

LINE Login Channel:

- Login ด้วย LINE
- ดึง Profile ผู้ใช้
- OAuth 2.0

Lab นี้ใช้ Messaging API Channel

LINE Webhook Events

ประเภท Event ที่ Bot รับได้:

Event Type	เมื่อไหร่
message	ผู้ใช้ส่งข้อความ
follow	ผู้ใช้เพิ่ม Bot เป็นเพื่อน
unfollow	ผู้ใช้ Block Bot
postback	ผู้ใช้กดปุ่ม (Rich Menu, Button)
join	Bot ถูกเชิญเข้า Group
leave	Bot ถูกเตะออกจาก Group

Lab นี้ใช้ message event

Webhook คืออะไร?

Webhook = การส่ง HTTP Request อัตโนมัติเมื่อเกิด Event



ข้อกำหนดของ LINE Webhook:

- ต้องเป็น **HTTPS** (ไม่รับ HTTP)
- SSL Certificate ต้อง valid
- ต้องเข้าถึงได้จาก Internet (Public URL)

ปัญหา: localhost ไม่มี HTTPS



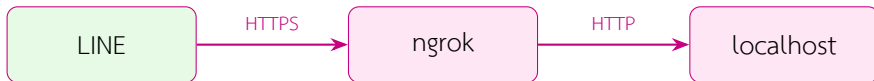
ปัญหา:

- n8n รันที่ `http://localhost:5678`
- LINE ต้องการ **HTTPS** + Public URL
- localhost เข้าถึงจาก Internet ไม่ได้

วิธีแก้: ใช้ ngrok สร้าง Tunnel!

ngrok: สร้าง HTTPS Tunnel

ngrok = Tunneling Service ที่สร้าง Public URL ให้ localhost



ngrok ทำอะไร:

- สร้าง Public URL: <https://xxxx.ngrok-free.app>
- Forward traffic ไปยัง localhost:5678
- จัดการ SSL/HTTPS ให้อัตโนมัติ

วิธีใช้ ngrok

1. สมัครบัญชี: <https://ngrok.com>

2. ติดตั้ง:

```
# Windows (Chocolatey)  
choco install ngrok
```

```
# macOS (Homebrew)  
brew install ngrok
```

3. เพิ่ม Authtoken:

```
ngrok config add-authtoken YOUR_TOKEN
```

4. รัน:

```
ngrok http 5678
```

ngrok Output

Session Status	online
Account	your-email@example.com
Version	3.x.x
Region	Asia Pacific (ap)
Forwarding	https://abc123.ngrok-free.app -> localhost:5678

URL ที่ได้: <https://abc123.ngrok-free.app>

Webhook URL สำหรับ LINE:

<https://abc123.ngrok-free.app/webhook/gold-bot>

หมายเหตุ: URL จะเปลี่ยนทุกครั้งที่รัน ngrok ใหม่ (Free plan)

HTTP Methods

Method	ใช้ทำอะไร	ตัวอย่าง
GET	ดึงข้อมูล (Read)	ดึงราคาทอง
POST	สร้างข้อมูล (Create)	ส่งข้อความ LINE
PUT	แก้ไขทั้งหมด (Update)	อัปเดต Profile
PATCH	แก้ไขบางส่วน	แก้ไขบาง field
DELETE	ลบข้อมูล	ลบ record

Lab นี้ใช้ GET และ POST

n8n HTTP Request Node

HTTP Request Node = Node สำหรับเรียก API ภายนอก

Settings ที่สำคัญ:

- **Method:** GET, POST, PUT, DELETE
- **URL:** API Endpoint
- **Authentication:** API Key, Bearer Token, OAuth
- **Headers:** Content-Type, Authorization
- **Body:** JSON data (สำหรับ POST)

GET vs POST ใน Lab นี้

GET - ดึงราคาทอง:

```
Method: GET
URL: https://api.chnwt.dev
    /thai-gold-api/latest

Headers: {}
Body: {}
```

POST - ส่งข้อความ LINE:

```
Method: POST
URL: https://api.line.me
    /v2/bot/message/reply

Headers:
  Authorization: Bearer xxx
  Content-Type: application/json

Body: { replyToken, messages }
```


Thai Gold API

API ดึงราคาทองจากสมาคมค้าทองคำ

รายการ	ค่า
URL	https://api.chnwt.dev/thai-gold-api/latest
Method	GET
Authentication	ไม่ต้อง (Free)
Rate Limit	ไม่จำกัด

ข้อมูลที่ได้:

- ราคาทองคำแท่ง (gold_bar)
- ราคาทองรูปพรรณ (gold)
- วันที่/เวลาอัปเดต

Thai Gold API Response

```
{
  "status": "success",
  "response": {
    "update_date": "03/02/2569",
    "update_time": "16:05 ๒.",
    "price": {
      "gold": {
        "buy": "71,631.00",
        "sell": "74,100.00"
      },
      "gold_bar": {
        "buy": "73,100.00",
        "sell": "73,300.00"
      }
    }
  }
}
```

โครงสร้างข้อมูลราคาทอง

Field	Path	คำอธิบาย
ทองคำแท่ง ขาย	response.price.gold_bar.sell	ราคาขายออก
ทองคำแท่ง ซื้อ	response.price.gold_bar.buy	ราคารับซื้อ
ทองรูปพรรณ ขาย	response.price.gold.sell	ราคาขายออก
ทองรูปพรรณ ซื้อ	response.price.gold.buy	ราคารับซื้อ
วันที่	response.update_date	วันที่อัปเดต
เวลา	response.update_time	เวลาอัปเดต

gold = ทองรูปพรรณ, gold_bar = ทองคำแท่ง

LINE Reply API

API สำหรับส่งข้อความตอบกลับผู้ใช้

รายการ	ค่า
URL	https://api.line.me/v2/bot/message/reply
Method	POST
Authentication	Bearer Token
Content-Type	application/json

ข้อควรรู้:

- ต้องใช้ **replyToken** จาก Webhook Event
- replyToken ใช้ได้ **ครั้งเดียว**
- ส่งได้สูงสุด **5 messages** ต่อ 1 reply

LINE Reply API Request Body

```
{
  "replyToken": "xxx-reply-token-xxx",
  "messages": [
    {
      "type": "text",
      "text": "Hello, World!"
    }
  ]
}
```

Headers ที่ต้องใส่:

```
Authorization: Bearer YOUR_CHANNEL_ACCESS_TOKEN
Content-Type: application/json
```

Reply API vs Push API

หัวข้อ	Reply API	Push API
ใช้เมื่อ	ตอบกลับ Event	ส่งเอง (ไม่ต้องรอ Event)
ต้องการ	replyToken	userId
ค่าใช้จ่าย	ฟรี	คิดเงิน (ถ้าเกิน quota)
Lab นี้ใช้	ใช่	ไม่ใช่

Lab นี้ใช้ Reply API เพราะฟรี!

Code Node ใน Lab นี้

Code Node ใช้ทำอะไร?

- 1 ดึงข้อมูลจาก Gold API Response
- 2 ดึง replyToken จาก LINE Event
- 3 จัดรูปแบบข้อความให้สวยงาม
- 4 สร้าง JSON สำหรับ LINE Reply API

Input:

- Gold API Response (ราคาทอง)
- LINE Webhook Data (replyToken)

Output:

- JSON พร้อมส่ง LINE Reply API

Code Node: ดึงข้อมูล

```
// ดึงข้อมูล Gold API
const goldData = $input.first().json;
const response = goldData.response;

// ดึงข้อมูล
const goldBarSell = response.price.gold_bar.sell;
const goldBarBuy = response.price.gold_bar.buy;

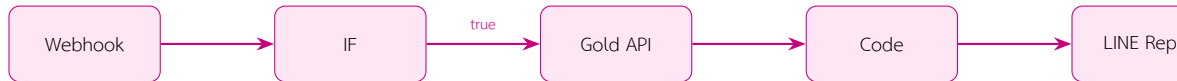
// ส่ง replyToken ไป Webhook
const webhookData = $('Webhook').first().json;
const replyToken = webhookData.body.events[0].replyToken;
```


Code Node: สร้างข้อความ

```
const message = `*****`
:
  *****: ${goldBarSell} 
  *****: ${goldBarBuy} `;

return [{
  json: {
    replyToken: replyToken,
    messages: [{
      type: "text",
      text: message
    }]
  }
}];
```

Workflow Overview



Node	Type	หน้าที่
Webhook	webhook	รับ LINE Event
IF	if	ตรวจสอบคำสั่ง
Gold API	httpRequest	ดึงราคาทอง
Code	code	จัดรูปแบบ
LINE Reply	httpRequest	ส่งตอบกลับ

Data Flow

1. LINE ส่ง Webhook:

- Event type, message text, replyToken

2. IF ตรวจสอบ:

- message.text มี "ราคาทอง" หรือ "gold"?

3. Gold API ตอบกลับ:

- gold_bar.sell, gold_bar.buy, gold.sell, gold.buy

4. Code จัดรูปแบบ:

- สร้างข้อความ + replyToken

5. LINE Reply ส่งกลับ:

- ผู้ใช้เห็นข้อความราคาทอง

วันนี้เรียนรู้:

- 1 LINE Messaging API - สร้าง Bot รับ-ส่งข้อความ
- 2 ngrok - สร้าง HTTPS Tunnel สำหรับ Webhook
- 3 HTTP Request Node - GET ดึงข้อมูล, POST ส่งข้อมูล
- 4 Thai Gold API - ดึงราคาทองฟรี
- 5 LINE Reply API - ส่งข้อความตอบกลับ
- 6 Code Node - จัดรูปแบบข้อความ

Lab 03: LINE Gold Price Bot

สิ่งที่ต้องทำ:

- ❶ สร้าง LINE Messaging API Channel
- ❷ ติดตั้งและรัน ngrok
- ❸ สร้าง Workflow ใน n8n (5 Nodes)
- ❹ ตั้งค่า Webhook URL ใน LINE
- ❺ ทดสอบส่ง "ราคาทอง" ใน LINE
- ❻ Export workflow.json และ push ขึ้น GitHub

คะแนน: 100 คะแนน (10 Tests)

มีคำถามไหม?

ดู Lab Instructions: README.md

ดูคู่มือตั้งค่า: SETUP_GUIDE.md