

n8n Installation

Topics

1. ติดตั้ง n8n self-host บนคอมพิวเตอร์ ของตัวเอง
2. ตั้งค่า ngrok เพื่อใช้กับ LINE Messaging API
3. เปิดบัญชี LINE ทดสอบการเชื่อมต่อ Messaging API
4. สร้าง Chatbot ด้วย Gemini

Docker N8N – Installation Steps

- Open Docker Desktop Terminal and run : `docker run -it --rm --name n8n -p 5678:5678 -v [your path]:/home/node/.n8n`

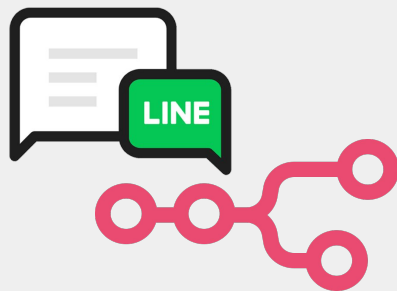
docker.n8n.io/n8nio/n8n

- Create a new container at :

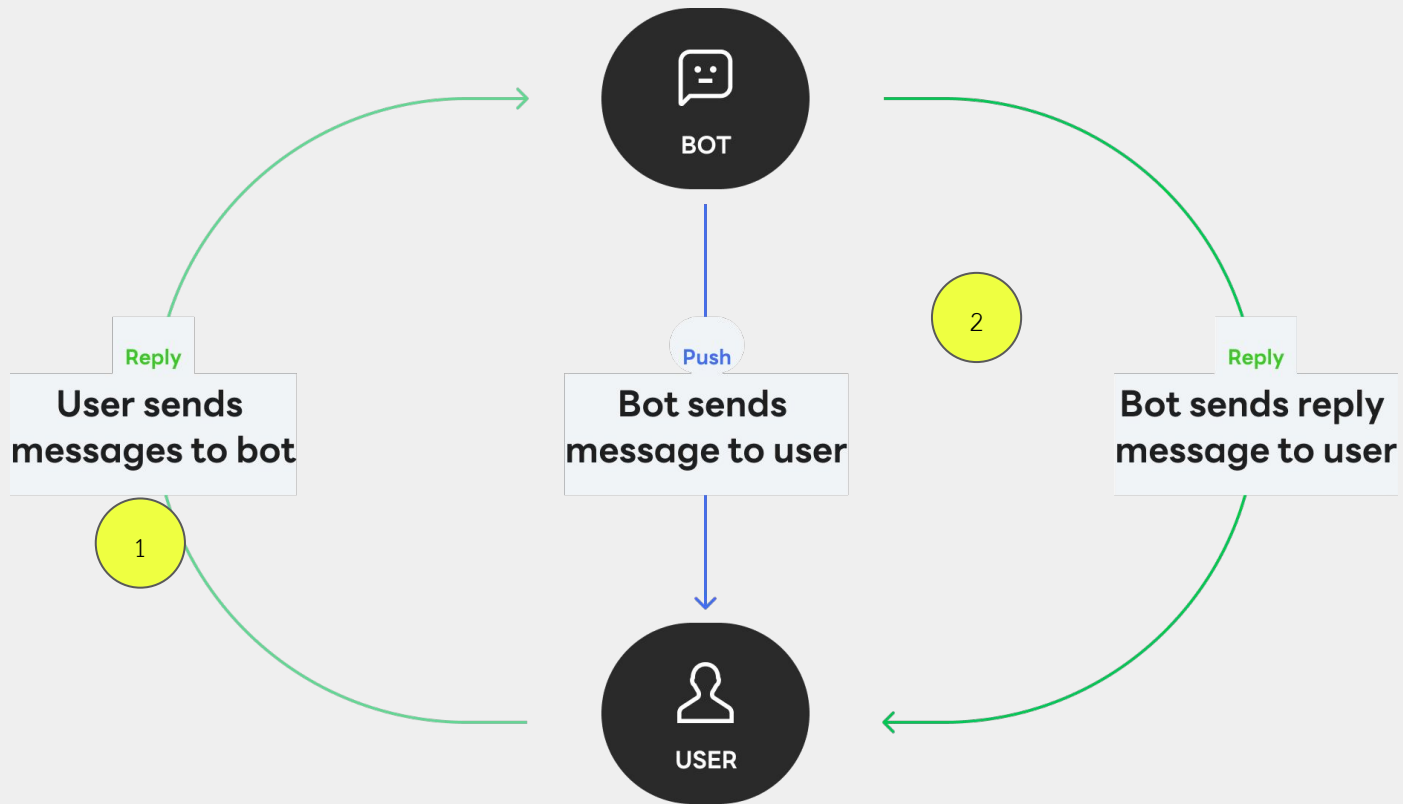
Host Ports:	[ANY_PORT]		
Host Path :	[Your Document path]	Container Path :	/home/node.n8n
Environment Variable :			
Variable :	N8N_COMMUNITY_PACKAGES_ALLOW_TOOL_USAGE	Value :	TRUE
Variable :	N8N_EDITOR_BASE_URL	Value :	[NGROK_PUBLIC_URL]
Variable :	WEBHOOK_URL	Value :	[NGROK_PUBLIC_URL]
Variable :	N8N_DEFAULT_BINARY_DATA_MODE	Value :	filesystem

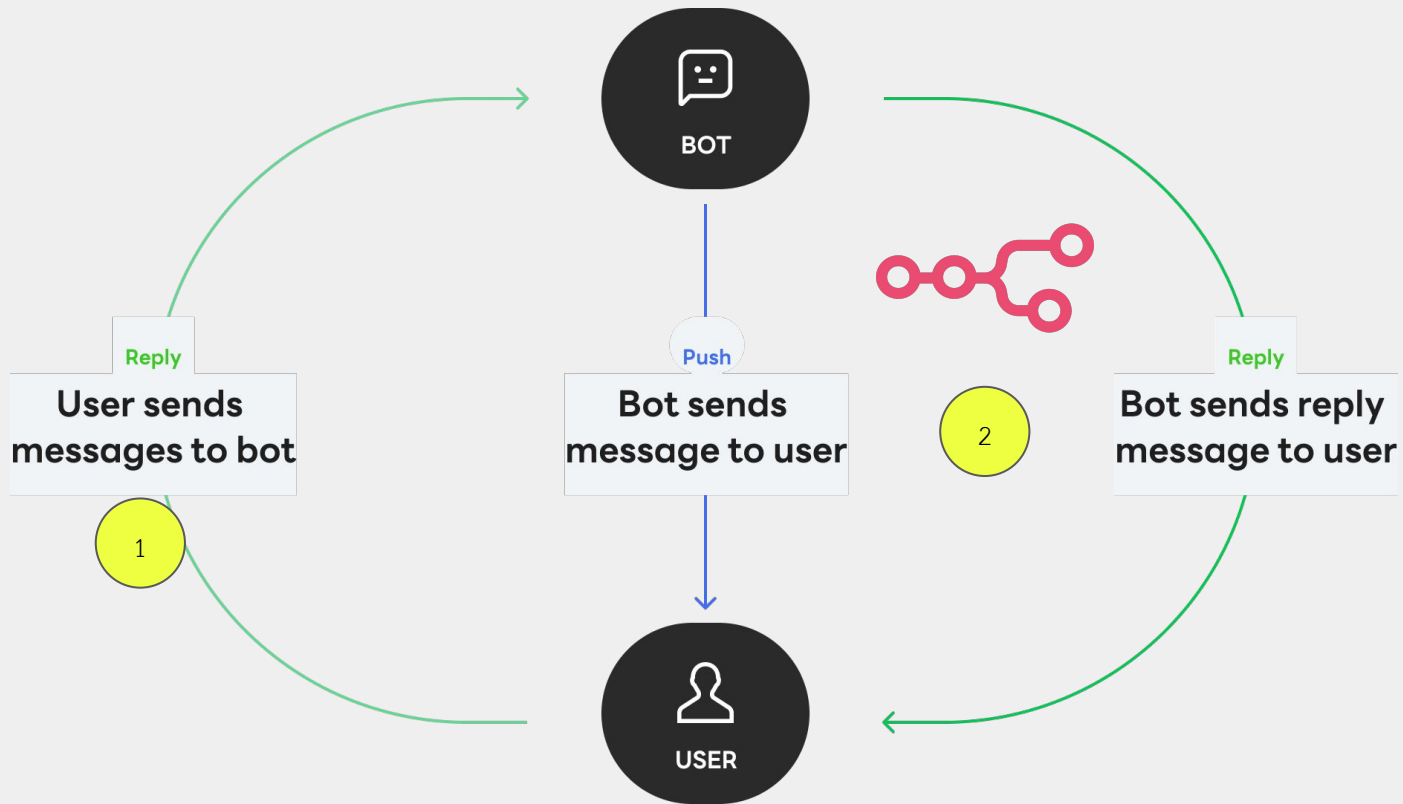
Ngrok – Installation Steps

- Login and Install : [ngrok Install](#)
- Open Ngrok terminal and run :
 - `ngrok config add-authtoken [AUTH_TOKEN]`
 - `ngrok http --url=[NGROK_PUBLIC_URL] [N8N_PORT]`



n8n + LINE API

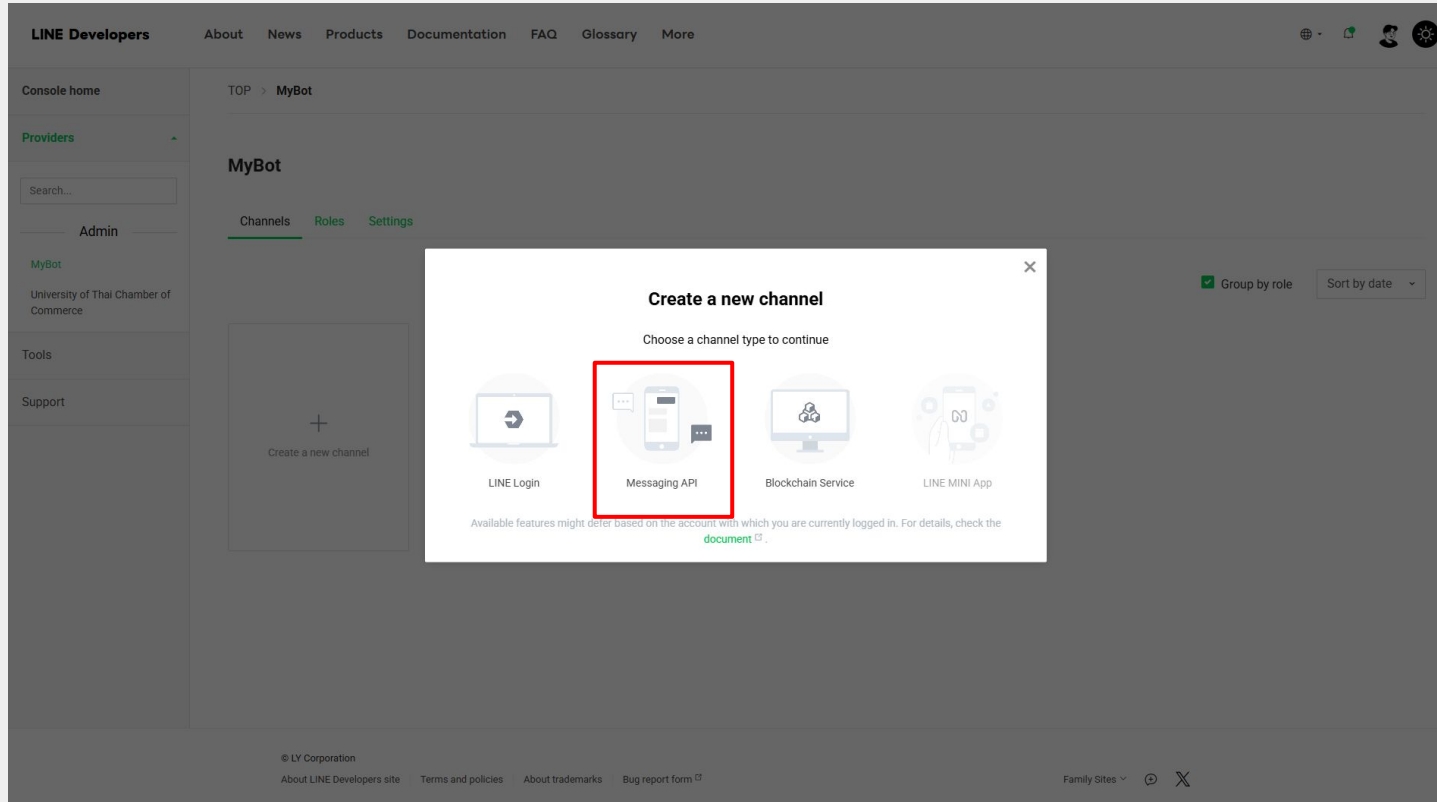




Line API Setup

- Create LINE [Developer](#) account [here](#)
- Add Webhook URL
- Config messaging API
- TEST

Step 1 : Create Line Developers Account choose **Messaging API**



The screenshot displays the LINE Developers console interface. A modal dialog titled "Create a new channel" is centered on the screen, prompting the user to "Choose a channel type to continue". The dialog lists four options: "LINE Login", "Messaging API", "Blockchain Service", and "LINE MINI App". The "Messaging API" option is highlighted with a red rectangular border. Below the options, a note states: "Available features might differ based on the account with which you are currently logged in. For details, check the [document](#)".

The background console shows the "MyBot" management page with tabs for "Channels", "Roles", and "Settings". The left sidebar includes navigation links for "Providers", "Admin", "Tools", and "Support". The footer contains copyright information for LY Corporation and links to the "About LINE Developers site", "Terms and policies", "About trademarks", and "Bug report form".

Step 2 : Select a Messaging API option and open the Use Webhook settings.

LINE Developers About News Products Documentation FAQ Glossary More

Console home
MyBot
Tools
Support

TOP > MyBot > QWERTY > **Messaging API**

[Basic settings](#) **Messaging API** [LIFF](#) [Security](#) [Statistics](#) [Roles](#)

Messaging API settings

Bot information

Bot basic ID

QR code

Scan this QR code with LINE to add your LINE Official Account as a friend. You can share the code with others.

Webhook settings

Webhook URL ⓘ :990 **1**


Use webhook ⓘ ☒ **2**

Step 2 : Copy channel access token and paste into the header auth account token

``Bearer [CHANNEL_ACCESS_TOKEN]``


Console home

TOP > MyBot > QWERTY > Messaging API


Error statistics aggregation 

LINE Official Account features


Edit the message text and other settings for these features in the LINE Official Account Manager

Allow bot to join group chats 

Disabled


Auto-reply messages 

Enabled


Greeting messages 

Disabled

Channel access token

Channel access token (long-lived) 

W/h
89/

1B04t 

Reissue

Header Auth account

Connection


Need help filling out these fields? [Open docs](#)

Name

Authorization

Value

To send multiple headers, use a "Custom Auth" credential instead

 Enterprise plan users can pull in credentials from external vaults. [More info](#)

Step 3 : Back into the Messaging API then click Verify (A success popup will be shown)

The screenshot displays the LINE Developers console interface. The top navigation bar includes links for About, News, Products, Documentation, FAQ, Glossary, and More. The left sidebar shows the Console home, MyBot, Tools, and Support sections. The main content area is titled 'Messaging API' and contains several tabs: Basic settings, Messaging API (selected), LIFF, Security, Statistics, and Roles. The 'Messaging API settings' section is active, showing Bot information (Bot basic ID), a QR code, and Webhook settings. The Webhook settings section includes a Webhook URL, a 'Verify' button (highlighted with a yellow circle and the number 1), an 'Edit' button, a 'Use webhook' toggle (checked), a 'Webhook redelivery' toggle, and an 'Error statistics aggregation' toggle. A modal window is overlaid on the right side of the screen, displaying a QR code and a 'Success' message with an 'OK' button. A yellow arrow points from the 'Verify' button in the Webhook settings to the modal window.

Step 4 : Go back to the n8n page and click on 'Listen for test Event'. If there is an OUTPUT, it indicates that the setup is correct.

← Back to canvas

Webhook

Listen for test event

Parameters

Settings

Docs

Webhook URLs

Test URL

Production URL

POST

https://n8n.srv798680.hstgr.cloud/webhook-test/63f45384-2f73-4061-b0ad-6ea328aac990

HTTP Method

POST

Path

63f45384-2f73-4061-b0ad-6ea328aac990

Authentication

None

Respond

Immediately

Options

No properties

Add option

Pull in events from Webhook

Listen for test event

Once you've finished building your workflow, run it without having to click this button by using the production webhook URL. [More info](#)

When will this node trigger my flow?

OUTPUT

1 item

headers	params	query	body	webhookUrl
host: n8n.srv798680.hstgr.cloud	empty	empty	destination: Ua0ab847f5f6ed11a244d7aac67	https://n8n.srv798680.hstgr.cloud
user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7; rv:109.0) Gecko/20100101 Firefox/109.0				
content-type: application/json				
content-length: 104				
x-forwarded-for: 192.168.1.1				
x-forwarded-host: n8n.srv798680.hstgr.cloud				
x-forwarded-port: 5424				
x-forwarded-proto: https				
x-lim: 1				
sk: 1				
XI: 1				
x-re: 1				
accept-encoding: gzip				

I wish this node would...

test workflow

HTTP REQUEST : JSON Config

```
{{  
  
  JSON.stringify(  
  
    replyToken: $('Edit Fields').item.json.body.events[0].replyToken,  
  
    messages: [  
  
      {  
  
        type: "text",  
  
        text: $json.output  
  
      }  
  
    ]  
  
  )  
  
}}
```