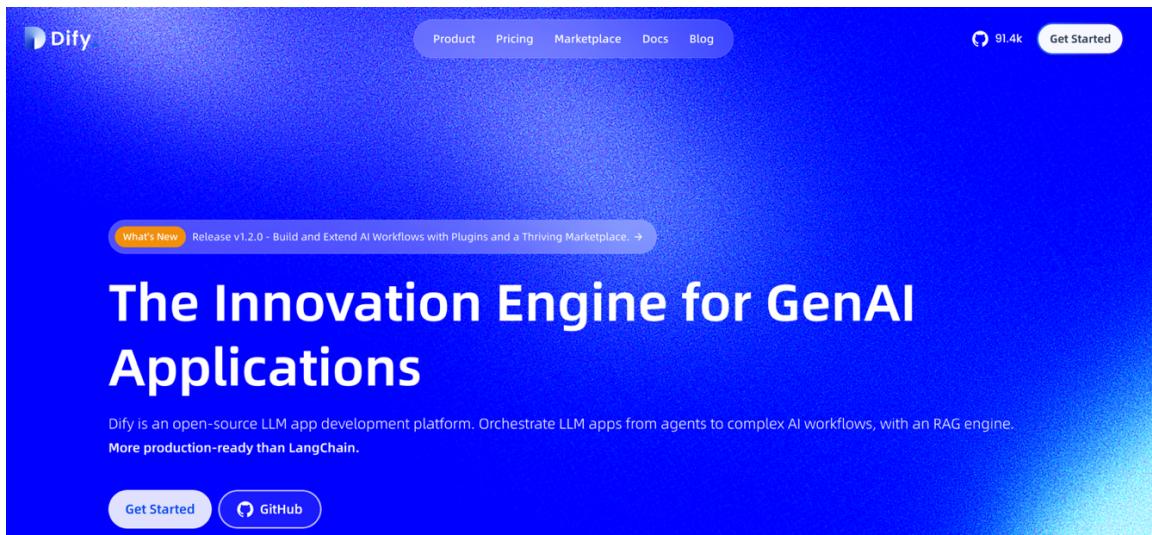


## 1. Introduction to Dify.ai

國立清華大學 區國良

Dify 是一個開源工具，可使用雲端版及自行下載安裝，後者需以 docker 方式安裝，雲端版則有容量及查詢次數的限制(且伺服器據說在對岸，使用時請自行評估資料機敏性)，本說明手冊以雲端版為例，以 Google 或 Github 帳號直接登入即可使用。請點選 Get Started

<https://dify.ai/>



登入後，上方頁簽提供「探索」、「工作室」、「知識庫」及「工具」選項，說明如下：



### 1.1. 探索

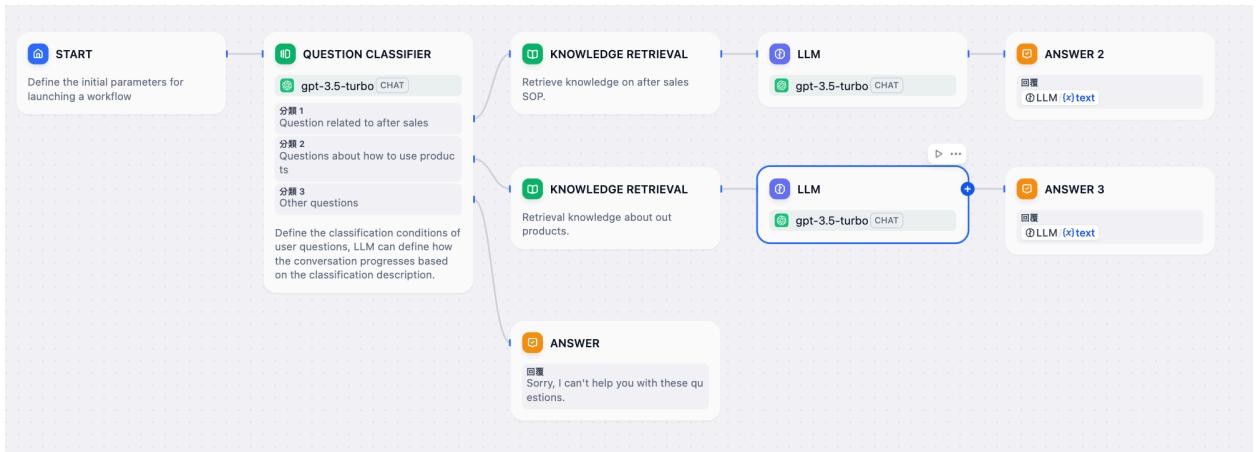
提供內建的樣板，提供快速布建，再依自己的需求修改，例如以下為 **Question Classifier + Knowledge + Chatbot** 的布建及修改過程，首先點擊「探索」頁簽後，再點擊 **Question Classifier + Knowledge + Chatbot** 下方的「新增到工作區」，接著可自訂應用名稱及圖示，若不需修改，直接點擊「建立」即可，不同的樣板可能會需要個別安裝第三方的外掛，部分外掛還需要另外註冊，請自行判斷是否需要。



免費版最多可製作或安裝 5 個應用程序，若達到上限時，在新增應用程式時則出現如右上的警告，如果想在免費額度內繼續新增應用程式，則請回到工作室中自行刪除不必要的項目即可。本範例預設使用 OpenAI 提供之 LLM，其它的 LLM 可利用右上角的外掛自行安裝，安裝完成後，右上角的外掛按鈕可見其支援的模型型號，以下為若想使用 Anthropic LLM 為例，需另外申請 [Claude](#)（Anthropic 的聊天機器人）的帳戶，即可利用其 [API](#) 串接使用。



完成外掛及應用程序的安裝後，將自動於「工作室」中展開，有些程序會貼心附上說明便利貼，使用者可自行依個人需求修改。以下圖為例，整個流程開始於最左方的 Start，點擊右上方的「預覽」可以試用看看。

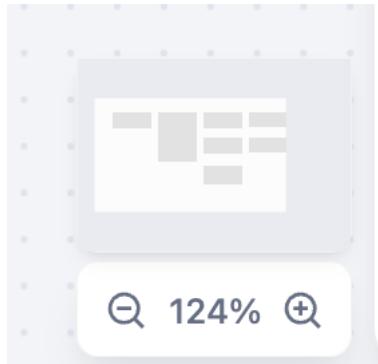


也可以展開「工作流」的內容，觀看處理的情況是否順暢，以及花費多少時間、多少 token 量等資訊，如下圖：

## 1.2. 工作室

工作室即流程控制的主要核心，原則上大部分模版應該是直接可用，但仍有少部分參數及流程需要稍微調整後才能使用。以 Question Classifier + Knowledge + Chatbot 模版為例，其最左方為 START 節點，為所有對話流程開啟時第一個會執行的節點，每一次對話都會由 START 開始執行。本模版中，若順利執行後，下一個流程為 QUESTIONCLASSIFIER，利用 GPT 模型判別使用者輸入的問題是屬於哪一種類型，分類 1 為 Question related to after sales，分類 2 為 Question about how to use products，若非以上分類，則屬於分類 3，其中分類 1 及分類 2 的問題分別引導到不同的 KNOWLEDGE RETRIEVAL 節點，以及不同的 LLM 節點，最後以 ANSWER 節點回答使用者，而分類 3 則直接回答”Sorry, I can't help you with these questions.”並且結束本回合的問答，每次使用者提問時都會由 START 開始新一回合的流程。

有時流程超過一個螢幕的空間時，可利用右下角的放大縮小鍵，以及該鍵的上方的地點圖示移動顯示區域。



若欲修改流程，可在每個節點的後方以滑鼠點擊添加或拖曳的方式修改流程線，當鼠標移到+號之上時，系統會提示可增加的節點或工具：

The image displays two side-by-side screenshots of the Dify workspace interface.

**Left Screenshot (Nodes Panel):**

- Header: 節點 工具
- Search bar: 搜尋節點
- List of nodes:
  - LLM
  - 知識檢索
  - 直接回覆
  - 代理
  - 問題理解
  - 問題分類器
  - 邏輯
  - 條件分支
  - 迭代
  - 循環
  - 轉換
  - 代碼執行
  - 模板轉換
  - 變量聚合器
  - 文件提取器
  - 變數分配器
  - 參數提取器
  - 工具
  - HTTP 請求
  - 清單運算子

**Right Screenshot (Tools Panel):**

- Header: 節點 工具
- Search bar: 搜尋工具...
- Filter buttons: 全部 插件 自定義 工作流 MCP
- List of tools:
  - Audio hjlarry
  - Code Interpreter Dify
  - CurrentTime Dify
  - DuckDuckGo yash\_parmar
  - DALL-E langgenius
  - Discord Ice Yao
  - Google Translate svcvit
  - Google Dify
  - Google Sheets omluc
  - Spider William Espegren
  - WebScraper Dify
- Text at the bottom: 在 Marketplace 中查找更多內容 ↗

若欲刪除節點間的連結線時，請點選該連線線後，按鍵盤的 Del 處理。

### 1.3. 知識庫

知識庫 RAG 工具，可預先上傳文件作為知識庫，在工作流中可設定成為 LLM 查詢回覆時使用，為比較使用原生 LLM 對話與使用知識庫對話的差異，我們先到工作室中建立一個單純聊天的機器人，首先在工作室中的左方「建立應用」選單中，點選「建立空白應用」，Dify 提供五種空白應用環境，上方為適合初學者的聊天助手、Agent 以及文字生成應用，下方為高級使用者適用之聊天流及工作流，請先點選聊天助手，並自行命名應用名稱：

#### 建立空白應用

##### 選擇 App Type

對於初學者



對於高級使用者



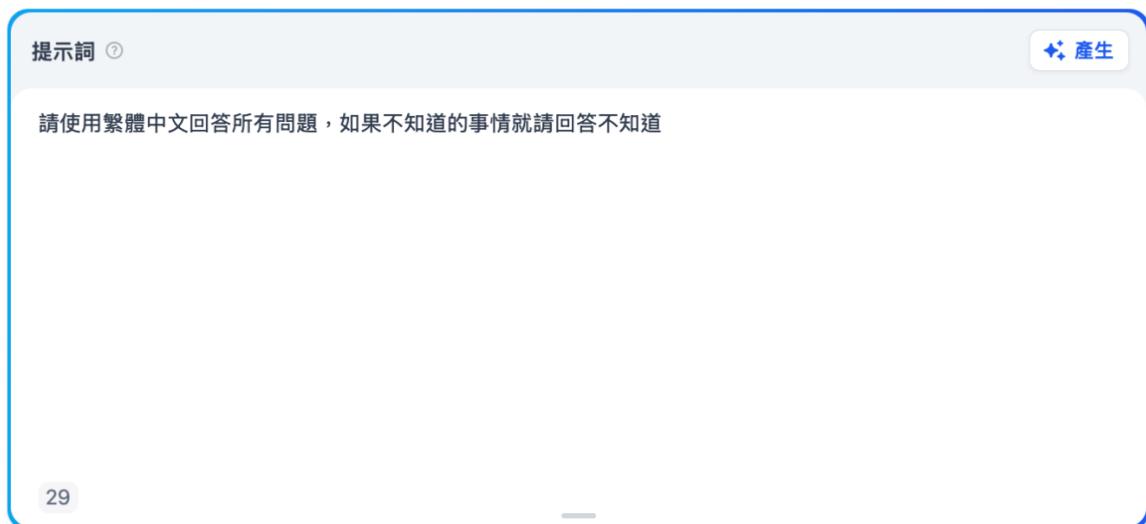
##### 應用名稱 & 圖示

純聊天與RAG測試



接著可利用「編排」欄位自行設定提示詞，也可以保留空白作為使用原生 LLM 預設環境的純聊天使用，但通常至少提示聊天時請 LLM 應配合的事項，例如使用的文字及偏好等，如果不善於自行定義提示詞，也可利用右上角的「產生」功能建立更完善的提示詞。

## 編排



目前建立的純聊天機器人，其實就是透過 Dify 去呼叫指定的 LLM（在發佈按鈕旁可切換不同的 LLM），此時的聊天內容無法聚焦在指定的文本之中，例如我在預覽對話功能中，請 ChatGPT 40-mini 介紹區國良，得到的結果如下，實在令人捏把冷汗。。。

The interface shows a generated response from 'gpt-4o-mini-2024-07-18' in the 'CHAT' mode. The input text is '請簡要介紹區國良' (Please introduce Qu Guoliang briefly). The response is: '區國良是一位中國知名的學者和社會活動家，專注於政治學、公共政策及社會科學等領域。他在學術界有著廣泛的影響力，並經常參與社會運動和公共事務，倡導民主和人權。區國良的研究主要集中在中國的社會變遷、政府治理以及公民社會等方面。他的著作和講座在學術界和社會上引起了不少關注與討論。' Below the response, it says '耗時 2.51 秒 花費 Token 155 · 11:05 PM'.

這時就需要 RAG 的功能將回答的範圍加以設限，請在編排的下方找到「上下文」，其右方的新增按鈕將引導對話內容需參考知識庫，若還沒建立知識庫，則會引導使用者建立知識庫。

### 選擇資料來源



匯入已有文字



同步自 Notion 內容



同步自 Web 站點

### 選擇供應商



Jina Reader



Firecrawl



WaterCrawl



未配置 Jina Reader

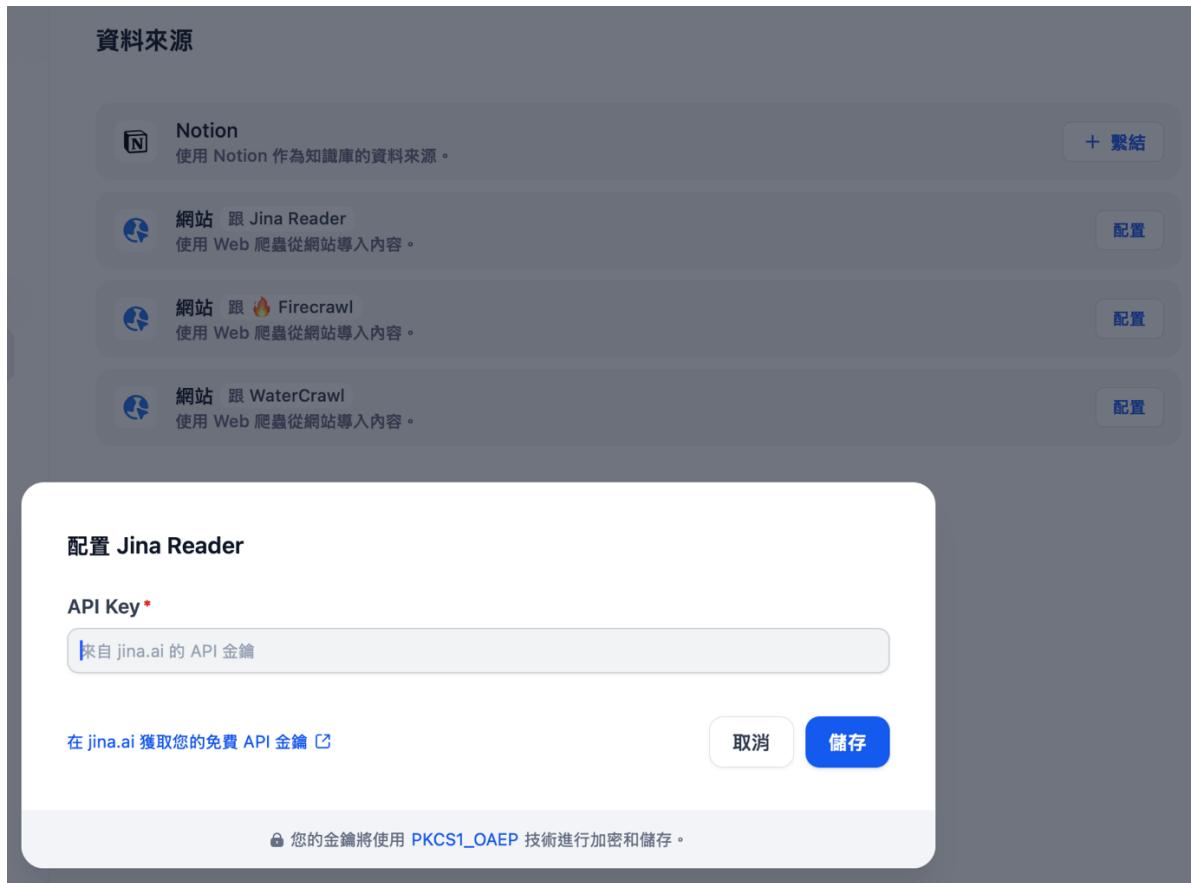
通過輸入您的免費 API 金鑰進行訪問來設置 Jina Reader。

配置

知識庫支援文字，notion 以及網站，我們可以先測試同步自 web 網站，並使用其中三個外部應用程式的供應商進行網站爬梳作業，以下是以 Jina reader 為例，請註冊登入後，將 api key 複製。



再回到 Dify 的資源來源選單中，配置 Jina Reader



成功後即可使用指定的網站作為資訊庫了，以下是以區國良的官方網站為例，當按下「跑」後，可見其爬梳的進度，並可預覽爬到的網頁是否符合需求，在下一步中，可指定其分段及索引的方式，影響最大的是 embedding 模型，以及 Rerank 服務（Reranker 又稱為二階檢索，依輸入的兩段文字之相似度，輸出一個 0~1 之間的相關性分數），目前 Dify.ai 支援的 Rerank 只有 Cohere，請自行測試看看。

#### 選擇資料來源



匯入已有文字



同步自 Notion 內容



同步自 Web 站點

#### 選擇供應商



Jina Reader



Firecrawl



WaterCrawl

將整個網站轉換為 Markdown

Configure Jina Reader

了解有關 Jina Reader 的更多資訊

<https://klou.ilst.nthu.edu.tw/>

跑

有了指定網站內容作為知識庫時，這時若再詢問一次相同的問題，答案除了參考知識庫的內容外，LLM 還會加油添醋地提出它自己的看法，真是孺子可教也！

The screenshot shows the Dify AI interface. At the top, there's a search bar with the placeholder "請簡要介紹區國良" and a blue circular button with a white letter "D". Below the search bar is a profile icon of a robot head. The main content area contains text about Professor Kuo Lian, followed by a "引用" (Citation) section with a link to <https://klou.ilst.nthu.edu.tw/?page...>.

知識庫可支援多種文件格式，若要將學術論文作為知識庫內容，可分批上傳，Dify 會將文件切割成若干個 Chunk，以作為日後回答問題時參考的來源，每個 chunk 將會與使用者的問題進行相似度分析，下例為某文件的「召回測試」，該文件為 Computers & Education 的期刊論文，標題為 Identifying supportive student factors for mindset interventions: A two-model machine learning approach，我們可以試著以 200 字以內的問題進行測試，例如在源文字處輸入 how to employ the machine learning techniques in educational mindset intervention，系統回應 3 個召回段落，很明顯 Score 為 0.94 的 chunk 比 Score 0.71 的 chunk 更接近問題字串。

The screenshot shows a recall test interface. On the left, a "源文字" (Source Text) input field contains the sentence "how to employ the machine learning techniques in educational mindset intervention". To its right is a "混合檢索" (Mixed Search) button. On the right, there are two search results under the heading "召回段落" (Recall Paragraph). The first result, "Chunk-09", has a score of "SCORE 0.94" and the text "However, it is worth noting that no type of model is better, in general, than any other (known as the "no free lunch" theorem; Schaffer, 1994;... #2019 #Intervent #machine #et #methods #models #complex #ai #learning #student #variables #dataset #intervent #via #students #methods #question #explore #predicto #student #variables #dataset #intervent #via #students #methods #question #explore #predicto #student". The second result, "Chunk-11", has a score of "SCORE 0.71" and the text "Computers & Education 167 (2021) 104190 4... #variables #dataset #intervent #via #students #methods #question #explore #predicto #student #variables #dataset #intervent #via #students #methods #question #explore #predicto #student". Both results include a "打開" (Open) link at the bottom right.

但這只是單純利用知識庫來源作為 LLM 回答的參考，真正的考驗是回工作室中再次詢問看看

是否對話機器人真的有參考知識庫內容，實測如下：

what is the most effective intervention of mindset in previous researches? D

 在先前的研究中，最有效的心態干預之一是針對增長心態的計算機輔助干預。尤其是在2016年的一項研究中，研究者們發現，這種干預顯著提高了第九年級學生的平均學業成績，尤其是對那些學業成績較低的學生。

例如，在一項名為“Refraction”的數學教育遊戲中，研究者加入了旨在促進增長心態的訊息，結果發現實驗組的學生在遊戲中的堅持時間（即參與的時間）有了顯著改善。這些研究結果表明，增長心態能夠幫助學生在面對挑戰和困難時保持堅持。

然而，這些心態干預的效果可能受到學生特徵和其他背景因素的影響，因此在不同的情境中，心態干預的有效性可能會有所不同。

引用

 1-s2.0-S0360131521000671-main... 耗時 6.19 秒 花費 Token 1,840 · 09:21 AM

回 | 叉 | ⌂

大致上可見系統的回答完全是根據論文而生成的，還不錯！

#### 1.4. 工具

依 dify.ai 官網所言，工具已升級為外掛的生態，大致上就是在這裏提供了第三方的外部工具以協助使用者設計工作流，在此不另外說明。

#### 1.5. 綜合練習

現在，一起來練習如何以清華大學新生入學時需閱讀的註冊須知，建立一個對話機器人，可依使用者的問題類型（目前只練習研究生及大學部的身份）提供必要的解答。依前面所建立的「Question Classifier + Knowledge + Chatbot」範例，請展開 QUESTION CLASSIFIER 節點，修改分類 1 及分類 2 的提示內容為「與學士班入學註冊有關的問題」以及「與研究生入學註冊有關的問題」



接下來，分類 1 的下一個節點應為 KNOWLEDGE RETRIEVAL，點開之後，在知識庫增加相對應的文件：

學士班入學須知：

<https://registra.site.nthu.edu.tw/var/file/211/1211/img/2979/557596209.pdf>

研究生入學須知：

<https://registra.site.nthu.edu.tw/var/file/211/1211/img/2979/417141842.pdf>

測試結果如下（測試日期為 2025/8/30）：



✓ 工作流 >

學士班註冊日是在8月27日。



我是大一新生，請問有什麼減免學雜費的方案？



✓ 工作流 >

大一新生可以參考以下減免學雜費的方案：

1. 身障人士子女
2. 身心障礙學生
3. 現役軍人子女
4. 原住民籍學生
5. 特殊境遇家庭之子女
6. 低收入戶子女及中低收入戶子女

申請減免學雜費時，請依照相關規定填寫申請表並上傳相關證明文件。



研究生的健檢日期是何時？



✓ 工作流 >

研究生的健檢日期是在8月26日。



校長是誰？



✓ 工作流 >

Sorry, I can't help you with these questions.

## 2. Dify 連接 Discord

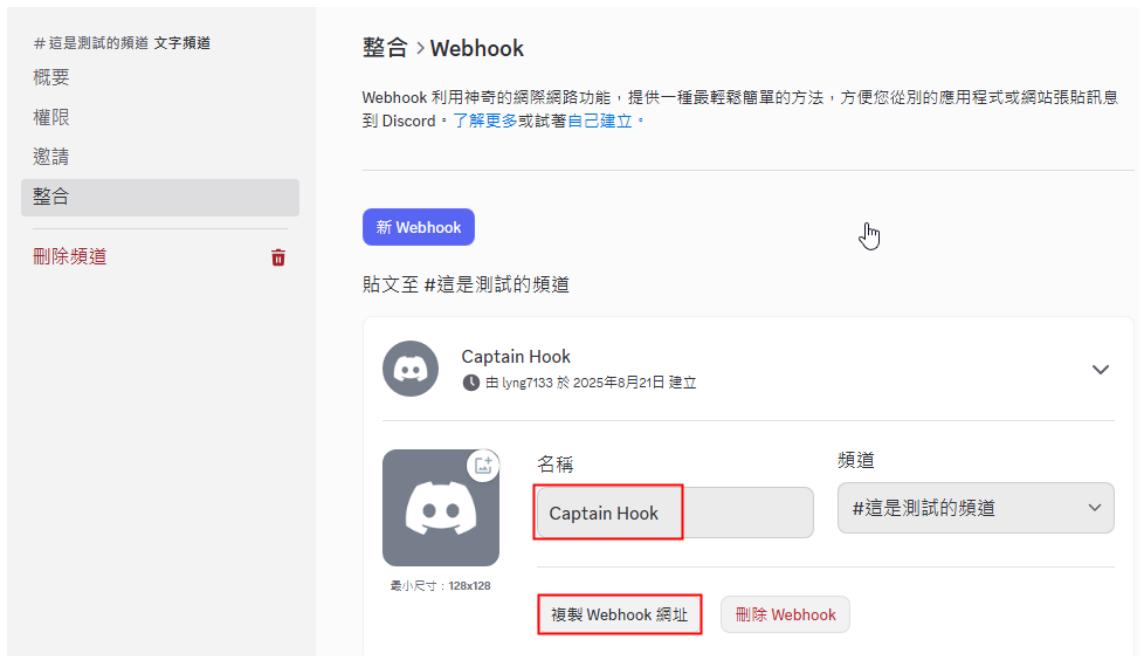
下面章節，說明如何將 Dify 內 OpenAI 的回應，同步回覆於 Discord 中。請注意，目前在 Dify.ai 的 marketplace 提供的外掛工具，只能單方面由 Dify.ai 送訊息到 Discord 指定頻道，不支援 Discord 使用者輸入文字啟動 Dify 工作流。

### 2.1. Discord Webhook 設定

首先在 Discord 系統中，新增一個「伺服器」及「頻道」，再點選「編輯頻道」進入，選擇「整合」，可以看見「建立 Webhook」功能。



按「建立 Webhook」按鈕之後，可以自行修改 Webhook 名稱，並且按「複製 Webhook 網址」，之後於 Dify 設定使用。另外，也可以按「新 Webhook」新增另一組新的 webhook



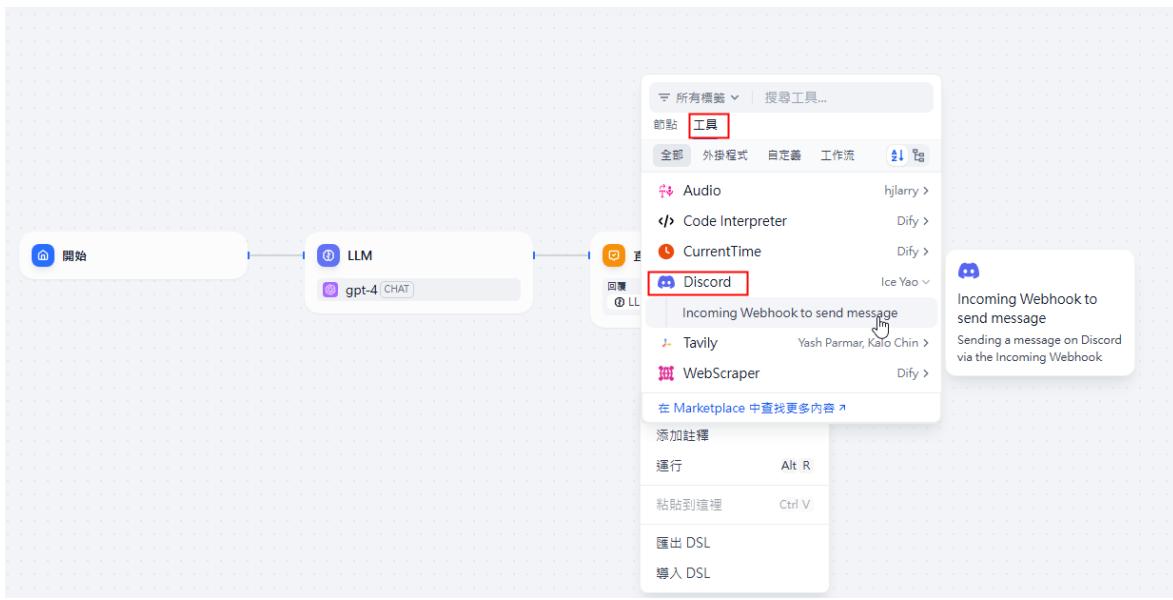
## 2.2. Dify 設定

由「外掛」->「探索 Marketplace」，找到 Discord 外掛，先進行安裝。

The screenshot shows the Dify marketplace interface. The search bar at the top has '探索 Marketplace' (Search Marketplace) highlighted. Below the search bar, there are several categories of plugins:

- Data**: Plugins for data crawling and processing. Examples include Jina AI, Firecrawl, JSON Process, and Regex.
- Developer**: Plugins for software development. Examples include GitHub, DevDocs, Vanna.AI, and WolframAlpha.
- Communication**: Plugins for communication platforms. Examples include Slack, Discord (highlighted with a red box around its 'Install' button), DingTalk, and Wecom.

回到「工作室」的流程中，新增 Discord 節點



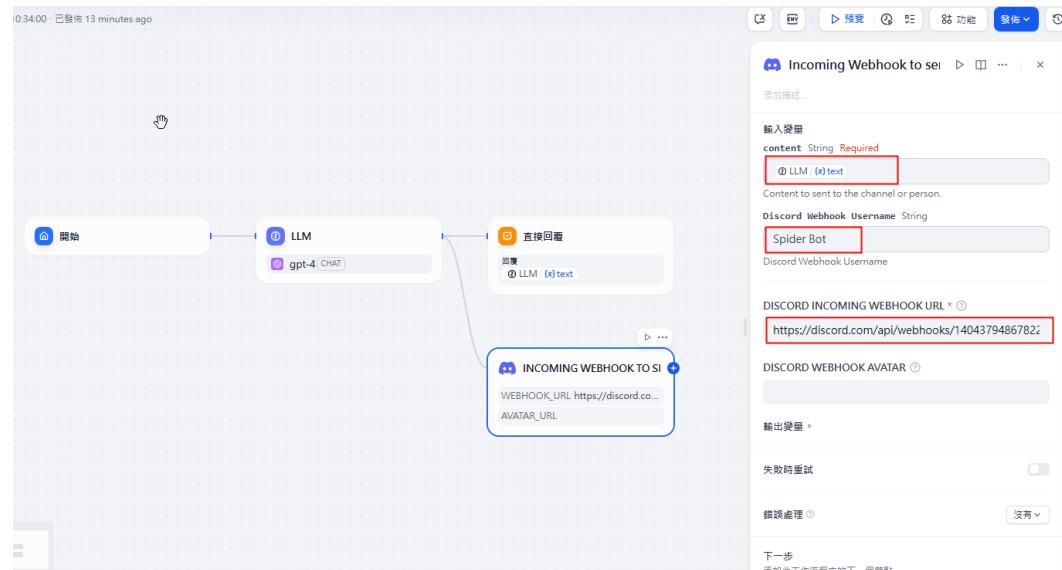
填寫 discord 節點所需的資料內容如下，並參考下圖。

**Content :** 填上希望回應於 Discord 的內容。

**Discord Webhook Username :** 回應時的代號稱呼。

**Discord Incoming Webhook url :** 貼上於 discord 設定時，複製的 webhook 網址。

修改完畢，執行「發佈」。

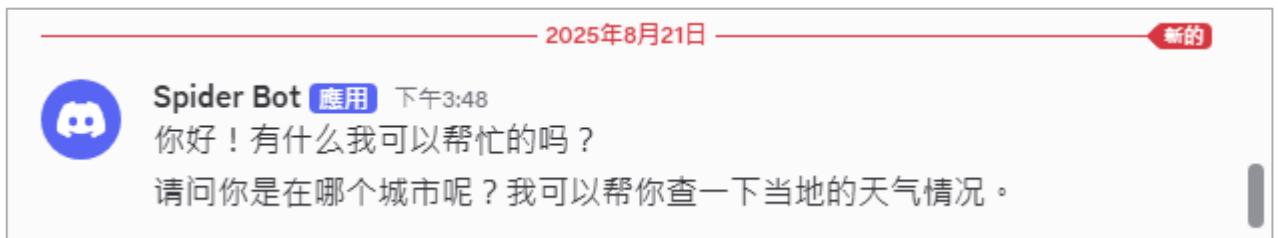


### 2.3. 執行結果

下圖是 Dify 執行結果。



由下圖可見，回應於 Dify 上的內容，同樣也回應於 Discord 我們設定的頻道中，回應者名稱是我們之前設定的 Spider Bot



### 3. LangBot 串接 Discord 與 Dify

下面章節說明使用 LangBot 串接 Discord 與 Dify，以支援 Dify 與 Discord 雙向互動，但只能以本地端安裝的 Dify 系統才能使用，雲端的 Dify.ai 無法提供相關服務。

LangBot 系統會擷取 Discord 上的使用者訊息，將其傳送至 Dify 進行處理，將回應內容自動回覆至 Discord。可以參考下面網址，閱讀 LangBot 的說明文件，

<https://docs.langbot.app/zh/insight/guide.html>。

#### 3.1. 安裝 LangBot

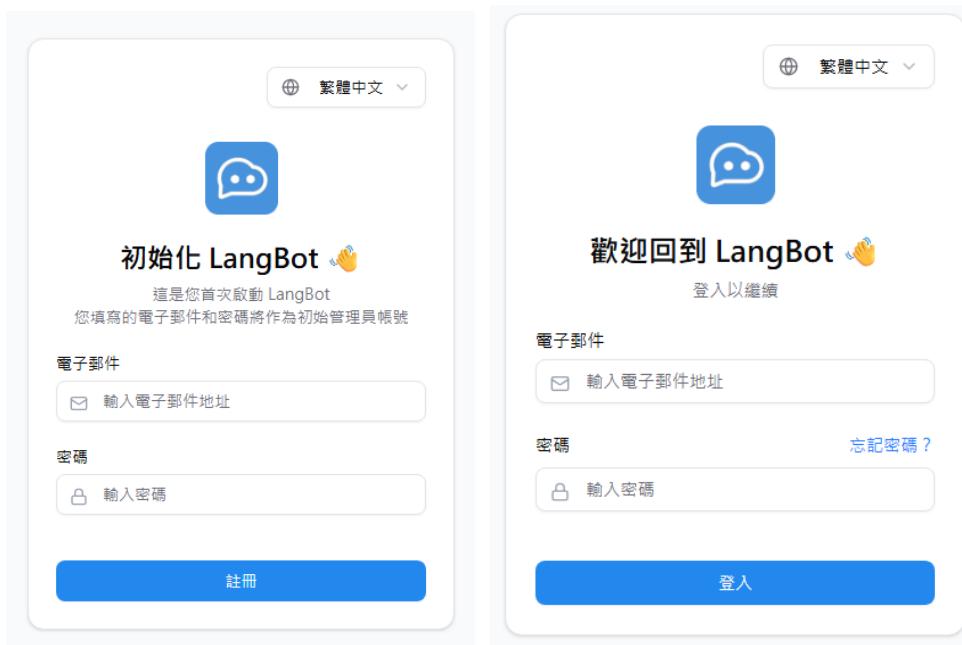
參考 <https://docs.langbot.app/zh/deploy/langbot/docker.html>，使用 docker 安裝。

- git clone https://github.com/langbot-app/LangBot
- cd LangBot
- docker compose up

執行 docker compose up 後，出現下列畫面，開啟 port 5300。可以連 <http://安裝的位置:5300>，開始進行 LangBot 的設定。

```
attaching to langbot
langbot | /app/.venv/lib/python3.12/site-packages/nakuru/misc.py:50: SyntaxWarning: invalid escape sequence '\['
langbot |     for m in re.compile("(\[Q:(.+?)\])").finditer(text):
langbot | [08-12 13:46:47.795] mgr.py (39) - [INFO] : Initializing database...
langbot | [08-12 13:46:48.014] mgr.py (57) - [INFO] : Creating initial metadata...
langbot | [08-12 13:46:48.039] mgr.py (71) - [INFO] : Creating default pipeline...
langbot | [08-12 13:46:48.070] mgr.py (118) - [INFO] : Migration 1 completed.
langbot | [08-12 13:46:48.079] mgr.py (118) - [INFO] : Migration 2 completed.
langbot | [08-12 13:46:48.086] mgr.py (118) - [INFO] : Migration 3 completed.
langbot | [08-12 13:46:48.093] mgr.py (118) - [INFO] : Migration 4 completed.
langbot | [08-12 13:46:48.093] mgr.py (120) - [INFO] : Successfully upgraded database to version 4.
langbot | [08-12 13:46:48.094] manager.py (73) - [INFO] : Loading all plugins...
langbot | [08-12 13:46:48.127] modelmgr.py (61) - [INFO] : Loading models from db...
langbot | [08-12 13:46:49.083] botmgr.py (200) - [INFO] : Loading bots from db...
langbot | [08-12 13:46:49.088] pipelinemgr.py (231) - [INFO] : Loading pipelines from db...
langbot | [08-12 13:46:49.090] kbmgr.py (162) - [INFO] : Loading knowledge bases from db...
langbot | [08-12 13:46:49.354] app.py (197) - [INFO] : =====
langbot | [08-12 13:46:49.354] app.py (197) - [INFO] : ➜ Access WebUI / 访问管理面板
langbot | [08-12 13:46:49.355] app.py (197) - [INFO] :
langbot | [08-12 13:46:49.355] app.py (197) - [INFO] : 🏠 Local Address: http://127.0.0.1:5300/
langbot | [08-12 13:46:49.356] app.py (197) - [INFO] : 🌐 Public Address: http://<Your Public IP>:5300/
langbot | [08-12 13:46:49.356] app.py (197) - [INFO] :
langbot | [08-12 13:46:49.357] app.py (197) - [INFO] : ⚡ Running this program in a container? Please ensure that the 5300 port is exposed
langbot | [08-12 13:46:49.357] app.py (197) - [INFO] : =====
langbot | [08-12 13:46:51.042] show_notes.py (32) - [INFO] : Current Version: v1.1.2
langbot | [2025-08-12 13:46:51 +0800] [10] [INFO] Running on http://0.0.0.0:5300 (CTRL + C to quit)
```

一開始會請使用者註冊帳號(下圖左)，註冊完之後，就可以使用這帳號登入系統(下圖右)，開始進行設定。



### 3.2. LangBot 連接 Discord 設定

參考 <https://docs.langbot.app/zh/deploy/platforms/discord.html>，使依照文件所指示的步驟建立 Discord Application、機器人，並且邀請至伺服器中。

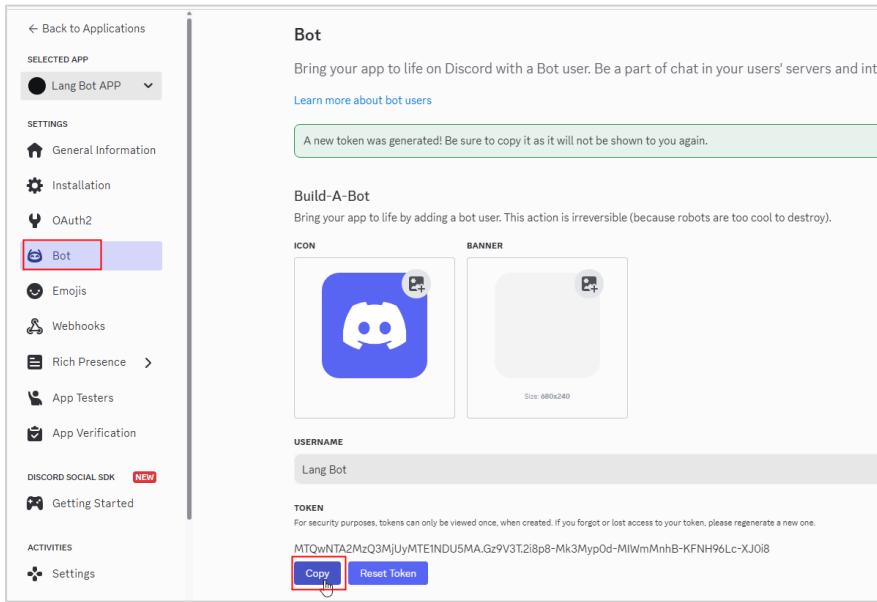
#### Discord :

進入建立的 application 中，選擇「OAuth2」，複製 CLIENT ID。

The screenshot shows the OAuth2 settings page for the 'Lang Bot APP'. On the left sidebar, under 'SETTINGS', the 'OAuth2' option is selected and highlighted with a red box. The main content area is titled 'OAuth2' and contains the following information:

- Client information**: Displays the CLIENT ID: 1405063472521154590, which is also highlighted with a red box, and a 'Copy' button below it.
- CLIENT SECRET**: Shows 'Hidden for security' and a 'Reset Secret' button.
- PUBLIC CLIENT**: A note stating 'Public clients cannot maintain the confidentiality of their client credentials (i.e. desktop/mobile)'.

選擇「Bot」，建立機器人，並且複製 TOKEN。



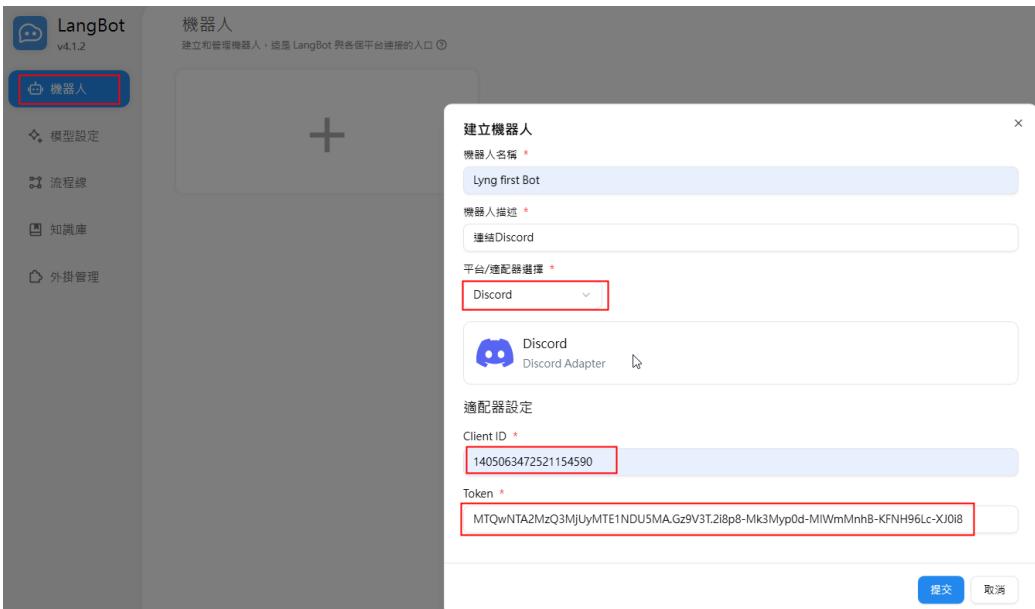
### LangBot:

使用剛剛註冊的帳密進入 LangBot 中，選擇機器人，新增一個機器人。

機器人名稱跟描述：可以自行提供。

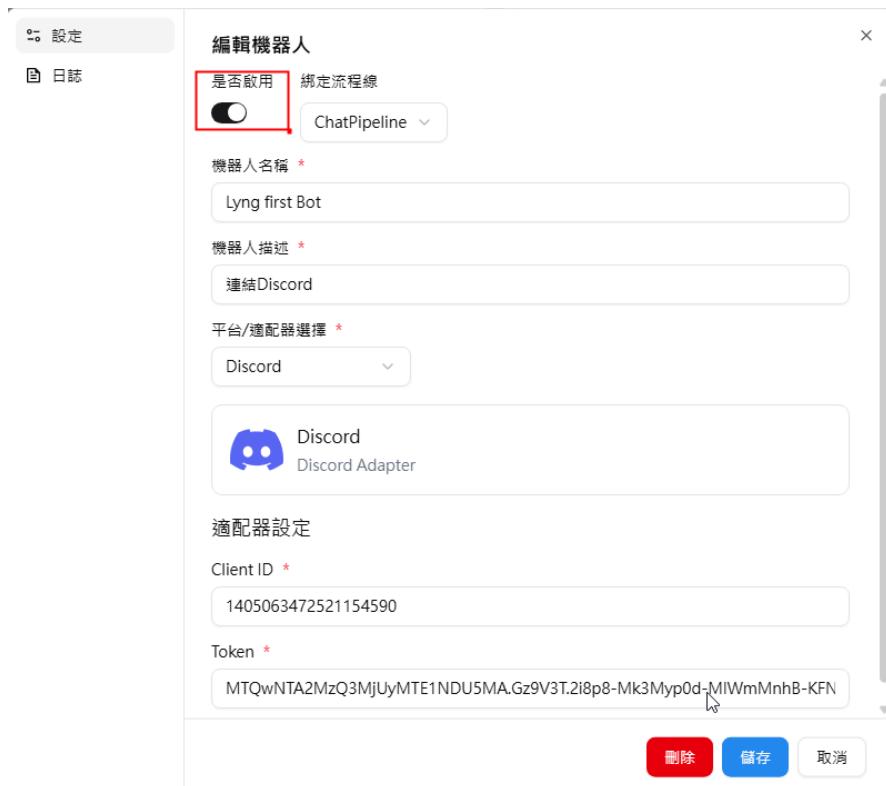
平台/適配器選擇：選擇 Discord

Client ID / Token：依照之前從 Discord 複製的 client id 與 token 資訊填入。



提交後，進入下面畫面，可以選擇啟用，然後儲存。啟用後，Discord 機器人即會上線。

流程線會先綁定預設的流程線 ChatPipeline，等之後有設定新的流程線，再進行修改。



也可以直接在外層選擇啟用或關閉機器人。

機器人

建立和管理機器人，這是 LangBot 與各個平台連接的入口 ②

+ Lyng first Bot

連結Discord

Discord

ChatPipeline

回到 Discord，可以看到我們剛剛串接的機器人已經上線。



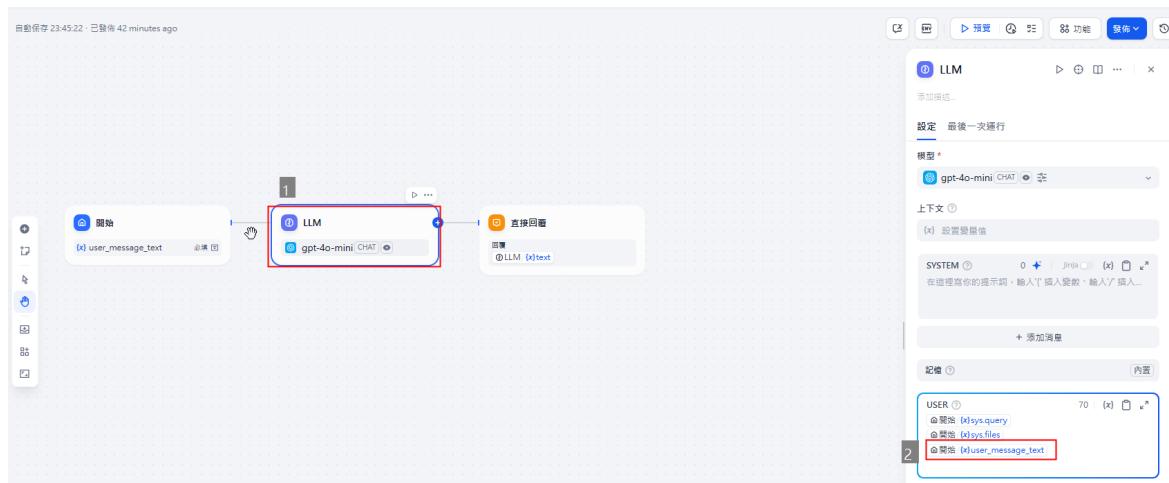
### 3.3. LangBot 連接 Dify 設定

參考下面兩個網站 <https://docs.langbot.app/zh/deploy/pipelines/dify.html> 與

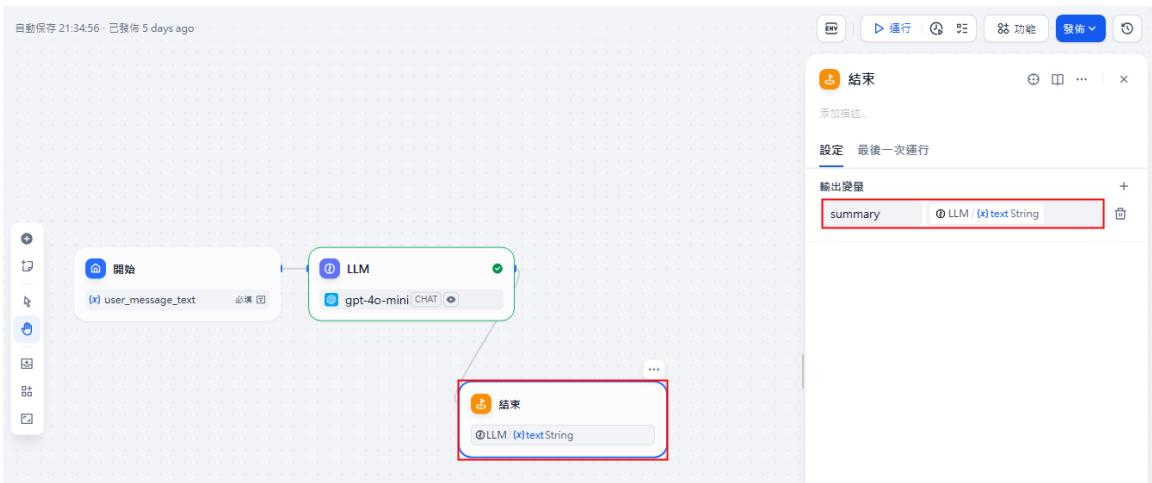
<https://docs.dify.ai/zh-hans/learn-more/use-cases/connect-dify-to-various-im-platforms-by-using-langbot>，關於 Dify 與 LangBot 的設定，如何連結 LangBot 與 Dify。

Dify-聊天流(ChatFlow)：

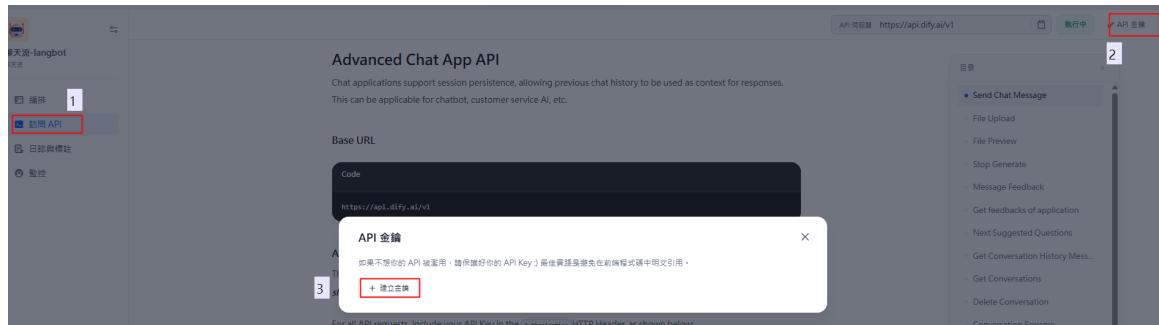
新增一個 chatflow，並新增變數 user\_message\_text 接收從 Discord 傳來的訊息，並且將 user\_message\_text 設定於 user prompt 中。



另外提醒，如果 Dify 使用的是工作流(workflow),需要在最後加上一個結束的節點，並且將 LLM 最後執行的結果放置 summary 變數中。

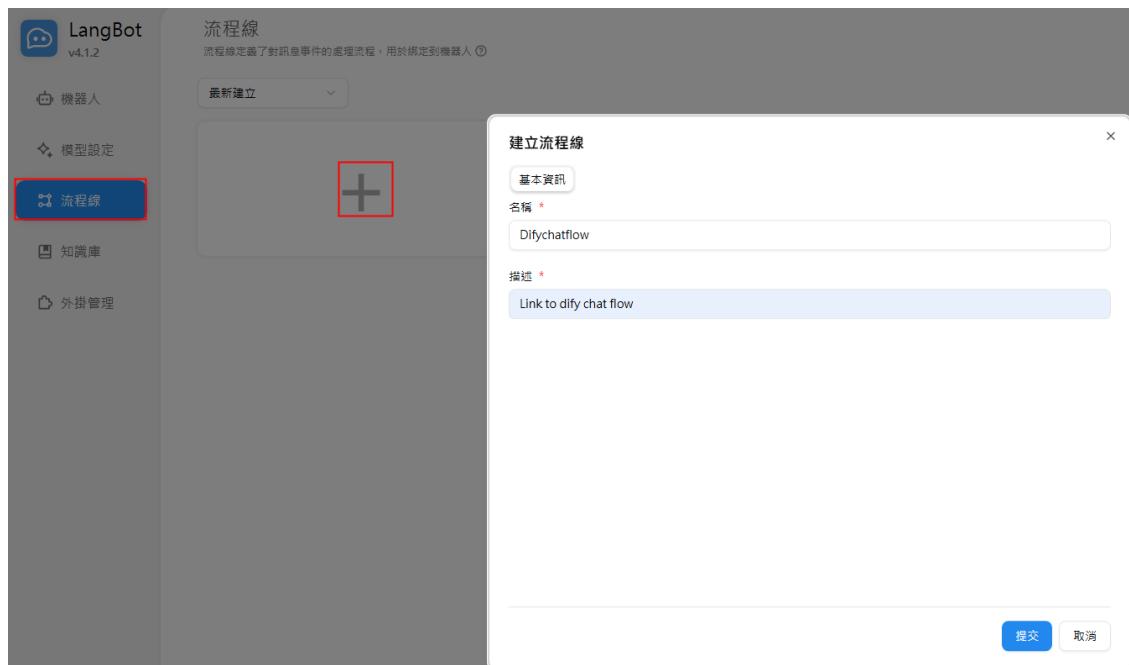


接下來，由下圖產生 API 金鑰，並將 API 金鑰複製。



**LangBot:**

連結 Dify 的設定，選擇「流程線」，建立一條新的流程線，自行輸入名稱與描述。



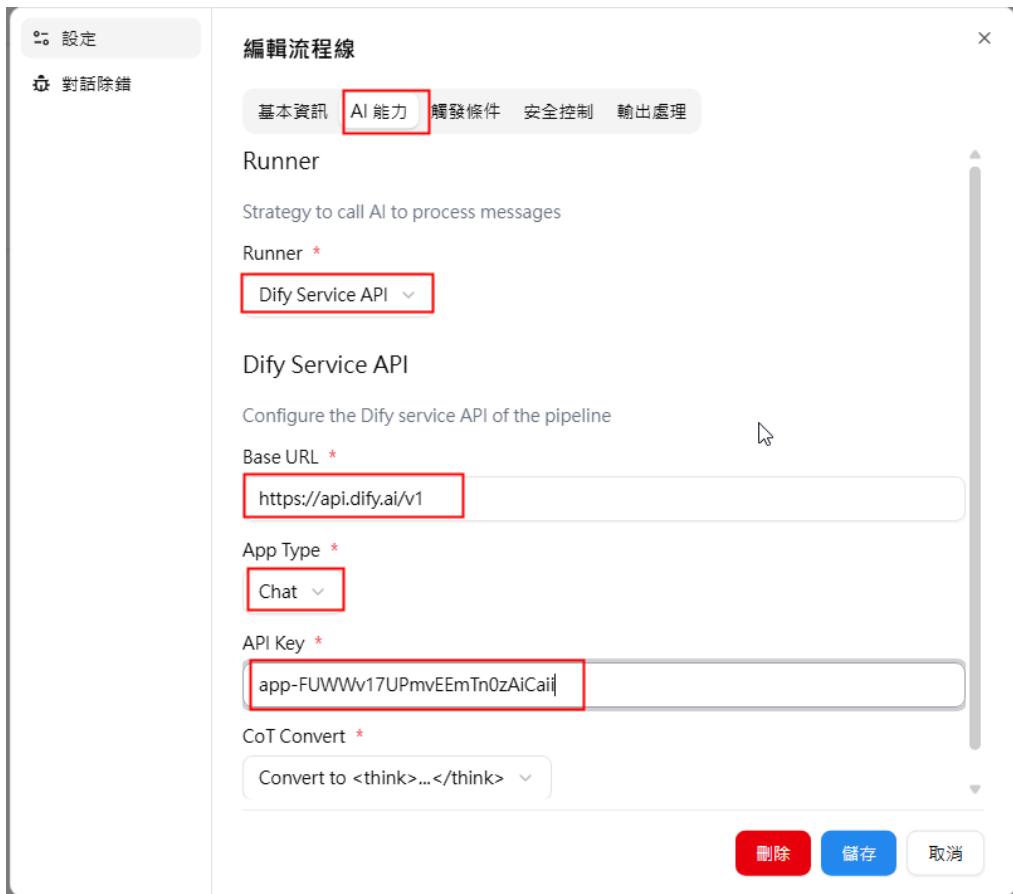
按「提交」後，如下圖，選擇 tab 「AI 能力」：

**Runner**：選 Dify Service API。

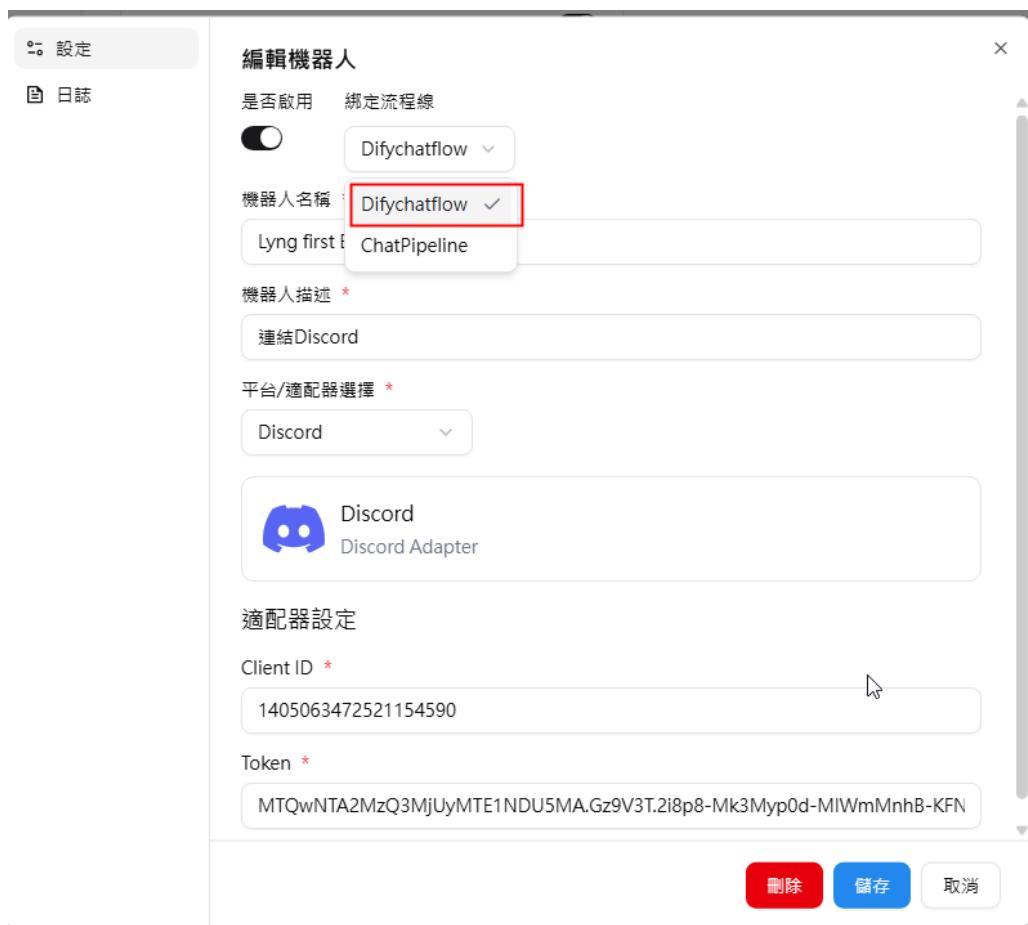
**Base URL**：輸入正確網址，如果使用雲端 Dify，不需要修改。

**App Type**：選 Chat /Work Flow/ Event

API Key：將剛剛從 Dify flow 產生的 API 金鑰，填入此處。然後儲存設定。



最後記得回到機器人，將「定流程線」修改成我們剛刷新建立的流程線。



### 3.4. 執行結果

在 Discord 提問，機器人透過連結 Dify 會自動回應。

lyng7133 下午3:56  
@Lang Bot 你今天好嗎?

Lang Bot 應用 下午3:56  
我今天很好，謝謝你！你今天怎麼樣？需要我幫忙什麼嗎？

lyng7133 下午3:56  
@Lang Bot 世界上最大的動物是什麼?

Lang Bot 應用 下午3:56  
世界上最大的動物是藍鯨 (Blue Whale)。藍鯨是地球上體型最大的動物，成年的藍鯨體長可達30米以上，體重可超過150噸。牠們主要以小型甲殼類動物—磷蝦為食，分布在全球的海洋中。你還想了解藍鯨的其他資訊嗎？

另外說明，流程線「觸發條件」中

**At**：如果打開，可以用@機器人，呼叫機器人回答。

**Prefix**：定義用來呼叫機器人回答的前綴詞，不能是空白。如下面第二個圖為例，用「ai 你喜歡下雨的天氣嗎？」ai 來啟動機器人。

