

Dify and Streamlit ChatBot

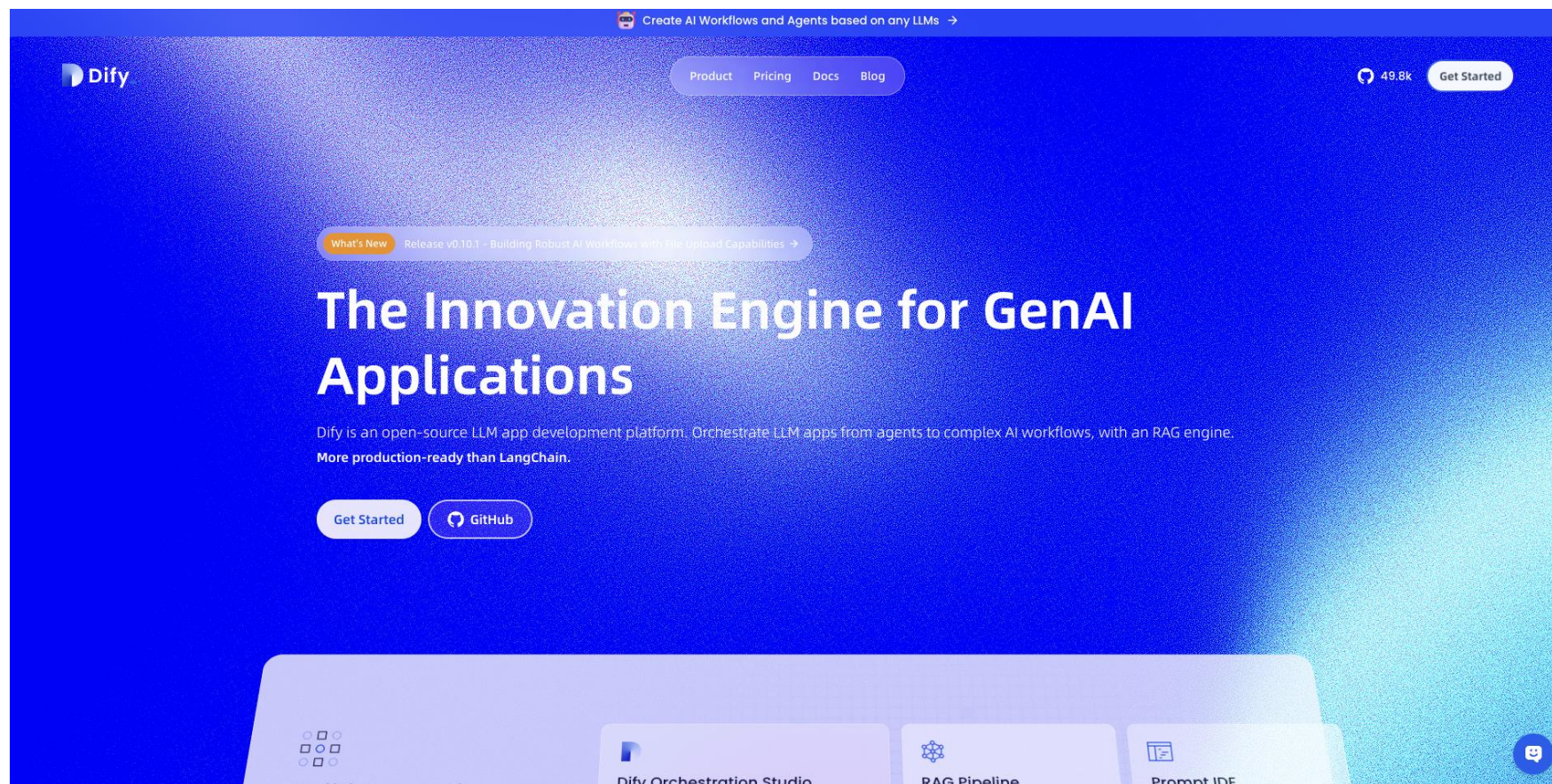
邱彦榕

zx12as3420@gmail.com

簡介

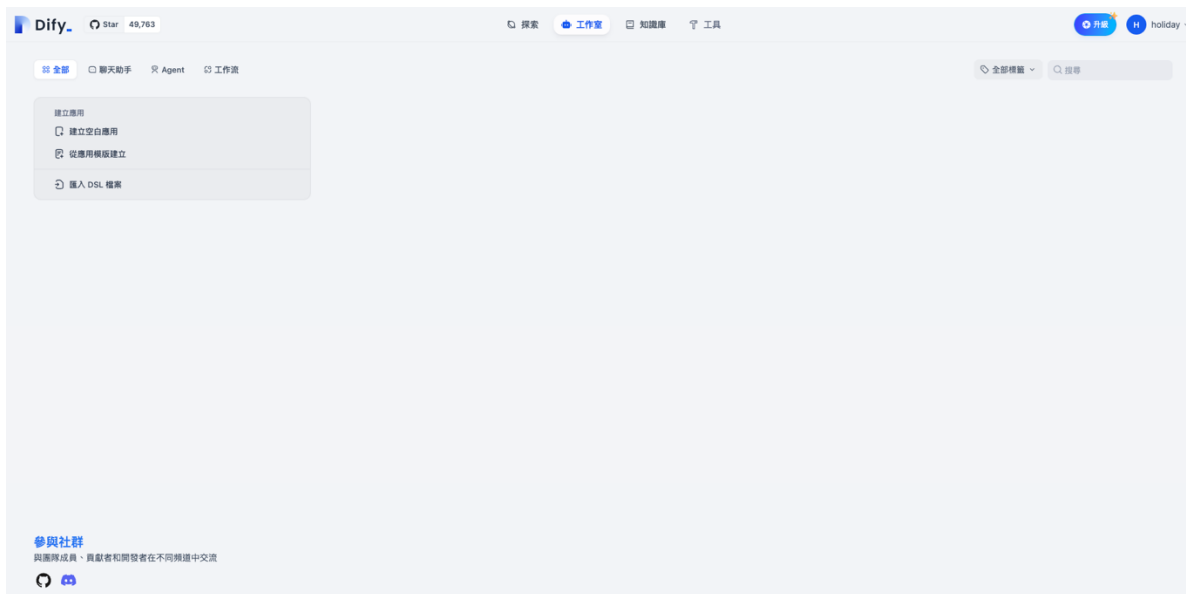
- Dify
- 程式講解
- Colab 建立聊天機器人

申請dify

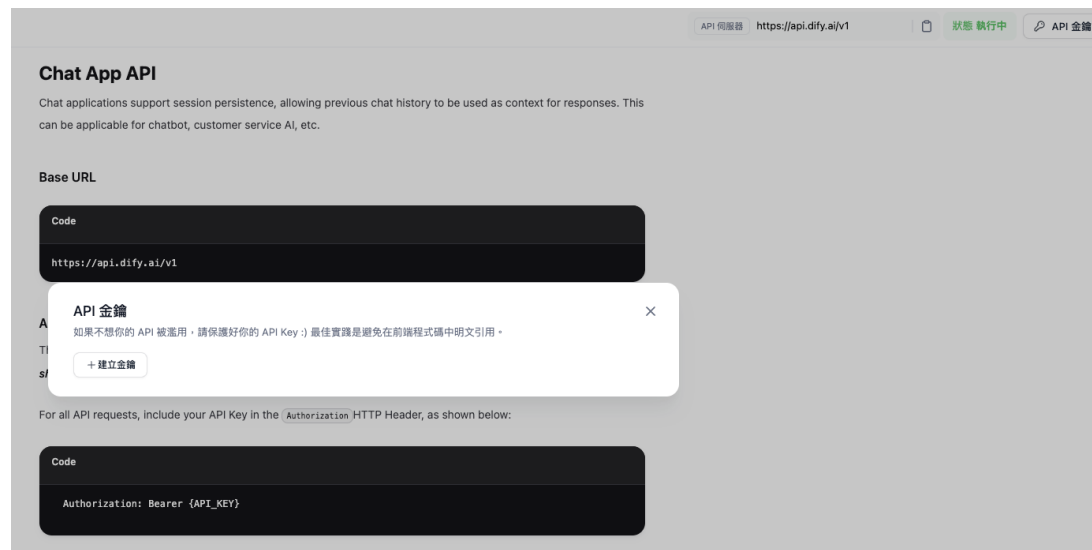


<https://github.com/langgenius/dify>
<https://dify.ai/>

登入後



方法一 直接建立金鑰



有兩個東西會用到

<https://api.dify.ai/v1>

&

API 金鑰 (記得複製存好)，想像成密碼就可以

方法二



試用已結束

在除錯之前需要設定 LLM 提供者的金鑰。

去設定 [↗](#)

用量沒了

模型供應商

顯示模型

ANTHROPIC

LLM

額度

0 Tokens

API-KEY

設定

顯示模型

+ 新增更多模型提供者



Azure OpenAI Service

+ 新增模型



Google's Gemini model.

LLM



Google Cloud

Vertex AI in Google Cloud Platform.

LLM

TEXT EMBEDDING

壓“去設定”

設定 Google

API Key *

AlzaSyAD-Cuoe-lABnZJ9SgyfYl

[Get your API Key from Google](#)

您的金鑰將僅

填入自己的google api，就可以使用

取得google API

- 查詢 Build with Gemini API

API keys


Cloud projects are subject to the [Google Cloud Platform Terms of Service](#), and use of Gemini API and Google AI Studio is subject to the [Gemini API Additional Terms of Service](#).

Remember to use API keys securely. Don't share or embed them in public code. Use of Gemini API from a billing-enabled project is subject to [pay-as-you-go pricing](#).

Quickly test the API by running a cURL command



[API quickstart guide](#)

```
curl \
-H 'Content-Type: application/json' \
-d '{"contents":[{"parts":[{"text":"Explain how AI works"}]}]' \
-X POST 'https://generativelanguage.googleapis.com/v1beta/models/gemini-1.5-flash-latest:generateContent?key=YOUR_API_KEY'
```

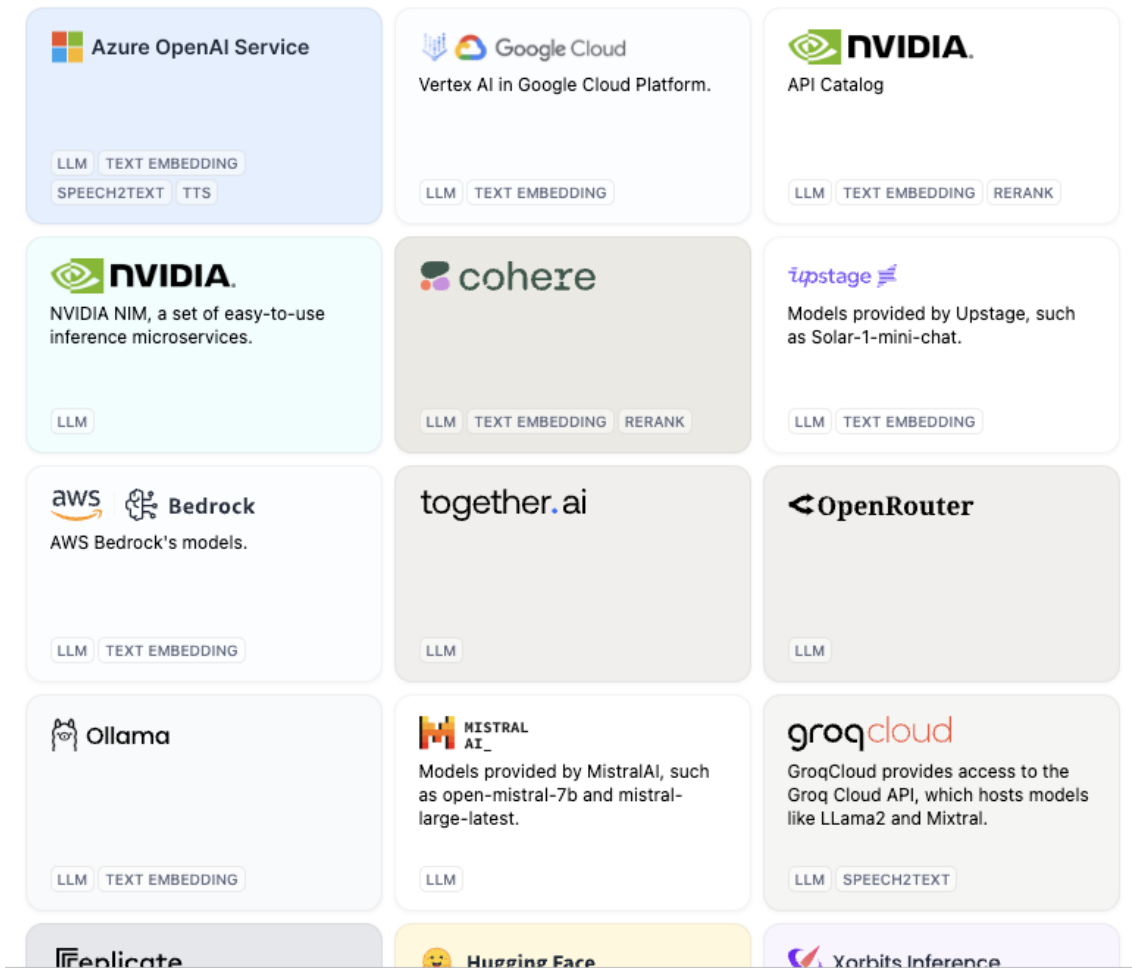
 Use code with caution.

[Create API key](#)

Your API keys are listed below. You can also view and manage your project and API keys in Google Cloud.

Project number	Project name	API key	Created	Plan
...4652	Generative Language Client 	...9S30	Jul 1, 2024	Free of charge Set up Billing View usage data 

串接各種LLM工具



更換內建模型，發佈



工具介紹

- Python
 - 程式語言
 - 套件：streamlit, request
- Colab
 - Google提供網路撰寫pytho語言的軟體

Python 免費學習網站

- 套件的官網
 - 如：[Numpy](#)
- [GeeksforGeeks](#)
- [W3school](#)
- Youtube

使用套件

- 在colab上安裝
 - !pip install requests
 - !pip install streamlit
- requests
 - Python應用HTTP操作的套件
- streamlit
 - Python 開發互動式網頁的套件

If...then...else

- If可以獨立存在，但else必須伴隨著if
- 透過條件判斷，使得在不同情況下，執行不同的操作



```
age = 18
if age > 18:
    print("大學生")
else:
    print("高中生")
```

載入套件與設定應用工具來源



```
import requests
import streamlit as st

dify_api_key = YOUR_KEY
url = "https://api.dify.ai/v1/chat-messages"
```

初始化與紀錄紀錄問答的過程



```
st.title("Dify Streamlit App")

if "conversation_id" not in st.session_state:
    st.session_state.conversation_id = ""

if "messages" not in st.session_state:
    st.session_state.messages = []
```

可以把它刪除看對話結果

把問答的結果顯示出來

問答是透過JSON的方法存放，取用方法參考程式碼

```
for message in st.session_state.messages:  
    with st.chat_message(message["role"]):  
        st.markdown(message["content"])
```

Streamlit顯示與後端儲存

Dify Streamlit App



hi



Hi there! How can I help you today?



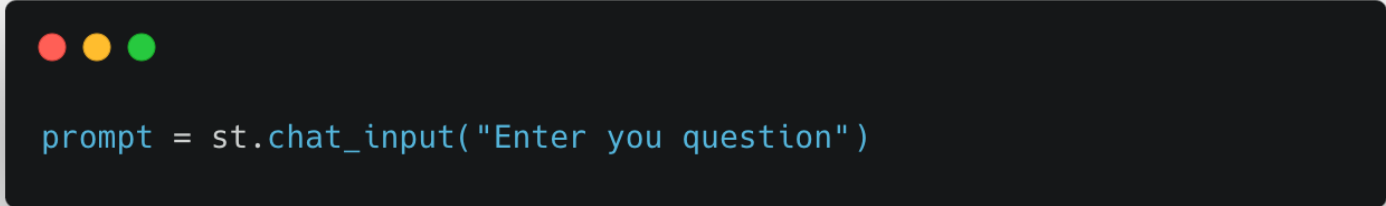
how are you



I'm doing well, thank you for asking! How are you? As a large language model, I don't experience emotions or feelings the way humans do, but I'm functioning optimally and ready to assist you.

```
{
  "messages": [
    {
      "role": "user",
      "content": "hi"
    },
    {
      "role": "assistant",
      "content": "Hi there! How can I help you today?\n"
    },
    {
      "role": "user",
      "content": "how are you"
    },
    {
      "role": "assistant",
      "content": "I'm doing well, thank you for asking! How are you? As a large language model, I don't experience emotions or feelings the way humans do, but I'm functioning optimally and ready to assist you.\n"
    }
  ],
  "conversation_id": "b7ab2afb-a04e-4f35-93fd-8c0edfdd2ac8"
}
```

建立一個輸入文字的功能



```
prompt = st.chat_input("Enter you question")
```

引號中可以替換想顯示的字

連線伺服器端，進行問答

```
if prompt:
    with st.chat_message("user"):
        st.markdown(prompt)
        st.session_state.messages.append({"role": "user", "content":
prompt})

    with st.chat_message("assistant"):
        message_placeholder = st.empty()

        headers = {
            'Authorization': f'Bearer {dify_api_key}',
            'Content-Type': 'application/json'
        }

        payload = {
            "inputs": {},
            "query": prompt,
            "response_mode": "blocking",
            "conversation_id": st.session_state.conversation_id,
            "user": "aianytime",
            "files": []
        }

        try:
            response = requests.post(url, headers=headers,
json=payload)
            response.raise_for_status()
            response_data = response.json()

            full_response = response_data.get('answer', '')
            new_conversation_id =
response_data.get('conversation_id',
st.session_state.conversation_id)
            st.session_state.conversation_id = new_conversation_id

        except requests.exceptions.RequestException as e:
            st.error(f"An error occurred: {e}")
            full_response = "An error occurred while fetching the
response."

        message_placeholder.markdown(full_response)

        st.session_state.messages.append({"role": "assistant",
"content": full_response})
```

if是當有問題傳入時，會執行以下的程式

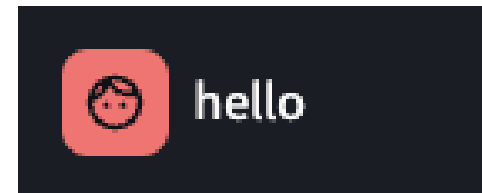
有兩個with

第一個用在顯示user提出的問題

第二個上傳伺服器，顯示回傳的答案

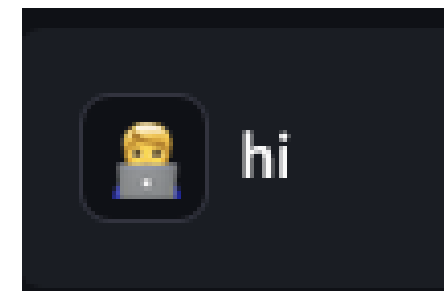
顯示使用者的對話資料

```
with st.chat_message("user"):
    st.markdown(prompt)
    st.session_state.messages.append({"role": "user", "content":
prompt})
```



human = user
修改avatar可以換造型

```
with st.chat_message("human", avatar="👤 "):
    st.markdown(prompt)
    st.session_state.messages.append({"role": "user", "content":
prompt})
```



設定Dify API 的使用

Request

POST · /chat-messages

```
curl -X POST 'https://api.dify.ai/v1/chat-messages' \
--header 'Authorization: Bearer {api_key}' \
--header 'Content-Type: application/json' \
--data-raw '{
  "inputs": {},
  "query": "What are the specs of the iPhone 13 Pro Max?",
  "response_mode": "streaming",
  "conversation_id": "",
  "user": "abc-123",
  "files": [
    {
      "type": "image",
      "transfer_method": "remote_url",
      "url": "https://cloud.dify.ai/logo/logo-site.png"
    }
  ]
}'
```

Dify 提供的API使用範例

```
headers = {
    'Authorization': f'Bearer {dify_api_key}',
    'Content-Type': 'application/json'
}

payload = {
    "inputs": {},
    "query": prompt,
    "response_mode": "blocking",
    "conversation_id": st.session_state.conversation_id,
    "user": "yjc",
    "files": []
}
```

headers對應到左圖的header
payload對應到左圖的data-raw

response_mode

The mode of response return, supporting:

- `streaming` Streaming mode (recommended), implements a typewriter-like output through SSE (**Server-Sent Events**).
- `blocking` Blocking mode, returns result after execution is complete.
(Requests may be interrupted if the process is long) Due to Cloudflare restrictions, the request will be interrupted without a return after 100 seconds. *Note: blocking mode is not supported in Agent Assistant mode*

conversation_id

`conversation_id` string

Conversation ID, to continue the conversation based on previous chat records, it is necessary to pass the previous message's conversation_id.

files

File list, suitable for inputting files (images) combined with text understanding and answering questions, available only when the model supports Vision capability.

- `type` (string) Supported type: `image` (currently only supports image type)
- `transfer_method` (string) Transfer method, `remote_url` for image URL / `local_file` for file upload
- `url` (string) Image URL (when the transfer method is `remote_url`)
- `upload_file_id` (string) Uploaded file ID, which must be obtained by uploading through the File Upload API in advance (when the transfer method is `local_file`)

```
"files": [  
  {  
    "type": "image",  
    "transfer_method": "remote_url",  
    "url": "https://cloud.dify.ai/logo/logo-site.png"  
  }  
]
```

try...except...



```
try:
    # 嘗試執行的程式碼
    ...
except ExceptionType:
    # 當出現指定的 ExceptionType 錯誤時，執行這段程式碼
    print("An error occurred:", ExceptionType)
```

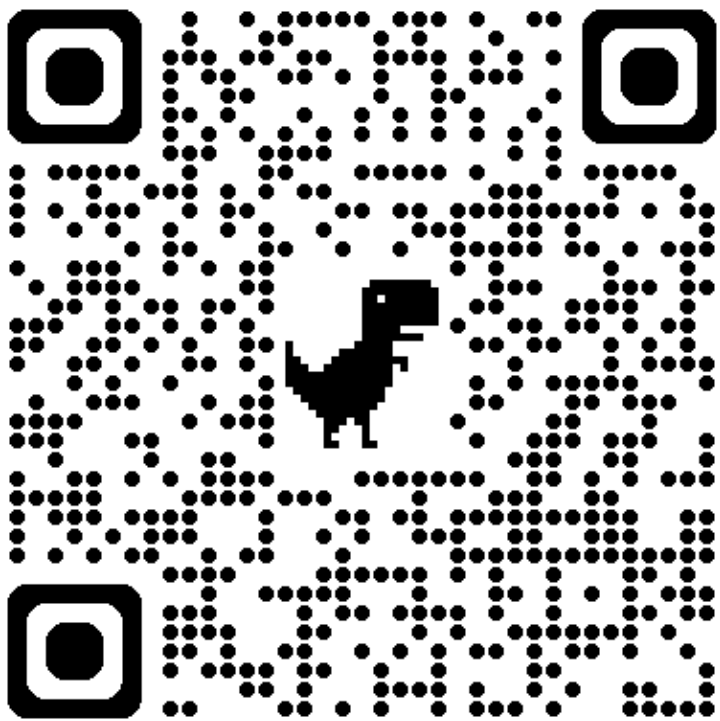
執行問答模型

```
try:
    response = requests.post(url, headers=headers, json=payload)
    response.raise_for_status()
    response_data = response.json()

    full_response = response_data.get('answer', '')
    new_conversation_id = response_data.get('conversation_id', st.session_state.conversation_id)
    st.session_state.conversation_id = new_conversation_id

except requests.exceptions.RequestException as e:
    st.error(f"An error occurred: {e}")
    full_response = "An error occurred while fetching the response."
```

取得完整程式



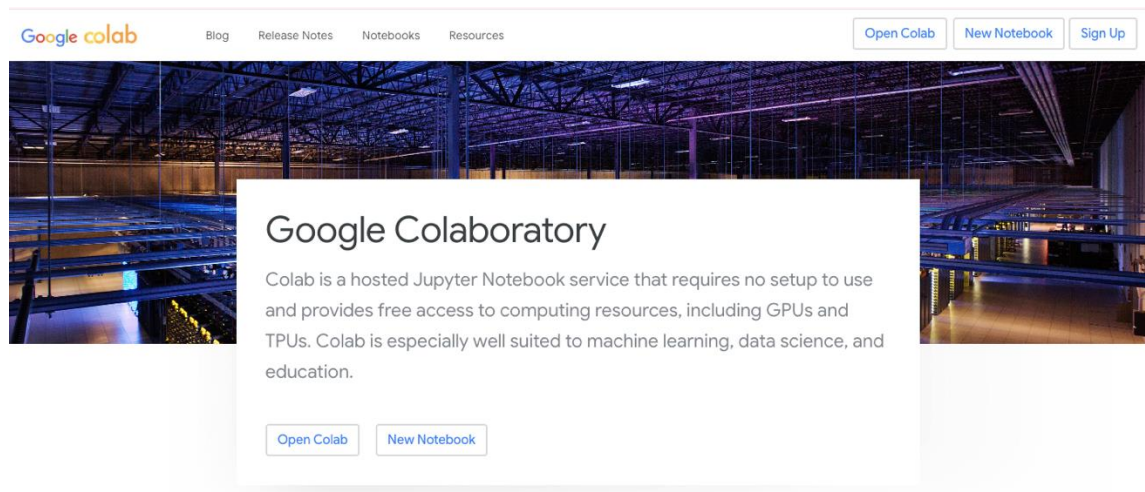
```
import requests
import streamlit as st

dify_api_key = YOUR_API

url = "https://api.dify.ai/v1/chat-messages"
```

把YOUR_API取代成自己的

Colab



開啟筆記本

範例 >

最近 >

Google 雲端硬碟 >

GitHub >

上傳 >

搜尋筆記本

🔍

🔄

🗑️

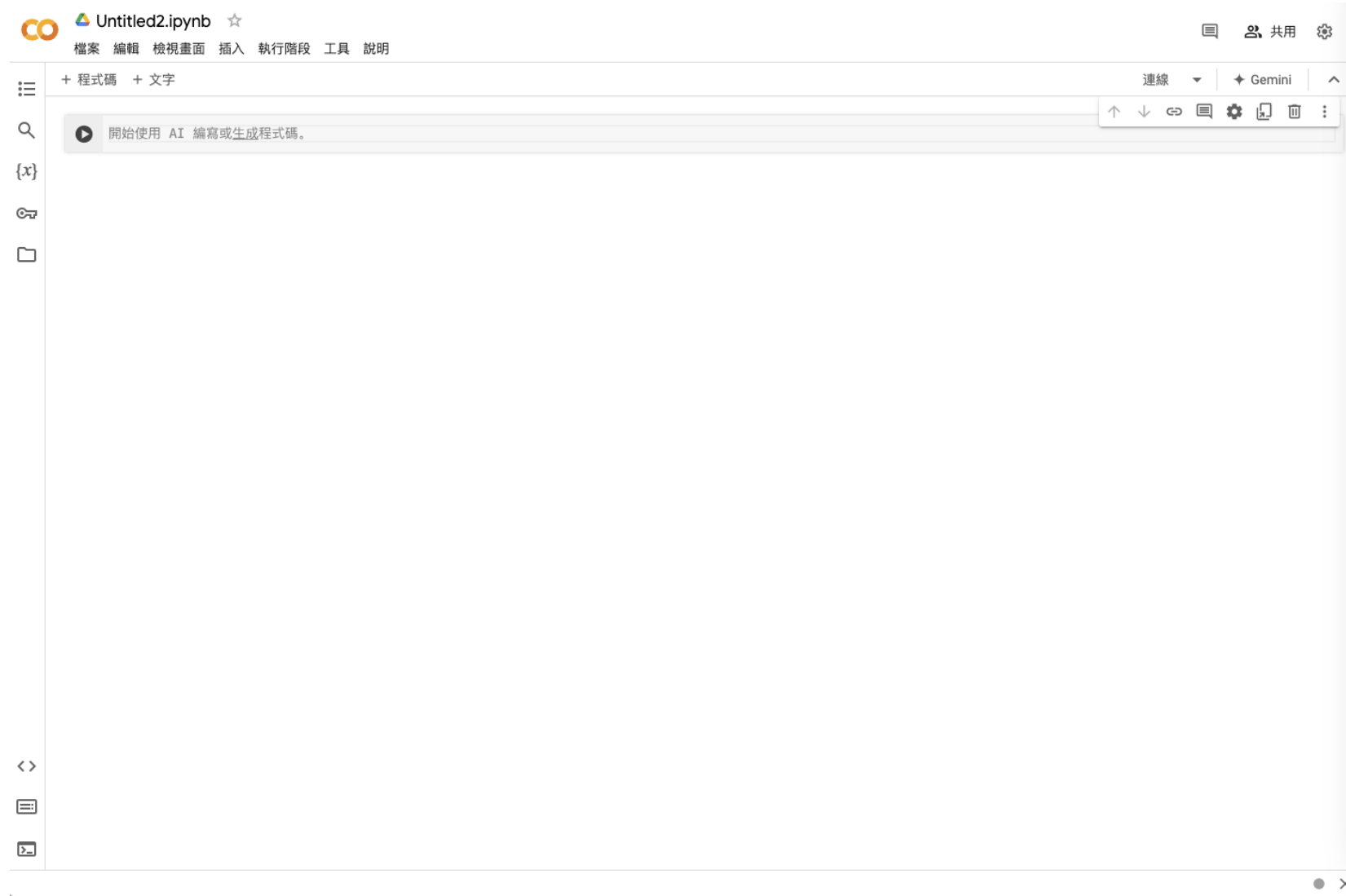
標題	上次開啟時間 ↓	初次開啟 ↑	
dify_streamlit.ipynb	10月28日	10月28日	
Untitled2.ipynb	10月28日	10月28日	
Icanhazip.Ipv4 Download	10月28日	10月28日	
歡迎使用 Colaboratory	10月28日	2023年3月15日	
Al.ipynb	5月23日	2023年5月21日	
Streamlit.ipynb	5月23日	2023年5月21日	
0523上課.inynb	5月23日	2023年5月23日	

+ 新增筆記本

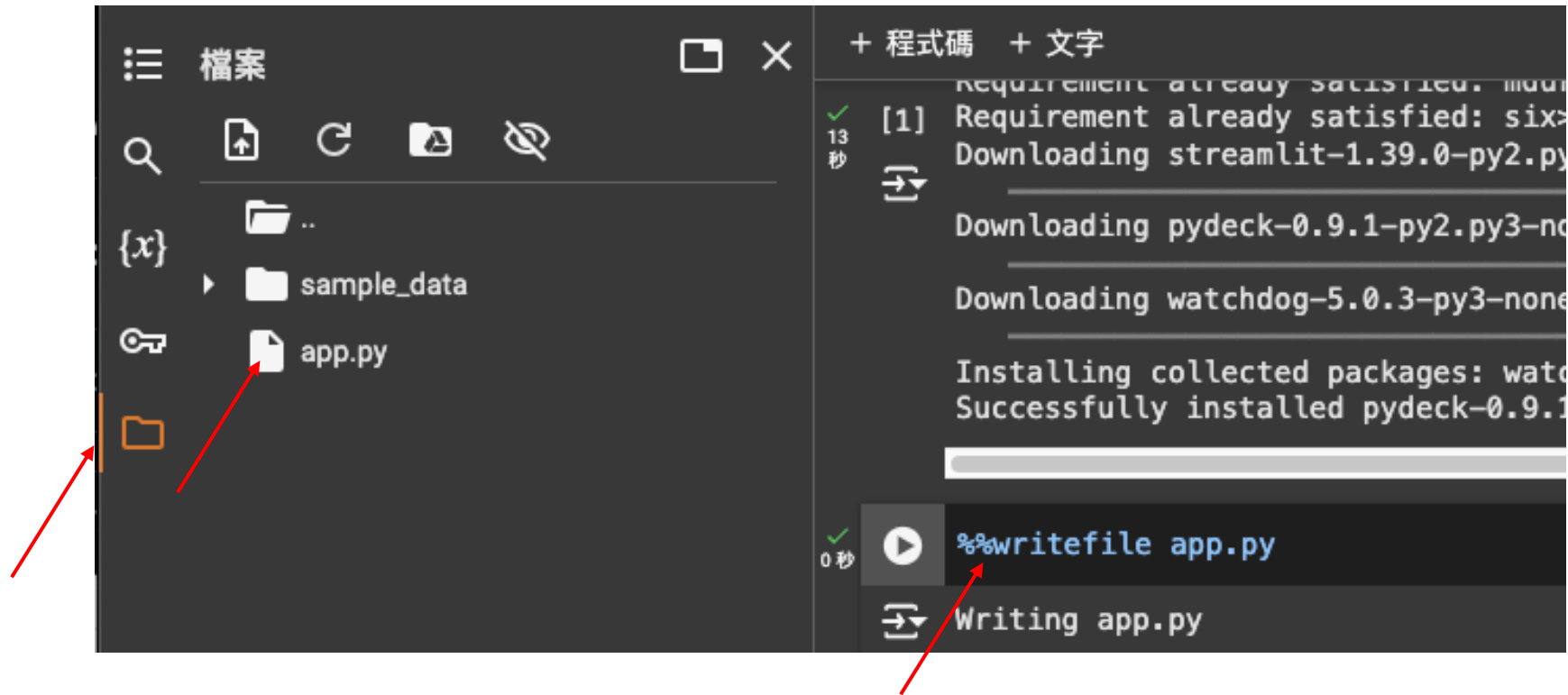
取消

<https://colab.google/>

Colab notebook



將github的app.py內容複製到這個app.py中



你的伺服器IP和密碼

```
[ ] !curl https://loca.lt/mytunnelpassword
```

```
⇒ 35.197.124.107
```

```
! streamlit run app.py & npx localtunnel --port 8501
...
Collecting usage statistics. To deactivate, set browser.gat

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://172.28.0.12:8501
External URL: http://34.81.110.213:8501

Need to install the following packages:
  localtunnel@2.0.2
Ok to proceed? (y) 
```

兩次y

```
Need to install the following packages:
  localtunnel@2.0.2
Ok to proceed? (y) y
your url is: https://petite-snails-rush.loca.lt
y

```

打開後，把剛剛那個密碼填入就可以

To access the website, please enter the tunnel password below.

If you don't know what it is, please ask whoever you got this link from.

Tunnel Password:

[Click to Submit](#)

Dify Streamlit App



hi



Hi there! How can I help you today?

Enter you question

成功~~~