market模塊 - 市集資料處理與訊息回覆內容：

匯入爬蟲取得的市集CSV資訊，存成pandas dataframe (df)

將檔案中的日期欄位轉成datetme物件

def 日期搜尋(df, 開始日期, 結束日期):

date\_list = []

for i in range(df的長度):

if df中第i列的日期有在開始日期和結束日期之間:

date\_list.append(i)

return date\_list

def 地點搜尋(df, 地點):

location\_list = []

for i in range(df的長度):

if df中第i列的地點符合要求的地點:

location\_list.append(i)

return location\_list

def 訊息回覆內容(df, 使用者的訊息):

# 判斷是否搜尋地點

if 訊息長度 == 3:

location\_list = 地點搜尋(df, 地點)

# 判斷是否搜尋特定日期

elif 使用者訊息符合特定日期搜尋格式:

date\_list =日期搜尋(df, 開始日期, 結束日期)

# 判斷是否搜尋地點與特定日期

elif使用者訊息符合地點和特定日期搜尋格式:

location\_list = 地點搜尋(df, 地點)

date\_list =日期搜尋(df, 開始日期, 結束日期)

# 其他為處理特定月份的訊息

else:

開始日期 = 此月份的第一天

if 月份為12月:

結束日期 = 明年的1/1減去一天

else:

結束日期 = 下個月的第一天減去一天

date\_list =日期搜尋(df, 開始日期, 結束日期)

# 將結果存成list

result\_list = []

if 沒有location\_list:

result\_list = date\_list

elif 沒有date\_list:

result\_list = location\_list

else:

result\_list為date\_list和location\_list的交集

# 避免訊息太長傳不出去

answer = []

用result\_list從df中用index提取內容，每10個結果為一個字串存到answer裡

return answer

discord api模塊：

import discord

# 避免token被別人更改利用，token存在.env檔

from dotenv import load\_dotenv

匯入.env檔

discord\_token = .env檔裡存的"DISCORD\_TOKEN"

import 寫好的market模塊

class 客戶(discord.Client):

# discord bot 上線時

async def on\_ready(self):

print("成功登入，目前身分為：discord bot名稱)

# 傳訊息

async def on\_message(self, message):

if 上一個傳訊息的人是discord bot:

print(self.user)

return

command, user\_message = None, None

try:

# openai api (目前沒有付費無法串接)

for text in ["Hi ", "hi "]:

if message.content.startswith(text):

command = message.content.split(" ")[0]

user\_message = message.content.replace(text, "")

# print(command, user\_message)

# 判斷是否有Hi字樣，以及將Hi和後面的字串內容切割開

if command in ["Hi", "hi"]:

await message.channel.send("抱歉，目前無法使用openai\n若要使用查詢市集功能，請輸入查詢+空格")

# openai api 聊天功能

# bot\_response = chatgpt\_response(prompt=user\_message)

# await message.channel.send(f"Answer: {bot\_response}")

# 判斷是否有查詢字樣，以及將查詢和後面的字串內容切割開

for text in ["查詢 "]:

if message.content.startswith(text):

command = message.content.split(" ")[0]

user\_message = message.content.replace(text, "")

if command in ["查詢"]:

mkt\_response = market\_new.market\_response(df, prompt=user\_message)

for i in mkt\_response:

await message.channel.send(f"{i}")

# 非"Hi", "hi"(呼叫openai)及"查詢"(查詢市集)

if command not in ["Hi", "hi", "查詢"]:

await message.channel.send(f"""{message.author.name}您好！

歡迎使用市集機器人

1. 呼叫openai請說Hi+空格

2. 查詢市集

(1)縣市名稱，格式為 查詢 縣市名稱 （例如：查詢 台北市）

(2)查詢特定月份的所有市集，格式為 查詢 YYYY/MM ( 例如：查詢 2024/06）

(3)查詢特定日期，格式為 查詢 YYYY/MM/DD-DD ( 例如：查詢 2024/06/01-05）

(4)同時查詢縣市和日期，格式為 查詢 縣市名稱 YYYY/MM/DD-DD ( 例如：查詢 台北市 2024/06/01-05）

\*請記得開頭要加上查詢""")

except Exception as e:

print(e)

await message.channel.send(f"""請重新確認查詢的格式

查詢市集

(1)縣市名稱，格式為 查詢 縣市名稱 （例如：查詢 台北市）

(2)查詢特定月份的所有市集，格式為 查詢 YYYY/MM ( 例如：查詢 2024/06）

(3)查詢特定日期，格式為 查詢 YYYY/MM/DD-DD ( 例如：查詢 2024/06/01-05）

(4)同時查詢縣市和日期，格式為 查詢 縣市名稱 YYYY/MM/DD-DD ( 例如：查詢 台北市 2024/06/01-05）

\*請記得開頭要加上查詢""")

intents = discord.Intents.default()

intents.message\_content = True

client = My\_Client(intents=intents)

主程式運行：

從discord\_api模塊 import client, Discord token

if 當前模塊為主程式

運行discord的client，用Discord token驗證