25.4 使用 Open AI 製作聊天機器人

25.4.1 ChatGPT 聊天機器人初體驗

如下是使用 OpenAI 製作 ChatGPT 聊天機器人的步驟，完整實作可參閱 ChatGPT01 範例。

Step 01　前往 OpenAI 申請金鑰 (Key)，步驟可參考 25.3 節。

Step 02　在專案安裝 OpenAI 套件。

Step 03　使用 OpenAI API 物件進行對話操作，並取得聊天機器人的回應 (ChatResut)，寫法如下：

```
// 使用 OpenAI 金鑰建立 OpenAIAPI 物件
OpenAIAPI oenai 物件 = new OpenAIAPI("OpenAI API 金鑰");

// 透過 OpenAIAPI 物件執行對話操作並取得回應結果 CharResult 物件
ChatResult chatResult =
    await oenai 物件.Chat.CreateChatCompletionAsync(new ChatRequest()
    {
        Model = Model.ChatGPTTurbo,                    //使用模型
        Messages = new ChatMessage[]                   //指定聊天訊息
        {
            new ChatMessage(ChatMessageRole.User, prompt)   //聊天提示
        }
    });
```

Step 04　ChatResult 物件包含 OpenAI 模型回應的相關資訊，如生成的文本、回應的信度…等。ChatResult 物件的 Choices[0].Message.Content 屬性可取得生成的文本。

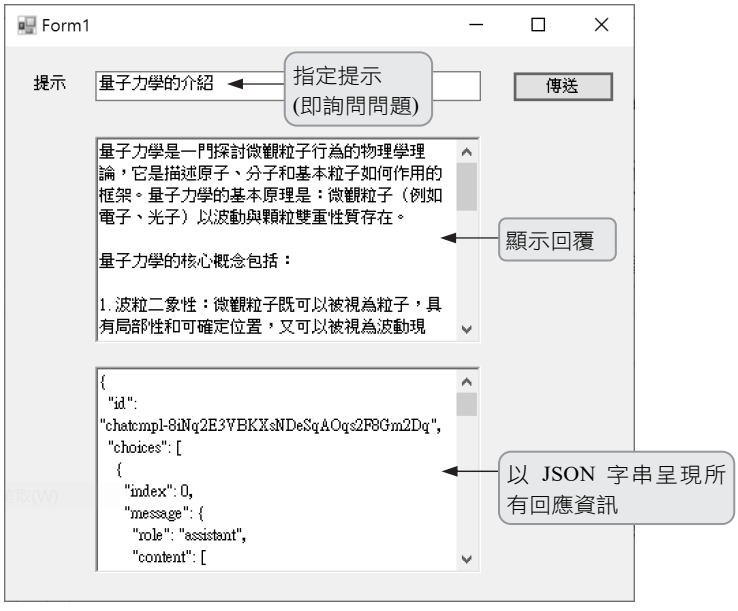
```
// 將回傳結果顯示在 richTextBox1 中
var reply = chatResult.Choices[0].Message.Content;
richTextBox1.Text = reply;
```

接著以 ChatGPT01 範例來練習製作聊天機器人。

範例：ChatGPT01.sln

使用 OpenAI 的 API 金鑰製作簡易的 ChatGPT 聊天機器人。當使用者指定提示(即詢問問題)後按下 傳送 鈕即將提示傳送給 OpenAI API。接著 OpenAI API 即會將回覆結果顯示於上方的多行文字方塊；而所有回應資訊會以 JSON 格式顯示在下方的多行文字方塊內。

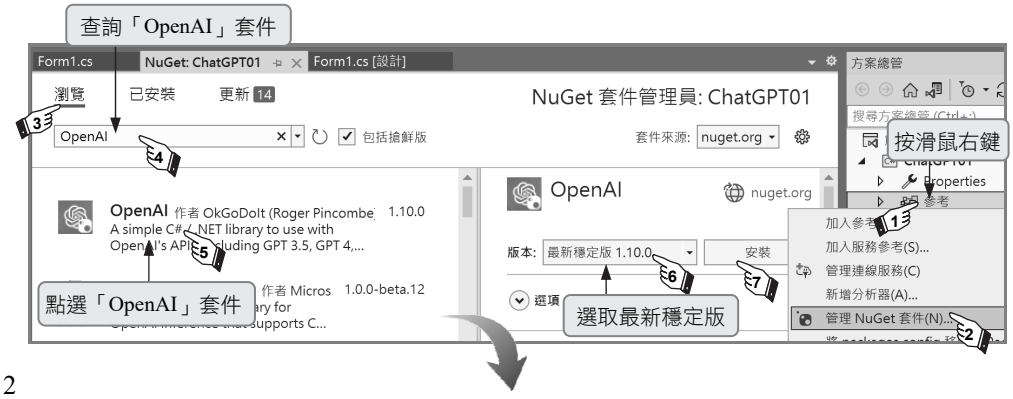
執行結果

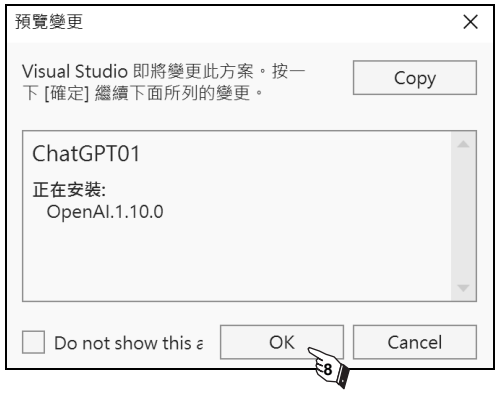


操作步驟

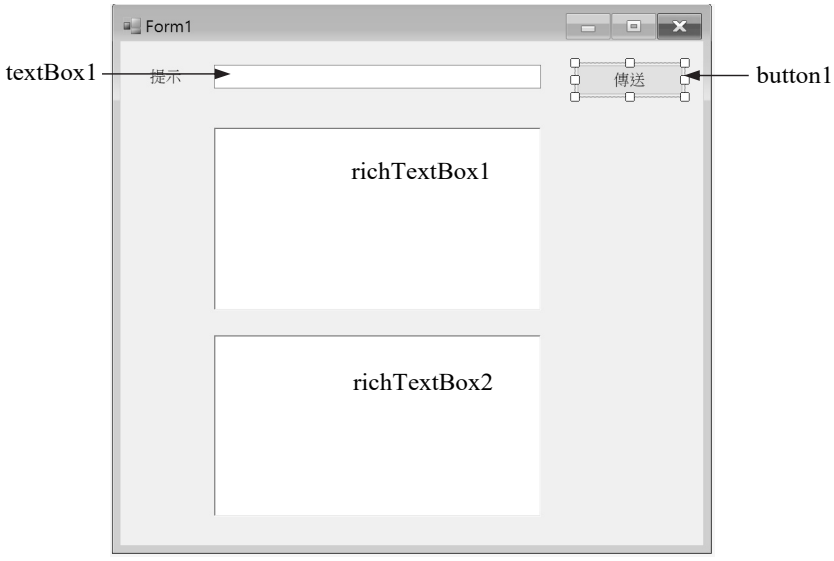
Step 01 前往 OpenAI 申請金鑰 (Key)，步驟可參考 25.3 節。

Step 02 安裝 OpenAI 套件。在方案總管視窗的「參考」按滑鼠右鍵執行【管理 NuGet 套件(N)】，接著依圖示操作安裝「OpenAI」套件。





Step 03　建立表單輸出入介面



Step 04　撰寫程式碼

程式碼　FileName:Form1.cs

```
01 using OpenAI_API;
02 using OpenAI_API.Chat;
03 using OpenAI_API.Models;
04 using Newtonsoft.Json;
05 using Newtonsoft.Json.Linq;
06
07 namespace ChatGPT01
08 {
```

```
09    public partial class Form1 : Form
10    {
11        public Form1()
12        {
13            InitializeComponent();
14        }
15
16        private async void button1_Click(object sender, EventArgs e)
17        {
18            // 從 textBox1 中取得使用者輸入的提示
19            string prompt = textBox1.Text;
20
21            // 判斷提示是否為空，若是則顯示「提示不能空白」並離開事件
22            if (string.IsNullOrEmpty(prompt))
23            {
24                MessageBox.Show("提示不能空白");
25                return;
26            }
27
28            try
29            {
30                //  使用 OpenAI API 金鑰建立 OpenAIAPI 物件 api
31                OpenAIAPI api = new OpenAIAPI("OpenAI API 金鑰");
32
33          // 透過 OpenAIAPI 物件執行對話(Chat)操作並取得回應結果 CharResult 物件
34                ChatResult chatResult = await
                      api.Chat.CreateChatCompletionAsync(new ChatRequest()
35                {
36                    Model = Model.ChatGPTTurbo,           //使用模型
37                    Messages = new ChatMessage[]         //指定聊天訊息
38                    {   //指定聊天提示
39                        new ChatMessage(ChatMessageRole.User, prompt)
40                    }
41                });
42
43                // 將回傳結果顯示在 richTextBox1 中
44                var reply = chatResult.Choices[0].Message.Content;
45                richTextBox1.Text = reply;
```

```
46
47      //將回傳結果序列化為 JSON 字串，同時將 JSON 字串格化顯示在 richTextBox2 中
48                  string jsonStr=JsonConvert.SerializeObject(chatResult);
49                  richTextBox2.Text = JObject.Parse(jsonStr).ToString();
50              }
51              catch (Exception ex)
52              {
53                  // 捕捉例外情況，顯示錯誤訊息
54                  MessageBox.Show($"發生錯誤：{ex.Message}");
55              }
56          }
57      }
58  }
```

說明

1. 第 1-5 行：引用相關命名空間。
2. 第 31 行：建立 OpenAIAPI 類別物件 api，同時指定金鑰。關於金鑰請讀者自行申請。
3. 第 48-49 行：將回應結果 CharResult 物件轉成 JSON 字串並顯示於 richTextBox2，下列使用 JSON 說明 CharResult 類別物件對應的屬性。

```
{
  "id": "chatcmpl-8imk3HM3ieW7K3HkyYUf1L43fxqHR",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": [
          {
            "type": "text",
            "text": "量子力學是一門研究微觀世界的物理學科，它揭示了原子、分子、基本粒子等微觀粒子的行為和性質。.....",
            "image_url": null
          }
        ],
        "name": null
      },
```

回覆內容可由 chatResult.Choices[0].Message.Content 取得

```
      "finish_reason": "stop",
      "delta": null
    }
  ],
  "usage": {
    "completion_tokens": 736,
    "prompt_tokens": 16,
    "total_tokens": 752
  },
  "created": 1705684487,
  "model": {
    "id": "gpt-3.5-turbo-0613",
    "owned_by": null,
    "object": null,
    "created": null,
    "permission": [],
    "root": null,
    "parent": null
  },
  "object": "chat.completion"
}
```


25.4.2 常用聊天參數

OpenAIAPI 物件執行 Chat.CreateChatCompletionAsync() 可指定 ChatResult 聊天結果物件 chatResult，此物件提供 Message、Temperature、PresencePenalty、FrequencyPenalty 以及 MaxTokens 參數，讓生成文本更加豐富。寫法如下：

```
// 透過 OpenAIAPI 物件執行對話(Chat)操作並取得回應結果 CharResult 物件
ChatResult chatResult = await
    oenai 物件.Chat.CreateChatCompletionAsync(new ChatRequest()
 {
    Model = Model.ChatGPTTurbo,       //使用模型
    Messages = new ChatMessage[]      //指定聊天訊息
    {
      new ChatMessage(ChatMessageRole.System, "你是網路行銷小編"),
      new ChatMessage(ChatMessageRole.User, "幫我寫人中之龍 IG 貼文")
    },
```

```
    Temperature =1.0,
    PresencePenalty = 0,
    FrequencyPenalty = 0,
    MaxTokens = 600,
});
```

1. Model：主要輸入參數，指定使用的模型。不同的模型具有不同的能力和特性，可根據應用場景和需求來選擇適合的模型。
2. Message：主要輸入參數，用於指定 ChatMessage 物件，每個物件代表一種角色，分別為「System」、「User」或「Assistant」；System 可指定 ChatGPT 的角色身份；User 代表使用者可進行提示(詢問問題)；Assistant 是 ChatGPT 回覆的結果，用來做回答的歷史紀錄，讓 ChatGPT 進行連續對話。
3. Temperature：預設值為 1，指定介於 0~2 之間的浮點數資料。用來指定生成文字的多樣性。值越小，生成結果越保守，但也可能較為狹隘。值越大，生成的結果越多樣，但可能會失去一些邏輯性。
4. PresencePenalty：預設值為 0，指定介於-2.0~2.0 之間的浮點數資料。用來指定對先前生成的文本中已經出現過的詞語進行懲罰力度，即鼓勵使用未出現詞語以增加生成的多樣性。例如：生成文本有關食物內容頻繁出現為米、雞肉、豬肉等。若指定 PresencePenalty 大於 0，例如指定 0.6 或 0.9，則下次生成食物會傾向選擇不重複詞彙，例如麵、牛肉、羊肉等，進而增加生成文本的多樣性。
5. FrequencyPenalty：預設值為 0，指定介於-2.0~2.0 之間的浮點數資料。用來指定對高頻率詞語出現機率進行懲罰力度，即鼓勵使用低頻率詞語以增加生成的多樣性。例如：生成文本有關食物內容常見為米、雞肉、豬肉等。若指定 FrequencyPenalty 大於 0，例如 1.3 或 1.6，則生成食物可能會出現較不常見的詞彙，例如鮭魚奶油飯、鴨肉羹等，增加生成文本的多樣性。
6. MaxTokens：指定生成文本內容最大 token 數量，可防止模型生成過於冗長或不適合的回應。

範例：ChatGPT02.sln

製作網路行銷聊天機器人小編， 透過提示與調整 MaxTokens、Temperature、PresencePenalty、FrequencyPenalty 參數，生成需要的行銷貼文與行銷策略。