

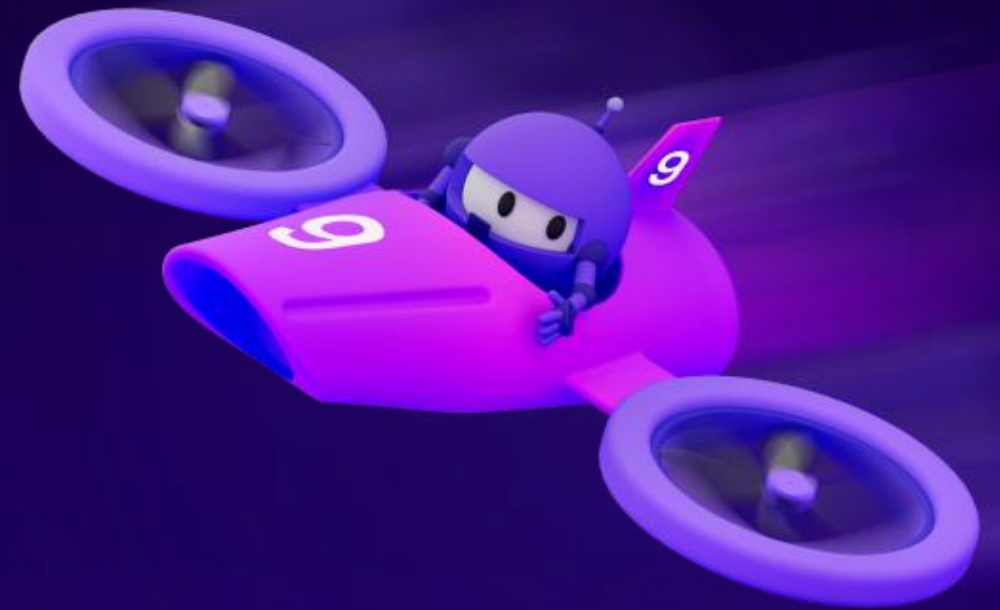


STUDY4

為 學 習 而 生

加入一點創意讓 ESG 更為輕鬆

James Fu



James Fu (傅士哲)

Data Platform MVP (2012~2022)

MCT (巨匠電腦-資料庫講師)

威煦軟體開發有限公司-技術總監

點部落：五餅二魚工作室

<https://www.dotblogs.com.tw/JamesFu>



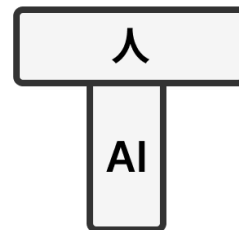
James Fu

一型



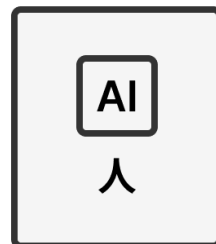
了解 AI 生態與應用

T型



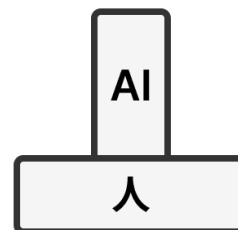
運用 AI 協助解決

O型



產生運用 AI 發生質變

逆T型



跨產品整合運作



Demo: ChatGPT

一本正經地胡說八道

高雄和大阪的時差



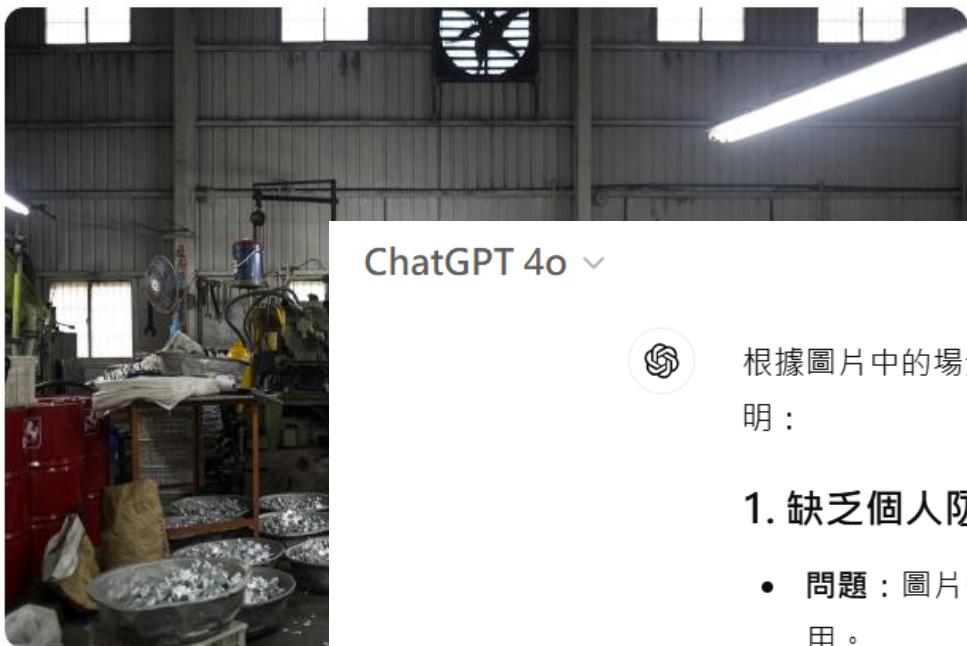
高雄（台灣）和大阪（日本）之間沒有時差。兩地都位於東八區（UTC+8），因此時間是一樣的。

你確定你的答案嗎？



謝謝提醒，我查了一下，確實應該修正我的答案。

高雄（台灣）是東八區時間（UTC+8），而大阪（日本）是東九區時間（UTC+9），因此大阪比高雄快一小時。例如，如果高雄是上午 9 點，大阪會是上午 10 點。



您是一個還安衛的專家，
哪些違反法規的地方，以

ChatGPT 4o ▾



根據圖片中的場景，以下是可能的違反勞工安全衛生相關法規的地方，以及對應的處理建議與罰則說明：

1. 缺乏個人防護設備 (PPE)

- **問題：**圖片中僅部分工人戴著口罩，其他個人防護裝備（如護目鏡、防護手套等）未見明顯使用。

- **相關法規：**

- 《職業安全衛生法》第6條、第25條：雇主應提供勞工適當的個人防護具。

- **罰則：**

- 違反者可處新臺幣30,000元以上150,000元以下罰款。

- **建議：**要求全體員工佩戴適當的PPE，包括護目鏡、防護手套和防塵口罩，並制定使用PPE的規範。

第 6 條

修訂

雇主對下列事項應有符合規定之必要安全衛生設備及措施：

- 一、防止機械、設備或器具等引起之危害。
- 二、防止爆炸性或發火性等物質引起之危害。
- 三、防止電、熱或其他之能引起之危害。
- 四、防止採石、採掘、裝卸、搬運、堆積或採伐等作業中引起之危害。
- 五、防止有墜落、物體飛落或崩塌等之虞之作業場所引起之危害。
- 六、防止高壓氣體引起之危害。
- 七、防止原料、材料、氣體、蒸氣、粉塵、溶劑、化學品、含毒性物質或缺氧空氣等引起之危害。
- 八、防止輻射、高溫、低溫、超音波、噪音、振動或異常氣壓等引起之危害。
- 九、防止監視儀表或精密作業等引起之危害。
- 十、防止廢氣、廢液或殘渣等廢棄物引起之危害。
- 十一、防止水患、風災或火災等引起之危害。
- 十二、防止動物、植物或微生物等引起之危害。
- 十三、防止通道、地板或階梯等引起之危害。
- 十四、防止未採取充足通風、採光、照明、保溫或防濕等引起之危害。

雇主對下列事項，應妥為規劃及採取必要之安全衛生措施：

- 一、重複性作業等促發肌肉骨骼疾病之預防。
- 二、輪班、夜間工作、長時間工作等異常工作負荷促發疾病之預防。
- 三、執行職務因他人行為遭受身體或精神不法侵害之預防。
- 四、避難、急救、休息或其他為保護勞工身心健康之事項。

前二項必要之安全衛生設備與措施之標準及規則，由中央主管機關定之。



顧問見解

樹寶博士

解釋令

調整 Retry 策略

如果 SDK 內部有內建的 Retry 機制，你也可以考慮修改這部分。通常 Retry 策略應該在 OpenAI SDK 的設定中有所配置。如果 SDK 支援傳入自定義的 Retry 策略，你可以將重試次數減少。例如：

csharp

複製程式碼

```
var apiClient = new OpenAIClient(new OpenAIClientOptions
{
    MaxRetries = 2, // 將重試次數減少到 2 次
    Timeout = TimeSpan.FromMinutes(5) // 設定較長的 Timeout 時間
});
```

設定 MaxRetry 和 Timeout

具體的設定方法取決於你所使用的 OpenAI SDK 版本。你可以檢查 SDK 的官方文件或者查看 `OpenAIClientOptions` 中是否有相關的設定。

這樣你就可以延長 Timeout 時間並且減少重試次數，避免過度的重試導致 `AggregateException` 錯誤。





Demo: Azure OpenAI

讓 ChatGPT 的經驗到系統內

```
public async Task<string> SendTextToChat(string userQuestion)
{
    // 初始化 Chat 客戶端
    var chatClient = _azureOpenAIClient.GetChatClient("gpt-4o");
    var messages = new List<ChatMessage>
    {
        new SystemChatMessage("請用繁體中文回應"),
        new UserChatMessage(userQuestion)
    };

    // 呼叫 API 並獲取回應
    var response = await chatClient.CompleteChatAsync(messages);
    return response.Value.Content[0].Text;
}
```

```
public async Task<string> SendImageToChat(string imagePath, string imagePrompt = null)
{
    if (string.IsNullOrEmpty(imagePath) || !File.Exists(imagePath))
    {
        Console.WriteLine($"圖片檔案不存在：{imagePath}");
        throw new FileNotFoundException("請確認路徑是否正確並重試。");
    }

    var chatClient = _azureOpenAIClient.GetChatClient("gpt-4o");
    var messages = new List<ChatMessage>
    {
        new UserChatMessage(
            ChatMessageContentPart.CreateTextPart(imagePrompt),
            ChatMessageContentPart.CreateImagePart(
                await BinaryData.FromStreamAsync(new FileStream(imagePath, FileMode.Open, FileAccess.Read)),
                "image/jpeg"))
    };

    var response = await chatClient.CompleteChatAsync(messages);
    return response.Value.Content[0].Text;
}
```

```
// 主方法：使用向量資料庫來解析 PDF
public async Task<string> AnalyzePdfUsingVectorDb(string pdfFileName)
{
    // 初始化需要的客戶端
    var assistantClient = _azureOpenAIClient.GetAssistantClient();
    var fileClient = _azureOpenAIClient.GetOpenAIFileClient();
    var vectorStoreClient = _azureOpenAIClient.GetVectorStoreClient();

    // 上傳 PDF 文件並獲取文件 ID
    string uploadedFileId = await UploadPdfFileAsync(fileClient, pdfFileName);

    // 獲取向量存儲 ID
    string vectorStoreId = await GetVectorStoreIdAsync(assistantClient, _assistantid);

    // 將上傳的文件新增到向量存儲中
    await vectorStoreClient.AddFileToVectorStoreAsync(vectorStoreId, uploadedFileId, true);

    // 執行文件解析，並返回結果
    string threadResult = await RunAssistantAnalysisAsync(assistantClient, uploadedFileId);

    // 刪除上傳的文件
    await DeletePdfFileAsync(fileClient, uploadedFileId);
    return threadResult;
}
```

// 執行文件解析，並返回解析結果

```
private async Task<string> RunAssistantAnalysisAsync(AssistantClient assistantClient, string fileId)
{
```

// 定義初始訊息，請求解析文件的詳細規範

```
var messages = new MessageContent[]
```

```
{
```

\$"請完整解析這份文件：{Path.GetFileName(fileId)}，只要結果不要解釋，也不要使用 Markdown 美化格式"，

"如果資料有多行時請合併一起，將資料用 JSON 的格式，採用 Name，Value 的陣列格式"，

"將內容用以下格式呈現{ SDS: [Catalog: 化學品與廠商資料, Data: [{ Name: 化學品名稱, Value: 絕對酒精 }]] }，

```
};
```

初始化執行緒訊息

```
var initialMessage = new ThreadInitializationMessage(MessageRole.User, messages);
```

```
var threadOptions = new ThreadCreationOptions { InitialMessages = { initialMessage } };
```

// 建立並執行 Assistant 分析執行緒

```
var threadRun = (await assistantClient.CreateThreadAndRunAsync(_assistantId, threadOptions)).Value;
```

// 持續輪詢執行狀態，直到完成

```
while (!threadRun.Status.IsTerminal)
```

```
{
```

```
    await Task.Delay(TimeSpan.FromSeconds(1));
```

```
    threadRun = await assistantClient.GetRunAsync(threadRun.ThreadId, threadRun.Id);
```

```
}
```

// 獲取執行緒的所有訊息並返回

```
return await AggregateMessagesAsync(assistantClient, threadRun.ThreadId);
```

```
}
```

主要差異點：

- 版本差異：Azure GPT-4o 更可能基於 **GPT-4 v1**，而我基於更進一步的 **GPT-4 v2**。
- 最佳化目標：
 - Azure GPT-4o 強調高效能、低成本運行，適合大規模的企業需求。
 - GPT-4 v2 則側重於提升對話品質、創意生成和語境理解。

如果你希望了解這些差異如何影響實際應用，或需要更深入的技術比較，可以再告訴我！ 😊