單元 15：NotebookLM 自動化 Web 系統操作

上一個單元介紹的 Chrome 擴充程式，解決了「邊瀏覽邊蒐集」的效率問題。但如果你需要更完整的自動化能力——不只是加入來源，而是要管理筆記本、生成各種內容、批量處理檔案——那就需要更強大的工具了。

這個單元要介紹的是「NotebookLM 自動化 Web 系統」——一個 Flask 架構的網頁應用程式，讓你透過自然語言控制 NotebookLM 的所有功能。

這是什麼系統？

簡單來說，這是一個「自然語言介面」。你不需要記住任何指令，只需要用日常的中文描述你想做的事，系統就會自動執行。

比如說：

- 你輸入「列出我所有的筆記本」→ 系統顯示筆記本清單
- 你輸入「建立一個叫做『AI研究』的筆記本」→ 系統建立新筆記本
- 你輸入「幫我生成 Podcast」→ 系統開始生成 Podcast
- 你輸入「做一份心智圖」→ 系統生成心智圖

這樣的操作方式，比在 NotebookLM 網頁上點來點去要快得多，也更適合需要批量處理的工作流程。

系統架構總覽

在深入使用之前，讓我們先理解這個系統是如何運作的。這不是為了讓你變成程式設計師，而是因為理解架構能幫助你更好地使用系統，並在遇到問題時知道該往哪裡找原因。

技術堆疊

這個系統的技術架構是這樣的：

讓我逐層解釋：

最上層：Flask Web GUI

這是你看到和操作的部分——一個在瀏覽器中運行的網頁介面。Flask 是一個 Python 網頁框架，輕量但功能強大。

第二層：NLP Parser

NLP 是 Natural Language Processing（自然語言處理）的縮寫。這個模組負責理解你輸入的中文，把它轉換成系統可以執行的「意圖」。比如它能理解「生成 Podcast」、「製作音訊」、「產生播客」都是同一個意思。

第三層：notebooklm-py

這是一個開源的 Python 函式庫，透過逆向工程 NotebookLM 的未公開 API，提供程式化的存取能力。它是整個系統的核心引擎。

第四層：Playwright

Playwright 是一個瀏覽器自動化工具。因為 NotebookLM 沒有公開的官方 API，所以 notebooklm-py 使用 Playwright 來模擬真人在瀏覽器上的操作——點擊按鈕、填入表單、等待結果。

最底層：NotebookLM 網頁

這就是 Google 的 NotebookLM 服務本身。所有的資料最終都是存在 Google 的伺服器上。

資料流向

當你在系統中輸入「幫我生成 Podcast」時，發生的事情是這樣的：

1. **前端**接收你的輸入文字
2. 文字送到 **Flask 後端**
3. **NLP Parser** 分析文字，識別出意圖是「生成音訊」
4. 後端呼叫 **notebooklm-py** 的對應函式
5. notebooklm-py 啟動 **Playwright** 控制的瀏覽器
6. Playwright 在 **NotebookLM 網頁**上執行操作：點擊「產生 Podcast」按鈕、等待生成完成
7. 結果層層回傳到前端顯示

整個過程對你來說是透明的——你只需要說一句話，然後等待結果。

系統能做什麼？

這個系統幾乎可以操控 NotebookLM 的所有功能，甚至包括一些網頁版不容易做到的事。

筆記本管理

功能	自然語言範例
列出所有筆記本	「列出我所有的筆記本」、「顯示筆記本清單」
建立新筆記本	「建立一個叫做『專案研究』的筆記本」
重新命名	「把這個筆記本改名為『2024 報告』」
刪除筆記本	「刪除這個筆記本」
分享筆記本	「分享這個筆記本給 xxx@gmail.com」

來源匯入

來源類型	自然語言範例
網頁 URL	「新增這個網址 https://example.com」
YouTube 影片	「加入這個 YouTube 影片」
PDF 文件	「匯入這份 PDF」
文字檔案	「加入這個 TXT 檔案」
Google Drive	「從我的雲端硬碟加入那份文件」
直接貼上文字	「把這段文字加入來源」

內容生成

這是最有價值的功能區塊——自動生成各種內容。

內容類型	自然語言範例
Podcast	「幫我生成 Podcast」、「製作音訊」
影片	「生成影片摘要」
簡報	「製作簡報」、「生成投影片」
測驗	「生成 10 題測驗」、「出考題」
閃卡	「製作學習閃卡」
報告	「做一份報告」、「生成文件摘要」
心智圖	「製作心智圖」、「生成腦圖」
資訊圖表	「生成資訊圖表」
數據表	「製作數據表」

問答查詢

你可以直接對文件提問：

- 「問一下這份文件的主要觀點是什麼？」
- 「查詢文件中提到的預算數字」
- 「這份研究的結論是什麼？」

下載匯出

這是網頁版不容易做到的功能——把生成的內容下載成可用的檔案。

內容	可匯出格式
Podcast 音訊	MP3
影片	影片檔案
簡報	PDF
閃卡	Markdown、Anki 格式
心智圖	可編輯格式
報告	文字檔案

安裝步驟詳解

現在讓我們開始安裝這個系統。

系統需求

在開始之前，請確認你的電腦符合以下條件：

項目	需求
作業系統	Windows 10/11、macOS、Linux
Python	3.9 以上版本
RAM	8GB 以上（瀏覽器自動化需要記憶體）
硬碟空間	500MB 以上
網路	需要連網
Google 帳號	需要能登入 NotebookLM

步驟一：安裝 notebooklm-py

首先，安裝核心的 notebooklm-py 函式庫。

開啟命令提示字元（Windows）或終端機（macOS/Linux），輸入：

```
# 安裝主程式
pip install notebooklm-py

# 安裝瀏覽器自動化套件
pip install "notebooklm-py[browser]"

# 安裝 Chromium 瀏覽器
playwright install chromium
```

這三個指令分別做了：

1. 安裝 notebooklm-py 主程式
2. 安裝 Playwright 瀏覽器自動化支援
3. 下載 Chromium 瀏覽器（這是 Playwright 用來模擬操作的瀏覽器）

步驟二：取得專案檔案

接下來，取得 Web 系統的專案檔案。

如果你有 Git，可以用以下指令：

```
git clone https://github.com/your-repo/notebooklm-automation.git
cd notebooklm-automation
```

如果沒有 Git，直接下載 ZIP 檔案並解壓縮到你想要的位置即可。

步驟三：安裝專案依賴

進入專案目錄，安裝 Flask 和其他依賴：

```
cd D:\NotebookLM課程\讓NotebookLM自動化
pip install -r requirements.txt
```

requirements.txt 的內容很簡單，主要就是 Flask：

```
flask>=3.0.0
```

步驟四：Google 帳號登入

這是關鍵步驟——讓系統能夠存取你的 NotebookLM。

在命令提示字元中輸入：

```
notebooklm login
```

執行後，會自動開啟一個瀏覽器視窗，顯示 Google 登入頁面。請用你的 Google 帳號登入。

登入成功後，回到命令提示字元視窗，按 Enter 確認。

重要：登入狀態會儲存在 `~/.notebooklm/storage_state.json` 這個檔案中。只要這個檔案存在，之後就不需要重新登入。如果登入狀態失效，重新執行 `notebooklm login` 即可。

步驟五：啟動應用程式

一切就緒，啟動系統：

```
python app.py
```

如果看到以下訊息，表示啟動成功：

步驟六：開啟瀏覽器使用

開啟你的瀏覽器，輸入網址：

你會看到系統的首頁。恭喜，安裝完成！

介面導覽

讓我們熟悉一下系統的介面。

首頁（執行介面）

這是你最常使用的頁面。

左側邊欄：

- 筆記本下拉選單：選擇要操作的筆記本
- 快捷按鈕：常用功能的快速存取圖示

中間區域：

- 自然語言輸入框：輸入你的指令
- 執行按鈕：送出指令
- 結果顯示區：顯示執行結果

右側區域：

- 登入狀態：顯示目前是否已登入 NotebookLM
- 任務狀態：顯示背景任務的進度

功能總覽頁

這個頁面列出了系統的所有功能，並提供每種功能的自然語言範例。

當你不確定該怎麼說時，可以來這裡參考範例，或直接點擊範例執行。

設定頁

在這裡可以調整系統的各種設定：

NLP 模式：選擇自然語言解析的方式

模式	說明	費用
關鍵字匹配	簡單快速，靠關鍵字判斷	免費
Gemini API	Google AI 理解語意	需 API Key
OpenAI API	GPT 理解語意	需 API Key

建議先用「關鍵字匹配」模式，如果覺得不夠智慧再升級。

介面主題：選擇你喜歡的視覺風格

- 簡約現代（預設）
- 豐富視覺
- 深色主題
- 淺色主題

基本操作流程

使用這個系統的基本流程是這樣的：

讓我們實際操作幾個範例。

範例一：列出筆記本

這是最簡單的操作，用來確認系統運作正常。

1. 在輸入框中輸入：「列出我所有的筆記本」
2. 點擊「執行」
3. 等待幾秒鐘
4. 結果區會顯示你的筆記本清單

如果成功看到筆記本清單，表示系統的連線和登入都正常。

範例二：加入來源

1. 先在左側選擇一個筆記本（或先建立一個）
2. 在輸入框中輸入：「新增這個網址 https://example.com」
3. 點擊「執行」
4. 等待系統處理
5. 成功後會顯示「來源已加入」的訊息

範例三：生成 Podcast

這是最受歡迎的功能之一。

1. 確保已選擇有來源的筆記本
2. 在輸入框中輸入：「幫我生成 Podcast」
3. 點擊「執行」
4. **注意**：Podcast 生成需要較長時間（約 2-5 分鐘）
5. 任務會在背景執行，你可以在右側「任務狀態」區域看到進度
6. 完成後會顯示結果，並提供下載連結

範例四：生成心智圖

1. 選擇有來源的筆記本
2. 輸入：「製作心智圖」
3. 執行
4. 結果區會以樹狀結構顯示心智圖

快捷按鈕說明

左側邊欄的快捷按鈕提供了常用功能的快速存取。滑鼠懸停在圖示上可以看到功能名稱。

圖示	功能
🎙️	生成 Podcast
🎬	生成影片
📊	生成簡報
❓	生成測驗
🃏	生成閃卡
📄	生成報告
🧠	生成心智圖
📈	生成資訊圖表
📋	生成數據表

點擊這些按鈕，相當於輸入對應的自然語言指令——系統會自動填入指令並執行。

自然語言的彈性

這個系統的一大優點是自然語言的彈性。你不需要記住精確的指令格式，用你習慣的方式說就好。

比如，以下這些說法都會被識別為「生成 Podcast」：

- 「幫我生成 Podcast」
- 「製作 Podcast」
- 「產生播客」

- 「生成音訊」
- 「做一個 Podcast」
- 「把這些內容變成 Podcast」

系統的 NLP Parser 會分析你的輸入，提取關鍵字，判斷你的意圖。

當然，說得越明確，系統理解得越準確。如果你發現系統誤解了你的意思，試著用更明確的關鍵字。

背景任務管理

有些操作需要較長時間，比如生成 Podcast 可能需要 2-5 分鐘。為了避免前端等待逾時，系統採用「背景任務」的機制。

當你執行一個長時間操作時：

1. 系統會在背景啟動一個任務
2. 前端會收到一個「任務 ID」
3. 你可以繼續在系統中做其他操作
4. 任務完成後，會在「任務狀態」區域顯示結果

這個機制讓你可以同時進行多個操作，不需要傻傻等待。

常見問題與解決方案

問題一：「No notebook specified」錯誤

原因：你想執行一個需要筆記本的操作（如加入來源、生成內容），但沒有先選擇筆記本。

解決：在左側下拉選單中選擇一個筆記本，或先執行「列出筆記本」然後選擇。

問題二：登入狀態失效

原因：儲存的 Cookie 過期或被清除。

解決：重新執行 `notebooklm login` 指令，完成登入流程。

問題三：生成 Podcast 很久沒反應

原因：Podcast 生成本來就需要時間，這是 NotebookLM 本身的處理時間。

解決：

1. 檢查右側「任務狀態」區域，看任務是否還在執行中
2. 如果顯示「進行中」，請耐心等待
3. 如果超過 10 分鐘沒有動靜，可能是發生了錯誤，嘗試重新執行

問題四：NLP 解析不準確

原因：關鍵字匹配模式有其限制，無法理解太口語或太創意的說法。

解決：

1. 使用更明確的關鍵字，例如「生成 Podcast」而非「做個音檔給我」
2. 參考「功能總覽」頁面的範例說法
3. 如果經常遇到這個問題，可以在設定頁切換為 Gemini 或 OpenAI 模式（需要 API Key）

問題五：無法遠端存取

原因：預設情況下，Flask 只監聽 localhost，外部電腦無法連入。

解決：修改 `app.py`，將 host 從 `127.0.0.1` 改為 `0.0.0.0`：

```
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

然後確保防火牆允許 5000 port 的連入。

進階使用技巧

技巧一：批量操作

雖然系統是單指令執行的，但你可以快速連續輸入多個指令，實現批量操作的效果。

比如，要加入多個來源：

1. 輸入「新增這個網址 https://example1.com」→ 執行

2. 輸入「新增這個網址 https://example2.com」→ 執行
3. 輸入「新增這個網址 https://example3.com」→ 執行

每次執行後，等幾秒鐘確認成功，再輸入下一個。

技巧二：結合不同功能

這個系統的強大之處在於它整合了 NotebookLM 的所有功能。你可以設計一個工作流程，結合不同的功能達成目標。

比如，要為一份研究資料製作完整的學習材料：

1. 「新增這個 PDF 檔案」
2. 「製作心智圖」→ 了解整體結構
3. 「生成 Podcast」→ 通勤時聽
4. 「製作學習閃卡」→ 複習用
5. 「生成 10 題測驗」→ 自我檢測

這樣一套流程下來，你從一份 PDF 得到了四種不同的學習材料。

技巧三：使用 Gemini/OpenAI 模式

如果你有 Google AI 或 OpenAI 的 API Key，可以在設定頁啟用更智慧的 NLP 模式。

這些 AI 模式的好處是：

- 理解更自然、更口語的說法
- 能處理較複雜的指令
- 更容易正確識別你的意圖

不過這會產生 API 費用，請根據自己的需求決定。

部署方式

這個系統可以用不同的方式部署，適應不同的使用情境。

方式一：本機使用（最簡單）

這是預設的使用方式，適合個人使用。

```
python app.py
```

訪問 http://localhost:5000

方式二：區域網路分享

如果你想讓同一個區域網路內的其他電腦也能使用這個系統，修改 `app.py`：

```
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

其他電腦可以用你的 IP 位址存取：`http://你的IP:5000`

注意：NotebookLM 的登入狀態是綁定在伺服器電腦上的。其他電腦使用時，實際上是用伺服器電腦的 Google 帳號。

方式三：開機自動啟動

如果你經常使用這個系統，可以設定成開機自動啟動。

在 Windows 上，建立一個 `start.bat` 檔案：

```
@echo off
cd /d D:\NotebookLM課程\讓NotebookLM自動化
python app.py
pause
```

然後把這個檔案（或它的捷徑）放入 Windows 的啟動資料夾：

1. 按 `Win + R`
2. 輸入 `shell:startup`
3. 把 `start.bat` 複製進去

這樣每次開機，系統就會自動啟動。

與其他工具的整合

這個 Web 系統可以和前面介紹的其他工具搭配使用。

與 Chrome 擴充程式搭配

- **Chrome 擴充程式**：適合「邊瀏覽邊蒐集」——快速加入單一來源
- **Web 系統**：適合「集中處理」——管理筆記本、生成內容、匯出結果

你可以用 Chrome 擴充程式快速蒐集一堆來源，然後用 Web 系統來生成各種內容。

與後製工具搭配

- 用 Web 系統生成簡報
- 用 PDNob 編輯文字
- 用 Canva Pro 提升視覺
- 用 Lovart 精細調整

這樣的組合拳，能讓你從原始資料到最終成品，全流程自動化。

本章小結

NotebookLM 自動化 Web 系統是一個強大的效率工具。它的核心價值是：

自然語言介面：用日常中文控制 NotebookLM，不需要記指令

完整功能覆蓋：筆記本管理、來源匯入、內容生成、問答查詢、下載匯出

背景任務管理：長時間操作不阻塞，可以同時進行多個任務

彈性部署：本機使用、區域網路分享、自動啟動

這個系統特別適合需要頻繁使用 NotebookLM 的人。一旦設定完成，你只需要用最直覺的方式說出需求，系統就會自動完成所有操作。

下一個單元，我們會介紹另一個專門處理簡報的系統：簡報編修系統。這個系統專注於 PDF 轉 PowerPoint 的功能，並提供完整的線上編輯能力。