12.8 本章小結

MCP（模型上下文協議）是 AI 開發領域的重要突破，它讓 Claude Code 從單純的程式碼生成工具，進化成能夠操作各種外部服務的強大 AI 助手。透過 MCP，開發者不再需要熟悉每個服務的 API 語法，只要用自然語言就能操作資料庫、檔案系統、GitHub、Jira 等各種工具，讓 Claude Code 從一個單純的程式碼產生器，有限的工具使用，成為一個程式開發的 Hub，能夠使用各種工具來完成任務。

本章也詳細介紹了 MCP 的架構原理、四種安裝方式（本機執行檔、npm 套件、SSE 伺服器、HTTP 伺服器），以及三種範圍管理策略（本機、專案、使用者範圍）。這些不同的安裝和管理方式，讓開發者能夠根據實際需求選擇最適合的部署策略，無論是個人開發、團隊協作還是企業級應用都能找到合適的解決方案。

MCP 的真正價值在於它建立了一個標準化的協議，讓全世界的服務都能透過統一的方式與 AI 溝通。隨著越來越多的服務支援 MCP，Claude Code 將成為開發者手中的「變形金鋼」，能夠適應各種開發場景的需求，大幅提升開發效率和體驗。在下一章，我們會介紹一些 Claude Code 最常用 MCP Server，以及如何使用 MCP 來完成各種任務。



13

20+20+20 個最重要的 MCP Server 大閱兵

上一章我們詳細了解了 MCP 的原理和安裝方式，這章就來看看目前最重要、最實用的 20 個 MCP Server。這些 MCP Server 涵蓋了開發、測試、部署、監控等各個環節，讓 Claude Code 真正成為你的開發利器。我會按照功能分類介紹每個 MCP Server，並提供實用的 Prompt 範例，讓你立刻上手。最後我們也會建議另外 20 + 20 個好用的 MCP Server。

另外提醒讀者，如上一章所示，MCP 安裝方式有多種，這邊會以官方推薦的方式來安裝。此外不同的 MCP Server 安裝時，可能會需要不同的認證，如 GitHub PAT、Docker Hub PAT 等。讀者們可以根據每一個 MCP Server 的官網說明，查看如何取得需要的變數。

13.1 開發協作工具

既然 Claude Code 是個開發工具，我們當然要先介紹開發協作工具。開發工具最重要的就是版本控制以及協作，因此我們會以 GitHub、GitLab、Linear 這三個工具來介紹。

13.1.1 GitHub MCP Server（官方）

GitHub 官方提供的 MCP Server 是開發者必備工具，讓你可以用自然語言操作 GitHub 的各種功能。

安裝方式：

```
claude mcp add --transport http github https://api.githubcopilot.com/mcp -H
"Authorization: Bearer YOUR_GITHUB_PAT"
```

說明：

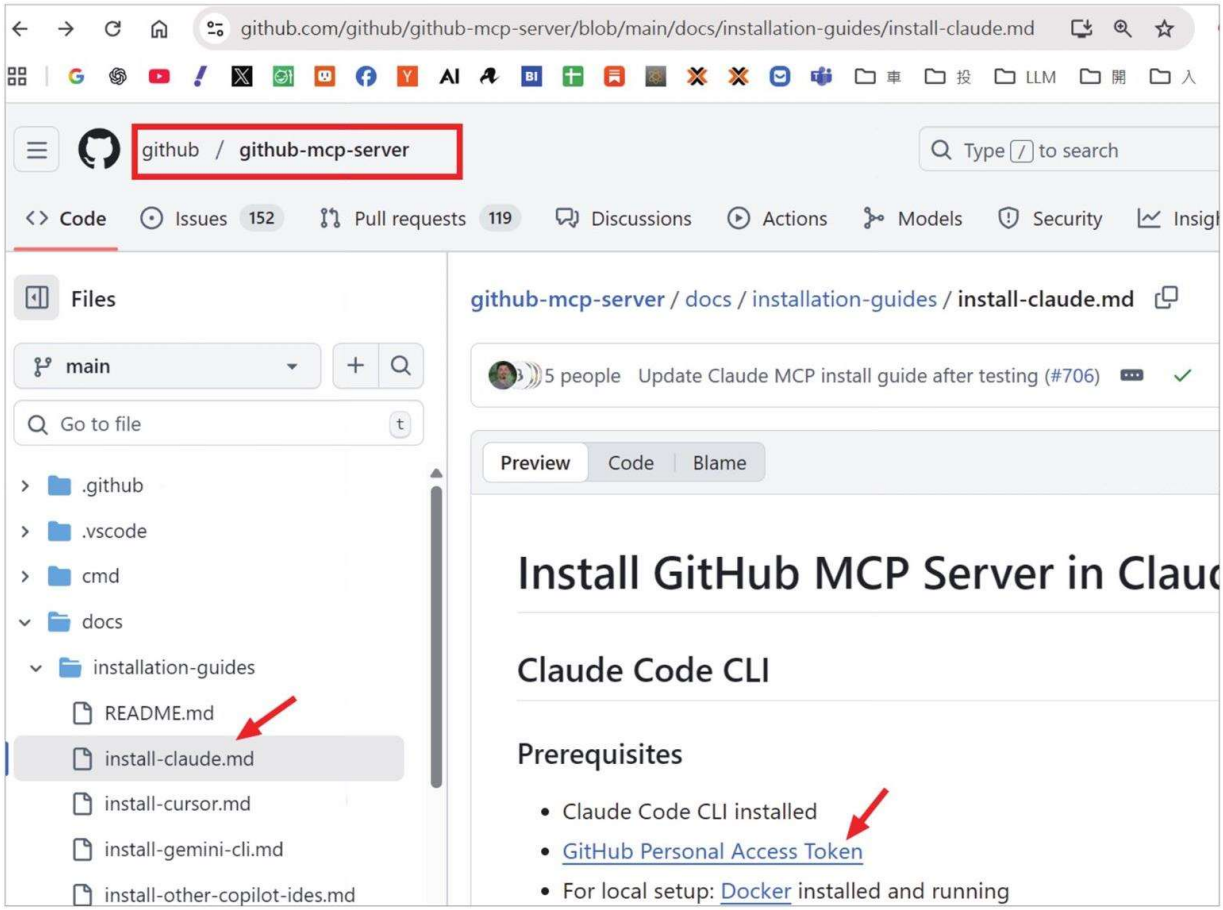
這個方式是使用 GitHub 的 MCP Server，因此需要一個 GitHub PAT (Personal Access Token)。你可以登入 GitHub 並且在 Settings-> Developer settings-> Personal access tokens 中建立一個新的 PAT。

常用 Prompt 範例：

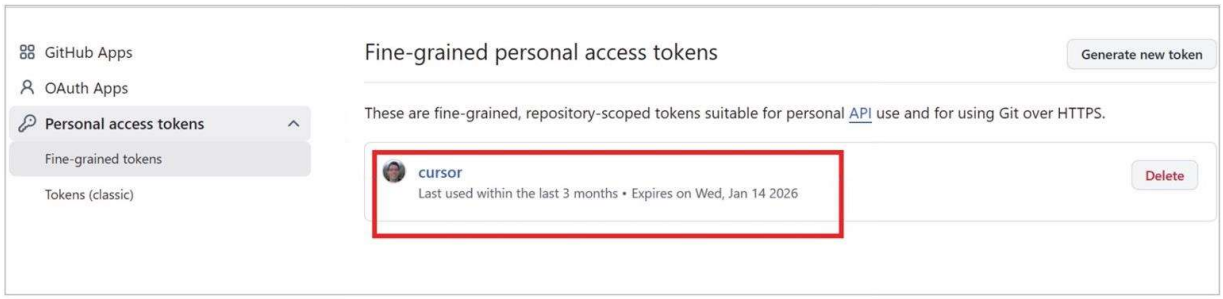
```
> 列出此 repo 未合併 PR，依最近更新排序，幫我加上總結與接下來要做的 3 點，然後建立一則彙整留言。
> 為 issue #123 產生可執行的重現步驟並開一支 hotfix 分支，推送後開 PR。
> 搜尋本組織內 'feature flag' 的使用處並產生影響面評估表。
```

來源：[GitHub 官方 MCP Server](https://github.com/github/github-mcp-server)

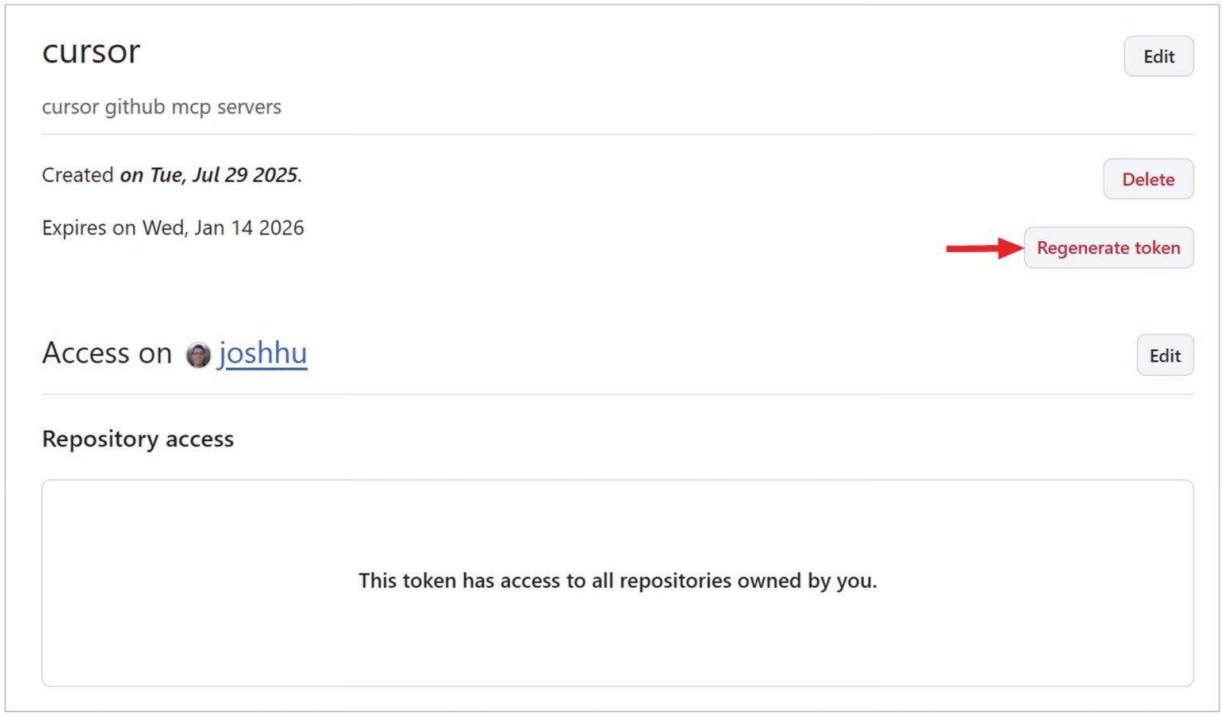
這是本章的第一個 MCP Server，因此我們詳細示範安裝方式，接下來的 MCP Server 就不再詳細示範安裝方式，讀者可以根據這個範例來安裝其它的 MCP Server。



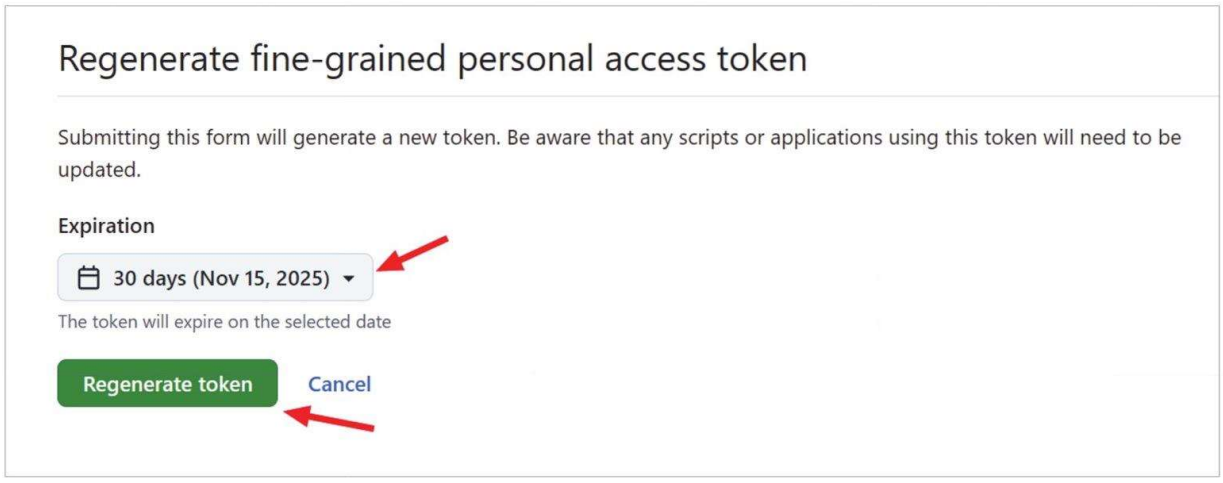
▲ 圖 13-1 先進入這個網頁



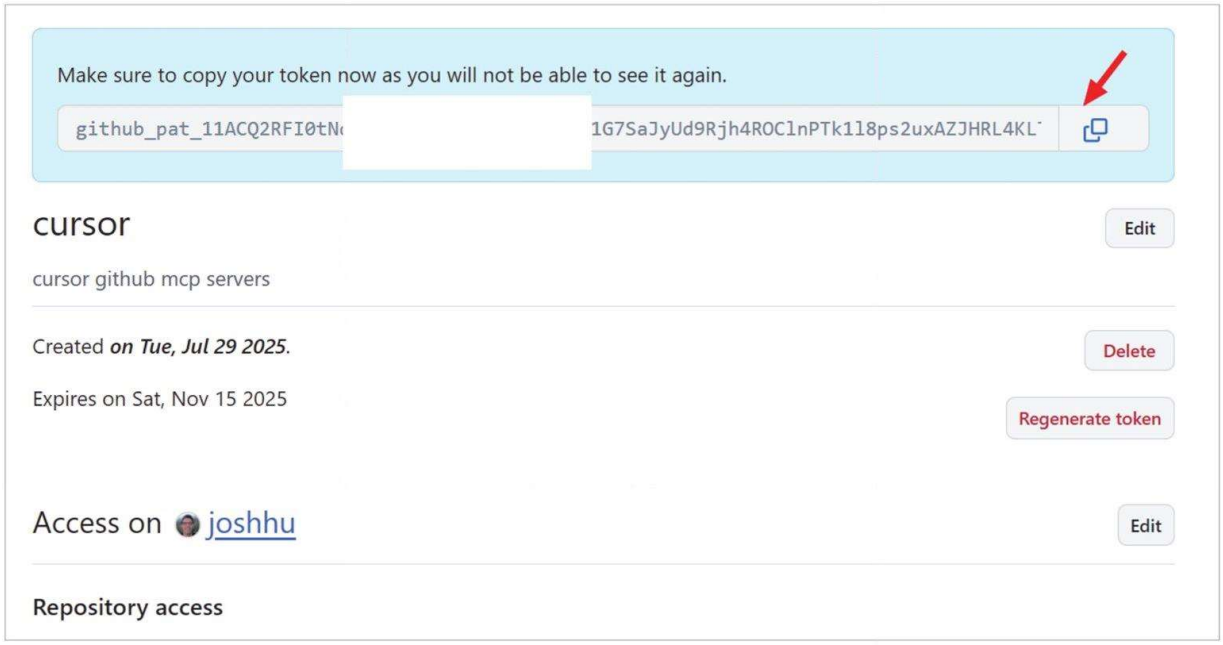
▲ 圖 13-2 這邊會出現之前建立的 PAT，可以沿用



▲ 圖 13-3 因為只出現一次就看不到了，因此重新產生



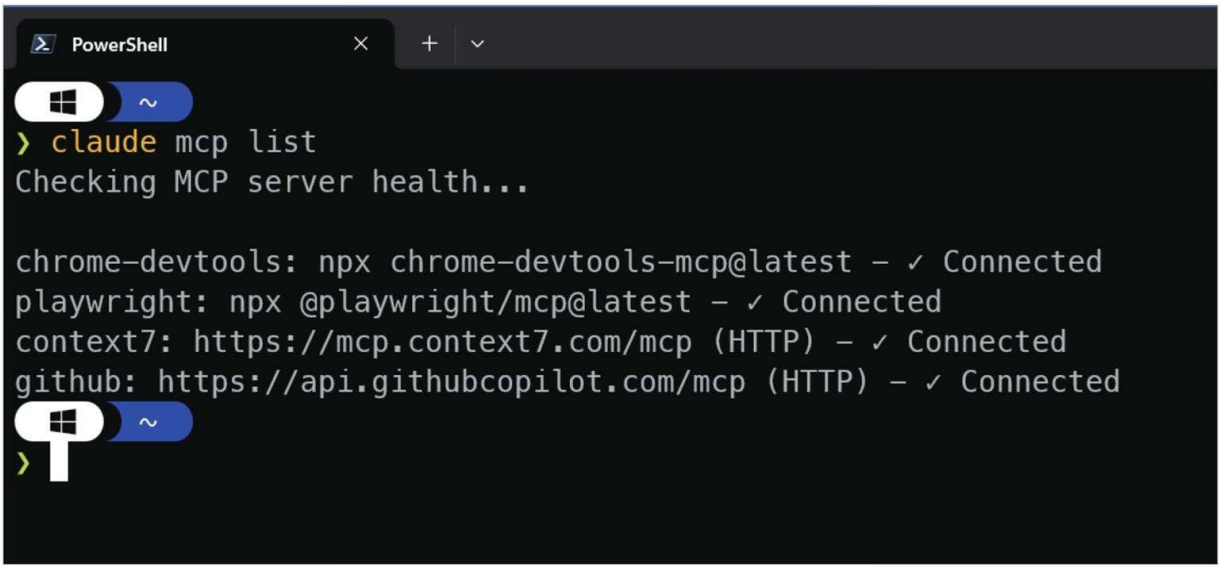
▲ 圖 13-4 選擇過期日，並且按下重新產生



▲ 圖 13-5 複製下來，以後就看不到了



▲ 圖 13-6 進入終端機，輸入前面提到的安裝指令，別忘記把 PAT 填入



▲ 圖 13-7 輸入 `claude mcp list` 來確認是否安裝成功

13.1.2 GitLab MCP（官方 / 社群）

GitLab 的 MCP 整合讓你可以管理 GitLab 專案的各個方面。

安裝方式：

官方建議是在 `.mcp.json` 中新增一個 GitLab 的 MCP Server。

```
{
  "mcpServers": {
    "GitLab": {
      "command": "npx",
      "args": [
        "mcp-remote",
        "https://<gitlab.example.com>/api/v4/mcp",
        "--static-oauth-client-metadata",
        "{\"scope\": \"mcp\"}"
      ]
    }
  }
}
```

常用 Prompt 範例：

```
> 列出此群組近 7 日的 MR 併入率與待審清單，並加總 reviewer 負載。
> 建立 CI 變更：在 test stage 加上快取與重試策略，開 MR 並附 diff 摘要。
> 把 issue #45 的討論結論寫回描述並關閉。
```

來源：[GitLab MCP 官方文件](https://docs.gitlab.com/user/gitlab_duo/model_context_protocol/mcp_server/)

13.1.3 Linear MCP（官方 Remote）

Linear 是新一代的專案管理工具，官方提供的 MCP Server 讓專案管理更加智慧。

安裝方式：

```
claude mcp add --transport sse linear-server https://mcp.linear.app/sse
```

常用 Prompt 範例：

```
> 搜尋 'payments' 相關的 open issues，依 Priority 排序後指派給對應小組。
> 建立 Epic 與三個子 Issue，連結 GitHub PR 並設定目標里程碑。
> 匯出本季完成度與 cycle time，生成週會圖表（附件 md）。
```

來源：[Linear MCP 官方文件](https://linear.app/docs/mcp)

13.2 自動化測試與瀏覽器操作

筆者在使用 Claude Code 時，最愛看的畫面就是看到 Claude Code 自己開啟瀏覽器，然後進行自動化測試，省去不少手動操作的麻煩。以本章內容為例，筆者就是用 Playwright 要求 Cursor 逐一檢查本章內容所提的連結。我們這邊看最重要的幾個工具。

13.2.1 Playwright MCP（瀏覽器自動化）

Microsoft Playwright 的 MCP 整合讓你可以用自然語言控制瀏覽器進行自動化測試。

安裝方式：

```
claude mcp add playwright npx @playwright/mcp@latest
```

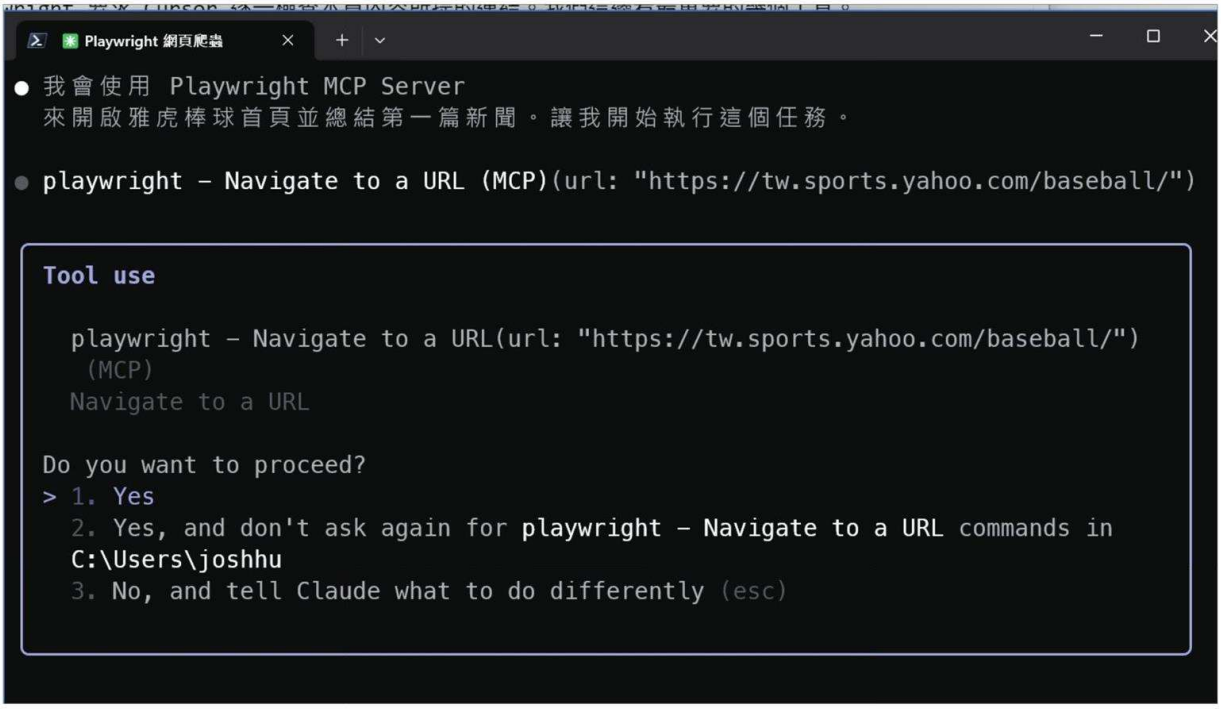
常用 Prompt 範例：

```
> 啟動無頭瀏覽，打開 staging 站 /checkout，輸入測試卡號完成下單並傳回可存證的可存取性快照。
> 巡檢 /pricing 頁面所有外鏈是否 200；失敗傳回網址與 HTTP 狀態。
> 對 /login 寫一支最小可行 E2E，保存成 e2e/login.spec.ts。
```

來源：[Microsoft Playwright MCP](https://github.com/microsoft/playwright-mcp)



▲ 圖 13-8 你可以在 Claude Code 中直接指定這個 MCP Server 來進行瀏覽器自動化測試



▲ 圖 13-9 就會看到一系列 Playwright 的工具使用，如果你沒設權限就會一直詢問



▲ 圖 13-10 過程中會一直呼叫不同的工具

13.2.2 Browser-use

強大的網頁瀏覽和擷取工具，適合需要複雜網頁互動的場景。

安裝方式：

```
{
  "mcpServers": {
    "browser-use": {
      "command": "uvx",
      "args": ["browser-use", "--mcp"],
      "env": {
        "OPENAI_API_KEY": "your-openai-api-key-here"
      }
    }
  }
}
```

由於是使用 uvx 來安裝，因此需要先安裝 uv。

常用 Prompt 範例：

```
> 用關鍵字 'MCP Registry' 搜尋，抓前 5 頁並產生重點摘要與引用清單。
> 抓取這 3 篇評測頁的規格表並對齊欄位輸出比較表。
> 登入內部入口網站，依指示下載最新報表並存到 reports/。
```

來源：[browser-use MCP 官方文件](https://docs.browser-use.com/customize/mcp-server)

13.3 知識與文件管理

如果只能讓我選擇一個 MCP Server，我會選擇 Context7。這個整合了全世界最新文件的 MCP Server，讓你使用 Claude Code 時，可以隨時查詢最新文件。尤有甚者，Claude Code 寫的程式都會照著最新文件的標準來編寫，保證你寫的程式都是最新版本，這是筆者的親身體驗。我們在這一節就來看看這些重要的文件管理工具。

13.3.1 Context7 MCP

Context7 提供即時的技術文件查詢，確保你使用的程式碼範例都是最新版本。

安裝方式：

```
claude mcp add --transport http context7 https://mcp.context7.com/mcp
```

常用 Prompt 範例：

```
> 撰寫一個 Next.js middleware 驗證 Cookie 內 JWT，請回填對應版本的 API 實作與官方範例連結，use context7
> 我用 Prisma v5，請產出 'soft delete' 的 schema 與查詢範例，保證與最新文件一致，use context7
> Cloudflare Workers 針對 JSON API 設 5 分鐘快取的程式碼與測試說明，use context7
```

來源：[Upstash Context7 MCP](https://github.com/upstash/context7)



▲ 圖 13-11 在任何時候，可以提示詞中直接加入 use context7 來使用 Context7 MCP



▲ 圖 13-12 就會看到呼叫各種工具來獲得我們要的資料

13.3.2 Notion MCP（官方 Hosted）

Notion 官方提供的 MCP Server 讓你可以直接操作 Notion 工作區。

安裝方式：

```
claude mcp add --transport http notion https://mcp.notion.com
```

常用 Prompt 範例：

```
> 在資料庫 'Roadmap' 建立 3 筆卡片，欄位（Team、Status、ETA）請與工作區定義一致。
> 把 'Release Notes' 頁整理成摘要，並反寫回同頁最後一節。
> 查 'OKR 2025' 裡屬於 Data 科的 KRs 與目前進度，產出周會簡報大綱。
```

來源：[Notion MCP 官方](https://github.com/makenotion/notion-mcp-server)

13.3.3 Atlassian Remote MCP（Jira/Confluence/Compass）

這邊要說明一下所謂的「票券」系統。在軟體開發時，如果使用者有問題，會建立一個「票券」來追蹤問題。這有點像硬體故障時要填報修單這概念，而這個方法目前已經是軟體工程中的重要方法，而管理這些票券的系統就是所謂的「票券」系統。Atlassian 就是這個領域的佼佼者。因此我們會以 Atlassian 的 MCP 整合涵蓋 Jira、Confluence 等多個產品。

安裝方式：

```
{
  "mcpServers": {
    "mcp-atlassian": {
      "command": "docker",
      "args": [
        "run",
        "--rm",
        "-i",
        "--env-file",
        "/path/to/your/mcp-atlassian.env",
        "ghcr.io/sooperset/mcp-atlassian:latest"
      ]
    }
  }
}
```

由於使用 Docker 來安裝，因此需要先安裝 Docker。

常用 Prompt 範例：

```
> 在 Jira 為 bug 'Checkout failed' 建立 ticket，欄位與範本照團隊規範，並連結相關
Confluence 文章。
> 把本週 Confluence 決議轉成 Jira 子工作並排程。
> 列出 PROJ 的 '過期未關閉' Issue 與責任人，貼回 Slack。
```

來源：[Atlassian Remote MCP 社群專案](https://github.com/sooperset/mcp-atlassian)

13.3.4 exa-code MCP

這是一個讓 context7 黯然失色的工具，因為他可以讓你查詢任何文件，包括程式碼、文件、網頁等等，提供查詢超過 10 億個文件庫，是真正的萬能工具。這個工具的前身就是 Exa Search，現在已經改名為 exa-code。

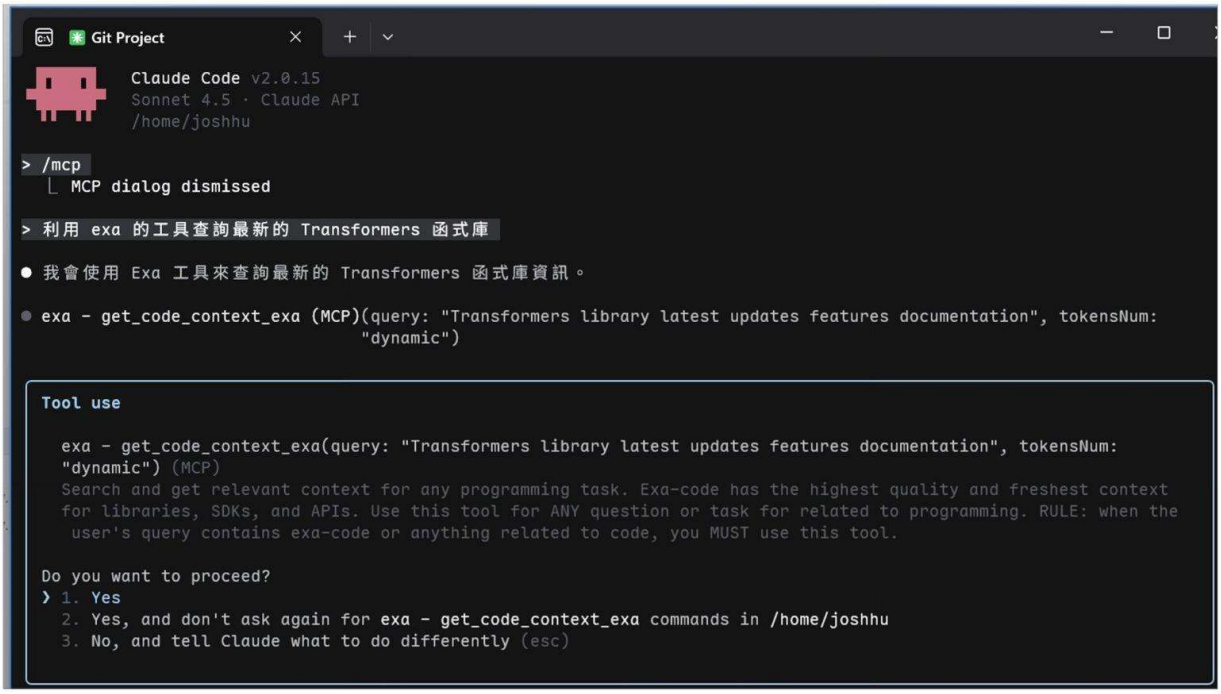
安裝方式：

```
claude mcp add exa -e EXA_API_KEY=[redacted] -- npx -y
exa-mcp-server
```

常用 Prompt 範例：

```
> 示範如何在 TypeScript 中使用 React hooks
> 找到使用 NextJS 實作身份驗證的範例
> 取得 pandas 函式庫的文件和範例
> 研究 exa.ai 公司並找到他們的定價資訊
> 開始一個關於人工智慧對醫療保健影響的深度研究專案，然後檢查完成狀態以獲得完整報告
```

來源：[Exa MCP 官方](https://docs.exa.ai/reference/exa-mcp)



▲ 圖 13-13 可以提示詞中直接加入 use exa 來使用 exa-code MCP

13.4 通訊協作工具

在開發過程中，和團隊成員的溝通是不可或缺的，Slack 是目前最流行的通訊工具，其類似 LINE 這種對話式通訊工具，是目前標準「鬆散式」管理的一個重要環境，我們在這節就來看看這些重要的通訊工具。

13.4.1 Slack MCP

Slack 的 MCP 整合讓你可以智慧管理團隊溝通，在需要通知成員時，可以在 Claude Code 中自動發送訊息。

安裝方式：

```
{
  "mcpServers": {
    "slack": {
      "command": "npx",
      "args": [
```

```
        "-y",
        "slack-mcp-server@latest",
        "--transport",
        "stdio"
      ],
      "env": {
        "SLACK_MCP_XOXC_TOKEN": "xoxc-...",
        "SLACK_MCP_XOXD_TOKEN": "xoxd-..."
      }
    }
  }
}
```

常用 Prompt 範例：

```
> 在 #eng 週會前 30 分鐘彙整上週含 'incident' 的訊息與連結。
> 將本討論串整理成決議與待辦，回貼到原訊息並標記相關成員。
> 搜尋本月含 'MCP' 的訊息與檔案，做一頁知識回顧。
```

來源：[Slack MCP 社群專案](https://github.com/zencoderai/slack-mcp-server)

13.4.2 Gmail MCP

Gmail 的 MCP 整合讓電子郵件管理變得更智慧。

安裝方式：

```
{
  "mcpServers": {
    "gmail": {
      "command": "npx",
      "args": [
        "@gongrzhe/server-gmail-autoauth-mcp"
      ]
    }
  }
}
```

常用 Prompt 範例：

```
> 用標籤 'invoices' 篩選近 30 日郵件，擷取發票金額與到期日做一張清單回覆我。
> 以 ' 採購單 #987' 為主旨回覆供應商並附上最新版報價單。
> 建立過濾器：寄件者 @example.com、自動套用 'vendor' 標籤並封存。
```

來源：[Gmail MCP 社群專案](https://github.com/GongRzhe/Gmail-MCP-Server)

13.5 雲端服務整合

開發時常常要和雲端的各種服務整合，Claude Code 當然也沒問題，接下來我們就來看看幾個常用的雲端專案。

13.5.1 Google Drive MCP

Google Drive 的整合讓檔案管理變得更加智慧。

安裝方式：

```
{
  "mcpServers": {
    "gdrive": {
      "command": "npx",
      "args": ["-y", "@isaacphi/mcp-gdrive"],
      "env": {
        "CLIENT_ID": "<CLIENT_ID>",
        "CLIENT_SECRET": "<CLIENT_SECRET>",
        "GDRIVE_CREDS_DIR": "/path/to/config/directory"
      }
    }
  }
}
```

常用 Prompt 範例：

```
> 在 'Quarterly Review' 資料夾中搜尋包含 'OKR' 的文件並匯出為 PDF。
> 把 data/report.csv 匯入到試算表新分頁，補上樞紐分析摘要。
> 比對兩份同名檔差異並產出合併建議。
```

來源：[Google Drive MCP 社群專案](https://github.com/isaacphi/mcp-gdrive)

13.5.2 Google Calendar MCP

Google Calendar 的整合讓行程管理更加便利。

安裝方式：

```
{
  "mcpServers": {
    "google-calendar": {
      "command": "npx",
      "args": [
        "-y",
        "@takumi0706/google-calendar-mcp"
      ],
      "env": {
        "GOOGLE_CLIENT_ID": "your_client_id",
        "GOOGLE_CLIENT_SECRET": "your_client_secret",
        "GOOGLE_REDIRECT_URI": "http://localhost:4153/oauth2callback"
      }
    }
  }
}
```

常用 Prompt 範例：

```
> 幫我在明天 10:00-11:00 建立『資料科學讀書會』，邀請 a@x.com，Google Meet 自動加入。
> 找出下週所有與 'roadmap' 相關會議並產生議程草案。
> 將 '一對一' 會議改為雙週一次並避免與 'Team Sync' 衝突。
```

來源：[Google Calendar MCP](https://github.com/takumi0706/google-calendar-mcp)

13.6 系統與基礎設施

系統與基礎設施是開發過程中不可或缺的一環，我們在這節就來看看這些重要的系統與基礎設施，包括最重要的容器方面以及排程方式的使用。

13.6.1 Kubernetes/EKS MCP

Kubernetes 的 MCP 整合讓叢集管理變得更加直覺，這個 MCP Server 是官方提供的，因此可以放心使用，而且除了支援 K8S 之外，也支援 OpenShift 等其他容器平台。

安裝方式：

```
{
  "mcpServers": {
    "kubernetes": {
      "command": "npx",
      "args": [
        "-y",
        "kubernetes-mcp-server@latest"
      ]
    }
  }
}
```

常用 Prompt 範例：

```
> 列出 prod 叢集 CPU/ 記憶體使用最高的 5 個 Deployment 並產生 HPA 建議值。
> 回報 checkout 服務從 v1 升級到 v2 的事件與失敗 Pod 日誌摘要。
> 比對 staging 與 prod Ingress 規則差異並草擬修補 PR。
```

來源：[Kubernetes MCP 官方](https://github.com/containers/kubernetes-mcp-server)

13.6.2 Docker Hub MCP Server

Claude Code 支援新一代的部署方式，目前使用 Docker Hub 的部署流程也早就成為容器的標準之一。這個 MCP Server 整合讓容器映像管理更方便，可以管理你在 Docker Hub 上的所有容器映像。

安裝方式：

```
{
    "mcpServers": {
        "docker-hub": {
            "command": "node",
            "args": [
                "/FULL/PATH/TO/YOUR/docker-hub-mcp-server/dist/index.js",
                "--transport=stdio",
                "--username=YOUR_DOCKER_HUB_USERNAME"
            ],
            "env": {
                "HUB_PAT_TOKEN": "YOUR_DOCKER_HUB_PERSONAL_ACCESS_TOKEN"
            }
        }
    }
}
```

常用 Prompt 範例：

```
> 替 'node + playwright' 搜尋最合適且下載量最高的 image，列出安全掃描結果與建議 tag。
> 比較三個 Postgres image（alpine、bitnami、官方），產生部署注意事項。
> 用最近 90 天趨勢挑選 CUDA 映像版本並附相容性說明。
```

來源：[Docker Hub MCP 官方](https://github.com/docker/hub-mcp)

13.6.3 Docker（本機容器操作 MCP）

本機 Docker 操作的 MCP 整合。

安裝方式：

```
"mcpServers": {
  "mcp-server-docker": {
    "command": "uvx",
    "args": [
      "mcp-server-docker"
```

```
    ],
    "env": {
      "DOCKER_HOST": "ssh://myusername@myhost.example.com"
    }
  }
}
```

常用 Prompt 範例：

```
> 列出本機異常的 container 與對應 docker logs 關鍵錯誤，輸出維護建議。
> 以 Compose 啟動 db + api + web 三服務，驗證健康檢查與連線。
> 建立一次性 Build Cache 清理腳本，保留最近兩週常用層。
```

來源：[Docker MCP 社群專案](https://github.com/ckreiling/mcp-server-docker)

13.7 資料庫與 API

資料庫與 API 是開發過程中不可或缺的一環，我們在這節就來看看這些重要的資料庫與 API，包括最重要的 PostgreSQL 等。

13.7.1 PostgreSQL/ 資料庫 MCP

操作資料庫是 Claude Code 常見的動作，當然也支援 MCP Server 來操作資料庫。

安裝方式：

```
{
  "mcpServers": {
    "postgres": {
      "command": "postgres-mcp",
      "args": [
        "--access-mode=unrestricted"
      ],
      "env": {
        "DATABASE_URI": "postgresql://username:password@localhost:5432/dbname"
```

```
      }
    }
  }
}
```

常用 Prompt 範例 **：

```
> 連線唯讀資料庫，列出慢查詢 Top 10 與對應索引建議。
> 為 orders 建立物化檢視與更新排程腳本（若僅允許唯讀，請傳回替代方案）。
> 產生 sales schema 的 ER 圖簡表與外鍵檢查。
```

來源：[PostgreSQL MCP 社群專案](https://github.com/crystaldba/postgres-mcp)



▲ 圖 13-14 我們可以隨時在程式中看到資料庫的內容

13.7.2 MySQL MCP Server

談到資料庫怎麼可以少了 MySQL，這個 MCP Server 整合讓你可以在 Claude Code 中操作 MySQL。

安裝方式：

```
npm install -g @benborla29/mcp-server-mysql
```

再來用 Claude Code 安裝

```
claude mcp add mcp_server_mysql \
  -e MYSQL_HOST="127.0.0.1" \
  -e MYSQL_PORT="3306" \
  -e MYSQL_USER="root" \
  -e MYSQL_PASS="your_password" \
  -e MYSQL_DB="your_database" \
  -e ALLOW_INSERT_OPERATION="false" \
  -e ALLOW_UPDATE_OPERATION="false" \
  -e ALLOW_DELETE_OPERATION="false" \
  -- npx @benborla29/mcp-server-mysql
```

常用 Prompt 範例 **：

```
> 查詢 users 資料表，並回傳所有資料。
> 新增一筆資料到 users 資料表，並回傳新增的資料。
> 更新 users 資料表中的一筆資料，並回傳更新的資料。
```

來源：[MySQL MCP 社群專案](https://github.com/benborla/mcp-server-mysql)

13.8 檔案系統與網路工具

13.8.1 Filesystem（檔案系統）

檔案系統的 MCP 整合是最基礎但也最重要的工具之一。

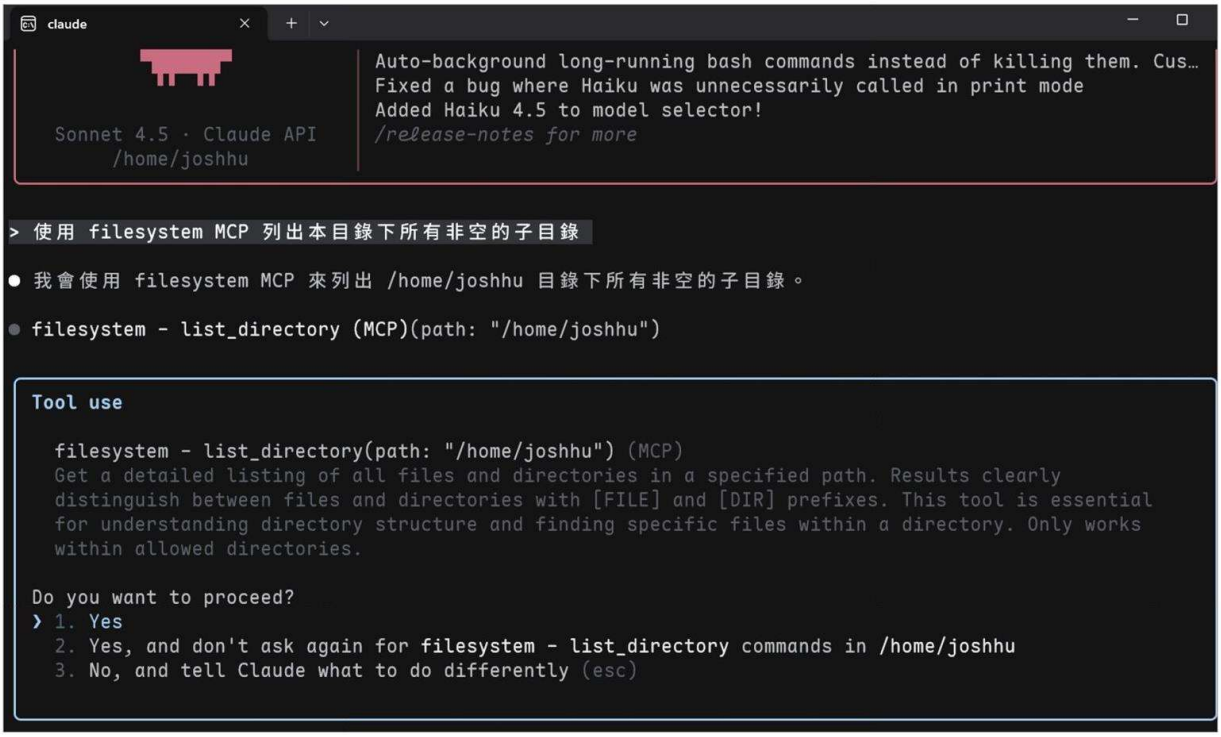
安裝方式：

```
claude mcp add filesystem npx @modelcontextprotocol/server-filesystem@latest
```

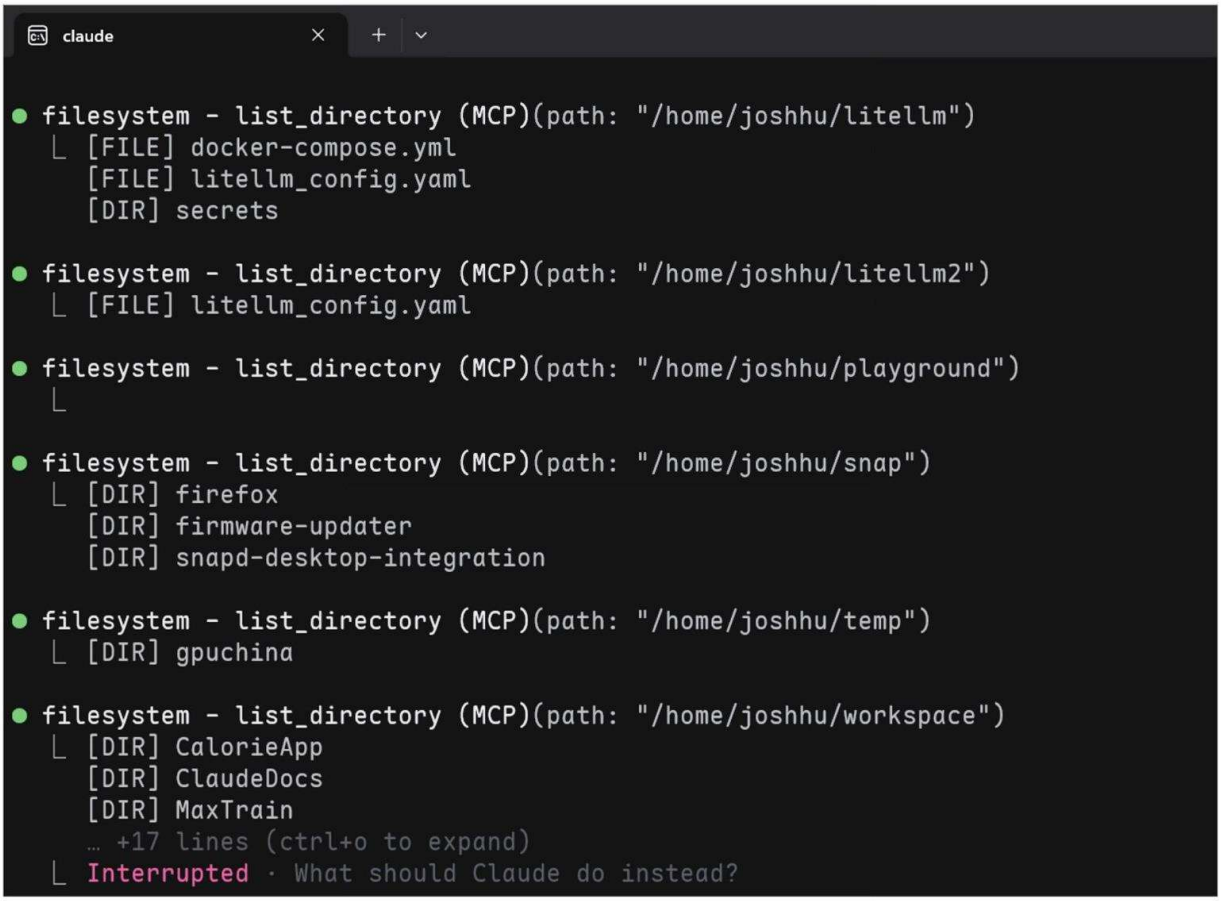
常用 Prompt 範例：

```
> 索引專案 src/ 與 config/，建立檔案地圖（含大小、最近修改），輸出成 repo_map.md。
> 將 docs/ 內的 PNG 轉 WebP 並更新引用路徑的 Markdown。
> 遞迴搜尋 TODO 標記並開一份修正清單（檔名、行號、建議修正）。
```

來源：[Filesystem MCP 官方](https://github.com/modelcontextprotocol/servers/tree/main/src/filesystem)



▲ 圖 13-15 使用這個 MCP



▲ 圖 13-16 幫我列出非空的子目錄，並回傳結果

13.8.2 Figma 網頁設計平台

Figma 是個網頁設計平台，這個 MCP Server 整合讓你可以在 Claude Code 中操作 Figma。

安裝方式：

```
claude mcp add --transport http figma-remote-mcp https://mcp.figma.com/mcp
```

常用 Prompt 範例：

```
> 在 Figma 建立一個新的設計，並設定為 'Product' 類型。
> 讀取 Figma 中的設計，並回傳設計的 JSON 格式。
> 在 Figma 中新增一個元件，並設定為 'Button' 類型。
```

來源：[Figma MCP 官方](https://help.figma.com/hc/en-us/articles/32132100833559-Guide-to-the-Figma-MCP-server)

13.9 最受歡迎的 20 個 MCP 伺服器

除了本章詳細介紹的核心 MCP Server 之外，以下是基於 Reddit 開發者社群真實回饋整理出的 20 個最受歡迎 MCP Server。這些伺服器經過實際使用驗證，具有高使用率和良好評價：

MCP 名稱	MCP 功能說明	範例 Prompt
Sequential Thinking MCP	程式碼邏輯一步步思考分析，適合除錯和演算法開發	「分析這個遞迴函數的執行流程，找出可能的效能問題」
Serena MCP	提供智慧程式建議，類似 LSP 的語言感知完成功能	「在這個函數中自動完成錯誤處理邏輯」
DesktopCommander MCP	程式碼導航、Git 操作和重構工具的桌面整合	「重構這個類別，將相關方法歸類到不同模組」
Supabase MCP	Supabase 後端服務管理，無需離開編輯器即可操作資料表和 API	「在 users 表中新增 email 驗證欄位，並更新相關 API」
Puppeteer MCP	瀏覽器動作自動化，支援點擊、填表單、UI 測試	「自動填寫註冊表單並截圖驗證結果」
DuckDuckGo MCP	輕量級網頁搜尋功能，適合快速查詢資訊	「搜尋 React 18 新功能的相關文章」
Knowledge Graph Memory MCP	建立結構化記憶圖，連接概念和檔案關係	「建立這個專案的知識圖譜，標示各模組間的依賴關係」
Markdownify MCP	將原始文字、文件或截圖轉換為乾淨的 Markdown 格式	「將這個 PDF 文件轉換為 Markdown 格式」

（續上表）

MCP 名稱	MCP 功能說明	範例 Prompt
Graphiti MCP	從結構化輸入生成圖表，視覺化程式碼依賴關係	「生成專案架構圖，顯示各個服務的關係」
Perplexity MCP	提供有來源引用的準確答案，適合研究和技術查詢	「查詢微服務架構的最佳實務，需要權威來源」
Magic UI MCP	根據描述生成前端組件，快速建立表單、導航列、儀表板	「建立一個包含搜尋功能的產品清單頁面」
Zen MCP	AI 模型路由器，自動選擇最適合的 AI 模型處理請求	「選擇最適合的模型來處理這個程式碼最佳化任務」
Figma Context MCP	連接 Figma 專案，使用自然語言搜尋和編輯設計檔案	「在設計檔案中找出所有按鈕元件並統一樣式」
Obsidian MCP	個人知識庫管理，支援雙向連結和標籤系統	「建立新筆記並自動連結到相關的專案文件」
MongoDB MCP	NoSQL 文件資料庫操作，支援彈性資料結構	「查詢所有活躍使用者的購買記錄，按時間排序」
Redis MCP	高效能記憶體資料結構儲存，常用於快取和會話管理	「設定使用者會話快取，過期時間 30 分鐘」
AWS MCP	Amazon Web Services 完整雲端服務整合	「部署 Lambda 函數並設定 API Gateway 路由」
Jira MCP	Atlassian 專業專案管理和問題追蹤系統	「建立新的衝刺，並指派優先順序高的任務」
Linear MCP	現代化軟體開發專案追蹤平台	「更新 issue 狀態並通知相關團隊成員」
MCP Compass	幫助探索和發現其他 MCP 伺服器的元工具	「推薦適合 Python 專案的 MCP 伺服器」

13.10 Claude Code 官方推薦的 MCP Server

Claude Code 官方也推薦了許多 MCP Server，我們整理成一張表如下：

分類	MCP 名稱	功能說明	安裝指令
開發與測試工具	Sentry	監控錯誤，除錯正式環境問題	`claude mcp add --transport http sentry https://mcp.sentry.dev/mcp`
	Socket	相依套件安全性分析	`claude mcp add --transport http socket https://mcp.socket.dev/`
	Hugging Face	存取 Hugging Face Hub 資訊和 Gradio AI 應用程式	`claude mcp add --transport http hugging-face https://huggingface.co/mcp`
	Jam	透過可存取 Jam 錄製內容（如影片、主控台紀錄、網路請求和錯誤）的 AI 代理來加速除錯	`claude mcp add --transport http jam https://mcp.jam.dev/mcp`
專案管理與文件	Asana	與你的 Asana 工作區互動，讓專案保持在軌道上	`claude mcp add --transport sse asana https://mcp.asana.com/sse`
	Atlassian	管理你的 Jira 票券和 Confluence 文件	`claude mcp add --transport sse atlassian https://mcp.atlassian.com/v1/sse`
	ClickUp	任務管理、專案追蹤	`claude mcp add clickup --env CLICKUP_API_KEY=YOUR_KEY --env CLICKUP_TEAM_ID=YOUR_ID -- npx -y @hauptsache.net/clickup-mcp`
	Intercom	存取即時客戶對話、票券和使用者資料	`claude mcp add --transport http intercom https://mcp.intercom.com/mcp`

（續上表）

分類	MCP 名稱	功能說明	安裝指令
專案管理與文件	Linear	與 Linear 的問題追蹤和專案管理整合	`claude mcp add --transport sse linear https://mcp.linear.app/sse`
	Notion	讀取文件、更新頁面、管理任務	`claude mcp add --transport http notion https://mcp.notion.com/mcp`
	Box	針對企業內容提問、從非結構化資料取得洞察、自動化內容工作流程	`claude mcp add --transport http box https://mcp.box.com/`
	Fireflies	從會議逐字稿和摘要中萃取有價值的洞察	`claude mcp add --transport http fireflies https://api.fireflies.ai/mcp`
	Monday	透過建立項目、更新欄位、指派負責人、設定時程、新增 CRM 活動和撰寫摘要來管理 monday.com 看板	`claude mcp add --transport sse monday https://mcp.monday.com/sse`
資料庫與資料管理	Airtable	讀取 / 寫入記錄、管理資料庫和資料表	`claude mcp add airtable --env AIRTABLE_API_KEY=YOUR_KEY -- npx -y airtable-mcp-server`
	Daloopa	提供來自 SEC 申報文件、投資者簡報等高品質基本財務資料	`claude mcp add --transport http daloopa https://mcp.daloopa.com/server/mcp`
	HubSpot	透過擷取聯絡人、公司和交易，以及建立和更新記錄來存取和管理 HubSpot CRM 資料	`claude mcp add --transport http hubspot https://mcp.hubspot.com/anthropic`

（續上表）

分類	MCP 名稱	功能說明	安裝指令
支付與商務	PayPal	整合 PayPal 商務功能、支付處理、交易管理	`claude mcp add --transport http paypal https://mcp.paypal.com/mcp`
	Plaid	分析、疑難排解和最佳化 Plaid 整合。銀行資料、金融帳戶連結	`claude mcp add --transport sse plaid https://api.dashboard.plaid.com/mcp/sse`
	Square	使用代理程式在 Square API 上建構。支付、庫存、訂單等	`claude mcp add --transport sse square https://mcp.squareup.com/sse`
	Stripe	支付處理、訂閱管理和金融交易	`claude mcp add --transport http stripe https://mcp.stripe.com`
設計與媒體	Figma	存取設計、匯出素材	`claude mcp add --transport http figma-dev-mode-mcp-server http://127.0.0.1:3845/mcp`
	Cloudinary	上傳、管理、轉換和分析你的媒體素材	多個服務可用，請參閱文件以了解特定伺服器網址
	invideo	在你的應用程式中建構影片創作功能	`claude mcp add --transport sse invideo https://mcp.invideo.io/sse`
	Canva	直接從 Claude 瀏覽、摘要、自動填入，甚至產生新的 Canva 設計	`claude mcp add --transport http canva https://mcp.canva.com/mcp`
基礎架構與 DevOps	Cloudflare	透過 Cloudflare 建構應用程式、分析流量、監控效能和管理安全設定	多個服務可用，請參閱文件以了解特定伺服器網址
	Netlify	在 Netlify 上建立、部署和管理網站。從建立機密到執行存取控制到彙總表單提交，控制網站的所有面向	`claude mcp add --transport http netlify https://netlify-mcp.netlify.app/mcp`

（續上表）

分類	MCP 名稱	功能說明	安裝指令
基礎架構與 DevOps	Stytch	設定和管理 Stytch 身份驗證服務、重新導向網址、電子郵件範本和工作區設定	`claude mcp add --transport http stytch http://mcp.stytch.dev/mcp`
	Vercel	Vercel 官方 MCP Server，讓你搜尋和導覽文件、管理專案和部署，以及分析部署紀錄——全部在一個地方	`claude mcp add --transport http vercel https://mcp.vercel.com/`
自動化與整合	Workato	透過 Workato 存取任何應用程式、工作流程或資料，為 AI 提供可存取性	MCP Server 以程式化方式產生
	Zapier	透過 Zapier 的自動化平台連接近 8,000 個應用程式	在 mcp.zapier.com 產生使用者專屬網址

13.11 MCP 集散地

提供 MCP Server 的網站很多，我們就來看幾個常見的。

https://mcp.so

https://github.com/modelcontextprotocol/servers

https://hub.docker.com/mcp

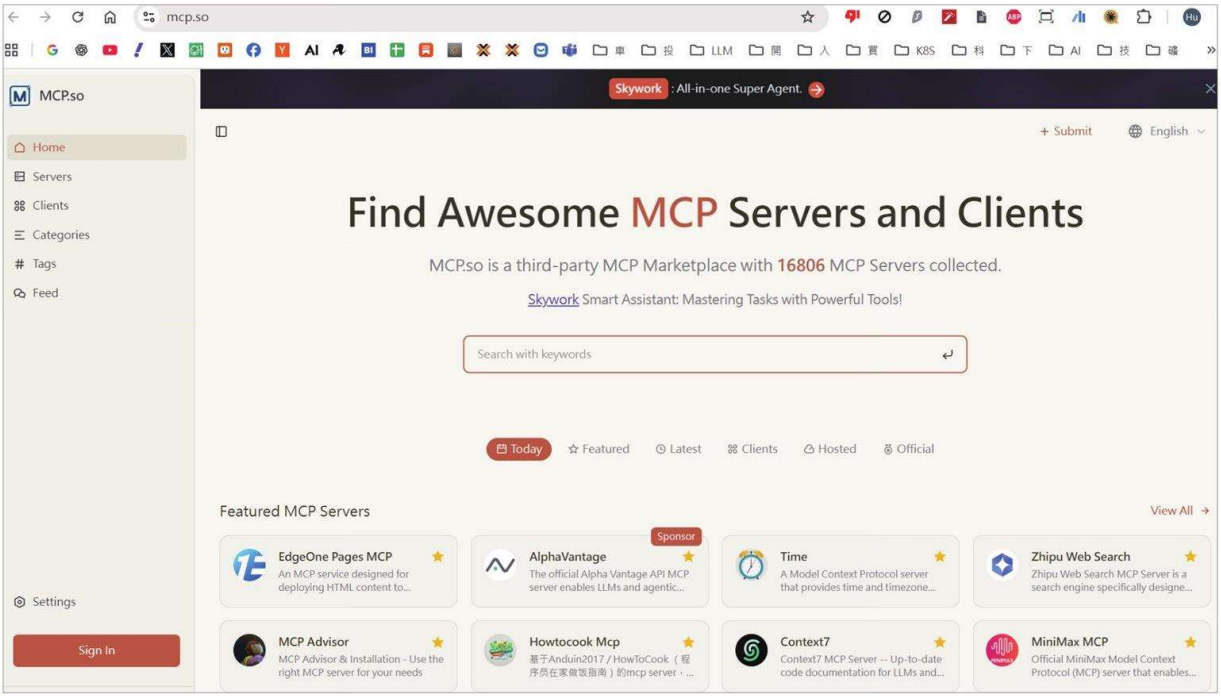
https://github.com/apappascs/mcp-servers-hub

https://mcpserverhub.net

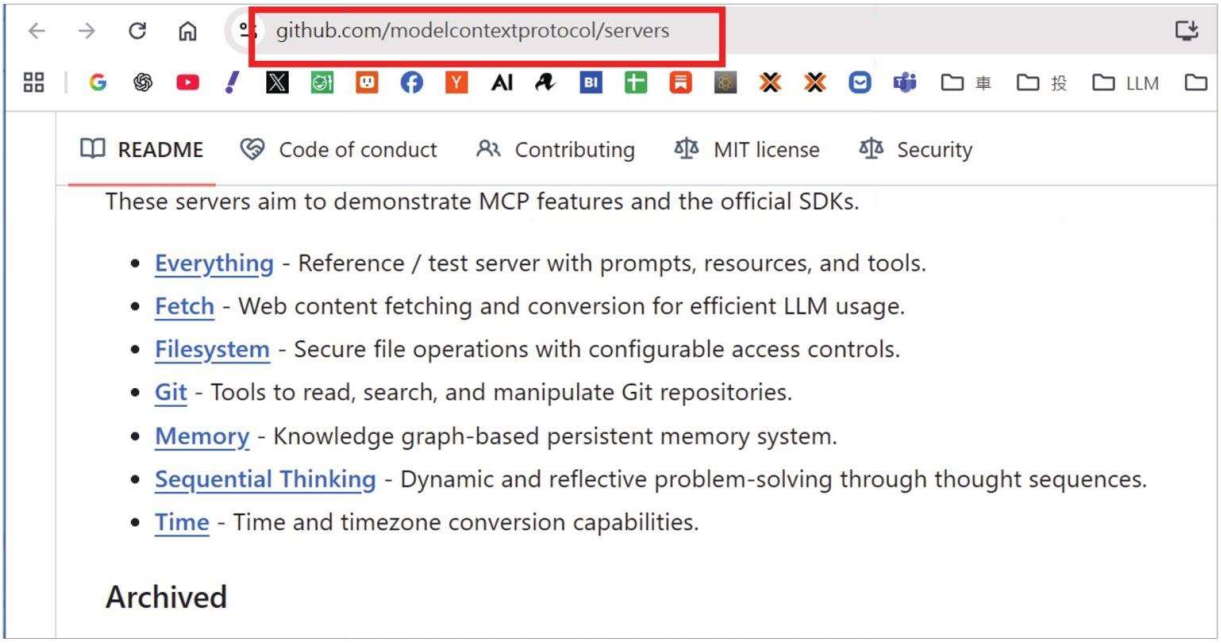
https://github.com/punkpeye/awesome-mcp-servers

https://mcpmarket.com/zh/leaderboards

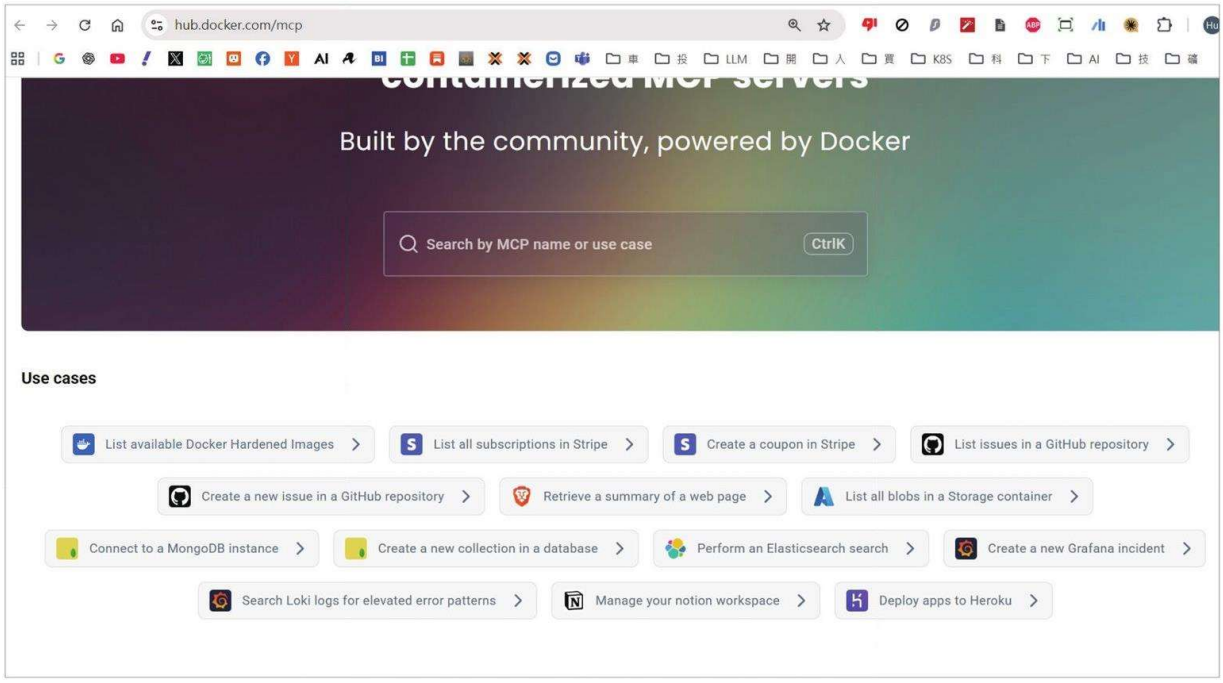
https://smithery.ai/



▲ 圖 13-17 mcp.so 是筆者每天都要來看的



▲ 圖 13-18 這邊是程式開發者的天堂



▲ 圖 13-19 原來 Docker Hub 也有提供 MCP Server

13.12 本章小結

這 20 個核心 MCP Server 加上 20 個熱門伺服器及官方推薦的 20 個 MCP Server，構成了一個完整的開發生態系統，涵蓋了從程式碼編寫到部署運維的各個環節。每個伺服器都有其專精領域，但真正的威力在於它們的組合使用。透過合理設定和使用這些 MCP Server，Claude Code 不再只是一個 CLI 工具，而是成為了超級厲害的八爪魚，它能夠理解需求，執行複雜的任務，並持續學習改進。這就是 MCP 的重要性 - 開發者能夠專注於創造性工作，而將重複性和技術性任務交給 AI 來處理。



14

Subagent 子代理協作系統

在前面的章節中，我們看到了 MCP Server 如何擴充 Claude Code 的能力，但有時候單一的 Claude Code 對話會變得太過複雜，上下文太多反而會讓效果變差。這時候就是 Subagent 派上用場的時候了。Subagent 讓你可以建立專門處理特定任務的 AI 助手，每個助手都有自己的專長和獨立的上下文空間，就像是組成一個專業分工的開發團隊。

上下文視窗管理是目前 LLM 系統中一大挑戰。太多資料不但放不進去，前後矛盾或無關的資料，更是影響輸出的品質。