



# 如何用AI建置 網站 (前台)

楊健宇

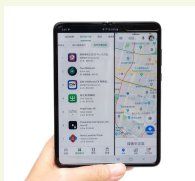
# 議程

- |    |                         |
|----|-------------------------|
| 00 | AI Coding前置作業           |
| 01 | 如何用AI建置商業網站(前台)         |
| 02 | 如何用AI建置商業網站(前台+資料庫)     |
| 03 | 如何設計商業網站(前台+控制項+資料庫)    |
| 04 | 如何設計商業網站(後台+Auth/OAuth) |
| 05 | 如何建立AI小助手(Agent Skills) |

# Web網站架構

Browser

Edge, Chrome, Safari, Firefox, App...



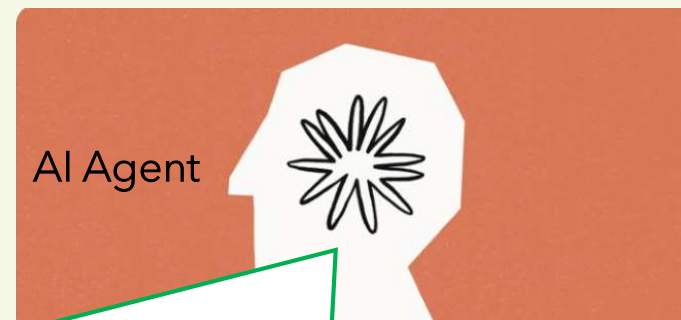
Web

PHP, ASP.NET Core...

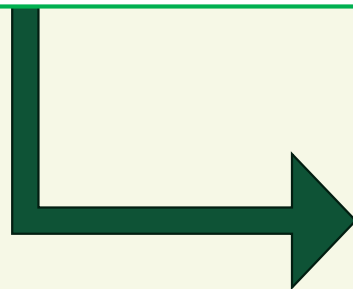


## 如何建置第一個Web應用程式?

# 用AI建立新網站



【AI Agent<提示詞>】建立一個.net core 8 MVC網站系統，專案名稱為MyNewWeb，建立基礎架構專案，提供一個預設首頁樣板。



# 感謝聆聽 Thank You

## 共創與合作

感謝您的時間與支持。我們以AI與團隊協作，啟動更好的可能。

## 保持連結

期待與您持續交流，共同推動創新與影響力。





# .NET Core

- .NET Core 是一個免費、開源、跨平台 (Windows, Linux, macOS) 的應用程式開發框架，由微軟和社群維護共同開發，用於建立高性能、可擴展的現代 Web 應用程式、微服務、雲端應用以及桌面和 IoT 應用



**.NET 10.0**

長期支援

下載 .NET SDK x64

SDK 版本 10.0.101, 已發行 2025年12月9日

[所有 .NET 10.0 下載](#) [.NET 10.0 執行階段](#) [所有 .NET 版本](#)

### .NET 10 的新增功能為何?

.NET 10 為現代化開發提供更快速、更安全且功能更強大的平台。透過對執行階段的主要升級以及對最新硬體的支援、讓使用 C# 14 進行編碼變得優雅並可延伸的精簡功能，您的應用程式可更快速且更有效率地執行。 .NET 10 讓安全驗證更加輕鬆且 API 甚至更快，並跨 Web 與行動裝置提供更順暢的體驗和更好的診斷。

[深入了解 →](#)

① 建議所有新產品開發都使用 [.NET 8 或更新版本](#)。這些較新的 .NET 版本是跨平台版本、支援更多應用程式類型，以及提供高效能。

<https://dotnet.microsoft.com/zh-tw/download>

# 快速安裝.Net SDK(Win10/11)

- 用Powershell安裝.NET SDK

```
winget install Microsoft.DotNet.SDK.<version number>
```

範例：

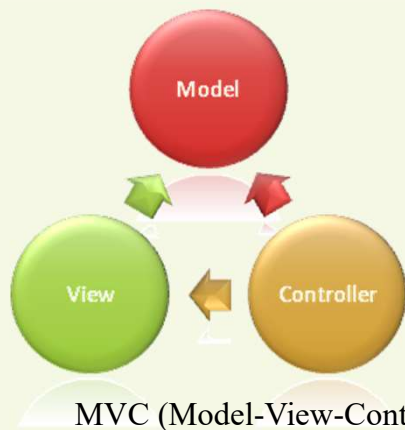
```
winget install Microsoft.DotNet.SDK.10
```

- 非Win10/11環境，至.Net SDK網站下載本機OS版本

<https://dotnet.microsoft.com/zh-tw/download>

# ASP.NET MVC 概觀

- MVC(模型 - 檢視 - 控制器)架構模式將一個應用程式劃分成三組主要元件。ASP.NET MVC 架構提供Web Form 模式替代方案。



MVC (Model-View-Controller) 是一種軟體設計模式，將應用程式分為三個互相關聯的原因，實踐關注點分離：

- 模型(Model)：處理資料邏輯和業務規則
- 視圖(View)：負責使用者介面和資料展示
- 控制器(Controller)：處理使用者輸入並協調模型和視圖

這種分離使得程式碼更易維護、擴展和測試，特別適合團隊協作開發。

ASP.NET MVC

## MVC 型 Web 應用程式的優點

ASP.NET MVC 架構具有下列優點：

- 該架構將應用程式分割成模型、檢視和控制器，可更輕鬆地管理複雜度。
- 該架構不使用檢視狀態或伺服器型表單。這讓 MVC 架構成為想要完全控制應用程式行為的開發人員的理想選擇。
- 該架構使用前端控制器模式，透過單一控制器處理 Web 應用程式要求。這可讓您設計支援豐富的路由基礎結構的應用程式。如需詳細資訊，請參閱[前端控制器](#)。
- 它對測試驅動開發 (TDD) 提供更好的支援。
- 它適用於大規模開發人員小組，和需要高度控制應用程式行為的 Web 設計工具小組所支援的 Web 應用程式。

## Web Forms 型 Web 應用程式的優點

以 Web Forms 為基礎的架構具有下列優點：

- 該架構支援透過 HTTP 保留狀態的事件模型，這有利於企業營運 Web 應用程式開發。以 Web Forms 為基礎的應用程式，提供了由數百個伺服器控制項支援的數十個事件。
- 它使用頁面控制器模式，將功能新增至個別頁面。如需詳細資訊，請參閱[頁面控制器](#)。
- 它使用檢視狀態或伺服器型表單，讓管理狀態資訊更容易執行。
- 它適用於想要利用大量可用於快速應用程式開發之元件的 Web 開發人員和設計工具小組。
- 一般而言，應用程式開發較不複雜，因為元件 (Page 類別、控制項等) 會緊密整合，而且通常需要比 MVC 模型更少的程式碼。



# 手動建立ASP.NET Core Web 應用程式

- 在本課程模組中，您將學會設定開發環境來建立及建置 ASP.NET Core Web 應用程式。然後完成下列步驟：
  - 準備工具
  - 檢閱 .NET SDK 中可用的 ASP.NET Core 預設專案範本。
  - 從範本建立 ASP.NET Core Web 應用程式專案。
  - 在本機執行您的 Web 應用程式，並在瀏覽器中檢視它。
  - 建置了第一個 ASP.NET Core Web 應用程式！

# 使用 .NET 和 C# 建立第一個 ASP.NET Core Web 應用程式(1/5)

- 準備工具
  - 安裝的 .NET SDK 最新版本(必須)
  - 安裝 Visual Studio Code 與 C# Dev Kit 擴充功能
- ASP.NET Core 預設專案範本
  - .NET SDK 包含內建 ASP.NET Core 範本，用於建立 ASP.NET Core 專案和檔案，常見預設專案範本包括：
    1. **web** 空的 ASP.NET Core：用於建立 ASP.NET Core 應用程式的空白專案範本。此範本中沒有任何範例內容。
    2. **blazor** Blazor Web 應用程式：用於建立 Blazor Web 應用程式的專案範本，其同時支援伺服器端轉譯和用戶端互動。此範本可用於具有豐富動態使用者介面 (UI) 的 Web 應用程式。
    3. **webapi** ASP.NET Core Web API：用於使用 ASP.NET Core 控制器或最小 API 建立 RESTful Web API 的專案範本，並針對 OpenAPI 和驗證具有選用的支援。
    4. **grpc** ASP.NET Core gRPC 服務：用於使用 ASP.NET Core 建立 gRPC 服務的專案範本。

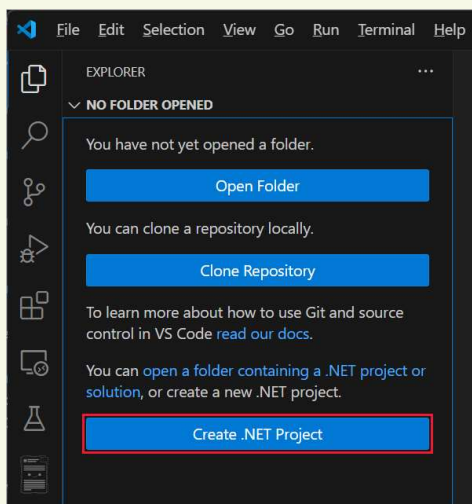
# 使用 .NET 和 C# 建立第一個 ASP.NET Core Web 應用程式(2/5)

建立名為 *MyWebApp* 的目錄中  
-o：輸出所在的目錄，如果目錄不存在，則 .NET CLI 會建立它

- 方式一：CLI建立新的專案

- `dotnet new web -o MyWebApp`

- 方式二：在 Visual Studio Code 中建立新專案



# 使用 .NET 和 C# 建立第一個 ASP.NET Core Web 應用程式(3/5)



The screenshot shows the Visual Studio IDE with a project named 'MyWebApp'. The left sidebar displays the 'Workspace' tree with the following structure:

- WORKSPACE
  - MyWebApp
    - bin
    - obj
    - Properties
    - appsettings.Development.json
    - appsettings.json
    - MyWebApp.csproj
    - Program.cs
    - workspace.slnx

The main editor window shows the 'Program.cs' file with the following code:

```
1 var builder = WebApplication.CreateBuilder(args);  
2 var app = builder.Build();  
3  
4 app.MapGet("/", () => "Hello World!");  
5  
6 app.Run();  
7
```

# 使用 .NET 和 C# 建立第一個 ASP.NET Core Web 應用程式(4/5)

- CLI建置並執行專案 `dotnet run --Project MyWebApp`

看起來很簡單吧!

但是...

如何新增控制(功能)項?

如何美工網頁?

如何增加資料庫?

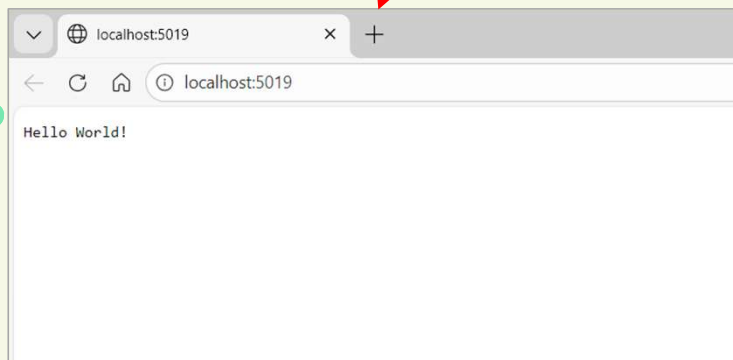
如何確保網站安全?

(過去)

Tools + Knowledge

`dotnet run --Project MyWebApp`

```
PS C:\Users\user\.vscode\workspace> dotnet run --project MyWebApp
使用來自 MyWebApp\Properties\launchSettings.json 的啟動設定...
正在建置...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5019
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\user\.vscode\workspace\MyWebApp
```



# 使用 .NET 和 C# 建立第一個 ASP.NET Core Web 應用程式(5/5)

- CLI建置並執行專案 `dotnet run --Project MyWebApp`

```
PS C:\Users\user\.vscode\workspace> dotnet run --project MyWebApp
使用來自 MyWebApp\Properties\launchSettings.json 的啟動設定...
正在建置...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5019
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\user\.vscode\workspace\MyWebApp
```

看起來很簡單吧!

但是...

如何新增控制(功能)項?

如何美工網頁?

如何增加資料庫?

如何確保網站安全?

(現在)

Vibe Coding

