



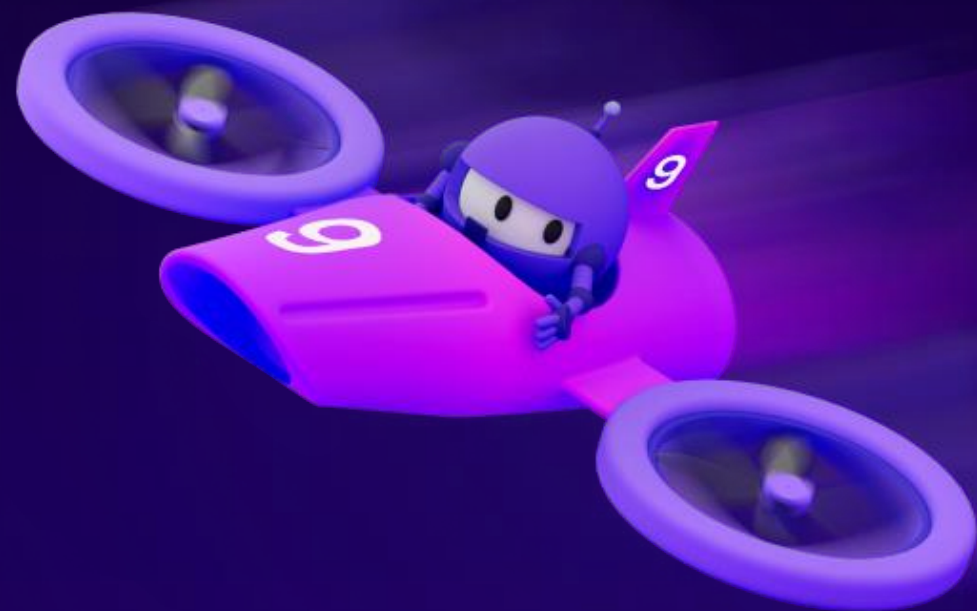
STUDY4

為 學 習 而 生

全面掌握 Swagger 文件的「念」能力：

讓你的 API 無懈可擊

泉富



Agenda

- 名詞定義
- Swashbuckle 說明與設定
- 標記 API 資訊
- 使用 Open API Spec 生成 Client 端程式碼
- 其他設定

01

名詞定義

Swagger

Swagger 是 SmartBear Software 開發出的 REST API 的商用軟體，專為設計、建構、記錄和使用 RESTful API 而設計。

- 01** 生成 OpenAPI 規格文件：將的 API 結構轉換為機器可讀的 API 規範 (JSON / YAML)
- 02** 生成互動式 API 文件：將 OpenAPI 定義渲染為互動式文件 Swagger UI，讓使用者能直接在瀏覽器中嘗試 API 呼叫。
- 03** 生成程式碼：透過 Swagger Codegen / [openapi-generator](#) 之類的工具，將 OpenAPI 規格生成 Server 端程式或 Client 程式，工具通常支援多種程式語言

OpenAPI

是一種專為 REST API 設計的 API 描述格式，使用 JSON或YAML 編寫。

內容包含：

- 01 API 路徑：例如 /book。
- 02 操作：例如 GET /book、POST /book。
- 03 操作參數：每個操作的輸入與輸出。
- 04 驗證方式 (Authentication methods) 。
- 05 聯絡資訊、授權條款以及其他相關資訊。

完整的 OpenAPI 規格說明可參考 GitHub 上的 [OpenAPI 3.1 Specification](#)

OpenAPI 與 Swagger

Swagger 專案於 2015 年已捐贈給 OpenAPI 方案，此後一直將其稱為 OpenAPI。
這兩個名稱會交替使用。

明確定義兩個名詞：

- OpenAPI 是指「規格」。
- Swagger 是使用 OpenAPI 規格的工具。例如，OpenAPIGenerator 和 SwaggerUI。

節錄自微軟官網：<https://learn.microsoft.com/zh-tw/aspnet/core/tutorials/web-api-help-pages-using-swagger?view=aspnetcore-8.0#openapi-vs-swagger>

02

Swashbuckle 說明與設定

Swashbuckle.AspNetCore 相依套件

Swashbuckle.AspNetCore.SwaggerGen

功能：幫助生成 OpenAPI 規範 (Swagger 文件物件)

用途：自動掃描程式碼中的控制器、路由、模型等，並生成對應的 OpenAPI 規範文件

使用時機：需要調整 **OpenAPI** 規格相關內容，例如標題、版本、描述

Swashbuckle.AspNetCore.Swagger

功能：提供處理 OpenAPI/Swagger 規範的核心元件，將 API 規範轉換成 JSON/YAML 格式的進入點

用途：作為 API 規範的「最終輸出點」，供其他工具（如 Swagger UI 或外部系統）使用

使用時機：需要更改 **OpenAPI** 規格網址進入點時

Swashbuckle.AspNetCore.SwaggerUI

功能：提供互動式 API 介面

用途：將 OpenAPI 規格轉換為互動式操作介面，用於瀏覽和測試 API。

使用時機：設定跟 **Swagger** 畫面有關的資訊，包含讀取 **OpenAPI** 網址、介面呈現資訊與模式

註冊 Swagger 服務、Middleware

```
var builder = WebApplication.CreateBuilder(args);  
builder.Services.AddSwaggerGen();
```

```
var app = builder.Build();  
app.UseSwagger();  
app.UseSwaggerUI();
```

將設定搬到擴充方法內

```
public static class SwaggerSettings
{
    /// <summary>
    /// Swagger 相關註冊內容
    /// </summary>
    /// <param name="services"></param>
    public static void AddSwaggerSettings(this IServiceCollection services)
    {
        services.AddSwaggerGen();
    }

    /// <summary>
    /// Swagger 中介軟體設定
    /// </summary>
    public static void UseSwaggerSettings(this WebApplication app)
    {
        app.UseSwagger();
        app.UseSwaggerUI();
    }
}
```

設定：標題、版本、描述

```
services.AddSwaggerGen(setupAction =>
{
    setupAction.SwaggerDoc("v1", // OpenAPI 文件名稱，預設為 v1，可自行設定，但要將其他參數調整一致
        new OpenApiInfo
        {
            Title = "Dotnet conf 2024 範例專案",
            Version = "v1",
            Description = "Dotnet conf 2024 Taiwan Swagger 範例",
            TermsOfService = new Uri("https://dotnetconf.study4.tw/Code-Of-Conduct"),
            Contact = new OpenApiContact
            {
                Name = "Study4tw",
                Url = new Uri("https://www.facebook.com/groups/study4tw/")
            },
            License = new OpenApiLicense
            {
                Name = "MIT License",
                Url = new Uri("https://opensource.org/licenses/MIT")
            },
        },
    });
});
```

設定：標題、版本、描述



Dotnet conf 2024 範例專案 v1 OAS 3.0

[/api/v1/swagger.json](#)

Dotnet conf 2024 Taiwan Swagger 範例

[Terms of service](#)

[Study4tw - Website](#)

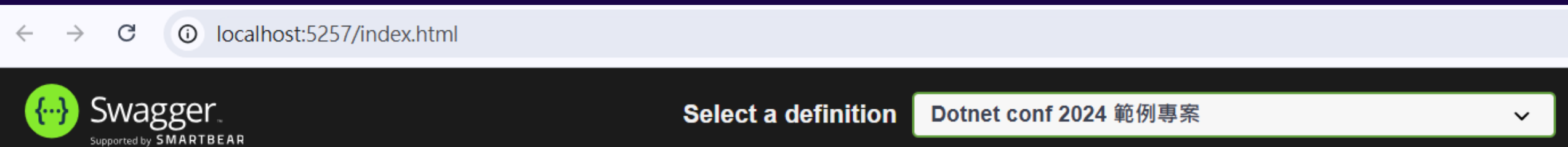
[MIT License](#)

設定 Swagger 畫面進入點

```
app.UseSwagger(setupAction =>
{
    // Open API 規格進入點，如果不需要異動，可以不用設定
    setupAction.RouteTemplate = "swagger/{documentName}/swagger.json";
});

app.UseSwaggerUI(setupAction =>
{
    // UI 讀取 OpenAPI 規格路徑
    setupAction.SwaggerEndpoint("/swagger/v1/swagger.json", "Dotnet conf 2024 範例專案");

    // UI 進入點設定為根目錄
    setupAction.RoutePrefix = string.Empty;
});
```



03

標記 API 資訊

類別成員的資訊標記

<summary>：成員的描述

<param name="name">：參數的名稱

<returns>：描述傳回值

<remarks>：補充 **summary** 的資訊

```
/// <summary>  
/// 根據 ID 查詢作者  
/// </summary>  
/// <param name="id">作者 ID</param>  
/// <returns>作者詳細資訊</returns>  
///  
///  
/// <remarks>  
/// 此方法會根據提供的作者 ID 查詢對應的作者詳細資訊  
/// </remarks>
```

更多 XML 標籤可參考：[建議用於 C# 文件註解的 XML 標記](#)

模型的資訊標記

<summary>：描述類型，或是成員

<example>：方法參數的名稱

[Required]：屬性為必填

[MinLength]：長度限制(最小值)

[MaxLength]：長度限制(最大值)

```
public class Book
{
    /// <summary>
    /// 書籍描述
    /// </summary>
    /// <example>一本描述你所不知道的 Swashbuckle 設定資訊
    /// 那些你知道、你不知道的小技巧，都在這裡了
    /// </example>
    [Required]
    [MinLength(10)]
    [MaxLength(150)]
    public string Description { get; set; }
}
```

更多 XML 標籤可參考：[建議用於 C# 文件註解的 XML 標記](#)

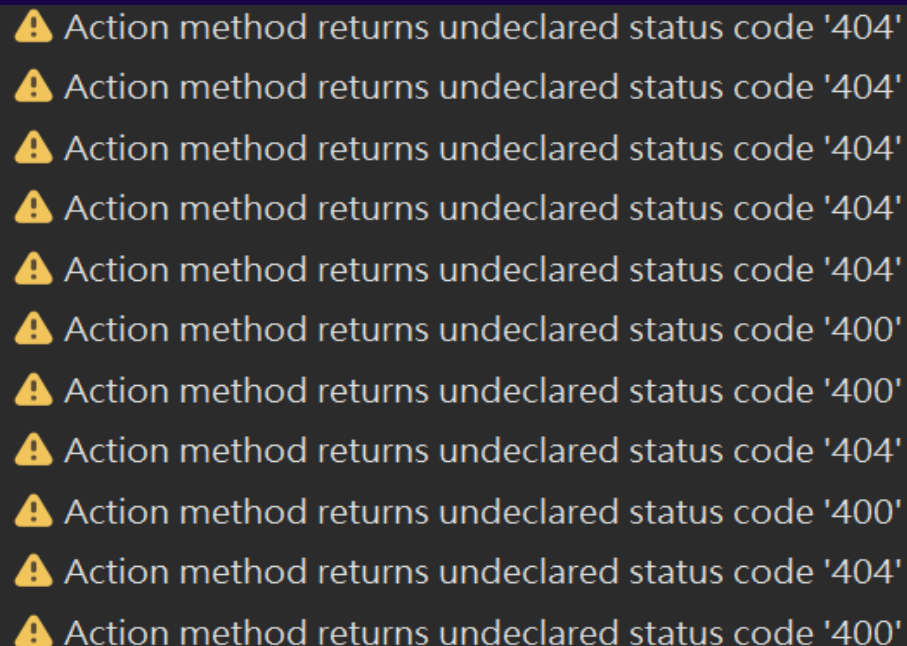
啟用 Open API 分析器

- 從 **ASP.NET Core 3.0** 開始，分析器已內建於 **.NET Core SDK**

- 在專案設定檔案開啟 **IncludeOpenAPIAnalyzers** 屬性

```
<IncludeOpenAPIAnalyzers>true</IncludeOpenAPIAnalyzers>
```

PS: 這個屬性沒有 IntelliSense



⚠ Action method returns undeclared status code '404'

⚠ Action method returns undeclared status code '404'

⚠ Action method returns undeclared status code '404'

⚠ Action method returns undeclared status code '404'

⚠ Action method returns undeclared status code '404'

⚠ Action method returns undeclared status code '400'

⚠ Action method returns undeclared status code '400'

⚠ Action method returns undeclared status code '404'

⚠ Action method returns undeclared status code '400'

⚠ Action method returns undeclared status code '404'

⚠ Action method returns undeclared status code '400'

⚠ Action method returns undeclared status code '400'

ProducesResponseType 屬性

用於描述特定動作方法可能產生的 HTTP 回應「型別」和「狀態碼」。

```
[HttpGet("{id}")]
[ProducesResponseType(StatusCodes.Status200OK, Type = typeof(Item))]
[ProducesResponseType(StatusCodes.Status404NotFound)]
public IActionResult GetItem(int id)
{
    var item = _repo.GetById(id);
    if (item == null)
    {
        return NotFound(); //API 有機會回傳 404 狀態
    }

    return Ok(item);
}
```

限定 API 輸入格式、輸出格式

[`Produces("application/json")`]

1. 限制**回應格式** (Response Format)，強制 Controller 或 Action 回傳指定的格式，例如 JSON。
2. 即使已啟用了其他格式器 (例如 XML)，且 Client 端的 Accept 也明確標記特定格式，API 也只會回傳指定格式

[`Consumes("application/json")`]

1. 限制**請求內容格式** (Request Format)，強制動作只接受特定格式的請求資料，例如 JSON。
2. 如果 Client 端的 Content-Type 未設為指定的格式，請求將被拒絕。

04

使用 Open API Spec 生成 Client 端程式碼



Demo:

使用 Visual Studio 連線參考



Demo:

使用 NSwagStudio

注意 API 是否有標記 Name 資訊！！

```
[HttpGet(Name = "GetAllAuthors")]
```

```
[HttpGet("{id:int}/books", Name = "GetBooksByAuthor")]
```

Name 參數設定值，最終會被 Client 端程式作為方法名稱使用
如果沒有特別設定，程式生成的方法名稱會變得不可控，切記！！

06

其他設定

其他跟 UI 相關的設定

- `DisplayRequestDuration` : 顯示請求時間。
- `DefaultModelRendering` : 預設顯示 `Model` / `Example`
- `EnableDeepLinking` : 啟用深度連結 (錨點連結)
- `EnableFilter` : 啟用篩選框
- `DefaultModelExpandDepth` : 模型展開深度, `0` 表示不展開, 預設為 `1`
- `DocExpansion` : 文件展開設定

關於我



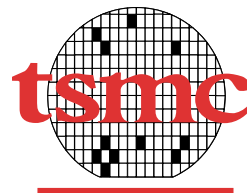
github.com/imall/DotnetConf2024

泉富

一個喜歡從零開始學習的工程師，寫過前端、寫過後端
深信基礎知識遠比追求新技術重要，於是修了不少基礎課程
目前任職於好房網

- ISO 27001：2022資訊安全管理系統主導稽核員訓練課程
- Amos - 暴力網頁新手入門班 (暴力班)：第 7~19 屆
- 五倍紅寶石 Kuro - Vue.js 3.x 與 Pinia 前端開發實戰
- SkillTree Bill 叔 - 輕鬆學會物件導向
- SkillTree Bill 叔 - 決戰設計模式
- 保哥 - ASP.NET Core 6 開發實戰

特別感謝



以及各位參與活動的各位



Thank you

