多奇·教育訓練

ASP.NET Core 5 開發實戰

應用開發篇 (MVC)

多奇數位創意有限公司

技術總監 黃保翕（Will 保哥）

https://blog.miniasp.com

ASP.NET Core MVC: Summary

ASP.NET Core MVC 重點摘要

從 ASP.NET MVC 升級 ASP.NET Core MVC

- Migrate from ASP.NET MVC to ASP.NET Core MVC
  - 移轉 `web.config` 內的**應用程式設定**與**連接字串**
  - 設定 `Startup.ConfigureServices` 與 `Startup.Configure`
  - `ActionResult` 全部轉成 `IActionResult`
  - 準備 `Views/_ViewImports.cshtml` 檔案 (Tag Helpers)
  - 改寫 `@Styles.Render` 與 `@Scripts.Render` 為 `<link>` 與 `<script>`
  - Migrate Authentication and Identity to ASP.NET Core
  - Bundle and minify static assets in ASP.NET Core

- A Step-by-Step Guide to Migrating a Project from ASP.NET MVC to ASP.NET Core

ASP.NET Core MVC 新增的功能

- 相依性注入 (**Dependency Injection**)
- 標籤協助程式 (**Tag Helpers**)
  - ASP.NET Core 中的 Tag Helpers / 內建的 Tag Helpers
- 檢視元件 (**View Components**)
  - 與 Partial View 相當類似
  - 不過除了 View 之外卻多了部分控制邏輯 (Component)
  - 除了內建的 View 以外，在使用元件的地方還可以覆寫！
  - 除了透過 `@await Component.InvokeAsync()` 執行之外，現在還可以透過 **Tag Helper** 載入該元件！
  - 也可以從 Controller 直接呼叫 View Component

MVC 與 WebAPI 的 Routing 差異 (1)

- MVC 專案使用 `endpoints.MapControllerRoute()` 設定預設路由

```
app.UseEndpoints(endpoints =>
{
    endpoints.MapControllerRoute(
        name: "default",
        pattern: "{controller=Home}/{action=Index}/{id?}");
});
```

- 控制器繼承 Controller 抽象類別

```
public class HomeController : Controller
{
    public IActionResult Index()
    {
        return View();
    }
}
```

MVC 與 WebAPI 的 Routing 差異 (2)

- Web API 專案使用 `endpoints.MapControllers()` 不包含任何預設路由

```
app.UseEndpoints(endpoints =>
{
    endpoints.MapControllers();
});
```

- 控制器繼承 ControllerBase 抽象類別

```
[ApiController]
[Route("api/[controller]")]
public class WeatherForecastController : ControllerBase
{
    [HttpGet]
    public IEnumerable<WeatherForecast> Get()
    {
        return [];
    }
}
```

擴充驗證屬性的改變

- 擴充程式碼產生器的實體模型(Entity Model)
  - .NET Framework
    - 使用 `[MetadataType]` 屬性
  - .NET Core / .NET 5
    - 使用 `[ModelMetadataType]` 屬性
      - 需引用 `Microsoft.AspNetCore.MVC` 命名空間
- 範例程式
  - https://github.com/doggy8088/ASPNETCore5ModelMetadataType/blob/main/WebApplication4/Models/Course.Partial.cs#L11

在 ASP.NET Core MVC 的 Views 注入服務

- 在 ASP.NET MVC 的 View 中注入 **IConfiguration** 物件

```
@inject IConfiguration Configuration
```

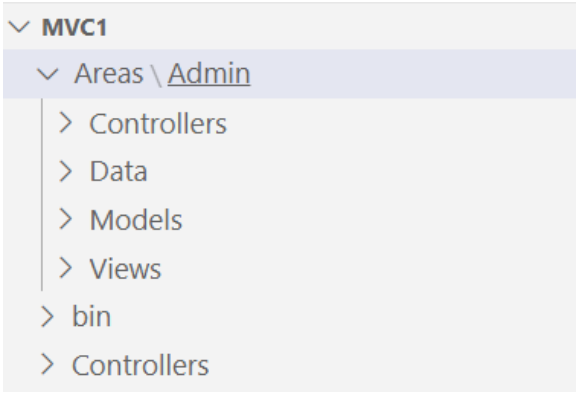
- 在 ASP.NET MVC 的 View 中注入 **IOptions<T>** 物件

```
@inject IOptions<AppSetting> OptionsAccessor
```

```
@inject IOptionsSnapshot<AppSetting> OptionsAccessor
```

建立 Area 目錄結構

- 更新 dotnet-aspnet-codegenerator 全域工具
  - `dotnet tool update -g dotnet-aspnet-codegenerator`
- 安裝 Microsoft.VisualStudio.Web.CodeGeneration.Design 套件
  - `dotnet add package Microsoft.VisualStudio.Web.CodeGeneration.Design`
- 建立新的 `Area` 區域 (以 `Admin` 為例)
  - `dotnet aspnet-codegenerator area Admin`
- 產生的目錄結構 (空目錄)



替 Area 設定路由

- Startup.cs 內的 Configure() 中
  - 在 app.UseEndpoints() 內加入 Area 路由設定

```
endpoints.MapControllerRoute(
    name : "areas",
    pattern : "{area:exists}/{controller=Home}/{action=Index}/{id?}"
);

endpoints.MapControllerRoute(
    name : "default",
    pattern : "{controller=Home}/{action=Index}/{id?}"
);
```

在 Area 的控制器加入 [Area] 屬性

- 在 Area 內建立的控制器需要加入 [Area] 屬性設定所屬 Area

```
[Area("Admin")]
public class HomeController : Controller
{
    public IActionResult Index()
    {
        return View();
    }
}
```

所有 ASP.NET Core 內建的 Tag Helpers

- Anchor Tag Helper
- Cache Tag Helper
- Component Tag Helper
- Distributed Cache Tag Helper
- Environment Tag Helper
- Form Tag Helper
- Form Action Tag Helper
- Image Tag Helper
- Input Tag Helper
- Label Tag Helper
- Link Tag Helper
- Partial Tag Helper
- Script Tag Helper
- Select Tag Helper
- Textarea Tag Helper
- Validation Message Tag Helper
- Validation Summary Tag Helper

超連結 Tag Helpers 用法

- `<a asp-controller="Speaker" asp-action="Index">All Speakers</a>`
- `<a asp-controller="Speaker" asp-action="Detail" asp-route-id="@Model.SpeakerId">SpeakerId: @Model.SpeakerId</a>`
- `<a asp-route="speakerevals">Speaker Evaluations</a>`
- `<a asp-route="speakerevalscurrent" asp-all-route-data="parms">Speaker Evaluations</a>`
- `<a asp-controller="Speaker" asp-action="Evaluations" asp-fragment="SpeakerEvaluations">Speaker Evaluations</a>`
- `<a asp-area="Blogs" asp-controller="Home" asp-action="AboutBlog">About Blog</a>`
- `<a asp-protocol="https" asp-controller="Home" asp-action="About">About</a>`
- `<a asp-protocol="https" asp-host="microsoft.com" asp-controller="Home" asp-action="About">About</a>`
- `<a asp-page="/Attendee">All Attendees</a>`
- `<a asp-page="/Attendee" asp-page-handler="Profile" asp-route-attendeeid="12">Attendee Profile</a>`

表單 Tag Helpers 起手式

- 宣告表單

```
<form asp-controller="Demo" asp-action="Register" method="post">
    <!-- Input and Submit elements -->
</form>
```

- 產生的表單 HTML

```
<form method="post" action="/Demo/Register">
    <!-- Input and Submit elements -->
    <input name="__RequestVerificationToken" type="hidden" value="<removed for brevity>" />
</form>
```

表單 Tag Helpers 的更多用法

- 具名路由

```
<form asp-route="register" method="post">
    <!-- Input and Submit elements -->
</form>
```

- 加入額外的路由參數

```
<form asp-controller="Account" asp-action="Login"
     asp-route-returnurl="@ViewData["ReturnUrl"]"
     method="post" class="form-horizontal" role="form">
```

表單欄位的 Tag Helpers 用法

- `<input asp-for="Property1.Property2" />`
  - 等同於 `@Html.EditorFor(m => m.Property1.Property2)`
- `<label asp-for="Age"></label>`
- `<textarea asp-for="Description"></textarea>`

- `<select asp-for="Country" asp-items="Model.Countries">`
  `</select>`
- `<select asp-for="CountryCodes" asp-items="Model.Countries">`
  `</select>`
  - 如果 CountryCodes 是個 IEnumerable<T> 型別，預設就會變成多選的下拉選單
- `<select asp-for="EnumCountry"`
  `asp-items="Html.GetEnumSelectList<CountryEnum>()">`
- `</select>`

表單欄位的 Tag Helpers 的更多用法

- `<span asp-validation-for="Email"></span>`

- `<div asp-validation-summary></div>`

快取 Tag Helpers 用法

- `<cache>@DateTime.Now</cache>` (預設快取 20 分鐘)

- `<cache enabled="false">@DateTime.Now</cache>`

- ```
  <distributed-cache name="my-distributed-cache-unique-key-101">
      Time Inside Cache Tag Helper: @DateTime.Now
  </distributed-cache>
  ```
  ```

快取 Tag Helpers 的更多用法

- ```
  <cache expires-on="@new DateTime(2025,1,29,17,02,0)">
      Current Time Inside Cache Tag Helper: @DateTime.Now
  </cache>
  ```

- ```
  <cache expires-after="@TimeSpan.FromSeconds(120)">
      Current Time Inside Cache Tag Helper: @DateTime.Now
  </cache>
  ```

- ```
  <cache expires-sliding="@TimeSpan.FromSeconds(60)">
      Current Time Inside Cache Tag Helper: @DateTime.Now
  </cache>
  ```
  ```

環境設定 Tag Helpers 用法

- ```
  <environment names="Staging,Production">
      EnvironmentName is Staging or Production
  </environment>
  ```

- ```
  <environment include="Staging,Production">
      <strong>EnvironmentName is Staging or Production</strong>
  </environment>
  ```

- ```
  <environment exclude="Development">
      EnvironmentName is Staging or Production
  </environment>
  ```
  ```

部分檢視 (PartialView) 的 Tag Helpers

- `<partial name="Shared/_ProductPartial.cshtml" for="Product" />`
- `<partial name="_ProductPartial" model='new Product { Number = 1, Name = "Test product", Description = "This is a test" }' />`
- `<partial name="_ProductViewDataPartial" for="Product" view-data="ViewData" />`

靜態資源的 Tag Helpers 用法

- 避免靜態資源被 cache 住

- `<img src="~/images/asplogo.png" asp-append-version="true" />`

- `<script src="~/js/site.js" asp-append-version="true"></script>`

- `<link href="~/css/site.css" asp-append-version="true" />`

```
<img src="/images/asplogo.png?v=BEwNS6fkSmH63J_QehH8ukXzzzgYThQtaUzD5F72mM8">
<script src="/js/site.js?v=4q1jwFhaPaZgr8WAUSrux6hAuh0XDg9kPS3xIVq36I0"></script>
<link href="/css/site.css?v=S2ihmzMFFc3FWmBWsR-NiddZWa8kbyaQYBx2FDkIoHs">
```

檢視元件 (View Components) 簡介

- 可以將**檢視頁面**與**商業邏輯**封裝到一個**元件**內！
  - **檢視頁面**
    - 呈現面的邏輯
  - **商業邏輯**
    - 原本應該出現在 **控制器** 的 **部分邏輯** 抽離成**元件**的一部份
    - 應用情境：登入框、購物車、標籤雲、動態導覽選單、...
- 使用 View Component 時
  - 可以覆寫原有 View Component 的檢視頁面 (Views)
- 文件說明 / 範例程式

建立 ViewComponent 類別

類別名稱必須使用 ViewComponent 結尾

必須繼承 ViewComponent

```
public class PriorityListViewComponent : ViewComponent
{
    private readonly ToDoContext db;

    public PriorityListViewComponent(ToDoContext context)
    {
        db = context;
    }

    public async Task<IViewComponentResult> InvokeAsync(int maxPriority, bool isDone)
    {
        var items = await db.ToDo.Where(x =>
                x.IsDone == isDone && x.Priority <= maxPriority).ToListAsync();
        return View(items);
    }
}
```

建立 ViewComponent 類別

```
public class PriorityListViewComponent : ViewComponent
{
    private readonly ToDoContext db;

    public PriorityListViewComponent(ToDoContext context)
    {
        db = context;
    }

    public async Task<IViewComponentResult> InvokeAsync(int maxPriority, bool isDone)
    {
        var items = await db.ToDo.Where(x => x.IsDone == isDone &&
                                x.Priority <= maxPriority).ToListAsync();
        return View(items);
    }
}
```

可以使用相依注入

建立 ViewComponent 類別

```
public class PriorityListViewComponent : ViewComponent
{
    private readonly ToDoContext db;

    public PriorityListViewComponent(ToDoContext context)
    {
        db = context;
    }

    public async Task<IViewComponentResult> InvokeAsync(int maxPriority, bool isDone)
    {
        var items = await db.ToDo.Where(x => x.IsDone == isDone &&
                                x.Priority <= maxPriority).ToListAsync();
        return View(items);
    }
}
```

建立 InvokeAsync 方法，可以自行撰寫更多邏輯，並回傳 View

建立 ViewComponent 的畫面

- 預設 View 路徑為
  - Views/Shared/Components/{View Component Name}/{View Name}.cshtml
  - 預設使用的 View 檔名為 **Default.cshtml**
- 範例
  - Views/Shared/Components/PriorityList/**Default.cshtml**

- View Component 搜尋 View 的路徑 (依照搜尋優先順序)
  - /Views/{Controller Name}/Components/{View Component Name}/{View Name}
  - /Views/Shared/Components/{View Component Name}/{View Name}
  - /Pages/Shared/Components/{View Component Name}/{View Name}

ViewComponent 使用方法

- 方法 1: 使用 Component.InvokeAsync

```
@await Component.InvokeAsync("PriorityList",
    new { maxPriority = 2, isDone = false }
)
```

- 方法 2: 使用 View Components Tag Helper
  - 需先使用 @addTagHelper 註冊包含 View Components 的組件

```
@addTagHelper *, MyWebApp
```

  - 使用 <vc:xxxx></vc:xxxx> 顯示內容

```
<vc:priority-list max-priority="999" is-done="false">
</vc:priority-list>
```

聯絡資訊

The Will Will Web

網路世界的學習心得與技術分享

http://blog.miniasp.com/

Facebook

Will 保哥的技術交流中心

http://www.facebook.com/will.fans

Twitter

https://twitter.com/Will_Huang

多奇·教育訓練

THANK YOU!

Q&A