

ASP.NET Core **Practices 03**

Chapter 3: ASP.NET Core MVC

Part 1: Create ASP.NET Core MVC project

Bài 3.1 Tạo ứng dụng ASP.NET Core Web App MVC

Tạo mới project ASP.NET Core, sử dụng template: **ASP.NET Core Web App (Model-View-Controller)**.

Tham khảo: <https://learn.microsoft.com/vi-vn/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-8.0&tabs=visual-studio>

Part 2: Add Controller

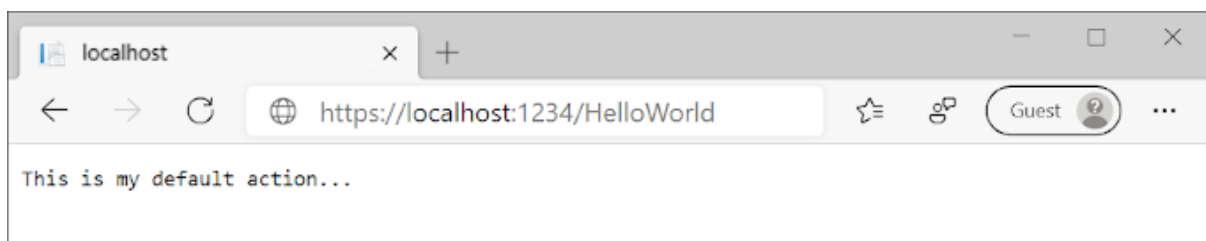
Bài 3.2 Tạo Controller

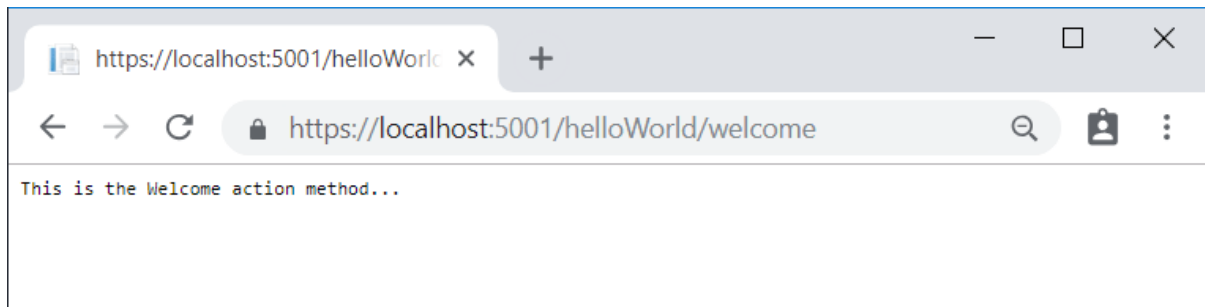
Trong thư mục Controllers, tạo mới một Controller có tên: HelloWorldController, gồm 2 actions (methods).

- **Index:** trả về kiểu string: `"This is my default action..."`
- **Welcome:** trả về kiểu string: `"This is the Welcome action method..."`

Sau đó, chạy ứng dụng và xem kết quả.

Tham khảo: <https://learn.microsoft.com/vi-vn/aspnet/core/tutorials/first-mvc-app/adding-controller?view=aspnetcore-8.0&tabs=visual-studio>





Bài 3.3 Tạo action với các tham số. Truyền giá trị cho tham số bằng Query String trên URL

Lưu ý: Các tham số của một action trong Controller có thể được truyền vào bằng cách:

- Query String in URL.
- Route Parameter in URL.

Yêu cầu:

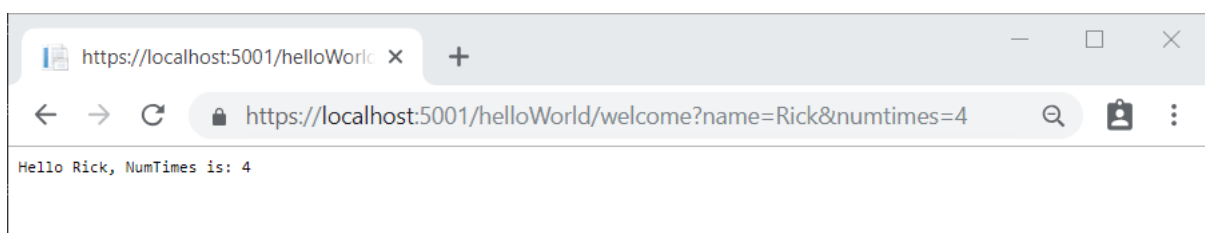
Hãy thay đổi code của action Welcome, bổ sung thêm 2 tham số name và numTimes như bên dưới.

```
public string Welcome(string name, int numTimes = 1)
{
    return HtmlEncoder.Default.Encode($"Hello {name}, NumTimes is: {numTimes}");
}
```

Sau đó, truy cập vào action này, và truyền giá trị cho 2 tham số này bằng cách sử dụng Query String (phần đằng sau dấu ?). Ví dụ:

`https://localhost:{PORT}/HelloWorld/Welcome?name=Rick&numtimes=4`

Tham khảo: <https://learn.microsoft.com/vi-vn/aspnet/core/tutorials/first-mvc-app/adding-controller?view=aspnetcore-8.0&tabs=visual-studio>



Bài 3.4 Tạo action với các tham số. Truyền giá trị cho tham số bằng Route Parameter trên URL

Hãy thay đổi code của action Welcome, bổ sung thêm 2 tham số name và ID như bên dưới.

```
public string Welcome(string name, int ID = 1)
```

```
{
    return HtmlEncoder.Default.Encode($"Hello {name}, ID: {ID}");
}
```

Sau đó, truy cập vào action này, và truyền giá trị cho tham số ID bằng Route Parameter trong URL và tham số name thông qua Query String. Ví dụ:

`https://localhost:{PORT}/HelloWorld/Welcome/3?name=Rick`

Lưu ý:

URL trong mô hình MVC thường có cấu trúc: `/[Controller]/[ActionName]/[Parameters]`

Định dạng này (route format) được quy định trong file Program.cs

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

Tham khảo: <https://learn.microsoft.com/vi-vn/aspnet/core/tutorials/first-mvc-app/adding-controller?view=aspnetcore-8.0&tabs=visual-studio>

Part 3: Add View

Bài 3.5 Tạo View cho action Index trong Controller

View là thành phần cung cấp giao diện cho người dùng. Nó chịu trách nhiệm hiển thị dữ liệu và cho phép người dùng tương tác với hệ thống.

Trong ứng dụng ASP.NET Core MVC, các View được tạo dựa trên Razor (Razor-based view templates):

- Có phần mở rộng là .cshtml.
- Cho phép tạo ra HTML sử dụng C#.

Yêu cầu: Hãy tạo một View cho action **Index** trong Controller **HelloWorldController**.

Bước 1: Trong HelloWorldController, action Index trả về IActionResult (thay vì string) với đoạn code bên dưới:

```
public IActionResult Index()
{
    return View();
}
```

Bước 2: Tạo View bằng cách: Nhấp chuột phải vào View() → Add View.

Visual Studio sẽ tạo một file **Index.cshtml** trong thư mục Views/HelloWorld.

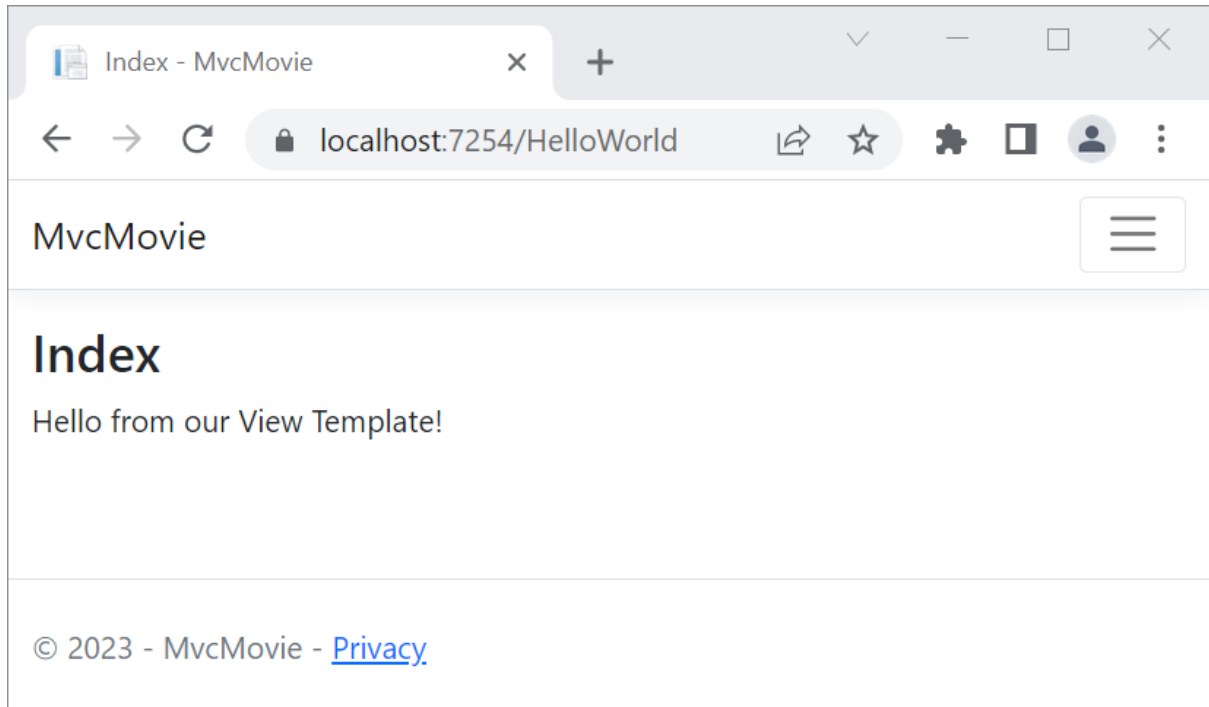
Thay đổi nội dung file Index.cshtml với đoạn code sau:

```
@{
    ViewData["Title"] = "Index";
}

<h2>Index</h2>

<p>Hello from our View Template!</p>
```

Truy cập ứng dụng qua URL: <https://localhost:{PORT}/HelloWorld> và giải thích kết quả.



Tham khảo: <https://learn.microsoft.com/vi-vn/aspnet/core/tutorials/first-mvc-app/adding-view?view=aspnetcore-8.0&tabs=visual-studio>

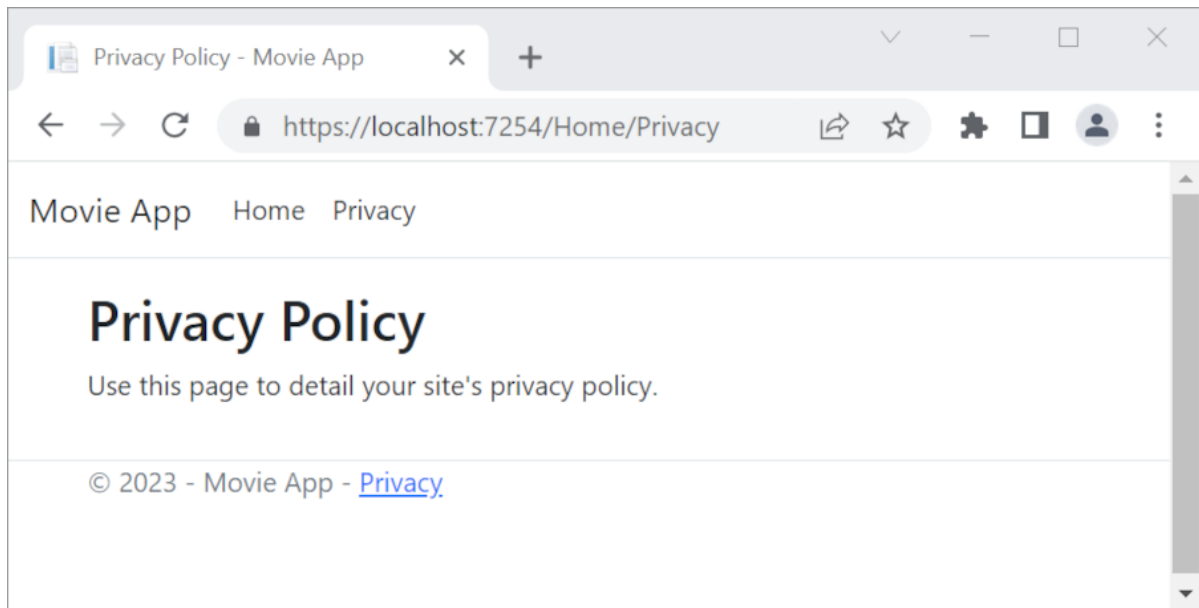
Bài 3.6 Thay đổi Layout

Yêu cầu:

Chỉnh sửa file `Views/Shared/_Layout.cshtml` với các nội dung sau (xem hình kết quả để rõ hơn):

- Thay đổi tiêu đề (Title) thành: Movie App.
- Thay đổi Footer thành: © 2023 Movie App.
- Thay đổi link trên thanh Menu thành Movie App và link đến Controller có tên “Movies” (lưu ý: hiện tại controller Movies này chưa được tạo).

Tham khảo: <https://learn.microsoft.com/vi-vn/aspnet/core/tutorials/first-mvc-app/adding-view?view=aspnetcore-8.0&tabs=visual-studio>



Bài 3.7 Thay đổi file Index.cshtml

Yêu cầu:

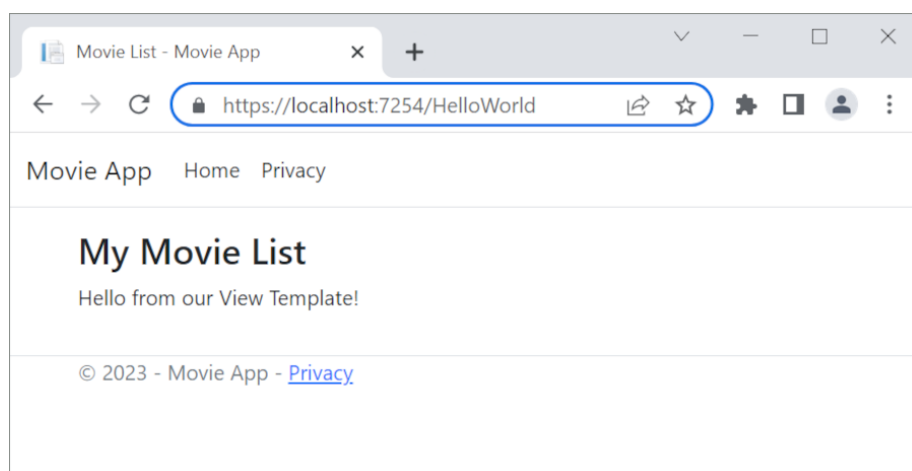
Thay đổi file Views/HelloWorld/Index.cshtml với nội dung bên dưới, sau đó chạy ứng dụng và giải thích kết quả.

Lưu ý: Có thể cần nhấn **Ctrl + F5** để refresh lại trang web từ server (nếu thấy nội dung mới không áp dụng vì có thể là bị cache ở trình duyệt).

```
@{
    ViewData["Title"] = "Movie List";
}

<h2>My Movie List</h2>

<p>Hello from our View Template!</p>
```



Bài 3.8 Truyền dữ liệu từ Controller vào View thông qua ViewData dictionary

Yêu cầu:

Thay đổi action Welcome trong controller HelloWorldController với đoạn code sau:

```
using Microsoft.AspNetCore.Mvc;
using System.Text.Encodings.Web;

namespace MvcMovie.Controllers;

public class HelloWorldController : Controller
{
    public IActionResult Index()
    {
        return View();
    }
    public IActionResult Welcome(string name, int numTimes = 1)
    {
        ViewData["Message"] = "Hello " + name;
        ViewData["NumTimes"] = numTimes;
        return View();
    }
}
```

Sau đó, tạo View cho action này (nhấp phải vào View() → Add View).

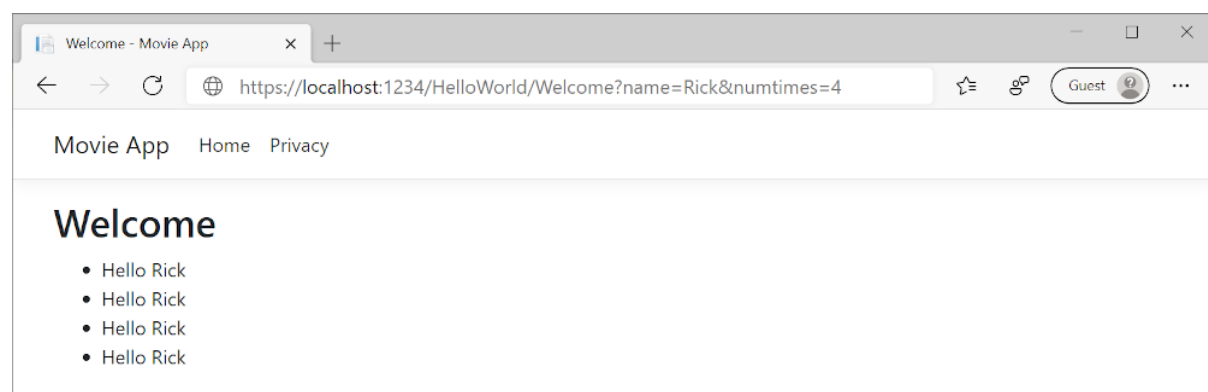
Mở file `Views/HelloWorld/Welcome.cshtml`, thay đổi với nội dung sau:

```
@{
    ViewData["Title"] = "Welcome";
}

<h2>Welcome</h2>

<ul>
    @for (int i = 0; i < (int)ViewData["NumTimes"]!; i++)
    {
        <li>@ViewData["Message"]</li>
    }
</ul>
```

Sau đó chạy ứng dụng và giải thích kết quả.



Part 4: Add Model

These model classes are used with Entity Framework Core (EF Core) to work with a database. EF Core is an object-relational mapping (ORM) framework that simplifies the data access code that you have to write.

The model classes created are known as POCO classes, from Plain Old CLR Objects. POCO classes don't have any dependency on EF Core. They only define the properties of the data to be stored in the database.

Bài 3.9 Thêm Model

Yêu cầu:

Thêm model Movie bằng cách nhấp phải vào thư mục Models → Add → Class, đặt tên Movie.cs

Cập nhật file Movie.cs với nội dung sau:

```
using System.ComponentModel.DataAnnotations;

namespace MvcMovie.Models
{
    public class Movie
    {
        public int Id { get; set; }
        public string? Title { get; set; }

        [DataType(DataType.Date)]
        public DateTime ReleaseDate { get; set; }
        public string? Genre { get; set; }
        public decimal Price { get; set; }
    }
}
```

Build project để đảm bảo không có lỗi nào.

Bài 3.10 Tạo các trang Thêm/Xóa/Sửa Movie

Yêu cầu:

Sử dụng công cụ **Scaffolding** để tạo các trang Create, Read, Update, và Delete (CRUD) cho model Movie.

Hướng dẫn: xem tại <https://learn.microsoft.com/vi-vn/aspnet/core/tutorials/first-mvc-app/adding-model?view=aspnetcore-8.0&tabs=visual-studio>

Bài 3.11 Sử dụng tính năng Migrations của Entity Framework Core để tạo và cập nhật Database

Yêu cầu:

Sử dụng lệnh **Add-Migration** và **Update-Database** để tạo và cập nhật Database.

Sau đó, chạy ứng dụng và xem kết quả (bấm vào link Movie App).

Hướng dẫn: xem tại: <https://learn.microsoft.com/vi-vn/aspnet/core/tutorials/first-mvc-app/adding-model?view=aspnetcore-8.0&tabs=visual-studio>

Part 5: Work with a database

Bài 3.12 Xem chuỗi kết nối ConnectionString

Đối với môi trường phát triển (local development), chuỗi kết nối được lưu tại file **appsettings.json**.

Hãy mở file này và xem chuỗi kết nối đến CSDL.

Hướng dẫn: xem tại <https://learn.microsoft.com/vi-vn/aspnet/core/tutorials/first-mvc-app/working-with-sql?view=aspnetcore-8.0&tabs=visual-studio>

Bài 3.13 Mở Cơ sở dữ liệu (CSDL) trong SQL Server Object Explorer (SSOX)

Giới thiệu về SQL Server Express LocalDB

LocalDB:

- Is a lightweight version of the SQL Server Express Database Engine, installed by default with Visual Studio.
- Starts on demand by using a connection string.
- Is targeted for program development. It runs in user mode, so there's no complex configuration.
- By default creates .mdf files in the C:/Users/{user} directory.

Hãy xem CSDL Movie với công cụ SSOX.

Hướng dẫn: xem tại <https://learn.microsoft.com/vi-vn/aspnet/core/tutorials/first-mvc-app/working-with-sql?view=aspnetcore-8.0&tabs=visual-studio>

Bài 3.14 Tạo một số dòng dữ liệu ban đầu cho model Movie

Hãy tạo một số dòng dữ liệu ban đầu cho model Movie.

Bước 1: Bổ sung thêm class `SeedData` để tạo một số dữ liệu cho bảng Movie.

Bước 2: Gọi phương thức `SeedData.Initialize` trong file `Program.cs`

Sau đó, chạy ứng dụng và xem kết quả.

Hướng dẫn: xem tại <https://learn.microsoft.com/vi-vn/aspnet/core/tutorials/first-mvc-app/working-with-sql?view=aspnetcore-8.0&tabs=visual-studio>

Index - Movie App

localhost:7254/Movies

Movie App

Home

Privacy

Index

[Create New](#)

Title	ReleaseDate	Genre	Price	
When Harry Met Sally	2/12/1989	Romantic Comedy	7.99	Edit Details Delete
Ghostbusters	3/13/1984	Comedy	8.99	Edit Details Delete
Ghostbusters 2	2/23/1986	Comedy	9.99	Edit Details Delete
Rio Bravo	4/15/1959	Western	3.99	Edit Details Delete

© 2023 - Movie App - [Privacy](#)