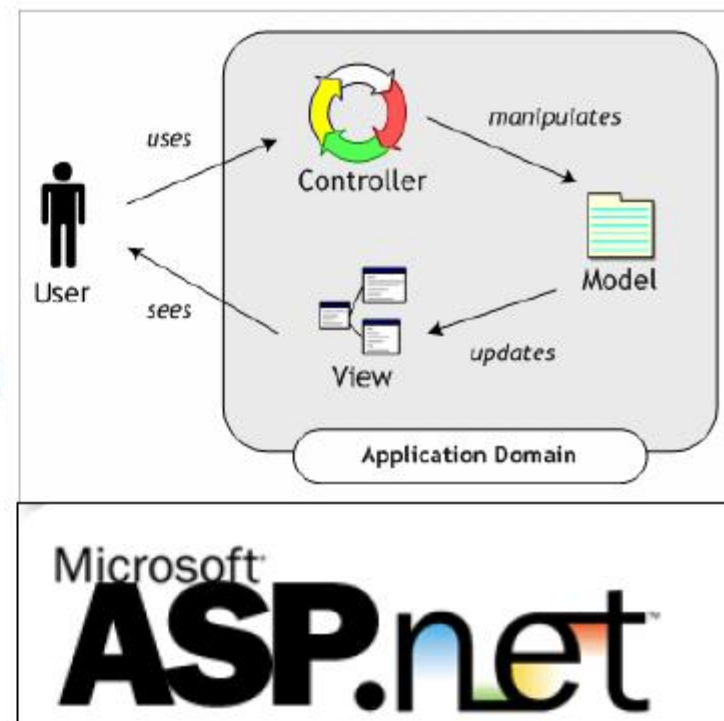


Tổng quan về ASP.Net MVC

Giới thiệu ASP.Net MVC

- ◆ ASP.Net MVC là một framework phát triển các ứng dụng web động dựa trên .Net Framework.
- ◆ Trước ASP.Net MVC, để phát triển các ứng dụng web động dựa trên .Net Framework chúng ta có thể dùng các công nghệ sau:
 - ◆ ASP.Net web forms
 - ◆ ASP.Net web pages

Microsoft
ASP.net MVC



Ưu điểm của ASP.Net MVC5

- ◆ Lập trình đơn giản với ngôn ngữ nền C# hoặc VB.Net
- ◆ Ứng dụng được tạo ra chạy ổn định trên máy chủ Windows
- ◆ Dễ quản lý nâng cấp
- ◆ An toàn và bảo mật
- ◆ Dễ tích hợp

MVC Pattern

◆ Controller

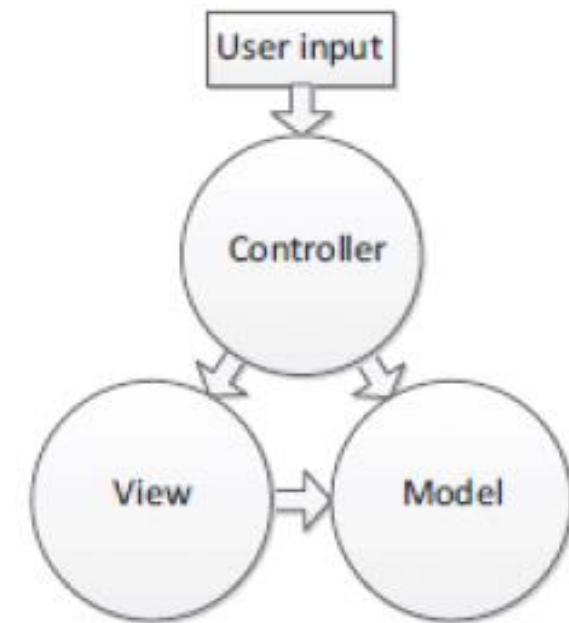
- ◆ Nhận yêu cầu từ user
- ◆ Để xử lý yêu cầu của user, có thể gọi tới model phù hợp
- ◆ Sau khi xử lý xong yêu cầu của user, có được model thì gửi model cho view

◆ View

- ◆ Tiếp nhận model từ controller để hiển thị dữ liệu cho người dùng

◆ Model

- ◆ Chứa dữ liệu chia sẻ giữa controller và view



MVC Pattern

- ◆ MVC Pattern

- ◆ Model

- ◆ View

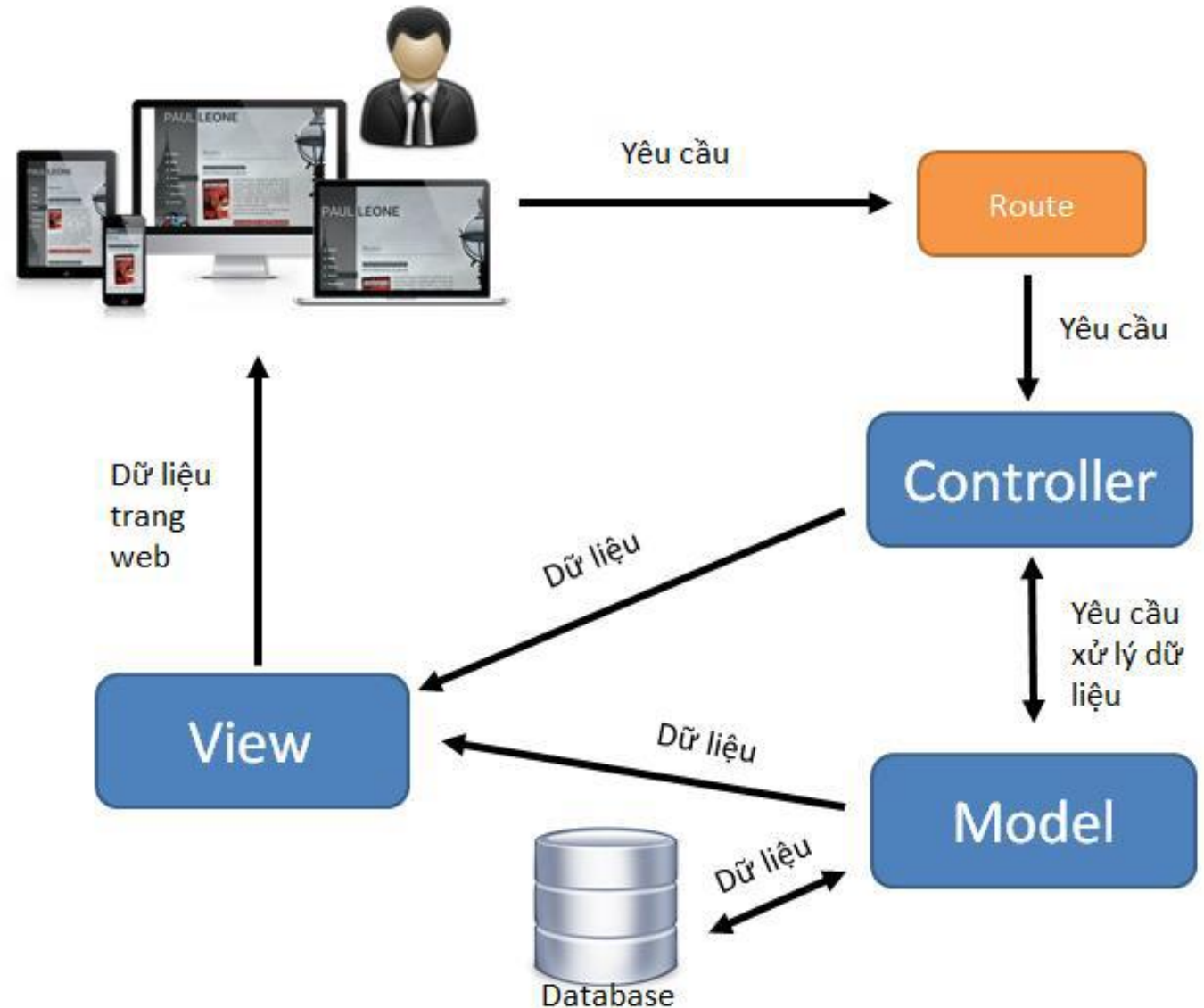
- ◆ Controller

- ◆ MVC Framework

- ◆ Java: Struts, Spring, JSF...

- ◆ PHP: Zend

- ◆ ASP.Net MVC



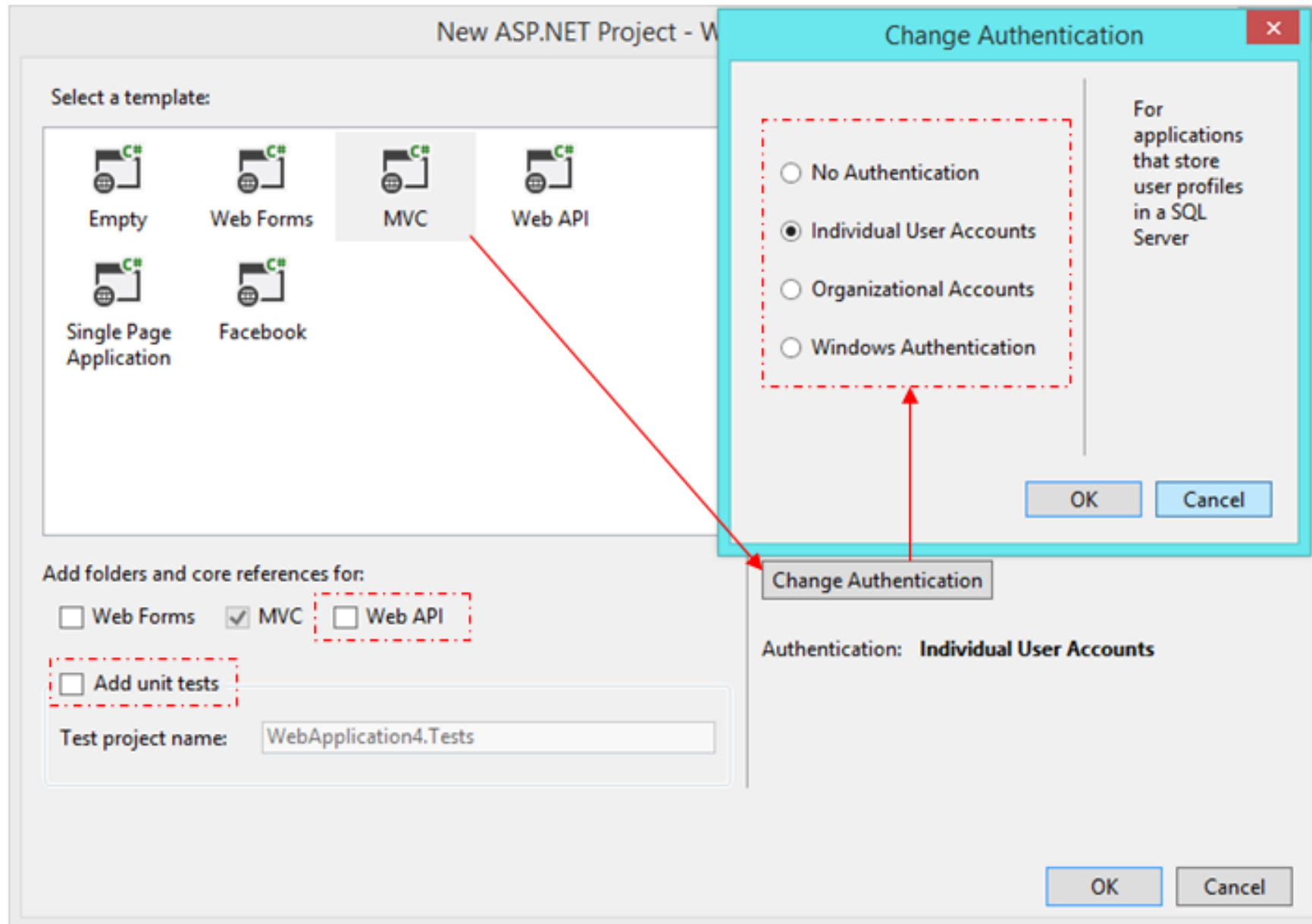
Đặc điểm của ASP.Net MVC

- ◆ MVC: Tách bạch các phần việc trong xử lý yêu cầu.
- ◆ Sát với giao thức web:
 - ◆ Không hướng sự kiện,
 - ◆ Không duy trì thông tin trạng thái như ASP.NET
- ◆ Tự động nhận diện thiết bị:
 - ◆ Tự lựa chọn view phù hợp.
 - ◆ Razor: hỗ trợ sinh giao diện.
- ◆ Kiểu dữ liệu động: ViewBag/DataView
- ◆ Cải thiện Ajax: JQuery+ Helper Ajax.

Đặc điểm của ASP.Net MVC

- ◆ Kiểm lỗi: lập trình một lần áp dụng cho cả 2 client và server.
- ◆ Web API: thư viện web dùng cho nhiều loại thiết bị
- ◆ Action Filter: kiểm soát các Action.
- ◆ Dễ test: dễ dàng test các Action của các Controller
- ◆ NuGet: quản lý các gói mở rộng.

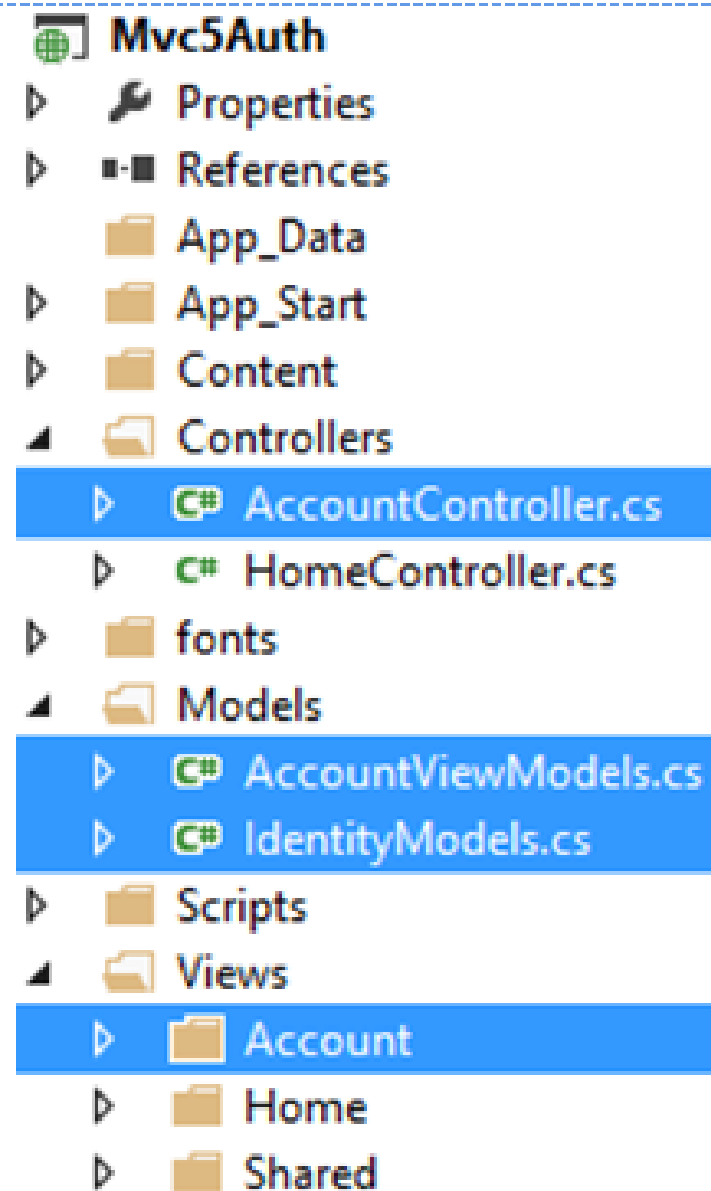
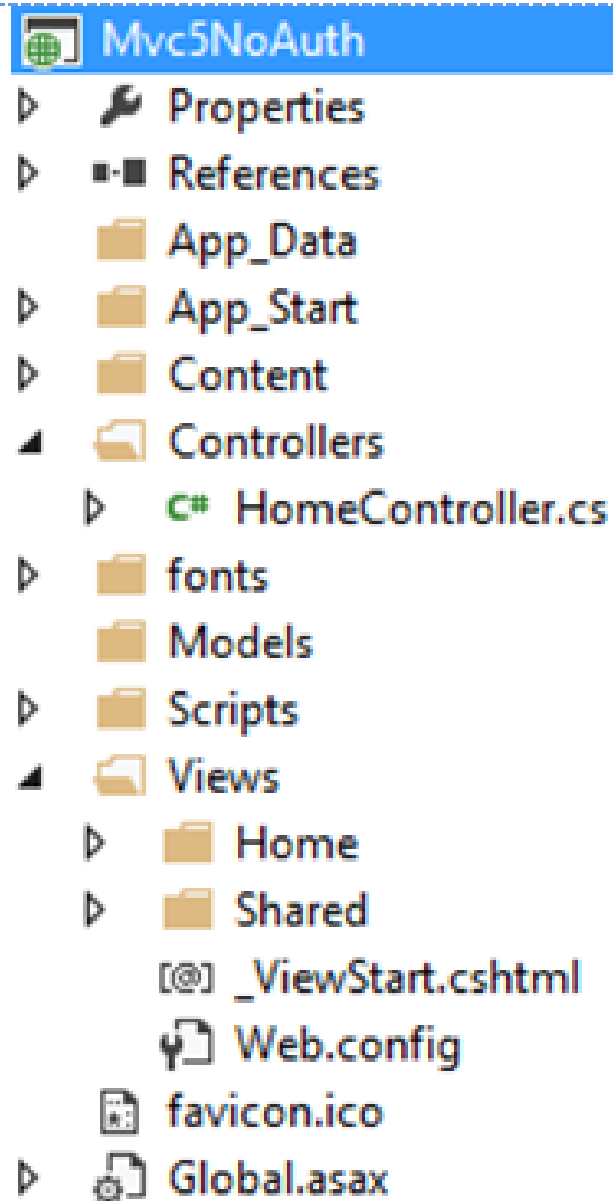
Tạo dự án ASP.Net MVC



Loại dự án

- ◆ **Individual User Accounts:** Dự án được tạo ra là loại dự án Internet bao gồm cả phần security
- ◆ **No Authentication:** Dự án được tạo ra là dự án Internet không bao gồm phần security.
- ◆ **Windows Authentication:** Dự án được tạo ra là loại dự án Intranet bao gồm cả phần security nhưng tài khoản được quản lý trên mạng nội bộ.
- ◆ **Organizational Accounts:** Dự án được tạo ra là loại dự án sử dụng tài khoản từ cloud computing.
- ◆ Ngoài ra: bạn có thể chọn Web API để bổ sung khả năng hỗ trợ xây dựng thư viện web.

Cấu trúc dự án



Các thư mục phân cấp

◆ Controllers

- ◆ Thư mục này chứa tất cả các Controller trong ứng dụng. Các Controller đáp ứng yêu cầu đến từ trình duyệt, quyết định phải làm gì và phản hồi kết quả.

◆ Models

- ◆ Các lớp mô hình dữ liệu được tổ chức để nắm giữ và thao tác dữ liệu

◆ Views

- ◆ View chứa các mẫu giao diện người dùng của ứng dụng

◆ App_Data

- ◆ Chứa các file cơ sở dữ liệu (nếu có)

◆ App_Start

- ◆ Chứa lớp được chạy một lần khi ứng dụng bắt đầu, thường là các file cấu hình

Các thư mục phân cấp

- ◆ Content

- ◆ Chứa tài nguyên tĩnh như hình ảnh, CSS...

- ◆ Scripts

- ◆ Chứa các file JavaScript cần thiết của ứng dụng

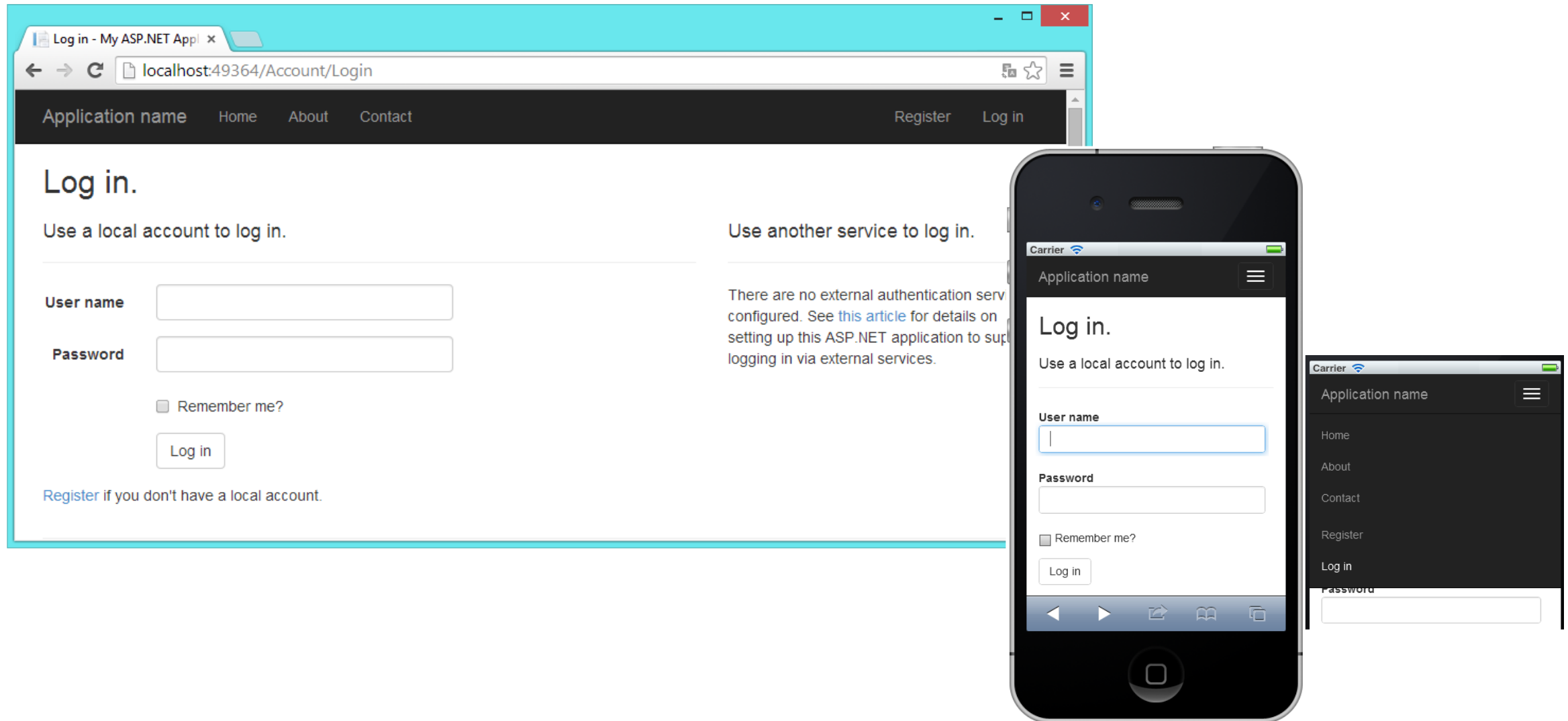
- ◆ Global.asax

- ◆ Tập tin sự kiện chứa các điều khiển sự kiện nhằm kiểm soát vòng đời của Request, Session, Application...

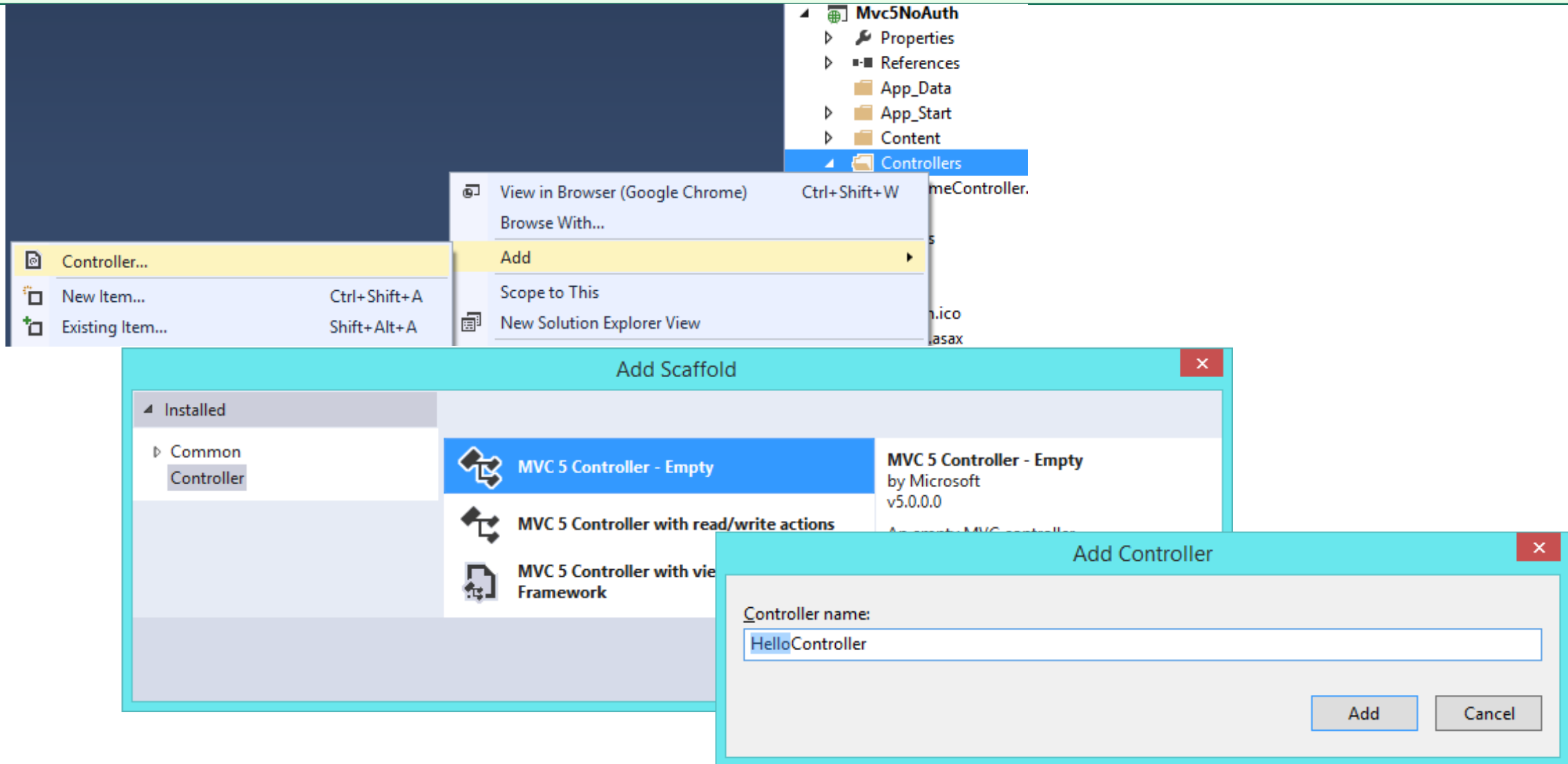
- ◆ Web.config

- ◆ Tập tin xml chứa thông tin cấu hình của ứng dụng web

Hiển thị trên các thiết bị khác nhau



Tạo Controller



Tạo Action

Visual Studio interface showing the `HelloController.cs` file and the `Solution Explorer`.

Code in `HelloController.cs`:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace Mvc5.Controllers
{
    public class HelloController : Controller
    {
        //
        // GET: /Hello/
        public ActionResult Index()
        {
            return View();
        }

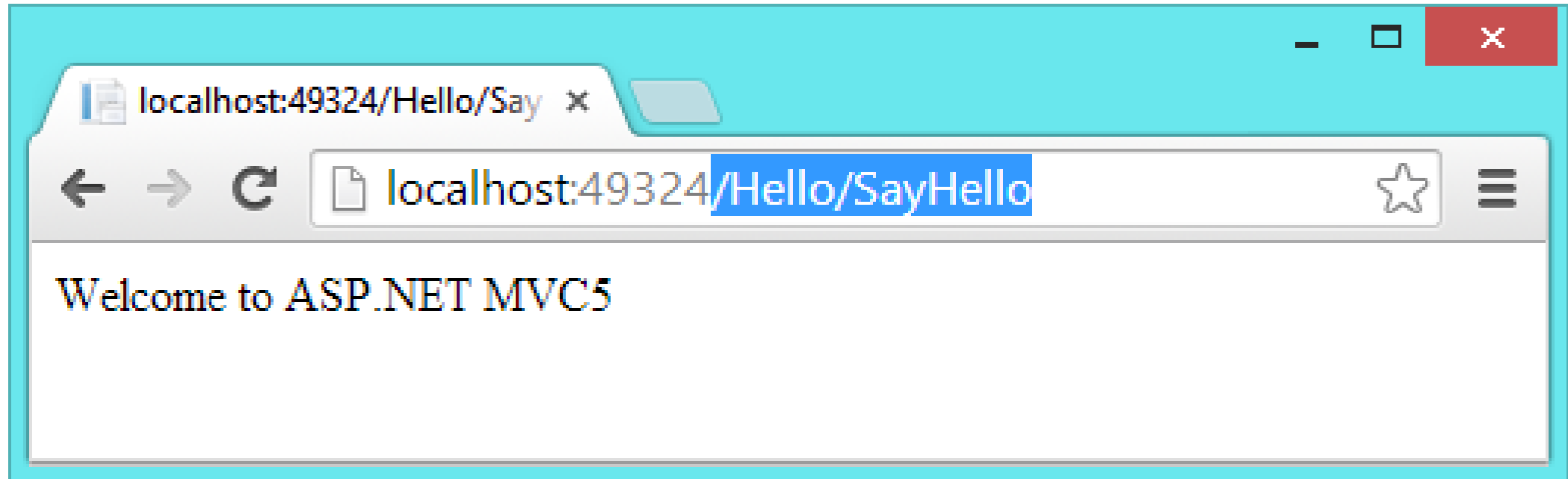
        //
        // GET: /Hello/SayHello
        public ActionResult SayHello()
        {
            return Content("Welcome to ASP.NET MVC5");
        }
    }
}
```

A yellow callout bubble points to the `SayHello()` method with the text: **Thêm một Action mới** (Add a new action).

The `Solution Explorer` on the right shows the project structure. A context menu is open over the `Controllers` folder, listing options: **Build**, **Rebuild**, **Clean**, **View**, **Publish...**, **Scope to This**, and **New Solution Explorer View**.

◆ Chú ý: `return Content("...");`

Kết quả chạy Action



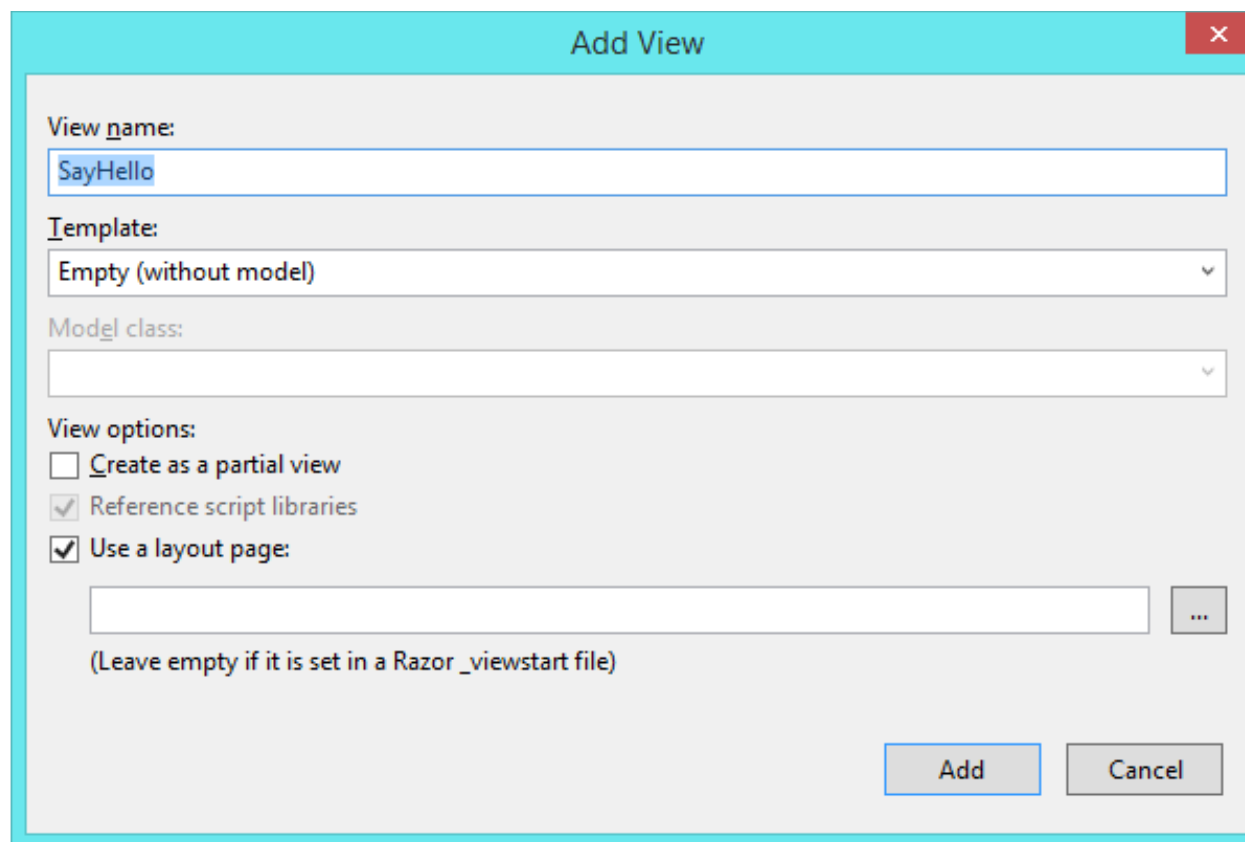
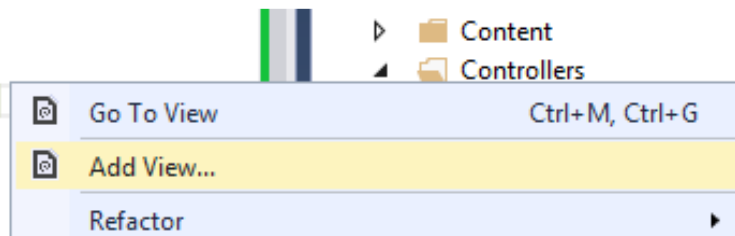
Tạo View cho Action

◆ Chú ý:

`return View()`

Tên của View

```
//  
// GET: /Hello/SayHello  
public ActionResult SayHello()  
{  
    return View();  
}
```



View name: SayHello

Template: Empty (without model)

Model class:

View options:

- ☐ Create as a partial view
- ☒ Reference script libraries
- ☒ Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)

Add Cancel

Code Razor cho View

The image shows a Visual Studio IDE with two main panels. The left panel displays the code for `SayHello.cshtml`, which is a Razor view. The code includes a metadata block, a title assignment, a heading, a horizontal line, and an author name. The right panel shows the `Solution Explorer` with a tree view of the project structure. A red arrow points from the `SayHello.cshtml` file in the Solution Explorer to the code in the left panel. Another red arrow points from the `Return View()` method in the `HelloController` class to the `SayHello.cshtml` file.

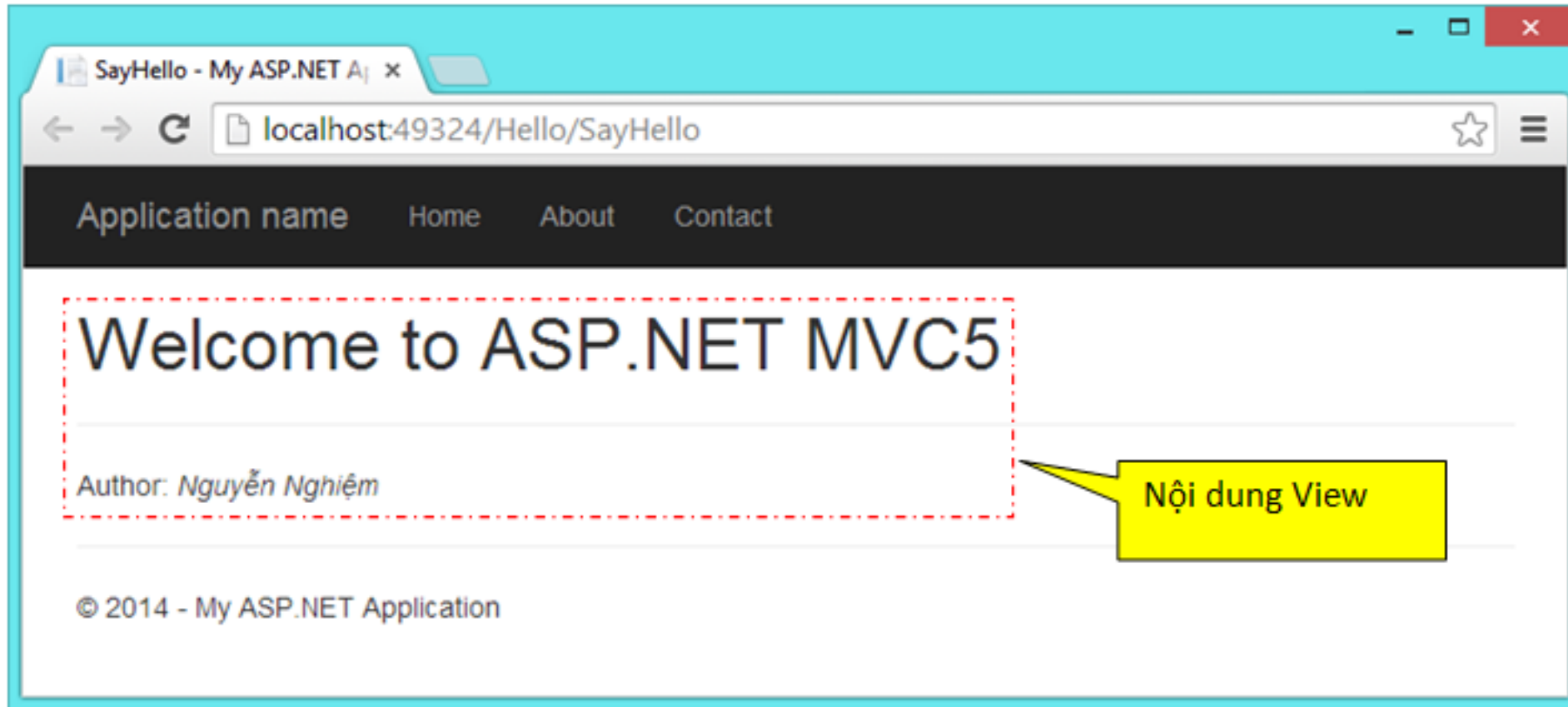
```
@{  
    ViewBag.Title = "Page Title";  
}  
  
<h1>Welcome to ASP.NET MVC5</h1>  
  
<hr />  
  
Author: <em>Nguyễn Nghiệm</em>
```

Solution Explorer

- App_Start
- Content
- Controllers
 - HomeController.cs
 - HomeController
 - Index() : ActionResult
 - SayHello() : ActionResult
- fonts
- Models
- Scripts
- Views
 - Hello
 - SayHello.cshtml

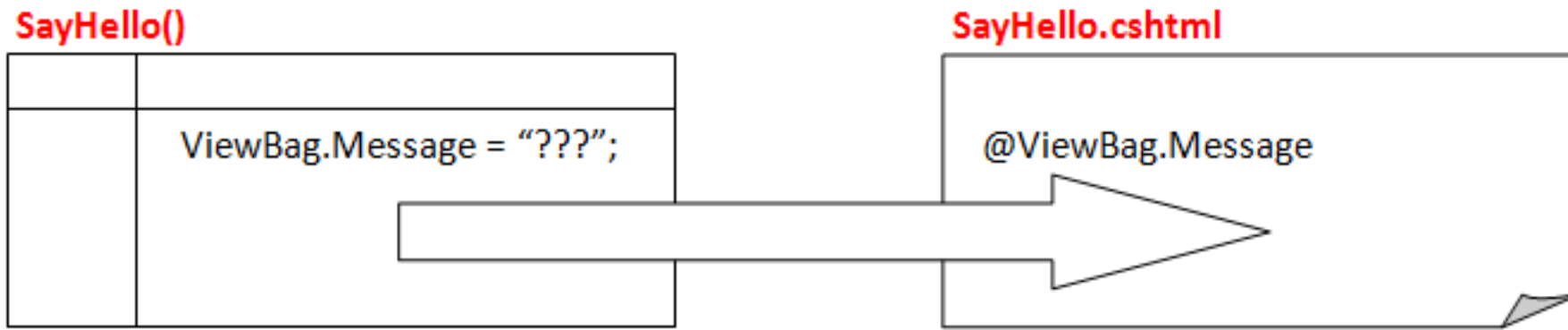
Return View()

Kết quả chạy Action



Truyền dữ liệu

◆ Truyền dữ liệu từ Controller sang View dùng ViewBag



```
public ActionResult SayHello()
{
    ViewBag.Message = "Welcome to ASP.NET MVC5";
    return View();
}
```

```
SayHello.cshtml  X HelloController.cs
@{
    ViewBag.Title = "Page Title";
}
<h1>@ViewBag.Message</h1>
<hr />
```

Định tuyến

Mvc5NoAuth

▶ Properties

▶ References

App_Data

▶ App_Start

▶ BundleConfig.cs

▶ FilterConfig.cs

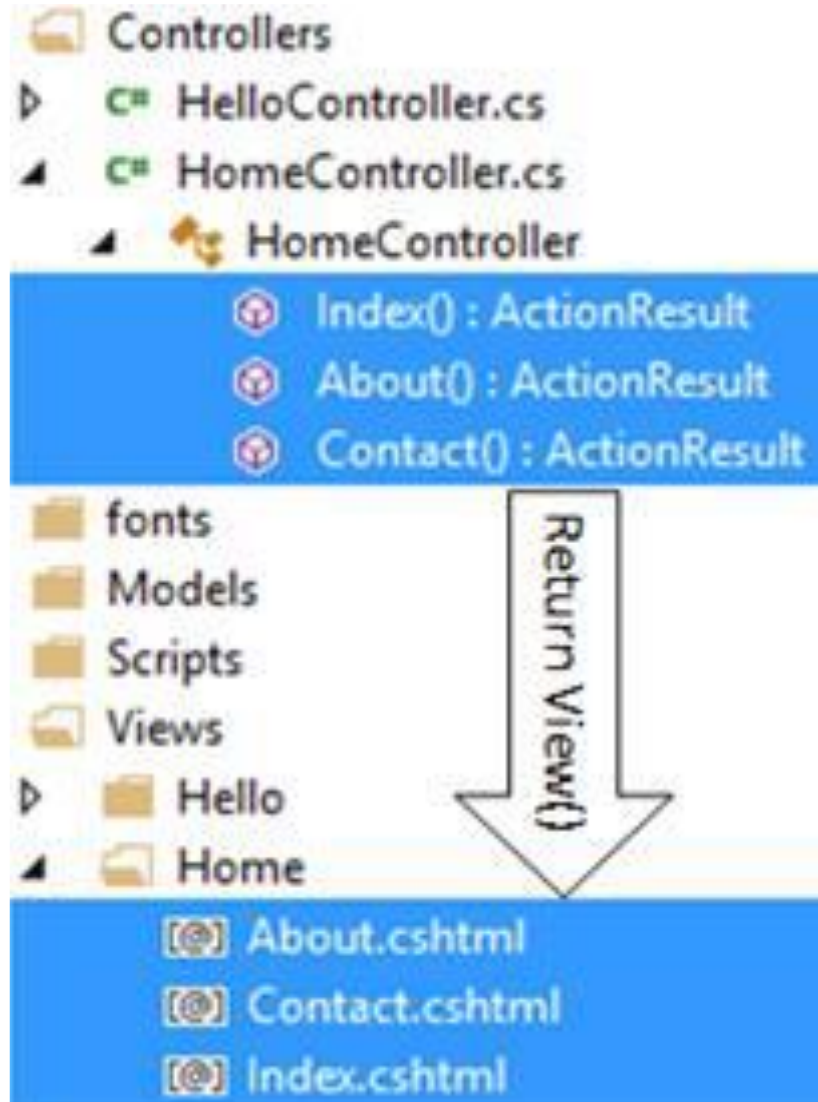
▶ RouteConfig.cs

Việc cấu hình này được thực hiện trong file RouteConfig.cs đặt tại thư mục App_Start. Sau đây là nội dung của file này.

```
public class RouteConfig
{
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

        routes.MapRoute(
            name: "Default",
            url: "{controller}/{action}/{id}",
            defaults: new
            {
                controller = "Home",
                action = "Index",
                id = UrlParameter.Optional
            }
        );
    }
}
```

Định tuyến



Gọi action Index() của HomeController:

- ✓ <http://localhost:49324/Home/Index/2000>
- ✓ <http://localhost:49324/Home/Index>
- ✓ <http://localhost:49324/Home>
- ✓ <http://localhost:49324>

Gọi action About() của HomeController:

- ✓ <http://localhost:49324/Home/About/2000>
- ✓ <http://localhost:49324/Home/About>

Gọi action Index() của HelloController:

- ✓ <http://localhost:49324/Hello/Index/2000>
- ✓ <http://localhost:49324/Hello/Index>
- ✓ <http://localhost:49324/Hello>

Định tuyến

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

    routes.MapRoute(
        name: "Product",
        url: "Product/{*pathInfo}",
        defaults: new
        {
            controller = "ProductManager",
            action = "Search"
        }
    );

    routes.MapRoute(
        name: "Default",
        url: "{controller}/{action}/{id}",
        defaults: new
        {
            controller = "Home",
            action = "Index",
            id = UrlParameter.Optional
        }
    );
}
```

<http://localhost:49324/Product/Hello World/>
<http://localhost:49324/Product/Hello/World>
<http://localhost:49324/Product>