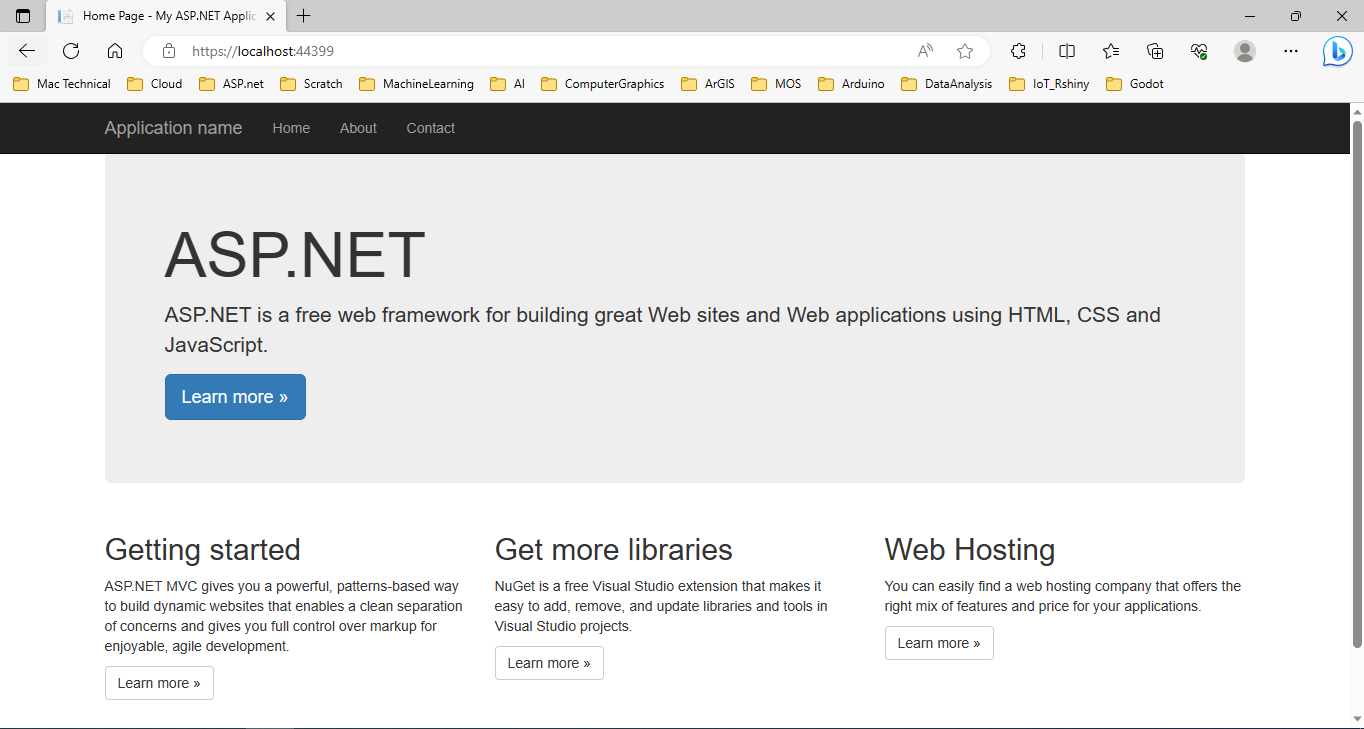
**Tạo và tương tác giữa các mô hình MVC**

Xem xét lại ví dụ trước:



Có thể nhận thấy trên menu của trang Web có 3 mục: Home, About, Contact

|  |  |
| --- | --- |
|  | Các menu này có thể tìm thấy ở Lớp VIEW bên trong thư mục **HOME**  Mối liên hệ giữa thư mục HOME với tập tin HomeController.cs trong lớp Controller. Nghĩa là khi tạo mới một controller đồng nghĩa với một folder mới có tên tương ứng được sinh ra và nằm trong lớp VIEW. |

Thử add thêm một controller mới tên là **Trangchu**:

* R\_click Controller - Add - Controller - MVC 5 Controller - **Empty**

Controller name: (tên\_controller)Controller: **TrangchuConntroller**

|  |  |
| --- | --- |
| Add Controller |  |

***Cấu trúc file TrangchuController.cs***

using System.Web.Mvc;

namespace Bai2.Controllers

{

public class TrangchuController : Controller

{

// GET: Trangchu

public ActionResult Index()

{

return View();

}

}

}

Có thể nhận ra ở đây, Controller tên Trangchu được dẫn xuất từ lớp cha Controller trong namspace System.Web.Mvc

Trong mọi contronller đều có phương thức (ví dụ Index) trả về một Result (Action method - trả về cho View): sẽ cần file **Index.cshtml**

* Thêm nội dung (Index.cshtml) cho phần Trangchu

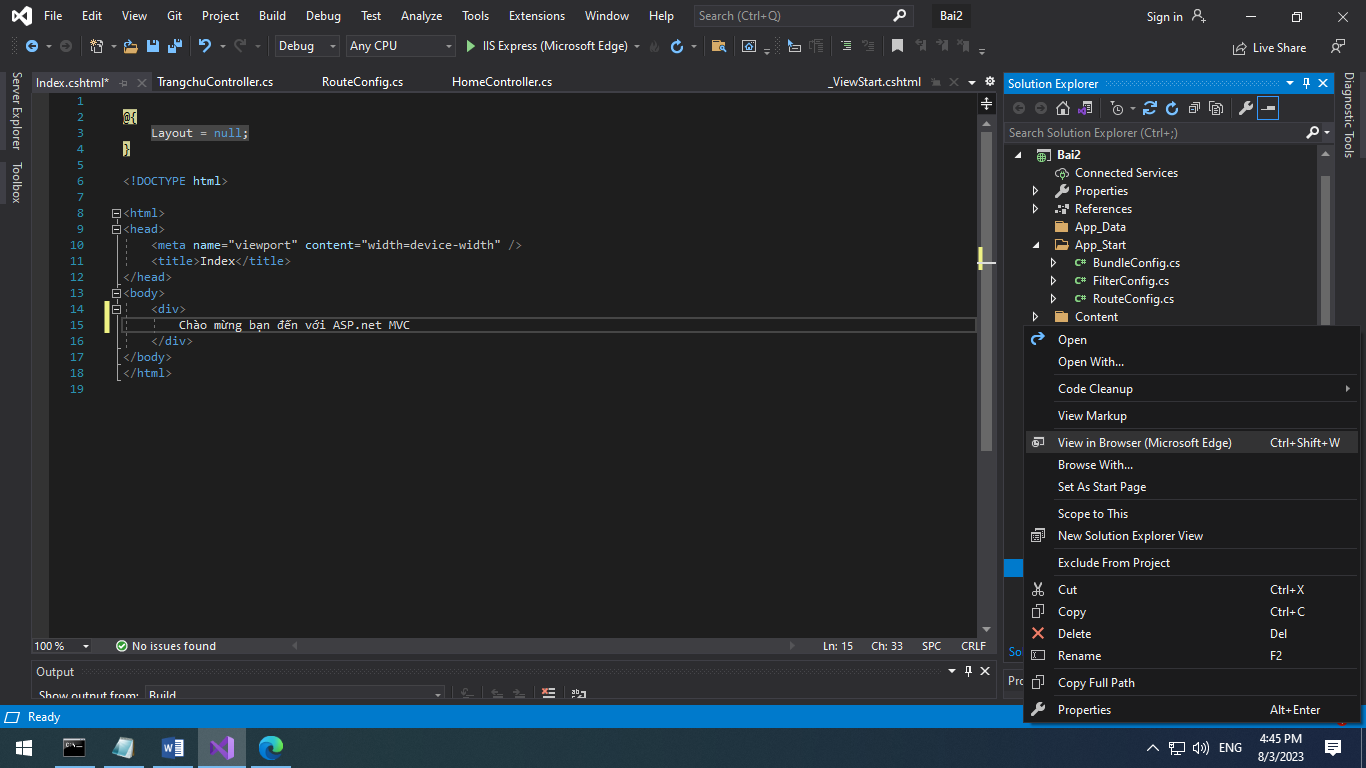
R\_Click - return View(); Add view - Add - như hình

|  |  |
| --- | --- |
| Add View |  |

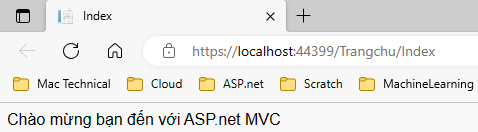
Chúng ta có thể điều hướng qua lại TrangchuController.cs với Index.cshtml trong thư mục trang chủ.

Thêm nội dung cho phần Index.cshtml

Thực thi: R\_click Index.cshtml - View in Browser



Câu hỏi đặt ra là làm sao biết được link nào đang thuộc controller nào?



Nhìn thanh địa chỉ của trang web: địa chỉ/Tên controller/Tên Method: <https://localhost:44399/Trangchu/Index>

Nguyên nhân là tập tin Global.asax có tác dụng điều hướng như sau:

|  |  |
| --- | --- |
|  |  |

Hãy để ý thư mục App\_Start

Tìm đến nội dung của RouteConfig bằng cách bôi đen RouteConfig và bấm F12

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 }

);

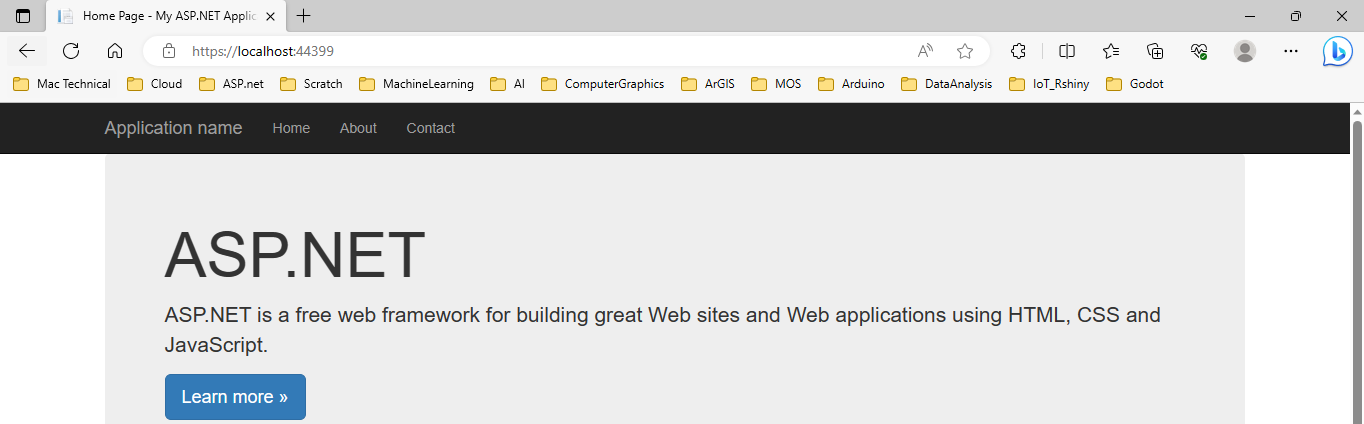
}

}

Hãy nhìn dòng URL: url: "{controller}/{action}/{id}" để thấy 3 thành phần

Với ID mà không truyền tham số vào: defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }

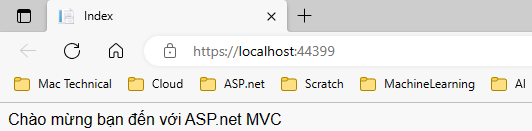
Đó là lý do tại sao khi thực hiện debug trang Web sẽ có trang như bên dưới mà không phải là Index.cshtml trong thư mục trang chủ



Vậy thì sau khi chỉnh sửa nội dung RouteConfig.cs:

defaults: new { controller = "Trangchu", action = "Index", id = UrlParameter.Optional }

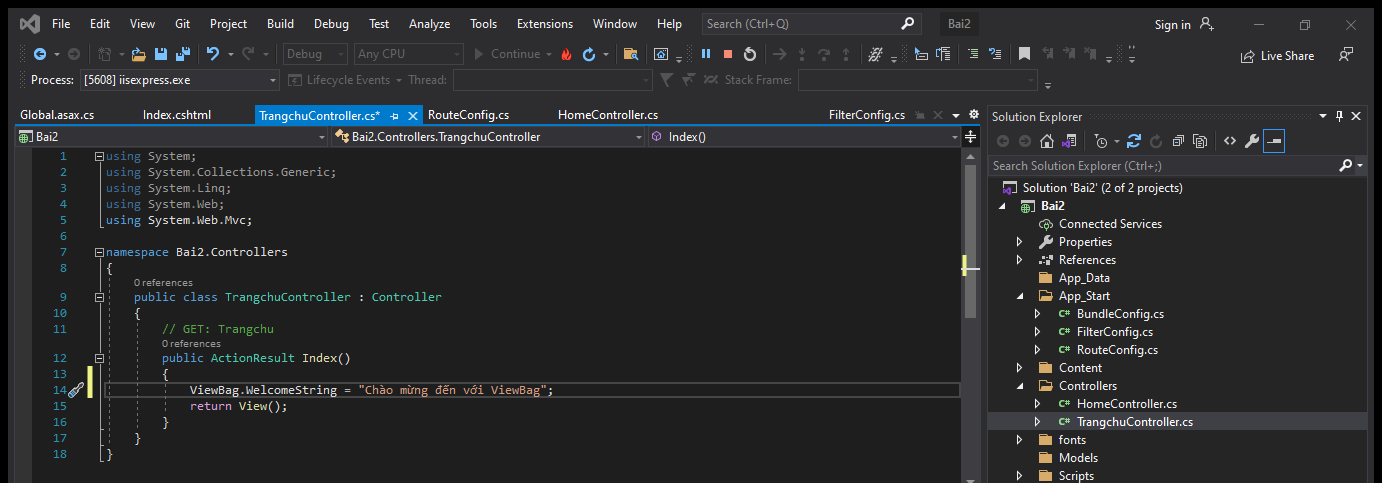
Kết quả là:



Làm thế nào để truyền dữ liệu từ Controller về cho VIEW?

Có 2 cách: (1) Dùng ViewBag; (2) Dùng Model

**(1). ViewBag (open TrangchuController.cs)**



Tiếp theo, làm thế nào để truyền dữ liệu từ controller đến View?

* Mở file **Index.cshtml** trong thư mục **Trangchu** của View:

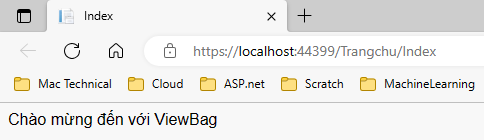
Giữa 2 thẻ <div> </div>

<div>

@ViewBag.WelcomeString

</div>

* Mở file **TrangchuController.cs** – Bấm F6 để build lại hệ thống
* Mở trình duyệt F5 để cập nhật lại nội dung ở trang chủ của website



**(2) Dùng Model**

Tạo mới model: R\_Click (model) - Add - Class - MessageModel.cs

public class MessageModel

{

public string Welcome { set; get; }

}

Set thuộc tính Welcome (hoặc gì đó)

Chuyển đến **TrangchuController.cs**, thay thế **ViewBag** thành Model như sau:

public class TrangchuController : Controller

{

// GET: Trangchu

public ActionResult Index()

{

var message = new MessageModel();

message.Welcome = "Chào mừng đến với Modeel";

return View(message);

}

}

Method sẽ trả về View có nội dung là message

Quay trở về trang Index.cshtml để khai báo model cần sử dụng:

@model Bai2.Models.MessageModel

(Nếu không hiển thị tự động Bai2.Model… thì click vào biểu tượng dấu +)

Khai báo như trên có nghĩa là toàn bộ trang Index sẽ sử dụng Model tên MessageModel.

<div>

@Model.Welcome

</div>

@Model sẽ đại diện cho cả Model