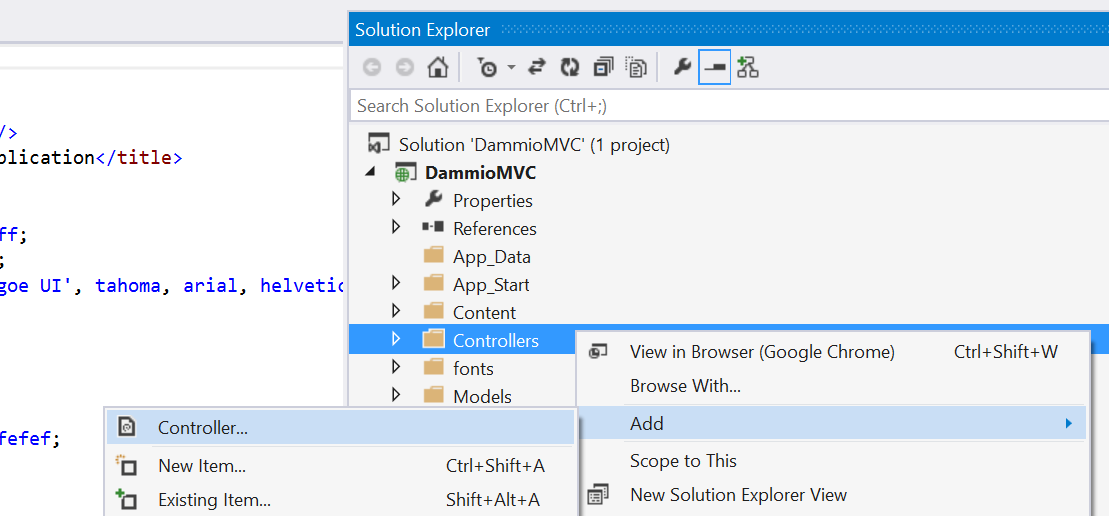
Phần 3: Thêm mới Controller[[1]](#footnote-1)

[**MVC**](https://www.dammio.com/glossary/mvc) (viết tắt model-view-controller) là một mẫu (pattern) dùng để phát triển các ứng dụng có kiến trúc tổ chức tốt, dễ duy trì và có khả năng thử nghiệm. Các ứng dụng dựa trên nguyên lý [MVC](https://www.dammio.com/glossary/mvc) gồm các thành phần:

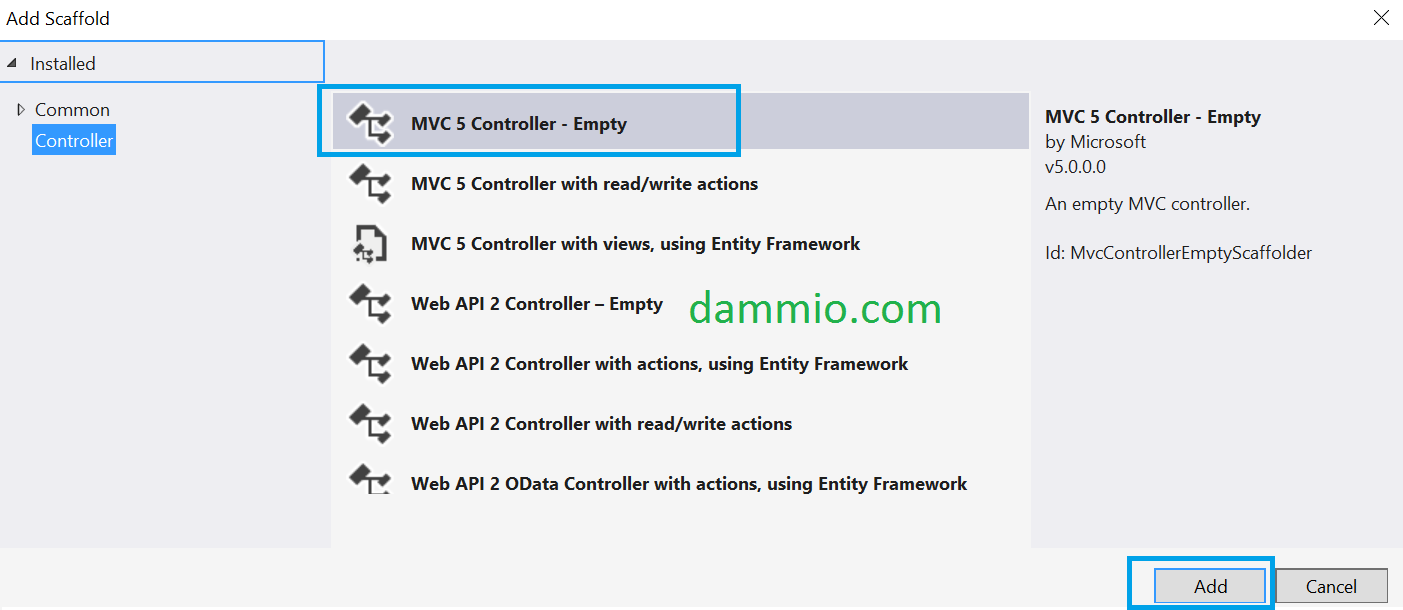
* **Models**: Các lớp thể hiện dữ liệu của ứng dụng và dùng nguyên lý xác nhận (validation) để thực thi các luật (rules) cho dữ liệu đó.
* **Views**: Các tập tin template (bản mẫu) mà ứng dụng dùng để tự động phát sinh các phản hồi HTML.
* **Controllers**: Các lớp xử lý các yêu cầu đến từ trình duyệt, rút trích mô hình dữ liệu, và đặc tả các template xem trả về một phản hồi từ trình duyệt.

Trong bài này, chúng ta sẽ học cách tạo 1 lớp controller mới và sử dụng controller này để hiển thị dữ liệu trên trình duyệt.

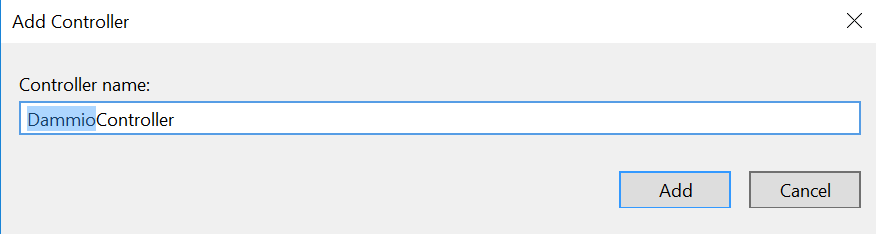
Để thêm 1 controller mới, ở cửa sổ **Solution Explorer** của dự án **DammioMVC**, chuột phải lên **thư mục Controllers** và sau đó nhấn **Add**, chọn **Controller**.



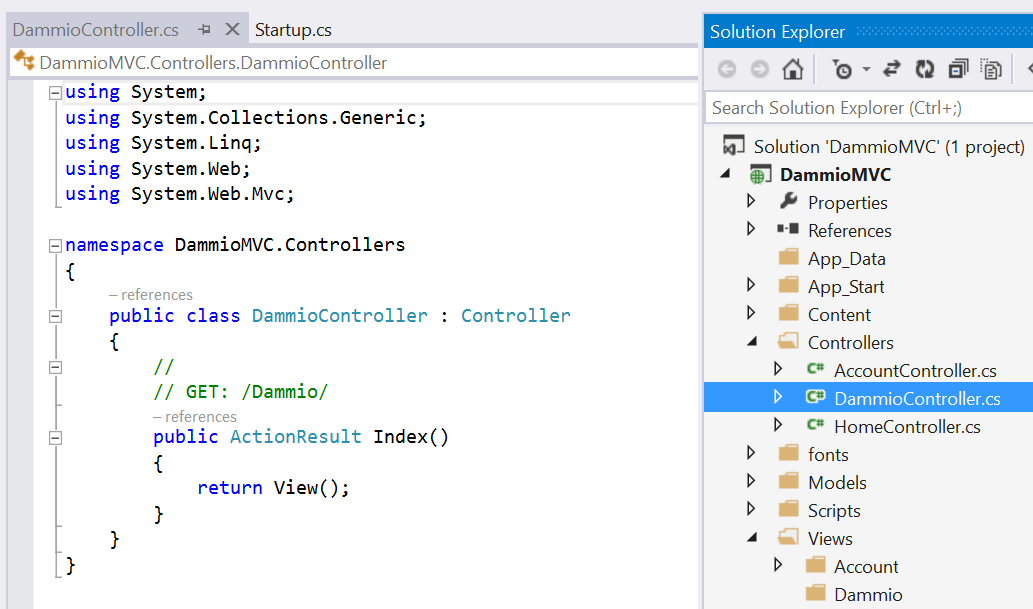
Ở hộp thoại **Add Scaffold**, chọn [**MVC**](https://www.dammio.com/glossary/mvc)**5 Controller – Empty** và nhấn **Add**. Ở bước này, do chúng ta mới làm quen với Controller cho nên chúng ta sử dụng mẫu Controller dạng Empty, tức là nội dung trống. Từ đó, chúng ta sẽ viết 1 số đoạn code làm quen. Ở những bài tiếp theo, bạn sẽ làm quen các dạng Controller khác.



Sau đó, đặt tên Controller là **DammioController**, lưu ý bạn nên để tiền tố “Controller” cuối cùng khi đặt tên để dễ phân biệt lớp code nào là Controller.



Ở bước này, thư mục Controller, lớp **DammioController.cs** được tạo ra với nội dung mặc định như trong hình sau.



Tiếp đến, bạn thêm 1 số đoạn mã vào tập tin **DammioController.cs** như sau:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace DammioMVC.Controllers

{

public class DammioController : Controller

{

//

// GET: /Dammio/

public string Index()

{

return "Đây là phương thức Index, phương thức mặc định của Controller Dammio.";

}

//

// GET: /Dammio/ChaoMung/

public string ChaoMung()

{

return "Đây là phương thức ChaoMung nằm trong Controller Dammio!";

}

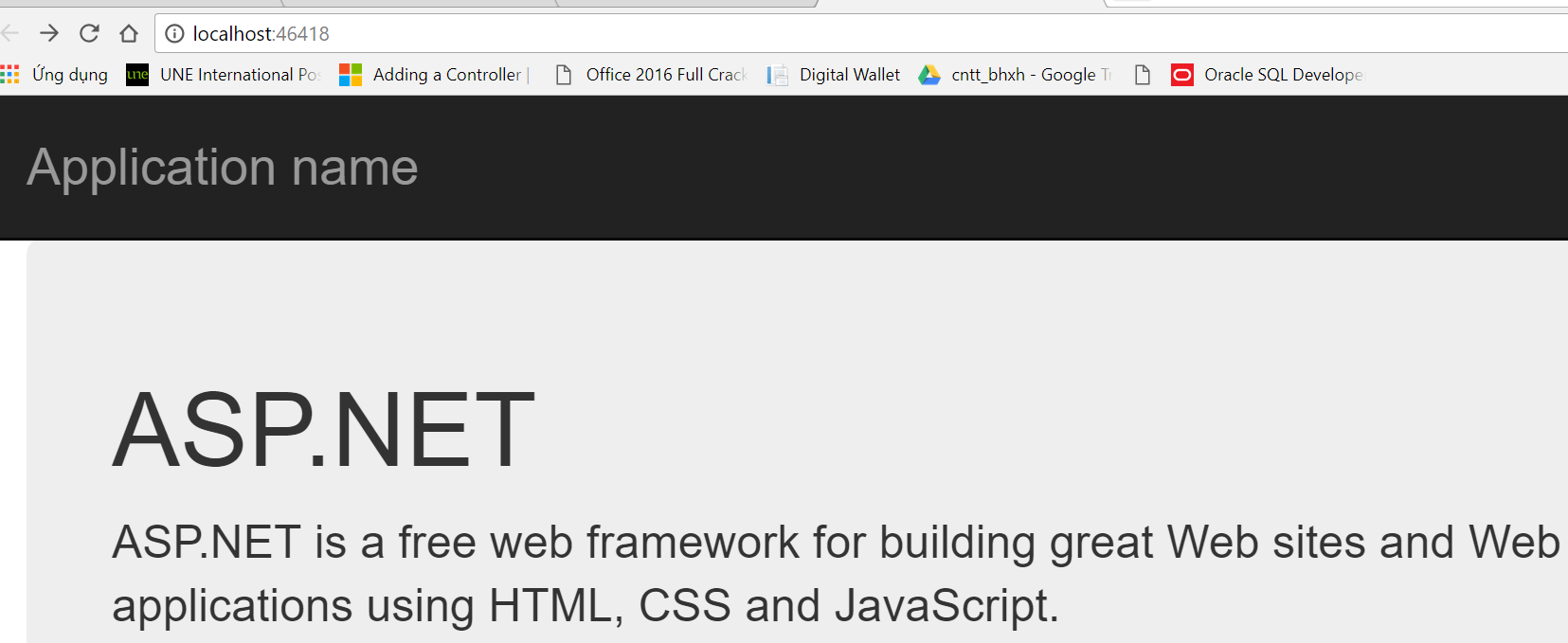
}

}

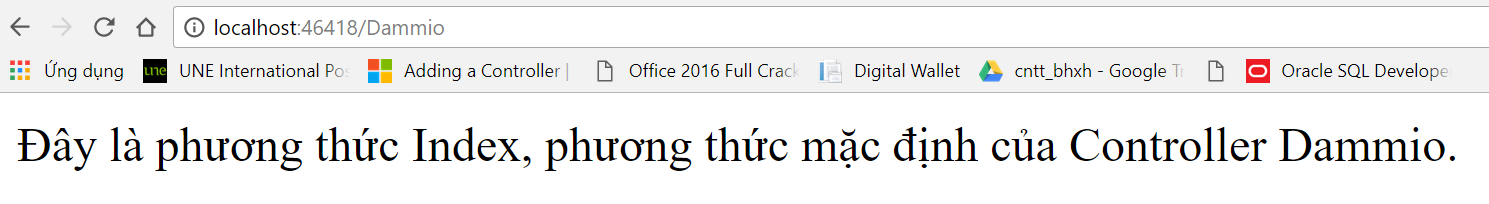
Giải thích đoạn mã trên:

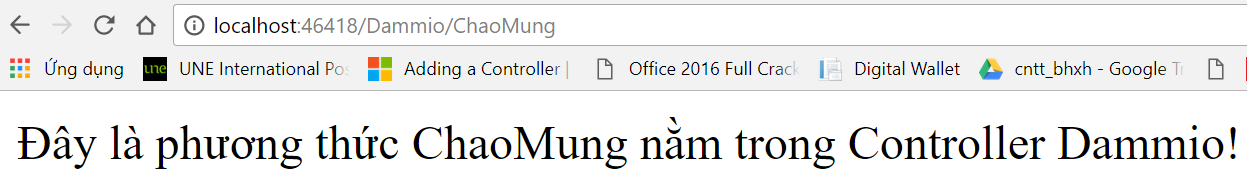
* Phương thức Index() trả về kiểu string với giá trị là “Đây là phương thức Index, phương thức mặc định của Controller Dammio.” Đây là phương thức mặc định của 1 Controller bất kỳ.
* Phương thức ChaoMung() cũng trả về kiểu string với giá trị “Đây là phương thức ChaoMung nằm trong Controller Dammio!”

Bạn có thể tạo nhiều phương thức thực thi ở tập tin DammioController.cs tùy ý. Tiếp theo, bạn thực thi ứng dụng **DammioMVC bằng cách nhấn F5 hoặc Ctrl + F5** (chế độ không cần Debug) để xem kết quả.

Giao diện website MVC mặc định

Ở trình duyệt, bạn thử chạy 2 địa chỉ **http://localhost:xxx/Dammio/** và **http://localhost:xxxx/Dammio/ChaoMung** để xem kết quả (với xxxx là số cổng tự động gieo bởi server **IIS Express** của Visual Studio, bạn không cần quan tâm số cổng này).





**ASP.NET**[**MVC**](https://www.dammio.com/glossary/mvc) sẽ gọi các lớp controller khác nhau (kèm theo các phương thức) tùy theo [URL](https://www.dammio.com/glossary/url) đầu vào. Mặc định [URL](https://www.dammio.com/glossary/url) có định dạng:

…/[Controller]/[ActionName]/[Parameters]

Phần … là địa chỉ website của bạn, ví dụ http://www.dammio.com/ hay http://localhost:xxxx/. Controller là tên phần đầu của Controller, ví dụ DammioController.cs thì có tên là Dammio. ActionName là tên phương thức, ví dụ phương thức ChaoMung. Parameters là các tham số đầu vào của các phương thức. Vì trong ví dụ trên, chúng ta chưa tạo Parameters cho phương thức vì vậy phần này trống.

Nếu bạn muốn thay đổi cấu trúc đường dẫn, bạn có thể vào tập tin App\_Start/RouteConfig.cs và chỉnh cấu trúc đường dẫn. Ví dụ chỉnh cấu trúc đường dẫn thành **{action}/{controller}/{id}**.

public static void RegisterRoutes(RouteCollection routes)

{

routes.IgnoreRoute("{resource}.axd/{\*pathInfo}");

\*/ Cấu trúc mặc định

routes.MapRoute(

name: "Default",

url: "{controller}/{action}/{id}",

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

);

/\*

// Cấu trúc đường dẫn thay đổi

routes.MapRoute(

name: "Default",

url: "{action}/{controller}/{id}",

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

}

Để thuận tiện, **bạn hãy để cấu trúc đường dẫn trong tập tin RouteConfig.cs**. Chúng ta sẽ có 1 bài viết về cách chỉnh cấu trúc này sau.

Quay trở lại đường dẫn <http://localhost:xxxx/Dammio/ChaoMung>, với cấu trúc đường dẫn mặc định ({controller}/{action}/{id}), bạn phải nắm rõ Dammio chính là tên Controller (DammioController.cs), ChaoMung là 1 phương thức trong lớp Controller Dammio. Vì chưa có tham số cho phương thức ChaoMung cho nên {id} ở đây là trống. Lưu ý, **đây là cách thức hoạt động đường dẫn cơ bản nhất trong**[**MVC**](https://www.dammio.com/glossary/mvc)**, bạn bắt buộc phải nắm vững**.

Để sử dụng các tham số (parameters) trên đường dẫn [URL](https://www.dammio.com/glossary/url), bạn hãy sửa nội dung phương thức ChaoMung như sau:

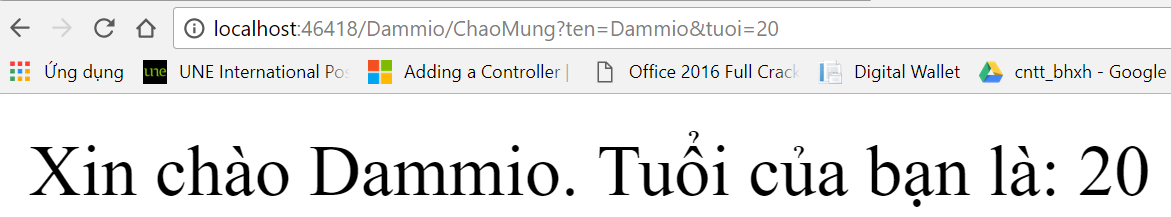
public string ChaoMung(string ten, int tuoi = 1)

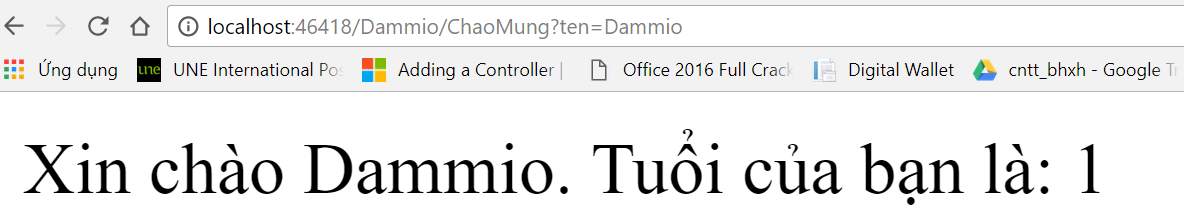
{

return HttpUtility.HtmlEncode("Xin chào " + ten + ". Tuổi của bạn là: " + tuoi);

}

Sau đó, bạn nhấn F5 hoặc Ctrl + F5 để chạy dự án, ở trình duyệt chạy [URL](https://www.dammio.com/glossary/url) như sau: **http://localhost:46418/Dammio/ChaoMung?ten=Dammio&tuoi=20**. Trong [URL](https://www.dammio.com/glossary/url) bạn có 2 tham số đầu vào là ten với giá trị Dammio và tuoi với giá trị 20. Lưu ý, tham số ten và tuoi phải giống như tham số ở phương thức ChaoMung(string ten, int tuoi = 1). Nếu bạn không mô tả cho giá trị ten, mặc định biến này có giá trị là 1.





Tiếp theo, bạn thử chạy đường dẫn **http://localhost:46418/Dammio/ChaoMung/?ten=Dammio&tuoi=20** thì cũng cho ra kết quả tương tự.

Bây giờ chúng ta thử chỉnh sửa đôi chút tập tin **App\_Start\RouteConfig.cs** để thay đổi cấu trúc đường dẫn theo đúng chuẩn SEO. Bạn chỉnh nội dung tập tin **RouteConfig.cs** như sau:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Routing;

namespace DammioMVC

{

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 }

);

routes.MapRoute(

name: "Dammio",

url: "{controller}/{action}/{ten}/{tuoi}"

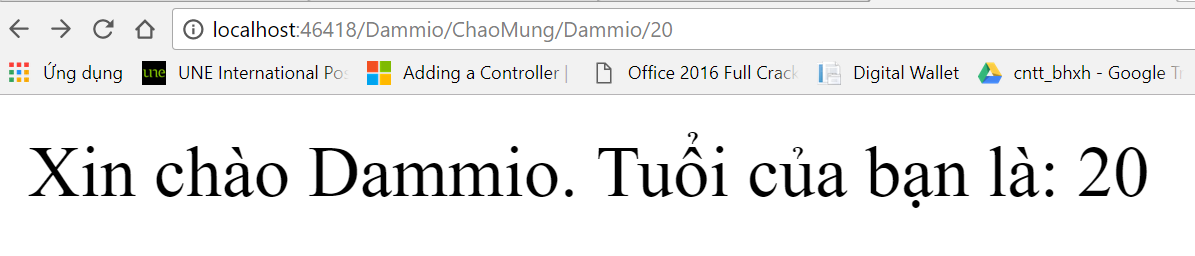
);

}

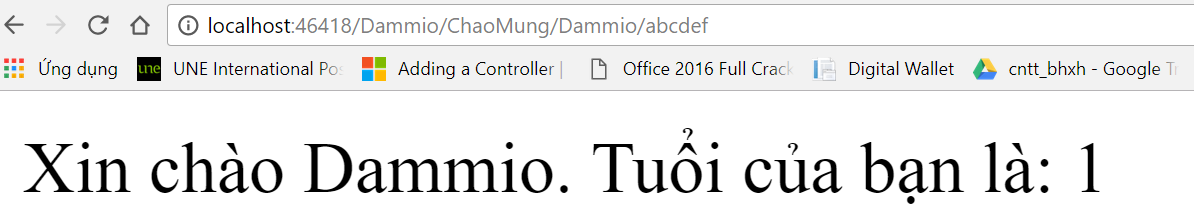
}

}

Bạn Build dự án (**Ctrl + Shift + B**) và chạy lại với F5 hoặc Ctrl + F5 (chế độ không cần Debug) và chạy đường dẫn **http://localhost:46418/Dammio/ChaoMung/Dammio/20**. Bạn cũng thấy kết quả tương tự, nhưng lúc này không cần ghi rõ tên biến (ten, tuoi) trong [URL](https://www.dammio.com/glossary/url) mà chỉ ghi theo thứ tự **{controller}/{action}/{ten}/{tuoi}**.



Chuyện gì xảy ra chạy đường dẫn **http://localhost:46418/Dammio/ChaoMung/Dammio/abcdef**. Nếu bạn để ý, giá trị “abdef” là giá trị cho tuoi, kiểu int. Trong trường hợp này do lỗi nên giá trị trả về mặc định, tức là 1.



Đối với nhiều ứng dụng [MVC](https://www.dammio.com/glossary/mvc), cấu trúc đường dẫn mặc định đủ để thực thi cho toàn ứng dụng. Bạn sẽ học sau cách đẩy dữ liệu dừng model binder mà không cần thay đổi đường dẫn mặc định.

**Kết luận**: Trong bài này, bạn đã học cách tạo 1 controller với các phương thức đơn giản và cả cách hiển thị dữ liệu từ các phương thức trên trình duyệt. Tuy nhiên, cách hiển thị này chưa đủ và đáp ứng nhu cầu, các bài tiếp theo sẽ giới thiệu về View (cách hiển thị) và Model (mô hình lấy dữ liệu từ nguồn dữ liệu như database) để bạn có thể hiển thị giao diện Web và dữ liệu từ database đầy đủ hơn.

1. Tài liệu được lấy từ website https://www.dammio.com/ [↑](#footnote-ref-1)