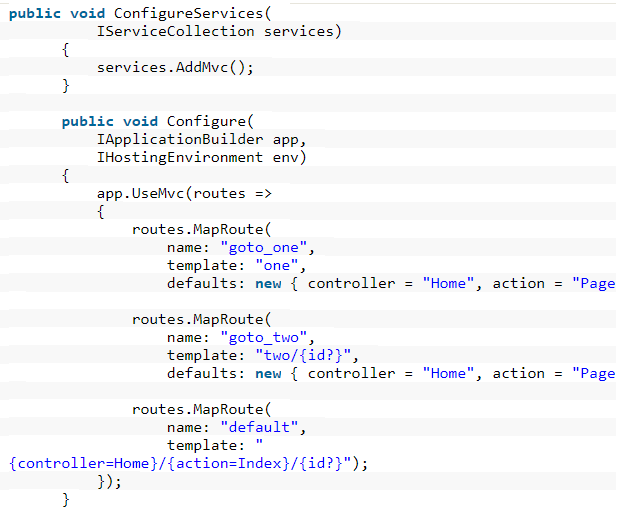
ASP.NET Core 2.0 MVC Routing

Để sử đụng routing trong Asp.net core ta thường thêm một lớp trung gian được định nghĩa sẵn là IroterBuilder để thực hiện Map Templete từ controller đến action.

Việc config thường thực hiện ở Startup Class và hàm configure.



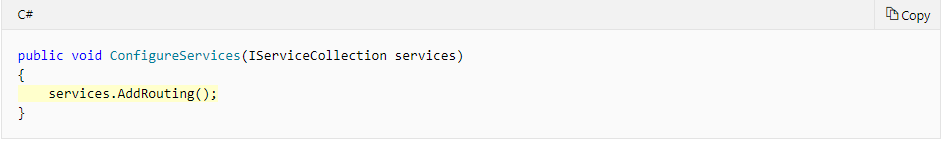
Routing trong asp.net core (MVC) thực hiện dựa trên việc ánh xạ các request đến controller gọi đến action của controller đó.

Thực hiện thông qua lớp trung gian sử dụng IroterBuilder để map đến templete của controller và action.

Cách sử dụng routing middleware

Thực hiện add [Microsoft.AspNetCore.App metapackage](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/metapackage-app?view=aspnetcore-2.2) vào app.

Thêm service routing vào Startup.ConfigureService.



Router sẽ được configured in the Startup.configure. Config router được thực hiện theo mẫu sau :

* [RouteBuilder](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.routing.routebuilder)
* [MapGet](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.routing.requestdelegateroutebuilderextensions.mapget) – Matches only HTTP GET requests.
* [UseRouter](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.builder.routingbuilderextensions.userouter)



Framework  cung cấp cho chúng ta một số phương thức mở rộng để tạo ra các router ([RequestDelegateRouteBuilderExtensions](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.routing.requestdelegateroutebuilderextensions)):

* [MapDelete](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.routing.requestdelegateroutebuilderextensions.mapdelete)
* [MapGet](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.routing.requestdelegateroutebuilderextensions.mapget)
* [MapMiddlewareDelete](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.routing.requestdelegateroutebuilderextensions.mapmiddlewaredelete)
* [MapMiddlewareGet](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.routing.requestdelegateroutebuilderextensions.mapmiddlewareget)
* [MapMiddlewarePost](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.routing.requestdelegateroutebuilderextensions.mapmiddlewarepost)
* [MapMiddlewarePut](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.routing.requestdelegateroutebuilderextensions.mapmiddlewareput)
* [MapMiddlewareRoute](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.routing.requestdelegateroutebuilderextensions.mapmiddlewareroute)
* [MapMiddlewareVerb](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.routing.requestdelegateroutebuilderextensions.mapmiddlewareverb)
* [MapPost](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.routing.requestdelegateroutebuilderextensions.mappost)
* [MapPut](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.routing.requestdelegateroutebuilderextensions.mapput)
* [MapRoute](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.routing.requestdelegateroutebuilderextensions.maproute)
* [MapVerb](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.routing.requestdelegateroutebuilderextensions.mapverb)

Phương thức Map[Verb] sẽ hạn giới hạn các kết nối tới HTTP verd ví dụ đọc thêm ở [MapGet](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.routing.requestdelegateroutebuilderextensions.mapget) và [MapVerb](https://docs.microsoft.com/dotnet/api/microsoft.aspnetcore.routing.requestdelegateroutebuilderextensions.mapverb).

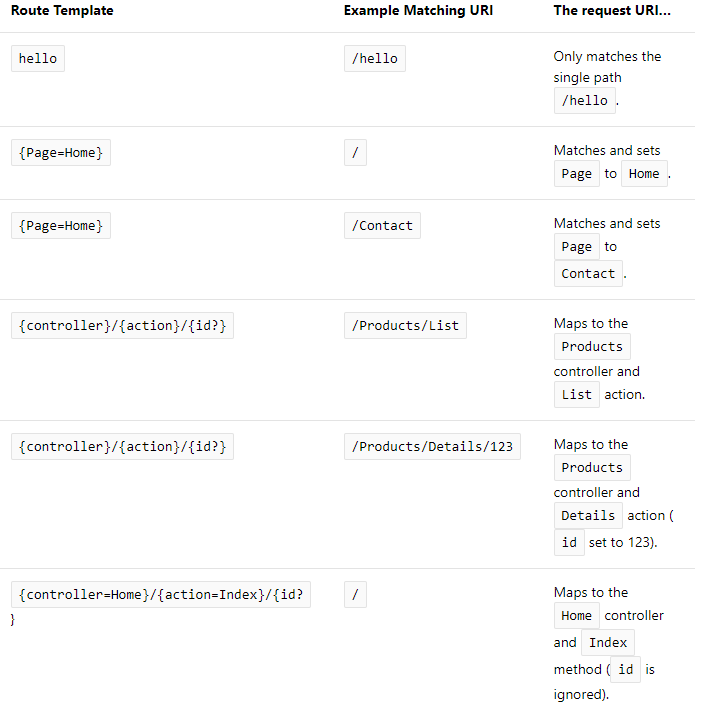
Route template reference

Để thực hiện định routing thì ta thực hiện định nghĩa ở trong đấu {…..}. Chúng ta có thể định nghĩa nhiều router parammater nhưng giữa các đấu ngoặc phải có đâu “/”

Cụ thể như sau:

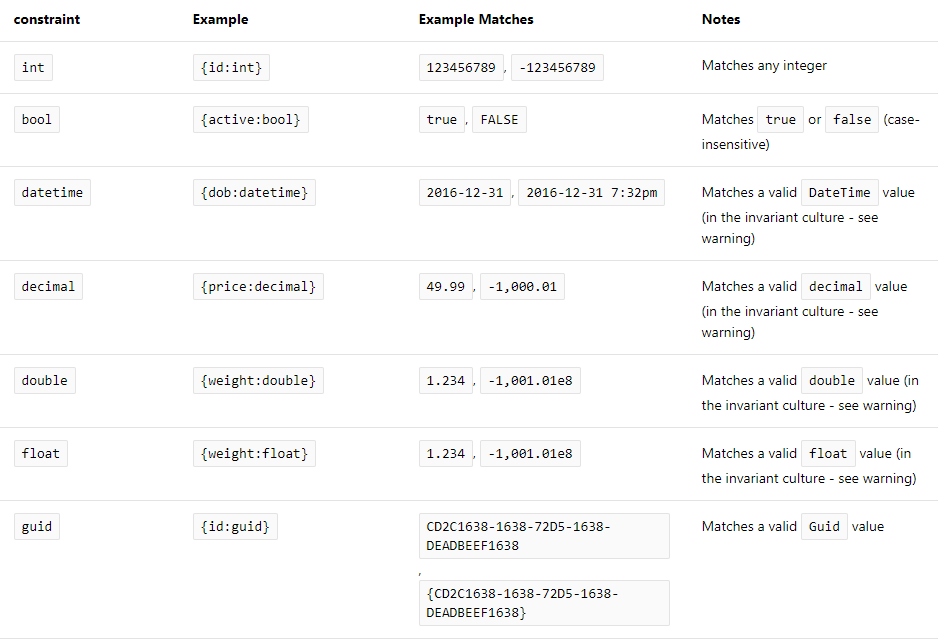
* {controller=Home}/{action=index}: đây là một router hợp lệ. và URL mặc định có giá trị “home/index”. Khi thực hiện gọi đến các controller khác ta thực hiện thay thế tên của controller và action theo đúng cấu trúc trên.
  + /home/index
* Url có thể lấy tên các file name và phần mở rộng của file. Ta có một route như sau “file/{filename}.{ext?}”. Nếu có nhiều file tồn tại trong “file” thì route sẽ lấy giá trị của file đó và phần mở rộng trên đưa lên url. Ngược lại nếu chỉ có một file thì url chỉ lấy tên của file vì phần sau “.” là phần tùy chọn.
  + /file/myfile.txt
  + /file/myFile
* Chúng ta có thể dùng (\*) hoặc (\*\*) để thực hiện lấy phần còn lại của parammater. Ví dụ {blog/\*\*bag} sẽ thực hiện tìm các param có giá trị của trước “bag”. Tất cả giá trị của param có thể chấp nhận chuỗi rỗng. Trở lại với ví dụ {blog/\*\*bag} với giá trị của {bag = “my/bag”} giá trị nhận được là “blog/my/bag”. Đối với trường hợp {blog/\*bag} với “bag=my/bag” ta sẽ được giá trị “blog/my%2bag”.
* Để truyền tham số vào url ta sử dụng đâu “?”
  + Home/index/id?
* Để ràng buộc giá trị nhập vào URL ta thực hiện thêm vào đấu “:” ví dụ: blog/{article:minlength(10)} có nghĩa độ dài ngắn nhất của url bằng 10. Xem thêm tại [Route constraint reference](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/routing?view=aspnetcore-2.2#route-constraint-reference).

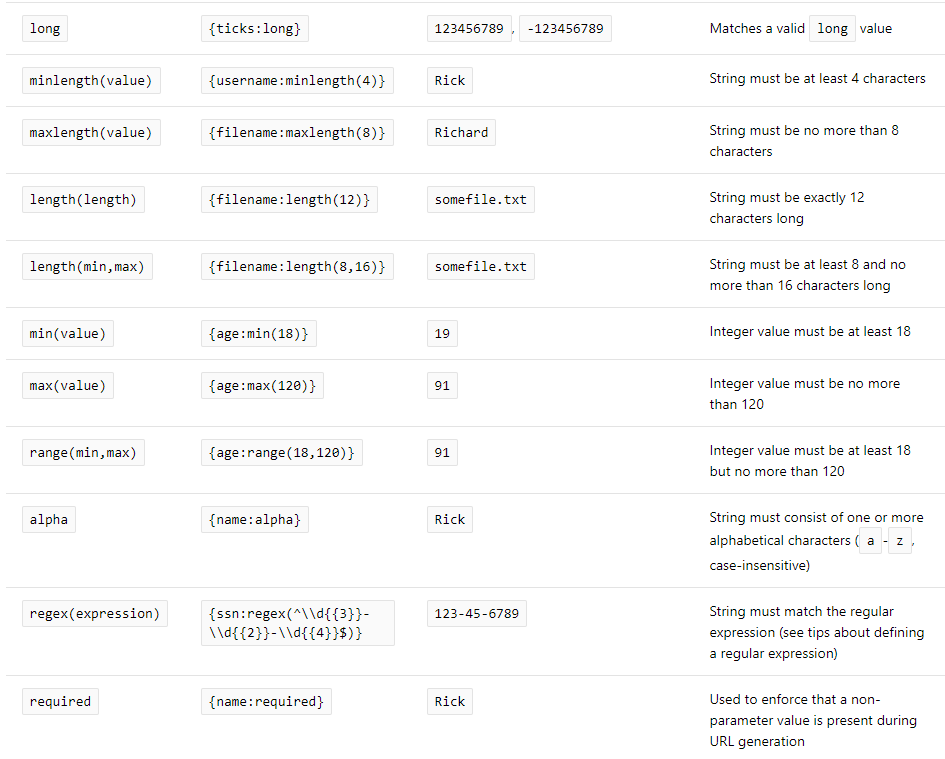
Ta có ví dụ như sau:



Route constraint reference

Như nói ở trên ta có thể ràng buộc giá trị của url. Ta có ví dụ về sử dụng ràng buộc của URL





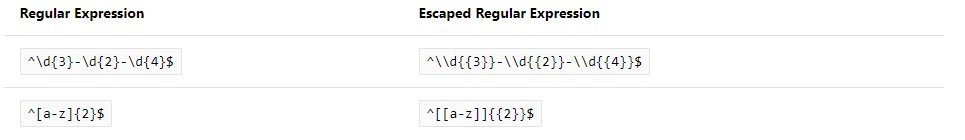
Sử dụng nhiều đâu “:” thường sử dụng trong việc ràng buộc tham số của url



## Regular expressions

Asp.net core đã thêm các thư viện như RegexOptions.IgnoreCase | RegexOptions.Compiled | RegexOptions.CultureInvariant hỗ trợ cho việc thực hiện quy định giá trị của URL

Việc sử dụng regualar expression giống như trong C#



Regular expressions được sử dụng trong routing được bắt đầu bằng (^) và kết thúc bằng ($) để đảm bảo chuỗi phù hợp với tất cả giá trị trong biểu thức mà không cần đầy đủ các yêu cầu của biểu thức. Nếu không có 2 ký tự trên thì chuỗi được so sánh phải có đầy đủ các yêu cầu quy định của biểu thức. Ví dụ ở bảng sau:



Tìm hiểu thêm thông tin về biểu thức chính quy (Regular Expression) ở: [.NET Framework Regular Expressions](https://docs.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference).