

# Sử dụng tag helper xây dựng sẵn của ASP.NET Core

[Hướng dẫn tự học lập trình ASP.NET Core toàn tập](#) > [Sử dụng tag helper xây dựng sẵn của ASP.NET Co...](#)

Tag helper là một tính năng mới trong ASP.NET Core cho phép tùy chỉnh HTML sinh ra từ mã Razor. Bạn đã học nguyên lý chung về Tag helper cũng như cách xây dựng [Tag helper trong ASP.NET Core](#) một bài học riêng.

Không chỉ vậy, ASP.NET Core tích hợp sẵn Tag Helper hướng tới hầu hết các thẻ HTML. ASP.NET Core khuyến khích sử dụng các Tag Helper tích hợp này trong mã Razor.

Trong bài học này chúng ta sẽ xem xét các Tag Helper hướng tới các thẻ HTML thường dùng nhất, bao gồm form, input, anchor, label và select. Đây là những thẻ bạn thường dùng khi xây dựng form.

## NỘI DUNG CỦA BÀI [ Ấn ]

1. Tag Helper xây dựng sẵn trong ASP.NET Core
2. Form Tag Helper và Anchor Tag Helper
3. Input Tag Helper
4. Select Tag Helper
5. Một số Tag Helper khác
  - 5.1. Label Tag Helper
  - 5.2. TextArea Tag Helper
6. Kết luận

## Tag Helper xây dựng sẵn trong ASP.NET Core

ASP.NET Core đã xây dựng Tag Helper cho hầu hết các thẻ HTML. Các Tag Helper này được gọi là Tag Helper xây dựng sẵn (built-in). Tuy nhiên, ASP.NET Core yêu cầu bạn phải chỉ định các Tag Helper sẽ sử dụng trên các trang Razor với directive `@addTagHelper` :

```
1. @addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
```

Thông thường bạn có thể đặt directive trên ở đầu mỗi trang Razor hoặc trong file `_ViewImports.cshtml` (nếu muốn sử dụng mặc định trên tất cả các trang Razor).

Khi sử dụng các template của ASP.NET Core như Web Application hay Web Application (Model-View-Controller), directive này đã được đặt sẵn trong file `_ViewImports.cshtml`. Do vậy bạn có thể sử dụng ngay Tag Helper xây dựng sẵn trên tất cả các trang Razor (trang nội dung của Razor Pages hoặc view của MVC).

Nếu muốn giới hạn, giả sử chỉ dùng Tag Helper cho thẻ a (anchor tag), bạn có thể điều chỉnh directive `@addTagHelper` như sau:

```
1. @addTagHelper Microsoft.AspNetCore.Mvc.TagHelpers.AnchorTagHelper, Microsoft.AspNetCore.
```

Tất cả Tag Helper xây dựng sẵn trong ASP.NET Core đều chứa các attribute bắt đầu bằng `asp-`. Ví dụ:

```
<a asp-action="Register" asp-controller="User">Register</a>
```

Trong Visual Studio Tag Helper được thể hiện bằng màu sắc đậm hơn so với thẻ HTML bình thường.

```
<form asp-action="Convert">
  <div class="form-group">
    <label asp-for="CurrencyFrom"></label>
    <input class="form-control" asp-for="CurrencyFrom" />
    <span asp-validation-for="CurrencyFrom"></span>
  </div>
  <div class="form-group">
    <label asp-for="Quantity"></label>
    <input class="form-control" asp-for="Quantity" />
    <span asp-validation-for="Quantity"></span>
  </div>
</form>
```

<https://tuhocict.com>

Như trong hình minh họa, thẻ form, label, input, span giờ là những Tag Helper chứ không còn là thẻ HTML bình thường nữa, mặc dù nhìn hình thức tương tự nhau.

Do Tag Helper và thẻ HTML tiêu chuẩn là hai loại đối tượng khác nhau, một số người muốn tách biệt chúng trên file Razor. ASP.NET Core cho phép chỉnh định tiền tố (prefix) sử dụng cùng Tag Helper với directive `@tagHelperPrefix`. Ví dụ, nếu sử dụng tiền tố `x:` sẽ viết như sau:

```
1. @tagHelperPrefix x:
```

Khi này bạn sử dụng Tag Helper như sau:

```
<x:a asp-action="Register" asp-controller="User">Register</x:a>
```

Khi đó, thẻ `<a></a>` sẽ không còn chịu tác động của Anchor Tag Helper nữa.

Tag Helper xây dựng sẵn của ASP.NET Core được đặt tên theo loại thẻ mà nó tác động: Form Tag Helper (thẻ form), Label Tag Helper (thẻ label), Input Tag Helper (thẻ input), Anchor Tag Helper (thẻ a), Select Tag Helper (thẻ select).

ASP.NET Core cũng xây dựng sẵn một số Tag Helper đặc biệt (không trùng tên với các thẻ HTML) như Validation Message Tag Helper, Validation Summary Tag Helper. Hai loại Tag Helper này sẽ được giới thiệu trong bài học về [thẩm định dữ liệu \(model validation\)](#).

### Về HTML helper và WebForms

Nếu bạn quen thuộc với ASP.NET MVC bạn hẳn biết về HTML helper. HTML helper vẫn có mặt trong ASP.NET Core và bạn vẫn có thể tiếp tục sử dụng nó. Tuy nhiên khi ASP.NET Core chuyển sang đa nền tảng và có nhiều editor khác ngoài Visual studio, HTML helper thể hiện nhiều nhược điểm. Tag Helper ra đời nhằm thay thế cho HTML helper. ASP.NET Core khuyến khích sử dụng Tag Helper. Vì vậy trong loạt bài học này chúng ta không học HTML helper mà chỉ tập trung vào Tag Helper.

Đối với những bạn quen thuộc với WebForms, mặc dù nhìn có nét tương tự, Tag Helper hoàn toàn không có điểm gì liên quan với WebForms server control.

Trên thực tế qua các bài học trước đây bạn đã sử dụng hầu hết các Tag Helper thông dụng. Trong bài học này chúng ta sẽ hệ thống lại chúng và đưa ra một số lưu ý khi sử dụng.

## Form Tag Helper và Anchor Tag Helper

Tag Helper dành cho Form (thẻ form) và Anchor (thẻ a) có chung các attribute sau đây:

Attribute	Ý nghĩa
asp-action	(MVC) chỉ định <a href="#">action</a> sẽ xử lý form khi nó được submit
asp-all-route-data	Chỉ định nhiều giá trị tham số chứa trong route hoặc query string (ghi chú 1)
asp-antiforgery	Chỉ định sử dụng chuỗi anti-forgery (ghi chú 2)
asp-controller	(MVC) chỉ định <a href="#">controller</a> chứa action xử lý form
asp-page	(Razor Pages) chỉ định page sẽ xử lý form khi nó được submit
asp-page-handler	(Razor Pages) chỉ định <a href="#">named-handler</a> sẽ xử lý form
asp-route-*	Chỉ định giá trị tham số chứa trong route (ghi chú 3)
asp-route	(MVC) chỉ định tên route template được sử dụng (nếu định nghĩa nhiều route template) trong Configure.

### Ghi chú 1

Để sử dụng `asp-all-route-data` bạn phải khai báo một từ điển để lưu các cặp khóa – giá trị:

```
1. @{
2.     var d = new Dictionary<string, string>
3.     {
4.         { "key1", "value1" },
5.         { "key2", "value2" }
6.     };
7. }
8. <form asp-all-route-data="d">...</form>
```

`asp-all-route-data` sẽ giúp điều chỉnh URL trong attribute action `<form action="/Page?key1=value1&key2=value2" method="post">...</form>`.

### Ghi chú 2

Anti-forgery token là cơ chế giúp [bảo vệ form khỏi tấn công CSRF](#) (Cross-site request forgery). Cơ chế này được sử dụng mặc định với form trong MVC hoặc Razor Pages.

`asp-antiforgery="false"` cho phép tắt chế độ kiểm tra trên form.

### Ghi chú 3

Giả sử bạn sử dụng route template mặc định của MVC ({controller}/{action}/{id?}), bạn có thể truyền giá trị id vào Url thông qua `asp-route-id="1"`. Khi này Tag Helper sẽ giúp bạn chèn giá trị "1" vào vị trí của tham số id trong Url.

Ở vị trí \* trong `asp-route-*`, bạn có thể đặt tên tham số của route. Attribute nhận giá trị tương ứng của tham số. Form Tag Helper sẽ sử dụng tên tham số và giá trị để tạo ra Url phù hợp cho bạn.

## Input Tag Helper

Input Tag Helper tác động lên thẻ `<input />`. Đây là Tag Helper hữu ích hàng đầu khi làm việc với form. Tuy nhiên Input Tag Helper chỉ có 1 attribute duy nhất: `asp-for`.

Tuy chỉ có 1 attribute, Input Tag Helper phân tích thông tin nó nhận được từ `asp-for` (bao gồm cả việc đọc Data Annotation của model class) và tự động sinh ra HTML phù hợp nhất cho thẻ input đó.

Ví dụ với domain class sau đây:

```
1. public class Member
2. {
3.     public int PersonId { get; set; }
4.     public string Name { get; set; }
5.     [EmailAddress]
6.     public string Email { get; set; }
7.     [DataType(DataType.Password)]
8.     public string Password { get; set; }
9.     [DataType(DataType.PhoneNumber)]
10.    public string Telephone { get; set; }
11.    [Display(Name="Date of Birth")]
12.    public DateTime DateOfBirth { get; set; }
13.    public decimal Salary { get; set; }
14.    [Url]
15.    public string Website { get; set; }
16.    [Display(Name="Send spam to me")]
17.    public bool SendSpam { get; set; }
18.    public int? NumberOfCats { get; set; }
19.    public IFormFile Selfie { get; set; }
20. }
```

Nếu xây dựng form nhập dữ liệu cho Member bạn có thể sử dụng Input Tag Helper trong form như sau:

```
1. <form method="post">
2.     <input asp-for="Member.PersonId" /><br />
3.     <input asp-for="Member.Name" /><br />
4.     <input asp-for="Member.Email" /><br />
5.     <input asp-for="Member.Password" /><br />
6.     <input asp-for="Member.Telephone" /><br />
7.     <input asp-for="Member.Website" /><br />
8.     <input asp-for="Member.DateOfBirth" /><br />
9.     <input asp-for="Member.Salary" /><br />
10.    <input asp-for="Member.SendSpam" /><br />
11.    <input asp-for="Member.NumberOfCats" /><br />
12.    <input asp-for="Member.Selfies" /><br />
13.    <button type="submit">Submit</button>
14. </form>
```

Loại Input Tag Helper trên sẽ sinh ra HTML như sau:

```

1. <form method="post">
2.   <input type="number" data-val="true" data-val-required="The PersonId field is requir
3.   <input type="text" id="Member_Name" name="Member.Name" value="" /><br />
4.   <input type="email" data-val="true" data-val-email="The Email field is not a valid e
5.   <input type="password" id="Member_Password" name="Member.Password" /><br />
6.   <input type="tel" id="Member_Telephone" name="Member.Telephone" value="" /><br />
7.   <input type="url" data-val="true" data-val-url="The Website field is not a valid ful
8.   <input type="datetime-local" data-val="true" data-val-required="The Date of Birth fi
9.   <input type="text" data-val="true" data-val-number="The field Salary must be a numbe
10.  <input data-val="true" data-val-required="The Send spam to me field is required." id
11.  <input type="number" id="Member_NumberOfCats" name="Member.NumberOfCats" value="" />
12.  <input type="file" id="Member_Selfie" name="Member.Selfie" /><br />
13.  <button type="submit">Submit</button>
14.  <input name="__RequestVerificationToken" type="hidden" value="xxxxx.." />
15.  <input name="Member.SendSpam" type="hidden" value="false" />
16. </form>

```

Bạn có thể để ý:

Input Tag Helper sinh thêm một số attribute lạ `data-val-`. Các attribute này được sinh ra khi bạn sử dụng data annotation ([EmailAddress], [DataType(DataType.Password)], [Url], v.v.) cho property của model class. Chúng giúp thực hiện [thẩm định dữ liệu](#) ở client sử dụng jQuery. Để sinh ra các thông tin này, Input Tag Helper sẽ tự phân tích các data annotation sử dụng cho property cung cấp qua asp-for.

Input Tag Helper tự động sinh id và name cho thẻ input dựa trên tên trường cung cấp cho asp-for. Giá trị của name trùng với tên property để sau cơ chế [model binding](#) sử dụng.

Input Tag Helper tự động xác định giá trị của attribute type dựa trên kiểu của dữ liệu nó nhận được. Ví dụ, property NumberOfCats có kiểu int thì tương ứng type="number". Bảng sau là danh sách kiểu .NET tương ứng với giá trị của type:

.NET Type	Input type
bool	checkbox
byte, short, int, long	number
decimal, double, float	text
string	text
DateTime	datetime-local
IFormFile	file

Input Tag Helper cũng xác định giá trị của type dựa trên data annotation của property theo quy tắc sau:

DataType Enumeration	Attribute	Input type
EmailAddress	EmailAddress	email
PhoneNumber	Phone	tel
Password		password
Url	Url	url
Date		date
Time		time
DateTime, Duration		datetime-local
HiddenInput		hidden

# Select Tag Helper

Select Tag Helper giúp tạo ra combo box (hộp chọn) từ một danh sách các phần tử có kiểu SelectListItem.

Hãy cùng xem ví dụ sau (trên Razor Pages):

```
1. @page
2.
3. <div class="text-center">
4.     <form asp-page="Index">
5.         <select asp-for="Number" asp-items="numbers">
6.             <option value="">Pick a number</option>
7.         </select>
8.         <select asp-for="Day" asp-items="Html.GetEnumSelectList<DayOfWeek>()">
9.             <option value="">Pick a day</option>
10.        </select>
11.        <button type="submit">Submit</button>
12.    </form>
13. </div>
14.
15. @functions {
16.     List<SelectListItem> numbers = Enumerable.Range(1, 5)
17.         .Select(n => new SelectListItem {
18.             Value = n.ToString(),
19.             Text = n.ToString()
20.         }).ToList();
21.
22.     [BindProperty]
23.     public int Number { get; set; }
24.
25.     [BindProperty]
26.     public int Day { get; set; }
27.
28.     public void OnPost() {
29.         Console.WriteLine($"Number: {Number}, Day: {Day}");
30.     }
31. }
```

Trong ví dụ trên bạn tạo ra hai danh sách:

(1) numbers là danh sách các số từ 1 đến 5: Danh sách này được tạo ra trong khối @functions của page Index. Lưu ý rằng danh sách này có kiểu là IEnumerable<SelectListItem>. Trong đó kiểu phần tử của danh sách bắt buộc phải là SelectListItem. Select Tag Helper giúp tự động xuất danh sách này thành thẻ <select> và các thẻ <option> bên trong như sau:

```
1. <select data-val="true" data-val-required="The Number field is required." id="Number" name="Number">
2.     <option value="">Pick a number</option>
3.     <option value="1">1</option>
4.     <option value="2">2</option>
5.     <option value="3">3</option>
6.     <option value="4">4</option>
7.     <option value="5">5</option>
8. </select>
```

(2) danh sách tên các ngày trong tuần: Danh sách này được tạo ra qua phương thức hỗ trợ Html.GetEnumSelectList<DayOfWeek>(). Phương thức này trả về dữ liệu có kiểu IEnumerable<SelectListItem>.

```
1. <select data-val="true" data-val-required="The Day field is required." id="Day" name="Day">
2.     <option value="">Pick a day</option>
```

```
3. <option selected="selected" value="0">Sunday</option>
4. <option value="1">Monday</option>
5. <option value="2">Tuesday</option>
6. <option value="3">Wednesday</option>
7. <option value="4">Thursday</option>
8. <option value="5">Friday</option>
9. <option value="6">Saturday</option>
10. </select>
```

Qua ví dụ trên bạn có thể thấy:

Select Tag Helper sử dụng attribute `asp-for` để xác định biến/property dùng trong model binding.

Attribute `asp-items` nhận một danh sách `SelectListItem` và xuất danh sách này thành các thẻ `<option>` tương ứng. `SelectListItem` có hai property được sử dụng khi sinh option là `Value` (dùng cho attribute `value` của option) và `Text` (là giá trị của mỗi option).

Nếu muốn chỉ định phần tử nào sẽ được chọn mặc định, bạn thiết lập property `Selected` của `SelectListItem` thành `true`. Như với danh sách `numbers` trên bạn có thể chỉ định `numbers[1].Selected = true`.

## Một số Tag Helper khác

### Label Tag Helper

Label Tag Helper chỉ có một attribute `asp-for` và nhận tên của property. Label Tag Helper tự động phân tích data annotation để lấy thông tin hỗ trợ hiển thị cho thẻ label.

Lấy ví dụ, nếu property `Email` được khai báo như sau:

```
1. [Display(Name = "Địa chỉ email")]
2. public string Email { get; set; }
```

Khi này Tag Helper `<label asp-for="Email"></label>` sẽ được chuyển thành HTML `<label for="Email">Địa chỉ Email</label>`.

Nếu property không sử dụng `Display` annotation, Label Tag Helper sẽ sử dụng luôn tên của property.

Label Tag Helper rất thường xuyên được sử dụng cùng Input Tag Helper khi xây dựng form.

### TextArea Tag Helper

TextArea Tag Helper điều chỉnh cho thẻ `textarea` cũng chỉ có một attribute `asp-for`. Attribute này nhận giá trị là property của model.

TextArea Tag Helper cũng phân tích data annotation sử dụng cho property để sinh ra HTML phù hợp và hỗ trợ thẩm định dữ liệu.

Ví dụ:

```
1. [BindProperty, MaxLength(300)]
2. public string MainText { get; set; }
```

Bạn sử dụng Textarea Tag Helper cho MainText như sau:

```
1. <textarea asp-for="MainText"></textarea>
```

HTML sinh ra có dạng như sau:

```
1. <textarea
2.     data-val="true"
3.     data-val-maxlength="The field MainText must be a string or array type with a maximum
4.     data-val-maxlength-max="300"
5.     id="MainText"
6.     name="MainText">
```

## Kết luận

Trong bài học này chúng ta đã hệ thống các Tag Helper xây dựng sẵn thông dụng của ASP.NET Core, bao gồm Form, Anchor, Input, Select, Label, Textarea. Đây là những Tag Helper quan trọng nhất giúp xây dựng form thu thập dữ liệu.

Bạn có thể tìm hiểu cách sử dụng các Tag Helper khác từ tài liệu của Microsoft:

<https://docs.microsoft.com/en-us/aspnet/core/mvc/views/tag-helpers/built-in/?view=aspnetcore-3.1>.

- + Nếu bạn thấy site hữu ích, trước khi rời đi hãy **giúp đỡ** site bằng một hành động nhỏ để site có thể phát triển và phục vụ bạn tốt hơn.
  - + Nếu bạn thấy bài viết hữu ích, hãy giúp **chia sẻ** tới mọi người.
  - + Nếu có thắc mắc hoặc cần trao đổi thêm, mời bạn viết trong phần **thảo luận** cuối trang.
- Cảm ơn bạn!