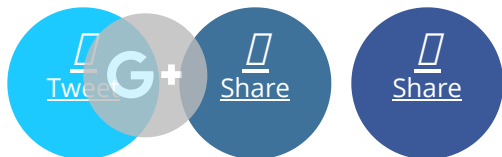


CodingBlast

[Blog posts](#) / [Hire me](#) / [Contact](#) / [About](#) / [Categories](#)

[Home](#) >> [ASP.NET Core](#) >> [ASP.NET Core MVC – All about Tag Helpers](#)

ASP.NET Core MVC – All about Tag Helpers



This entry is part 1 of 5 in the [ASP.NET Core Tag Helpers](#) series

All posts in [ASP.NET Core Tag Helpers](#) series:

- ♦ [ASP.NET Core MVC – All about Tag Helpers](#)
- ♦ [ASP.NET Core MVC – Caching Tag Helpers](#)
- ♦ [ASP.NET Core MVC – Form Tag Helpers](#)
- ♦ [ASP.NET Core MVC – Custom Tag Helpers](#)
- ♦ [ASP.NET Core MVC – Tag Helper Components](#)

Get Free updates!
ASP.NET Core, Angular,
React.js, SignalR!

Sign Up!

POPULAR POSTS


- [EntityFramework Core implementation of IDesignTimeDbContext](#)
- [ASP.NET Core SignalR C React.js](#)
- [Entity Framework Core Repository](#)

Introduction

Tag Helpers provide us with means to change and enhance existing HTML elements in our views. Therefore, we add them to our views, and afterwards, they are processed by the Razor Templating engine which in return creates an HTML and serves it to the browser. Few Tag Helpers actually act as elements or actual tags (environment, cache, etc.).

They make working with HTML while using the power of Razor and C# so simple and HTML friendly. Since using Tag Helpers feels so natural and looks like regular HTML front-end people can also quickly jump in and change things without learning about C#. Also, they get to have nice IntelliSense support within the scope of existing elements.

Simple example, do notice the difference:



```
  

```

Visual Studio automatically colours the *img* element because we used Tag Helper inside of it.

Why the name Tag Helper? Well, we refer to HTML elements as tags (<div>, <body>,) and we want to improve and possibly change the behaviour of these HTML elements.

They might remind you of HTML Helpers in Razor. Indeed, they share some of the features. However, Tag Helpers are much more natural to use. We will most usually use them to extend existing HTML elements. We can also use HTML Helpers to create HTML elements from scratch.

- ASP.NET Core SignalR C Angular 5
- nameof expression is a NO to magic strings!

Categories

- .NET Core (4)
- Angular (4)
- ASP.NET Core (33)
 - SignalR (3)
 - Tag Helpers (6)
- C# (5)
- Entity Framework Core
- JavaScript (7)
- React (2)
- RxJS (6)
- TypeScript (5)

Tags

Another thing they remind me of is AngularJS directives. In AngularJS 1 (before components came to AngularJS 1.5) two most common types of directives were an element and attributed directives. Element directives would create a new DOM element and attribute directives would change the behaviour of existing DOM element. The main difference, of course, is that Tag Helpers will be processed on the server and AngularJS works in the browser.

Ways of using TagHelpers

We can:

- ◆ Use them as HTML attribute
- ◆ Or as an HTML tag/element
- ◆ Or we use them as a child element of existing HTML element

HTML Attribute:

We can use built-in, or custom made TagHelpers as part of existing HTML elements. That's usually in a role of an attribute.





#.NET Core #angular #ai
#ASP.NET Core
#ASP.NET Core Best Pract
#ASP.NET Core Configurat
#ASP.NET Core EntityFram
#ASP.NET Core MVC
#ASP.NET Core SignalR
#ASP.NET Core Tips
#ASP.NET Core Tricks #as
#C# 7 #chat #decorators
#Entity Framework
#Entity Framework Core
#generators
#IDesignTimeDbContextF
#iterators #javascript #n
#node #node.js #observ
#observer #Razor #Razc
#React #reactivex #real-
#rxjs operators #selectec
#selected-post #SignalR
#Tag Helper Component
#Tag Helpers #tuples #t
#typescript #websocket
#websockets

```

```

More about image tag helper later in the post!

HTML Element:



```
<super-cool-element></super-cool-element>  
<environment names="Development">  
  <script src="~/lib/jquery/dist/jquery.js"></script>  
</environment>
```

On the image, we can see a custom 'super-cool-element' tag helper as custom HTML tag. We can also see a built-in Tag Helper **environment**. It is quite useful for purposes of rendering content based on current environment.

List of built-in Tag Helpers

[Form Related Tag Helpers:](#)

- ◆ Form
- ◆ FormAction
- ◆ Input
- ◆ Label
- ◆ Option
- ◆ Select
- ◆ TextArea
- ◆ ValidationMessage
- ◆ ValidationSummary

Caching

- ◆ Cache
- ◆ Distributed

Others:

- ◆ Image
- ◆ Anchor
- ◆ Script

- ♦ Link
- ♦ Environment

It is important to know that all built-in Tag Helpers are prefixed with *asp-* prefix.

In this post, we will see some of the Tag Helpers in action. To be precise, Tag Helpers from *Others* category. However, in the next [few posts](#) expect to see most important [Form Tag Helpers](#) covered. Also, we will cover most important details about [caching Tag Helpers](#).

Image Tag Helper

This Tag Helper makes sure that the picture that will display is the latest image from our file system.

In this example, we have image tag helper which appends version to the provided image tag:

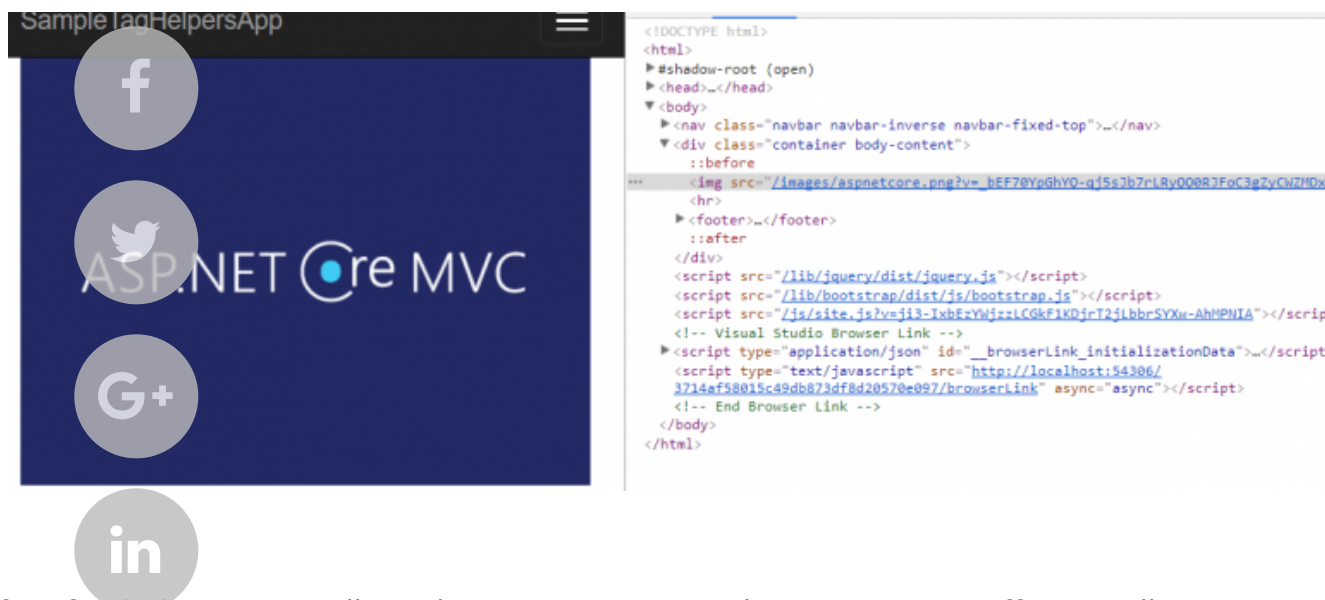
```

```

The code above indicates that every time we change the actual picture on the server (*~/images/aspnetcore.png*) the image tag helper will append a different string to it and the new image will not be cached so users can see a new image every time the picture is changed.

However, if the picture is not changed on the server, we will still be getting a cached version of the image.

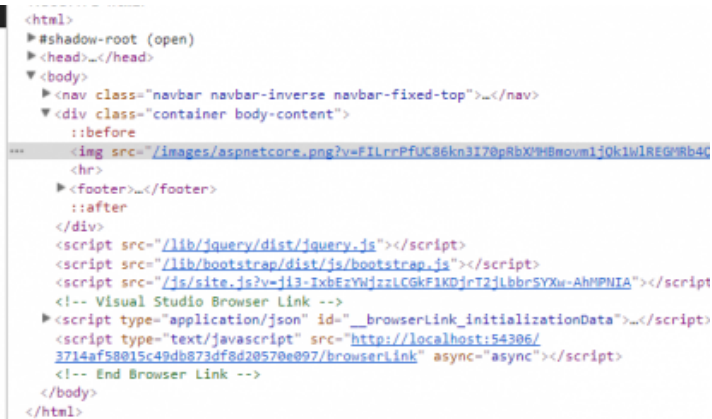
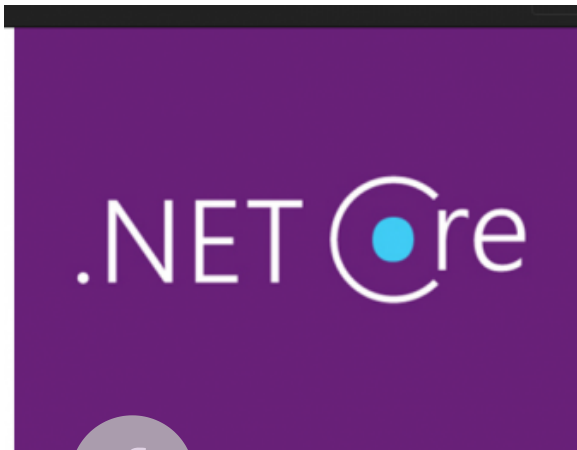
Let us see what happens:



If I refresh the page, I will get the same content and same version suffix as well.

While the MVC app is running, I will copy new image with the same file name to the images folder.

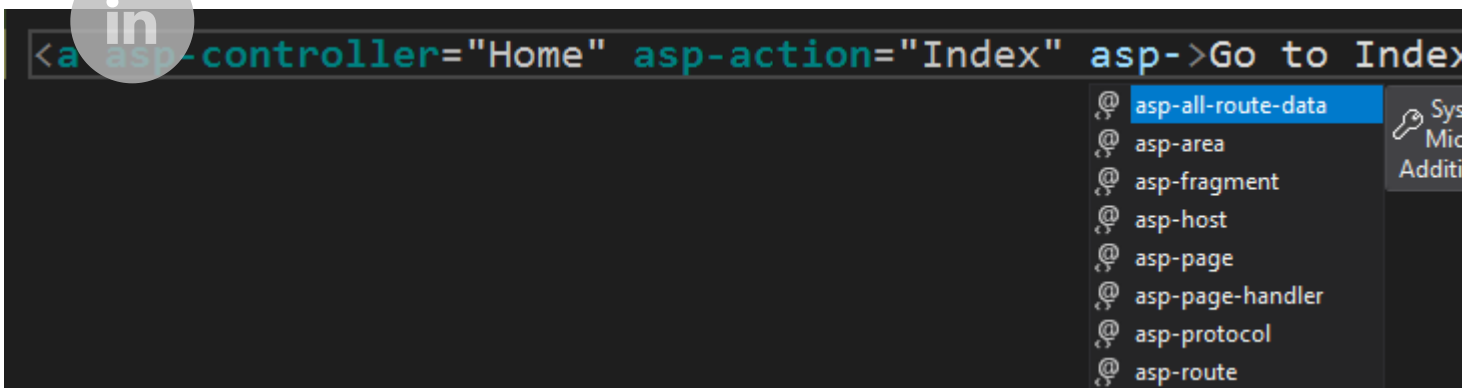
If I refresh the page now:



Everything is the same, except the version suffix.

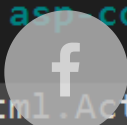


Anchor Tag Helpers




The famous **a** or **anchor** tag can be extended with various tag helpers. Two tag helpers that you will probably use will be *asp-controller* and *asp-action*. These will help us to create links to a specific Action method quickly.


If you are familiar with MVC, you probably used `@Html.ActionLink` helpers to achieve the same result for the anchor tag.



```
<a asp-controller="Home" asp-action="Index" asp->Go to Index</a>  
@Html.ActionLink("Go to Index", "Index", "Home")
```



In this particular example, the old way might seem shorter and easier. However, the new way is much more natural since we are extending existing HTML tags with 'custom attributes' or TagHelpers. Furthermore, it is more clear, since we can easily see which part is the action and which one is the controller.



in

Script and Link Tag Helpers

These two make adding JS and CSS references to our HTML much easier. They also enable us to provide a local fallback to URL if we are using an external resource like a CDN.

Furthermore, they greatly simplify referencing of a vast number of files using globbing patterns.

Just like image Tag Helper, Script and Link Tag Helpers can help us with caching of local files.

```
<script src="https://ajax.aspnetcdn.com/ajax/bootstrap/3.3.7/bootstrap.min.js"
asp-append-version="true"
asp-fallback-src="~/lib/bootstrap/dist/js/bootstrap.min.js"
asp-fallback-test="window.jQuery && window.jQuery.fn && window.jQuery.fn.modal">
</script>
```

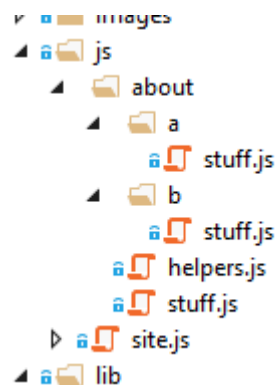
This example shows that we are referencing Bootstrap's JS file from *aspnetcdn*. If that site is not available the local file will be referenced instead. Hence, the *asp-fallback-src* Tag Helper. What about *asp-fallback-test*? Well, that is a way of knowing if we will use fallback source or not. In this case, we use a specific code that is used to check if Bootstrap's JS functionality is present.

If expression inside of *asp-fallback-test* returns false(falsy) then source from *asp-fallback-src* will be used.

Referencing multiple files – *asp-src-include*, *asp-src-exclude*

Referencing multiple files is one of the features that I like the most about TagHelpers. It saves me time to reference all or multiple files from a particular folder and its children sub folders.

Let us see an example. I have these folders and files under **js** folder in **wwwroot**:



Now, let us see what happens when I use *asp-src-include* TagHelper to reference multiple files from my About.cshtml view:

```
<script asp-src-include="/js/about/*.js"></script>
```

Now when I run the app and I inspect the contents of the page:

```
<div class="container body-content">
  ::before
  <script src="/js/about/helpers.js"></script>
  <script src="/js/about/stuff.js"></script>
  ...
</div>
```

Sweet! I don't have to reference files, one by one manually. I can use glob patterns and reference multiple folders and also exclude some of them as well.


Let us see how I can use *asp-src-exclude* TagHelper:

```
<script asp-src-include="/js/about/**/*.*.js"  
        asp-src-exclude="/js/about/a/**/*.*.js"></script>
```


Now the output is:



```
::before  
<script src="/js/about/helpers.js"></script>  
<script src="/js/about/stuff.js"></script>  
<script src="/js/about/b/stuff.js"></script>  
/b/
```



I got all js files from **about** folder with the exclusion of **a** folder. With only one script tag we manage to generate so much HTML. Say goodbye to gulp, grunt or whatever you used to reference your JS files from HTML!



All this makes me happy because that is yet another thing I'm not going to miss when not developing Angular / React applications in favour of developing with ASP.NET Core MVC.

Link Tag Helper

Link Tag Helper has almost all the same functionality except it has *href* attribute instead of *src* and it has few different fallback test attributes:

```
<link rel="stylesheet"
      href="http://ajax.aspnetcdn.com/ajax/bootstrap/3.3.7/css/bootstrap.css"
      asp-fallback-href="~/lib/bootstrap/dist/css/bootstrap.css"
      asp-fallback-test-class="hidden"
      asp-fallback-test-property="visibility"
      asp-fallback-test-value="hidden" />
```



This code will check for class, property and value. If any of those are not found then the local Bootstrap file will be used.



Environment Tag Helper



Environment Tag Helper will render the content which it wraps based on the environment that application is running in.

Let us say we have the following code:

```
<environment include="Development">  
    This will run in Development <br />  
</environment>
```

```
<environment include="Production">  
    This will run in Production <br />  
</environment>
```

```
<environment exclude="Production">  
    This will run in every environment except Production <br />  
</environment>
```

```
<environment names="Development,Staging">  
    This will run in Development and in Staging <br />  
</environment>
```

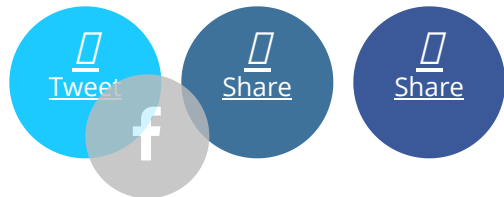
Running on my local machine in development environment produces following page output:

```
This will run in Development  
This will run in every environment except Production  
This will run in Development and in Staging
```

[ASP.NET Core Tag Helpers](#) Series Navigation:





Previous post:

Next post: [ASP.NET Core MVC – Caching Tag Helpers >>](#)



Posted by Ibrahim Šuta
in ASP.NET Core , Tag Helpers

Related Posts

  [ASP.NET Core Dependency Injection – Cannot consume scoped service](#) [Hosting ASP.NET Core on Heroku \(with Docker&CircleCI\) for free](#) [ASP.NET Core Authorization – Using attributes for handlers and requirements](#)

ASP.NET Core 2 – Global Model validation

📄 ASP.NET Core – Unit testing `IAuthorizationService` and dependencies inside of your controller

📄 ASP.NET Core – `InvalidOperationException`: Cannot find compilation library location for package 'Microsoft.Win32.Registry'



[Ibrahim Šuta](#)

Software Consultant interested and specialising in ASP.NET Core, C#, JavaScript, Angular, React.js.



Sponsored**Many failed before. Will you complete the trial?**

Hero Wars

!احصل على بطاقة خضراء أمريكية. قدم الآن

Global USA

**Join egFWD's Free initiative for a chance to increase your income**

egFWD's Initiative from MCIT

**Play this Game for 1 Minute and see why everyone is addicted**

Desert Order

**These Wedding Moments Are Unforgettable For All The Wrong Reasons**

Gadgettheory

**Amazing Adjustable Reading Glasses for Seniors!**

Self Adjusting Glasses

The World's Worst Tattoo Fails: Try Not to Laugh**ALSO ON CODINGBLAST****ASP.NET Core MVC –
Tag Helper ...****ASP.NET Core – ...****ASP.NET Core SignalR
Chat with Angular 5****Why you should
ASP.NET Core**

3 years ago • 4 comments

Tag Helper Components work in conjunction with Tag Helpers. They let us to ...

3 years ago • 1 comment

TL;DR Add following line to your .

3 years ago • 7 comments

ASP.NET Core SignalR chat with latest Angular. Building a small chat application ...

3 years ago • 1 comment

What is .NET Core? ASP.NET Core? better, easier to

4 Comments CodingBlast  Disqus' Privacy Policy

 Login ▾

 Recommend

 Tweet

 Share

Sort by Best ▾



 Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 



Name



Mohammad • 3 years ago

JazakAllah Khair for this fine article.

Would it be possible for you to make printer friendly version of your articles, since I like to print them out and then read them.



• Reply • Share ▾



MichaelDennis • 3 years ago

Where is the Microsoft reference page that documents all the tag helpers?

^ | ▾ • Reply • Share ▾



CodingBlast Admin → MichaelDennis • 3 years ago

There is this page <https://docs.microsoft.com/...>

And the one for the form Tag Helpers <https://docs.microsoft.com/...>

^ | ▾ • Reply • Share ▾



MichaelDennis → CodingBlast • 3 years ago

Thanks very much.

^ | v • Reply • Share ›

 [Subscribe](#)  [Add Disqus to your site](#) [Add Disqus](#) [Add](#)  [Do Not Sell My Data](#)

Sponsored

Many failed before. Will you complete the trial?

Hero Wars



!احصل على بطاقة خضراء أمريكية. قدم الآن

Global USA



Get a Free Udacity Certificate for a chance to increase your income

egFWD's Initiative from MCIT



6 Awesome Workout Exercises for 6 Pack Abs - Active Health 101

Active Health 101



Play this Game for 1 Minute and see why everyone is addicted

Desert Order

These Wedding Moments Are Unforgettable For All The Wrong Reasons

Gadgettheory

[←Previous post](#)

[Next post→](#)

[Blog posts](#) / [Hire me](#) / [Contact](#) / [About](#) / [Categories](#)

Copyright text 2017 by CodingBlast.

