

ASP.NET 5 Tag Helpers

By Franc Stratton

Tag Helpers Introduction

- Tag Helpers enable server-side code to participate in creating and rendering HTML elements in Razor files.
- Tag Helpers are authored in C#, and they target HTML elements based on element name, attribute name, or parent tag.
- For example, the built-in [ImageTagHelper](#) can append a version number to the image name. Whenever the image changes, the server generates a new unique version for the image, so clients are guaranteed to get the current image (instead of a stale cached image).
- There are many built-in Tag Helpers for common tasks - such as creating forms, links, loading assets and more - and even more available in public GitHub repositories and as NuGet packages.

Tag Helper Defined with an Example

- A tag helper is any class that implements the **ITagHelper** interface. However, when you author a tag helper, you generally derive from **TagHelper**, doing so gives you access to the **Process** method.
- E.g., A tag helper that updates an email tag:
`<email>Support</email>`
- The server will use the new email tag helper to convert that markup into the following:
`franc.c.stratton@tn.gov`
- This anchor tag makes an email link and sends an email to me at the tn.gov domain. This will be an authoring demo later.

Tag Helpers vs. HTML Helpers

- Tag helpers replace HTML helpers from earlier versions of ASP.NET MVC
- Tag helpers use HTML helpers in views by extending tag semantics in markup rather than through Razor calls that disrupt the flow of HTML markup with embedded .NET code
- HTML output from tag helpers should in most cases be identical to that of HTML helpers

Tag Helpers vs. HTML Helpers (Cont'd)

What's a Tag Helper?

It's a set of code that allows you to no longer have to use Razor helpers to build your cshtml forms. What this means is that when you had to write this:

```
@Html.ActionLink("Add a Movie", "Add", "Movie")
```

You can now write this:

```
<a asp-action="Add" asp-controller="Movie">Add a Movie</a>
```

Because of this, you can now write views that look like HTML rather than an unwieldy mix of HTML and C#.

What Tag Helpers Provide

- Front-end designers conversant with HTML/CSS/JavaScript can edit Razor without learning C# Razor syntax.
- Even developers experienced with Razor C# syntax are more productive using Tag Helpers than writing C# Razor markup.
- Most of the built-in Tag Helpers target existing HTML elements and provide server-side attributes for the element.

Several built-in tag helpers

- Anchor (for generating hyperlinks)
- Cache (for managing partial page caching)
- Environment (for controlling content rendering based on the runtime environment)
- Form (for generating form elements)
- Input (generation of input elements)
- Label (outputs label elements)
- Link (processes link elements)
- Option (targets individual options in a select list)
- Script (processes script tags)
- Select (generates dropdown lists)
- TextArea (processes textarea tags)
- ValidationMessage (generates individual validation errors)
- ValidationSummary (renders the validation summary message)

Built-in Tag Helpers: First Demo

- Most of the built-in Tag Helpers target existing HTML elements and provide server-side attributes for the element. For example, the `<input>` element used in many of the views in the *Views/Account* folder contains the `asp-for` attribute, which extracts the name of the specified model property into the rendered HTML. The following Razor markup:

```
<label asp-for="Email"></label>
```

- Generates the following HTML:

```
<label for="Email">Email</label>
```

Tag Helper Scope

- Tag Helpers scope is controlled by a combination of `@addTagHelper`, `@removeTagHelper`, and the `!"` opt-out character.
- The `@addTagHelper` directive makes .NET Tag Helpers available to the view in the view file is (shown in demo) `Views/_ViewImports.cshtml` inherited by all view files in the `Views` folder.
- The code uses the wildcard syntax ("`*`") to specify that all Tag Helpers is available to all views:
`@addTagHelper "*, Microsoft.AspNetCore.Mvc.TagHelpers"`

Opting Out of Individual Elements

- You can disable a Tag Helper at the element level with the Tag Helper opt-out character ("!").
- For example, Email validation is disabled in the `` with the Tag Helper opt-out character:

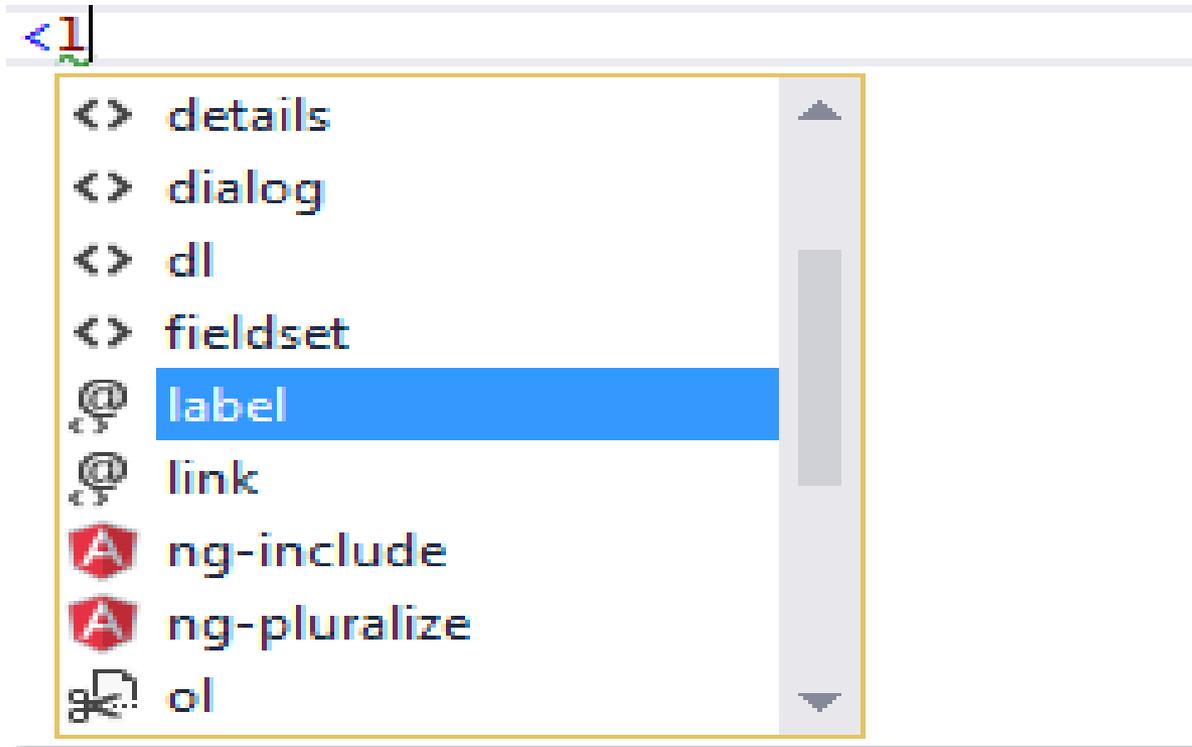
```
<!span asp-validation-for="Email" class="text-danger"></!span>
```
- **Note**: Apply the Tag Helper opt-out character to the opening and closing tags.

IntelliSense Support for Tag Helpers

- Creating a new ASP.NET web app in Visual Studio, it adds “Microsoft.AspNet.Tooling.Razor” to the *project.json* file.
- This is the package that adds Tag Helper tooling.
- (See *project.json* file in Demo code)

IntelliSense Support (Cont'd)

- Writing an HTML <label> element: as soon as you enter <l in the Visual Studio editor, IntelliSense displays matching elements:



Validation Summary Tag Helper

- The validation summary tag helper is used to display validation messages that apply to the entire MVC view (See Index View in Demo 1)

```
<div asp-validation-summary="ValidationSummary.All" class="text-danger"></div>
```
- The attribute can be `ValidationSummary.All` which display both property and model level validations messages while `ValidationSummary.ModelOnly` will display only validation messages that apply to the model level. If `ValidationSummary.None` is specified, the tag helper will do nothing which would be odd.

Form Tag Helper

- Form tag helper looks something like this:

```
<form asp-controller="Home" asp-action="Register" asp-anti-forgery="true" method="post" class="form-horizontal" role="form">
```
- Notice the "asp-anti-forgery" property is set to true and is the same as using the `@Html.AntiForgeryToken` helper in Razor ([which you should always be using](#)).
- Now run demo 1 and see the Index view

Authoring Tag Helpers: 2nd Demo

- Add the EmailTagHelper class to the TagHelpers folder.
- Tag helpers use a naming convention that targets elements of the root class name (minus the TagHelper portion of the class name).
- In the demo example, the root name of EmailTagHelper is email, so the <email> tag will be targeted.
- This naming convention should work for most tag helpers.
- The EmailTagHelper class derives from TagHelper.

Tag Helper Process Method

- The overridden Process method controls what the tag helper does when executed.
- The TagHelper class also provides an asynchronous version (ProcessAsync) with the same parameters.
- The context parameter to Process (and ProcessAsync) contains information associated with the execution of the current HTML tag.
- The output parameter to Process (and ProcessAsync) contains a stateful HTML element representative of the original source used to generate an HTML tag and content.
- The class name has a suffix of **TagHelper**, which is *not* required, but it's considered a best practice convention

Add Tag Helper Directive

- To make the EmailTagHelper class available to all Razor views, add the addTagHelper directive to the Views/_ViewImports.cshtml file:

```
@using AuthoringTagHelpers  
@addTagHelper "*", Microsoft.AspNet.Mvc.TagHelpers"  
@addTagHelper "*", AuthoringTagHelpers"
```

- The code above uses the wildcard syntax to specify all the tag helpers in the assembly will be available.
- The first string after @addTagHelper specifies the tag helper to load (we are using "*" for all tag helpers), and the second string "AuthoringTagHelpers" specifies the assembly the tag helper is in.

The HTML to Implement Tag Helper

- Add the HTML to the Contact View

```
<address>
```

```
  400 Deaderick Street<br />
```

```
  Nashville, TN 37243<br />
```

```
  <abbr title="Phone">P:</abbr>
```

```
  615-555-1212
```

```
</address>
```

```
<address>
```

```
  <strong>DHS:</strong><email>DHS</email><br />
```

```
  <strong>CSS:</strong><email>CSS</email>
```

```
</address>
```

Run the Demo

- Franc Stratton is rendered as a link, but they don't have an href attribute yet to make it functional.

DHS and CSS contact page.

Franc Stratton
400 Deaderick Street
Nashville, TN 37243
P: 615-555-1212

Franc Stratton:[@tn.gov](#)

A Working Email Tag Helper

- Update the EmailTagHelper class with the following:

```
private const string EmailDomain = "tn.gov";  
// Can be passed via <email mail-to="..." />.  
// Pascal case gets translated into lower-kebab-case.  
1 reference  
public string MailTo { get; set; }  
  
0 references  
public override void Process(TagHelperContext context, TagHelperOutput output)  
{  
    output.TagName = "a";    // Replaces <email> with <a> tag  
  
    var address = MailTo + "@" + EmailDomain;  
    output.Attributes["href"] = "mailto:" + address;  
    output.Content.SetContent(address);  
}
```

A Working Email Tag Helper (Cont'd)

- Pascal-cased class and property names for tag helpers are translated into lower kebab case.
- To use the MailTo attribute, use `<email mail-to="value"/>` equivalent.
- The last line sets the completed content for a minimally functional tag helper.
- The following line shows the syntax for adding attributes:

```
output.Attributes["href"] = "mailto:" + address;
```

A Working Email Tag Helper (Cont'd)

- Use `output.Attributes.Add` method to add a tag helper attribute to the end of the collection of tag attributes in *Contact.cshtml*

```
<address>  
    <strong>Franc Stratton</strong><email mail-to="Franc.C.Stratton"></email>  
</address>
```

- Put debug stops in the controller to view how the application demonstration works.

A Working Email Tag Helper (Cont'd)

AuthoringTagHelpers Home About Contact

Contact Us.

DHS and CSS contact page.

Franc Stratton
400 Deaderick Street
Nashville, TN 37243
Phone: 615-555-1212

Franc Stratton Franc.C.Stratton@tn.gov

© 2016 - CSS Authoring Tag Helpers

Tag Helpers Summary

- **Tag Helpers Defined with Examples**
- **Tag Helpers versus HTML Helpers Explained**
- **Built-in Tag Helpers Delineated and Demonstrated**
- **Authoring Tag Helpers Covered and Demonstrated**

Tag Helper References

- <https://app.pluralsight.com/player?course=asp-dot-net-5-understanding&author=roland-guijt&name=asp-dot-net-5-understanding-m2&clip=4&mode=live>
- <https://www.exceptionnotfound.net/tag-helpers-in-asp-net-core-1-0-an-overview/>
- <https://docs.asp.net/en/latest/mvc/views/tag-helpers/intro.html>
- <https://app.pluralsight.com/player?course=asp-dot-net-5-understanding&author=roland-guijt&name=asp-dot-net-5-understanding-m2&clip=4&mode=live>