



C# Corner

Kolkata Chapter



@csharpcorner



<https://www.facebook.com/CsharpCorner>

# Deep Drive Into ASP.Net Core 2.1 (Day 4)

Presented by

**DEBASIS SAHA**



C# Corner MVP

| Team Lead | Author | Blogger | Speaker



[debasis.ds@outlook.com](mailto:debasis.ds@outlook.com)



@debasiskolsaha



<https://github.com/debasis-saha>

## About Me

### Debasis Saha

- | **Team Lead and Consultant**
- | **C# Corner MVP**
- | **DZone MVB**

Email : [debasis.ds@outlook.com](mailto:debasis.ds@outlook.com)

Twitter : @debasiskolsaha

LinkedIn : <https://www.linkedin.com/in/sahadebasis>

- ❖ HTML Helpers
- ❖ Types of HTML Helpers
- ❖ Tag Helpers
- ❖ Custom Tag Helpers
- ❖ Partial View
- ❖ Q n A

- ☐ **HTML Helpers are just similar to the ASP.Net Server Controls**
- ☐ **It is the Extension Method of HtmlHelper Class**
- ☐ **Always return Html Markup to create Form Controls**
- ☐ **Can Create own Custom Helpers or use Build-In HTML Helpers.**

- ❑ Built – In HTML Helpers
  - Standard HTML Helpers
  - Strongly Typed HTML Helpers
  - Templated HTML Helpers
- Custom HTML Helpers

- ☐ **Enables Server Side Code to Create and Render HTML Elements in Razor Files**
- ☐ **Tag Helpers written in C#**
- ☐ **Always target HTML Elements based on**
  - **Element Name**
  - **Attribute Name**
  - **Parent Tag Name**
- ☐ **It reduce explicit transitions between C# and HTML in Razor View**

- ☐ **An HTML Friendly Development Experience**
- ☐ **Front-end Designer can edit Razor page without knowledge of C# Razor Engine**
- ☐ **Provide an environment for create HTML & Razor Markup**
- ☐ **Provides an abstract Layer between UI & Server Side**
- ☐ **Provides to create reusable UI components with Business Logic**

- ☐ **Provide Server-Side Binding without any Server-Side Code**
- ☐ **Helper Objects are very much usefull for HTML Developers**
- ☐ **Provides an experience of working on Basic HTML environment**
- ☐ **Provides Rich Intellisense environment support for Markup between Razor and HTML**



```
public class Employee
{
    public int Id { get; set; }
    public string EmployeeName { get; set; }
    public DateTime DateOfJoin { get; set; }
    public string Department { get; set; }
    public decimal Salary { get; set; }
}
```

Employee  
Class

```
<label asp-for="Employee.EmployeeName"></label>
```

Razor Markup  
Code

```
<label for="Employee_EmployeeName">
    EmployeeName</label>
```

Generated HTML

- ☐ **Anchor Tag Helpers**
- ☐ **Cache Tag Helpers**
- ☐ **Environment Tag Helpers**
- ☐ **Form Tag Helpers**
- ☐ **Image Tag Helpers**
- ☐ **Input Tag Helpers**
- ☐ **Label Tag Helpers**
- ☐ **Select Tag Helpers**
- ☐ **Text Area Tag Helpers**

If we want to expose all Tag Helpers for any views, need to add the Tag Helpers Reference in the ***Views/\_ViewImports.cshtml*** file

```
@using DemoTagHelpers  
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers  
@addTagHelper *, DemoTagHelpers
```

In ASP.NET web app in Visual Studio, the NuGet package ***"Microsoft.AspNetCore.Razor.Tools"*** is responsible for adding Tag Helper tooling.

HTML Helpers	Tag Helpers
HTML Helpers are normally invoked as Method	It Attached to HTML Elements in Razor Views
E.g. <code>@Html.Label("FirstName", "First Name:", new {@class="caption"})</code>	E.g. <code>&lt;label class="caption" asp-for="FirstName"&gt;&lt;/label&gt;</code>
No Intellisence Support	Provide Intellisence Support
The Visual Studio Editor display C# code with a Grey Background Like <code>@Html.AntiForgeryToken()</code>	The Visual Studio editor displays markup targeted by a Tag Helper in a distinctive font.

# DEMO

## Built – In Tag Helpers

## Step 1 – Create Hello-World Tag Helper

1. Create a New Class which need to Inherited from **TagHelper** Class
2. Override the **Process** Method of Tag Helper Class

```
public class HelloWorldHelper : TagHelper
{
    [HtmlTargetElement("hello-world", TagStructure = TagStructure.WithoutEndTag)]
    public override void Process(TagHelperContext context, TagHelperOutput output)
    {
        output.TagName = "label";
    }
}
```

## Step 2 – Set Custom Attribute Property

### 1. To create an attribute called **Message** -

```
public class HelloWorldHelper : TagHelper
{
    public string Message { get; set; }

    [HtmlTargetElement("hello-world", TagStructure = TagStructure.WithoutEndTag)]
    public override void Process(TagHelperContext context, TagHelperOutput output)
    {
        output.TagName = "label";

        var strSb = new StringBuilder();
        strSb.AppendFormat("<span>{0}</span>", this.Message);
        output.PreContent.SetHtmlContent(strSb.ToString());
    }
}
```

## Step 3 – Use Tag Helpers in Razor View

### 1. Now use Created Tag Helpers in Razor Views -

```
<div class="row">  
    <hello-world Message="Debasis Saha">  
    </hello-world>  
</div>
```



- ☐ **A Partial View is a simple which rendered within another view**
- ☐ **Partial View HTML part rendered in the Parent View**
- ☐ **The file extension of Partial view is also .cshtml**

- ☐ **For Breaking Large View into Small Views**
- ☐ **To Create Reusable View Parts.**
- ☐ **Also implement Separation of Concern in case of View also**

# DEMO

## Custom Tag Helpers

# **THANK YOU For Attending the Session**

**Session Resource is Available at –**

- **Presentations & Sample Code :-**

**[https://github.com/debasis-saha/Asp.Net\\_Core\\_Session](https://github.com/debasis-saha/Asp.Net_Core_Session)**