C# Web

PRO/PRW - C# Web 1

**DE HOGESCHOOL
MET HET NETWERK**

Doel

- ASP.Net Core toepassing
  - MVC Web Application
    - MVCTagHelper
      - Table Reference Keys
      - TagHelper

ASP.NET CORE

MVC Web Application
MVCTagHelper

Teams – Attachments – MVCTaghelper.zip

MVCTagHelper

- Models
    - Afdeling
    - Locatie
    - Land

- Data
    - TagHelperDbContext

- Packages
    - SQLServer
    - Tools

MVCTagHelper

- Nuget Package Manager Console
  - Update-database
    - Connection string
      - Appsettings.json

```
builder.Services.AddDbContext<TagHelperDbContext>
(
    options =>
    {
        options.UseSqlServer
        (
            builder.Configuration.GetConnectionString
            (
                "TagHelperConnection"
            )
        );
    }
);
```

MVCTagHelper

- Nuget Package Manager Console
  - Update-database

MVCTagHelper

```csharp
public class Afdeling
{
    public int AfdelingId { get; set; }
    public string AfdelingNaam { get; set; }
    public int LocatieId { get; set; }
    public Locatie Locatie { get; set; }
}
```

MVCTagHelper

```csharp
public class Locatie
{
    public int LocatieId { get; set; }
    public string Stad { get; set; }
    public int LandId { get; set; }
    public ICollection<Afdeling>? Afdelingen { get; set; }
    public Land Land { get; set; }
}
```

MVCTagHelper

```csharp
public class Land
{
    public int LandId { get; set; }
    public string LandCode { get; set; }
    public string LandNaam { get; set; }
    public ICollection<Locatie>? Locaties { get; set; }
}
```

TagHelper

Name	Description
TagName	Get or Set the tag name for the output element.
Attributes	It returns a dictionary containing all the attributes of the output element.
Content	Used to set the contents of the output element.
PreElement	Used to insert Elements before the output element.
PostElement	Used to insert Elements after the output element.
PreContent	Used to insert contents into the output element before any existing content.
PostContent	Used to insert contents into the output element after any existing content.

TagHelper - ViewImports

- **Views/_ViewImports.cshtml**

```
@using MVCTagHelper
@using MVCTagHelper.Models
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
@addTagHelper *, MVCTagHelper
```

MVCTagHelper

- Create Folder
  - TagHelpers
  - Create class
    - BoldTagHelper

```csharp
namespace MVCTagHelper.TagHelpers
{
    [HtmlTargetElement("bold")]
    public class BoldTagHelper : TagHelper
    {
        public override void Process(TagHelperContext context, TagHelperOutput output)
        {
            output.PreContent.SetHtmlContent("<b>");
            output.PostContent.SetHtmlContent("</b>");
        }
    }
}
```

MVCTagHelper

- Views/Home/Index
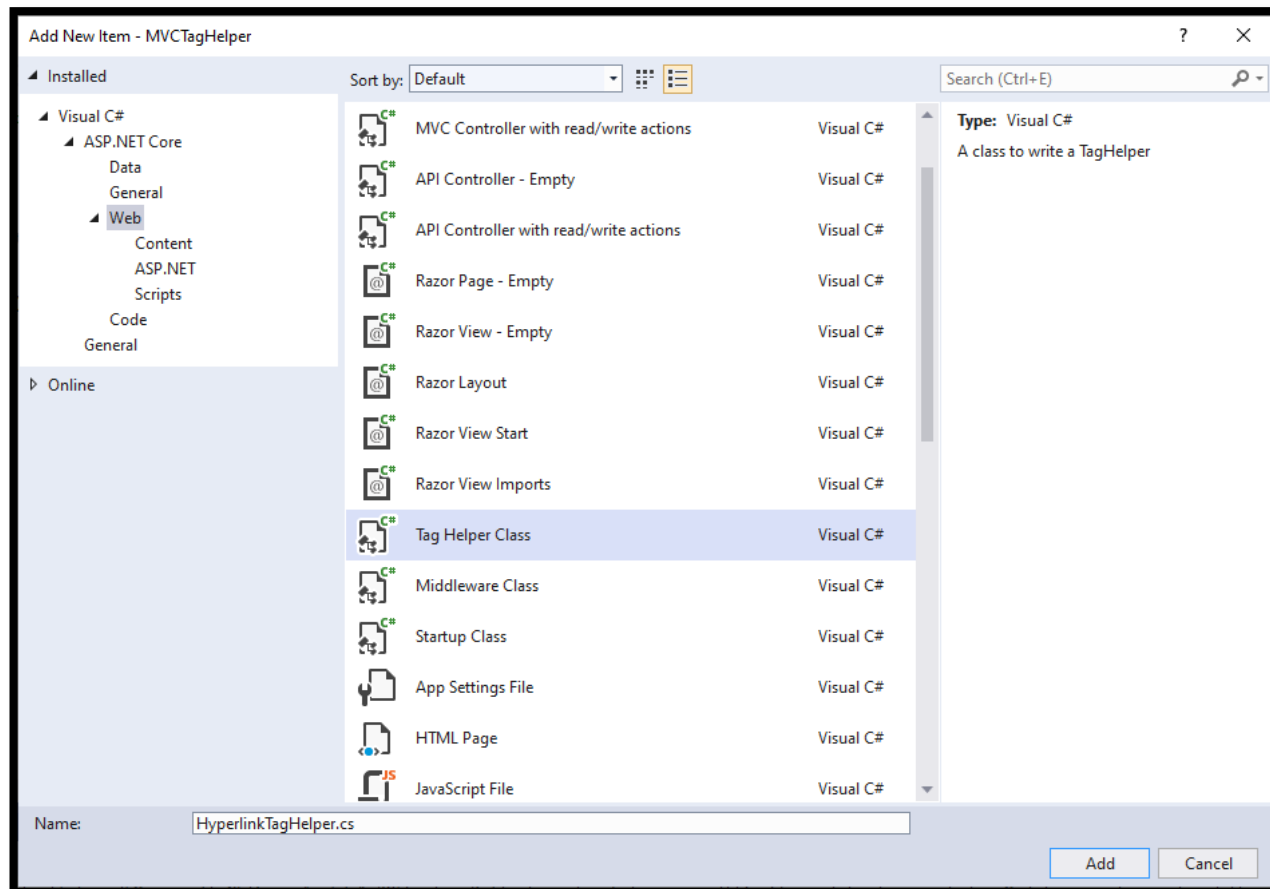
```
@{
    ViewData["Title"] = "Home Page";
}

<div class="text-center">
    <h1 class="display-4">TagHelper</h1>
    <div>
        <p>Gewone tekst</p>
        <bold>
            Dit is een informatieve website voor het gebruik van een
taghelper.
        </bold>
    </div>
</div>
```

MVCTagHelper

- Create taghelper
  - TagHelpers
    - HyperlinkTagHelper

MVCTagHelper

- HyperlinkTagHelper

```csharp
namespace MVCTagHelper.TagHelpers
{
    [HtmlTargetElement("a")]
    public class HyperlinkTagHelper : TagHelper
    {
        public override void Process(TagHelperContext context,
TagHelperOutput output)
        {
            output.Attributes.SetAttribute("class", "btn btn-danger");
        }
    }
}
```

MVCTagHelper

- HyperlinkTagHelper
  - Views/Shared/_Layout.cshtml

```
<div class="bg-primary row">
    <div class="col-1">
        <a asp-controller="Landen" asp-action="Index">Landen</a>
    </div>
    <div class="col-1">
        <a asp-controller="Locaties" asp-action="Index">Locaties</a>
    </div>
    <div class="col-1">
        <a asp-controller="Afdelingen" asp-action="Index">Afdelingen</a>
    </div>
</div>
```

MVCTagHelper

- HyperlinkTagHelper

```
namespace MVCTagHelper.TagHelpers
{
    [HtmlTargetElement("a", Attributes = "background-color")]
    public class HyperlinkTagHelper : TagHelper
    {
        public string BackgroundColor { get; set; } = "danger";
        public override void Process(TagHelperContext context, TagHelperOutput output)
        {
            output.Attributes.SetAttribute("class", $"btn btn-{BackgroundColor}");
        }
    }
}
```

MVCTagHelper

- HyperlinkTagHelper
  - Views/Shared/_Layout.cshtml

```
<div class="bg-primary row">
    <div class="col-1">
        <a background-color="danger" asp-controller="Landen" asp-action="Index">Landen</a>
    </div>
    <div class="col-1">
        <a background-color="success" asp-controller="Locaties" asp-
action="Index">Locaties</a>
    </div>
    <div class="col-1">
        <a background-color="warning" asp-controller="Afdelingen" asp-
action="Index">Afdelingen</a>
    </div>
</div>
```

MVCTagHelper

- HeaderFooterTagHelper

Name	Description
TagName	Get or Set the tag name for the output element.
Attributes	It returns a dictionary containing all the attributes of the output element.
Content	Used to set the contents of the output element.
PreElement	Used to insert Elements before the output element.
PostElement	Used to insert Elements after the output element.
PreContent	Used to insert contents into the output element before any existing content.
PostContent	Used to insert contents into the output element after any existing content.

MVCTagHelper

- HeaderFooterTagHelper

```csharp
public class HeaderFooterTagHelper : TagHelper
{
  public override void Process(…)
  {}

  private TagBuilder GetFormattedDiv(string content)
  {
    TagBuilder title = new TagBuilder("h1");
    title.InnerHtml.Append(content);

    TagBuilder container = new TagBuilder("div");
    container.Attributes["class"] = "bg-info m-1 p-1";
    container.InnerHtml.AppendHtml(title);
    return container;
  }
}
```

MVCTagHelper

- HeaderFooterTagHelper

```csharp
[HtmlTargetElement("div", Attributes = "header, footer")]
public class HeaderFooterTagHelper : TagHelper
{
    public string Header { get; set; }
    public string Footer { get; set; }
    public override void Process(TagHelperContext context, TagHelperOutput output)
    {
        if (!string.IsNullOrEmpty(Header) || !string.IsNullOrEmpty(Footer))
        {
            output.Attributes.SetAttribute("class", "m-1 p-1");
            if (!string.IsNullOrEmpty(Header))
            {
                output.PreElement.SetHtmlContent(GetFormattedDiv(Header));
            }
            if (!string.IsNullOrEmpty(Footer))
            {
                output.PostElement.SetHtmlContent(GetFormattedDiv(Footer));
            }
        }
    }
```

MVCTagHelper

- HeaderFooterTagHelper
  - Views/Locaties/Index.cshtml

```
@{
    ViewData["Title"] = "Index";
    var header = "Locaties";
    var footer = $"Locaties - {DateTime.Now.ToString("dd/MM/yyyy")}";
}

<div header=@header footer=@footer>

    <p>
        <a asp-action="Create">Create New</a>
    </p>
…
```

MVCTagHelper

- AfdelingCardTagHelper

- Create Folder (project niveau)
  - ViewModels
    - Create class
      - AfdelingCard.cs

MVCTagHelper

- AfdelingCardTagHelper - AfdelingCard.cs

```csharp
namespace MVCTagHelper.ViewModels
{
    public class AfdelingCard
    {
        public AfdelingCard(Afdeling afdeling)
        {
            AfdelingId = afdeling.AfdelingId;
            AfdelingNaam = afdeling.AfdelingNaam;
            Locatie = afdeling.Locatie.Stad;
            Land = afdeling.Locatie.Land.LandNaam;
            LandCode = afdeling.Locatie.Land.LandCode;
            Afdeling = afdeling;
        }
        public Afdeling Afdeling { get; set; }
        public int AfdelingId { get; set; }
        public string AfdelingNaam { get; set; }
        public string Locatie { get; set; }
        public string LandCode { get; set; }
        public string Land { get; set; }
    }
}
```

MVCTagHelper

- AfdelingCardTagHelper

```
[HtmlTargetElement(Attributes = "afdeling-card-view-model")]
    public class AfdelingCardTagHelper : TagHelper
    {
        public AfdelingCard AfdelingCardViewModel { get; set; }
        public override void Process(TagHelperContext context, TagHelperOutput output)
        {
            string content = $@"<div class='card'>";
            content += $@"<h4 class='card-title'>
{AfdelingCardViewModel.AfdelingNaam}</h4>";

            content += $@"<p class='card-position'>
{AfdelingCardViewModel.LandCode} - {AfdelingCardViewModel.Locatie}</p>";

            content += $@"<p class='card-description'>De afdeling
{AfdelingCardViewModel.AfdelingNaam} ligt in {AfdelingCardViewModel.Land}. </p>";

            output.TagName = "div";
            output.Content.SetHtmlContent(content);
        }
    }
```

MVCTagHelper

- AfdelingCardTagHelper
  - AfdelingenController - Details

```csharp
public IActionResult Details(int? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var afdeling = _context.Afdelingen
        .Include(a => a.Locatie)
        .ThenInclude(l => l.Land)
        .FirstOrDefault(m => m.AfdelingId == id);
    if (afdeling == null)
    {
        return NotFound();
    }
    var afdelingCard = new AfdelingCard(afdeling);
    return View(afdelingCard);
}
```

MVCTagHelper

- AfdelingCardTagHelper
  - Views/Afdelingen/Details

```
@model MVCTagHelper.ViewModels.AfdelingCard

<afdeling-card afdeling-card-view-model="@Model"></afdeling-card>
```

Verkoop

BE - Hasselt

De afdeling Verkoop ligt in België.

ASP.NET CORE

MVC Web Application
MVCTagHelper
Blackboard – Lesmateriaal – Oefeningen

Oefening 13