Headspring

Views In Depth

Razor Extensibility

Headspring

Duplication and Organization in Core MVC

- HtmlHelper extensions
- Custom Tag Helpers
- Layouts
- Partials
- View Components
- Areas

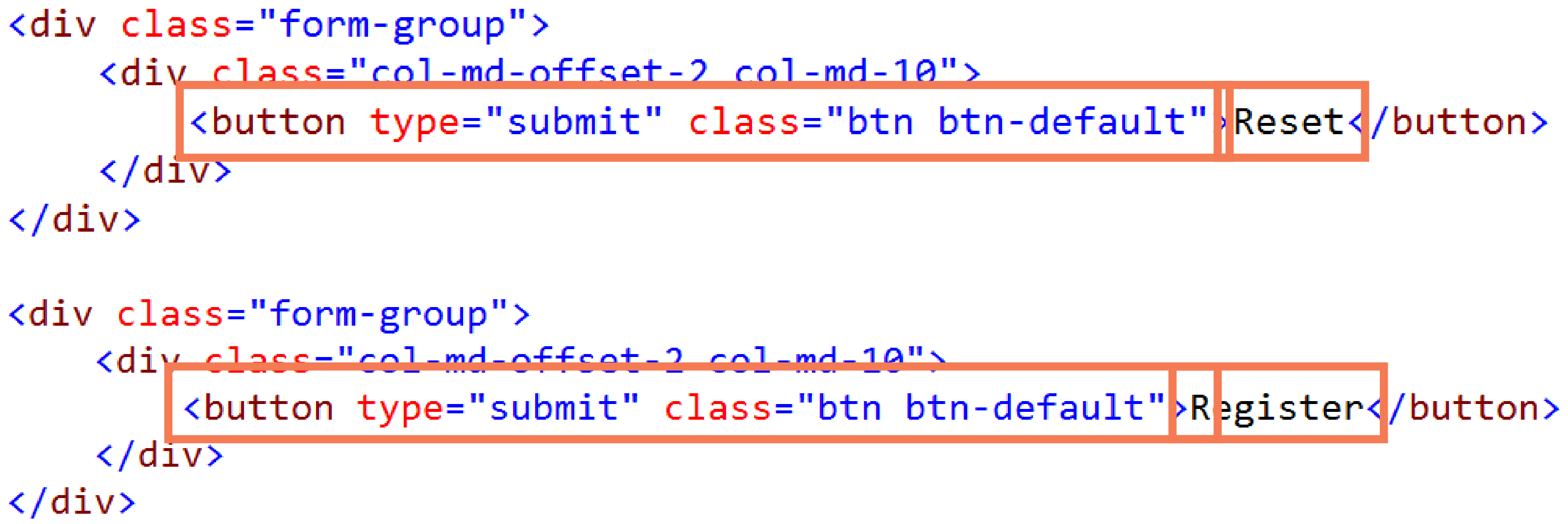
Duplication Antidote Decisions

- More or less HTML?
- More or less code?
- Conceptual or concrete?

HtmlHelper/Tag Helper extensions

- For small amounts of HTML
- For equivalent or highly similar HTML
- Code-centric
- Not easy to compose

Duplicate HTML Snippets



```
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <button type="submit" class="btn btn-default">Reset</button>
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <button type="submit" class="btn btn-default">Register</button>
    </div>
</div>
```

HtmlHelper Extension

```
public static IHtmlContent Button(this IHtmlHelper helper, string text)
{
    var content = $"<button type=\"submit\" class=\"btn btn-default\">{text}</button>";
    var html = new HtmlString(content);
    return html;
}
```

Duplication Removed

```
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        @Html.Button("Reset")
    </div>
</div>


<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        @Html.Button("Register")
    </div>
</div>
```

TagBuilder

```
public static IHtmlContent Button(this IHtmlHelper helper, string text)
{
    var builder = new TagBuilder("button");
    builder.Attributes["type"] = "submit";
    builder.AddCssClass("btn");
    builder.AddCssClass("btn-default");
    builder.InnerHtml.SetContent(text);

    return builder;
}
```

Composite TagBuilder

```
public static IHtmlContent ButtonBlock(this IHtmlHelper helper, string text)
{
    //<div class="form-group">
    var outerDiv = new TagBuilder("div");
    outerDiv.AddCssClass("form-group");

    //<div class="col-md-offset-2 col-md-10">
    var innerDiv = new TagBuilder("div");
    innerDiv.AddCssClass("col-md-offset-2");
    innerDiv.AddCssClass("col-md-10");

    var button = helper.Button(text);

    innerDiv.InnerHtml.AppendHtml(button);

    outerDiv.InnerHtml.AppendHtml(innerDiv);

    return outerDiv;
}
```

Composite Before/After

```
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        @Html.Button("Register")
    </div>
</div>
```

```
@Html.ButtonBlock("Register")
```

Custom Tag Helpers

Extending attributes with HtmlHelper

```
<label asp-for="Email" class="col-md-2 control-label"></label>

@Html.LabelFor(m => m.Email, new {@class = "col-md-2 control-label" })
```

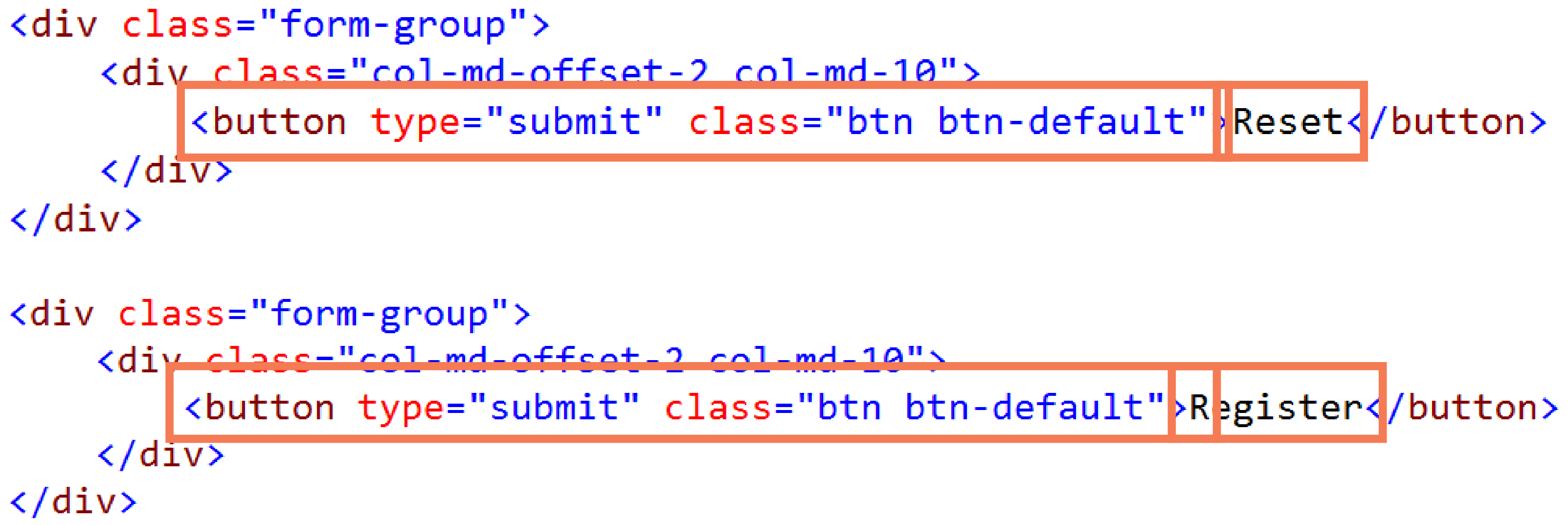
```
public static IHtmlContent ButtonBlock(this IHtmlHelper helper,
    string text,
    object htmlAttributes)
{
```

Headspring

Building a custom TagHelper

- Create class inheriting TagHelper
- Target element/attribute
- Create new element or modify existing
- Can have inner content
- Can modify other tag helpers

Duplicate HTML Snippets



```
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <button type="submit" class="btn btn-default">Reset</button>
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <button type="submit" class="btn btn-default">Register</button>
    </div>
</div>
```

TagHelper

```
public class SubmitTagHelper  TagHelper
{
    public override void Process(TagHelperContext context, TagHelperOutput output)
    {
        output.TagName = "button";

        var builder = new TagBuilder("ignored");
        builder.Attributes["type"] = "submit";
        builder.AddCssClass("btn");
        builder.AddCssClass("btn-default");

        output.MergeAttributes(builder);
    }
}
```

Registering TagHelpers

```
@using WebApplication1
@using WebApplication1.Models
@using WebApplication1.Models.AccountViewModels
@using WebApplication1.Models.ManageViewModels
@using Microsoft.AspNetCore.Identity
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
@addTagHelper WebApplication1.TagHelpers.SubmitTagHelper, WebApplication1
```

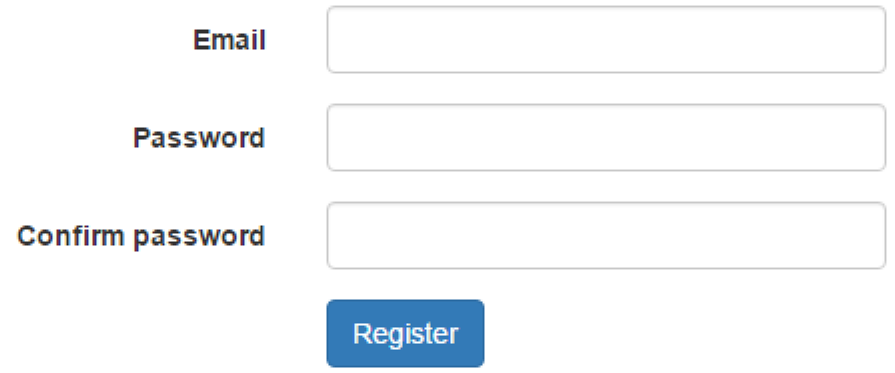
Using a TagHelper

```
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <submit>Register</submit>
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <submit>Reset</submit>
    </div>
</div>
```

Headspring

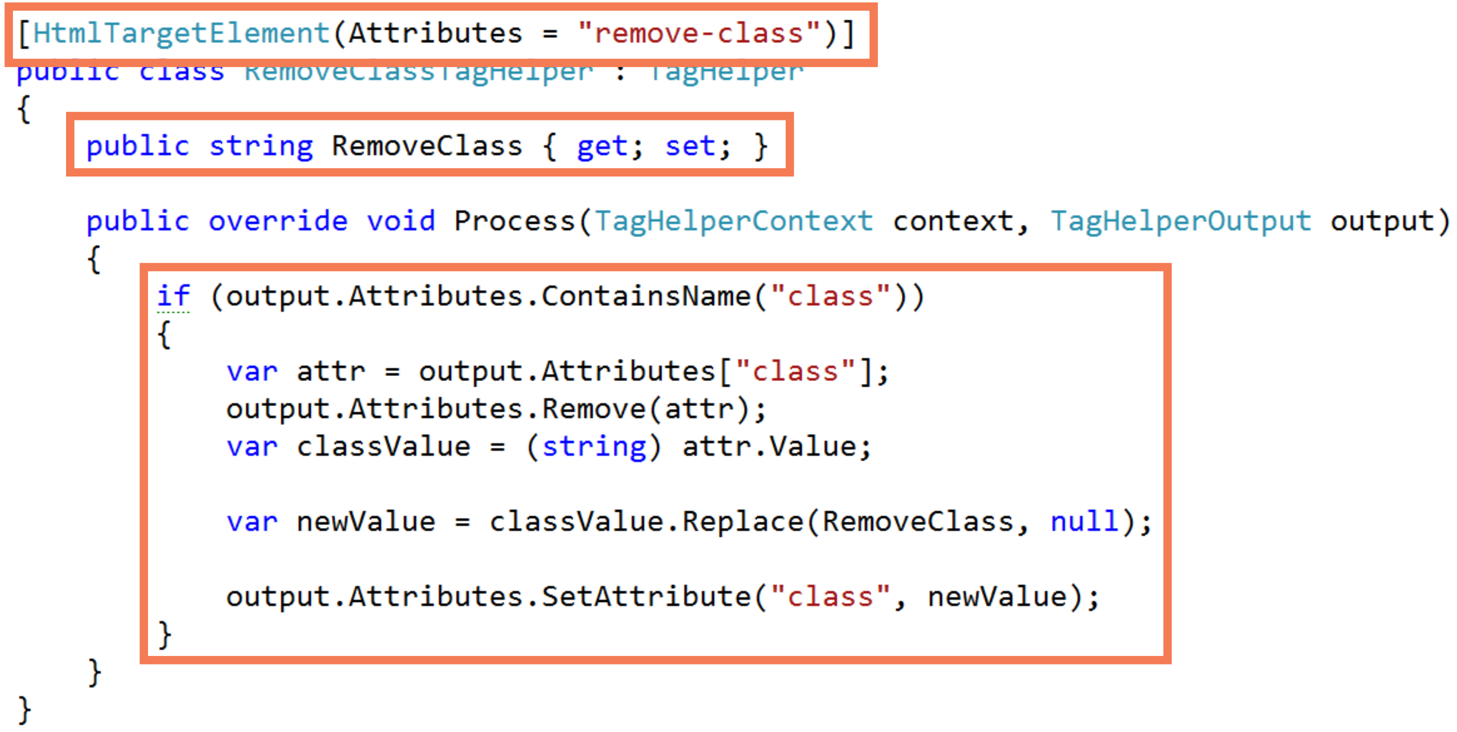
Attribute merging

```
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <submit class="btn-primary">Register</submit>
    </div>
</div>
```



```
▼<div class="col-md-offset-2 col-md-10">
   <button class="btn-primary btn-default btn" type="submit">Register</button>
  </div>
```

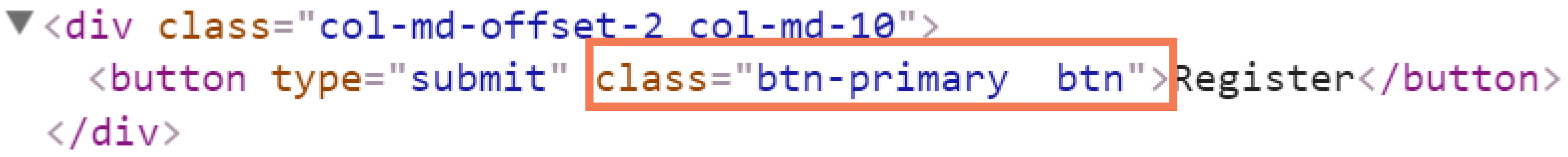
TagHelper targets



```
[HtmlTargetElement(Attributes = "remove-class")]
public class RemoveClassTagHelper : TagHelper
{
    public string RemoveClass { get; set; }

    public override void Process(TagHelperContext context, TagHelperOutput output)
    {
        if (output.Attributes.ContainsName("class"))
        {
            var attr = output.Attributes["class"];
            output.Attributes.Remove(attr);
            var classValue = (string) attr.Value;

            var newValue = classValue.Replace(RemoveClass, null);

            output.Attributes.SetAttribute("class", newValue);
        }
    }
}
```

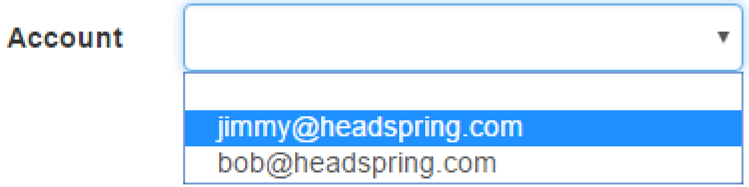
Applying attributes

```
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <submit class="btn-primary" remove-class="btn-default">Register</submit>
    </div>
</div>
```



```
▼<div class="col-md-offset-2 col-md-10">
   <button type="submit" class="btn-primary  btn">Register</button>
 </div>
```

Data-bound elements


Account

jimmy@headspring.com
bob@headspring.com

Headspring

Dependencies in Tag Helpers

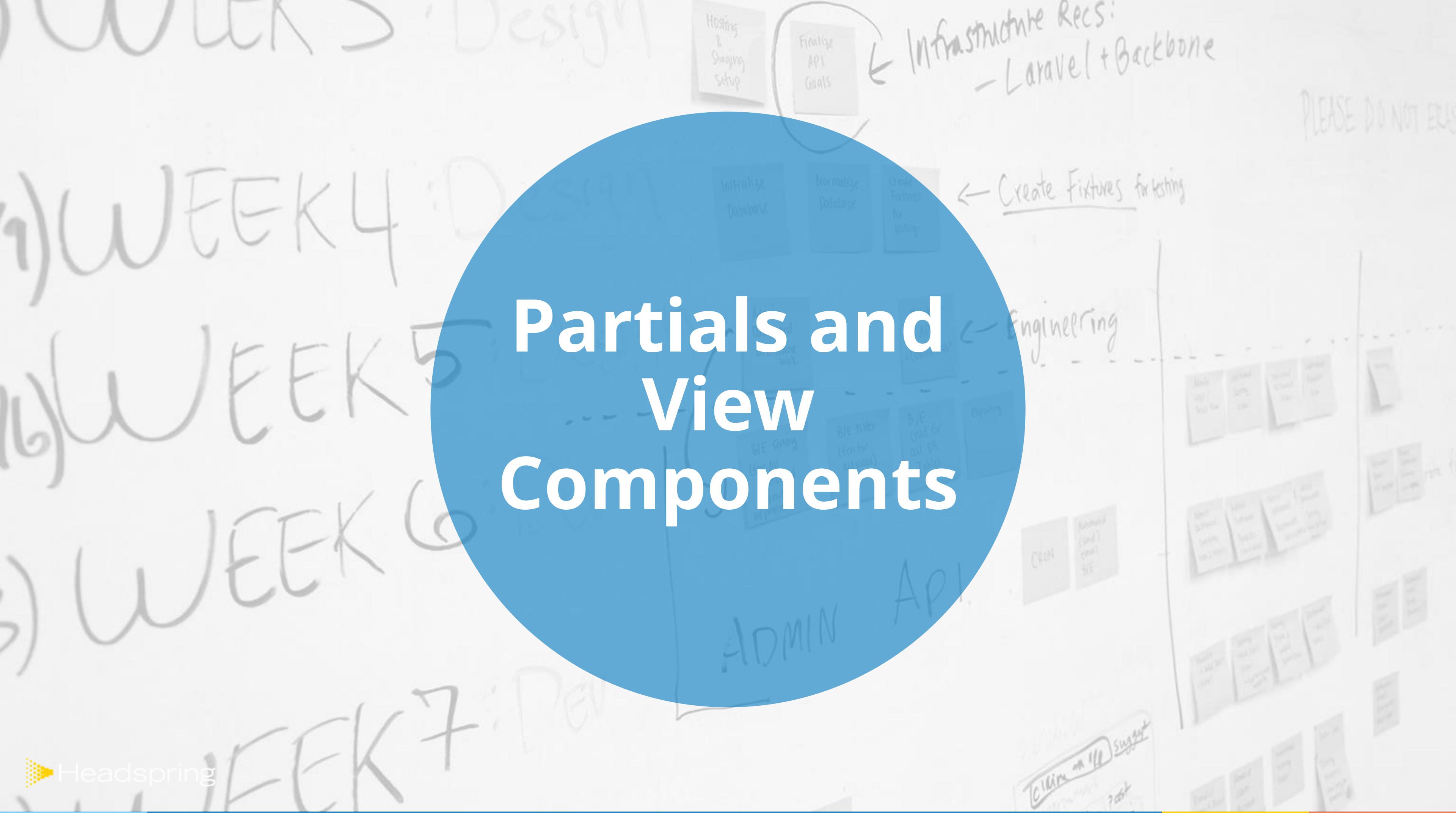
```
[HtmlTargetElement("account-select")]
public class AccountSelectTagHelper : TagHelper
{
    private readonly ApplicationDbContext _dbContext;
    private readonly IHtmlGenerator _htmlGenerator;
    private ICollection<string> _currentValues;
    private const string ForAttributeName = "asp-for";

    [HtmlAttributeNotBound]
    [ViewContext]
    public ViewContext ViewContext { get; set; }

    [HtmlAttributeName(ForAttributeName)]
    public ModelExpression For { get; set; }

    public AccountSelectTagHelper(ApplicationDbContext dbContext,
        IHtmlGenerator htmlGenerator)
    {
        _dbContext = dbContext;
        _htmlGenerator = htmlGenerator;
    }
```

TagHelper initialization

```
public override void Init(TagHelperContext context)
{
    _currentValues = _htmlGenerator.GetCurrentValues(
        ViewContext,
        For.ModelExplorer,
        expression: For.Name,
        allowMultiple: false);
}
```

Composing tag builders

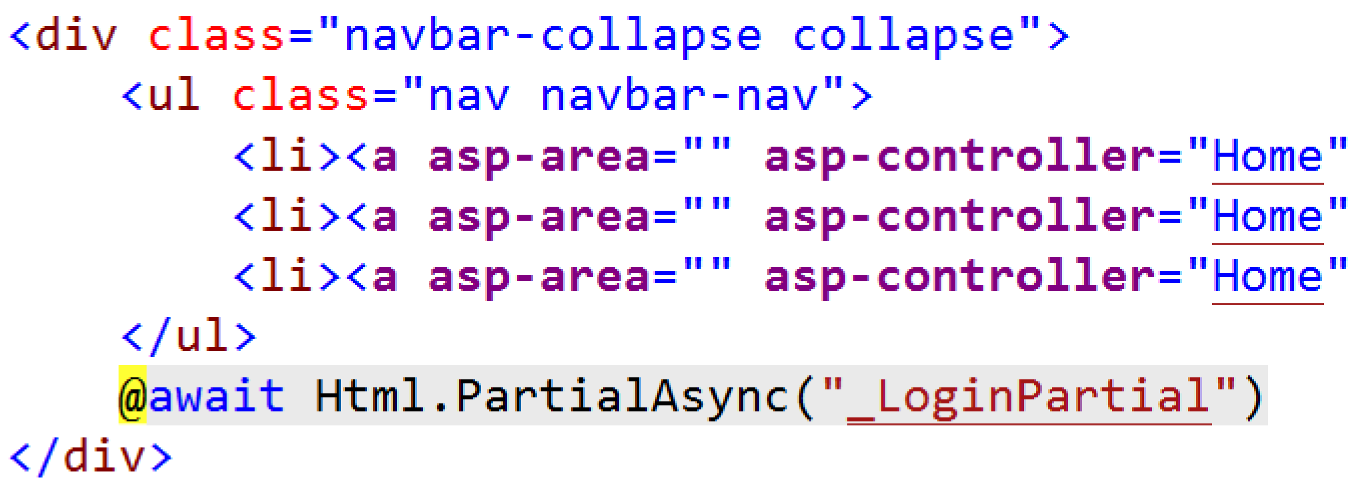
```
public override void Process(TagHelperContext context, TagHelperOutput output)
{
    var accounts = _dbContext.Users.ToList();

    var items = accounts.Select(a => new SelectListItem
    {
        Text = a.UserName,
        Value = a.Id
    }).ToList();
    items.Insert(0, new SelectListItem());

    var tagBuilder = _htmlGenerator.GenerateSelect(
        ViewContext,
        For.ModelExplorer,
        optionLabel: null,
        expression: For.Name,
        selectList: items,
        currentValues: _currentValues,
        allowMultiple: false,
        htmlAttributes: null);

    output.TagName = "select";
    if (tagBuilder != null)
    {
        output.MergeAttributes(tagBuilder);
        output.PostContent.AppendHtml(tagBuilder.InnerHtml);
    }
}
```



Partials and View Components

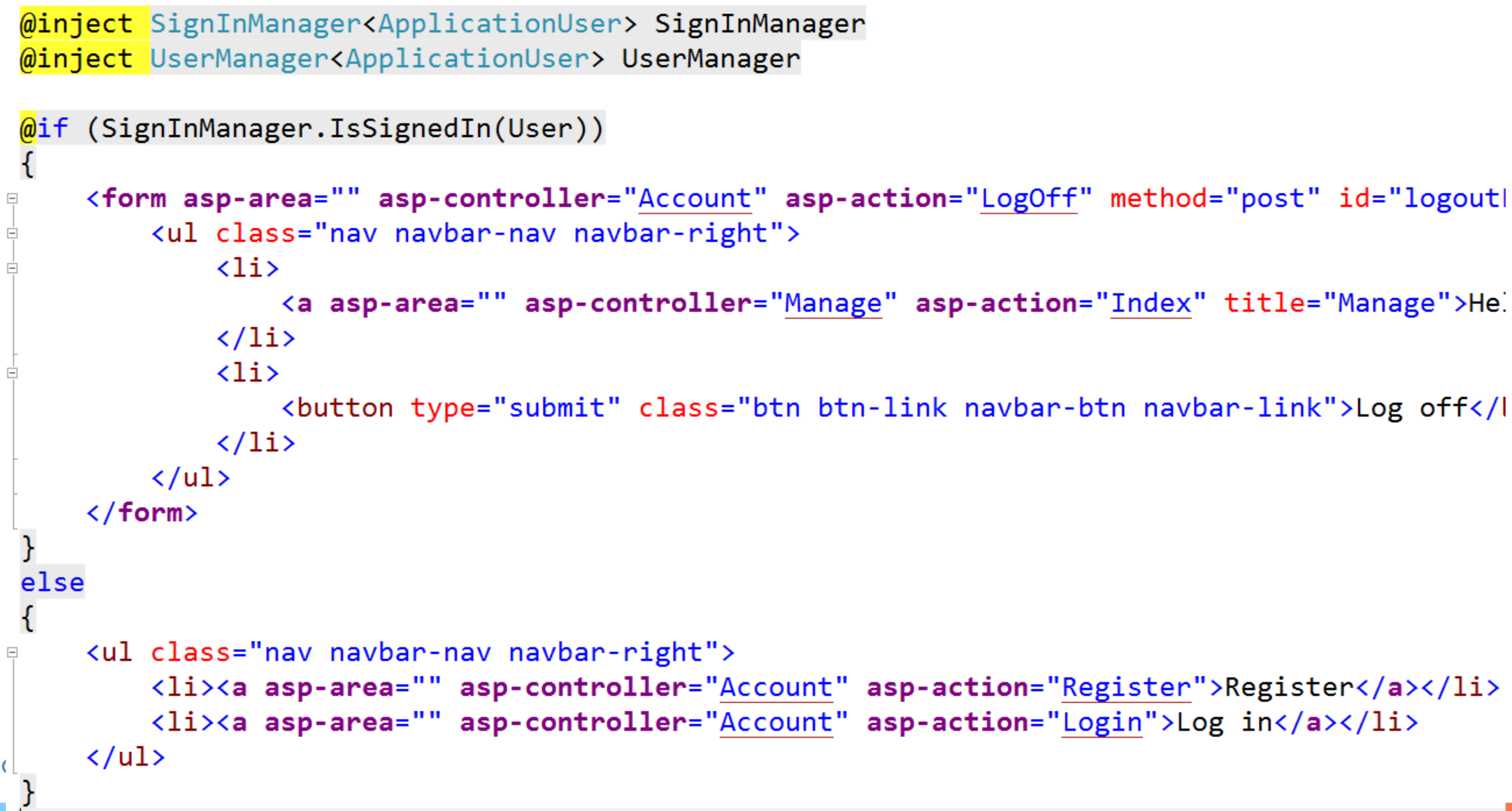
Partials

- For shared content
- For encapsulated content
- Can be rendered sync or async



```
<div class="navbar-collapse collapse">
    <ul class="nav navbar-nav">
        <li><a asp-area="" asp-controller="Home"
        <li><a asp-area="" asp-controller="Home"
        <li><a asp-area="" asp-controller="Home"
    </ul>
    @await Html.PartialAsync("_LoginPartial")
</div>
```

Dependency Injection in Views



```
@inject SignInManager<ApplicationUser> SignInManager
@inject UserManager<ApplicationUser> UserManager

@if (SignInManager.IsSignedIn(User))
{
    <form asp-area="" asp-controller="Account" asp-action="LogOff" method="post" id="logout
        <ul class="nav navbar-nav navbar-right">
            <li>
                <a asp-area="" asp-controller="Manage" asp-action="Index" title="Manage">He
            </li>
            <li>
                <button type="submit" class="btn btn-link navbar-btn navbar-link">Log off</
            </li>
        </ul>
    </form>
}
else
{
    <ul class="nav navbar-nav navbar-right">
        <li><a asp-area="" asp-controller="Account" asp-action="Register">Register</a></li>
        <li><a asp-area="" asp-controller="Account" asp-action="Login">Log in</a></li>
    </ul>
}
```

Designing Partials

- Named with a leading underscore
- Model passed in from parent
- Most common for sharing widgets across multiple views
- Also for making larger views more manageable
- Common for Partials to show ActionFilter-injected information

View Components

- Successor to child actions
- Render a chunk of HTML
- Like a Controller-Model-View
    - ViewModel/ViewData
    - Parameters instead of model binding
    - No filters
- Typically invoked in the layout

Creating a View Component

- Inherit from ViewComponent
- Decorate class with [ViewComponent]
- Class with name ending in “ViewComponent”
- Method InvokeAsync that returns an IViewComponentResult

View Component

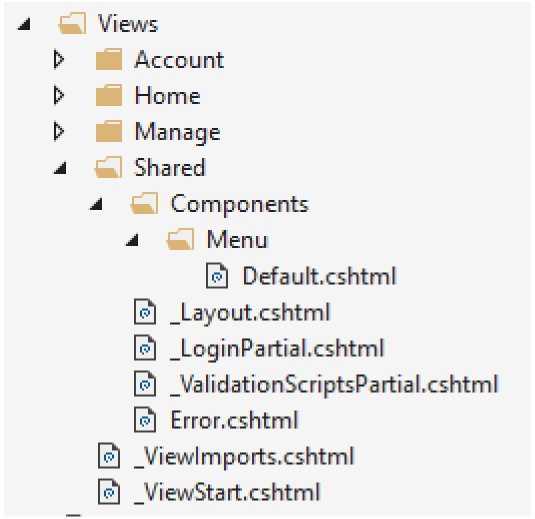
```csharp
public class MenuViewComponent : ViewComponent
{
    private readonly AppContext _dbContext;

    public MenuViewComponent(AppContext dbContext)
    {
        _dbContext = dbContext;
    }

    public async Task<IViewComponentResult> InvokeAsync(int currentMenuId)
    {
        var menuItems = await _dbContext.MenuItems.ToListAsync();

        var model = new MenuComponentModel
        {
            Items = menuItems,
            Current = currentMenuId
        };

        return View(model);
    }
}
```

View Component Results

- ViewViewComponentResult
  - Invokes the Default view
  - Can be strongly-typed view
- ContentViewComponentResult
  - HTML-encoded text
  - No content type necessary
- HtmlContentViewComponentResult
  - IHtmlContent result
  - No view invoked

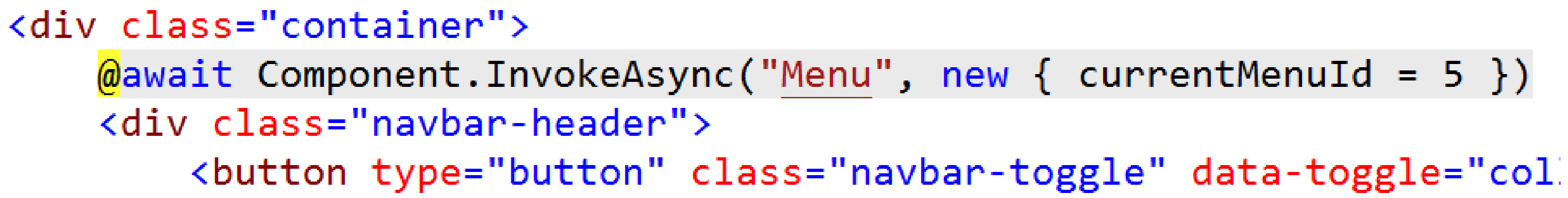
Locating a View Component

- Views/<controller_name>/Components/<view_component_name>/<view_name>.cshtml
- Views/Shared/Components/<view_component_name>/<view_name>.cshtml
- Default view name is "Default"


```
Views
  Account
  Home
  Manage
  Shared
    Components
      Menu
        Default.cshtml
    _Layout.cshtml
    _LoginPartial.cshtml
    _ValidationScriptsPartial.cshtml
    Error.cshtml
  _ViewImports.cshtml
  _ViewStart.cshtml
```

Headspring

Invoking a View Component



```
<div class="container">
    @await Component.InvokeAsync("Menu", new { currentMenuId = 5 })
    <div class="navbar-header">
        <button type="button" class="navbar-toggle" data-toggle="col
```

Partials + Filters vs. Component

- Partials
  - Everything on main controller
  - Everything in main ViewData
- View Component
  - No magic strings
  - View dictates necessity
  - No contrived controller hierarchy