Improving the Views in the Application

Gill Cleeren

CTO Xpirit Belgium

@gillcleeren – xpirit.com/gill

Module overview

Using partial views

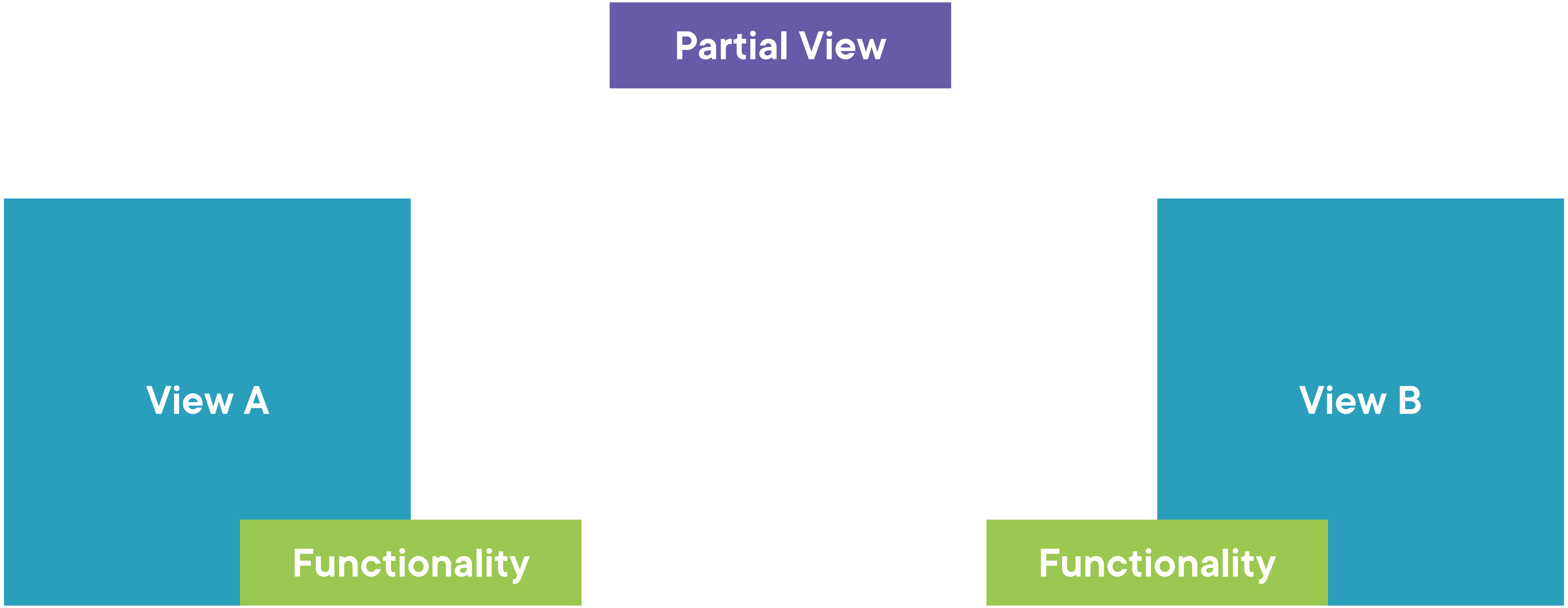
Creating the shopping cart

Working with view components

Creating a custom tag helper

Using Partial Views

Partial Views



Partial View

View A

Functionality

View B

Functionality

```
@model Pie
<div>
    <div class="thumbnail">
        <img src="@Model.ImageThumbnailUrl" alt="">
        <h3>@Model.Price.ToString("c")</h3>
    </div>
</div>
```

A Sample Partial View

- Name typically start with _

```
@foreach (var pie in Model.Pies)
{
    <partial name="_PieCard" model="pie"/>
}
```

Using the Partial View

Partial Views

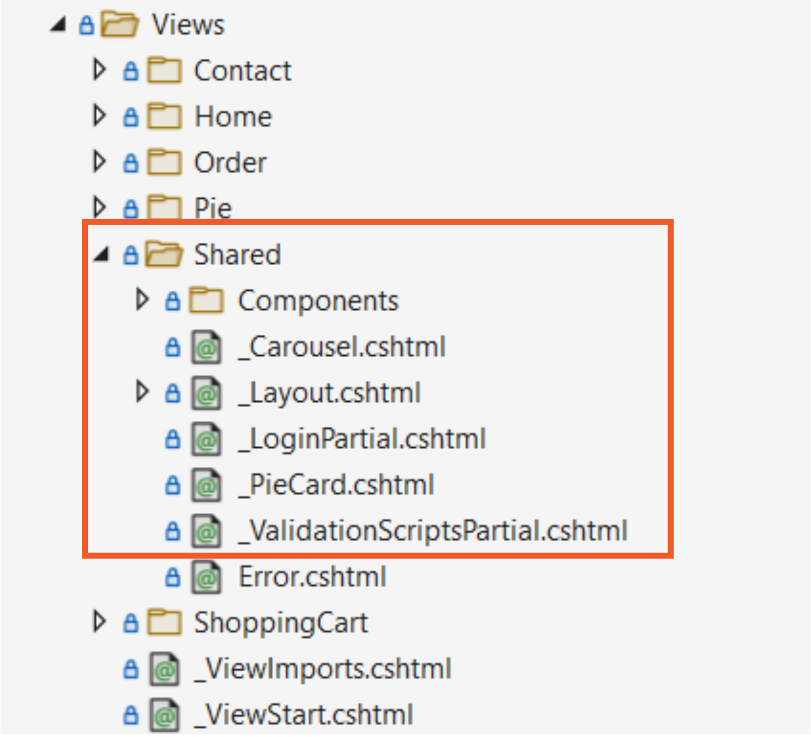
Avoid duplication

Break up large views into multiple smaller views

Don’t execute _ViewStart

Shared or specific folder, convention-based

Partial Views Folder Location



- Views
  - Contact
  - Home
  - Order
  - Pie
  - Shared
    - Components
    - _Carousel.cshtml
    - _Layout.cshtml
    - _LoginPartial.cshtml
    - _PieCard.cshtml
    - _ValidationScriptsPartial.cshtml
    - Error.cshtml
  - ShoppingCart
  - _ViewImports.cshtml
  - _ViewStart.cshtml

Demo

Creating and using a partial view

Creating the Shopping Cart

Creating the shopping cart

- Cart per user
- GUID created as ID for the cart
- Session ID is stored in cookie

Using Session State

Information stored by app to persist data across requests

Based on cookie

Linked to the browser

```
builder.Services.AddSession();

...

app.UseSession();
```

Adding Session Support in the App

Demo

Creating the model

Creating the shopping cart

Enabling session management

Performing a database migration

Adding the shopping cart view

Navigation updates

Working with View Components

```
@model Pie
<div>
  <div>
    <img src="@Model.ImageThumbnailUrl">
      <h3>@Model.Price.ToString("c")</h3>
      <p>@Model.ShortDescription</p>
```

Partial View Limitations

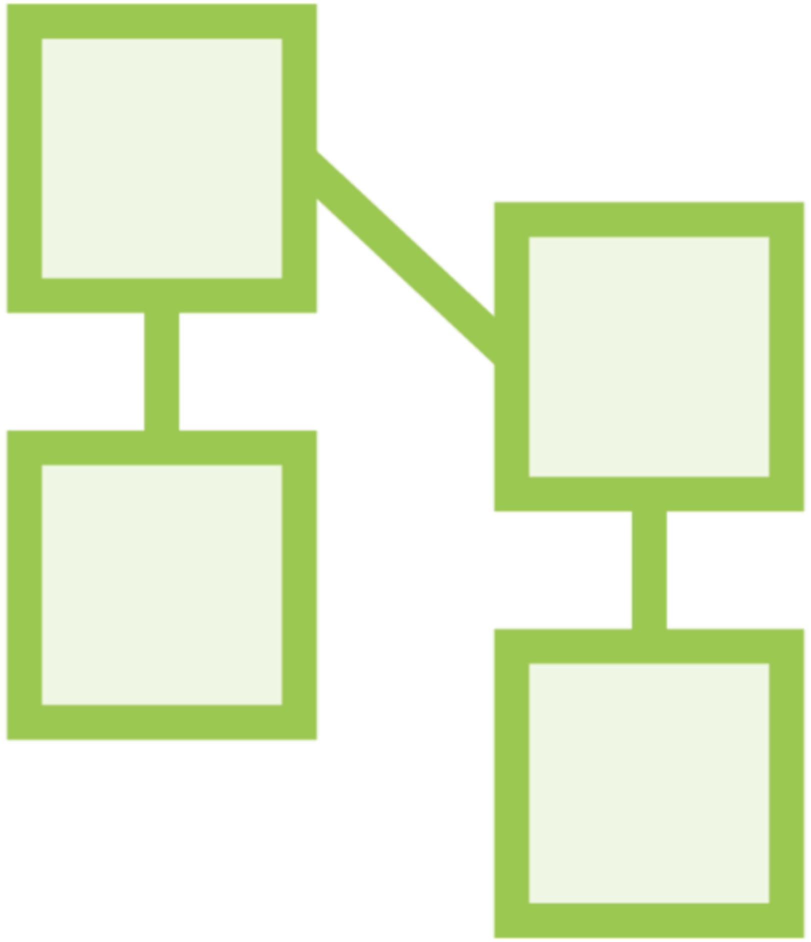
Introducing View Components

Partial content

Logic and view

Support for dependency injection

Always linked to parent view



View Component usage

- Login panel
- Dynamic navigation
- Shopping cart
- …

Creating a View Component

Derive from ViewComponent

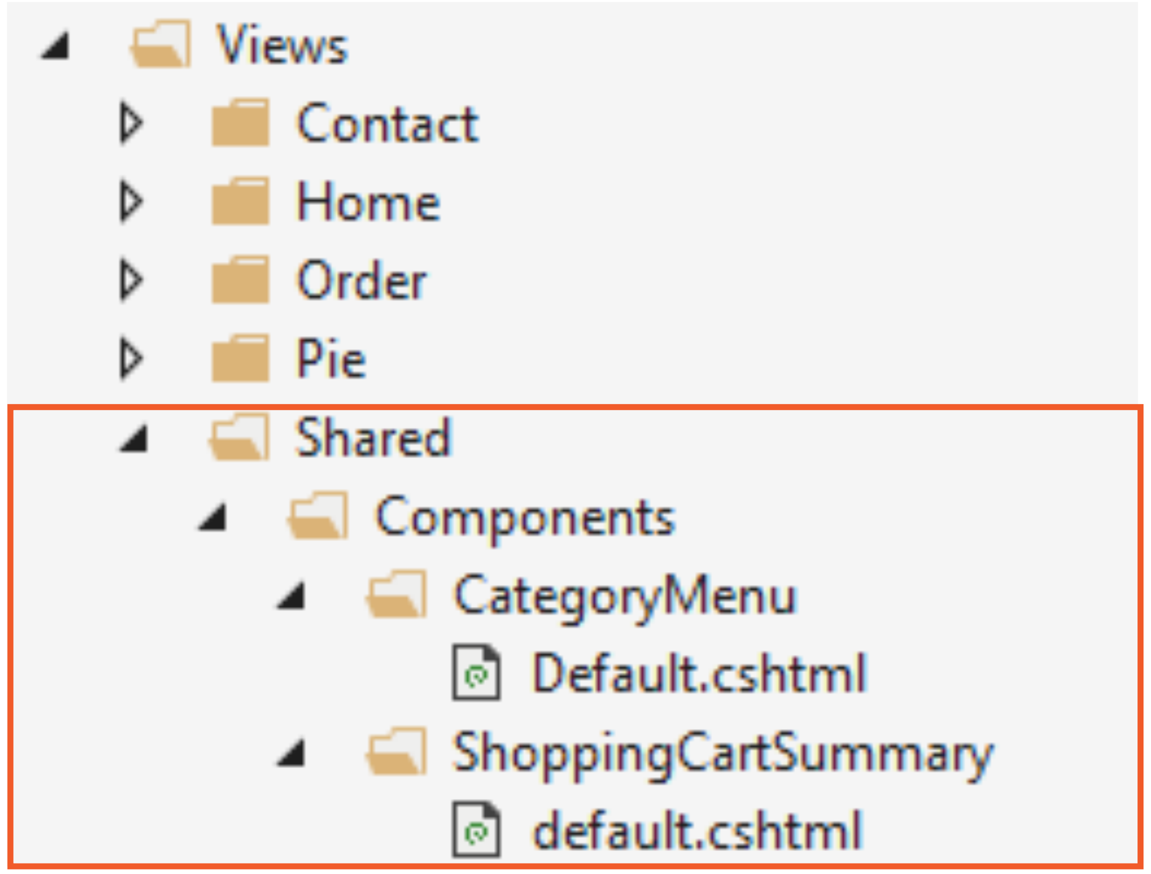
[ViewComponent]

Suffix with ViewComponent

```
public class ShoppingCartSummary : ViewComponent
{
    public IViewComponentResult Invoke()
    {
        return View(model);
    }
}
```

Creating a View Component

View Components Search Path



```
Views
    Contact
    Home
    Order
    Pie
    Shared
        Components
            CategoryMenu
                Default.cshtml
            ShoppingCartSummary
                default.cshtml
```

```
@await Component.InvokeAsync("ShoppingCartSummary")
```

Using the View Component

```
<vc:shopping-cart-summary></vc:shopping-cart-summary>
```

Invoking a View Component with a Tag Helper

Demo

Creating a navigation view component

Creating a shopping cart view component

Creating a Custom Tag Helper

Tag Helpers enable server-side C# code to participate in creating and rendering HTML elements in Razor files.

Creating a Custom Tag Helper

Inherit from base TagHelper class

<Name>TagHelper

Override Process/ProcessAsync

Output is HTML

Creating a Custom Tag Helper

```
public class EmailTagHelper: TagHelper
{
    public override void Process(
        TagHelperContext context, TagHelperOutput output)
    {
        ...
    }
}
```

```
<email
     address="info@@bethanyspieshop.com"
     content="Contact us">
</email>
```

Using the Tag Helper

Registering the Custom Tag Helper

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

Demo

Creating the contact page

Creating a tag helper

Adding the tag helper to the contact page

Summary

Partial views promote reuse of view code

View Components can be considered an upgrade of partial views

Custom tag helpers wrap common functionality

Up next:

Creating the order form