

# Working with Partial Views, Sections, and View Components

---



**Alex Wolf**

[www.crywolfcode.com](http://www.crywolfcode.com)



# Understanding Partial Views

---



# Introducing Partial Views

Wired Brain Coffee

Home Feedback Menu Contact

## Contact

We'd love to hear from you.

Life gets busy but we try to respond with 24 hours. If you have a more immediate question, give us a ring using the number over on the right. We are crazy friendly!

Your Name:

Phone Number:

Email:

Message:

Send Message!

### Keep in Touch

Sign up for our mailing list - no spam, we promise.


Email

Join

### Contact Details

1234 Wired Brain Blvd  
Awesome City, MM 55555

Phone: (123) 456-7890  
Hours: Monday - Friday: 6:00 AM to 9:00 PM



**WIRED BRAIN**  
COFFEE

Copyright © Wired Brain

Contact Info:

Phone: 555-555-1234  
Address: 1234 Test  
Email: Care@WiredBrain.com

Coffee Fact of the Day

More cups of coffee are drank every day than there are people on earth. *(May not be true)*



Layout File



Razor Page



Partial View



```
<div class="container">  
    <partial name="MobileWidget" /> <!-- Tag Helper way -->  
  
    @Html.Partial("MobileWidget") <!-- HTML Helper way -->  
</div>
```

## Including Partial Views on Razor Pages

The Partial Tag Helper can include a Partial View on the page

There is also an older style HTML helper that can include Partial Views



# Partial View Search Locations

## Search Directories for “MobileWidget” Partial

Pages/{parent-page-directory}/MobileWidget.cshtml

Pages/MobileWidget.cshtml

Pages/Shared/MobileWidget.cshtml



```
// Partial View
```

```
@model MenuItem
```

```
<h1>@Model.Name</h1>
```

```
<p>@Model.Description</p>
```

```
// Parent View
```

```
<partial name="ItemWidget"  
model="@Model.Item" />
```

◀ Model declared for the Partial View

◀ Access and display properties on that Model

◀ Reference the Partial View and pass in an instance of the Model (from parent view)



# Demo



## Demo: Reusing Markup with Partial Views



# Razor Sections

---





# Introducing Razor Sections

Wired Brain Coffee

Home Feedback Menu Contact

## Contact

We'd love to hear from you.

Life gets busy but we try to respond with 24 hours. If you have a more immediate question, give us a ring using the number over on the right. We are crazy friendly!

Your Name:

Phone Number:

Email:

Message:

Send Message!

### Keep in Touch

Sign up for our mailing list - no spam, we promise.


Email

Join

### Contact Details

1234 Wired Brain Blvd  
Awesome City, MM 55555

Phone: (123) 456-7890  
Hours: Monday - Friday: 6:00 AM to 9:00 PM



**WIRED BRAIN**  
COFFEE

Copyright © Wired Brain

Contact Info:

Phone: 555-555-1234  
Address: 1234 Test  
Email: Care@WiredBrain.com

Coffee Fact of the Day

More cups of coffee are drank every day than there are people on earth. *(May not be true)*



Layout File



Dynamic Sections



```
// Layout
```

```
@if(IsSectionDefined("Footer")){  
    RenderSection("Footer");  
}  
else {  
    <p>Default footer content</p>  
}
```

```
// Razor Page
```

```
@section Footer {  
    <p>Custom footer content</p>  
}
```

◀ Render section if defined on inner Razor Page

◀ Otherwise show default content

◀ Override the layout section from inner Razor Page



# Demo



## Demo: Dynamic Layouts Using Sections

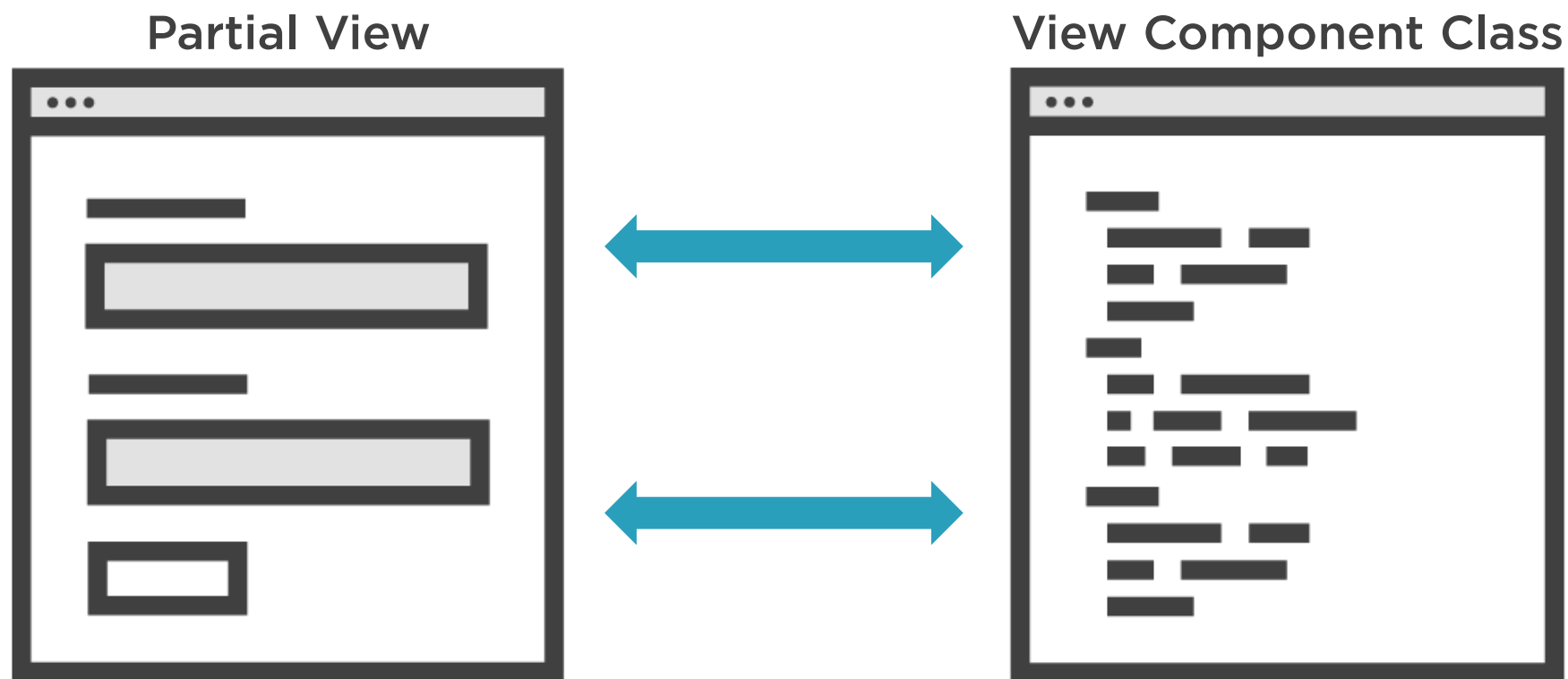


# Introducing View Components

---



# Empowering Partial Views with Logic



```
public class MenuWidget : ViewComponent
{
    public async Task<IViewComponentResult>
    InvokeAsync()
    {
        var headers = Request.Headers;

        // Build mortgage rates model

        return View(mortgageModel);
    }
}
```

- ◀ Inherit from ViewComponent
- ◀ Define InvokeAsync method
- ◀ Access framework context through inheritance
- ◀ Main logic of the component
- ◀ Return a view, optionally passing in a model



```
<div id="widgets">
    <div class="widget-wrapper">
        @await Component.InvokeAsync("MenuWidget", new { count = 1 })
        <vc:menu-widget count="1" />
    </div>
</div>
```

## Rendering View Components with Razor

**View Components can be rendered inside Pages and Layouts**

**Can be invoked with or without Model parameters**



# View Component Search Paths

MenuWidget.cs

```
public void InvokeAsync()  
{  
    return View()  
}
```



Search template

Pages/Components/  
{ **ComponentName** }/{ **ViewName** }



Pages/Components/  
**MenuWidget/Default.cshtml**





# View Components vs. Razor Pages

**Razor Page**

**View Component**



# Demo



## Demo: Reusable Widgets with View Components



# Demo



## Demo: View Components as Tag Helpers



# Summary



**Partial Views are the simplest option for reusing markup**

**Sections allow for dynamic content regions in layout files**

**View Components can package both markup and logic for reuse**

**View Components can be rendered using Tag Helpers**

**View Components support page specific customizations using parameters**

