ASP.NET Core 6: Razor Pages Fundamentals

Getting Set up to Work with Razor Pages

Alex Wolf
.NET Developer

www.thecodewolf.com

Razor Pages

A server-side, paged-based framework for building dynamic web user interfaces with ASP.NET Core

Why use Razor Pages?

The Case for Razor Pages

Why should I use Razor Pages?

- **Create dynamic, data driven web apps**
- **Built on top of ASP.NET Core**
- **Equally powerful and approachable**
- **A great foundation for other frameworks**

ASP.NET Core 6: Razor Pages Fundamentals

Version Check

Version Check

This version was created by using:

- ASP.NET Core 6.0
- C# 10
- Visual Studio 2022

Not Applicable

This course is NOT applicable to:

- ASP.NET Core 1.x or earlier
- Visual Studio 2015 or earlier

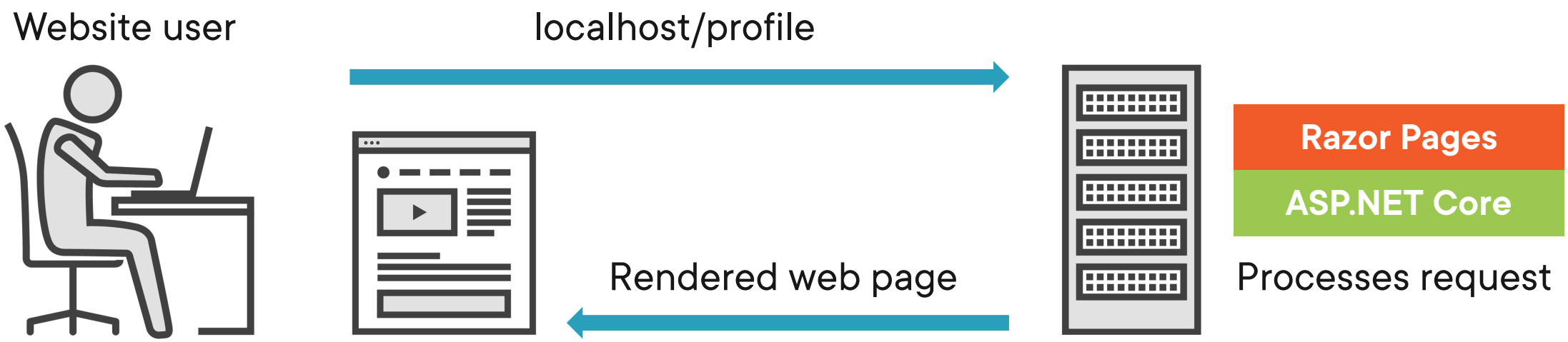
Razor Pages Core Concepts

Razor Pages

A server-side, page-based framework for building dynamic web user interfaces with ASP.NET Core.

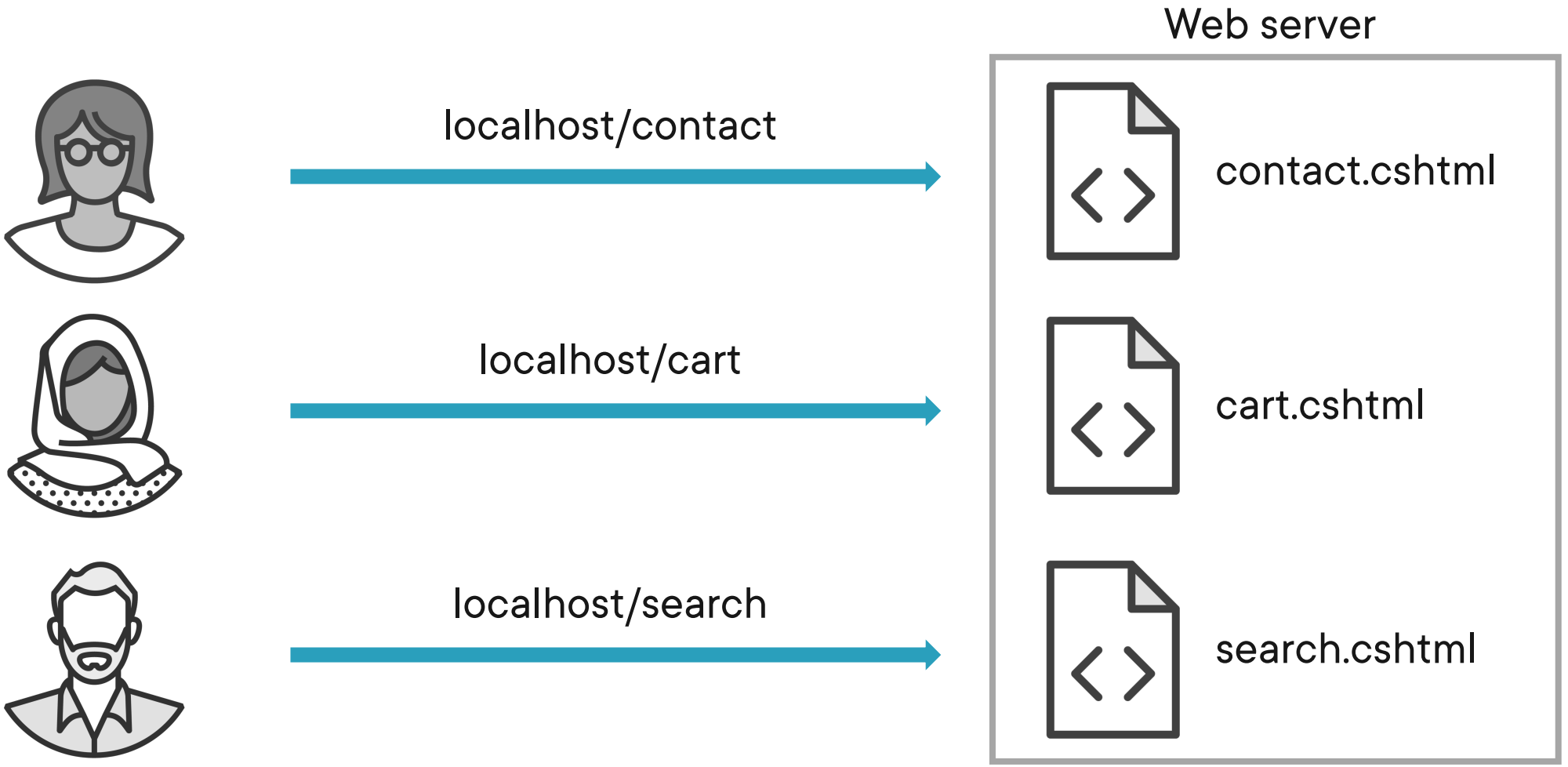
Key concepts

Razor Pages Is a Server-side Framework

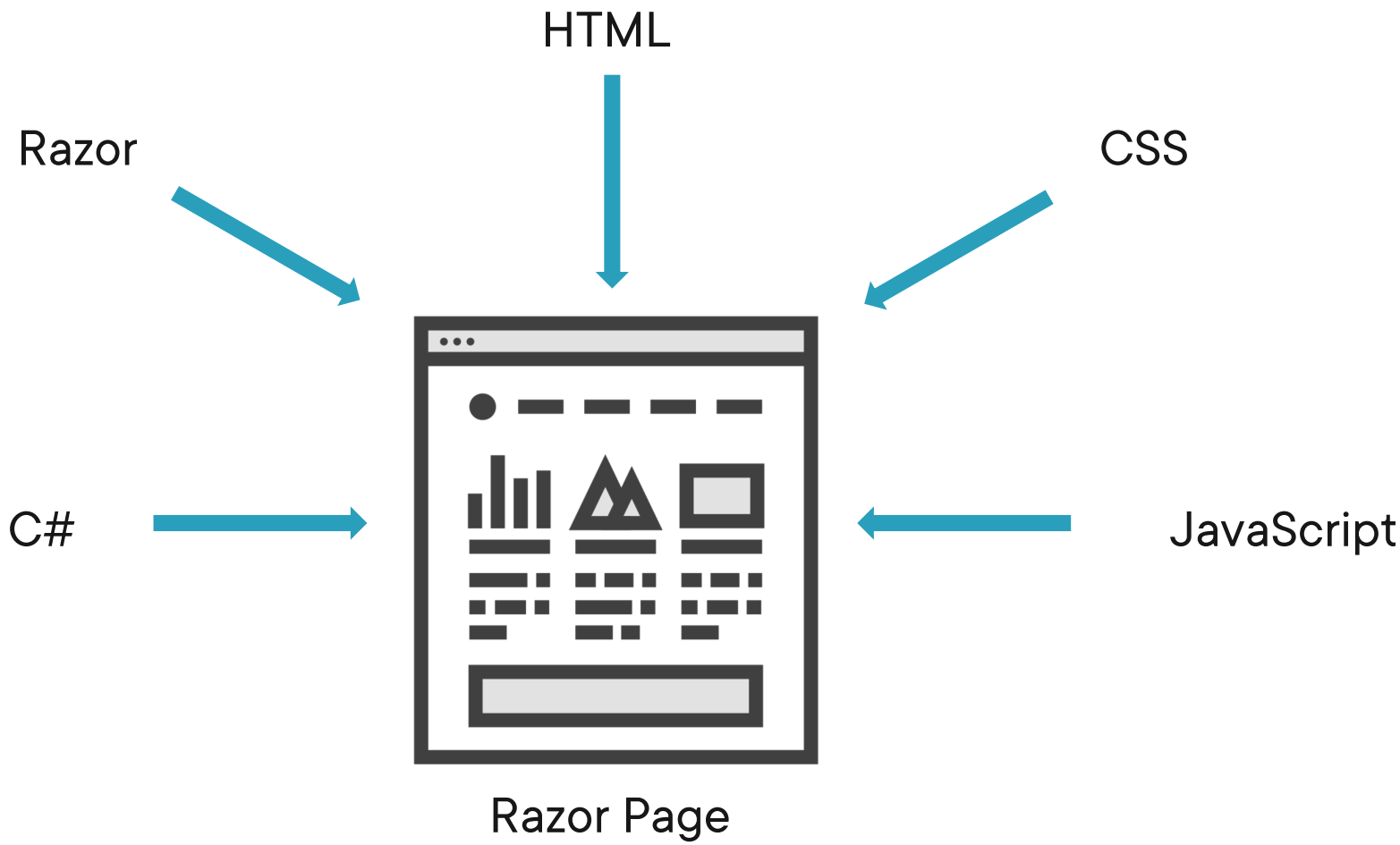


Comparing Client-side Frameworks

Razor Pages Is a Paged-based Framework



Razor Pages Build Web User Interfaces



```
<h1>Reminders</h1>

@{
   var day = DateTime.Today.Day;
}

@if (day == “Friday”)
{
        <p>Today is payday!</p>
}
else
{
        <p>Today is a normal day.</p>
}


<p>Take care!</p>
```

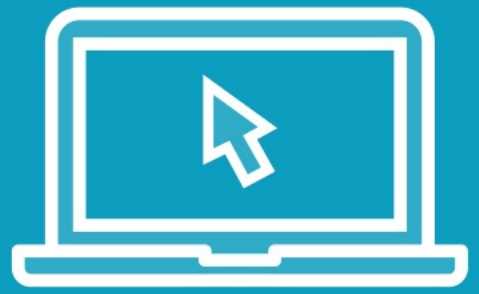
- Standard HTML
- Razor code block for C# logic
- Razor if/else statement
- Conditionally rendered HTML
- Automatic transition back to HTML

Razor Pages Is Built on ASP.NET Core

Razor Pages:
User interface features

ASP.NET Core:
Foundational features

Demo



Installing and setting up Razor Pages

Installing Razor Pages and ASP.NET Core 6.0

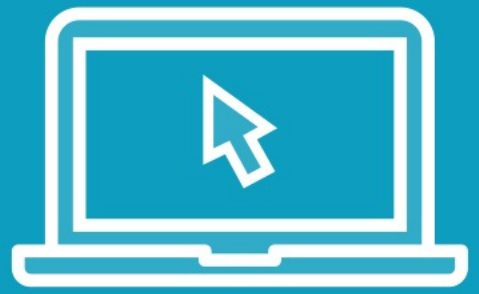
Install Visual Studio 2022

Razor Pages and ASP.NET 6.0 come bundled with it

Install ASP.NET Core 6.0 Directly

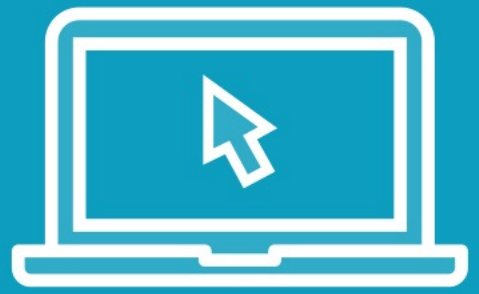
A great option when using VS Code or other IDEs

Demo



Creating and exploring a Razor Pages project

Demo



Handling requests with Razor Pages

Exploring Razor Pages and ASP.NET Core

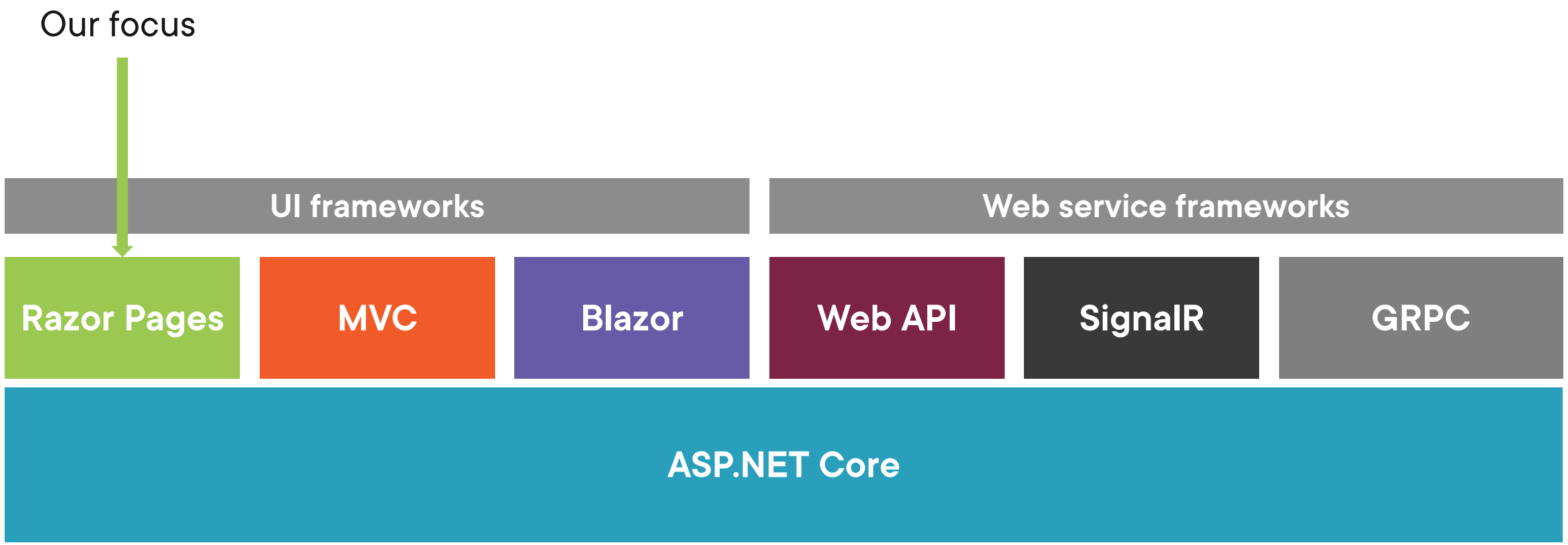
ASP.NET Core

A foundational framework for building cross platform, high performance, modern web applications.

ASP.NET Core Capabilities

Dependency injection	Routing	Authentication
Data access	Cross-platform support	Caching

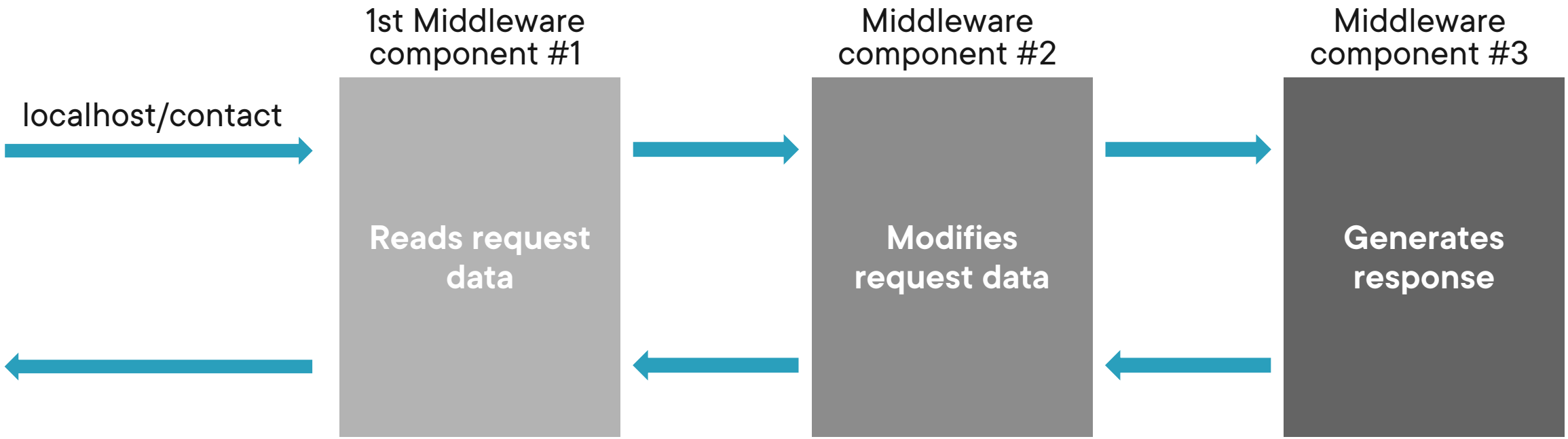
Razor Pages Is One of Many ASP.NET Core Tools



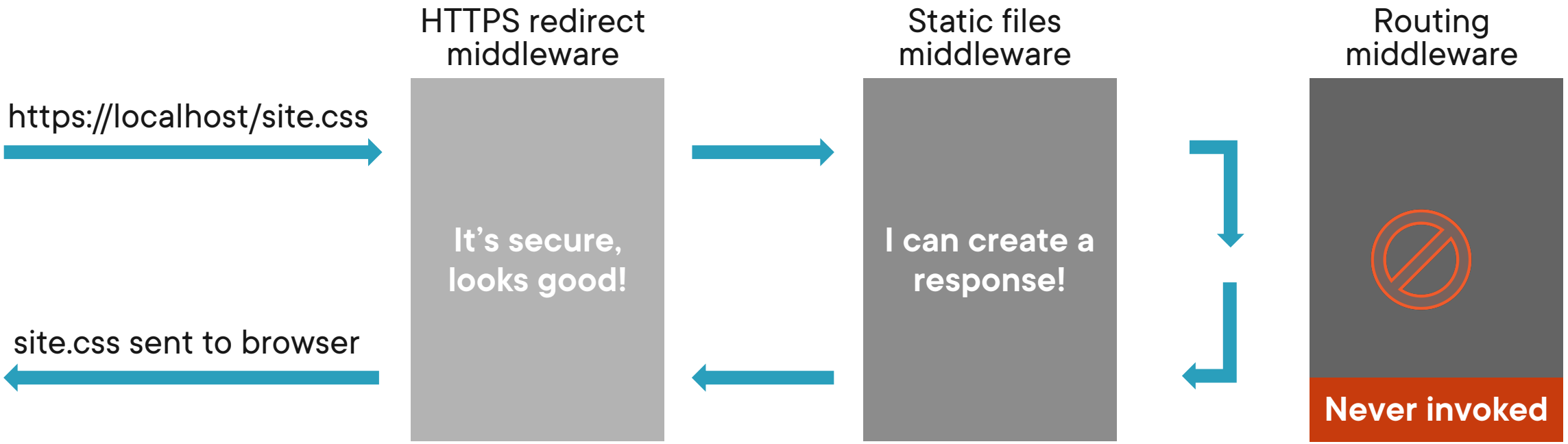
ASP.NET Core Middleware

A series of components that form the application request pipeline.

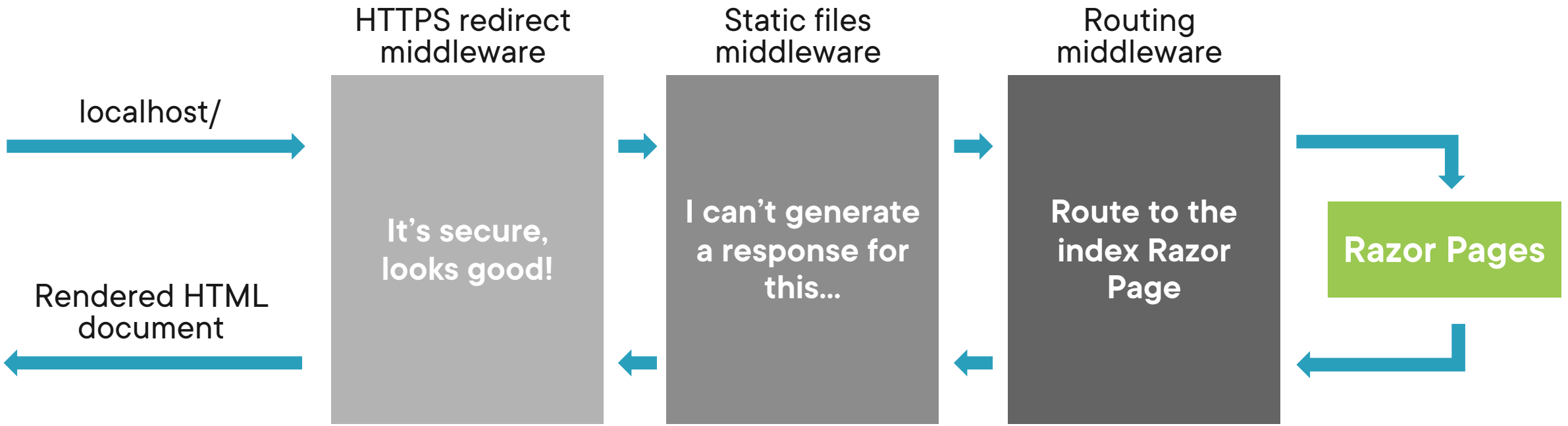
Understanding the Middleware Pipeline



Handling a Request with Middleware



Handling a Request with Middleware and Razor Pages

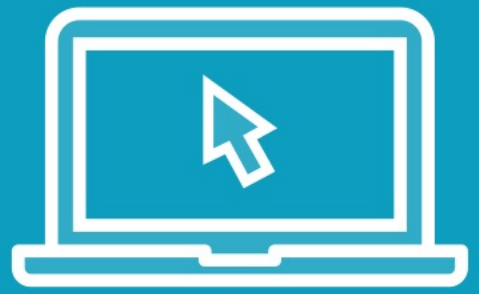


More Information

ASP.NET Core 6 Fundamentals

Gill Cleeren

Demo

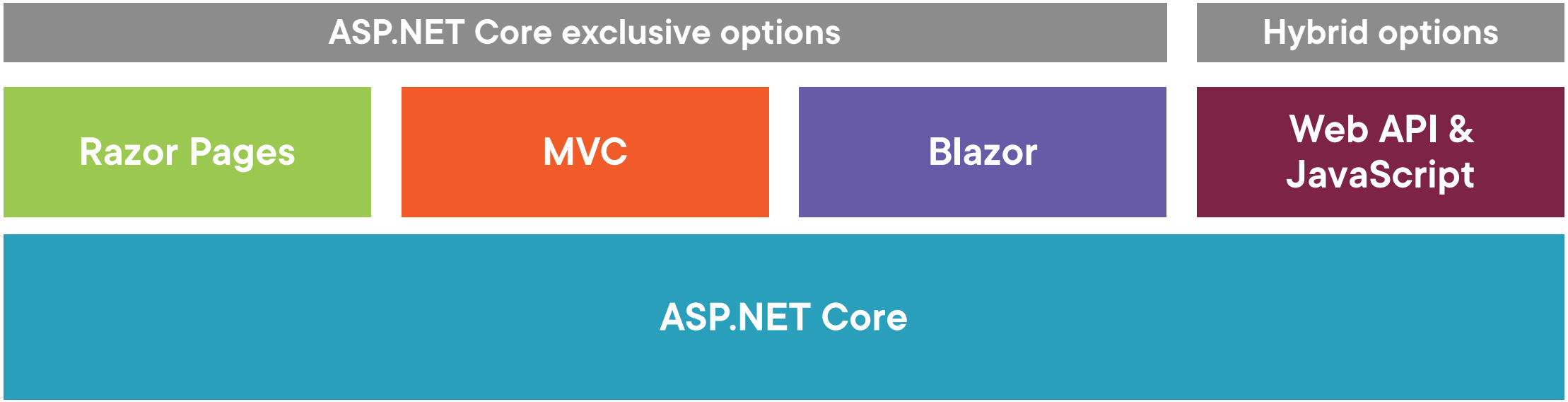


Working with middleware

Comparing Razor Pages

Our focus remains on Razor Pages.

UI Framework Options with ASP.NET Core



Exploring ASP.NET Core UI Options

Razor Pages

Simple, approachable framework for page-oriented workflows

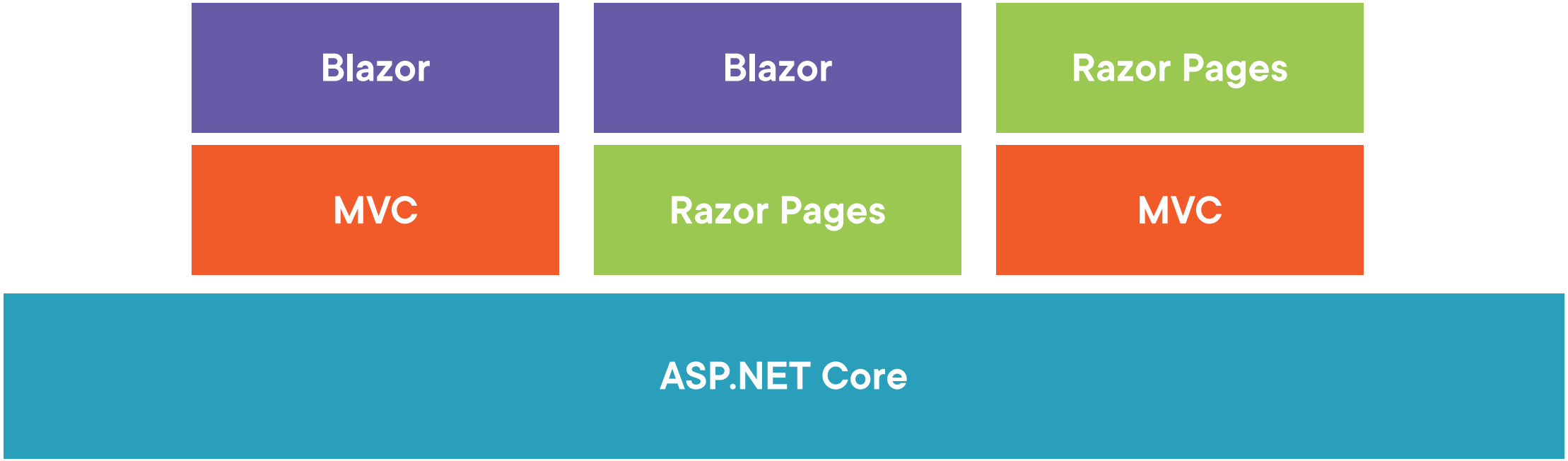
MVC

Utilizes the Model View Controller pattern for building complex apps

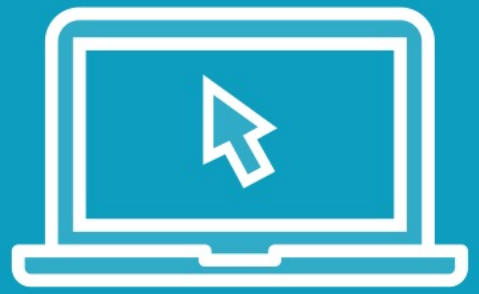
Blazor

A component driven framework for building highly dynamic UIs

Combining ASP.NET Core UI Tools



Demo



Setting up the sample project

Introducing the Wired Brain Coffee app

- Manage menu items
- View order data or reports
- Submit help tickets

GitHub Links

https://github.com/alex-wolf-ps/aspnet6-razor-pages-fundamentals

Summary

Razor Pages is a server-side, paged-based framework for building web UIs with ASP.NET Core

Razor Pages dynamically generate full page responses on the server to send to the browser

Page templates physically live on the server and correspond directly to application URLs

Razor templating syntax allows us to dynamically generate data-driven markup

ASP.NET Core provides many powerful foundational features of Razor Pages

Many ASP.NET Core features are implemented through Middleware

Other ASP.NET Core tools such as MVC and Blazor can also be used to build web UIs