Blazor: Getting Started

BUILDING YOUR FIRST BLAZOR APPLICATION

Gill Cleeren
CTO XPIRIT BELGIUM

@gillcleeren www.snowball.be

Overview

What will you learn from this course?

Hello Blazor

The different hosting models of Blazor

Understanding File → New Project

Creating a first Blazor app

Debugging a Blazor app

What Will You Learn from This Course?

This course will teach you how to build your first Blazor application.

How We Will Approach This Course

File → New Project...

Quite some code...

And then, a full Blazor application!

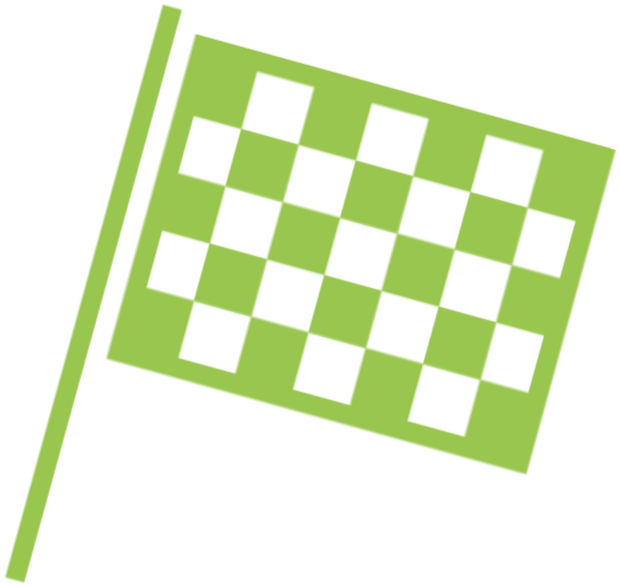
What I Assume You Know

HTML and CSS

C#

Some Razor is recommended

What You Need to Have Installed



Visual Studio 2019 (16.6 or higher)

- https://visualstudio.microsoft.com/vs/

.NET Core 3.1 SDK

- https://dotnet.microsoft.com/download/dotnet-core/3.1

.NET 5 compatible

A browser

The Scenario: Bethany's Pie Shop HRM

- List of employees
- Detail page
- Add new employee
- Navigation

Demo

Taking a look at the finished application

Blazor: Getting Started

Start Course Bookmark Add to Channel Download Course

Table of contents Description Transcript Exercise files Discussion Recommended

Hello Blazor

Blazor is a framework
to build interactive web UIs
using C# and HTML.

Hello Blazor

Based on WebAssembly or run on server

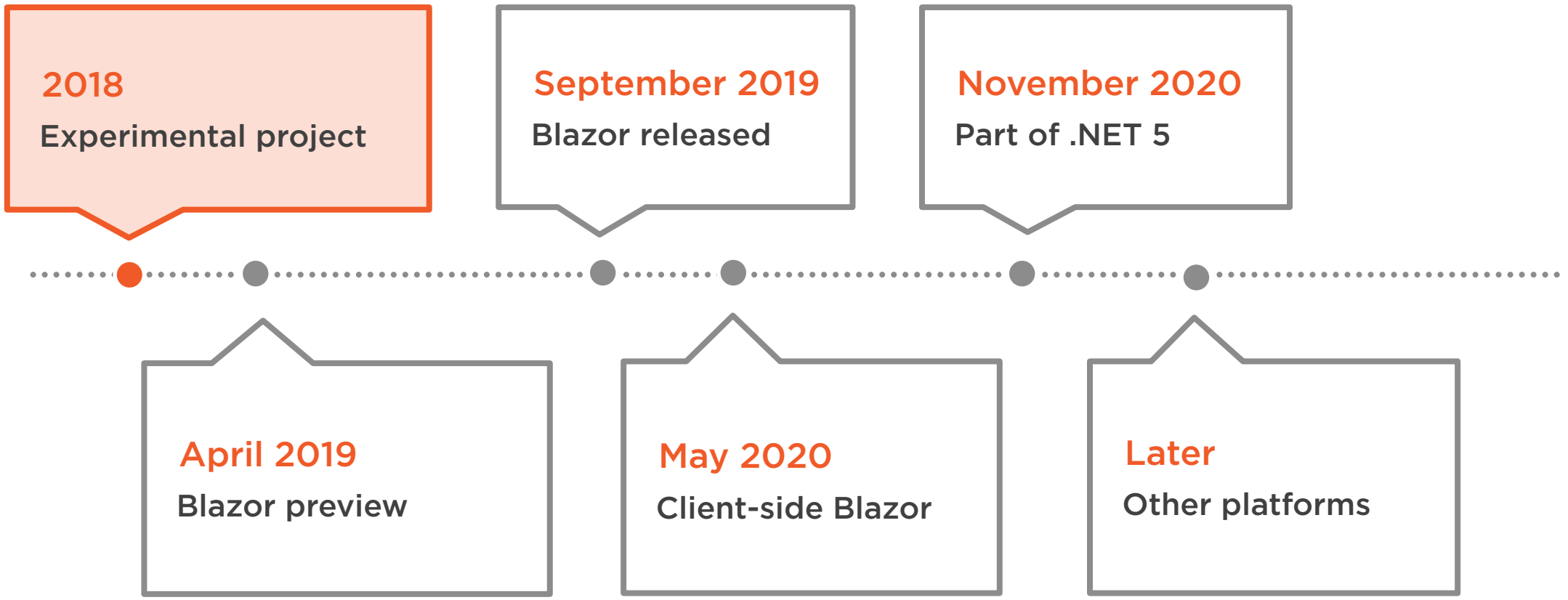
No plugin, based on web standards

Integrate with JavaScript

Benefits of Visual Studio and .NET including performance and libraries

Leverage your C# skills
to build interactive web applications.

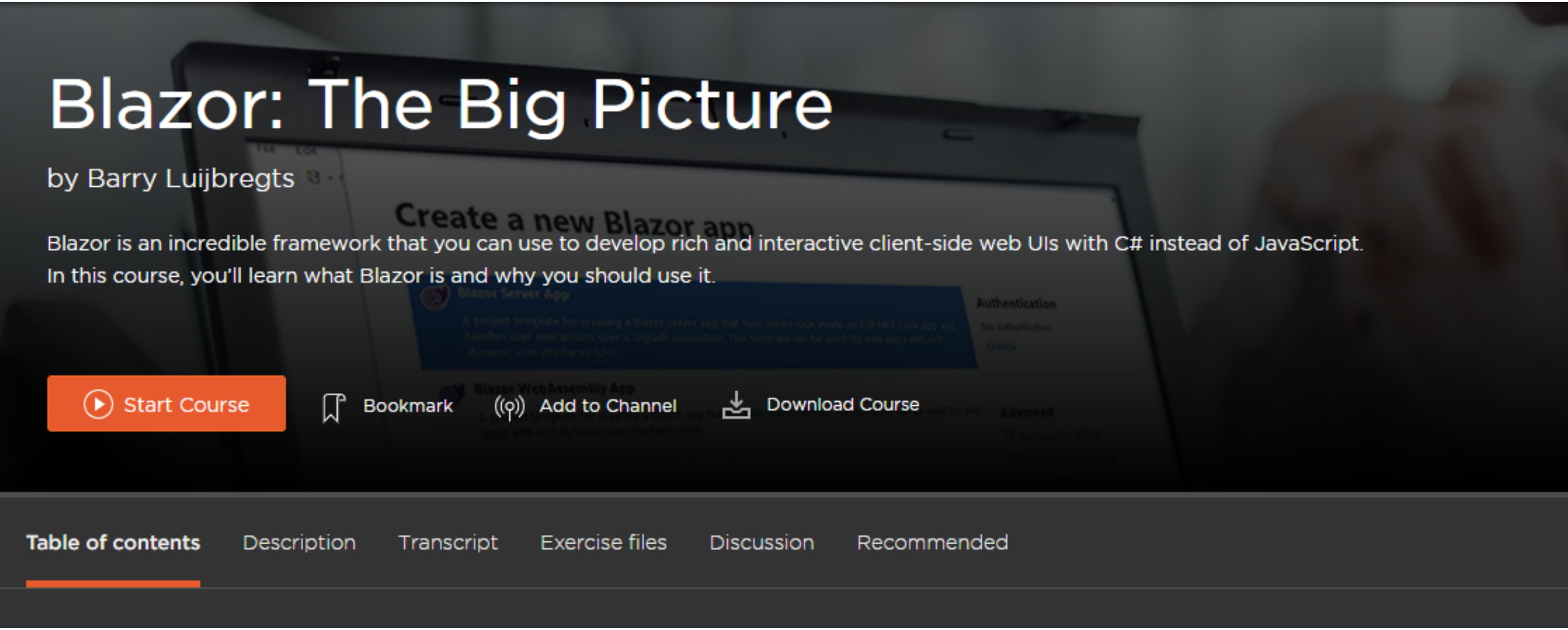
Blazor Roadmap



Demos are built using .NET Core 3.1

All code is compatible with .NET 5

.NET 5 module at the end of this course



https://www.pluralsight.com/library/courses/blazor-big-picture

The Different Hosting Models of Blazor

Different Flavors of Blazor

Client-side

Server-side

Client-side Blazor

Blazor app
WebAssembly

Runs on all modern browsers

No .NET required on server

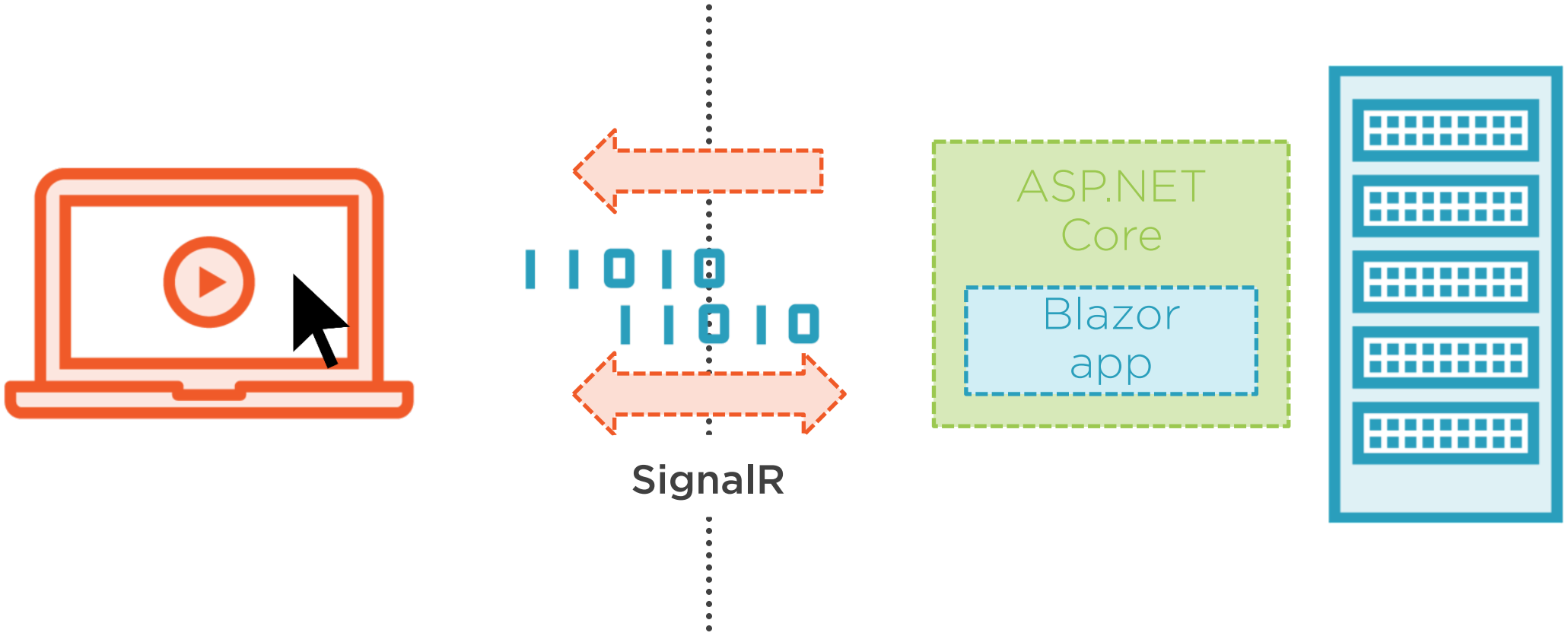
SPA user experience

Older browsers might not be supported

Initial app download is larger

Debugging support

Server-side Blazor



- Small download
- Works with all server-side APIs
- Full debugging support
- Blazor apps in non-supported browsers
- No offline support
- Network delay
- Scalability, although not a big problem

We'll use client-side Blazor and convert to server-side later.

Let’s now dive into Blazor code!

Understanding File → New Project

Visual Studio Templates

Client-side WebAssembly

Standalone

Client-side WebAssembly

ASP.NET Core Hosted

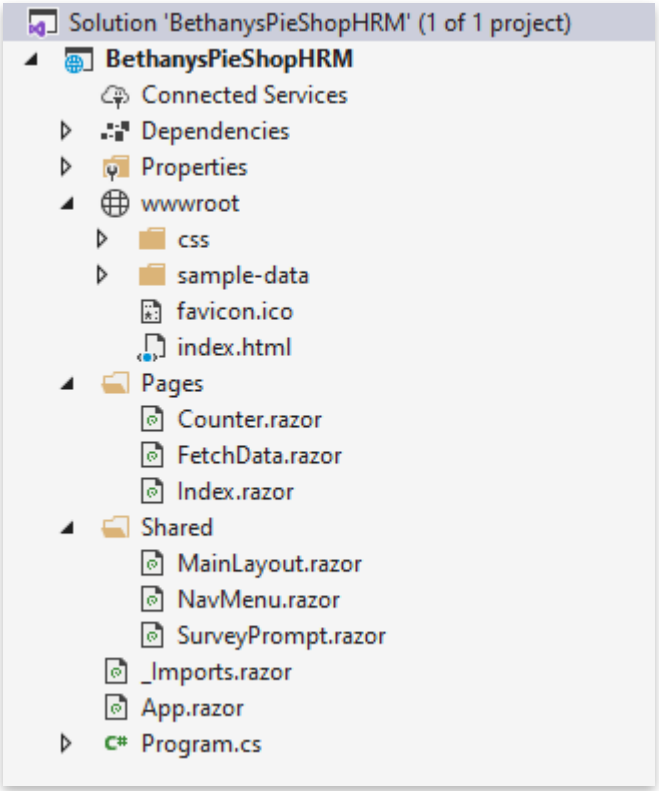
Server-side

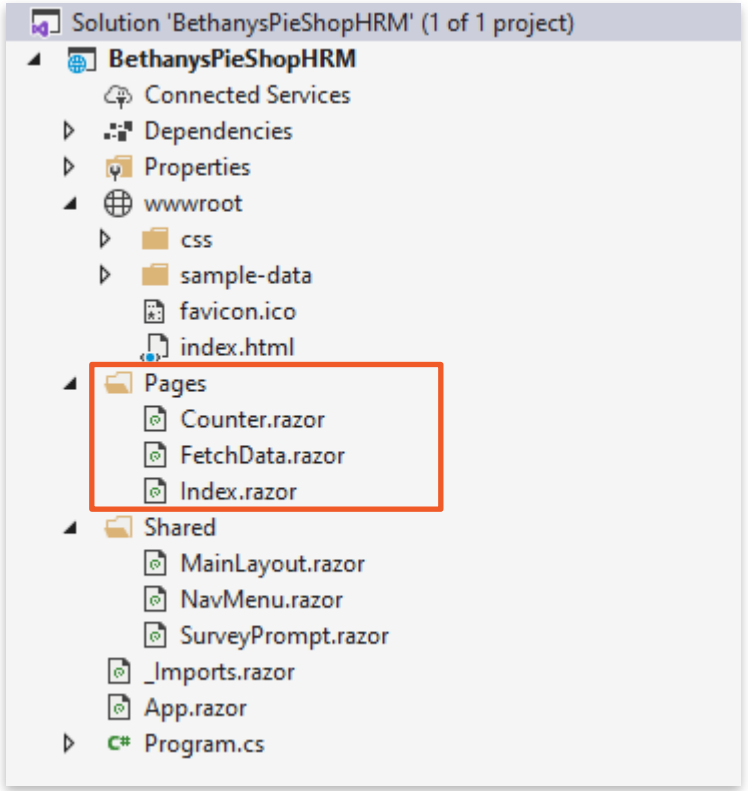
New Project

C# and .razor files

Structure similar to ASP.NET Core project

- Program.cs





*.razor files

Components are building blocks

Name must start with uppercase

Class generated upon compilation

Looking at a the First Component

```
@page "/counter"
<h1>Counter</h1>
<p>Current count: @currentCount</p>
<button class="btn btn-primary" @onclick="IncrementCount">Click me</button>
@code {
    int currentCount = 0;

    void IncrementCount()
    {
        currentCount++;
    }
}
```

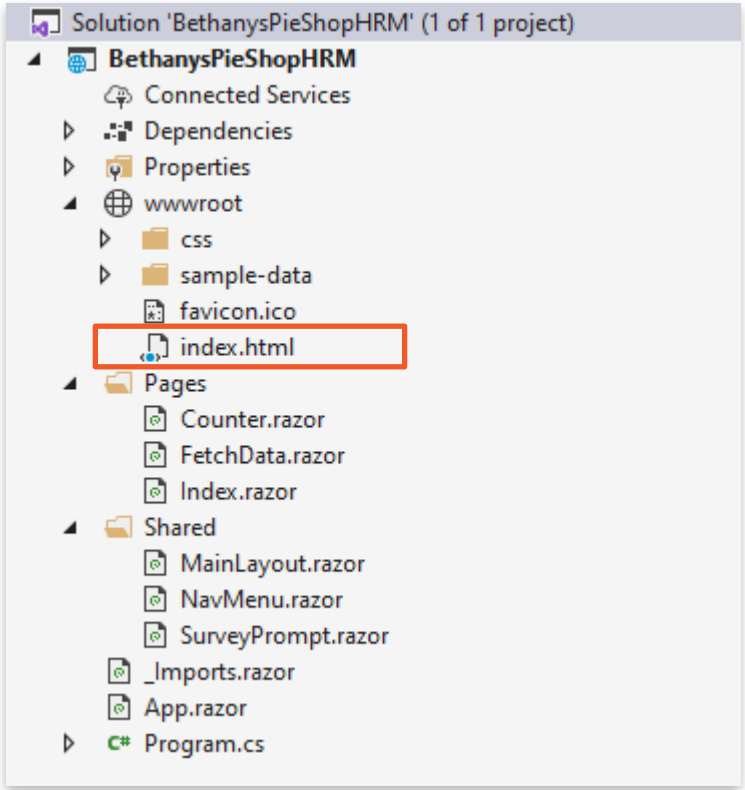
```
@page "/"

<h1>Hello, world!</h1>

Welcome to your new app.

<Counter />
```

Using a Component



Hosting page

Plain HTML

Trigger loading of your Blazor app

- blazor.webassembly.js

Demo

Creating a new project

Looking at the created files

New to ASP.NET Core?
Take a look at the
ASP.NET Core Learning Path!

Creating Your First Blazor App

Using Code

Mixed approach using @code

"Code behind" using partial

Mixed Approach

```
@page "/counter"

<h1>Counter</h1>

<p>Current count: @currentCount</p>

<button class="btn btn-primary" @onclick="IncrementCount">Click me</button>

@code {
    int currentCount = 0;


    void IncrementCount()
    {
        currentCount++;
    }
}
```

```
public partial class EmployeeOverview
{

}
```

Using Partial Classes

Demo

Creating your first app

Demo

Adding your own layout

Demo

Debugging a Blazor app in Visual Studio

Summary

Blazor allows creating rich experiences in the browser using C# and HTML

Server-side and client-side

Components are a building block

Up next:

Working with data