Introduction to Razor – Episode 2

DEBASIS SAHA

Solution Architect, MCT, MVP

DEBASIS SAHA

Solution Architect, MCT, MVP

Author, Speaker, Blogger

Currently I am working as a Solution Architect at WeFiveSoft Pvt Ltd, Pune. I am a Microsoft Certified Trainer & C# Corner MVP. Having 12+ years of experience in the IT Industry. I am always been a great fan of Microsoft Technologies and love working on them. I have expertise in Asp.Net, .NET Core, MVC, C#, Azure, Asp.Net Core, SQL Server, ReactJs, Angular, and NoSQL Databases like MongoDB, CosmosDB, etc. He loves to write articles about these technologies.

debasis.ds@outlook.com

https://www.linkedin.com/in/sahadebasis/

@debasiskolsaha

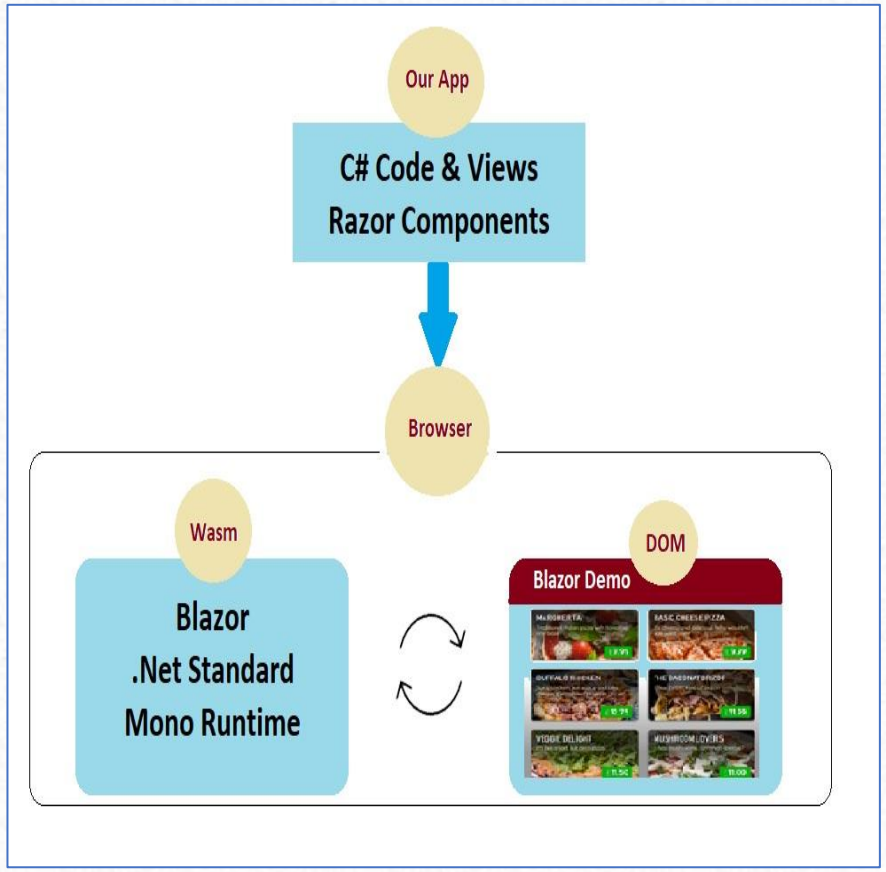
Agenda

- What is Blazor Framework?
- What is Razor Syntax?
- Overview of Razor Syntax
- Use Razor Syntax in Blazor Project
- Use Class Object Structure in Razor Syntax
- Use Conditional Statement in Razor Syntax
- Use Loop Statement in Razor Syntax
- Use Markup String in Razor Syntax

What is Blazor Framework?

What is Blazor?

- Microsoft introduced the **Blazor Framework** in the year of **2018**.
- Blazor is an open-source .Net web framework run on the Client-Side.
- Blazor combines two words – **Browser** and **Razor** (used as .Net HTML View generate engine).
- Can use **C#, HTML,** and **CSS** to create the **web-based UI** components
- Blazor executes the **Razor Views directly** on the **client's browser**.
- Can build **server-side** and **client-side** code with the help of the C#.
- Can use the **same class or code** as a **shared code or library**.



Benefits of Blazor Framework

- Helps us to **develop single-page application**s just like React, Vue, and Angular.
- It is **fast, reusable, and open source**, providing the way for the tremendous support from the community.
- **Server-side** rendering
- Provide **component-based architecture** for building reusable UI sections.
- Forms and validation
- Dependency injection
- Routing
- Publishing and app size reducing
- Rich IntelliSense and tooling
- Live reloading in the browser during development
- Can run the application on older browsers also by using asm.js.

When we can use Blazor

- If the target users use the **browser that supports WebAssembly**, then we can use the **Blazor Framework**.
- We can also use **Blazor for Desktop or Mobile** except for the web. However, this functionality is still in preview. But we need to use either Electron or WebWindows in this scenario.
- We can use the **Blazor Server app** for any type of **intranet or internal** application.
- If the application **does not require many concurrent users**, then we can use the **Blazor Server App**.
- If the application requires a **large volume of concurrent users at a time**, then **Blazor WebAssembly** is the best option.
- Also, if we want to implement the **offline or PWA-based** functionality, **Blazor WebAssembly** needs to be selected.

What is Razor Syntax?

What is Razor Syntax?

- Razor is **a markup syntax for embedding .NET based code into webpages**.
- The Razor syntax consists of Razor markup, C#, and HTML.
- Files containing Razor generally have a .cshtml file extension.
- The **default Razor** language is **HTML**. Rendering HTML from Razor markup is no different from rendering HTML from an HTML file. HTML markup in .cshtml Razor files is rendered by the server unchanged.

Razor Syntax

- Razor is a **simple programming syntax** for embedding server code in web pages.
- Razor syntax is based on the **ASP.NET framework**.
- The **Razor syntax provides** the power of **ASP.NET**, but is using a simplified syntax that's **easier to learn**.
- Razor web pages can be described as HTML pages with two kinds of content: **HTML content** and **Razor code**.
- When the **server reads the page**, it runs the **Razor code first**, before it sends the HTML page to the browser.
- The code that is **executed on the server** can perform tasks that **cannot be done in the browser**, for example accessing a server database.
- **Server code** can create **dynamic HTML con**tent on the fly.

Razor Syntax

- Razor supports C# and uses the @ symbol to transition from HTML to C#.
- Razor evaluates C# expressions and renders them in the HTML output.
- When an @ symbol is followed by a Razor reserved keyword, it transitions into Razor-specific markup.
- Otherwise, it transitions into plain HTML.

Demo

Use Razor Syntax in Blazor Project

Demo

Use Class Object Structure in Razor Syntax

Demo

Use Conditional Statement in Razor Syntax

Demo

Use Loop Statement in Razor Syntax

You can connect with me through –


debasis.ds@outlook.com


https://www.linkedin.com/in/sahadebasis


https://github.com/debasis-saha


@debasiskolsaha

Source Code of the Session will available at GitHub Repo –

https://github.com/debasis-saha/csharcorner-blazor-series

Thank You

Questions