**Blazor Integration to Existing ASP.Net Core Web App**

**Author:** Sampath Mannevaru

**Introduction**

This article can be used mainly to Integrate Blazor Server-side components.There are a lot of applications that use ASP.Net MVC as their UI project. There are many good reasons for having Blazor components on the ASP.Net MVC project. Some include writing UI functionality in C# and the ability to componentize your .Net application UI architecture.

**Integration of Blazor to the existing app as partial page components.**

(This can replace partial views or completely develop new components that can be used in the existing app.)

In startup.cs file, add the below code.

* ConfigureServices method, which adds the server-side Blazor.

services.AddServerSideBlazor();

* Configure the below methods, which map Blazor.

endpoints.MapRazorPages();

endpoints.MapBlazorHub();

* Add \_Imports .razor file to the project & Place this below code in it.

@using System.Net.Http

@using Microsoft.AspNetCore.Authorization

@using Microsoft.AspNetCore.Components.Authorization

@using Microsoft.AspNetCore.Components.Forms

@using Microsoft.AspNetCore.Components.Routing

@using Microsoft.AspNetCore.Components.Web

@using Microsoft.JSInterop

@using WebProject //in my case this is my current project.

* Under the \_Layout.cshtml file, at the end of the body.

Add Blazor reference

<script src="~/\_framework/blazor.server.js"></script>

* Create the folder Pages under the project, as Blazor will look for this folder.

We will place our Blazor components here.

**That's It! We are all set!**

**Important:**

1. Blazor server-side works by creating the SignalR connection from browse to server.
2. Layout changes, event handling, and JavaScript calls are done through a SignalR connection.

We have to make this connection to be laid out for that.

The controller action which is returning the views, needs to be added as a

[Route("/{<ActionMethodName, Which returns view>}")]

In my case, I have added [Route("/GetExampleComponent")] on the Example

controller class and the GetExampleComponentAsync Action method.

So whenever the blazor.server.js file is loaded in the browser, it will try to

establish a SignalR call and treat the URL before "/" as the host address.

Example:

In my case, if I replace the above route with

[Route("/Example/GetComponent")],

then the SignalR connection is not laid, and we cannot use Blazor components.

1. For Blazor, as a new Page.

Add the below code in the configure method in startup.cs file.

endpoints.MapFallbackToPage("/\_Host");

Add the \_Host.cshtml file in /Pages Folder. This is like the index page or \_Layout.cshtml for Blazor.

Add the App.razor component to the project, where it routes to the components.

**Advantages of Blazor:**

* We can use C# features on the UI without client-side scripting.
* Component-driven frameworks like angular & react, are easily

understood & testable.

* We want to convert from the server side to the web assembly, we can easily move.
* Two-way binding.

**References:**

* https://www.telerik.com/blogs/why-you-should-use-blazor-over-javascript-frameworks-to-build-your-single-page-application