Converting .NET Framework to .NET Core Notes:

1. Set Portability Analyzer to .NET 5.0 Target
2. Run Portability Analyzer and look at resulting report
3. In the case of SignalRChat2, report shows two unresolved assemblies:

"Microsoft.AspNet.SignalR.Core, Version=2.2.2.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35",

"Microsoft.Owin, Version=4.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"

1. Information to move from ASP.NET to ASP.NET core is found here:

<https://docs.microsoft.com/en-us/aspnet/core/migration/proper-to-2x/?view=aspnetcore-5.0>

1. We need to figure out the replacements of these dependencies in .NET. Looking for replacement of Microsoft.AspNet.SignalR.Core we find that there is an equivalent library:

<https://docs.microsoft.com/en-us/aspnet/core/tutorials/signalr?view=aspnetcore-5.0&tabs=visual-studio>

1. For the Microsoft.Owin replacement we find that ASP.NET core has a built-in mechanism to provide web services
2. The .NET Core project settings is greatly simplified compared to .NET Framework. The only thing need is the right project settings:

<Project Sdk="Microsoft.NET.Sdk.Web">

<PropertyGroup>

<TargetFramework>net5.0</TargetFramework>

</PropertyGroup>

</Project>

1. .NET Core project needs a Program.cs class with a static main method as an entry point. A general definition of program class can be found here: [App startup in ASP.NET Core | Microsoft Docs](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/startup?view=aspnetcore-5.0)
2. We move index.html and dependent scripts file to the wwwroot folder as those are web resources
3. We update the index.html file to use the proper signalr method calls