

Blazor, a new framework for browser-based .NET apps - Steve Sanderson

From 08:00-09:00 if you haven't already; go watch:
<https://www.youtube.com/watch?v=Khn7sDUSEJM>

Web Applications with C#

Rasmus Lystrøm
Associate Professor
ITU
rne@itu.dk

The image shows a Visual Studio Code editor with two C# files open: ProgramTests.cs and Program.cs. The ProgramTests.cs file contains a namespace HelloWorld.Tests with a public class ProgramTests. It has a [Fact] attribute and a public void Main_given_no_args_p method. The Program.cs file contains a namespace HelloWorld with a public class Program and a public static void Main method that writes "Hello World!" to the console. The terminal window at the bottom shows the output of a dotnet test command, indicating that the test was successful.

```
ProgramTests.cs
1 using System;
2 using System.IO;
3 using Xunit;
4
5 namespace HelloWorld.Tests
6 {
7     0 references | Run All Tests | Debug All Tests
8     public class ProgramTests
9     {
10         [Fact]
11         0 references | Run Test | Debug Test
12         public void Main_given_no_args_p
13         {
14             // Arrange
15             using var writer = new StringWriter();
16             Console.SetOut(writer);
17
18             // Act
19             Program.Main(new string[0]);
20
21             // Assert
22             var output = writer.GetStringBuilder().ToString();
23             Assert.Equal("Hello World!", output);
24         }
25     }
26 }
```

```
Program.cs
1 using System;
2
3 namespace HelloWorld
4 {
5     1 reference
6     public class Program
7     {
8         1 reference
9         public static void Main(string[] args)
10         {
11             Console.WriteLine("Hello World!");
12         }
13     }
14 }
```

```
Terminal
1: pwsh
Loading personal and system profiles took 1008ms.
C:\HelloWorld> dotnet test
Test run for C:\HelloWorld\HelloWorld.Tests\bin\Debug\netcoreapp3.0\HelloWorld.Tests.dll (.NETCoreApp, Version=v3.0)
Microsoft (R) Test Execution Command Line Tool Version 16.3.0
Copyright (c) Microsoft Corporation. All rights reserved.

Starting test execution, please wait...

A total of 1 test files matched the specified pattern.

Test Run Successful.
Total tests: 1
Passed: 1
Total time: 3,4618 Seconds
C:\HelloWorld>
```

Agenda

Message from Study & Career Guidance

Catch up on last weeks lecture

Web Frameworks for .NET

Other web frameworks

Building your own components

How to choose a framework

(The Gilded Rose Recap)

Exam preparation and Efficient Learning workshop

Study & Career Guidance are hosting the workshop [Exam preparation and Efficient Learning](#).

It is held on Zoom, today at 16:00-18:00.

Sign up: [here at ITU Student](#).

Catch-up on last weeks lecture

For a demo on Xamarin.Forms go to:

<https://github.com/ondfisk/BDSA2019> and go to BDSA2019.Lecture10

Web Frameworks for .NET

Blazor

Blazor is a framework for building interactive client-side web UI with .NET:

- Create rich interactive UIs using C# instead of JavaScript.
- Share server-side and client-side app logic written in .NET.
- Render the UI as HTML and CSS for wide browser support, including mobile browsers.
- Integrate with modern hosting platforms, such as Docker.

```
$ dotnet new blazorserver
```

```
$ dotnet new blazorwasm
```

Source: <https://blazor.net> and <https://docs.microsoft.com/en-us/aspnet/core/blazor>

gRPC Remote Procedure Calls

gRPC is a modern, open source, high-performance remote procedure call (RPC) framework that can run anywhere. gRPC enables client and server applications to communicate transparently and simplifies the building of connected systems.

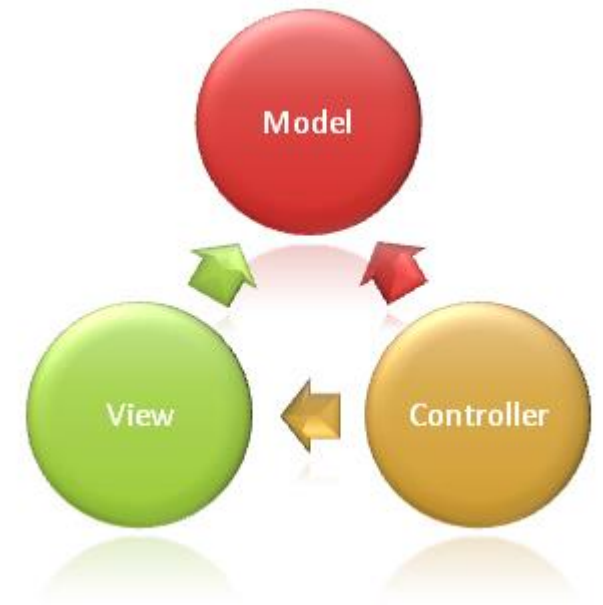
```
$ dotnet new grpc
```

Source: <https://github.com/grpc/grpc> and <https://docs.microsoft.com/en-us/aspnet/core/grpc>

ASP.NET Core MVC

ASP.NET Core MVC is a rich framework for building web apps and APIs using the Model-View-Controller design pattern.

```
$ dotnet new mvc
```



Source: <https://docs.microsoft.com/en-us/aspnet/core/mvc/overview>

Honorable mentions

Web Forms

Deprecated – was the default in ASP.NET from 2002 to around 2010-2012-ish

Razor Pages

Simple page centric approach without all the fuss

SignalR

Real-time functionality for existing apps (core part of Blazor Server)

Demo

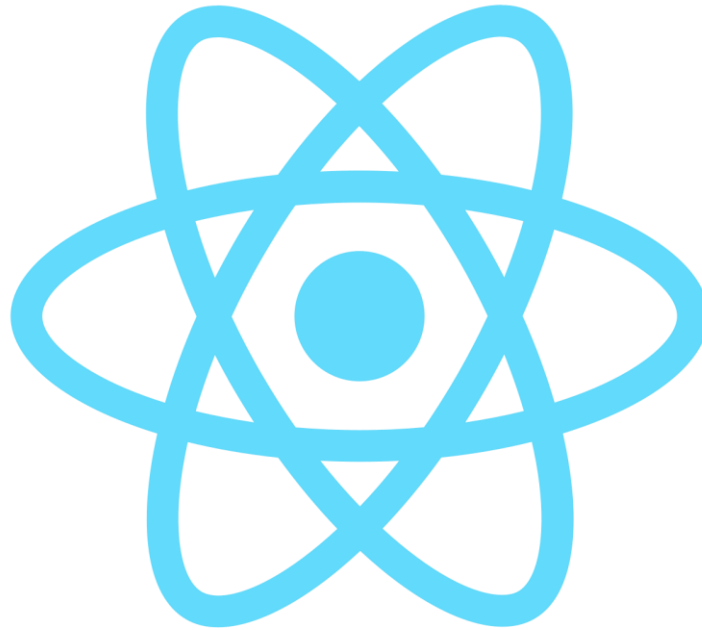
ASP.NET Web Applications

Other Web Frameworks

Other Web Frameworks



Angular
`$ dotnet new angular`



React
`$ dotnet new react`



Vue

Other Web Frameworks

Written in JavaScript (or TypeScript)

Backend could be:

- ASP.NET Core Web API
- Node.js
- Anything

No code sharing with backend – context switching hard.

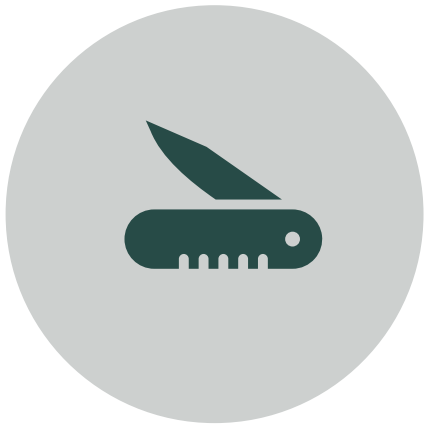
Building your own components

Building your own components



Source: <https://www.pmstudio.be/dont-re-invent-the-wheel/>

Component Vendors



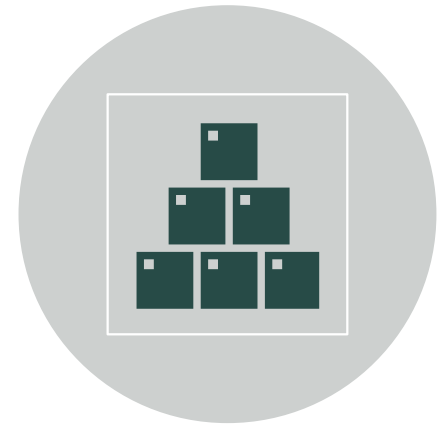
TELERIK:

[HTTPS://WWW.TELERIK.COM/](https://www.telerik.com/)



DEVEXPRESS:

[HTTPS://WWW.DEVEXPRESS.COM/](https://www.devexpress.com/)

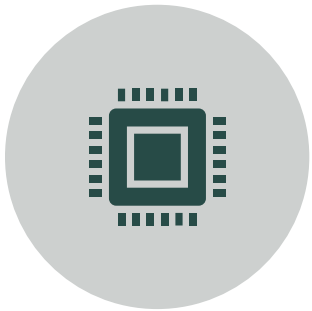


RADZEN:

[HTTPS://BLAZOR.RADZEN.COM/](https://blazor.radzen.com/)

How to choose a framework

How to choose a framework



INSTALL VISUAL STUDIO
PROFESSIONAL/ENTERPRISE/FOR
MAC



TEST OUT ALL THE TEMPLATES ONE
BY ONE



GET YOUR OWN FEEL FOR
PROS/CONS OF EACH FRAMEWORK
FOR YOUR SCENARIO

No, really



Pick the one where you believe you
will be the most productive



Don't be afraid to change your mind



Be prepared: The next great thing
which is better than all the others will
arrive tomorrow or next year approx.