

# Blazor WebAssembly in .NET 5



Flash is dead!..  
Long live WebAssembly!

**Blazor WebAssembly –  
потенциально  
приложение в  
браузере**

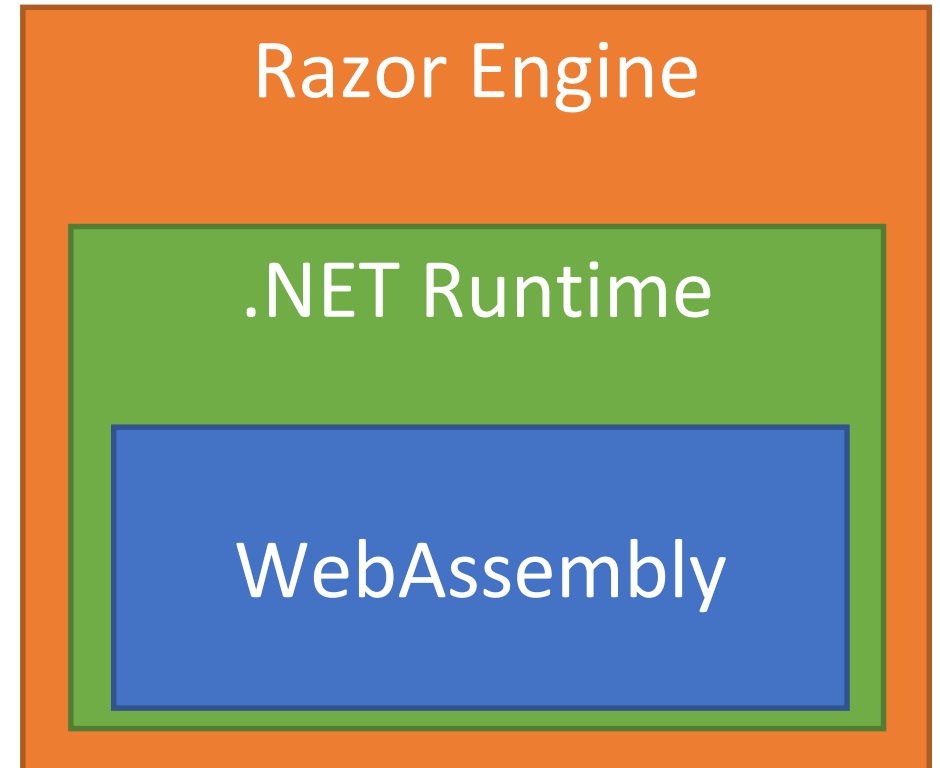
**Server-side Blazor –  
эволюция razor  
pages... и еще один  
web фреймворк**

# Blazor






- Открытый web стандарт (поддержка в браузерах)
- Front-end - смесь HTML, CSS и C# + JS interop
- What you *code* is what you get
- Back-end - ASP.NET Core (по умолчанию) или любой другой сервер или Web API

[Tutorial: Building a static web app with Blazor in Azure Static Web Apps | Microsoft Docs](#)

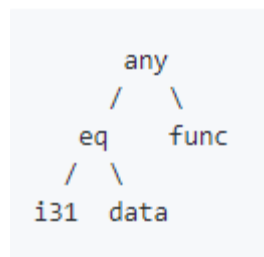
[Blazor WebAssembly on Azure Static Web Apps - Scott Hanselman's Blog](#)



# WebAssembly

	Your browser	 Chrome <sup>86</sup>	 Firefox <sup>79</sup>	 Safari <sup>13.1</sup>	 Wasmtime <sup>0.22</sup>	 Wasmer <sup>1.0</sup>
Standardized features						
JS BigInt to Wasm i64 integration	✓	✓	✓	✗	n/a	n/a
Bulk memory operations	✓	✓	✓	✗	✓	✓
Multi-value	✓	✓	✓	✓	✓	✓
Import & export of mutable globals	✓	✓	✓	✓	✓	✓
Reference types	✗	⌚	✓	⌚	✓	✗
Non-trapping float-to-int conversions	✓	✓	✓	✗	✓	✓
Sign-extension operations	✓	✓	✓	✗	✓	✓
In-progress proposals						
Exception handling	✗	✗	✗	✗	✗	✗
Module Linking	✗	✗	✗	✗	⌚	✗
Fixed-width SIMD	✗	⌚	⌚	✗	⌚	⌚
Tail calls	✗	✗	✗	✗	✗	✗
Threads and atomics	✓	✓	✓	✗	⌚	⌚

Иерархия классов в Wasm  
из GC proposal  
[gc/MVP.md at master · WebAssembly/gc · GitHub](https://github.com/WebAssembly/gc/blob/master/MVP.md)



# .NET Runtime

Mono Runtime в режиме интерпретатора IL.

AOT(Native) Compilation первый пункт в .NET 6 roadmap

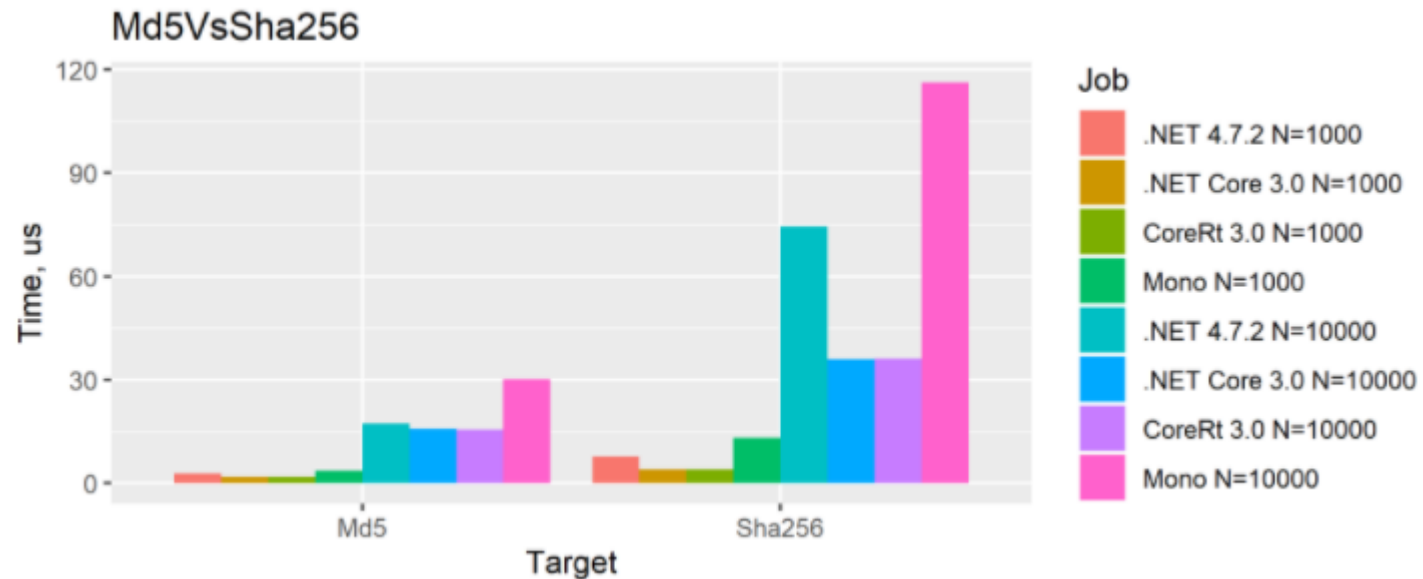


График с: [BenchmarkDotNet](#)

# Transport

## HTTP

### REST API

Старый, хорошо знакомый,  
“односторонний” и медленный

### gRPC-Web

Свежий, более быстрый,  
“односторонний” и все еще не до  
конца gRPC

## TCP

### SignalR

Старый, но менее распространенный  
чем REST, “двусторонний”, с  
возможностью откатиться на HTTP,  
back-end только ASP.NET

### WebSockets

Старый добрый TCP с минимумом  
web-магии,  
текстовый или бинарный формат  
данных, “двусторонний”,  
максимальный результат при  
маскированных вложениях

# Hello World? Razor синтакс

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



# Наиболее подходящий шаблон?

## MVVM

vs

## MVC / MVP

- + Bi-directional data binding
- ICommand support

Classic Model-View-  
Controller  
~~ASP.NET MVC~~

ReactiveUI???

[ReactiveUI - ReactiveUI On The Web with Blazor](#)

# Counter.razor – разметка

```
<h1 class="counter-header">Counter</h1>
```

```
<p class="counter">Current count: @_currentCount</p>
```

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

# Counter.razor.cs – code-behind

```
public partial class Counter
{
    private int _currentCount = 0;

    private void IncrementCount()
    {
        _currentCount++;
    }
}
```

# Counter.razor.css - стиль

```
.counter{  
    border: solid red 1px;  
    font-size: 1.2rem;  
}  
  
.counter-header {  
    color: darkblue;  
    font-size: 2rem;  
    border-bottom: solid red 1px;  
}
```

# CounterViewModel.cs

```
public class CounterViewModel : INotifyPropertyChanged
{
    private int _counter;
    public event PropertyChangedEventHandler PropertyChanged;

    public int Counter
    {
        get => _counter;
        set
        {
            if (_counter == value) return;
            _counter = value;
            NotifyPropertyChanged();
        }
    }

    public void IncrementCounter() => Counter++;

    private void NotifyPropertyChanged([CallerMemberName] string propertyName = "")
    {
        PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
    }
}
```

# Counter.razor.cs => MVVM?

```
public partial class Counter
{
    private bool _changed;
    [Inject]
    private CounterViewModel ViewModel { get; set; }

    protected override void OnInitialized()
    {
        ViewModel.PropertyChanged += (s, e) => StateHasChanged();
    }
}
```

# Counter.razor

```
<h1 class="counter-header">Counter</h1>
```

```
<p class="counter">Current count: @ViewModel.Counter</p>
```

```
<button class="btn btn-primary" @onclick="ViewModel.IncrementCounter">Click me</button>
```

# Program.cs

```
public class Program
{
    public static async Task Main(string[] args)
    {
        var builder = WebAssemblyHostBuilder.CreateDefault(args);
        builder.RootComponents.Add<App>("#app");

        builder.Services
            .AddScoped(sp => new HttpClient { BaseAddress = new
Uri(builder.HostEnvironment.BaseAddress) })
            .AddTransient<CounterViewModel>();

        await builder.Build().RunAsync();
    }
}
```



# Вопрос производительности

Стоит ли ожидать 20 кратное падение производительности по сравнению с десктопом?

[Developer Decries WinForms-to-Blazor Performance Degradation -- Visual Studio Magazine](#)

Мой заказчик не видит разницы...

Жду ли я AOT-compile?.. Конечно

# Component-based оптимизация отрисовки

```
public partial class Counter
{
    [Inject]
    private CounterViewModel ViewModel { get; set; }
    private bool _changed;

    protected override void OnInitialized()
    {
        ViewModel.PropertyChanged += (s, e) => Render();
    }

    protected override void OnAfterRender(bool firstRender) => _changed = false;
    protected override bool ShouldRender() => _changed;

    private void Render()
    {
        if (_changed) return;
        _changed = true;
        StateHasChanged();
    }
}
```

# RenderFragment-based оптимизация отрисовки

```
<div class="chat">
  @foreach (var message in messages)
  {
    @DisplayChatMessage(message)
  }
</div>
```

```
@code {
  RenderFragment<ChatMessage> DisplayChatMessage =
message => __builder =>
  {
    <div class="chat-message">
      <span class="author">@message.Author</span>
      <span class="text">@message.Text</span>
    </div>
  };
}
```

# Index.razor – динамическое обновление стиля

```
<style>
    .sidebar-adj-visibility {
        display: @_sidebarVisibility;
    }

    .btn-adj-visibility {
        display: @_btnVisibility;
        height: 3rem;
    }

    .preferences-adj-visibility {
        display: @_preferencesVisibility;
    }
    app {
        background-color: @Preferences?.General.MainBackground.Back;
        height: 100vh;
    }
</style>
```

# Альтернатива



[Bolero: F# in WebAssembly \(fsbolero.io\)](https://fsbolero.io) F# bindings to Blazor



[Uno Platform](#) – WinUI 3 port



[Unity Real-Time Development Platform | 3D, 2D VR & AR Engine](#)



[Wave Engine The graphics development engine for business and industry - WAVE cross platform engine for business and industry](#)

# Что дальше?



Blazor vs Angular / React /  
Vue  
no REST for the wicked...  
...until .NET 6



Blazor vs Xamarin / Flutter / Cocoa Touch



# Ссылки

<https://dotnet.microsoft.com/apps/aspnet/web-apps/blazor>  
(blazor.net) официальная документация

<https://blazor-university.com/> версия более похожая на документацию чем официальная документация

<https://blog.stevensanderson.com/> Стив разрабатывает Blazor и время от времени выкладывает примеры с функционалом до того как он станет частью фреймворка