



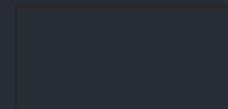
JavaScript теперь можно не учить?

Родион Мостовой

ex. Руководитель разработки в “Доставка от Гудвина”

Свободный разработчик, исследователь технологий, преподаватель

<https://twitter.com/RodionMostovoy>



Зачем нам это?

1. Это интересно
2. 11 из 67 докладов на .NET Conf 2021 было посвящено Blazor
3. Единый язык программирования ускоряет разработку

Razor Pages

- Первая версия появилась в 2010 г.
- Шаблонный движок, который позволяет верстать страницы
- Сейчас входит в ASP.NET Core
- Сервер каждый раз рендерит HTML страницу и возвращает результат браузеру

Razor Pages: Недостатки

- UI создавать неудобно, поэтому используют js фреймворки + ASP.NET Web API
 - Angular
 - React
 - Vue
- Может работать только в связке с сервером
- **Клиентская логика на JavaScript**

Что не так в JS?

- Динамическая типизация
 - Менее безопасна, чем статическая
 - Множество ошибок обнаруживается только в runtime
- Синтаксический, лексический анализ
 - Снижает производительность
- Доступен только дробный тип Number, либо тяжелый BigInt
 - Снижает производительность

WTF, javascript?

```
[] == ![];    // -> true
```

```
NaN === NaN;  // -> false
```

```
Math.min() > Math.max() // -> true
```

```
parseInt("freak"); // -> NaN
```

```
parseInt("freak", 16); // -> 15
```

```
"b" + "a" + +"a" + "a"; // -> 'baNaNa'
```

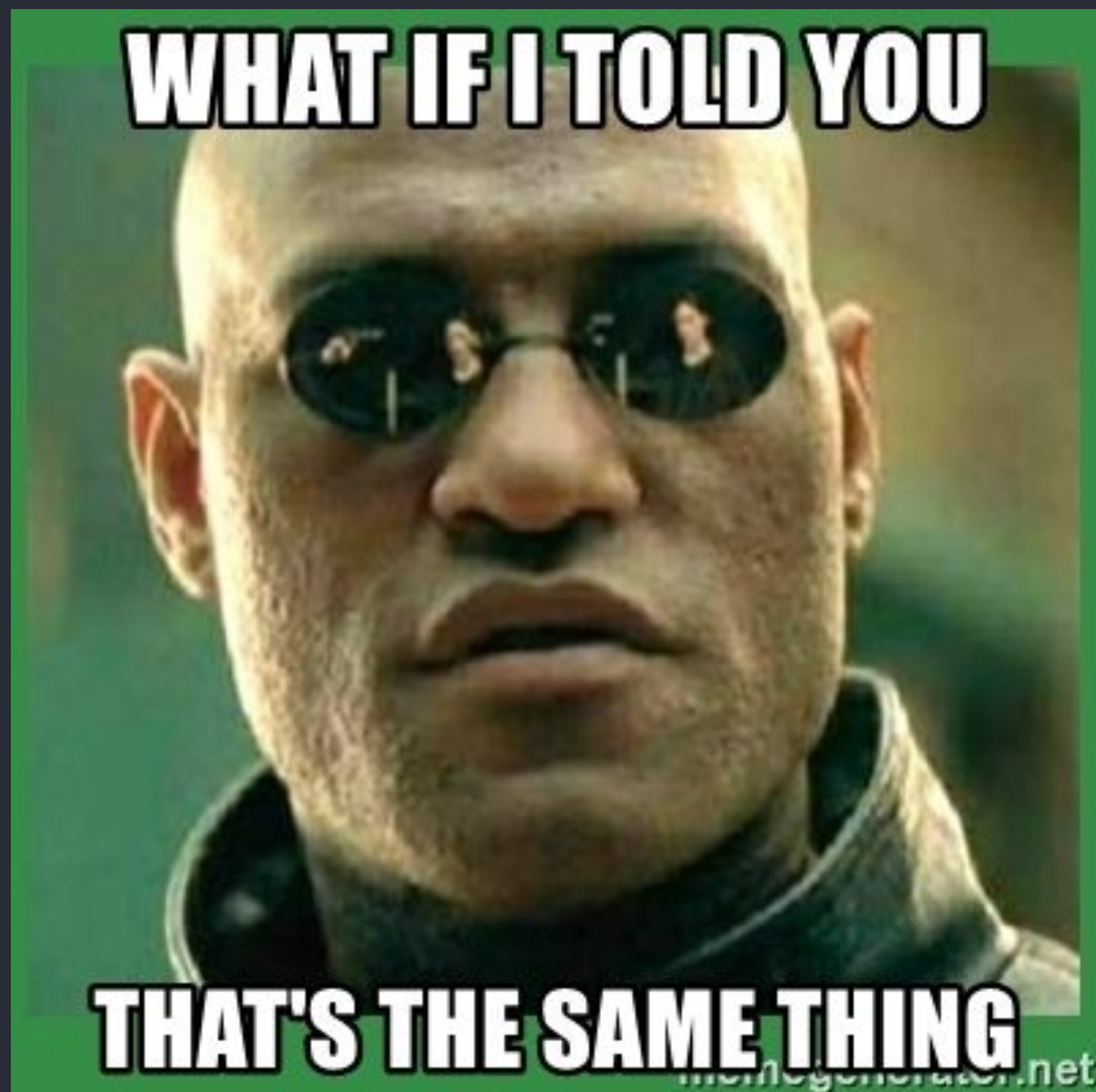
```
[null] == '' // -> true
```

```
[1, 2, 3] + [4, 5, 6]; // -> '1,2,34,5,6'
```



TypeScript!

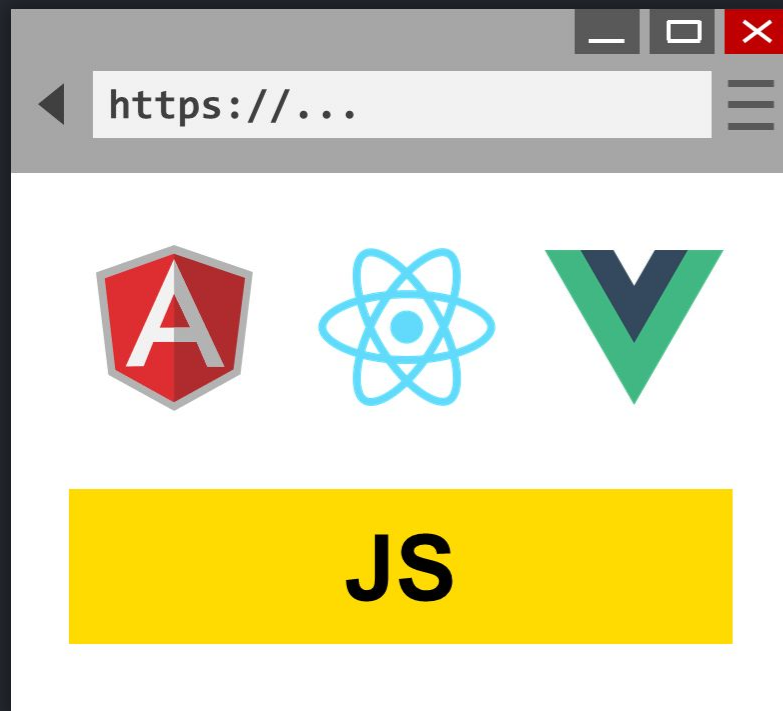
TypeScript!



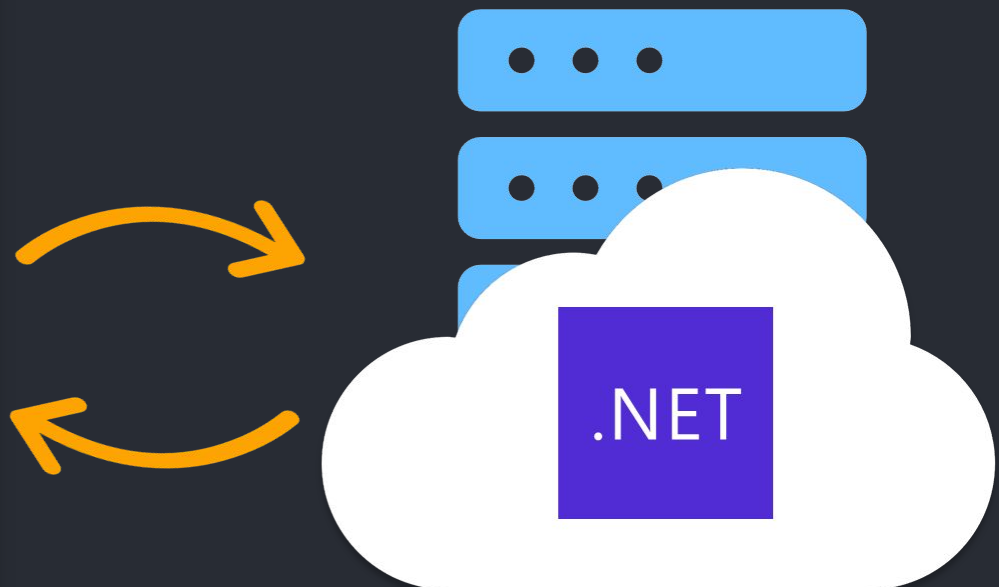
Blazor

- Появился в 2018 г.
- Клиентскую логику теперь можно писать на C#
- Позволяет создавать переиспользуемые веб компоненты
- Возможен вызов функций из JavaScript
- Является частью ASP.NET Core

JS – клиент

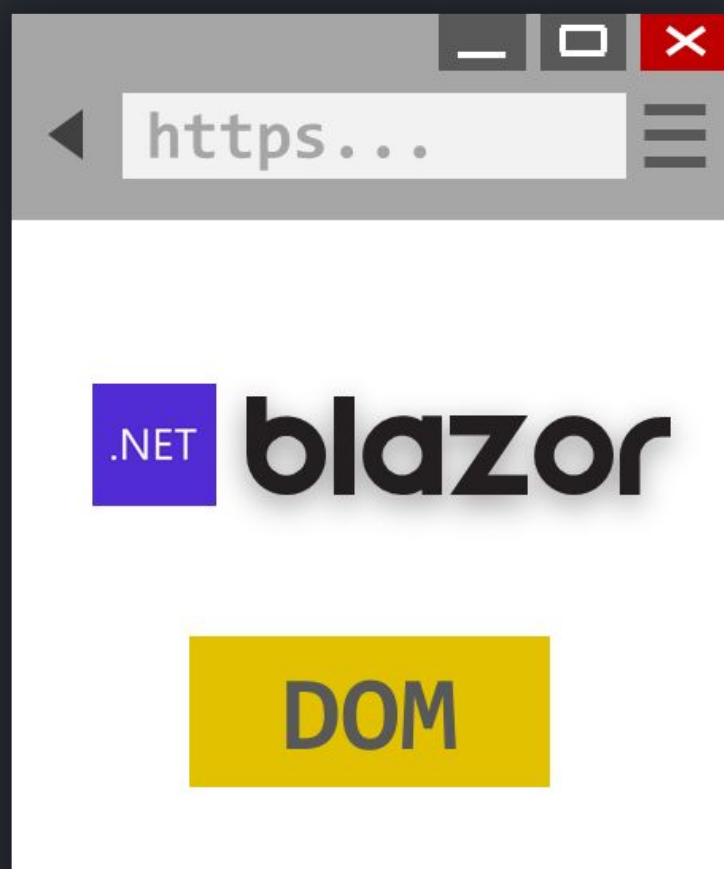


C# – сервер
(ASP.NET)

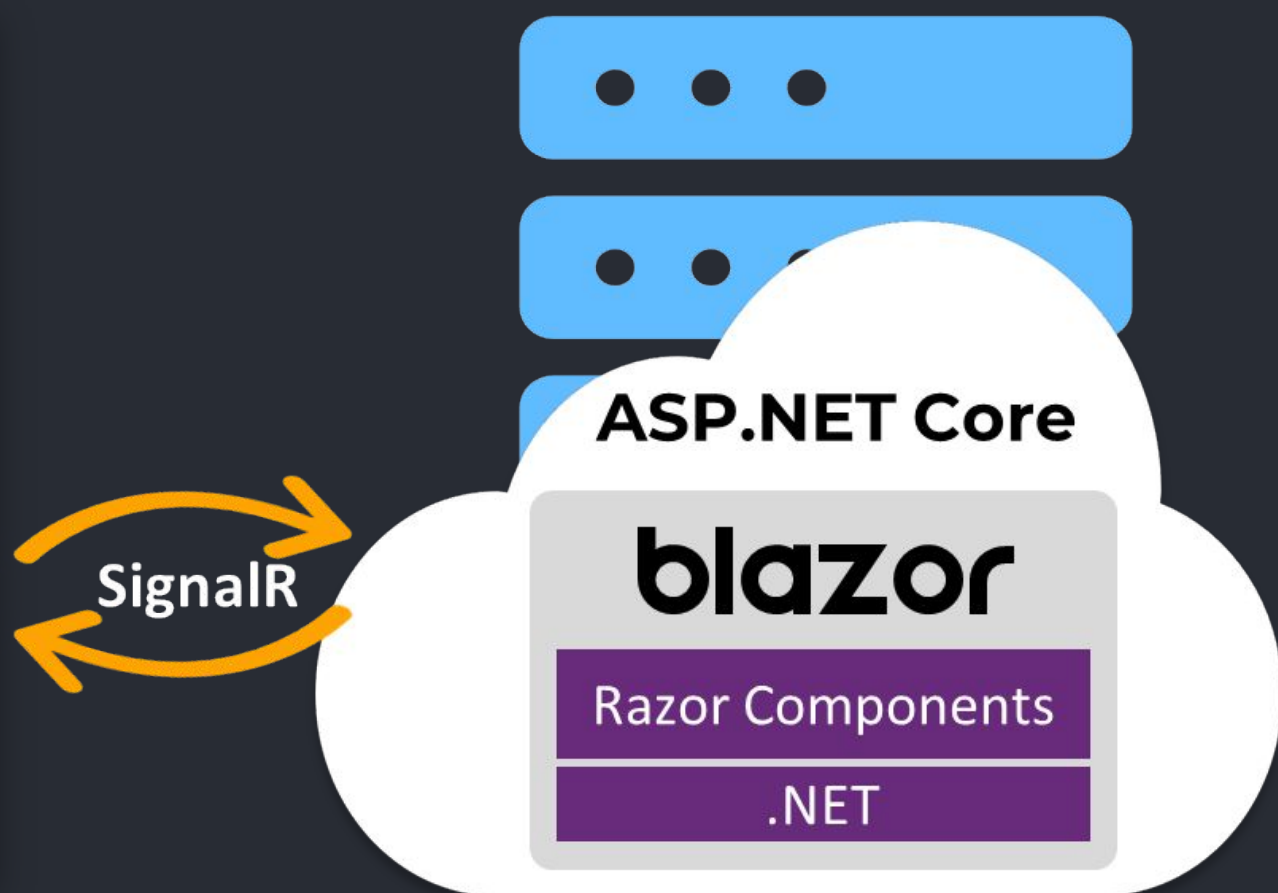


Blazor Server App

Blazor –
клиент (C#)



C# – сервер
(ASP.NET)



SignalR

- Дает возможность серверу мгновенно отправить данные подключенным клиентам, а не ждать, пока эти данные они запросят сами
- Общение работает в обе стороны
- Работает через WebSockets, либо HTTP
- Наиболее популярные примеры применения:
 - пуш сообщения
 - чаты
 - другие real-time приложения

Blazor Server: Преимущества

- ✓ Доступ к возможностям сервера
 - К базе, шифрованию и т. д.
 - Ускорение процесса разработки
- ✓ Real-time обновления
- ✓ Логика на сервере – это безопаснее

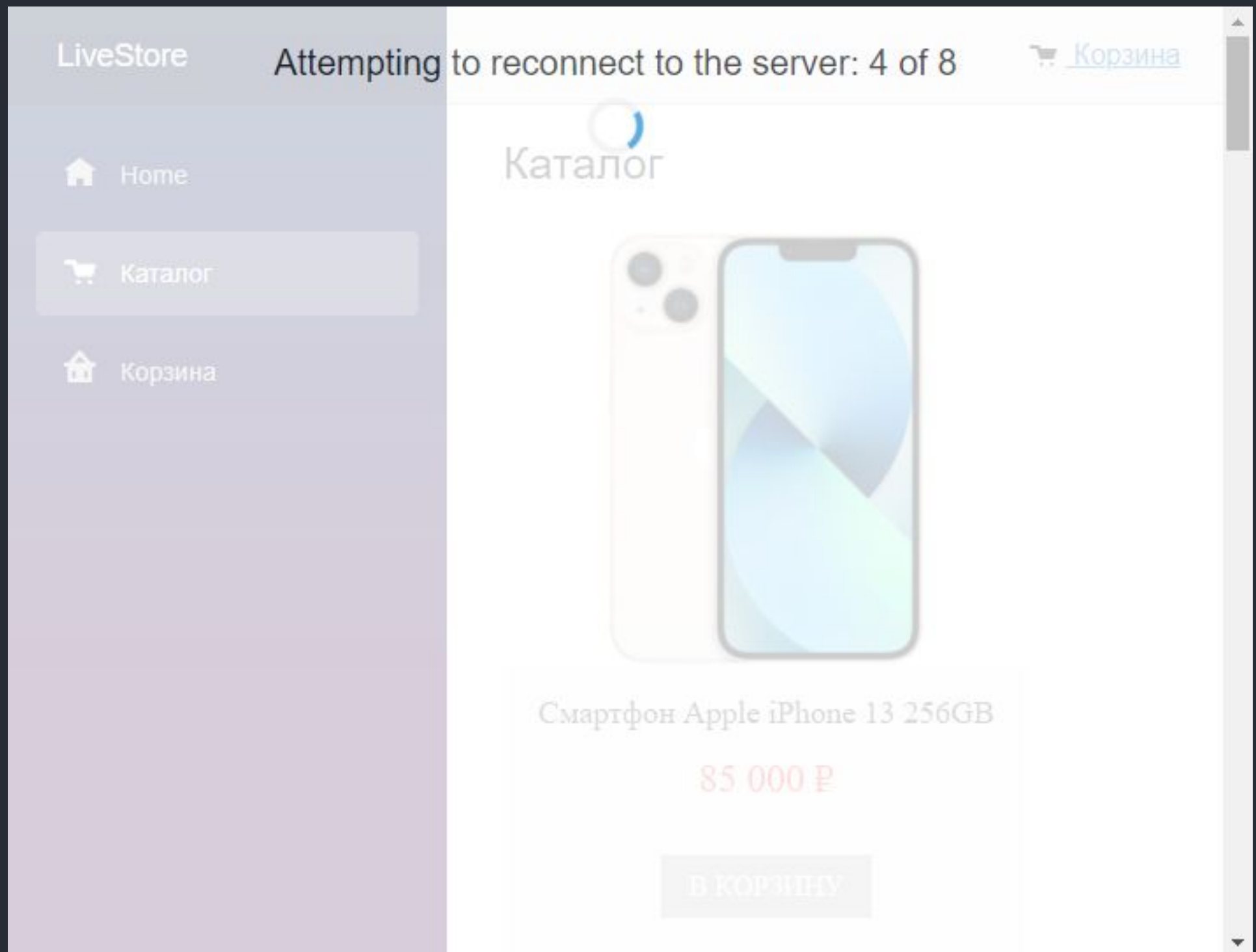
Blazor Server: Недостатки

- ✗ Нужен сервер с рантаймом .NET
- ✗ Требуется стабильное подключение
- ✗ Возможны подвисания UI
- ✗ Создает высокую нагрузку на сервер
- ✗ Не поддерживает автономную работу

Blazor Server: Применение

- Сайты с обновлением данных в реальном времени
- Админки
- MVP (минимально ценный продукт)
- Функциональные сайты с невысокой нагрузкой

При нестабильном соединении



А если мы хотим, чтобы
приложение могло работать
оффлайн?

Blazor WebAssembly!

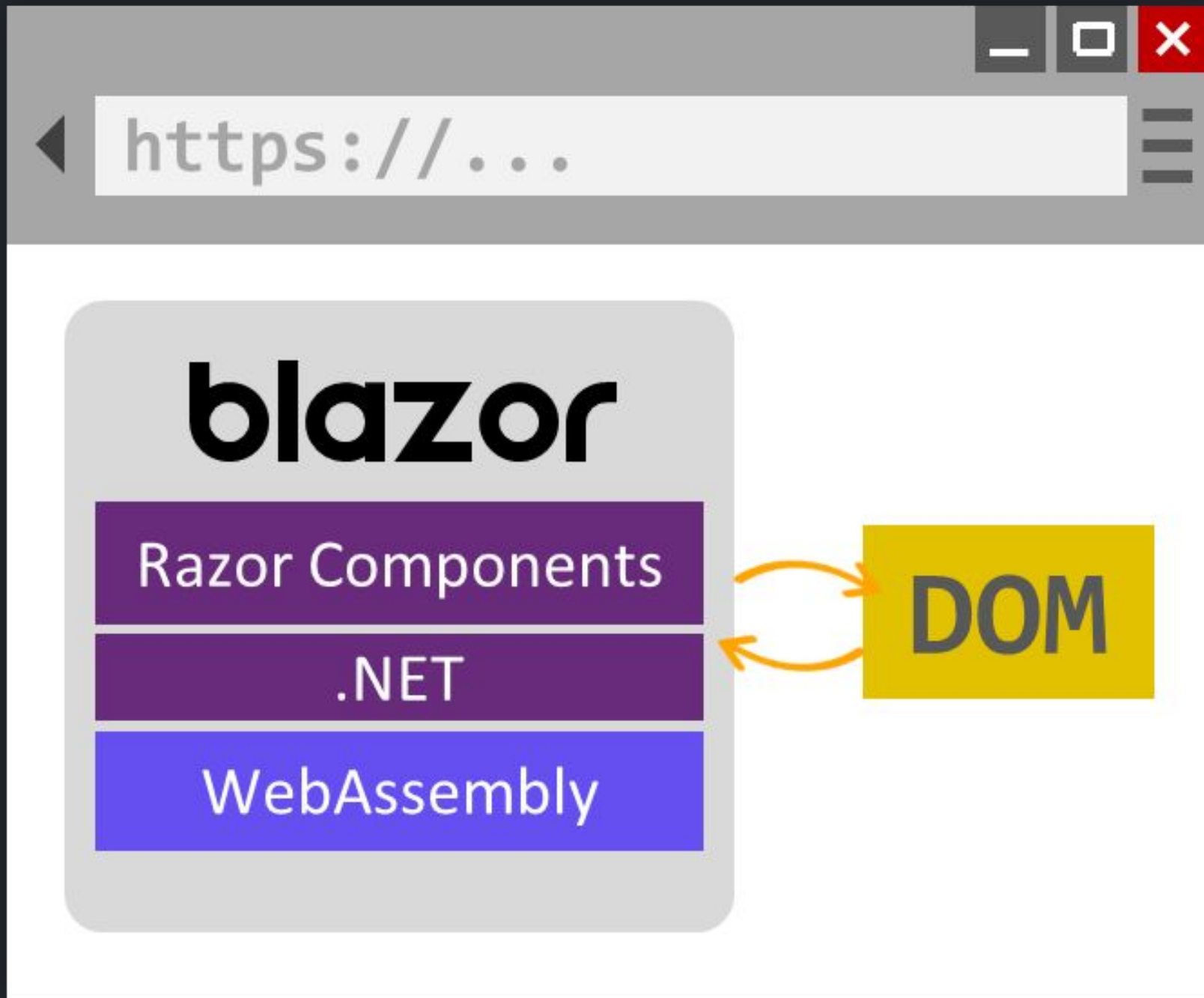
WebAssembly (WASM)

- WebAssembly (или WASM) это:
 - бинарный формат, запускаемый в браузере
 - виртуальная машина
 - результат компиляции с языка высокого уровня
- Поддержка в ЯП: C++, Rust, C# (.NET) и т. д.
- Выполняется значительно быстрее JavaScript
- Поддержка многотопоточности (кроме Safari)

Google Earth



Blazor WebAssembly



Blazor WebAssembly: Преимущества

- Может работать полностью оффлайн (PWA)
- Сервер не обязателен, т. е. можно хостить где угодно
- Тяжелые вычисления можно делать на стороне клиента
- Поддерживает интероп с модулями на других ЯП (например, с Rust)

Blazor WebAssembly: Недостатки

- ✗ Загружает много данных при первом запуске (минимум 1 мб.)
- ✗ Для общения с бекендом требуется использовать HttpClient или gRPC
- ✗ Есть проблемы с SEO
- ✗ Работает в 1 поток, производительность может уступать десктопной
- ✗ Не получится вызывать функции ОС

Blazor WebAssembly: AOT

Максимальная производительности достигается с AOT (Ahead Of Time):

- Прирост до 20-ти раз
- Но размер приложения увеличивается

<https://blog.elmah.io/ahead-of-time-compilation-for-blazor-wasm/>
<https://www.youtube.com/watch?v=X6FnfzUfrew>

Blazor WebAssembly: Применение

- Графические, 3D редакторы
- Кодировщики аудио, видео
- Почтовый клиент
- Игры, игровые движки, VR/AR
- Другие задачи, нагружающие ЦП

А если мы хотим?

- Избежать долгой первой загрузки
- И совместить онлайн и оффлайн режимы работы
- Как в Google Slides :)

Blazor Server + WebAssembly!

Blazor Server + WebAssembly

- Связать их вручную
 - Статья “Blazor: Server и WebAssembly одновременно в одном приложении”:
<https://habr.com/ru/post/546414/>
- Использовать библиотеку [Stl.Fusion](#)
 - Доклад о ней: <https://youtu.be/rrNqwOTgQKI>

А если мы хотим работать с
файлами в ОС?

MAUI Blazor App (Preview)

- Выглядит как нативное приложение
- Кроссплатформенный
 - Windows
 - Mac
 - Android
 - iOS
- Априори работает оффлайн
- Многопоточный
- Может вызывать функции ОС
 - Работать с файлами
 - Запускать процессы
 - Обращаться к API ОС (например, WinAPI)

Mobile Blazor Bindings (MBB)

Он же Blazor Hybrid.

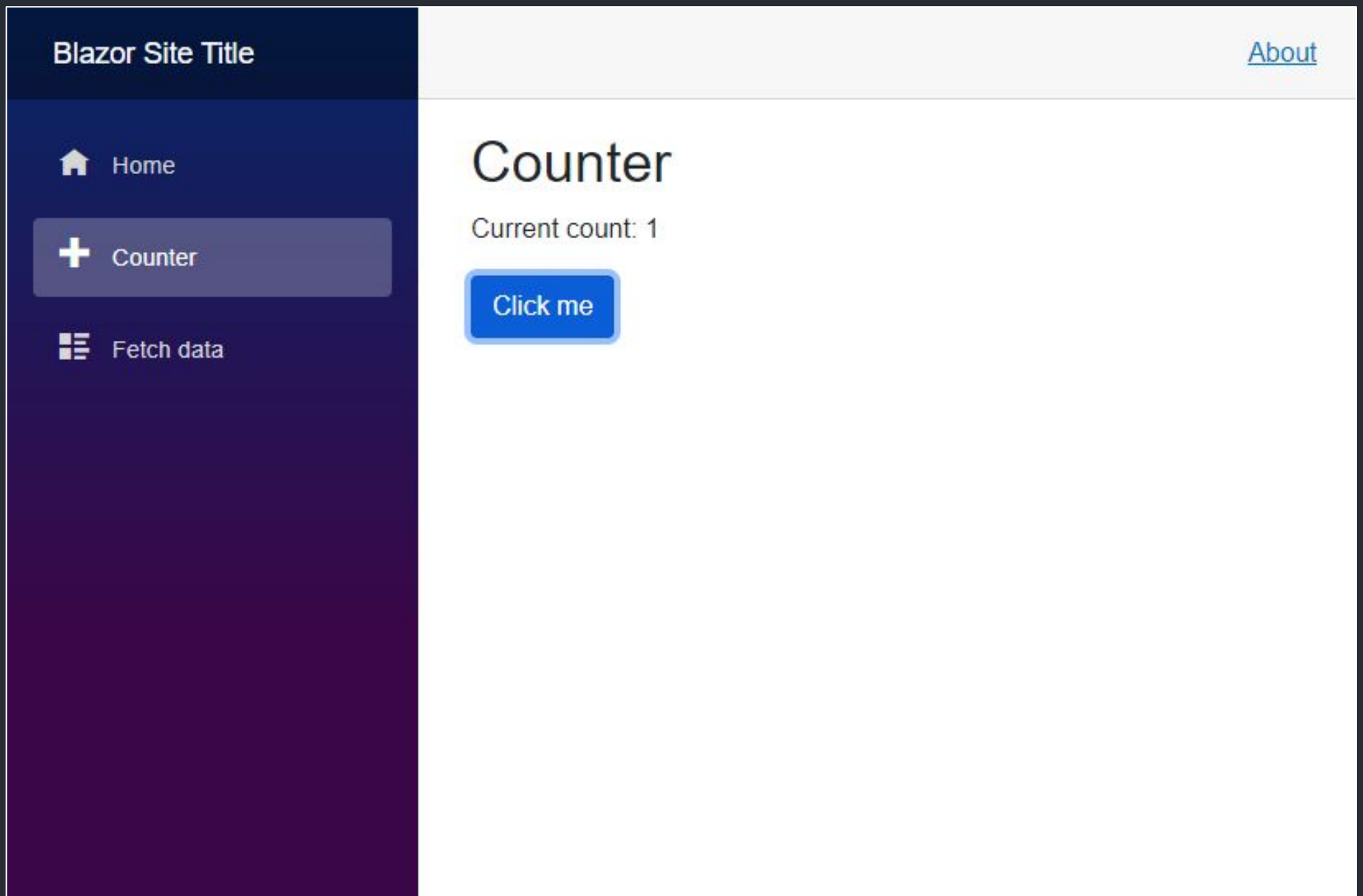
Принципиально иной подход, в основе уже не веб. Компоненты мапятся в контроллы WPF/Xamarin.

```
<StackLayout>
    <Label FontSize="30">You pressed @count times </Label>
    <Button Text="+1" OnClick="@HandleClick" />
</StackLayout>
```

```
@code {
    int count;

    void HandleClick()
    {
        count++;
    }
}
```


Пример страницы



Структура Blazor компонента

```
@page "/counter"

<PageTitle>Counter</PageTitle>

<h1>Counter</h1>

<p role="status">Current count: @currentCount</p>

<button @onclick="IncrementCount">Click me</button>
```

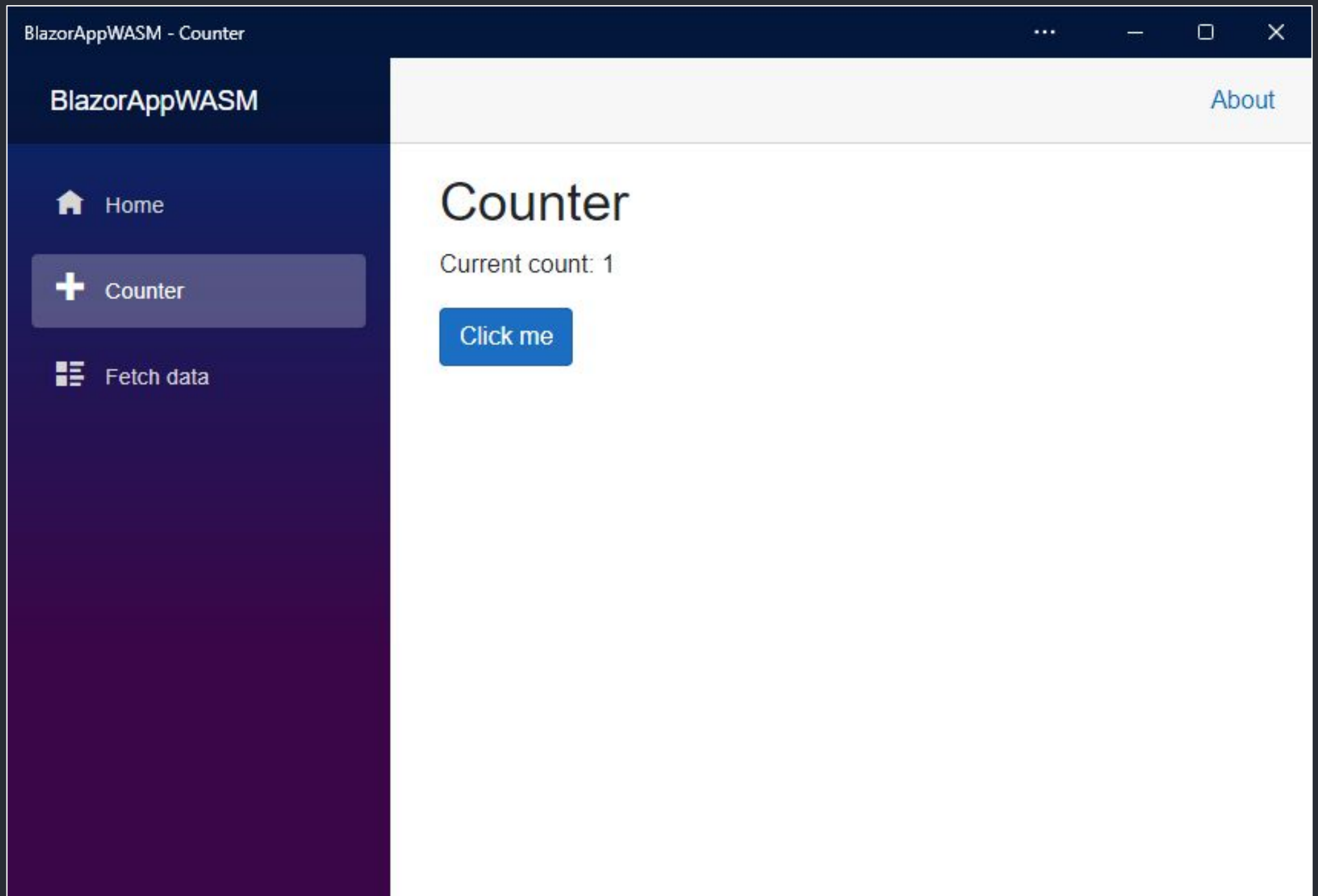
HTML код страницы,
включая Blazor компоненты

```
@code {
    private int currentCount = 0;

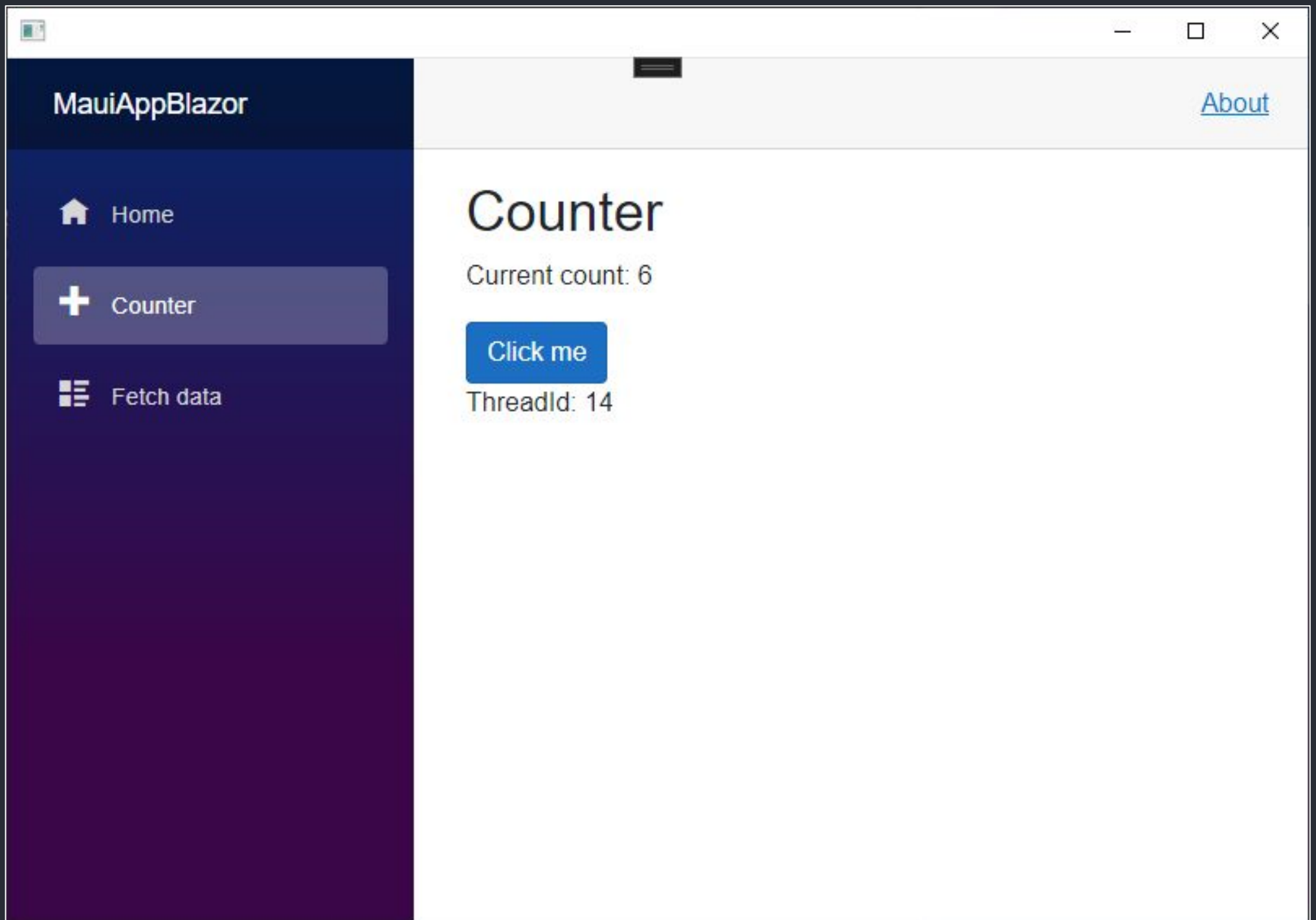
    private void IncrementCount()
    {
        currentCount++;
    }
}
```

Логика страницы на языке C#!

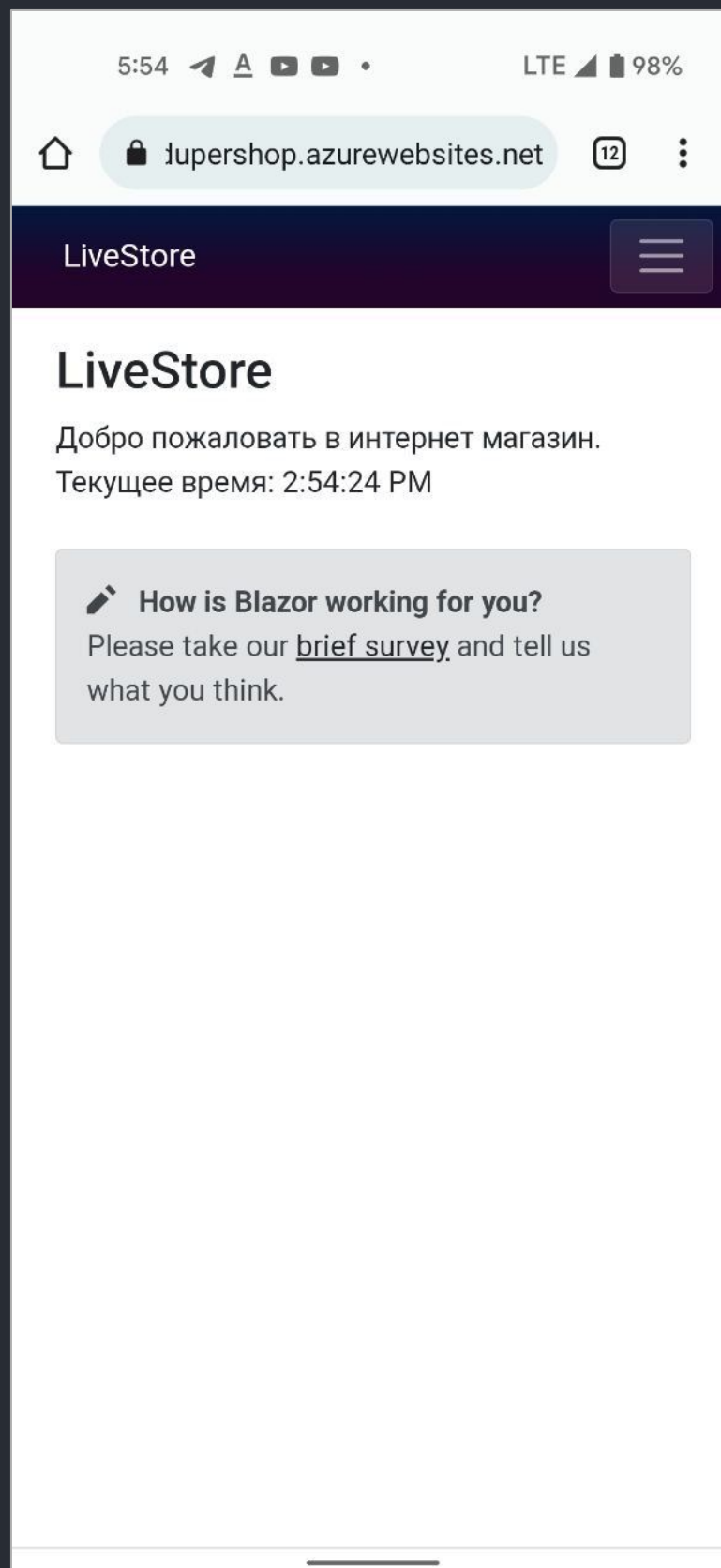
Blazor WebAssembly (PWA)



MAUI Blazor



Blazor на мобильном



Интерактив

rodion-m.ru

“Живая корзина”

Real-time корзина

1. Откройте корзину в другом окне браузера или даже на другом устройстве
2. Добавьте товары в корзину из каталога на устройстве 1
3. Корзина должна обновляться в реальном времени

Как мы этого добились?

```
@code {  
...  
    protected override void OnInitialized()  
    {  
        OBasket.OnChanged += RefreshBasket;  
    }  
  
    private void RefreshBasket()  
    {  
        _products = OBasket.GetBasketProducts();  
        InvokeAsync(StateHasChanged);  
    }  
...  
}
```


Примеры сайтов на Blazor

Игры на Blazor

<https://blazorgames.net/>

<https://boardgames.alex yakunin.com/>

Примеры других сайтов

<https://trends.builtwith.com/websitelist/Blazor>

Где учиться?

- Курс от Microsoft:
<https://docs.microsoft.com/ru-ru/learn/paths/build-web-apps-with-blazor/>
- Сайт о том, как работает Blazor:
<https://blazor-university.com/>
- Русскоязычные чаты по Blazor в Telegram:
 - [@BlazorRuChat](#)
 - [@BlazorNetRuChat](#)
- Международный чат по Blazor в Gitter:
 - <https://gitter.im/aspnet/Blazor>

Подведем итоги

- Так можно отказаться от JavaScript или нет?
 - Зависит. Но в целом, можно.
- Blazor Server
 - Не взлетит без сервера
 - Годится для админок, real-time сайтов и MVP
- Blazor WebAssembly
 - Может работает оффлайн
 - Хорош для игр и при высокой нагрузки на клиент
- MAUI Blazor App
 - Если ваше приложение должно работать как нативное
 - Если нужна производительность и доступ к функциям ОС

Спасибо за просмотр!

Спасибо Марку Шевченко за помощь в
подготовке и Александру Якунину
(разработчик Stl.Fusion) за консультации.