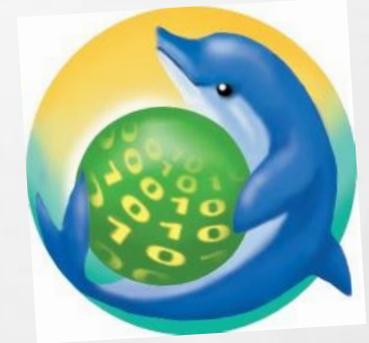


BLAZORと踊ろう！



WRITE BY 松井 敏

.MEETUP APP OSAKA@3

2018-12-22



MYSELF

- 松井 敏 A.K.A 森理 麟 **BELONG CODEER**
- **MY COMMUNITY : C#読書会**



前提

- 約**1.5**年前からホリデープロジェクトとしてコツコツと**WEB**サービスを作っている
- 去年、**MEETUP APP OSAKA@2**で「**AZURE WEBJOBS**四方山話」を話しました
- **WEBJOBS**で自動スクレイピングをしている話です。もちろん今日も自動スクレイピングは続いている。

[HTTP://MANGA-YONDA.COM](http://manga-yonda.com)

- 「マンガを読んだ！！」はクリックするだけで簡単に読んだマンガを記録できるサービス
- 作者やシリーズ全部でまとめて1回で登録も簡単！
- マンガは集英社、講談社、小学館などから**8**万冊以上！毎日自動更新で、日々増加中！！

前提

- 去年の時点で**A**版。その後最低限必要と置いていた機能を入れた**B**版までは公開済み。
- その時点から技術見直しを行う
- **ASP.NET MVC**から**VUE.JS**。

SINGLE PAGE APPLICATION

- **SPA**。簡単に言うと**HTML**を**JS**でバンバン動的に書き換えて作られた**APPLICATION**。
- どれだけ動きがあっても**HTML1**枚なので**SINGLE PAGE**。
- が、僕はどうにもこうにも**JS**つらくなった。

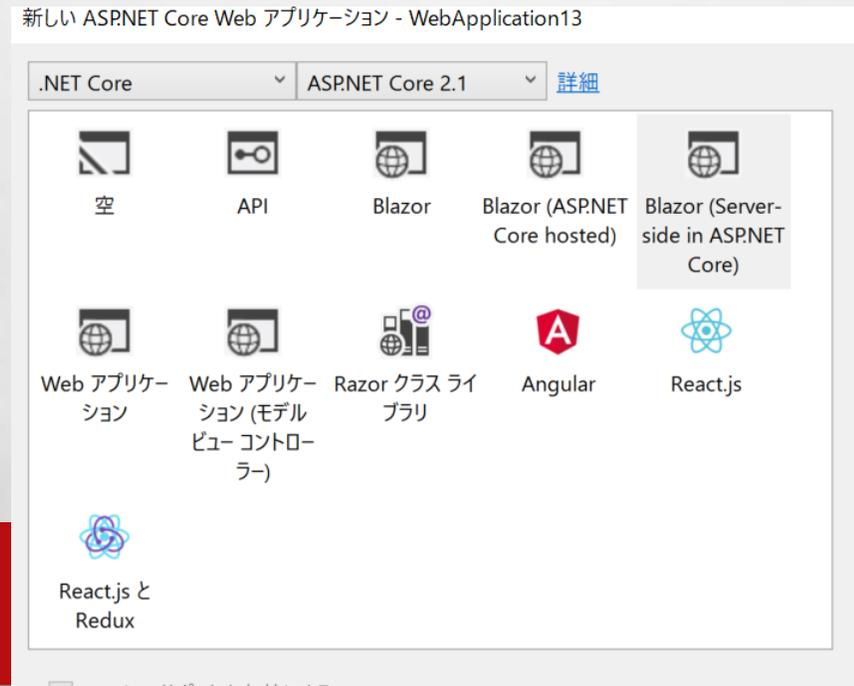
SINGLE PAGE APPLICATION

- **そこでBLAZOR!**
- **BLAZOR**は**WEB ASSEMBLY**を使った**.NET**で動く**SPA**フレームワーク
- **JS**をほぼほぼ使わず**C#**で動的ページが作れます!

BLAZOR.

BLAXOR

- **BLAZOR(ASP.NET CORE HOSTED)** を選びます



SERVER, CLIENT, SHARED

- **SERVER, CLIENT, SHARED**の3つのプロジェクトが作られる。
- **SERVER**は**ASP.NET WEB API**。**CLIENT**がクライアントで**BLAZOR**本体。**SHARED**が共通。

RAZOR

- **BLAZOR**は**RAZOR**がベース。
- **RAZOR**はビューエンジン。**C#**言語用の**CSHTML**で**HTML**に**@**で直接**C#**を書ける

BLAZOR

- **BLAZOR**は**FUNCTIONS**の中でプロパティやメソッドが定義出来る。それを**CSHTML**から呼ぶことが出来る。

SOURCE

```
Counter.cshtml  + X
1  @page "/counter"
2
3  <h1>Counter</h1>
4
5  <p>Current count: @currentCount</p>
6
7  <button class="btn btn-primary" onclick="@IncrementCount">Click me</button>
8
9  @functions {
10     int currentCount = 0;
11
12     void IncrementCount()
13     {
14         currentCount++;
15     }
16 }
```

Routing

cshtml

Functions (c#)

BIND

- **CHECKBOX**などは**BIND**で双方向バインディング

```
<ul>
  @foreach (var todo in todos)
  {
    <li>
      <input type="checkbox" bind="@todo.IsDone" />
      <input bind="@todo.Title" />
    </li>
  }
</ul>
```

EVENT

```
@functions {  
    protected override void OnInit()  
    {  
        UriHelper.OnLocationChanged += async (_, __) =>  
        {  
            Console.WriteLine("OnLocationChanged");  
            await StateChangeAsync();  
        };  
    }  
  
    protected override async Task OnInitAsync()  
    {  
        AllList = new List<MangaNavi> { new MangaNavi() { Name = "すべ" };  
        PublisherList = await Http.GetJsonAsync<IEnumerable<MangaNavi>>();  
        DateList = await GetDateListAsync();  
        AuthorList = await Http.GetJsonAsync<IEnumerable<MangaNavi>>($"  
        SeriesList = await Http.GetJsonAsync<IEnumerable<MangaNavi>>($"  
        MangaList = await SetPagenationAndGetMangaListAsync();  
    }  
}
```

COMPONENT

- コンポーネントですよ。コンポーネント。これが最強です。

HTTPS://MANGAYONDA.AZUREWEBSITES.NET/

- こんな感じです

DEMERIT

- まだ**EXPERIMENTAL**
- **ASSEMBLY**と言っても特に速くもない。起動はむしろ遅いぐらい？
- サンプルも少ないし、ほぼ英語。ライブラリもあんまない

DEMERIT

- 例えば**PAGINATION**もないので作った
- **ASP.NET IDENTITY**は画面が書けない。**BLAZOR**はタグヘルパー対応してないので、認証どうしよう。
- **COOKIE**もよくわかってない。**JS**かな。

DEMERIT

- 特に**IDE**のサポートがまだまだ感。
- 普通に**DEBUG**も出来ないし。
- コンパイルがよく応答不能になるし。←これが一番不便。
- OBJ**と**BIN**の中にクリーンしても残るゴミができたり。
- リネームが失敗して**CSHTML**ではもうヤラナイ。。

MERIT

- **C#**で書ける！

まとめ

- **JS**はつらかった。**VS**の世界観とは違うものを無理に乗っけただけで使っている感じ。
- **CSHTML**でもそういうところはあるけれど、言語切り替えなくて書けるのが本当に楽。**C#**で書けるのが楽しい！
- **C#**大統一理論でより良い**BLAZOR**ライフを踊ろう！

最後に

- 踊ろうと書いたのは、正直踊らされる可能性はある。それなりに。
- でも、**SPA**のノウハウは結構そのまま**JS**に使える。もし**BLAZOR**が駄目でも他の**WEBASSEMBLY**フレームワークは出ると思う
- 踊る阿呆にみる阿呆同じ阿呆なら踊らにや損損かなーと