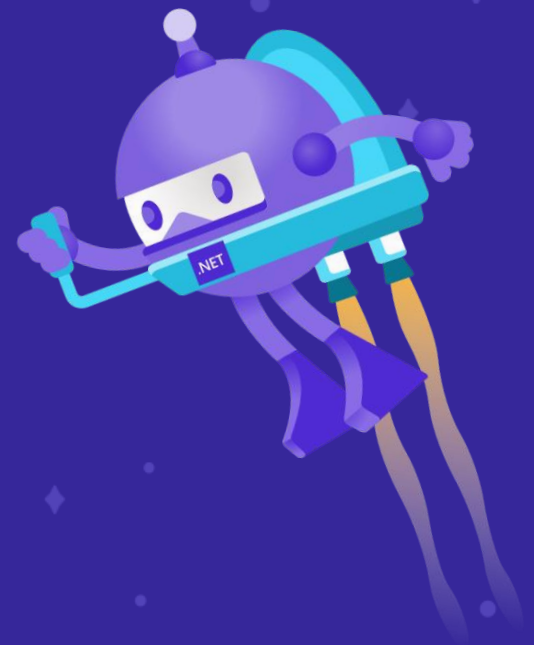


.NET Conf 2022 x Seoul

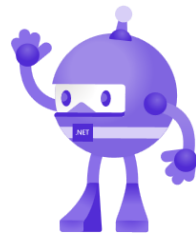
풀 스택과 **사랑**에 빠질 준비, 되셨나요?

Blazor WebAssembly + Webpack + TypeScript + Sass

김예건



시작하기에 앞서...



.NET CLI



webpack



VISUAL STUDIO 2022



TypeScript

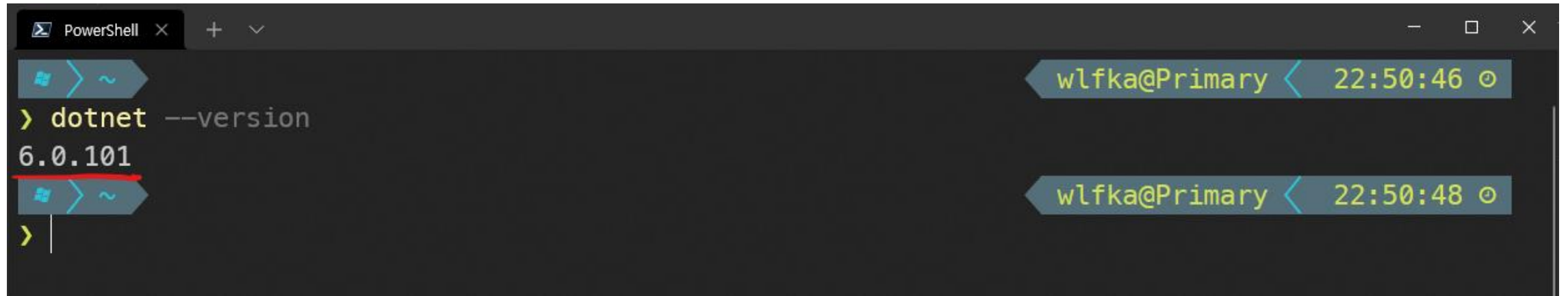


Blazor



<https://ibocon.tistory.com/271>

.NET 6




A screenshot of a Windows PowerShell terminal window. The window has a dark theme and a title bar with the text "PowerShell" and standard window controls. The terminal shows a command prompt with a blue prompt character ">". The command "dotnet --version" has been entered, and the output "6.0.101" is displayed below it. The version number "6.0.101" is underlined with a red line. To the right of the terminal window, there are two chat messages from "wlfka@Primary" with timestamps "22:50:46" and "22:50:48".

```
PowerShell x + v  
> dotnet --version  
6.0.101  
> |
```

wlfka@Primary < 22:50:46 ○
wlfka@Primary < 22:50:48 ○

Visual Studio 2022

 **Visual Studio Community 2022**
17.0.4
강력한 IDE, 학생, 오픈 소스 제공자 및 개인을 위해 무료로 제공
[릴리스 정보](#)

수정(M)





시작(L)

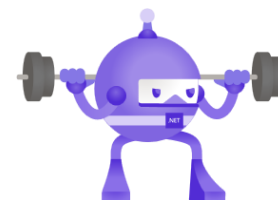
자세히 ▼

수정 — Visual Studio Community 2022 — 17.0.4

워크로드 개별 구성 요소 언어 팩 설치 위치

웹 및 클라우드 (4)

 ASP.NET 및 웹 개발 <input checked="" type="checkbox"/> Docker 지원이 포함된 ASP.NET Core, ASP.NET, HTML/JavaScript 및 컨테이너를 사용하여 웹 애플리케이션을 빌...	 Azure 개발 <input checked="" type="checkbox"/> .NET 및 .NET Framework를 사용하여 클라우드 앱을 개발하고 리소스를 만들기 위한 Azure SDK, 도구 및 프로젝트입...
 Python 개발 <input checked="" type="checkbox"/> Python에 대한 편집, 디버깅, 대화형 개발 및 소스 제어입니다.	 Node.js 개발 <input checked="" type="checkbox"/> 비동기 이벤트 구동 JavaScript 런타임인 Node.js를 사용하여 확장 가능한 네트워크 애플리케이션을 빌드합니다.



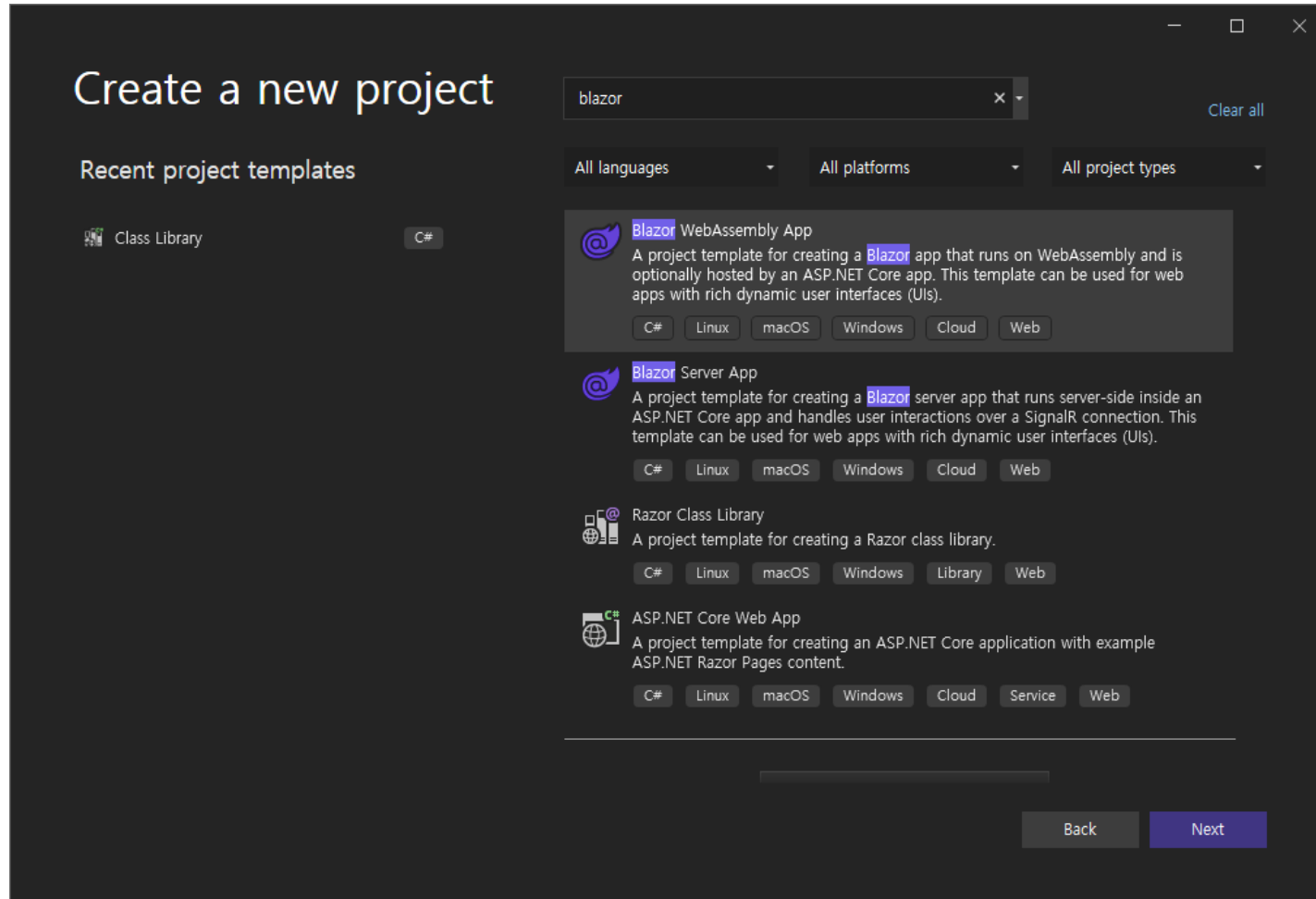
Blazor



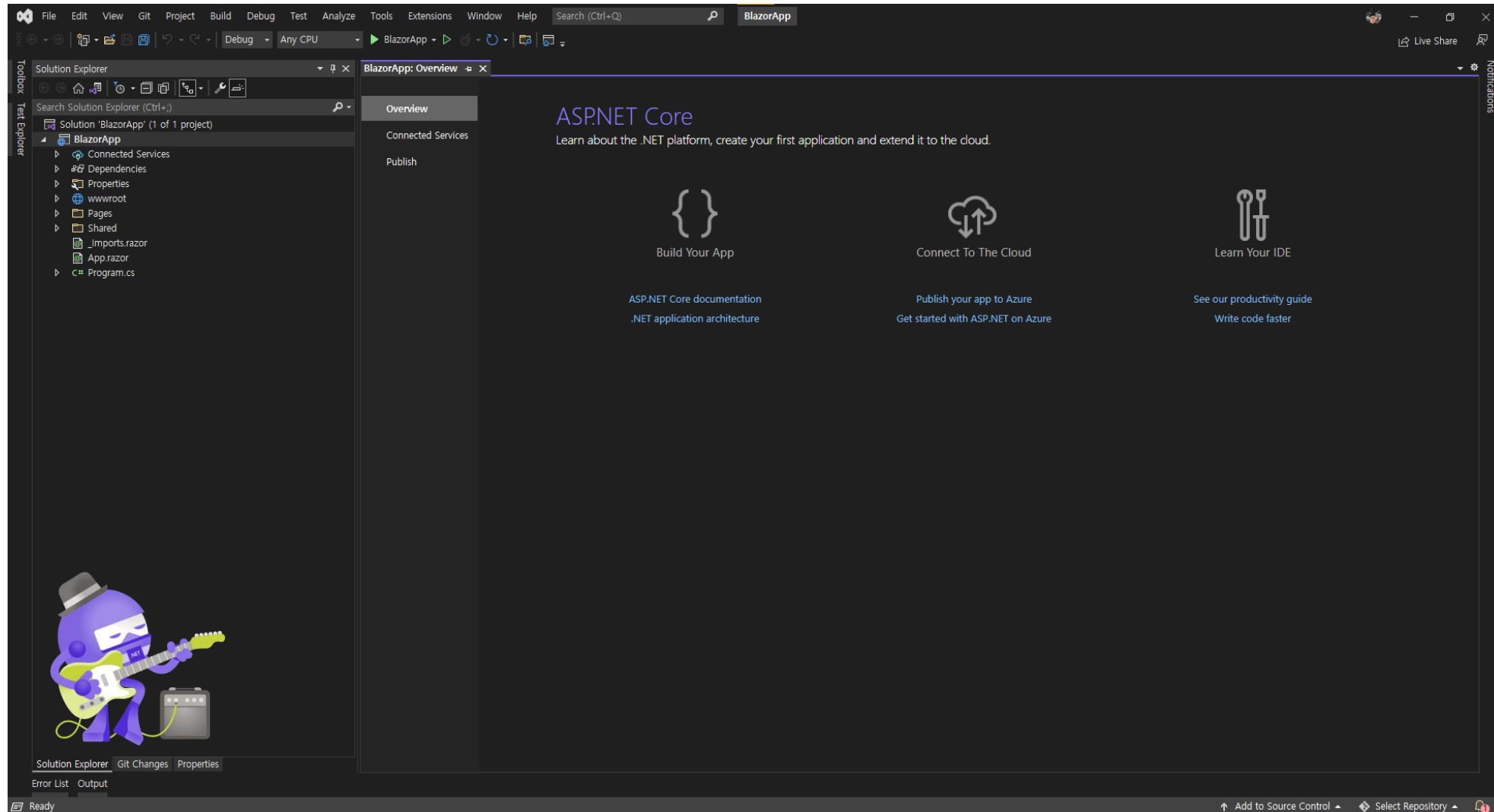
Blazor



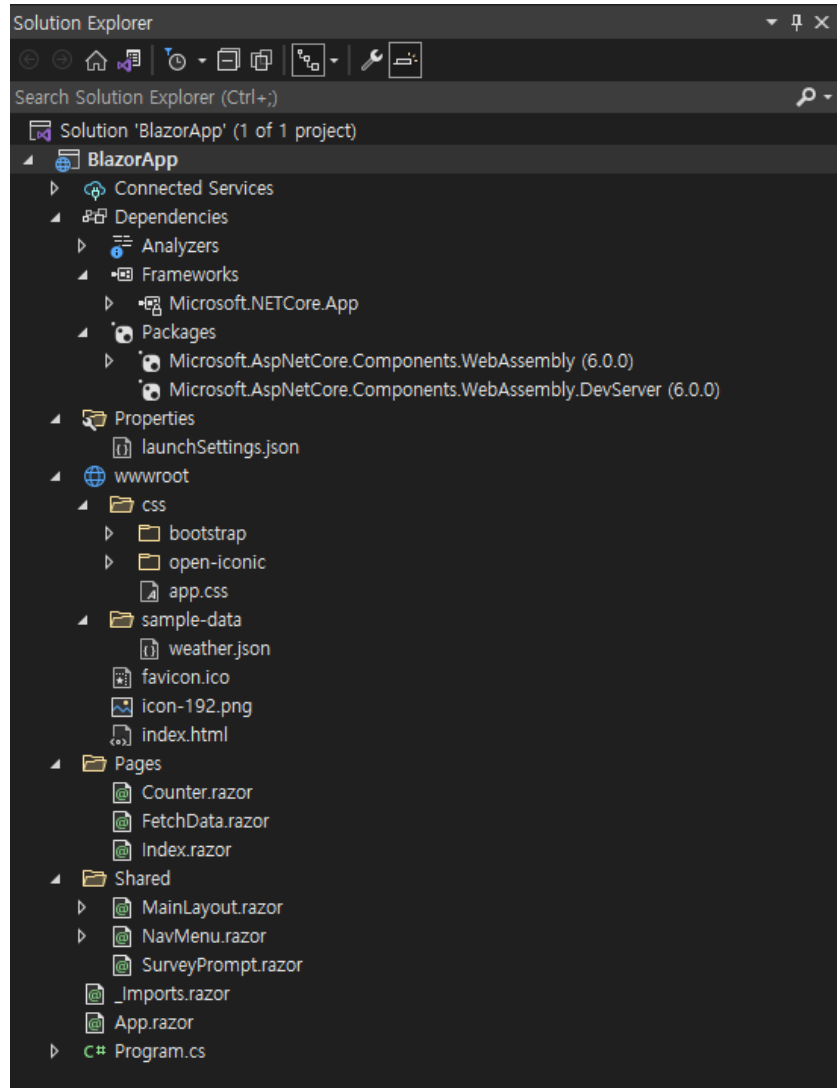
Blazor WebAssembly App



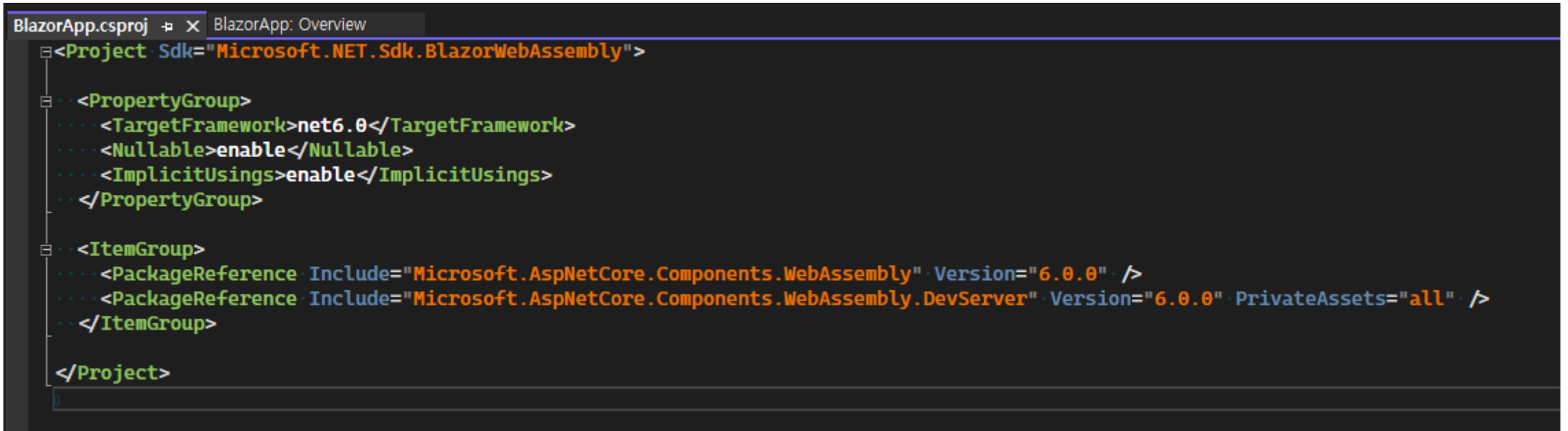
Blazor WebAssembly App



Blazor WebAssembly App



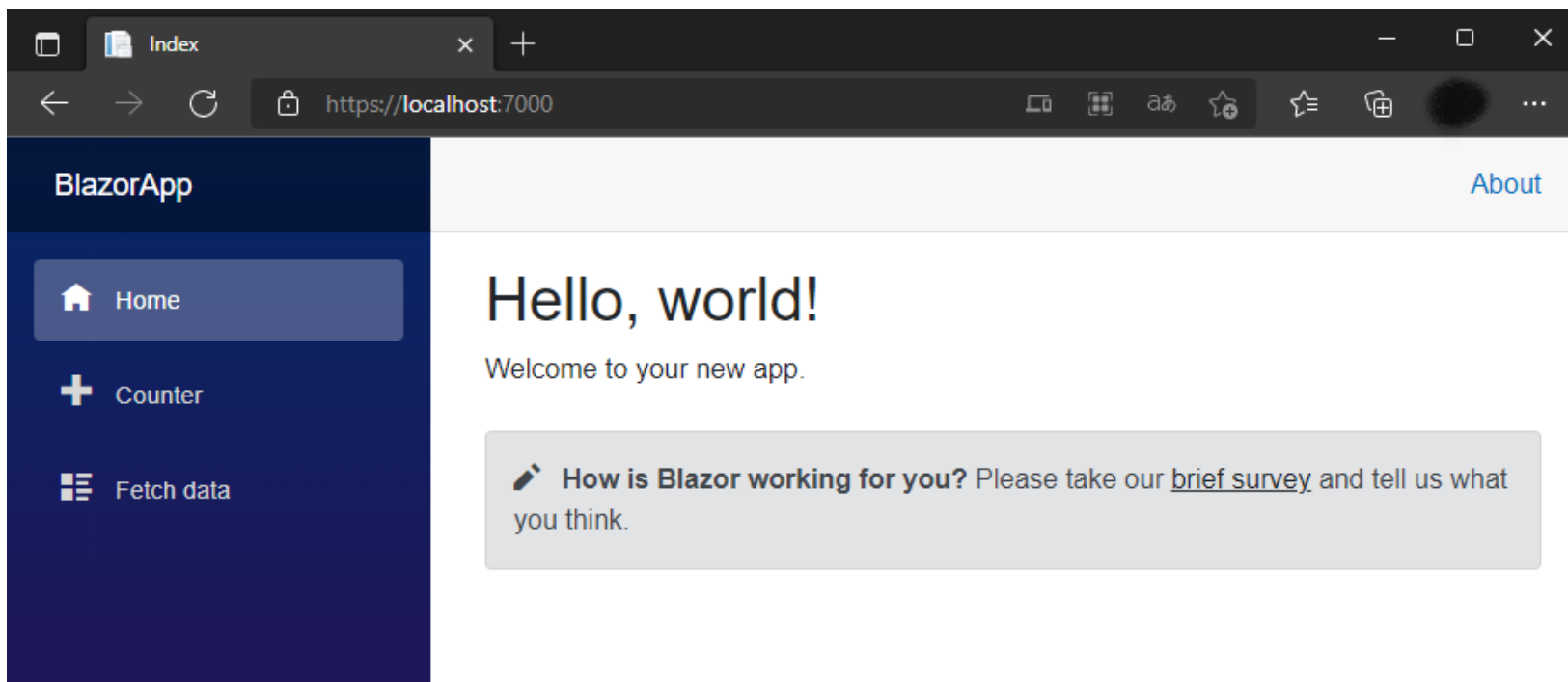
Blazor WebAssembly App



```
<?xml version='1.0' encoding='utf-8'>
<Project Sdk="Microsoft.NET.Sdk.BlazorWebAssembly">
  <PropertyGroup>
    <TargetFramework>net6.0</TargetFramework>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>
  </PropertyGroup>
  <ItemGroup>
    <PackageReference Include="Microsoft.AspNetCore.Components.WebAssembly" Version="6.0.0" />
    <PackageReference Include="Microsoft.AspNetCore.Components.WebAssembly.DevServer" Version="6.0.0" PrivateAssets="all" />
  </ItemGroup>
</Project>
```

Blazor WebAssembly App

```
C:\Program Files\dotnet\dotnet.exe
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7000
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: E:\Source\DotNetConf2022\BlazorApp
```

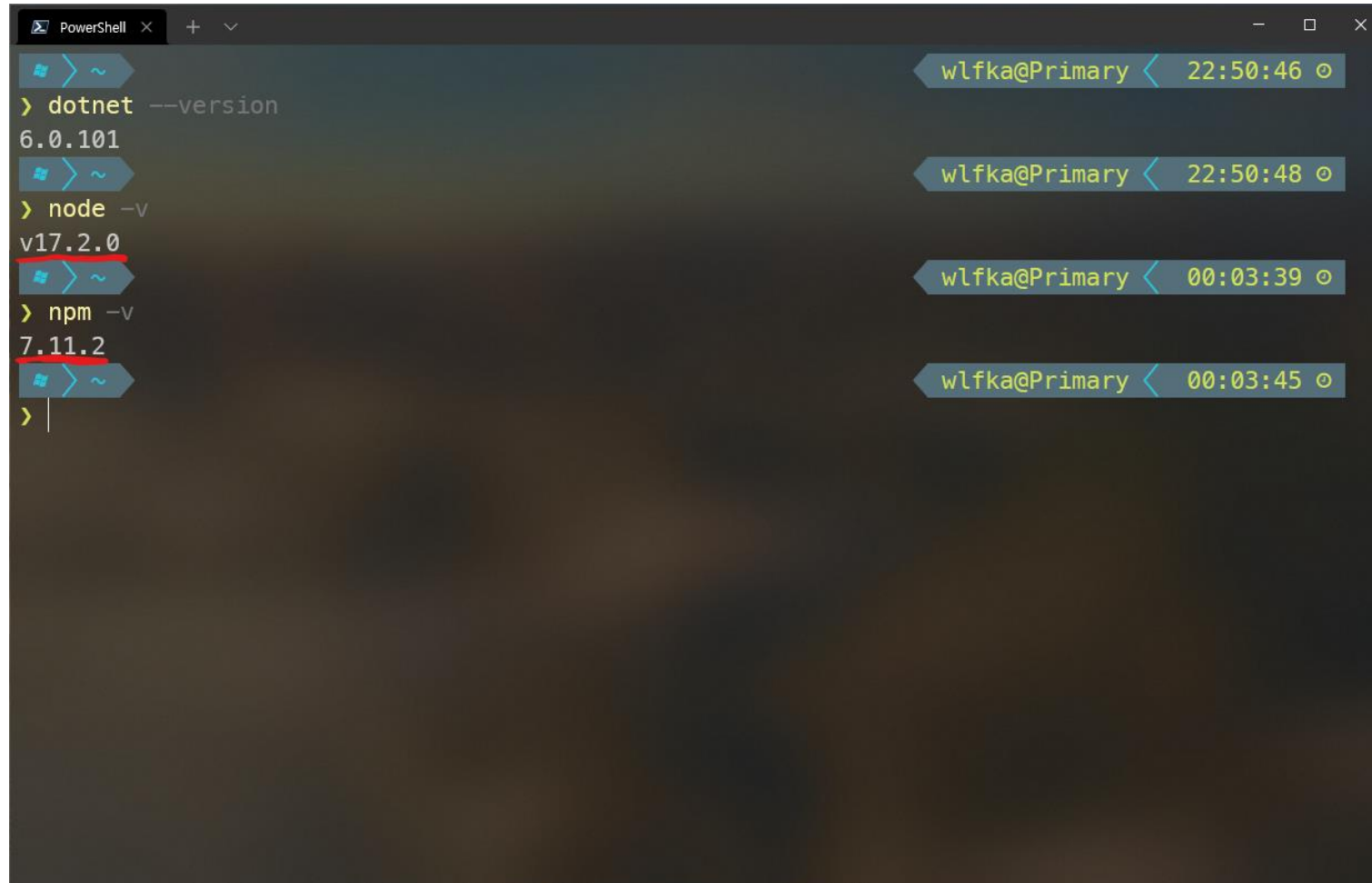


Webpack



webpack

Webpack



```
PowerShell x + v
> dotnet --version
6.0.101
> node -v
v17.2.0
> npm -v
7.11.2
> npm install webpack --save-dev
```

wlfka@Primary < 22:50:46 ○

wlfka@Primary < 22:50:48 ○

wlfka@Primary < 00:03:39 ○

wlfka@Primary < 00:03:45 ○

Webpack

```
PowerShell 7.2.1
Copyright (c) Microsoft Corporation.


https://aka.ms/powershell
Type 'help' to get help.

Loading personal and system profiles took 650ms.
> E:\Source\DotNetConf2022\BlazorApp > master
> ls

Directory: E:\Source\DotNetConf2022\BlazorApp

Mode                LastWriteTime         Length Name
----                -
d-----         12/16/2021 11:22 PM             bin
d-----         12/16/2021 11:28 PM             obj
d-----         12/16/2021 11:22 PM            Pages
d-----         12/16/2021 11:22 PM          Properties
d-----         12/16/2021 11:22 PM           Shared
d-----         12/16/2021 11:22 PM          wwwroot
-a-----         12/16/2021 11:22 PM           381 _Imports.razor
-a-----         12/16/2021 11:22 PM           477 App.razor
-a-----         12/16/2021 11:22 PM          488 BlazorApp.csproj
-a-----         12/16/2021 11:22 PM         1123 BlazorApp.sln
-a-----         12/16/2021 11:22 PM           434 Program.cs

> E:\Source\DotNetConf2022\BlazorApp > master
> |
```



Webpack

```
npm init
```

```
➤ > E:\Source\DotNetConf2022\BlazorApp > #master wlfka@Primary < 23:45:02
> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (blazorapp)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to E:\Source\DotNetConf2022\BlazorApp\package.json:

{
  "name": "blazorapp",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes)
➤ > E:\Source\DotNetConf2022\BlazorApp > #master wlfka@Primary < 23:49:31
> |
```

```
package.json  BlazorApp.csproj  BlazorApp: Overview
Schema: https://json.schemastore.org/package.json

1  {
2    "name": "blazorapp",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "",
10   "license": "ISC"
11  }
12
```

Webpack

```
npm install webpack webpack-cli webpack-dev-server --save-dev
```

```
> E:\Source\DotNetConf2022\BlazorApp > |master wlfka@Primary < 00:05:42
> npm install webpack webpack-cli webpack-dev-server --save-dev
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParams
API instead.

added 326 packages, and audited 327 packages in 19s

42 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
> E:\Source\DotNetConf2022\BlazorApp > |master
> |
```

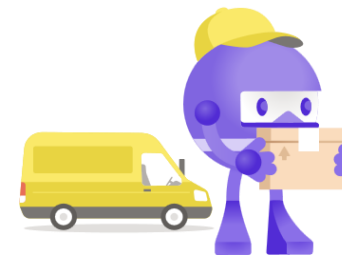
```
package.json x BlazorApp.csproj BlazorApp: Overview
Schema: https://json.schemastore.org/package.json

1  {
2    "name": "blazorapp",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "",
10   "license": "ISC",
11   "devDependencies": {
12     "webpack": "^5.65.0",
13     "webpack-cli": "^4.9.1",
14     "webpack-dev-server": "^4.6.0"
15   }
16 }
17
```


Webpack

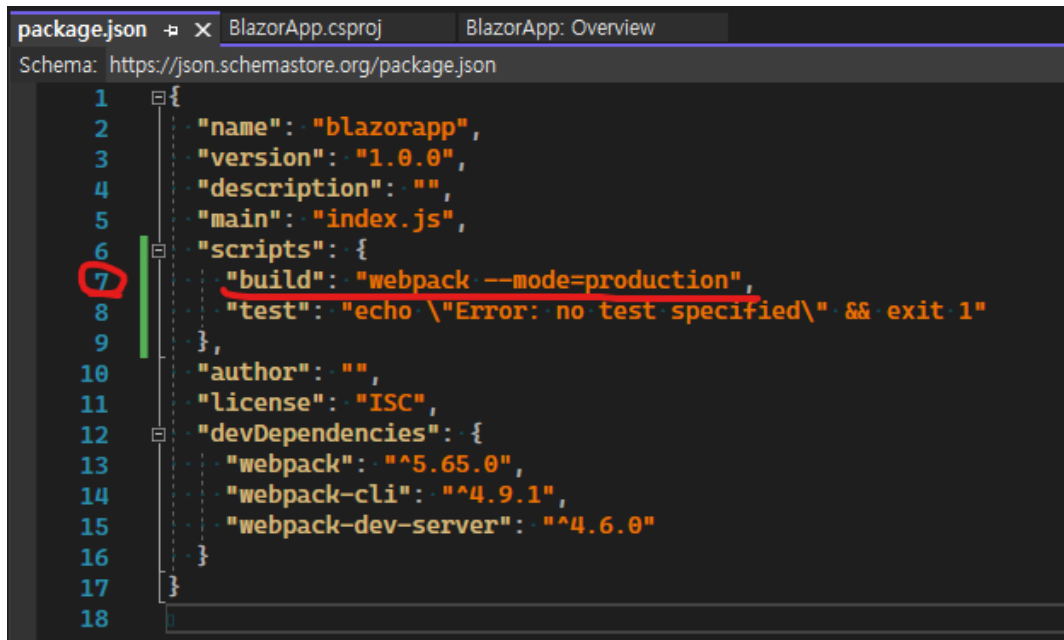
```
npm install --save-dev webpack webpack-cli webpack-dev-server
```

```
package.json  x BlazorApp.csproj  BlazorApp: Overview
Schema: https://json.schemastore.org/package.json
1  {
2    "name": "blazorapp",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "",
10   "license": "ISC",
11   "devDependencies": {
12     "webpack": "^5.65.0",
13     "webpack-cli": "^4.9.1",
14     "webpack-dev-server": "^4.6.0"
15   }
16 }
17
```



Webpack

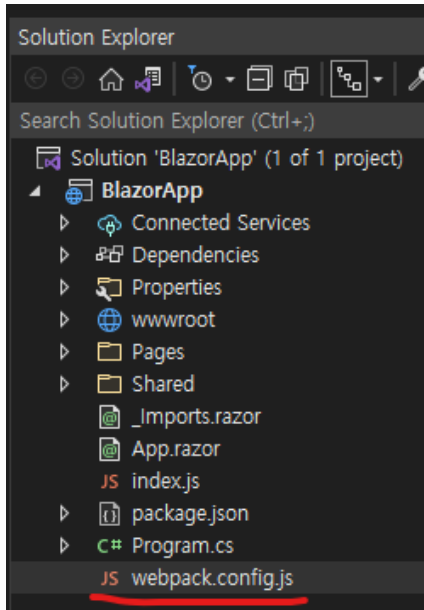
```
“build” : “webpack --mode=production”,
```



The screenshot shows a code editor with a dark theme. The top bar displays three tabs: 'package.json' (active), 'BlazorApp.csproj', and 'BlazorApp: Overview'. Below the tabs, the schema URL 'https://json.schemastore.org/package.json' is visible. The main area shows the 'package.json' file content with line numbers 1 through 18 on the left. The 'scripts' object is expanded, and the 'build' script is highlighted with a red circle and underline. The 'test' script is also visible. The 'devDependencies' section is partially visible at the bottom.

```
1 {
2   "name": "blazorapp",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "build": "webpack --mode=production",
8     "test": "echo \"Error: no test specified\" && exit 1"
9   },
10  "author": "",
11  "license": "ISC",
12  "devDependencies": {
13    "webpack": "^5.65.0",
14    "webpack-cli": "^4.9.1",
15    "webpack-dev-server": "^4.6.0"
16  }
17 }
18
```

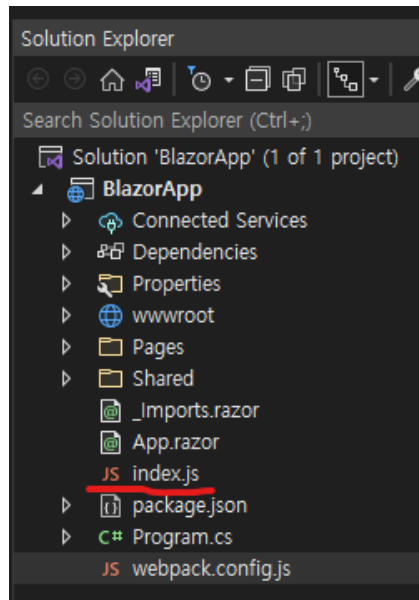
Webpack



```
index.js  webpack.config.js  x  package.json  BlazorApp: Overview
BlazorApp JavaScript Content Files  <global>

1  const webpack = require("webpack");
2  const path = require("path");
3
4  const isDevelopment = process.env.NODE_ENV !== "production";
5
6  module.exports = {
7    mode: isDevelopment ? "development" : "production",
8    entry: [
9      "./index.js"
10   ],
11   output: {
12     path: path.resolve(__dirname, "wwwroot"),
13     filename: "BlazorApp.js",
14   }
15 };
16
```

Webpack

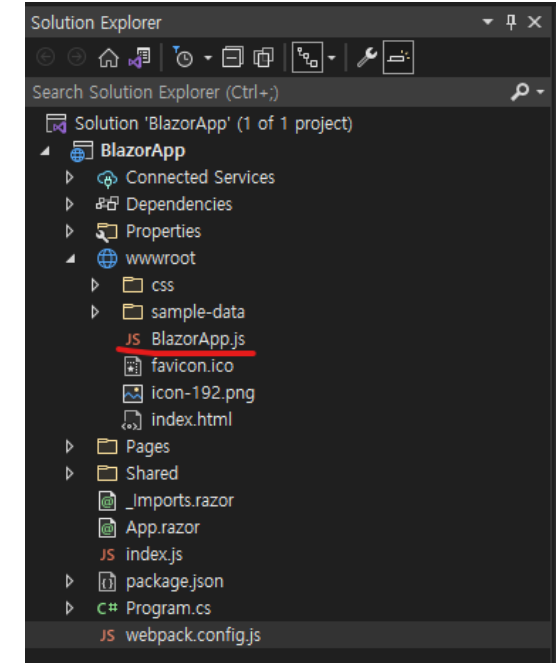


npm run build

```
> E:\Source\DotNetConf2022\BlazorApp > %master
> npm run build

> blazorapp@1.0.0 build
> webpack --mode=production

asset BlazorApp.js 0 bytes [emitted] [minimized] (name: main)
./index.js 3 bytes [built] [code generated]
webpack 5.65.0 compiled successfully in 223 ms
> E:\Source\DotNetConf2022\BlazorApp > %master
> |
```



TypeScript



TypeScript

```
npm install --save-dev typescript ts-loader
```

```
nothing added to commit but untracked files present
> E:\Source\DotNetConf2022\BlazorApp > |master
> npm install --save-dev typescript ts-loader

added 10 packages, and audited 337 packages in 3s

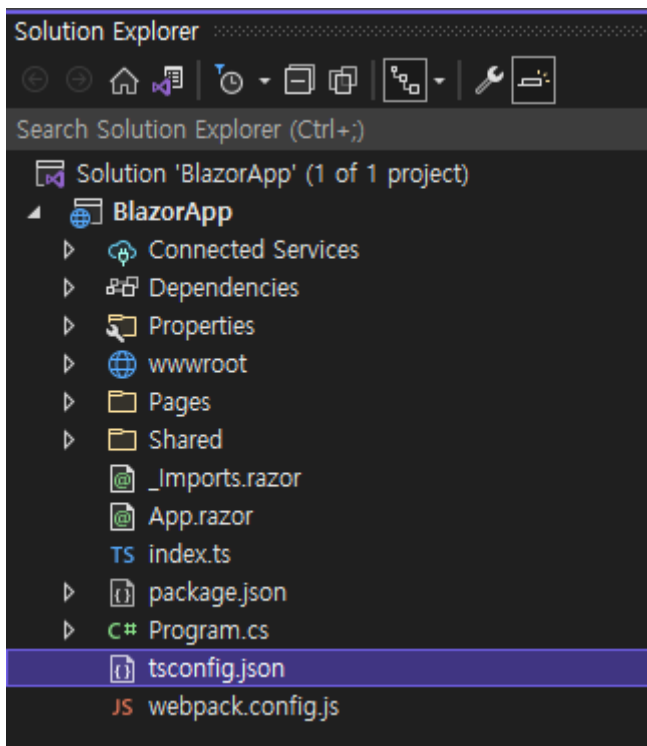
44 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
> E:\Source\DotNetConf2022\BlazorApp > |master
> |
```

```
package.json  X
Schema: https://json.schemastore.org/package.json

1  {
2    "name": "blazorapp",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "build": "webpack --mode=production",
8      "test": "echo \"Error: no test specified\" && exit 1"
9    },
10   "author": "",
11   "license": "ISC",
12   "devDependencies": {
13     "ts-loader": "^9.2.6",
14     "typescript": "^4.5.4",
15     "webpack": "^5.65.0",
16     "webpack-cli": "^4.9.1",
17     "webpack-dev-server": "^4.6.0"
18   }
19 }
20
```

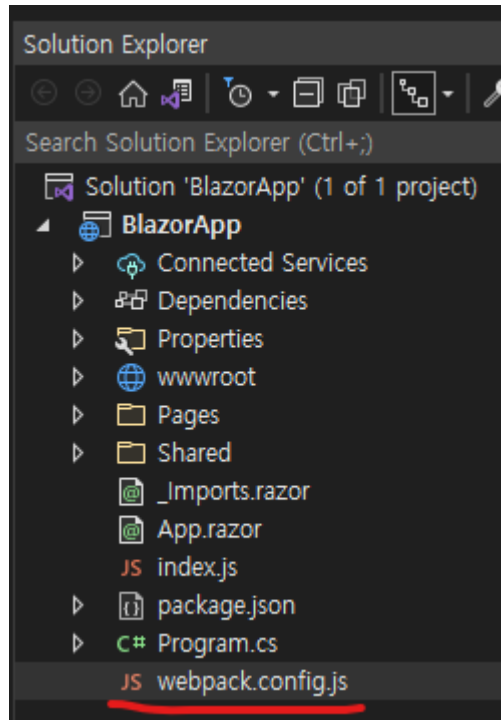
TypeScript



```
{
  "compilerOptions": {
    "baseUrl": "./", // 상대 경로의 base 폴더를 정한다. '.' 는 tsconfig.json 가 있는 폴더다.
    "outDir": "wwwroot" // TS 가 컴파일한 결과가 저장될 경로다.
  },

  "exclude": [ "node_modules" ]
}
```

TypeScript

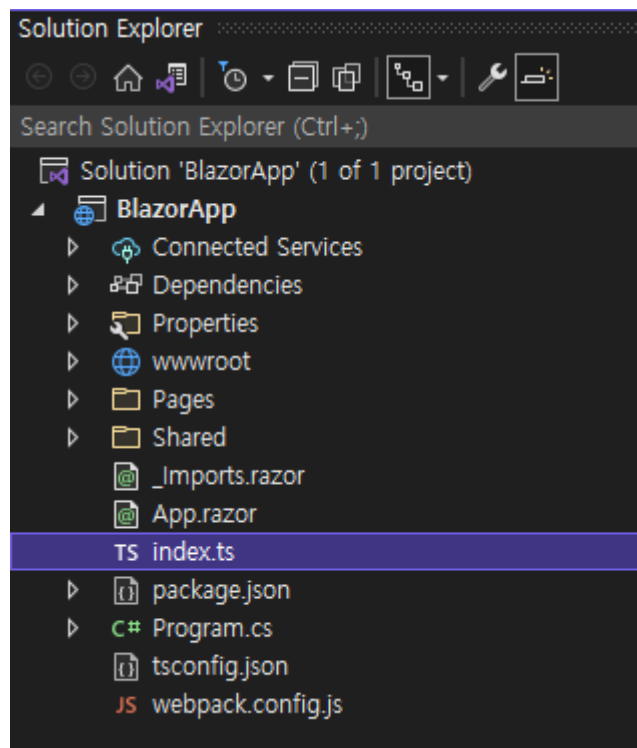


```
const webpack = require("webpack");
const path = require("path");

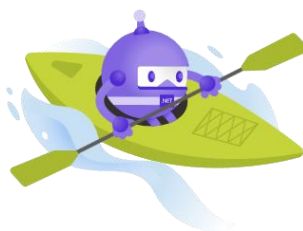
const isDevelopment = process.env.NODE_ENV !== "production";

module.exports = {
  mode: isDevelopment ? "development" : "production",
  entry: [
    "./index.ts"
  ],
  output: {
    path: path.resolve(__dirname, "wwwroot"),
    filename: "BlazorApp.js"
  },
  module: {
    rules: [
      // TypeScript
      {
        test: /\.tsx?$/,
        exclude: /node_modules/,
        use: "ts-loader"
      }
    ]
  }
},
};
```


TypeScript

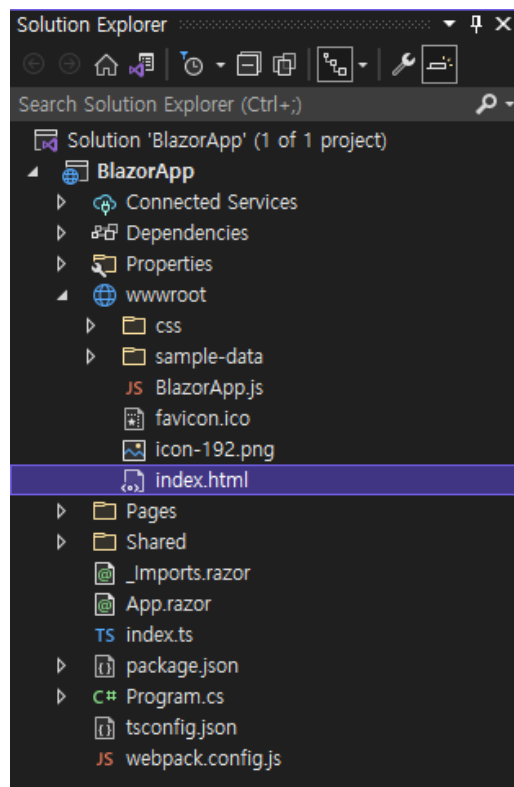


```
index.ts  [X]
E:\Source\DotNetConf2022\BlazorApp\tsconfig.json
1 console.log("Blazor");
2
```

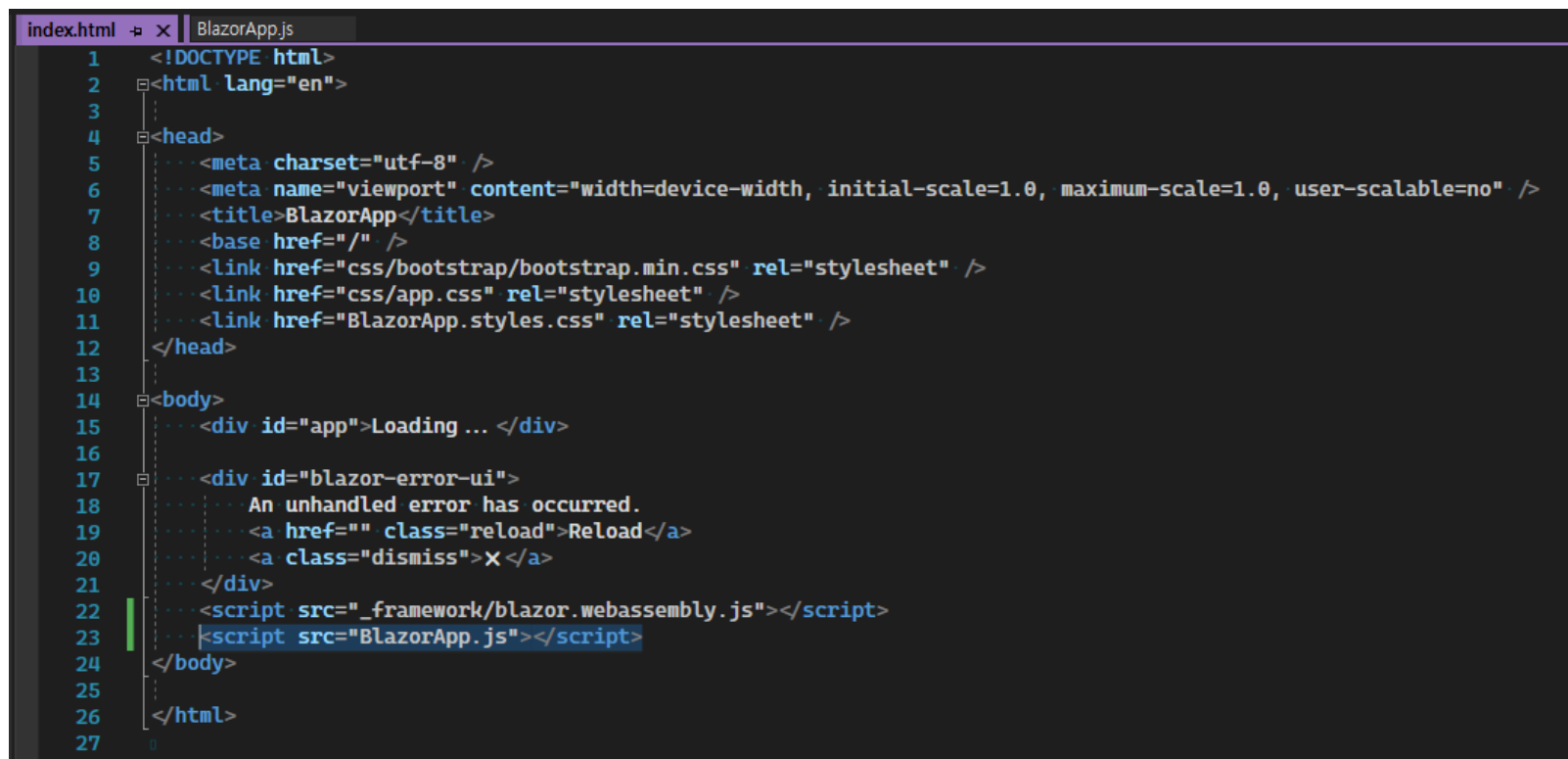


```
BlazorApp.js  [X]
Miscellaneous
1 console.log("Blazor");
```

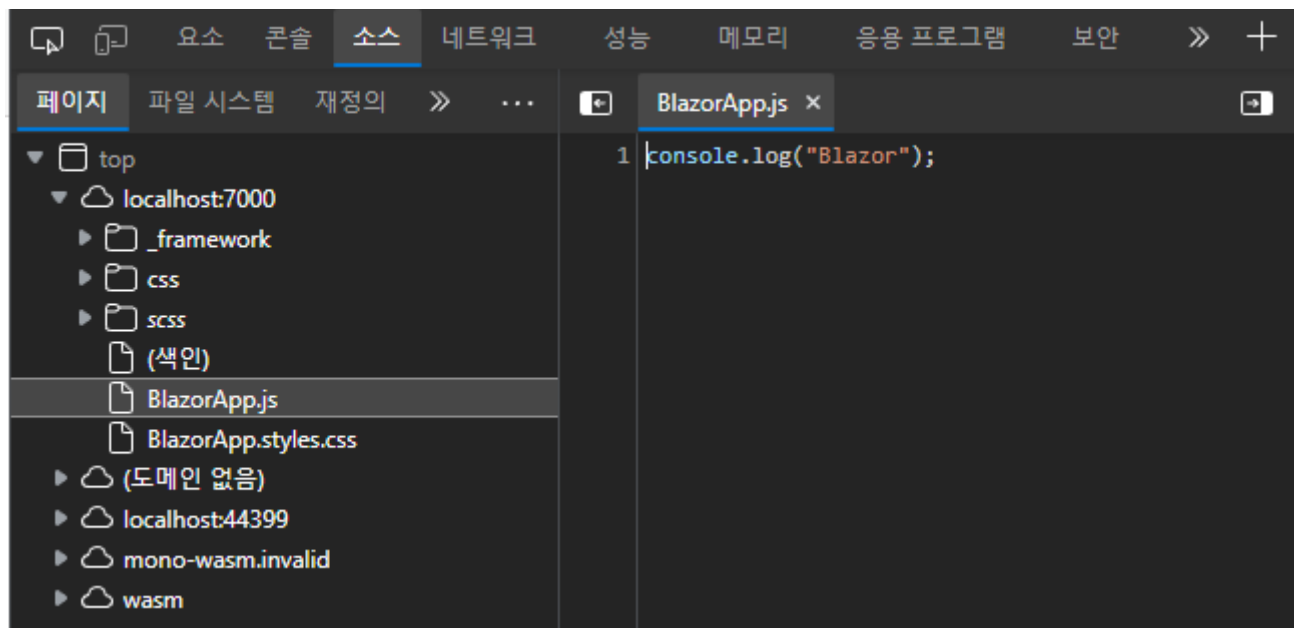
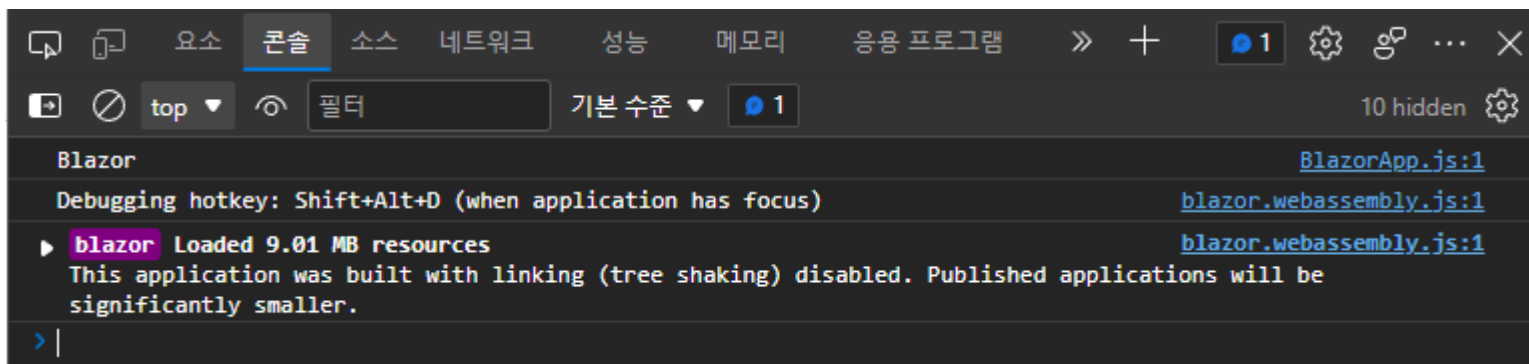
TypeScript



```
<script src="BlazorApp.js"></script>
```



TypeScript

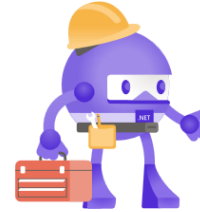


Sass



Sass

Sass

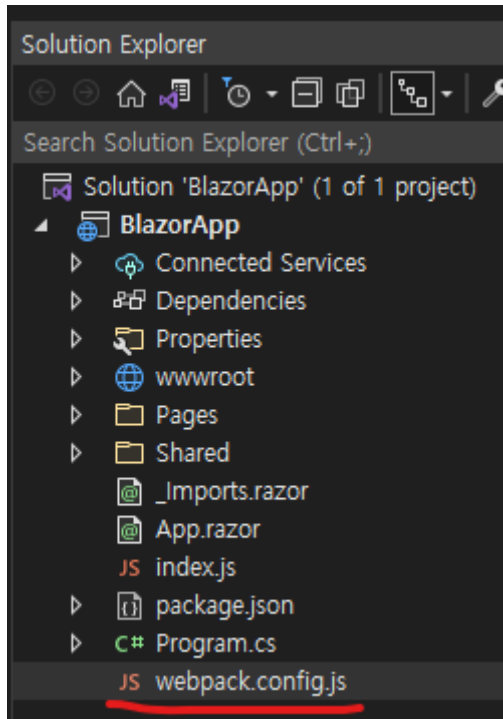


```
npm install --save-dev sass style-loader css-loader sass-loader
```

```
package.json  ✎ ✕
Schema: https://json.schemastore.org/package.json

1  {
2    "name": "blazorapp",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "build": "webpack --mode=production",
8      "watch": "webpack serve",
9      "test": "echo \"Error: no test specified\" && exit 1"
10  },
11  "author": "",
12  "license": "ISC",
13  "devDependencies": {
14    "css-loader": "^6.5.1",
15    "sass": "^1.45.0",
16    "sass-loader": "^12.4.0",
17    "style-loader": "^3.3.1",
18    "ts-loader": "^9.2.6",
19    "typescript": "^4.5.4",
20    "webpack": "^5.65.0",
21    "webpack-cli": "^4.9.1",
22    "webpack-dev-server": "^4.6.0"
23  }
24  }
25
```

Sass

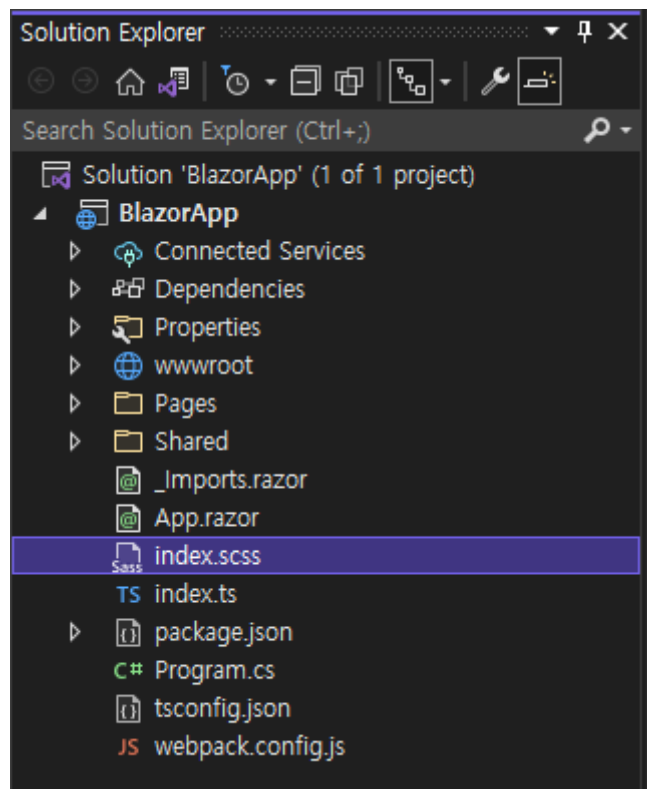


```
const webpack = require("webpack");
const path = require("path");

const isDevelopment = process.env.NODE_ENV !== "production";

module.exports = {
  mode: isDevelopment ? "development" : "production",
  entry: [
    "./index.ts"
  ],
  output: {
    path: path.resolve(__dirname, "wwwroot"),
    filename: "BlazorApp.js",
  },
  module: {
    rules: [
      // TypeScript
      {
        test: /\.tsx?$/,
        exclude: /node_modules/,
        use: "ts-loader",
      },
      // Style
      {
        test: /\.s[ac]ss$/i,
        exclude: /node_modules/,
        use: [
          "style-loader",
          "css-loader",
          "sass-loader",
        ],
      },
    ],
  },
};
```

Sass



```
body {  
  background-color: yellow;  
}
```



```
index.ts  x  
E:\Source\DotNetConf2022\BlazorApp\tsconfig.json  
1  import './index.scss';  
2  
3  console.log("Blazor");  
4
```

webpack 번들링에 추가하는 방법이 반드시 ts 파일에 scss 파일을 import 하는 방법만 있는건 아닙니다. scss 파일만 따로 모아 BlazorApp.css 파일로 만드는 방법도 있으나, 저는 번거롭다 판단하여 하나의 js 파일로 만들기 위해 현재 방법을 선택한 것 뿐입니다.

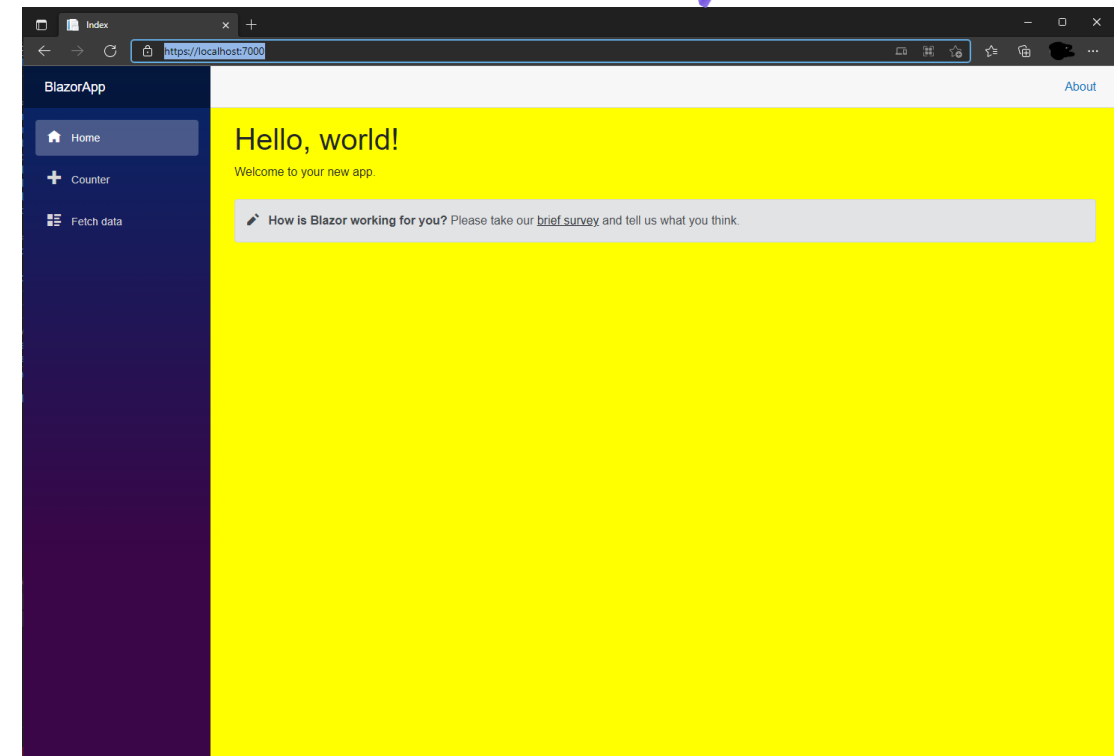
Sass

```
npm run build
```

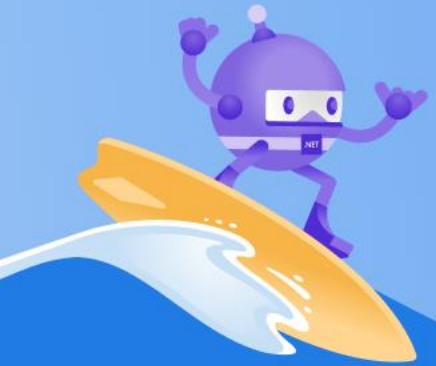
```
➤ E:\Source\DotNetConf2022\BlazorApp > #master wlfka@Primary < 01:18:16
> npm run build

> blazorapp@1.0.0 build
> webpack --mode=production

asset BlazorApp.js 4.23 KiB [emitted] [minimized] (name: main)
runtime modules 937 bytes 4 modules
modules by path ./node_modules/ 8.07 KiB
  modules by path ./node_modules/style-loader/dist/runtime/*.js 5.75 KiB 6 modules
  modules by path ./node_modules/css-loader/dist/runtime/*.js 2.33 KiB
    ./node_modules/css-loader/dist/runtime/noSourceMaps.js 64 bytes [built] [code generated]
    ./node_modules/css-loader/dist/runtime/api.js 2.26 KiB [built] [code generated]
modules by path .//*.scss 1.63 KiB
  ./index.scss 1.18 KiB [built] [code generated]
  ./node_modules/css-loader/dist/cjs.js!./node_modules/sass-loader/dist/cjs.js!./index.scss 452 bytes [built] [code generated]
./index.ts 93 bytes [built] [code generated]
webpack 5.65.0 compiled successfully in 2695 ms
➤ E:\Source\DotNetConf2022\BlazorApp > #master wlfka@Primary < 01:18:36
```



빌드 프로세스 개선



VISUAL STUDIO 2022



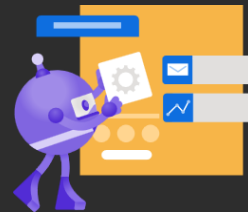
빌드 프로세스 개선

```
<Project Sdk="Microsoft.NET.Sdk.BlazorWebAssembly">

  <PropertyGroup>
    <TargetFramework>net6.0</TargetFramework>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="Microsoft.AspNetCore.Components.WebAssembly" Version="6.0.0" />
    <PackageReference Include="Microsoft.AspNetCore.Components.WebAssembly.DevServer" Version="6.0.0" PrivateAssets="all" />
  </ItemGroup>

</Project>
```



빌드 프로세스 개선

Running npm tasks when building a .NET project - Meziantou's blog

```
<!--  
  https://www.meziantou.net/running-npm-tasks-when-building-a-dotnet-project.htm  
  1. npm 패키지를 설치한다. 'Incremental builds' 가 적용되어 있기 때문에 package.json 파일이 수정되어야 실행된다.  
  https://docs.microsoft.com/en-us/visualstudio/msbuild/incremental-builds?view=vs-2022  
-->  
<Target Name="NpmInstall" Inputs="package.json" Outputs="node_modules/.install-stamp">  
  <!-- https://devblogs.microsoft.com/nuget/enable-repeatable-package-restores-using-a-lock-file/ -->  
  <PropertyGroup>  
    <RestorePackagesWithLockFile>true</RestorePackagesWithLockFile>  
  </PropertyGroup>  
  <Exec Command="npm ci" Condition="$(RestorePackagesWithLockFile) == 'true'" />  
  <Exec Command="npm install" Condition="$(RestorePackagesWithLockFile) != 'true'" />  
  
  <!-- 'Incremental builds' 가 되도록 stamp 파일을 업데이트한다. -->  
  <Touch Files="node_modules/.install-stamp" AlwaysCreate="true" />  
</Target>  
  
<!-- 2. 빌드를 시작하기 전에, webpack 으로 BlazorApp.js 파일을 만든다. -->  
<Target Name="NpmRunBuild" DependsOnTargets="NpmInstall" BeforeTargets="BeforeBuild">  
  <Exec Command="npm run build" />  
</Target>
```

빌드 프로세스 개선

```
BlazorApp.csproj  X
<Project Sdk="Microsoft.NET.Sdk.BlazorWebAssembly">

  <PropertyGroup>
    <TargetFramework>net6.0</TargetFramework>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>
  </PropertyGroup>

  <!--
  <<https://www.meziantou.net/running-npm-tasks-when-building-a-dotnet-project.htm
  <<<1. npm 패키지를 설치한다. 'Incremental builds'가 적용되어 있기 때문에 package.json 파일이 수정되어야 실행된다.
  <<<https://docs.microsoft.com/en-us/visualstudio/msbuild/incremental-builds?view=vs-2022
  <<-->
  <<<Target Name="NpmInstall" Inputs="package.json" Outputs="node_modules/.install-stamp">
  <<<<!-- https://devblogs.microsoft.com/nuget/enable-repeatabe-package-restores-using-a-lock-file/ -->
  <<<<PropertyGroup>
  <<<<<RestorePackagesWithLockFile>true</RestorePackagesWithLockFile>
  <<<<</PropertyGroup>
  <<<<<Exec Command="npm ci" Condition="$(RestorePackagesWithLockFile) == 'true'" />
  <<<<<Exec Command="npm install" Condition="$(RestorePackagesWithLockFile) != 'true'" />

  <<<<!-- 'Incremental builds'가 되도록 stamp 파일을 업데이트한다. -->
  <<<<<Touch Files="node_modules/.install-stamp" AlwaysCreate="true" />
  <<<<</Target>

  <<!-- 2. 빌드를 시작하기 전에, webpack으로 BlazorApp.js 파일을 만든다. -->
  <<<Target Name="NpmRunBuild" DependsOnTargets="NpmInstall" BeforeTargets="BeforeBuild">
  <<<<Exec Command="npm run build" />
  <<<<</Target>

  <<<ItemGroup>
  <<<<<PackageReference Include="Microsoft.AspNetCore.Components.WebAssembly" Version="6.0.0" />
  <<<<<PackageReference Include="Microsoft.AspNetCore.Components.WebAssembly.DevServer" Version="6.0.0" PrivateAssets="all" />
  <<<<</ItemGroup>

</Project>
```

빌드 프로세스 개선

```
package.json  X BlazorApp.csproj  BlazorApp: Overview
Schema: https://json.schemastore.org/package.json
1  {
2    "name": "blazorapp",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "build": "webpack --mode=production",
8      "test": "echo \\\"Error: no test specified\\\" && exit 1"
9    },
10   "author": "",
11   "license": "ISC",
12   "devDependencies": {
13     "webpack": "^5.65.0",
14     "webpack-cli": "^4.9.1",
15     "webpack-dev-server": "^4.6.0"
16   }
17 }
18
```

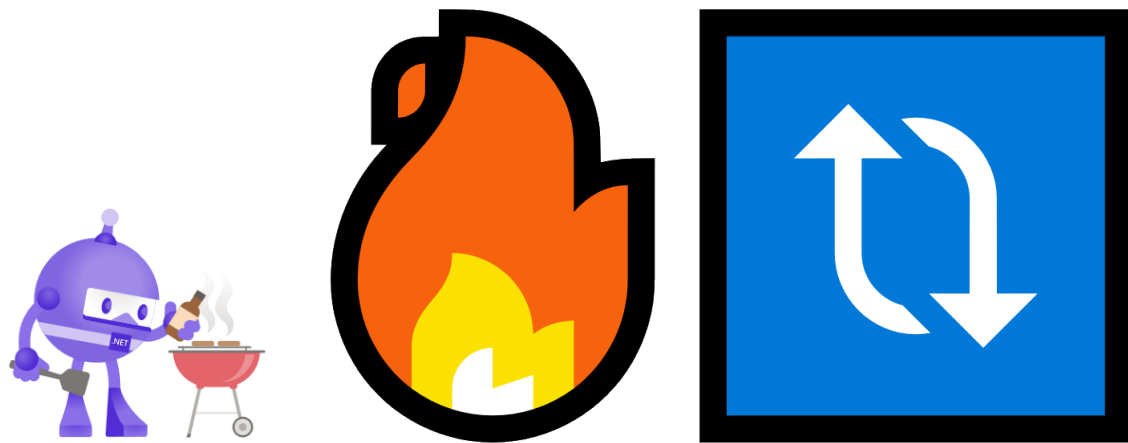
```
Output
Show output from: Build
Build started...
I>----- Build started: Project: BlazorApp, Configuration: Debug Any CPU -----
I>npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParams API instead.
I>
I>added 353 packages, and audited 354 packages in 23s
I>
I>48 packages are looking for funding
I>  run `npm fund` for details
I>
I>found 0 vulnerabilities
I>
I>> blazorapp@1.0.0 build
I>> webpack --mode=production
I>
I>asset  $\leftarrow$  [1m $\leftarrow$ [32mBlazorApp.js $\leftarrow$ [39m $\leftarrow$ [22m 4.23 KiB  $\leftarrow$ [1m $\leftarrow$ [32m[emitted] $\leftarrow$ [39m $\leftarrow$ [22m  $\leftarrow$ [1m $\leftarrow$ [32m[minimized] $\leftarrow$ [39m $\leftarrow$ [22m (name: main)
I>runtime modules 937 bytes 4 modules
I>modules by path  $\leftarrow$  [1m./node_modules/ $\leftarrow$ [39m $\leftarrow$ [22m 8.07 KiB
I>  modules by path  $\leftarrow$  [1m./node_modules/style-loader/dist/runtime/*.js $\leftarrow$ [39m $\leftarrow$ [22m 5.75 KiB 6 modules
I>    modules by path  $\leftarrow$  [1m./node_modules/css-loader/dist/runtime/*.js $\leftarrow$ [39m $\leftarrow$ [22m 2.33 KiB
I>       $\leftarrow$  [1m./node_modules/css-loader/dist/runtime/noSourceMaps.js $\leftarrow$ [39m $\leftarrow$ [22m 64 bytes  $\leftarrow$ [1m $\leftarrow$ [33m[built] $\leftarrow$ [39m $\leftarrow$ [22m  $\leftarrow$ [1m $\leftarrow$ [33m[code generated] $\leftarrow$ [39m $\leftarrow$ [22m
I>       $\leftarrow$  [1m./node_modules/css-loader/dist/runtime/api.js $\leftarrow$ [39m $\leftarrow$ [22m 2.26 KiB  $\leftarrow$ [1m $\leftarrow$ [33m[built] $\leftarrow$ [39m $\leftarrow$ [22m  $\leftarrow$ [1m $\leftarrow$ [33m[code generated] $\leftarrow$ [39m $\leftarrow$ [22m
I>modules by path  $\leftarrow$  [1m./s.css $\leftarrow$ [39m $\leftarrow$ [22m 1.62 KiB
I>   $\leftarrow$  [1m./index.scss $\leftarrow$ [39m $\leftarrow$ [22m 1.18 KiB  $\leftarrow$ [1m $\leftarrow$ [33m[built] $\leftarrow$ [39m $\leftarrow$ [22m  $\leftarrow$ [1m $\leftarrow$ [33m[code generated] $\leftarrow$ [39m $\leftarrow$ [22m
I>    ./node_modules/css-loader/dist/cjs.js!./node_modules/sass-loader/dist/cjs.js!./index.scss $\leftarrow$ [39m $\leftarrow$ [22m 451 bytes  $\leftarrow$ [1m $\leftarrow$ [33m[built] $\leftarrow$ [39m $\leftarrow$ [22m  $\leftarrow$ [1m $\leftarrow$ [33m[code generated] $\leftarrow$ [39m $\leftarrow$ [22m
I> $\leftarrow$  [1m./index.ts $\leftarrow$ [39m $\leftarrow$ [22m 93 bytes  $\leftarrow$ [1m $\leftarrow$ [33m[built] $\leftarrow$ [39m $\leftarrow$ [22m  $\leftarrow$ [1m $\leftarrow$ [33m[code generated] $\leftarrow$ [39m $\leftarrow$ [22m
I>webpack 5.65.0 compiled  $\leftarrow$ [1m $\leftarrow$ [32msuccessfully $\leftarrow$ [39m $\leftarrow$ [22m in 3115 ms
I>BlazorApp -> E:\Source\DotNetConf2022\BlazorApp\bin\Debug\net6.0\BlazorApp.dll
I>BlazorApp (Blazor output) -> E:\Source\DotNetConf2022\BlazorApp\bin\Debug\net6.0\wwwroot
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

빌드 프로세스 개선

Update on .NET Hot Reload progress and Visual Studio 2022 Highlights

*In Visual Studio 2022 GA release Hot Reload support for Blazor WebAssembly when using the Visual Studio debugger isn't enabled yet. You can still get Hot Reload If you start your app through Visual Studio without the debugger, and we are working to resolve this in the next Visual Studio update.

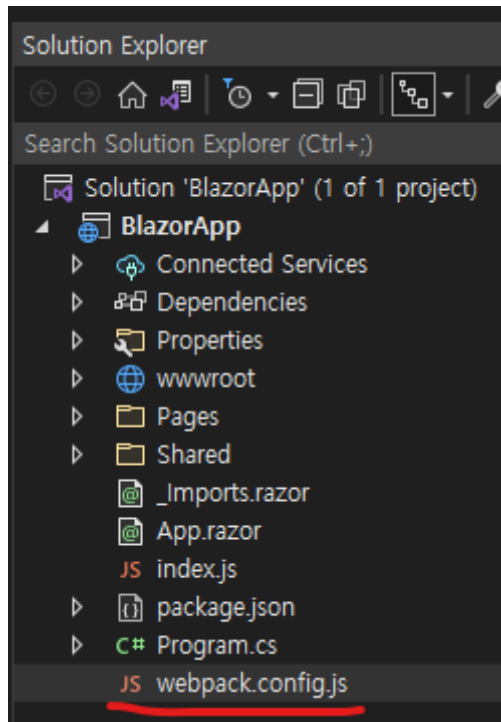
빌드 프로세스 개선



“Update on .NET Hot Reload progress and Visual Studio 2022 Highlights”

*In Visual Studio 2022 GA release Hot Reload support for Blazor WebAssembly when using the Visual Studio debugger isn't enabled yet. You can still get Hot Reload If you start your app through Visual Studio without the debugger, and we are working to resolve this in the next Visual Studio update.

빌드 프로세스 개선



```
devtool: "inline-source-map",
devServer: {
  static: {
    directory: path.resolve(__dirname, "wwwroot"),
  },
  port: 9000,
  historyApiFallback: true,
  hot: true,
  devMiddleware: {
    writeToDisk: true,
  }
}
```


빌드 프로세스 개선

```
webpack.config.js  X
Miscellaneous  <global>

1  const webpack = require("webpack");
2  const path = require("path");
3
4  const isDevelopment = process.env.NODE_ENV !== "production";
5
6  module.exports = {
7    mode: isDevelopment ? "development" : "production",
8    entry: [
9      "./index.ts"
10   ],
11   output: {
12     path: path.resolve(__dirname, "wwwroot"),
13     filename: "BlazorApp.js",
14   },
15   module: {
16     rules: [
17       // TypeScript
18       {
19         test: /\.tsx?$/,
20         exclude: /node_modules/,
21         use: "ts-loader",
22       },
23       // Style
24       {
25         test: /\.s[ac]ss$/i,
26         exclude: /node_modules/,
27         use: [
28           "style-loader",
29           "css-loader",
30           "sass-loader",
31         ],
32       },
33     ],
34   },
35   devtool: "inline-source-map",
36   devServer: {
37     static: {
38       directory: path.resolve(__dirname, "wwwroot"),
39     },
40     port: 9000,
41     historyApiFallback: true,
42     hot: true,
43     devMiddleware: {
44       writeToDisk: true,
45     }
46   }
47 }
```

빌드 프로세스 개선

```
package.json  X
Schema: https://json.schemastore.org/package.json

1  {
2    "name": "blazorapp",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "build": "webpack --mode=production",
8      "watch": "webpack serve",
9      "test": "echo \"Error: no test specified\" && exit 1"
10   },
11   "author": "",
12   "license": "ISC",
13   "devDependencies": {
14     "css-loader": "^6.5.1",
15     "sass": "^1.45.0",
16     "sass-loader": "^12.4.0",
17     "style-loader": "^3.3.1",
18     "ts-loader": "^9.2.6",
19     "typescript": "^4.5.4",
20     "webpack": "^5.65.0",
21     "webpack-cli": "^4.9.1",
22     "webpack-dev-server": "^4.6.0"
23   }
24 }
25
```

빌드 프로세스 개선

npm run watch

```
> E:\Source\DotNetConf2022\BlazorApp > ymaster wlfka@Primary 01:33:55
> npm run watch

> blazorapp@1.0.0 watch
> webpack serve

<i> [webpack-dev-server] Project is running at:
<i> [webpack-dev-server] Loopback: http://localhost:9000/
<i> [webpack-dev-server] On Your Network (IPv4): http://192.168.219.100:9000/
<i> [webpack-dev-server] Content not from webpack is served from 'E:\Source\DotNetConf2022\BlazorApp\wwwroot' directory
<i> [webpack-dev-server] 404s will fallback to '/index.html'
asset BlazorApp.js 716 KiB [emitted] (name: main)
runtime modules 27.1 KiB 13 modules
modules by path ./node_modules/ 208 KiB
  modules by path ./node_modules/webpack-dev-server/client/ 52.8 KiB 12 modules
  modules by path ./node_modules/style-loader/dist/runtime/*.js 5.75 KiB 6 modules
  modules by path ./node_modules/webpack/hot/*.js 4.3 KiB 4 modules
  modules by path ./node_modules/html-entities/lib/*.js 81.3 KiB 4 modules
  modules by path ./node_modules/url/ 37.4 KiB 3 modules
  modules by path ./node_modules/querystring/*.js 4.51 KiB 3 modules
  modules by path ./node_modules/css-loader/dist/runtime/*.js 2.94 KiB 2 modules
  ./node_modules/ansi-html-community/index.js 4.16 KiB [built] [code generated]
  ./node_modules/events/events.js 14.5 KiB [built] [code generated]
modules by path ./*.scss 2.97 KiB
  ./index.scss 2.35 KiB [built] [code generated]
  ./node_modules/css-loader/dist/cjs.js!./node_modules/sass-loader/dist/cjs.js!./index.scss 636 bytes [built] [code generated]
  ./index.ts 93 bytes [built] [code generated]
webpack 5.65.0 compiled successfully in 5216 ms
```


빌드 프로세스 개선

npm run watch

```
E:\Source\DotNetConf2022\BlazorApp > master wlfka@Primary 01:33:55
> npm run watch

> blazorapp@1.0.0 watch
> webpack serve

<i> [webpack-dev-server] Project is running at:
<i> [webpack-dev-server] Loopback: http://localhost:9000/
<i> [webpack-dev-server] On Your Network (IPv4): http://192.168.219.100:9000/
<i> [webpack-dev-server] Content not from webpack is served from 'E:\Source\DotNetConf2022\BlazorApp\wwwroot' directory
<i> [webpack-dev-server] 404s will fallback to '/index.html'
asset BlazorApp.js 716 KiB [emitted] (name: main)
runtime modules 27.1 KiB 13 modules
modules by path ./node_modules/ 208 KiB
  modules by path ./node_modules/webpack-dev-server/client/ 52.8 KiB 12 modules
  modules by path ./node_modules/style-loader/dist/runtime/*.js 5.75 KiB 6 modules
  modules by path ./node_modules/webpack/hot/*.js 4.3 KiB 4 modules
  modules by path ./node_modules/html-entities/lib/*.js 81.3 KiB 4 modules
  modules by path ./node_modules/url/ 37.4 KiB 3 modules
  modules by path ./node_modules/querystring/*.js 4.51 KiB 3 modules
  modules by path ./node_modules/css-loader/dist/runtime/*.js 2.94 KiB 2 modules
  ./node_modules/ansi-html-community/index.js 4.16 KiB [built] [code generated]
  ./node_modules/events/events.js 14.5 KiB [built] [code generated]
modules by path ./*.scss 2.97 KiB
  ./index.scss 2.35 KiB [built] [code generated]
  ./node_modules/css-loader/dist/cjs.js!./node_modules/sass-loader/dist/cjs.js!./index.scss 636 bytes [built] [code generated]
./index.ts 93 bytes [built] [code generated]
webpack 5.65.0 compiled successfully in 5216 ms
asset BlazorApp.js 716 KiB [emitted] (name: main)
asset main.6e750a7549cc0562af5d.hot-update.js 1.24 KiB [emitted] [immutable] [hmr] (name: main)
asset main.6e750a7549cc0562af5d.hot-update.json 28 bytes [emitted] [immutable] [hmr]
Entrypoint main 717 KiB = BlazorApp.js 716 KiB main.6e750a7549cc0562af5d.hot-update.js 1.24 KiB
cached modules 211 KiB [cached] 38 modules
runtime modules 27.1 KiB 13 modules
./index.ts 104 bytes [built] [code generated]
webpack 5.65.0 compiled successfully in 108 ms
|
```



빌드 프로세스 개선

dotnet watch run

```
* > E:\Source\DotNetConf2022\BlazorApp > ?master wlfka@Primary < 21:52:39 e
> dotnet watch run
watch : Hot reload enabled. For a list of supported edits, see https://aka.ms/dotnet/hot-reload. Press "Ctrl + R" to restart.
watch : Building...
복원할 프로젝트를 확인하는 중...
복원할 모든 프로젝트가 최신 상태입니다.

> blazorapp@1.0.0 build
> webpack --mode=production

asset BlazorApp.js 32.7 KiB [compared for emit] [minimized] (name: main)
runtime modules 937 bytes 4 modules
modules by path ./node_modules/ 8.68 KiB
  modules by path ./node_modules/style-loader/dist/runtime/*.js 5.75 KiB 6 modules
  modules by path ./node_modules/css-loader/dist/runtime/*.js 2.94 KiB
    ./node_modules/css-loader/dist/runtime/sourceMaps.js 688 bytes [built] [code generated]
    ./node_modules/css-loader/dist/runtime/api.js 2.26 KiB [built] [code generated]
modules by path ./*.scss 1.79 KiB
  ./index.scss 1.18 KiB [built] [code generated]
  ./node_modules/css-loader/dist/cjs.js!./node_modules/sass-loader/dist/cjs.js!./index.scss 620 bytes [built] [code generated]
./index.ts 104 bytes [built] [code generated]
webpack 5.65.0 compiled successfully in 2186 ms
BlazorApp -> E:\Source\DotNetConf2022\BlazorApp\bin\Debug\net6.0\BlazorApp.dll
BlazorApp (Blazor output) -> E:\Source\DotNetConf2022\BlazorApp\bin\Debug\net6.0\wwwroot
watch : Started
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7000
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: E:\Source\DotNetConf2022\BlazorApp
```

빌드 프로세스 개선

Microsoft Store

앱, 게임, 영화 등 검색

Windows Terminal

Microsoft Corporation

열기

4.6 ★ 76

평균 등급

Windows 터미널은 명령 프롬프트, PowerShell 및 WSL과 같은 명령 줄 도구 및 셸 사용자를 위한 최신의 빠르고 효율적이며 강력한 생산성의 터미널 응용 프로그램입니다. 주요 기능으로는 여러 탭, 창, 유니 코드 및 UTF-8 문자 지원, GPU 가속 텍스트 렌더링 엔진 및 사용자 정의 테마, 스타일 및 구성이 있습니다.

이것은 오픈 소스 프로젝트이며 커뮤니티 참여를 환영합니다. 참여하려면 <https://github.com/microsoft/terminal>을 방문하십시오

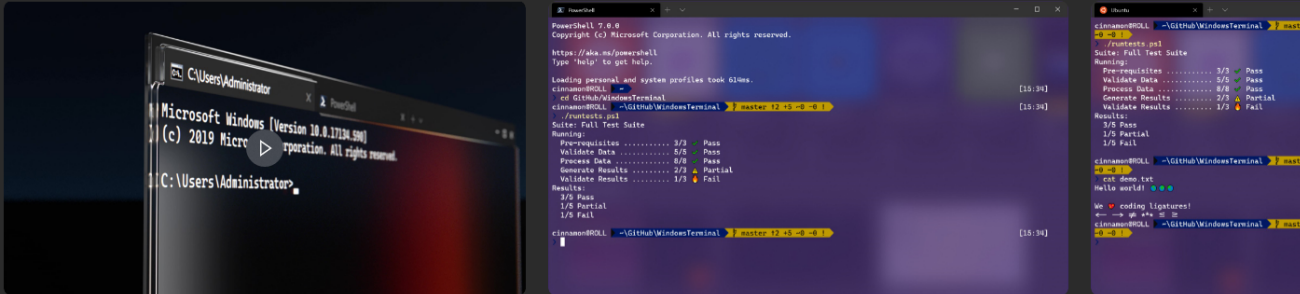
개발자 도구

라이브러리

3+ 연령 등급: IARC 3+

도움말

스크린샷



설명

Windows 터미널은 명령 프롬프트, PowerShell 및 WSL과 같은 명령 줄 도구 및 셸 사용자를 위한 최신의 빠르고 효율적이며 강력한 생산성의 터미널 응용 프로그램입니다. 주요 기능으로는 여러 탭, 창, 유니 코드 및 UTF-8 문자 지원, GPU 가속 텍스트 렌더링 엔진 및 사용자 정의 테마, 스타일 및 구성이 있습니다.

이것은 오픈 소스 프로젝트이며 커뮤니티 참여를 환영합니다. 참여하려면 <https://github.com/microsoft/terminal>을 방문하십시오

평점 및 리뷰

4.6

평가 76개

리뷰 남기기

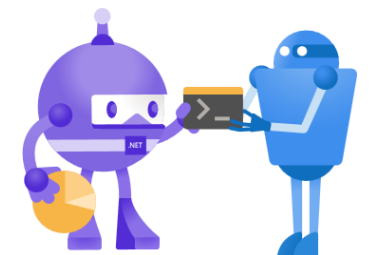
우리 마소가 달라졌어요

★★★★★ 가장 유용하고 긍정적인 리뷰

드디어 까만 화면에 하얀 글씨에서 진화

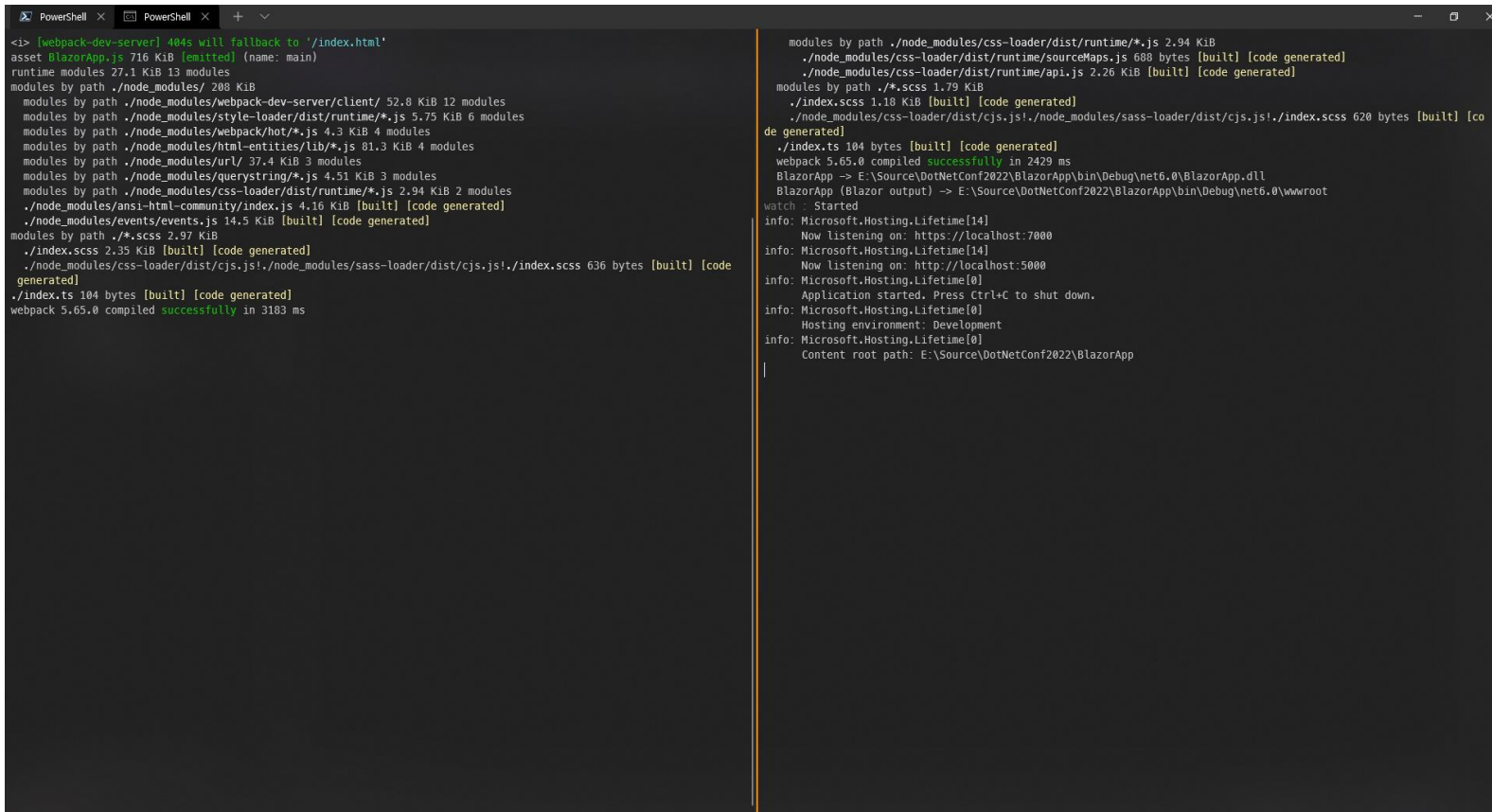
수민 2년 전

18 0



빌드 프로세스 개선

```
wt --window 0 --focus -d . PowerShell -c { npm run watch }  
wt --window 0 split-pane --focus -d . PowerShell -c { dotnet watch run }
```



```
<i> [webpack-dev-server] 404s will fallback to '/index.html'  
asset BlazorApp.js 716 KiB [emitted] (name: main)  
runtime modules 27.1 KiB 13 modules  
modules by path ./node_modules/ 208 KiB  
  modules by path ./node_modules/webpack-dev-server/client/ 52.8 KiB 12 modules  
  modules by path ./node_modules/style-loader/dist/runtime/*.js 5.75 KiB 6 modules  
  modules by path ./node_modules/webpack/hot/*.js 4.3 KiB 4 modules  
  modules by path ./node_modules/html-entities/lib/*.js 81.3 KiB 4 modules  
  modules by path ./node_modules/url/ 37.4 KiB 3 modules  
  modules by path ./node_modules/querystring/*.js 4.51 KiB 3 modules  
  modules by path ./node_modules/css-loader/dist/runtime/*.js 2.94 KiB 2 modules  
  ./node_modules/ansi-html-community/index.js 4.16 KiB [built] [code generated]  
  ./node_modules/events/events.js 14.5 KiB [built] [code generated]  
modules by path ./*.scss 2.97 KiB  
  ./index.scss 2.35 KiB [built] [code generated]  
  ./node_modules/css-loader/dist/cjs.js!/node_modules/sass-loader/dist/cjs.js!/index.scss 636 bytes [built] [code generated]  
  ./index.ts 104 bytes [built] [code generated]  
webpack 5.65.0 compiled successfully in 3183 ms  
  
modules by path ./node_modules/css-loader/dist/runtime/*.js 2.94 KiB  
  ./node_modules/css-loader/dist/runtime/sourceMaps.js 680 bytes [built] [code generated]  
  ./node_modules/css-loader/dist/runtime/api.js 2.26 KiB [built] [code generated]  
modules by path ./*.scss 1.79 KiB  
  ./index.scss 1.18 KiB [built] [code generated]  
  ./node_modules/css-loader/dist/cjs.js!/node_modules/sass-loader/dist/cjs.js!/index.scss 620 bytes [built] [code generated]  
  ./index.ts 104 bytes [built] [code generated]  
webpack 5.65.0 compiled successfully in 2429 ms  
BlazorApp -> E:\Source\DotNetConf2022\BlazorApp\bin\Debug\net6.0\BlazorApp.dll  
BlazorApp (Blazor output) -> E:\Source\DotNetConf2022\BlazorApp\bin\Debug\net6.0\wwwroot  
watch : Started  
info: Microsoft.Hosting.Lifetime[14]  
  Now listening on: https://localhost:7000  
info: Microsoft.Hosting.Lifetime[14]  
  Now listening on: http://localhost:5000  
info: Microsoft.Hosting.Lifetime[0]  
  Application started. Press Ctrl+C to shut down.  
info: Microsoft.Hosting.Lifetime[0]  
  Hosting environment: Development  
info: Microsoft.Hosting.Lifetime[0]  
  Content root path: E:\Source\DotNetConf2022\BlazorApp
```

고맙습니다!

