jongio / BlazorDocker Public

Code Issues 2 Pull requests 1 Actions Projects Security Insights

master 2 Branches 0 Tags Go to file Code

jongio Updated Hosted to serve from ASP.NET and created a new ClientServer p... eb5992a · 4 years ago

ClientServer	Updated Hosted to serve from ASP.NET and created a ne...	4 years ago
Hosted	Updated Hosted to serve from ASP.NET and created a ne...	4 years ago
Server	Updated Hosted to serve from ASP.NET and created a ne...	4 years ago
Standalone	Updated Hosted to serve from ASP.NET and created a ne...	4 years ago
.gitignore	Init commit	4 years ago
LICENSE	Init commit	4 years ago
README.md	Updated Hosted to serve from ASP.NET and created a ne...	4 years ago

README MIT license

Blazor Docker - How to Containerize Blazor WebAssembly Standalone, WebAssembly Hosted, and Blazor Server apps

This repo contains working examples for containerizing Blazor applications with four different hosting models.

ClientServer - A Blazor WebAssembly client app hosted on nginx and ASP.NET Core server app hosted in ASP.NET Core.

```
  - Created with `dotnet new blazorwasm --hosted -o . -n BlazorClientServer`
```

Hosted - A Blazor WebAssembly client app hosted on ASP.NET Core server.

```
  - Created with `dotnet new blazorwasm --hosted -o . -n BlazorHosted`
```

Server - A Blazor Server app hosted by ASP.NET Core.

```
  - Created with `dotnet new blazorserver -o . -n BlazorServer`
```

Standalone - A Blazor WebAssembly client app hosted on nginx.

```
  - Created with `dotnet new blazorwasm -o . -n BlazorStandalone`
```

How to run

Run `docker-compose up --build` from ClientServer, Hosted, Server, or Standalone folders to run each version and open the corresdponding localhost endpoint that is found in the docker-compose file for the Blazor app.

ClientServer

A Blazor WebAssembly client app hosted on nginx and ASP.NET Core server app hosted in ASP.NET Core. It is broken up into three projects "Client", "Server", and "Shared".

- docker-compose.yml - builds and runs both the Client and the Server projects.

How to run

Run `docker-compose up --build` from ClientServer folder. Open browser and go to http://localhost:5080/

Client Project - uses nginx on alpine base image.

- **/Pages/FetchData.razor** - includes the following code to show you what endpoint is being hit.

  ```
  @using Microsoft.Extensions.Configuration
  @inject IConfiguration Configuration

  <h1>Weather forecast</h1>
  <p>SERVER_HOST: @Configuration["SERVER_HOST"]</p>
  ```

- **/wwwroot/appsettings.json** - includes SERVER_HOST setting for production or when running in a container. You'll want to change this to match your production endpoint.
- **/wwwroot/appsettings.Development.json** - includes SERVER_HOST setting for development machine.
- **/Dockerfile** - builds, publishes, and uses nginx to host it.
- **/nginx.conf** - the nginx.conf file needed to serve the site.
- **/Program.cs** - includes the following code to read the SERVER_HOST setting from appsettings.json

  ```
  var serverHost = string.IsNullOrEmpty(builder.Configuration["SERVER_HOST"]) ?
      builder.HostEnvironment.BaseAddress :
      builder.Configuration["SERVER_HOST"];

      builder.Services.AddTransient(sp => new HttpClient { BaseAddress = new Uri(serverHost) });
  ```

Server Project- uses ASP.NET runtime image

- **/Dockerfile** - builds and starts ASP.NET app
- **/Startup.cs** - includes code to enable Cors, so the client app can call from a different host.

```
readonly string CorsOrigins = "CorsOrigins";

services.AddCors(options =>
       {
           options.AddPolicy(CorsOrigins,
               builder => builder.AllowAnyOrigin()
               .AllowAnyMethod()
               .AllowAnyHeader());
       });

app.UseCors(CorsOrigins);
```

Hosted

A Blazor WebAssembly client app hosted on ASP.NET Core server.

How to run

Run `docker-compose up --build` from Hosted folder. Open browser and go to http://localhost:6080/

Client Project

- No changes made to Client project from base template.

Server - uses ASP.NET runtime image

- **/Dockerfile** - builds and starts ASP.NET app

Server

How to run

Run `docker-compose up --build` from Server folder. Open browser and go to http://localhost:7080/

Server - Uses ASP.NET Core runtime host

- **/Dockerfile** - builds, publishes, and starts the ASP.NET site.

Standalone

How to run

Run `docker-compose up --build` from Standalone folder. Open browser and go to http://localhost:8080/

Standalone - uses nginx on alpine base image.

- **Dockerfile** - builds, publishes, and uses nginx to host it.
- **/nginx.conf** - the nginx.conf file needed to serve the site.

Releases

No releases published

Packages

No packages published

Languages

HTML 41.5% **C#** 29.6% **CSS** 23.9% **Dockerfile** 5.0%