.NET Core

In this lesson, you will be provided with the basic application image for .Net Core.

WE'LL COVER THE FOLLOWING

- Result
- Files

Result

The resulting image is published as *learnbook/aspnetcore-server*. You can run a container from it with the following command:

```
docker run --rm -it -p 8088:80 learnbook/aspnetcore-server
```

Then point your browser to `http://localhost:8088`

Files

> You can find these files in the *code/common-development-profiles/demos/aspnet-core* folder.

A static HTML file to be served:

wwwroot/index.htm

```
<h1>Hello !</h1>
<p>This page is served by ASP.NET Core</p>
<p>Try our <a href="/v1/square/4">multiply API</a>.</p>
```

A REST API coded with ASP.NET Core MVC:

ApiController.cs

```
using System;
using Microsoft.AspNetCore.Mvc;

namespace aspnet_core
{
    [Route("v1")]
    [ApiController]
    public class ValuesController : ControllerBase
    {
        // GET api/values
        [HttpGet("square/{value}")]
        public IActionResult Square(double value)
        {
            var result = Math.Pow(value, 2);
            return Ok(new {
                value = value,
                result = result
            });
        }
    }
}
```

Configuration of the HTTP server (middleware and dependency injection):

Startup.cs

```
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;

namespace aspnet_core
{
    public class Startup
    {
        public void ConfigureServices(IServiceCollection services)
        {
            services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_1);
        }

        public void Configure(IApplicationBuilder app, IHostingEnvironment env)
        {
            app.UseDeveloperExceptionPage();
            app.UseDefaultFiles();
            app.UseStaticFiles();
            app.UseMvc();
        }
```

```
    }
}
```

Spin up the HTTP server:

Program.cs

```
using Microsoft.AspNetCore;
using Microsoft.AspNetCore.Hosting;

namespace aspnet_core
{
    public class Program
    {
        public static void Main(string[] args)
        {
            WebHost.CreateDefaultBuilder(args)
                .UseStartup<Startup>()
                .Build()
                .Run();
        }
    }
}
```

The .csproj file containing build instructions:

aspnet-core.csproj

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

  <PropertyGroup>
    <TargetFramework>netcoreapp2.2</TargetFramework>
  </PropertyGroup>

  <ItemGroup>
    <Folder Include="wwwroot\" />
  </ItemGroup>

  <ItemGroup>
    <PackageReference Include="Microsoft.AspNetCore.App" />
    <PackageReference Include="Microsoft.AspNetCore.Razor.Design" Version="2.1.2" PrivateAsse
  </ItemGroup>

</Project>
```

Definition of the image to build:

Dockerfile

```
# Use an image with the SDK for compilation
FROM microsoft/dotnet:2.2-sdk AS builder
WORKDIR /app

# Get the build file
COPY *.csproj  .
# Optional. Run this first so that it is cached
RUN dotnet restore

# Get the source code inside the image
COPY . .
RUN dotnet publish --output /out/ --configuration Release

# Create a lightweight image
FROM microsoft/dotnet:2.2-aspnetcore-runtime-alpine
WORKDIR /app
# Copy compiled artifacts from previous image
COPY --from=builder /out .
EXPOSE 80
ENTRYPOINT ["dotnet", "aspnet-core.dll"]
```

Let’s look at another image in the next lesson.