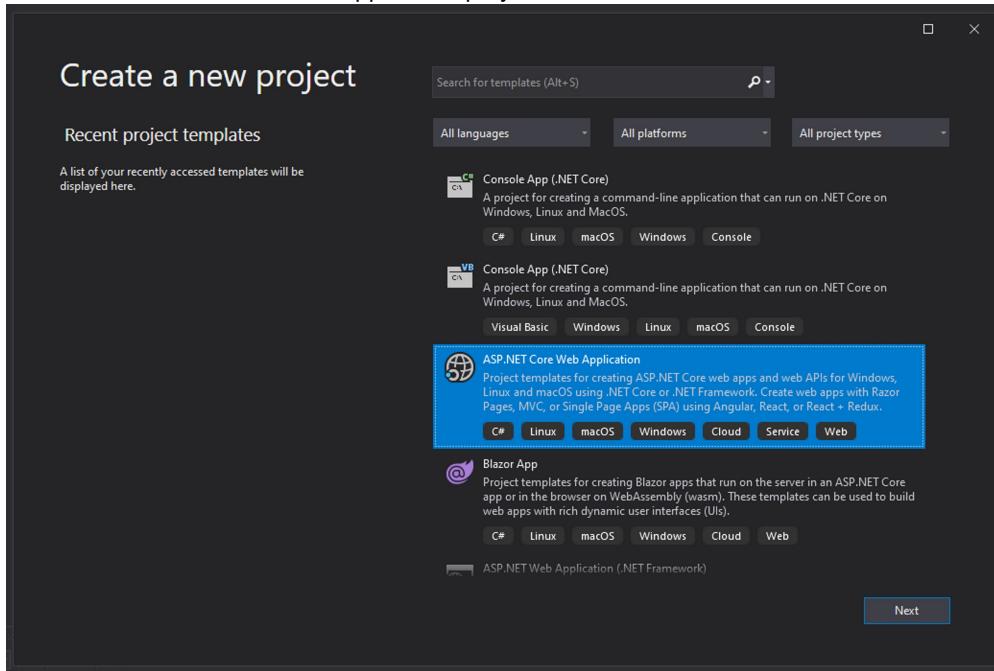


# .NET Core Project Setup

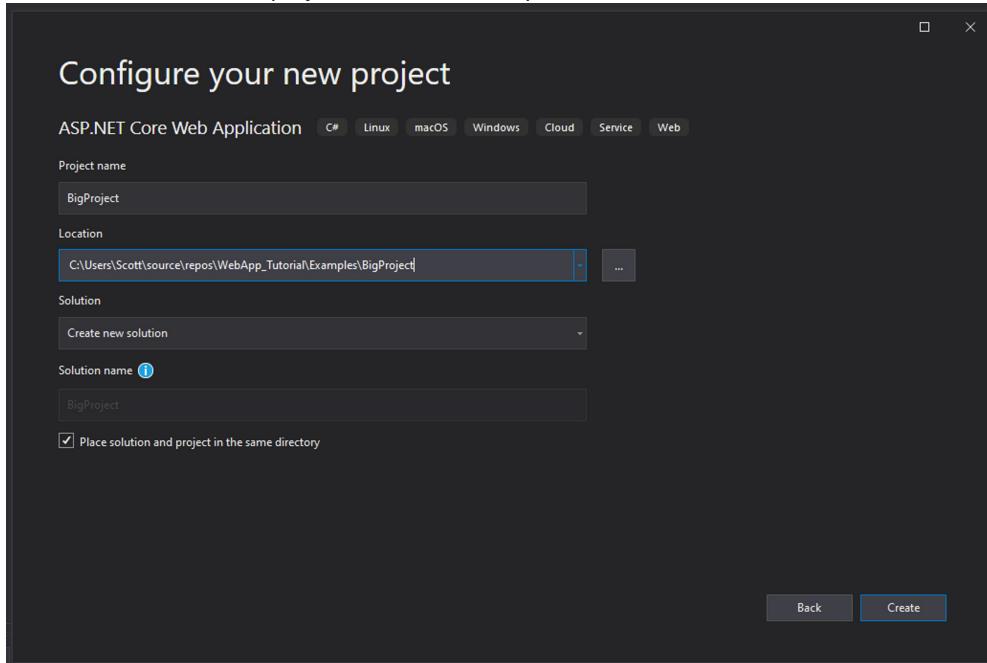
Thursday, December 17, 2020 11:47 PM

## Basic Settings

Create new ASP.NET Core Web Application project.

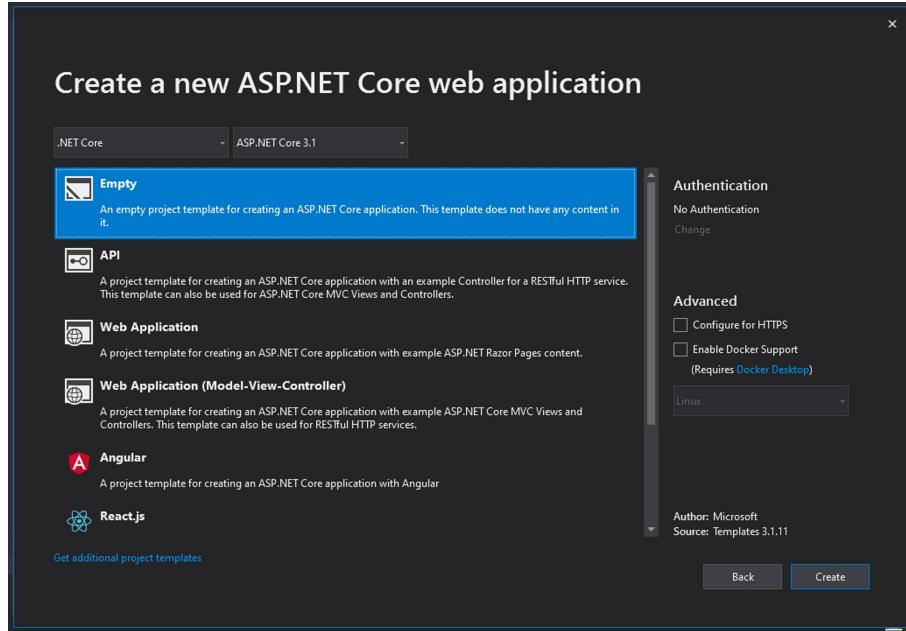


Note: Place solution and project in same directory enabled.



## Empty Template

Empty template with HTTPS configuration turned off for self-setup.



Creates a void Main method listening for web requests.

```
1  using Microsoft.AspNetCore.Hosting;
2  using Microsoft.Extensions.Configuration;
3  using Microsoft.Extensions.Hosting;
4  using Microsoft.Extensions.Logging;
5  using System;
6  using System.Collections.Generic;
7  using System.Linq;
8  using System.Threading.Tasks;
9
10 namespace DutchTreat
11 {
12     public class Program
13     {
14         public static void Main(string[] args)
15         {
16             CreateHostBuilder(args).Build().Run();
17         }
18
19         public static IHostBuilder CreateHostBuilder(string[] args) =>
20             Host.CreateDefaultBuilder(args)
21                 .ConfigureWebHostDefaults(webBuilder =>
22                 {
23                     webBuilder.UseStartup<Startup>();
24                 });
25     }
26 }
27
```

The screenshot shows the Visual Studio IDE with the 'DutchTreat' project open in Solution Explorer. The 'Program.cs' file is the active code editor, showing the following C# code:

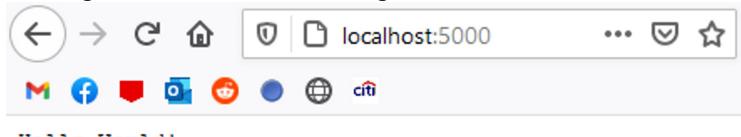
```
1  using Microsoft.AspNetCore.Hosting;
2  using Microsoft.Extensions.Configuration;
3  using Microsoft.Extensions.Hosting;
4  using Microsoft.Extensions.Logging;
5  using System;
6  using System.Collections.Generic;
7  using System.Linq;
8  using System.Threading.Tasks;
9
10 namespace DutchTreat
11 {
12     public class Program
13     {
14         public static void Main(string[] args)
15         {
16             CreateHostBuilder(args).Build().Run();
17         }
18
19         public static IHostBuilder CreateHostBuilder(string[] args) =>
20             Host.CreateDefaultBuilder(args)
21                 .ConfigureWebHostDefaults(webBuilder =>
22                 {
23                     webBuilder.UseStartup<Startup>();
24                 });
25     }
26 }
27
```

The 'Solution Explorer' pane shows the project structure with files like 'appsettings.json', 'Program.cs', and 'Startup.cs'.

Configuration further stripped down, removing Routing and Endpoint references from initial "Empty" template.

```
13  public class Startup
14  {
15      // This method gets called by the runtime. Use this method to add services to the container.
16      // For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?LinkID=398940
17      public void ConfigureServices(IServiceCollection services)
18      {
19      }
20
21      // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
22      public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
23      {
24          app.Run(async context =>
25          {
26              await context.Response.WriteAsync("Hello World!");
27          });
28      }
29  }
```

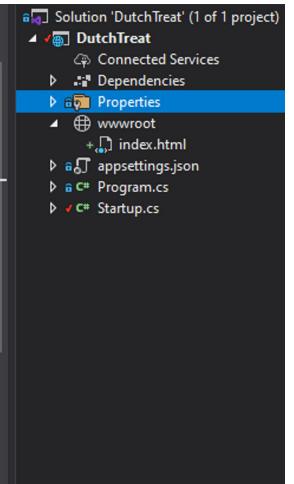
Running from console and accessing in browser.



Hello World!

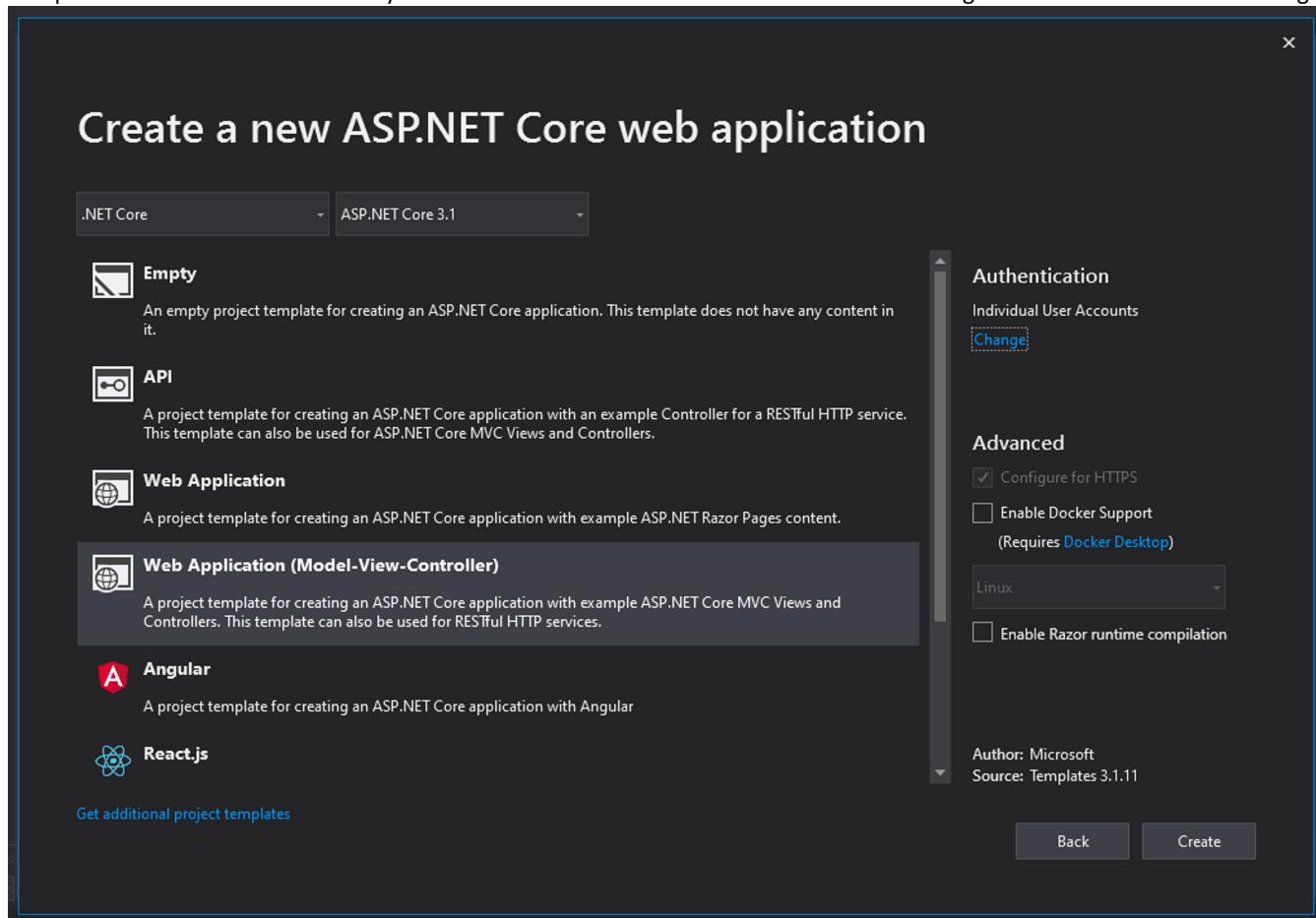
Order of logic in Configure method important for correctly initializing middleware. Here UseDefaultFiles detects the "index.html" file and changes its path to root at runtime, so when UseStaticFiles is run it picks up that route and serves it as the landing page. If UseStaticFiles came first, the path wouldn't have changed yet, so changing it afterward would not result in it being served once UseStaticFiles has already been executed.

```
11  namespace DutchTreat
12  {
13      public class Startup
14      {
15          // This method gets called by the runtime. Use this method to add services to the container.
16          // For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?LinkID=398940
17          public void ConfigureServices(IServiceCollection services)
18          {
19          }
20
21          // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
22          public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
23          {
24              /*---- Organizing Order of Middleware for Application ----*/
25
26              // Detects "index.html" as a default file for landing page and serves it as root.
27              app.UseDefaultFiles();
28
29              // Static file serving, but only from wwwroot directory - which had to be created.
30              app.UseStaticFiles();
31          }
32      }
33  }
```

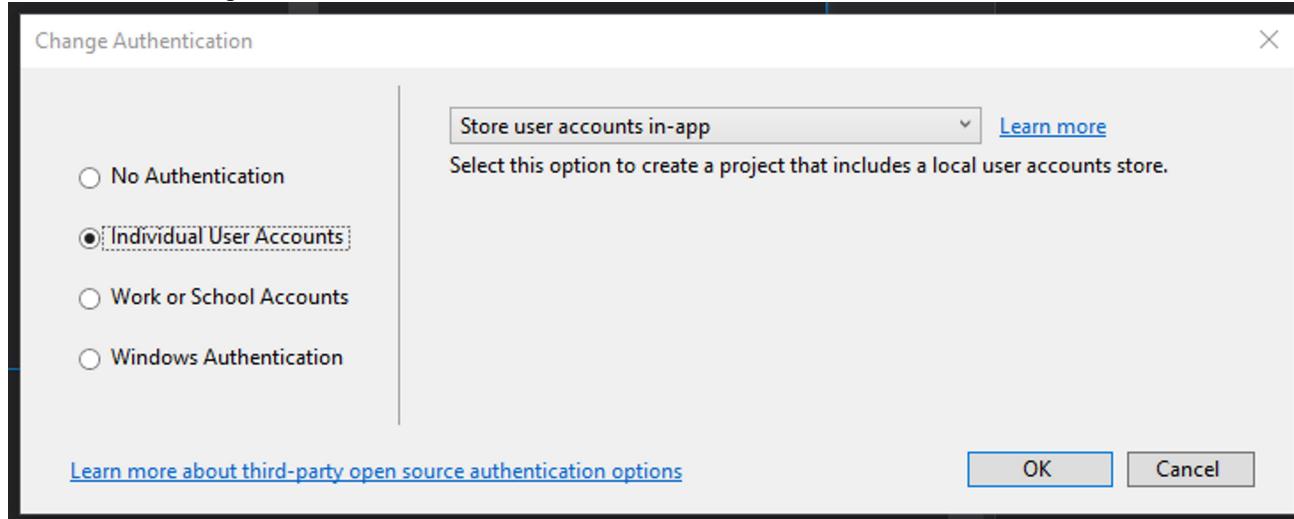


## MVC Web App Template

Template with MVC and user identity stuff included. Note: "Individual User Accounts" setting under "Authentication" is being used.



Authentication configuration.



## Console Project Creation (Hello World)

### Console Commands:

dotnet	Executable
dotnet new console	Creates console project within current directory.
dotnet restore	Downloads dependencies.
dotnet build	Build a version of the current dotnet project.
dotnet run	Execute current dotnet project. Builds in background if not built.
dotnet add package [name]	Add dependency package to current project.

Setting up "Hello World" project directly from console.

```
Scott@Mini-Tower MINGW64 ~/source/repos/WebApp_Tutorial/HelloWorld (main)
$ dotnet new console
The template "Console Application" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on C:\Users\Scott\source\repos\WebApp_Tutorial\HelloWorld\HelloWorld.csproj...
Determining projects to restore...
Restored C:\Users\Scott\source\repos\WebApp_Tutorial\HelloWorld\HelloWorld.csproj (in 46 ms).
Restore succeeded.

Scott@Mini-Tower MINGW64 ~/source/repos/WebApp_Tutorial/HelloWorld (main)
$ ls
HelloWorld.csproj  Program.cs  bin/  obj/

Scott@Mini-Tower MINGW64 ~/source/repos/WebApp_Tutorial/HelloWorld (main)
$ dotnet restore
Determining projects to restore...
All projects are up-to-date for restore.

Scott@Mini-Tower MINGW64 ~/source/repos/WebApp_Tutorial/HelloWorld (main)
$ dotnet build
Microsoft (R) Build Engine version 16.8.0+126527ff1 for .NET
Copyright (C) Microsoft Corporation. All rights reserved.

Determining projects to restore...
All projects are up-to-date for restore.
HelloWorld -> C:\Users\Scott\source\repos\WebApp_Tutorial\HelloWorld\bin\Debug\net5.0\HelloWorld.dll

Build succeeded.
0 Warning(s)
0 Error(s)

Time Elapsed 00:00:01.40

Scott@Mini-Tower MINGW64 ~/source/repos/WebApp_Tutorial/HelloWorld (main)
$ dotnet run
Hello World!
```

Visual Studio used only as a text editor to show what was generated.

