Setting up the Solution

Gill Cleeren

ARCHITECT

@gillcleeren www.snowball.be

Overview

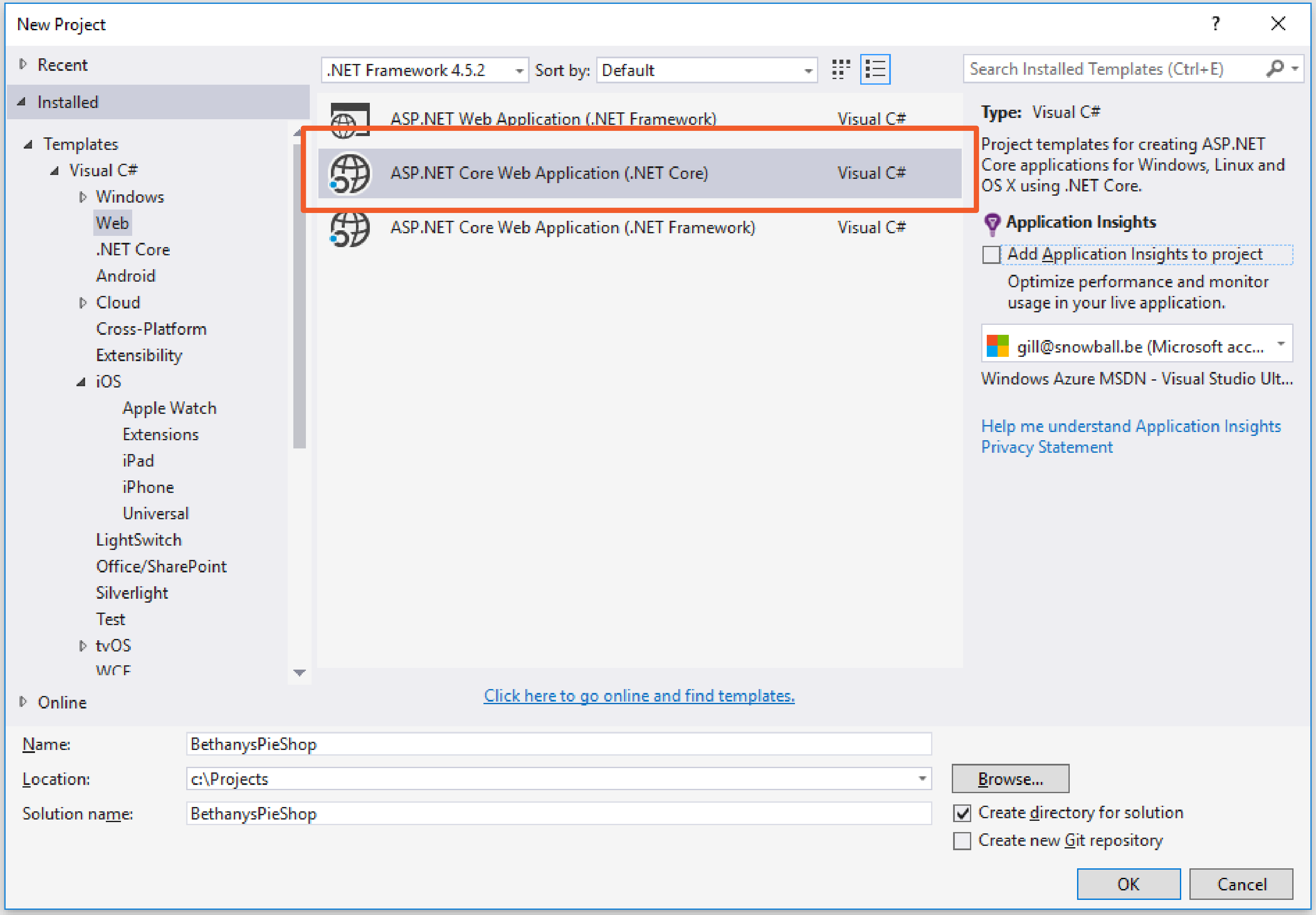
File → New Project

Looking at the project structure

Configuring the application

File → New Project

Project Creation



ASP.NET Web Application (.NET Framework) Visual C#

ASP.NET Core Web Application (.NET Core) Visual C#

ASP.NET Core Web Application (.NET Framework) Visual C#

Type: Visual C#

Project templates for creating ASP.NET Core applications for Windows, Linux and OS X using .NET Core.

Application Insights

□ Add Application Insights to project

Optimize performance and monitor usage in your live application.

gill@snowball.be (Microsoft acc...

Windows Azure MSDN - Visual Studio Ult...

Help me understand Application Insights

Privacy Statement

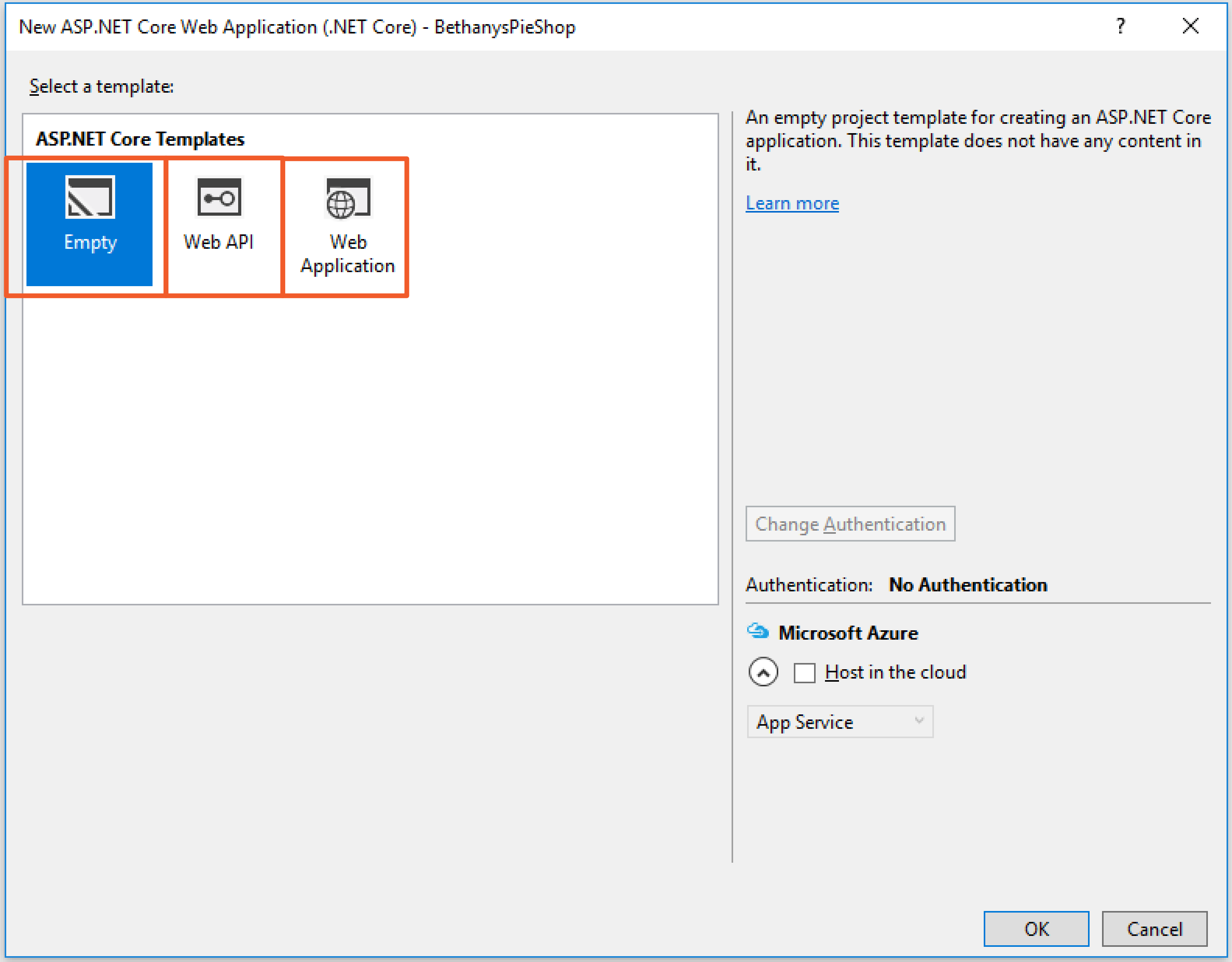
Name: BethanysPieShop

Location: c:\Projects

Solution name: BethanysPieShop

☑ Create directory for solution

□ Create new Git repository



New ASP.NET Core Web Application (.NET Core) - BethanysPieShop

Select a template:

ASP.NET Core Templates

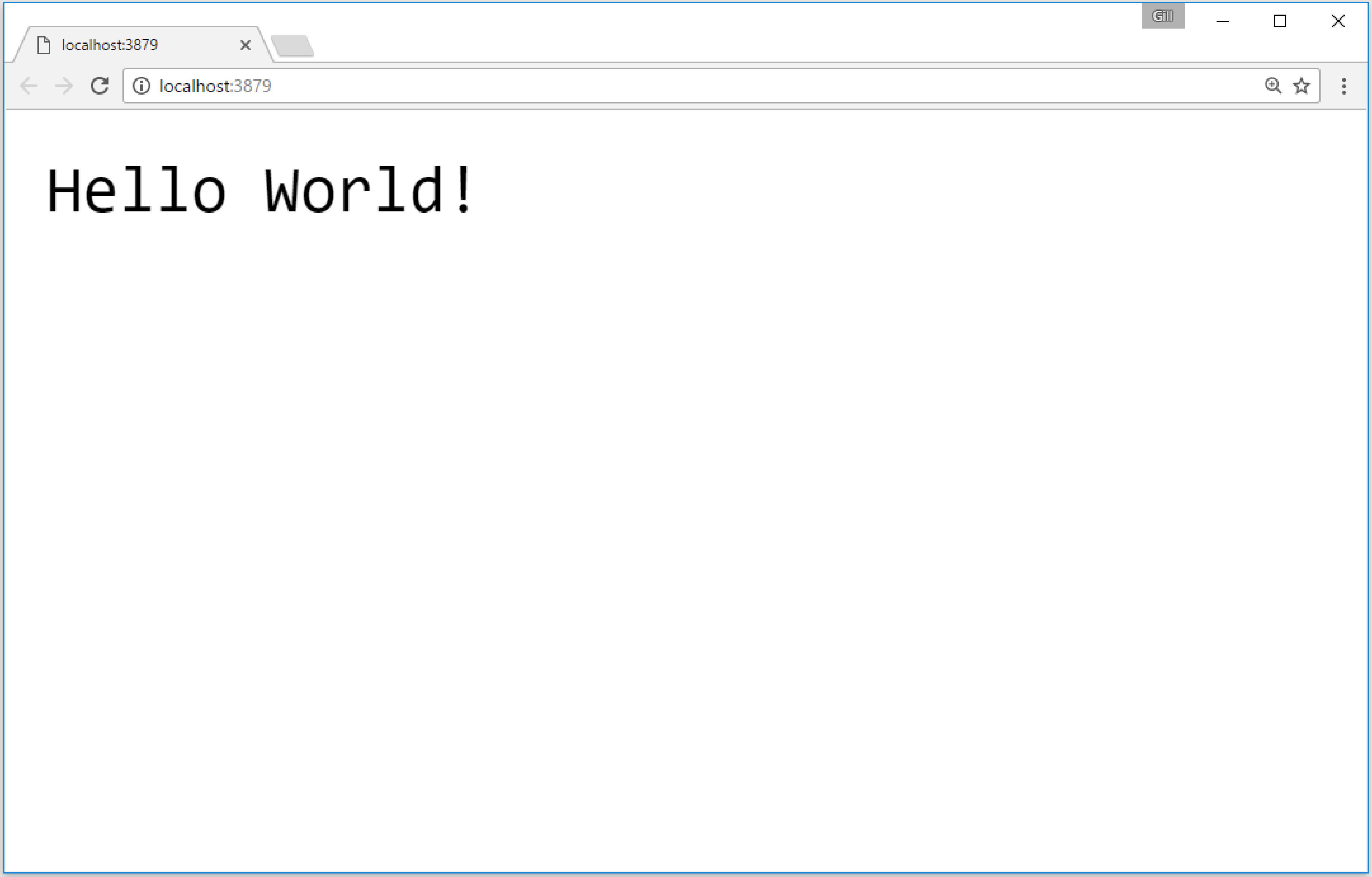
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

Authentication: **No Authentication**

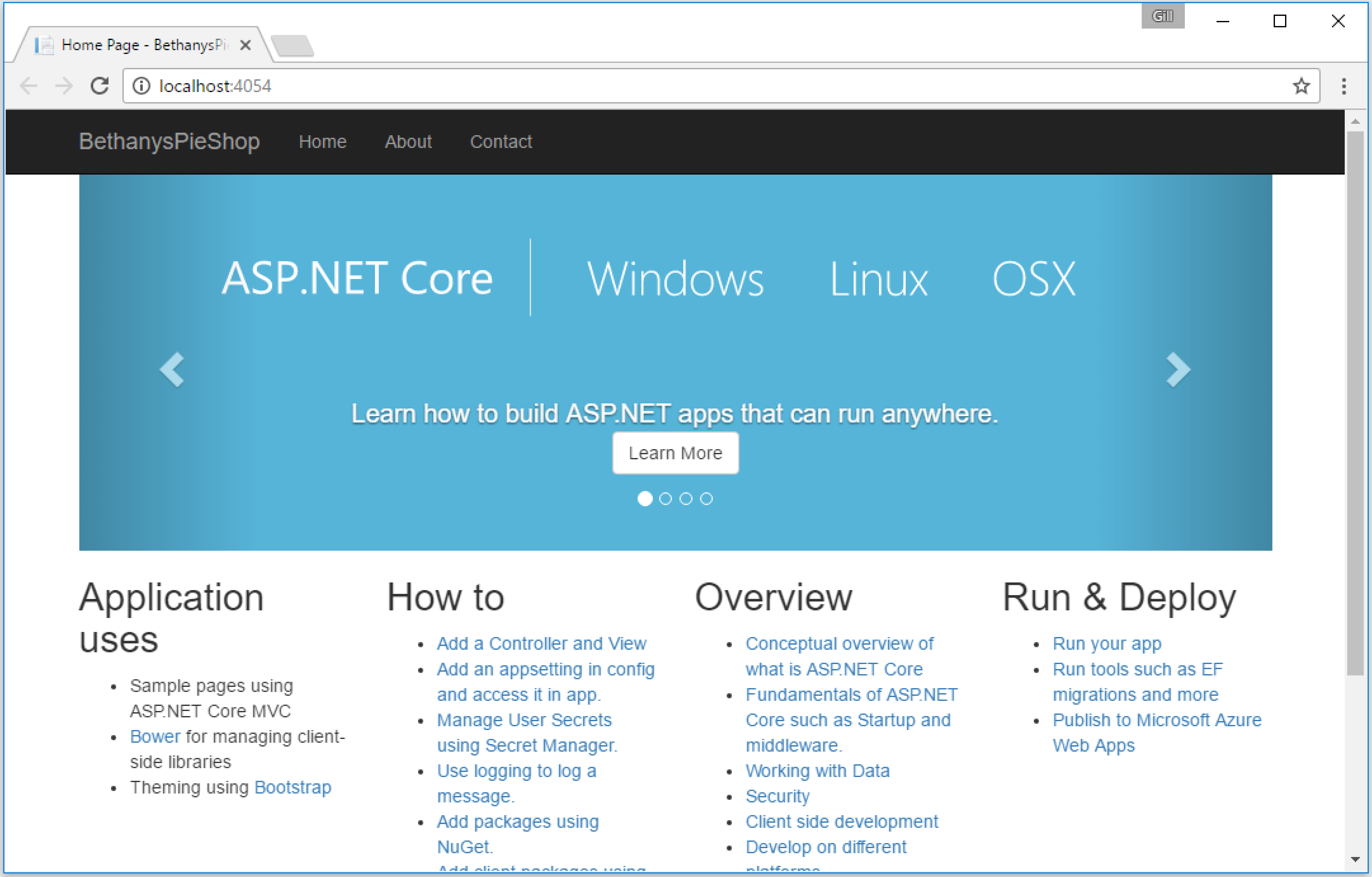
Microsoft Azure

□ Host in the cloud

App Service



Hello World!



Learn how to build ASP.NET apps that can run anywhere.

Application uses

- Sample pages using ASP.NET Core MVC
- Bower for managing client-side libraries
- Theming using Bootstrap

How to

- Add a Controller and View
- Add an appsetting in config and access it in app.
- Manage User Secrets using Secret Manager.
- Use logging to log a message.
- Add packages using NuGet.
- Add client packages using

Overview

- Conceptual overview of what is ASP.NET Core
- Fundamentals of ASP.NET Core such as Startup and middleware.
- Working with Data
- Security
- Client side development
- Develop on different platforms

Run & Deploy

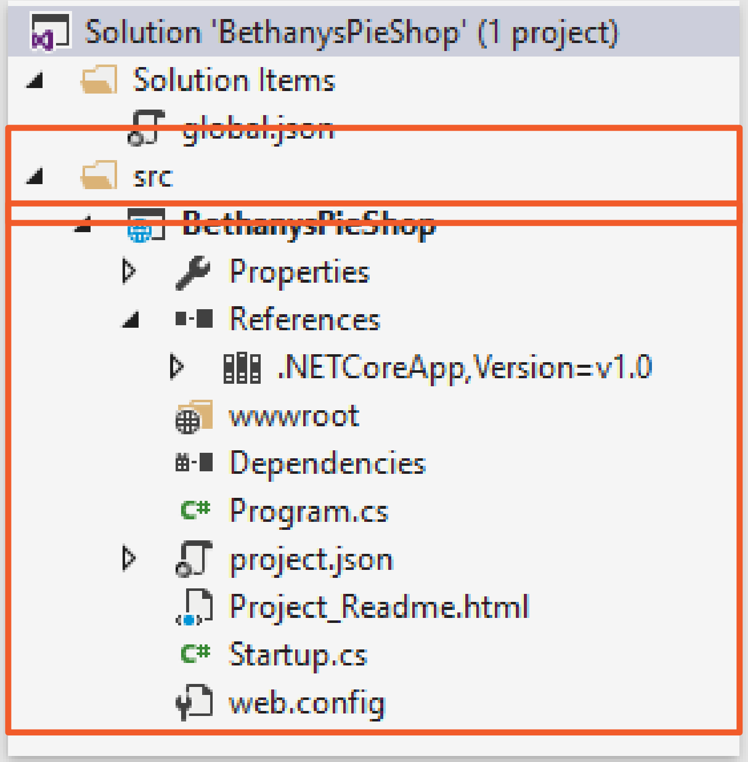
- Run your app
- Run tools such as EF migrations and more
- Publish to Microsoft Azure Web Apps

Demo

Creating the new solution

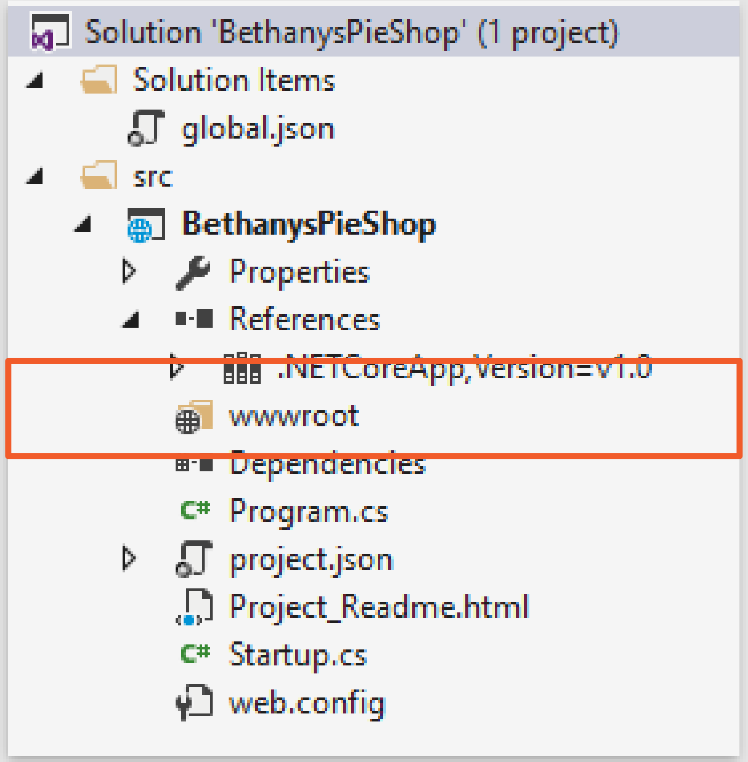
Looking at the Project Structure

Project Structure



Solution 'BethanysPieShop' (1 project)
- Solution Items
  - global.json
- src
  - BethanysPieShop
    - Properties
    - References
      - .NETCoreApp,Version=v1.0
    - wwwroot
    - Dependencies
    - Program.cs
    - project.json
    - Project_Readme.html
    - Startup.cs
    - web.config

The wwwroot Folder

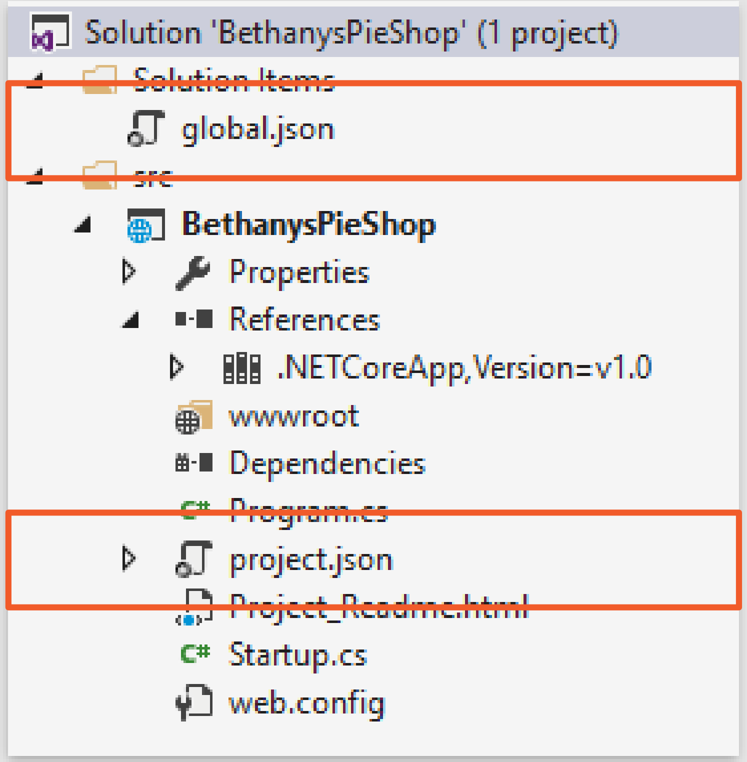


Solution 'BethanysPieShop' (1 project)
- Solution Items
  - global.json
- src
  - BethanysPieShop
    - Properties
    - References
      - .NETCoreApp,Version=v1.0
    - wwwroot
    - Dependencies
    - Program.cs
    - project.json
    - Project_Readme.html
    - Startup.cs
    - web.config

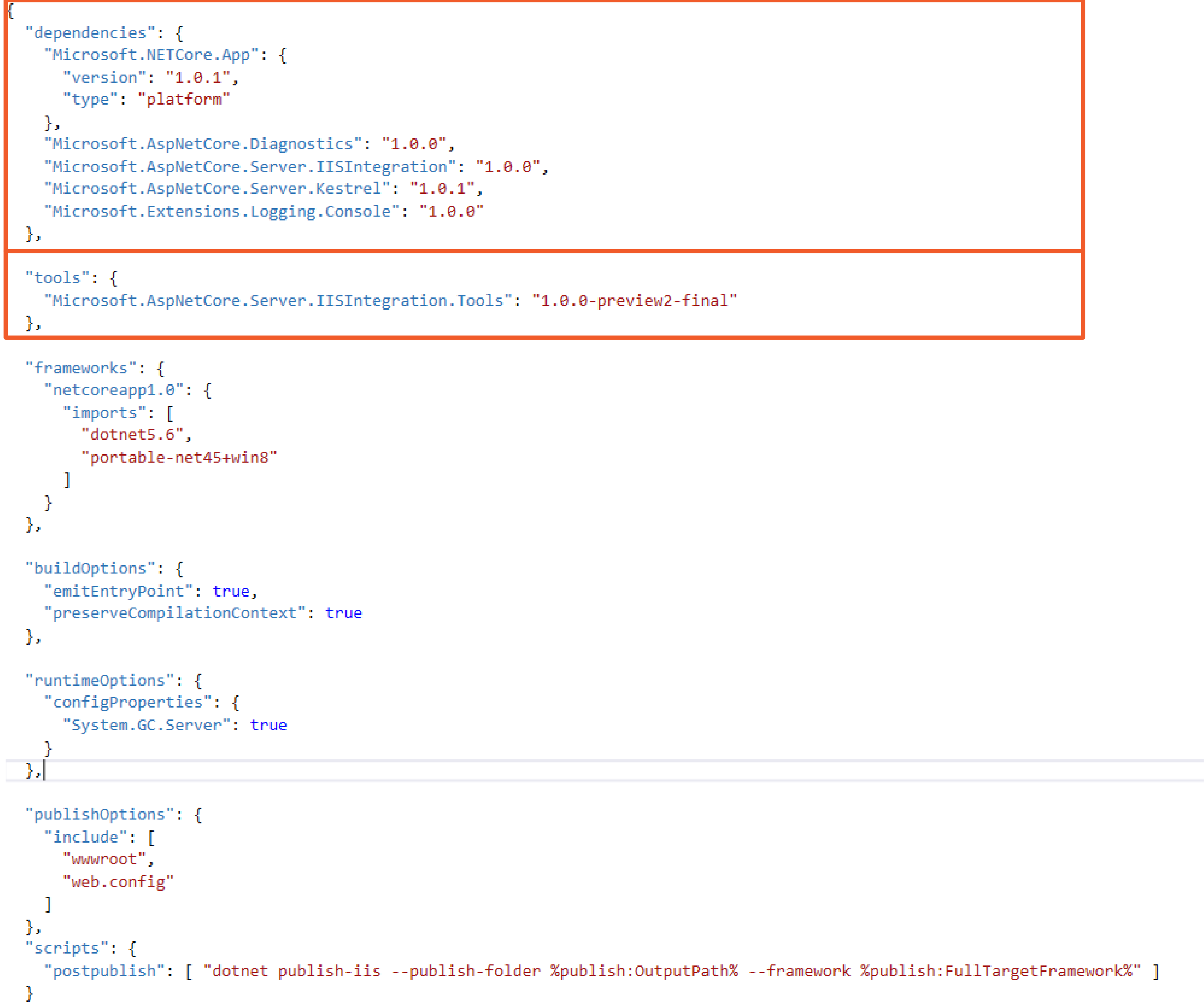
JSON Configuration

global.json

project.json



Solution 'BethanysPieShop' (1 project)
- Solution Items
  - global.json
- src
  - BethanysPieShop
    - Properties
    - References
      - .NETCoreApp,Version=v1.0
    - wwwroot
    - Dependencies
    - Program.cs
    - project.json
    - Project_Readme.html
    - Startup.cs
    - web.config



```
{
  "dependencies": {
    "Microsoft.NETCore.App": {
      "version": "1.0.1",
      "type": "platform"
    },
    "Microsoft.AspNetCore.Diagnostics": "1.0.0",
    "Microsoft.AspNetCore.Server.IISIntegration": "1.0.0",
    "Microsoft.AspNetCore.Server.Kestrel": "1.0.1",
    "Microsoft.Extensions.Logging.Console": "1.0.0"
  },

  "tools": {
    "Microsoft.AspNetCore.Server.IISIntegration.Tools": "1.0.0-preview2-final"
  },

  "frameworks": {
    "netcoreapp1.0": {
      "imports": [
        "dotnet5.6",
        "portable-net45+win8"
      ]
    }
  },

  "buildOptions": {
    "emitEntryPoint": true,
    "preserveCompilationContext": true
  },

  "runtimeOptions": {
    "configProperties": {
      "System.GC.Server": true
    }
  },

  "publishOptions": {
    "include": [
      "wwwroot",
      "web.config"
    ]
  },
  "scripts": {
    "postpublish": [ "dotnet publish-iis --publish-folder %publish:OutputPath% --framework %publish:FullTargetFramework%" ]
  }
```

Adding NuGet Packages

```
"dependencies": {
    "Microsoft.NETCore.App": {
      "version": "1.0.1",
      "type": "platform"
    },
    "Microsoft.AspNetCore.Diagnostics": "1.0.0",
    "Microsoft.AspNetCore.Server.IISIntegration": "1.0.0",
    "Microsoft.AspNetCore.Server.Kestrel": "1.0.1",
    "Microsoft.Extensions.Logging.Console": "1.0.0"




}
```

Adding NuGet Packages

```
"dependencies": {
    "Microsoft.NETCore.App": {
      "version": "1.0.1",
      "type": "platform"
    },
    "Microsoft.AspNetCore.Diagnostics": "1.0.0",
    "Microsoft.AspNetCore.Server.IISIntegration": "1.0.0",
    "Microsoft.AspNetCore.Server.Kestrel": "1.0.1",
    "Microsoft.Extensions.Logging.Console": "1.0.0"
    "Microsoft.AspNetCore.Razor.Tools": {
      "version": "1.0.0-preview2-final",
      "type": "build"
    },
    "Microsoft.AspNetCore.StaticFiles": "1.0.0",
    "Microsoft.AspNetCore.Mvc": "1.0.1"
}
```

Adding NuGet Packages

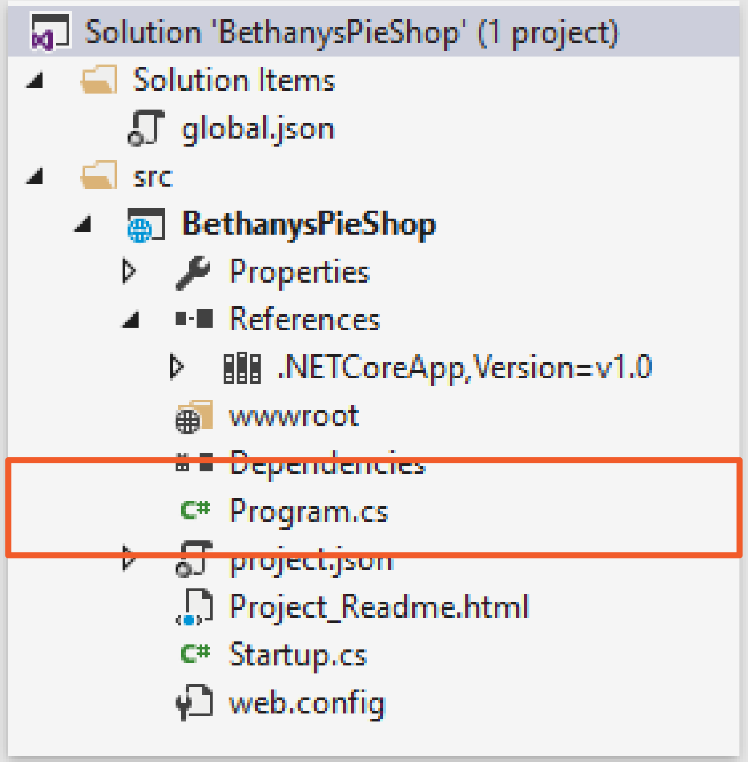
```
"tools": {
    "Microsoft.AspNetCore.Razor.Tools": "1.0.0-preview2-final",
    "Microsoft.AspNetCore.Server.IISIntegration.Tools": "1.0.0-preview2-final"
}
```

Demo

Exploring the generated project

Adding the correct NuGet packages

Program.cs



Solution 'BethanysPieShop' (1 project)
- Solution Items
  - global.json
- src
  - BethanysPieShop
    - Properties
    - References
      - .NETCoreApp,Version=v1.0
    - wwwroot
    - Dependencies
    - Program.cs
    - project.json
    - Project_Readme.html
    - Startup.cs
    - web.config

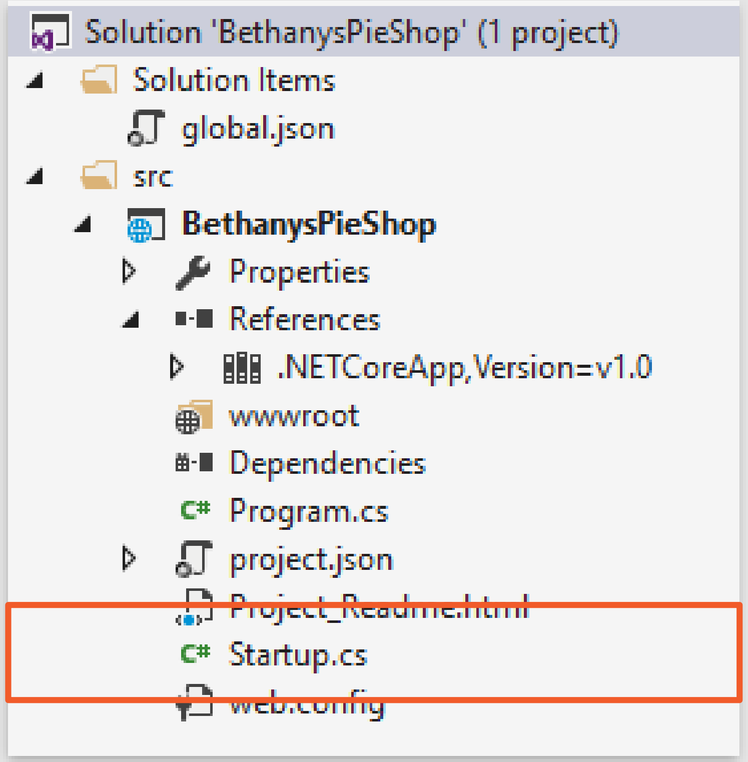
Program.cs

```
public class Program
{
    public static void Main(string[] args)
    {
        var host = new WebHostBuilder()
            .UseKestrel()
            .UseContentRoot(Directory.GetCurrentDirectory())
            .UseIISIntegration()
            .UseStartup<Startup>()
            .Build();

        host.Run();
    }
}
```

Configuring the Application

Startup.cs



Solution 'BethanysPieShop' (1 project)
Solution Items
global.json
src
BethanysPieShop
Properties
References
.NETCoreApp,Version=v1.0
wwwroot
Dependencies
Program.cs
project.json
Project_Readme.html
Startup.cs
web.config

Startup.cs

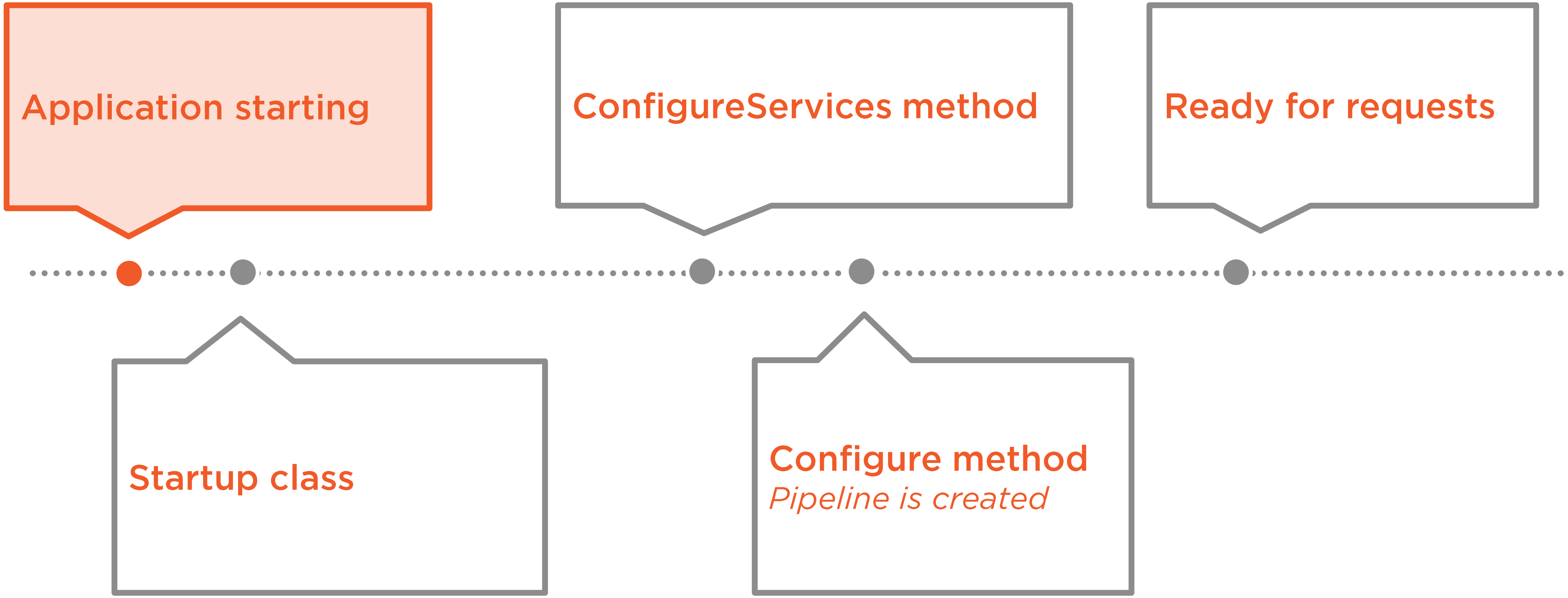
```
public class Startup
{
    public void ConfigureServices(IServiceCollection services) { }

    public void Configure(IApplicationBuilder app, IHostingEnvironment env, ILoggerFactory loggerFactory)
    {
        loggerFactory.AddConsole();

        if (env.IsDevelopment())
          app.UseDeveloperExceptionPage();

        app.Run(async (context) =>
             { await context.Response.WriteAsync("Hello World!"); });
    }
}
```

The Startup of the Application



Application starting

Startup class

ConfigureServices method

Configure method

Pipeline is created

Ready for requests

Configuring Our Application

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddMvc();
}

public void Configure(IApplicationBuilder app, ...)
{
    app.UseDeveloperExceptionPage();
    app.UseStatusCodePages();
    app.UseStaticFiles();
    app.UseMvcWithDefaultRoute();
}
```

Demo

Configuring the application

Summary

Modular configuration

New project structure