

How-To Get started with C# and ASP.NET Core in Visual Studio

In this tutorial for C# development with ASP.NET Core using Visual Studio, you'll create a C# ASP.NET Core web app, make changes to it, explore some features of the IDE, and then run the app.

Prerequisites

1. Install Visual Studio

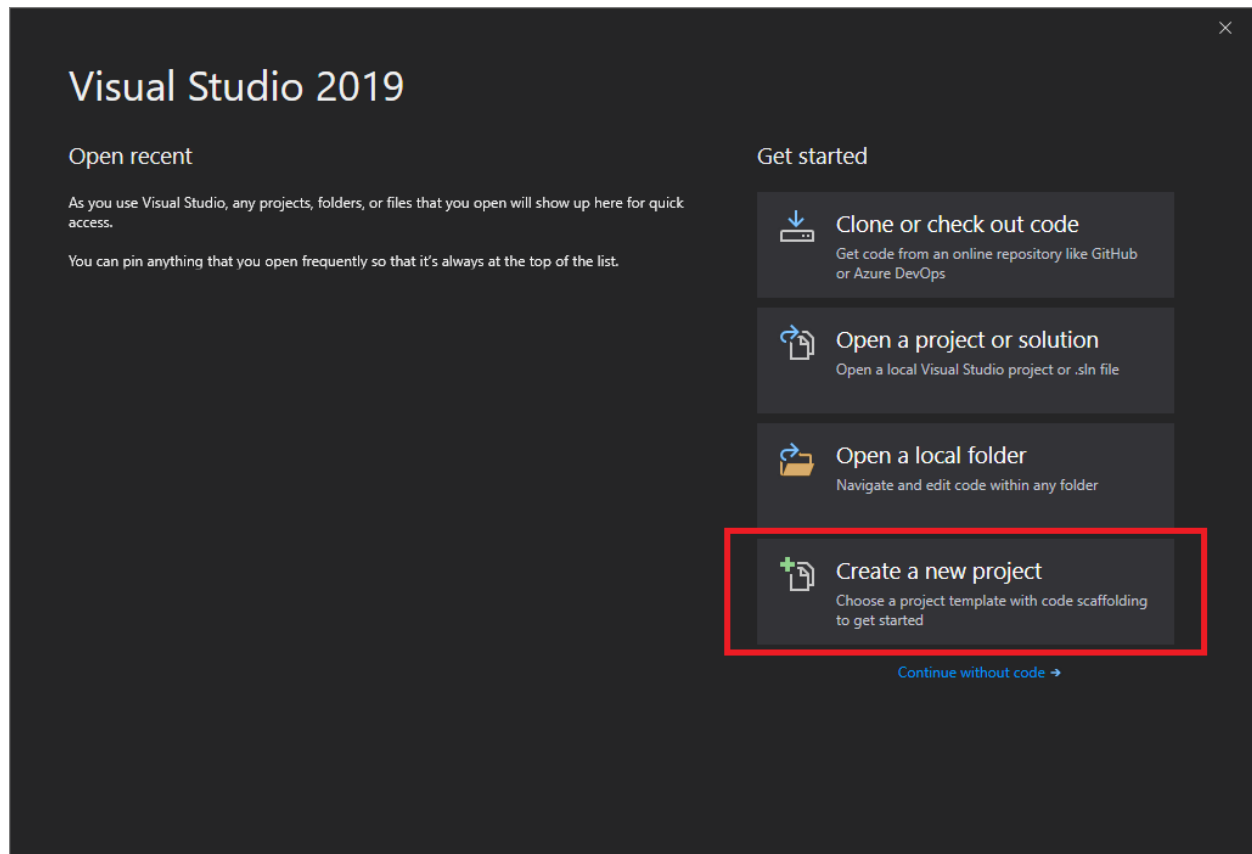
If you haven't already installed Visual Studio, go to the [Visual Studio downloads](#) page to install it for free.

2. Update Visual Studio - If you've already installed Visual Studio, make sure that you're running the most recent release. For more information about how to update your installation, see the [Update Visual Studio to the most recent release](#) page.
3. Choose your theme (optional) - This tutorial includes screenshots that use the dark theme. You can [Personalize the Visual Studio IDE and Editor](#) page to learn how.

Create a project

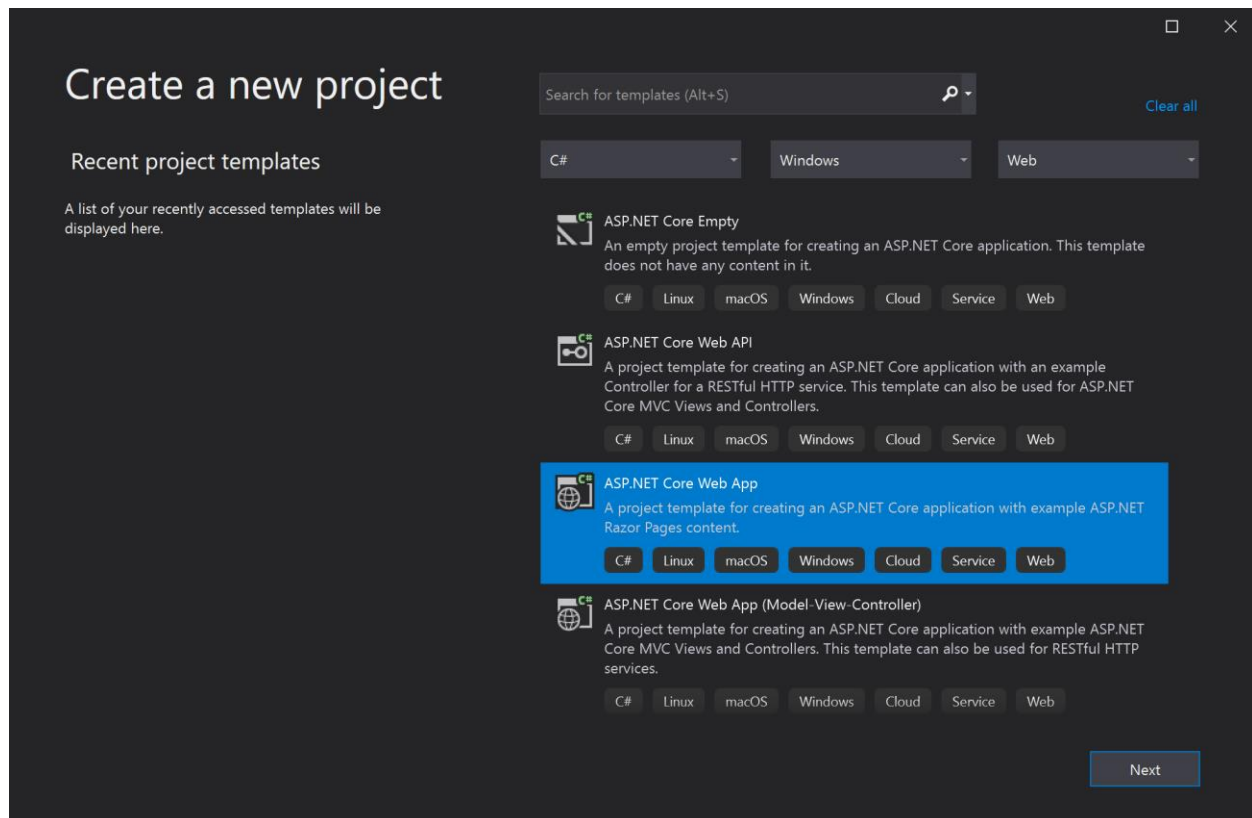
First, you'll create a ASP.NET Core project. The project type comes with all the template files you'll need for a fully functional website, before you've even added anything!

1. In the start window, choose **Create a new project**.



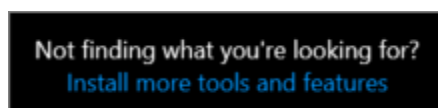
2. In the **Create a new project** window, choose **C#** from the Language list. Next, choose **Windows** from the Platform list, and **Web** from the project types list.

After you apply the language, platform, and project type filters, choose the **ASP.NET Core Web App** template, and then choose **Next**.

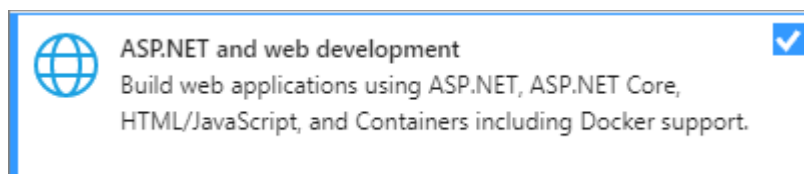


Note

If you don't see the **ASP.NET Core Web App** template, you can install it from the **Create a new project** window. In the **Not finding what you're looking for?** message, choose the **Install more tools and features** link.



Then, in the Visual Studio Installer, choose the **ASP.NET and web development** workload.



After that, choose the **Modify** button in the Visual Studio Installer. If you're prompted to save your work, do so. Next, choose **Continue** to install the workload. Then, return to step 2 in this "[Create a project](#)" procedure.

3. In the **Configure your new project** window, type or enter *MyCoreApp* in the **Project name** box. Then, choose **Next**.

Configure your new project

ASP.NET Core Web App C# Linux macOS Windows Cloud Service Web

Project name
MyCoreApp

Location
C:\Users\jornella\source\repos

Solution
Create new solution

Solution name ⓘ
MyCoreApp

☐ Place solution and project in the same directory

Back Next

4. In the **Additional information** window, verify that **.NET Core 3.1** appears in the top drop-down menu. Note that you can choose to enable Docker support by checking the box. You can also add authentication support by clicking the change Authentication button. From there you can choose from:
 - None: no authentication.
 - Individual accounts: these are stored in a local or Azure-based database.
 - Microsoft identity platform: this option uses Active Directory, Azure AD, or Microsoft 365 for authentication.
 - Windows: suitable for intranet applications.

Leave the **Enable Docker** box unchecked, and select **None** for Authentication Type. Then, select **Create**.

Additional information

ASP.NET Core Web App C# Linux macOS Windows Cloud Service Web

Target Framework

.NET Core 3.1 (Long-term support)

Authentication Type

None

☒ Configure for HTTPS

☐ Enable Docker

Docker OS

Linux

☐ Enable Razor runtime compilation

Back Create

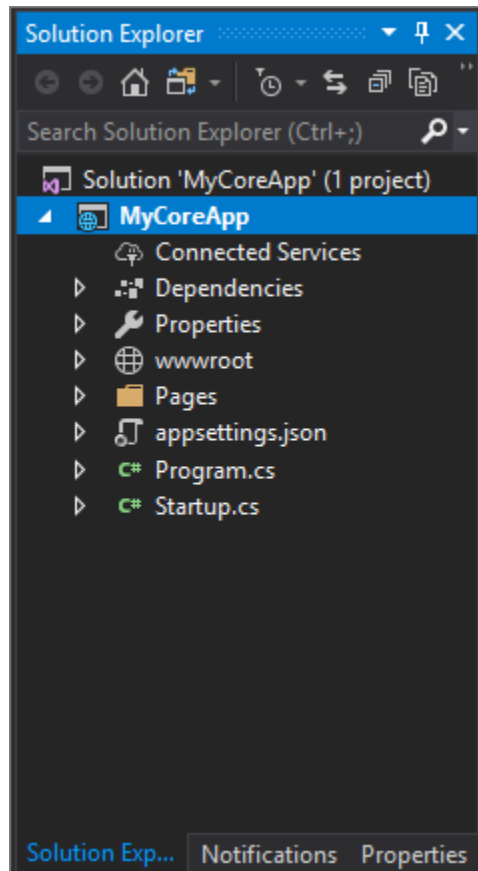
Visual Studio will open up your new project.

About your solution

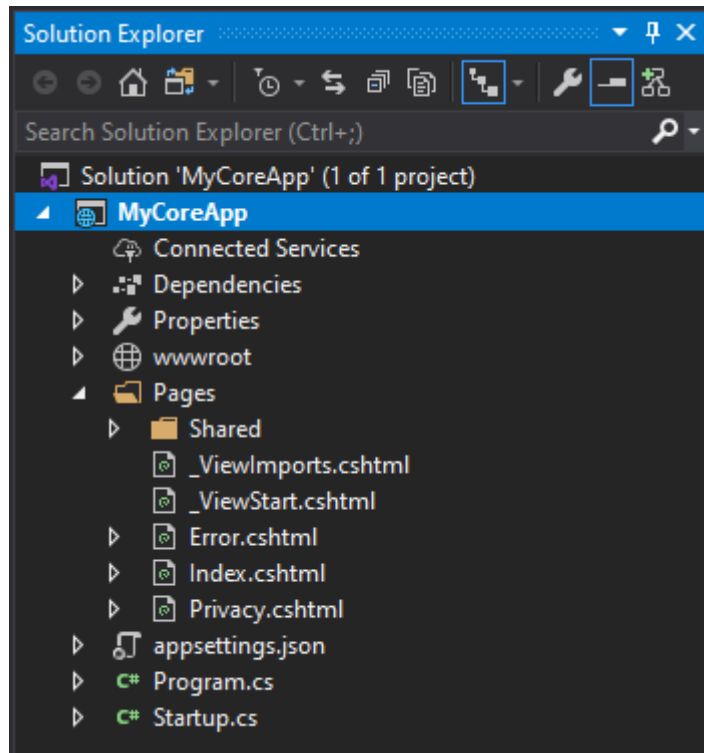
This solution follows the **Razor Page** design pattern. It's different than the [Model-View-Controller \(MVC\)](#) design pattern in that it's streamlined to include the model and controller code within the Razor Page itself.

Tour your solution

1. The project template creates a solution with a single ASP.NET Core project that is named *MyCoreApp*. Choose the **Solution Explorer** tab to view its contents.



2. Expand the **Pages** folder.

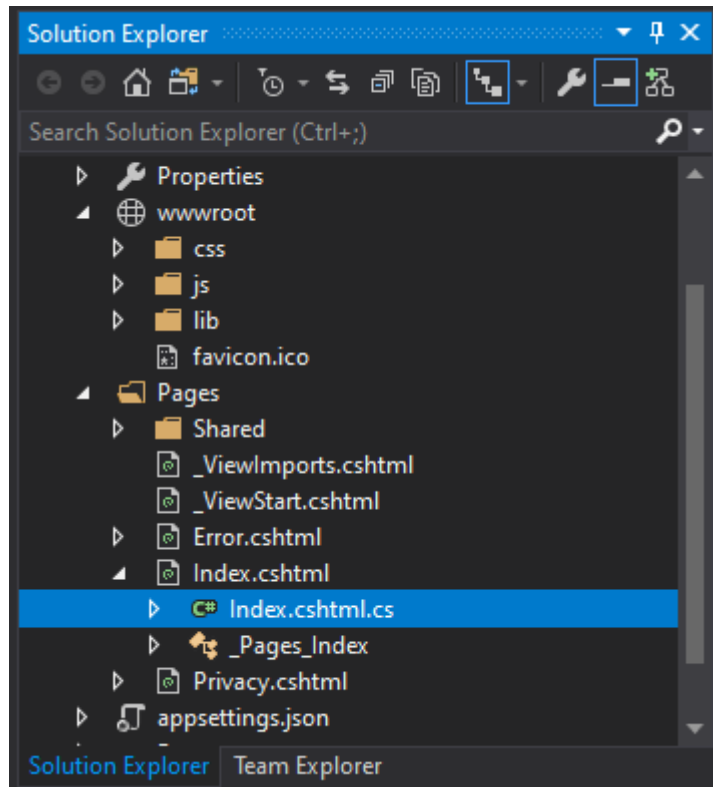


3. View the **Index.cshtml** file in the code editor.

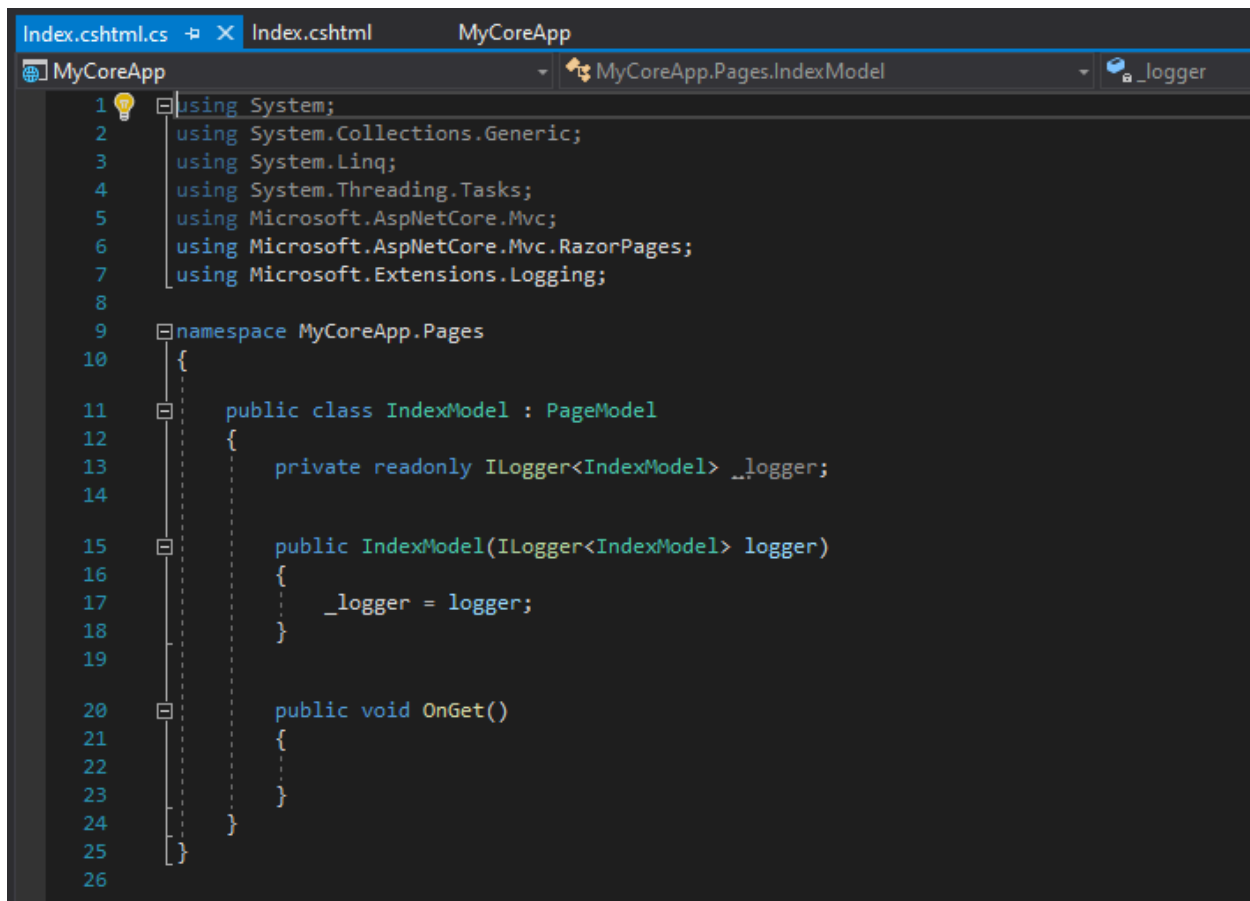
The image shows the code editor with the file 'Index.cshtml' open. The code is as follows:

```
1 @page
2 @model IndexModel
3 @*
4   ViewData["Title"] = "Home page";
5   *@
6
7 <div class="text-center">
8   <h1 class="display-4">Welcome</h1>
9   <p>Learn about <a href="https://docs.microsoft.com/aspnet/core">building Web apps with ASP.NET Core</a>.</p>
10 </div>
11
```

4. Each .cshtml file has an associated code file. To open the code file in the editor, expand the **Index.cshtml** node in Solution Explorer, and choose the **Index.cshtml.cs** file.

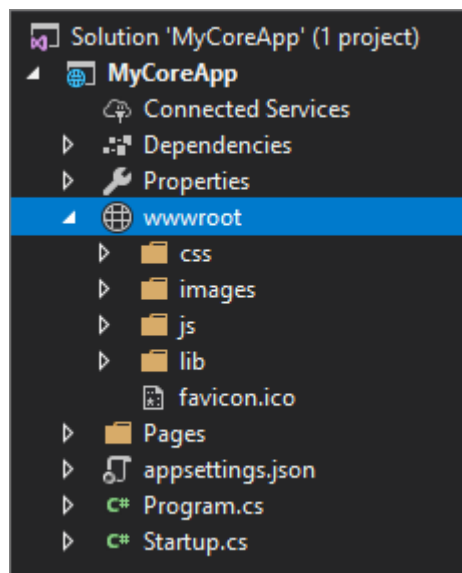


5. View the **Index.cshtml.cs** file in the code editor.



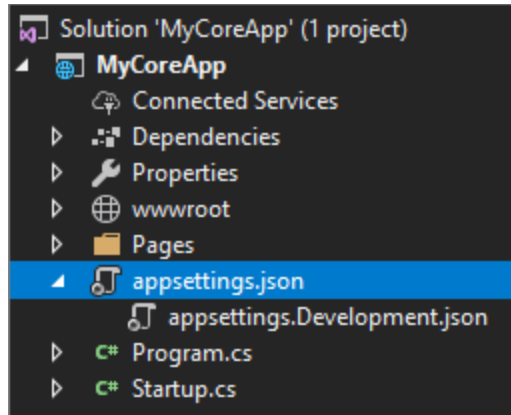
```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using Microsoft.AspNetCore.Mvc;
6 using Microsoft.AspNetCore.Mvc.RazorPages;
7 using Microsoft.Extensions.Logging;
8
9 namespace MyCoreApp.Pages
10 {
11     public class IndexModel : PageModel
12     {
13         private readonly ILogger<IndexModel> _logger;
14
15         public IndexModel(ILogger<IndexModel> logger)
16         {
17             _logger = logger;
18         }
19
20         public void OnGet()
21         {
22         }
23     }
24 }
25
26
```

6. The project contains a **wwwroot** folder that is the root for your website. Expand the folder to view its contents.



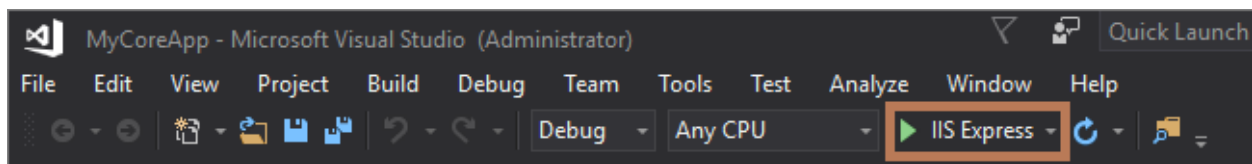
You can put static site content—such as CSS, images, and JavaScript libraries—directly in the paths where you want them.

7. The project also contains configuration files that manage the web app at run time. The default application [configuration](#) is stored in *appsettings.json*. However, you can override these settings by using *appsettings.Development.json*. Expand the **appsettings.json** file to view the **appsettings.Development.json** file.



Run, debug, and make changes

1. Choose the **IIS Express** button in the IDE to build and run the app in Debug mode. (Alternatively, press **F5**, or choose **Debug** > **Start Debugging** from the menu bar.)



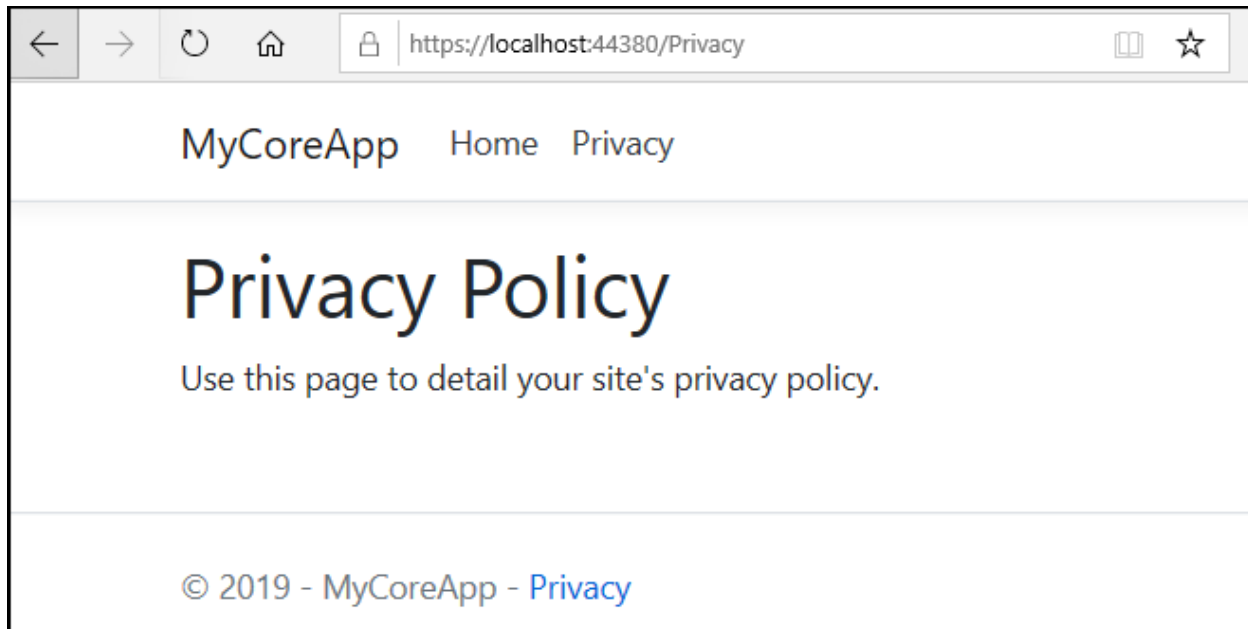
Note

If you get an error message that says **Unable to connect to web server 'IIS Express'**, close Visual Studio and then open it by using the **Run as administrator** option from the right-click or context menu. Then, run the application again.

You might also get a message that asks if you want to accept an IIS SSL Express certificate. To view the code in a web browser, choose **Yes**, and then choose **Yes** if you receive a follow-up security warning message.

2. Visual Studio launches a browser window. You should then see **Home**, and **Privacy** pages in the menu bar.
3. Choose **Privacy** from the menu bar.

The **Privacy** page in the browser renders the text that is set in the *Privacy.cshtml* file.

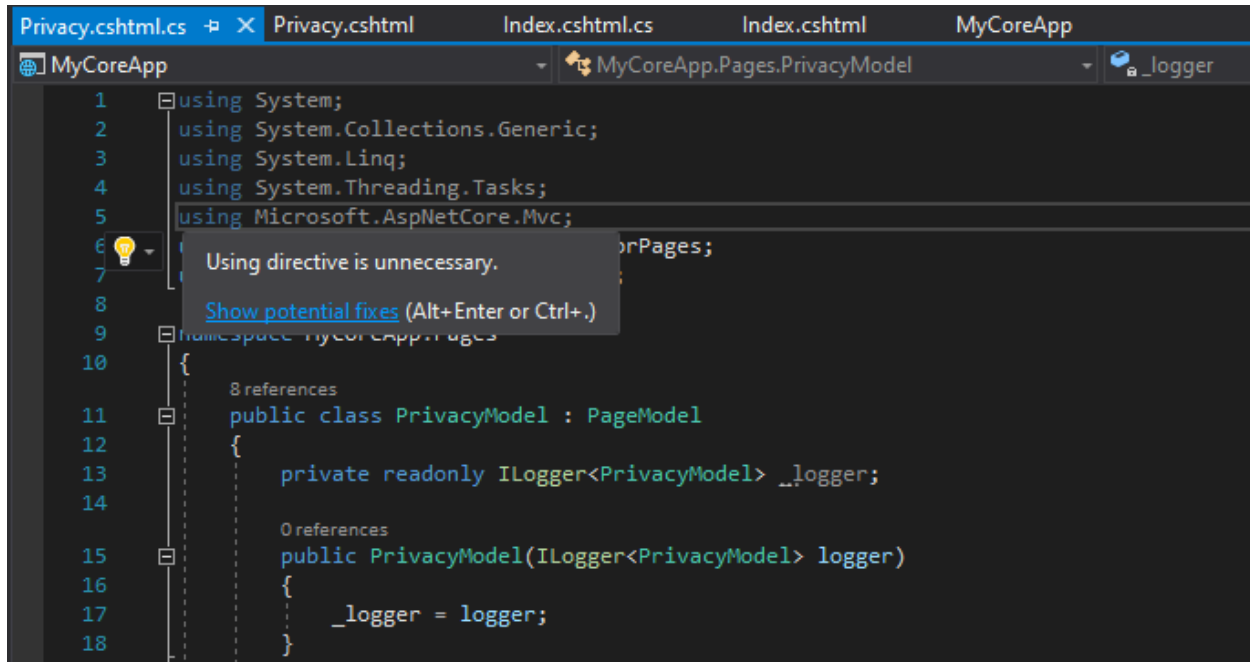


4. Return to Visual Studio, and then press **Shift+F5** to stop Debug mode. This also closes the project in the browser window.
5. In Visual Studio, open **Privacy.cshtml** for editing. Then, delete the words *Use this page to detail your site's privacy policy* and in its place, add the words *This page is under construction as of @ViewData["TimeStamp"]*.

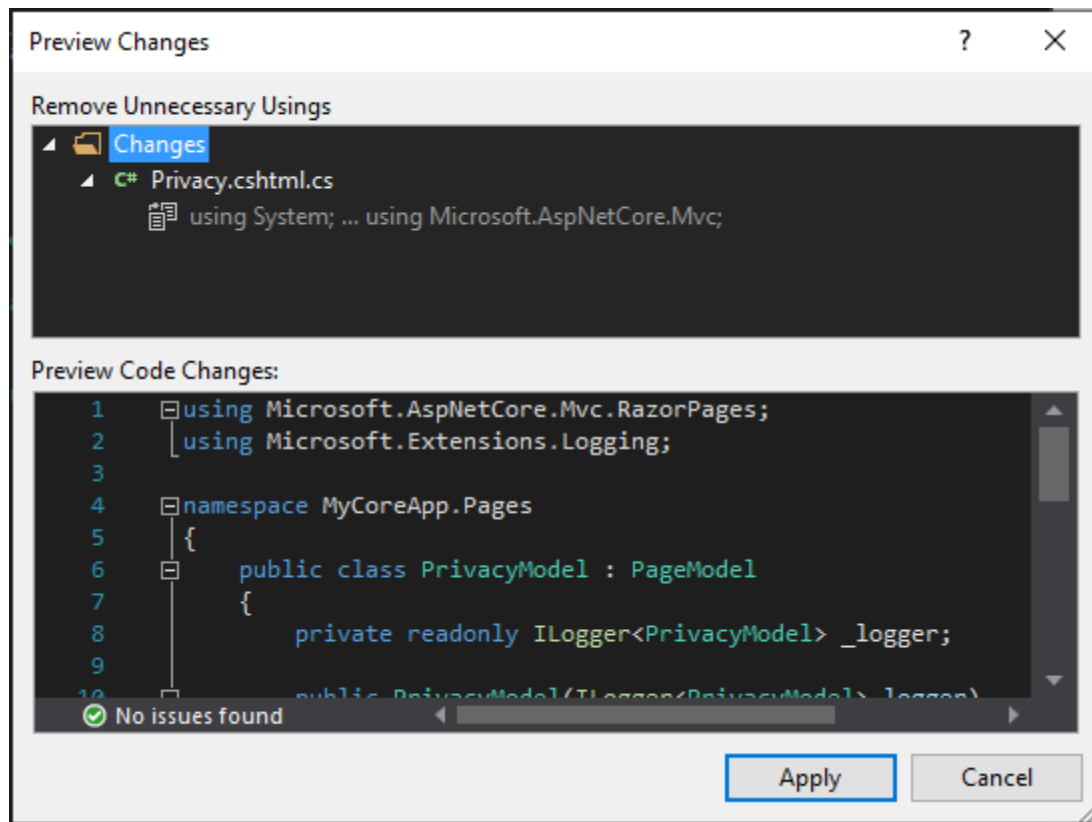
```
1 @page
2 @model PrivacyModel
3 @{
4     ViewData["Title"] = "Privacy Policy";
5 }
6 <h1>@ViewData["Title"]</h1>
7
8 <p>This page is under construction as of @ViewData["TimeStamp"].</p>
9
```

6. Now, let's make a code change. Choose **Privacy.cshtml.cs**. Then, clean up the using directives at the top of the file by using the following shortcut:

Choose any of the grayed-out using directives and a [Quick Actions](#) light bulb will appear just below the caret or in the left margin. Choose the light bulb, and then hover over **Remove unnecessary usings**.



Now choose **Preview changes** to see what will change.



Choose **Apply**. Visual Studio deletes the unnecessary using directives from the file.

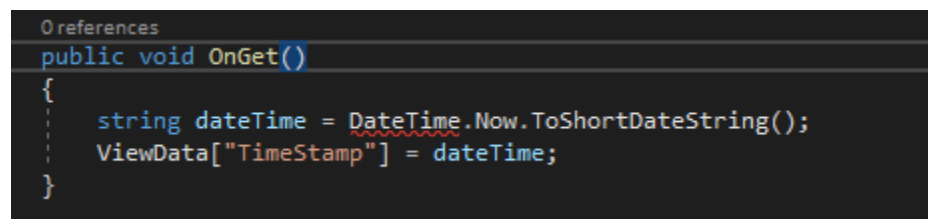
- Next, in the `OnGet()` method, change the body to the following code:

```

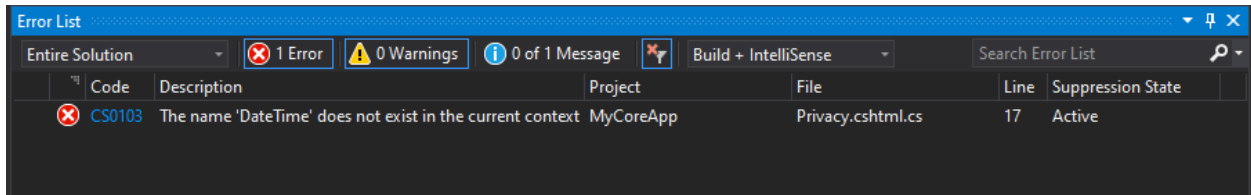
C#Copy
public void OnGet()
{
    string dateTime = DateTime.Now.ToShortDateString();
    ViewData["TimeStamp"] = dateTime;
}

```

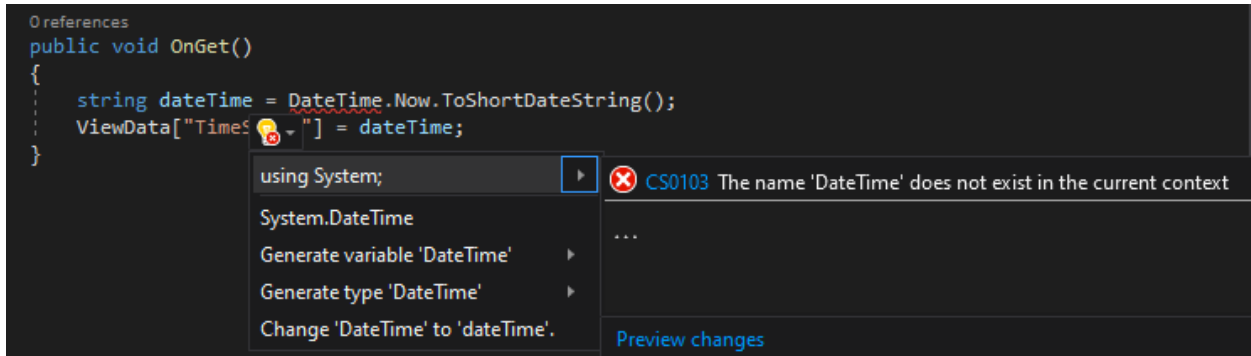
- Notice that two wavy underlines appear under **DateTime**. The wavy underlines appear because this type isn't in scope.



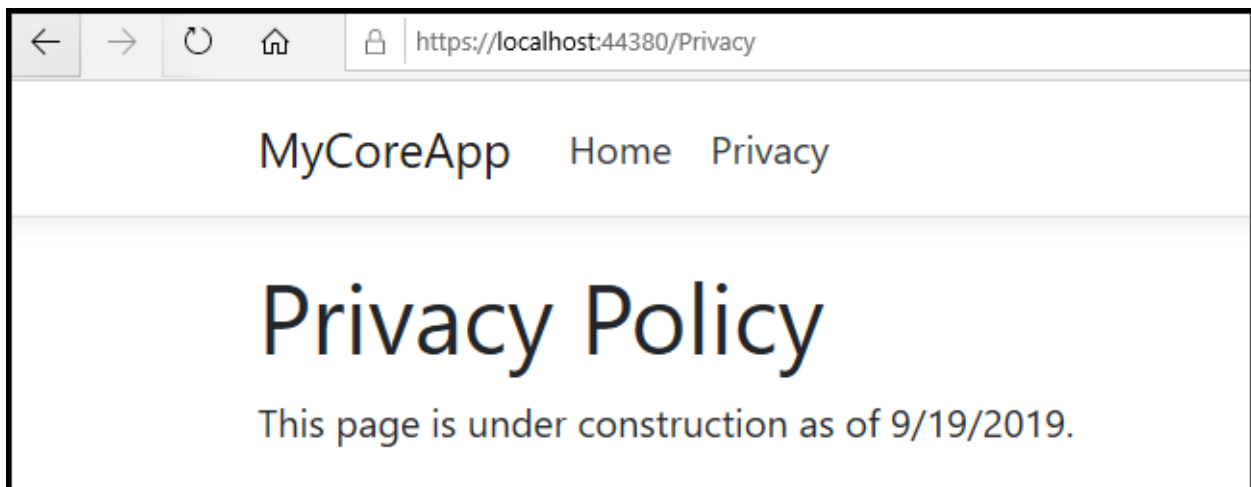
Open the **Error List** toolbar to see the same errors listed there. (If you don't see the **Error List** toolbar, choose **View > Error List** from the top menu bar.)



9. Let's fix this. In the code editor, place your cursor on either line that contains the error, and then choose the Quick Actions light bulb in the left margin. Then, from the drop-down menu, choose **using System;** to add this directive to the top of your file and resolve the errors.



10. Press **F5** to open your project in the web browser.
11. At the top of the web site, choose **Privacy** to view your changes.



12. Close the web browser, press **Shift+F5** to stop Debug mode, and then close Visual Studio.

Quick answers FAQ

Here's a quick FAQ to highlight some key concepts.

What is C#?

[C#](#) is a type-safe and object-oriented programming language that's designed to be both robust and easy to learn.

What is ASP.NET Core?

ASP.NET Core is an open-source and cross-platform framework for building internet-connected applications, such as web apps and services. ASP.NET Core apps can run on either .NET Core or the .NET Framework. You can develop and run your ASP.NET Core apps cross-platform on Windows, Mac, and Linux. ASP.NET Core is open source at [GitHub](#).

What is Visual Studio?

Visual Studio is an integrated development suite of productivity tools for developers. Think of it as a program you can use to create programs and applications.

Next steps

Congratulations on completing this tutorial! We hope you learned a little bit about C#, ASP.NET Core, and the Visual Studio IDE. To learn more about creating a web app or website with C# and ASP.NET, continue with other tutorials in this webpage:

<https://dokollolutionsinc.com/JournalsAspDotNet.html>

Courtesy: <https://docs.microsoft.com/en-us/visualstudio/get-started/csharp/tutorial-aspnet-core?view=vs-2019>

Modified: 2021.10.06.9.22.PM

Dököll Solutions, Inc.