Tutorial: Get started with Razor Pages in ASP.NET Core

09/15/2020 • 15 minutes to read • +10

In this article

By Rick Anderson

This is the first tutorial of a series that teaches the basics of building an ASP.NET Core Razor Pages web app.

For a more advanced introduction aimed at developers who are familiar with controllers and views, see Introduction to Razor Pages.

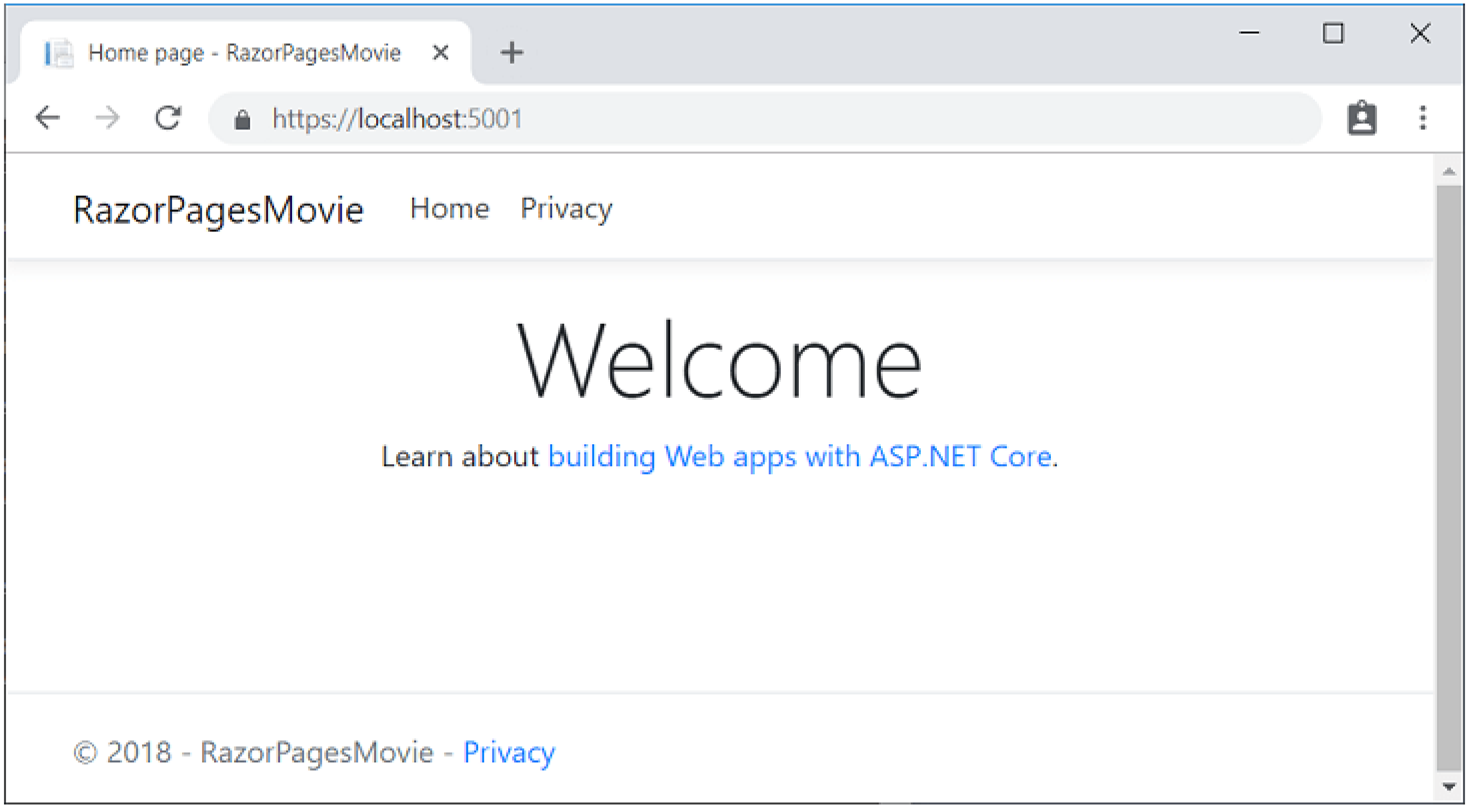
At the end of the series, you'll have an app that manages a database of movies.

View or download sample code (how to download).

In this tutorial, you:

- ✓ Create a Razor Pages web app.
- ✓ Run the app.
- ✓ Examine the project files.

At the end of this tutorial, you'll have a working Razor Pages web app that you'll build on in later tutorials.



Welcome

Learn about building Web apps with ASP.NET Core.

© 2018 - RazorPagesMovie - Privacy

Prerequisites

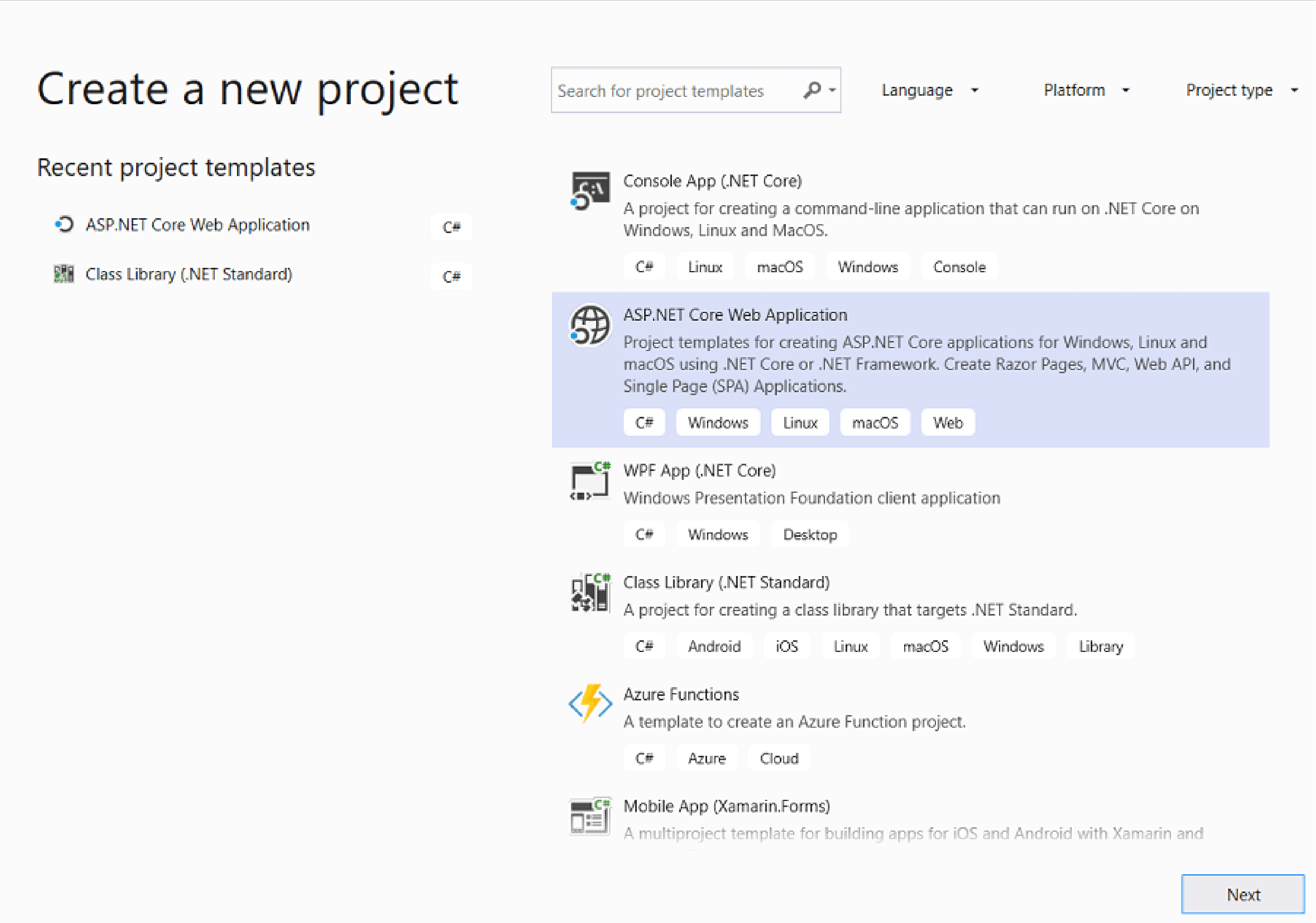
Visual Studio | Visual Studio Code | Visual Studio for Mac

- Visual Studio 2019 16.4 or later with the **ASP.NET and web development** workload
- .NET Core 3.1 SDK or later

Create a Razor Pages web app

Visual Studio | Visual Studio Code | Visual Studio for Mac

- From the Visual Studio **File** menu, select **New** > **Project**.
- Create a new ASP.NET Core Web Application and select **Next**.



Create a new project

Recent project templates

ASP.NET Core Web Application C#

Class Library (.NET Standard) C#

Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

C# Linux macOS Windows Console

ASP.NET Core Web Application

Project templates for creating ASP.NET Core applications for Windows, Linux and macOS using .NET Core or .NET Framework. Create Razor Pages, MVC, Web API, and Single Page (SPA) Applications.

C# Windows Linux macOS Web

WPF App (.NET Core)

Windows Presentation Foundation client application

C# Windows Desktop

Class Library (.NET Standard)

A project for creating a class library that targets .NET Standard.

C# Android iOS Linux macOS Windows Library

Azure Functions

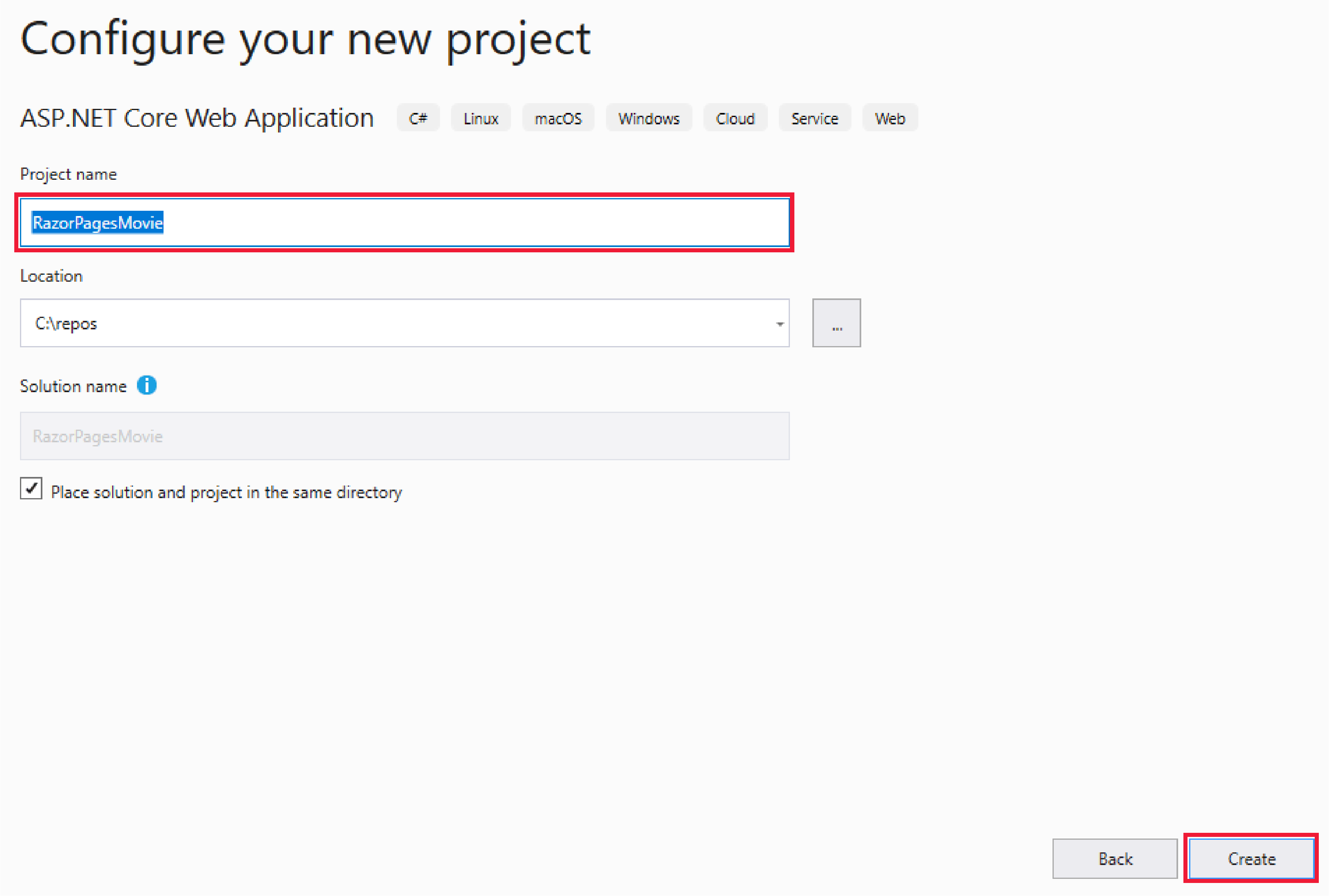
A template to create an Azure Function project.

C# Azure Cloud

Mobile App (Xamarin.Forms)

A multiproject template for building apps for iOS and Android with Xamarin and

- Name the project **RazorPagesMovie**. It's important to name the project *RazorPagesMovie* so the namespaces will match when you copy and paste code.



Configure your new project

ASP.NET Core Web Application

Project name

RazorPagesMovie

Location

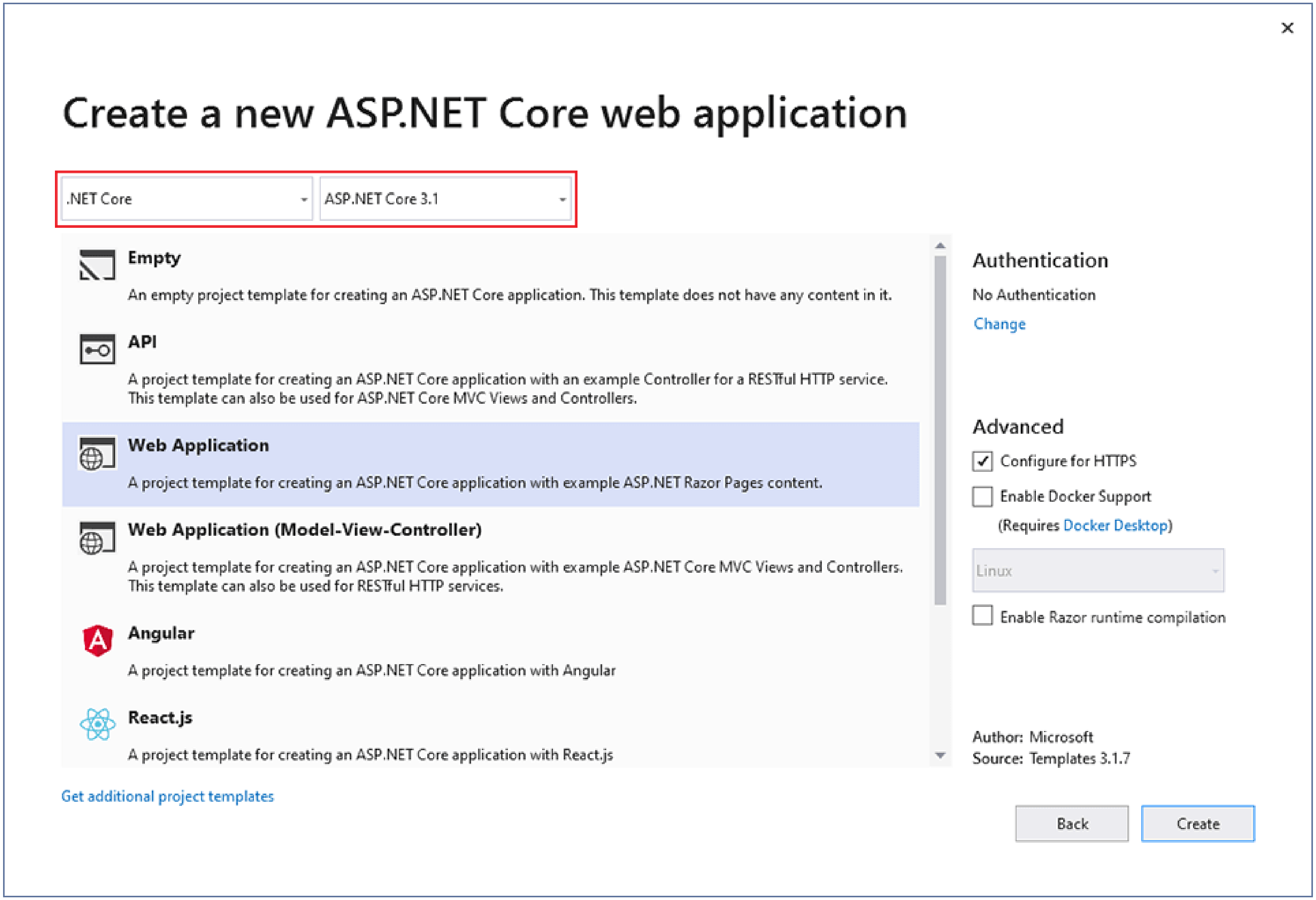
C:\repos

Solution name

RazorPagesMovie

☑ Place solution and project in the same directory

- Select **ASP.NET Core 3.1** in the dropdown, **Web Application**, and then select **Create**.



Create a new ASP.NET Core web application

.NET Core ▾ ASP.NET Core 3.1 ▾

Empty

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

API

A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

Web Application

A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

Web Application (Model-View-Controller)

A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

Angular

A project template for creating an ASP.NET Core application with Angular

React.js

A project template for creating an ASP.NET Core application with React.js

Authentication

No Authentication

Change

Advanced

☑ Configure for HTTPS

☐ Enable Docker Support

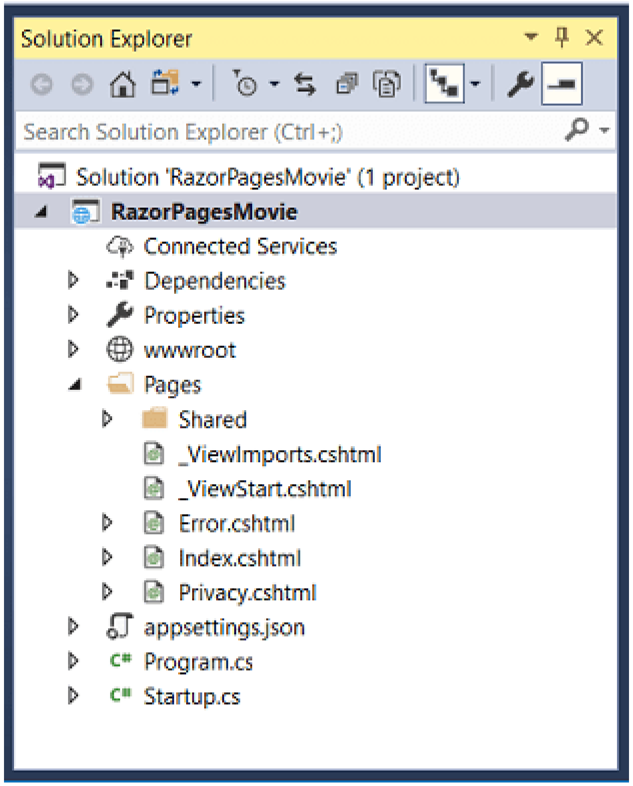
(Requires Docker Desktop)

Linux

☐ Enable Razor runtime compilation

Author: Microsoft
Source: Templates 3.1.7

The following starter project is created:



Solution Explorer

Search Solution Explorer (Ctrl+;)

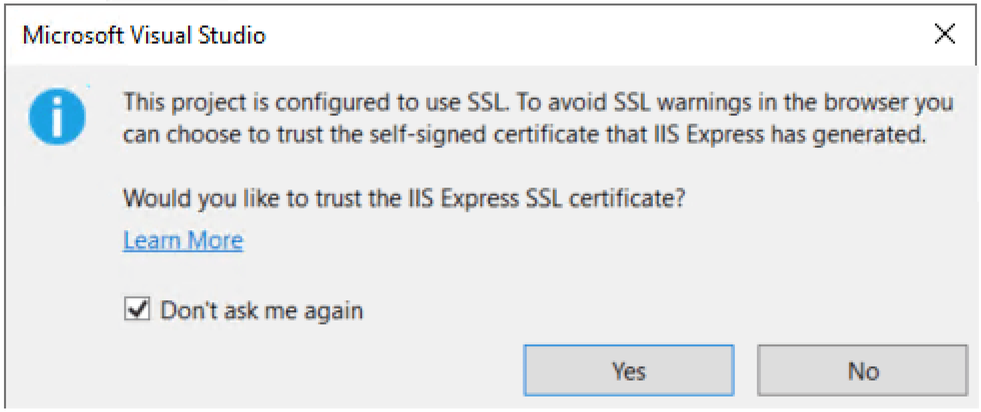
```
Solution 'RazorPagesMovie' (1 project)
◢ RazorPagesMovie
    Connected Services
  ▷ Dependencies
  ▷ Properties
  ▷ wwwroot
  ◢ Pages
    ▷ Shared
      _ViewImports.cshtml
      _ViewStart.cshtml
    ▷ Error.cshtml
    ▷ Index.cshtml
    ▷ Privacy.cshtml
  ▷ appsettings.json
  ▷ Program.cs
  ▷ Startup.cs
```

Run the app

Visual Studio | Visual Studio Code | Visual Studio for Mac

- Press Ctrl+F5 to run without the debugger.

  Visual Studio displays the following dialog:



Microsoft Visual Studio

This project is configured to use SSL. To avoid SSL warnings in the browser you can choose to trust the self-signed certificate that IIS Express has generated.

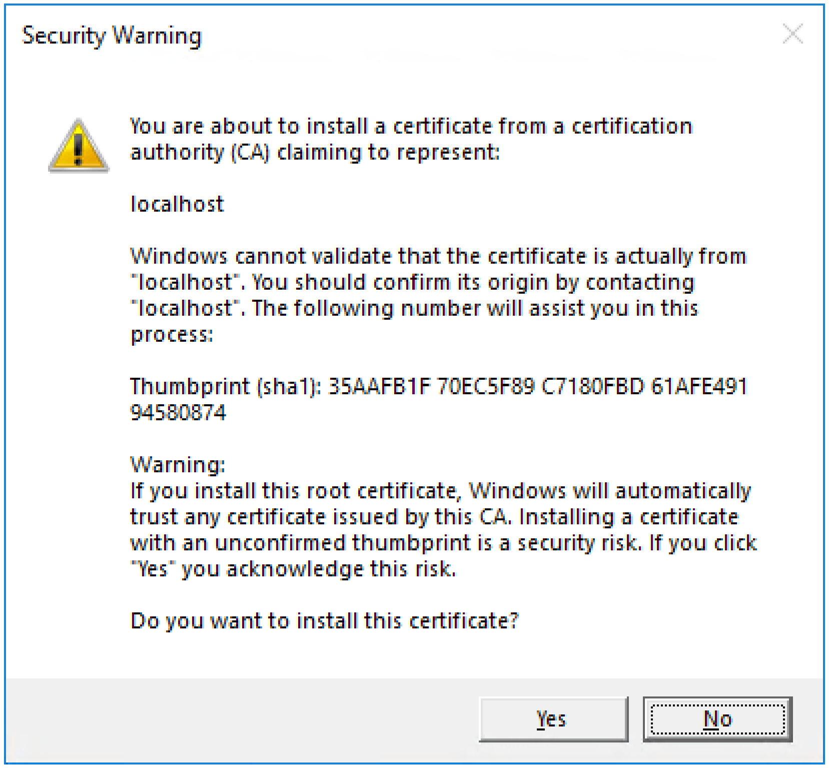
Would you like to trust the IIS Express SSL certificate?

Learn More

☑ Don't ask me again

Select **Yes** if you trust the IIS Express SSL certificate.

The following dialog is displayed:



Security Warning

You are about to install a certificate from a certification authority (CA) claiming to represent:

localhost

Windows cannot validate that the certificate is actually from "localhost". You should confirm its origin by contacting "localhost". The following number will assist you in this process:

Thumbprint (sha1): 35AAFB1F 70EC5F89 C7180FBD 61AFE491 94580874

Warning:
If you install this root certificate, Windows will automatically trust any certificate issued by this CA. Installing a certificate with an unconfirmed thumbprint is a security risk. If you click "Yes" you acknowledge this risk.

Do you want to install this certificate?

Select **Yes** if you agree to trust the development certificate.

Visual Studio starts IIS Express and runs the app. The address bar shows `localhost:port#` and not something like `example.com`. That's because `localhost` is the standard hostname for the local computer. Localhost only serves web requests from the local computer. When Visual Studio creates a web project, a random port is used for the web server.

Examine the project files

Here's an overview of the main project folders and files that you'll work with in later tutorials.

Pages folder

Contains Razor pages and supporting files. Each Razor page is a pair of files:

- A *.cshtml* file that has HTML markup with C# code using Razor syntax.
- A *.cshtml.cs* file that has C# code that handles page events.

Supporting files have names that begin with an underscore. For example, the *_Layout.cshtml* file configures UI elements common to all pages. This file sets up the navigation menu at the top of the page and the copyright notice at the bottom of the page. For more information, see Layout in ASP.NET Core.

wwwroot folder

Contains static files, like HTML files, JavaScript files, and CSS files. For more information, see Static files in ASP.NET Core.

appSettings.json

Contains configuration data, like connection strings. For more information, see Configuration in ASP.NET Core.

Program.cs

Contains the entry point for the program. For more information, see .NET Generic Host in ASP.NET Core.

Startup.cs

Contains code that configures app behavior. For more information, see App startup in ASP.NET Core.

Next steps

Next: Add a model

Is this page helpful?

👍 Yes 👎 No