

# Getting Started with Orchard Core CMS using Visual Studio

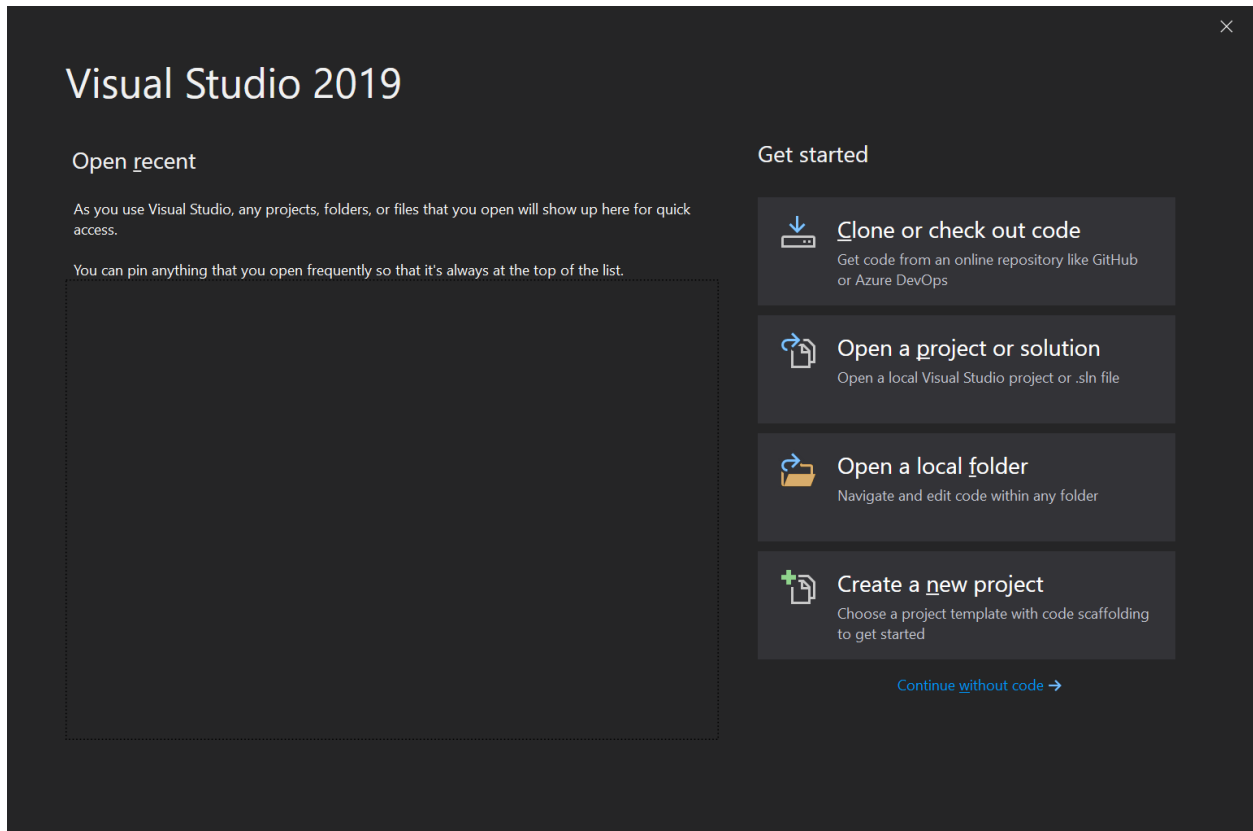


## Introduction

In this article, you're going to see how easy it is to get Orchard Core running in your very own ASP.NET Core Web Application using Visual Studio. You'll be using the power of the Orchard Core NuGet packages to simplify the process.

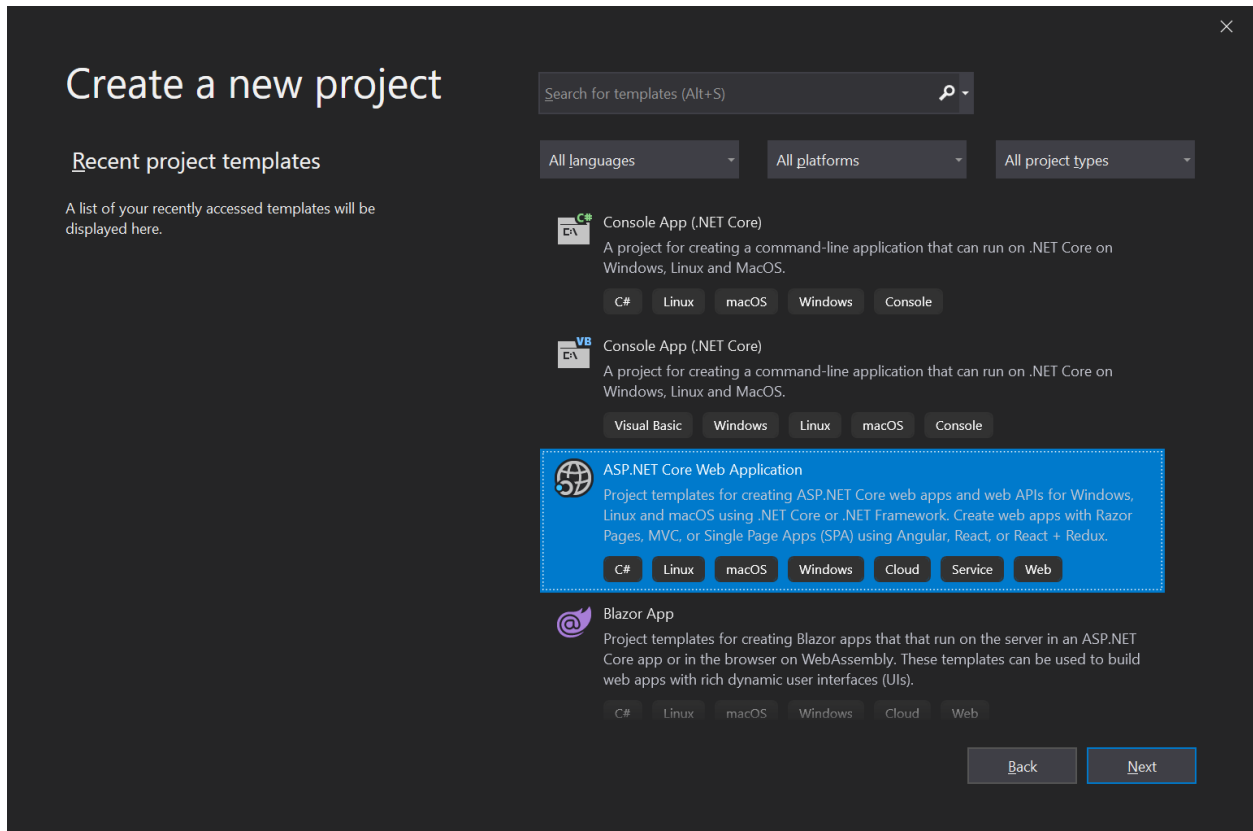
## Create a new ASP.NET Core Web Application

First, launch Visual Studio, and create a new ASP.NET Core Web Application:

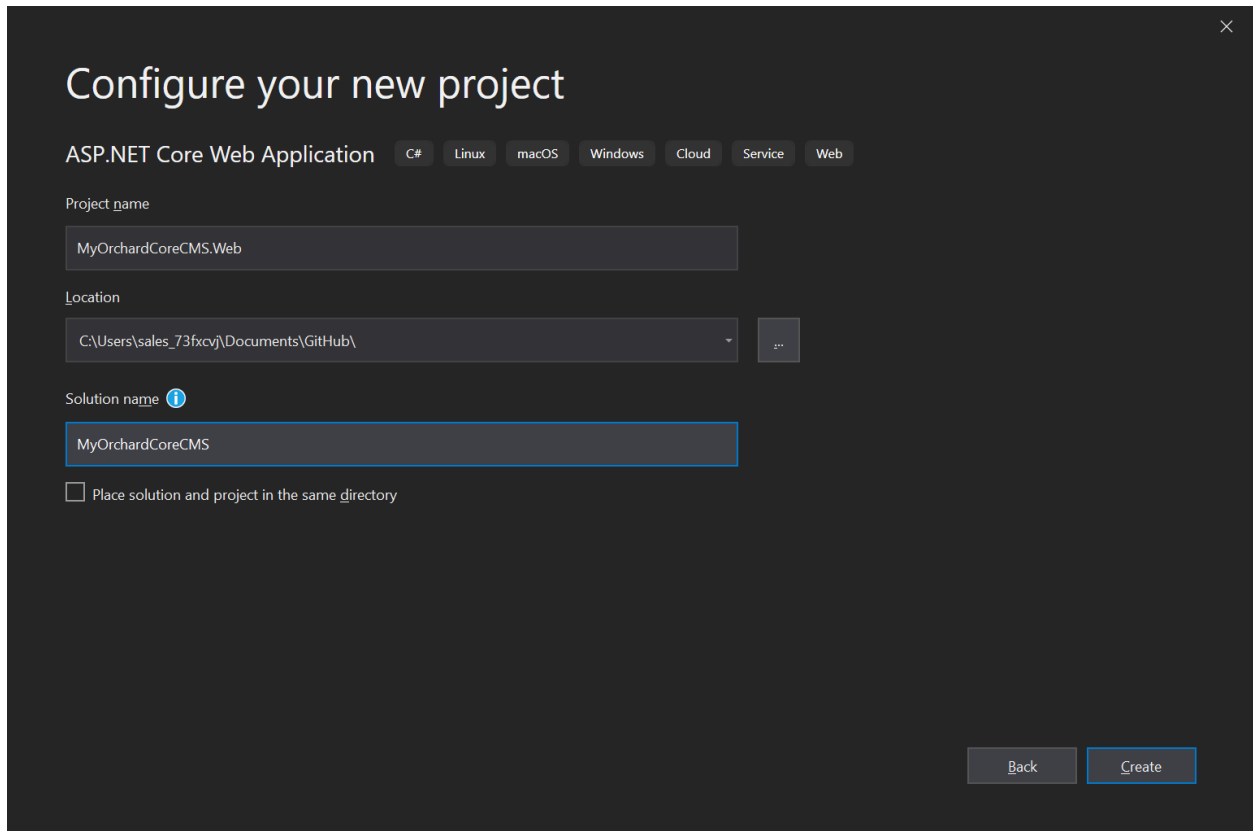


Select the box button at the lower left of the screen, “Create a new project”.

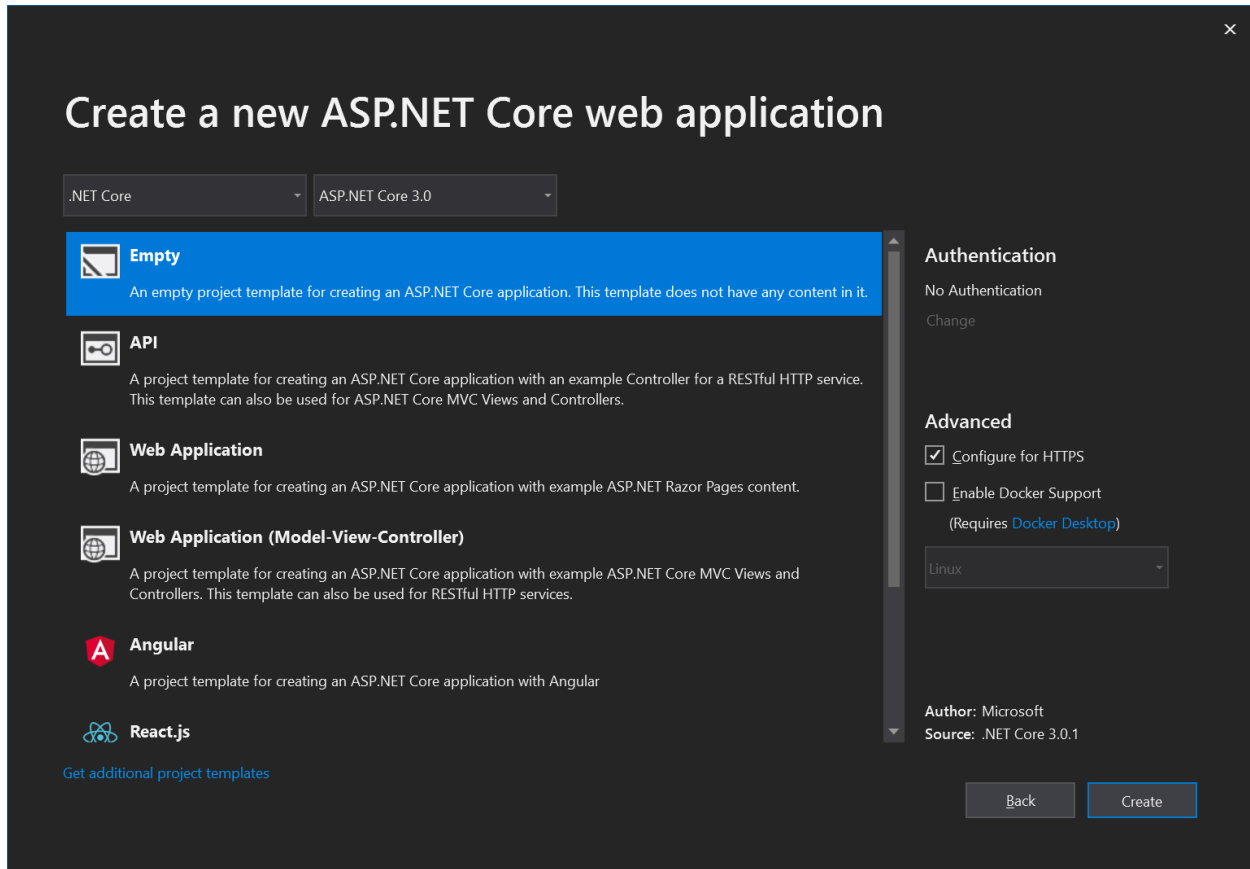
## Getting Started with Orchard Core CMS using Visual Studio



Select the ASP.NET Core Web Application that is highlighted in blue and press the “Next” button.



Enter a project name, location, and a solution name. It's customary to add a ".Web" to specify the project is a web application. The Solution can be whatever you decide. Another customary practice is to put all the source code in a root "src" folder. Again, it's up to you to decide how you want to structure your code.

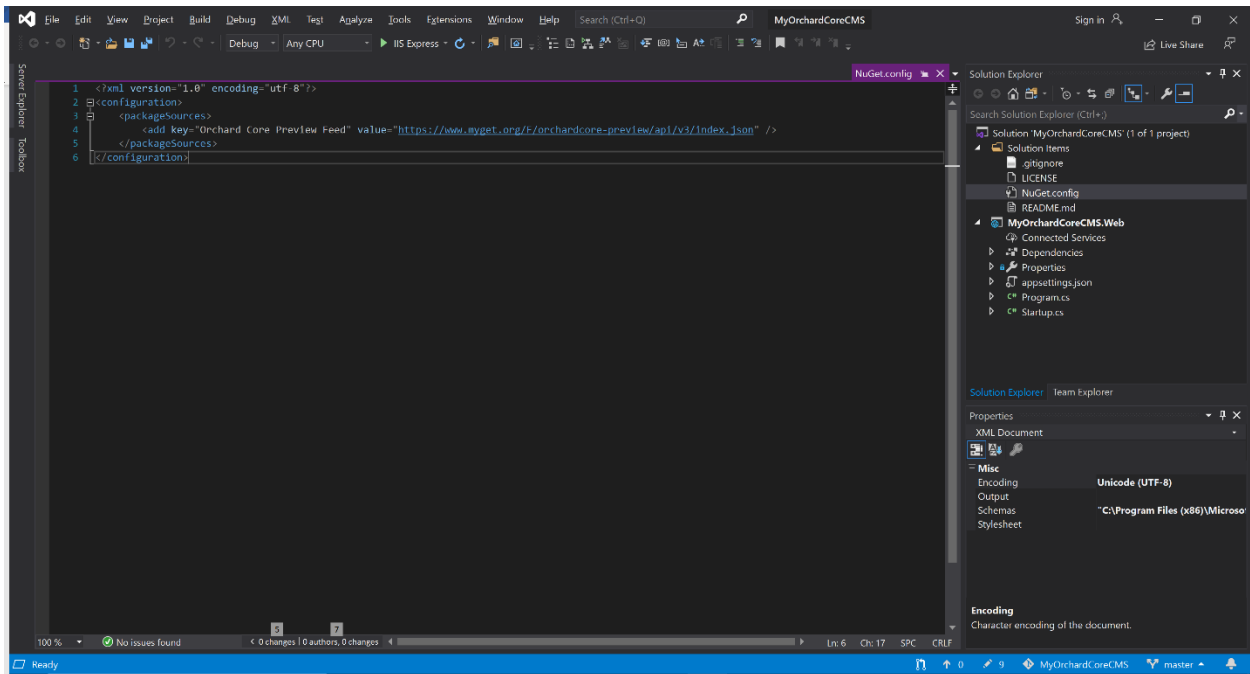


At the time of writing this article, Orchard Core is based on ASP.NET Core version 3.0. Make sure you select “ASP.NET Core 3.0” in the top right combo. Select the “Empty” template and then press the “Create” button.

## Create “NuGet.config”

In order to utilize the latest code, use the dev packages, by adding the Orchard Core Preview Feed. Create the file “NuGet.config” in the root folder and add the following configuration.

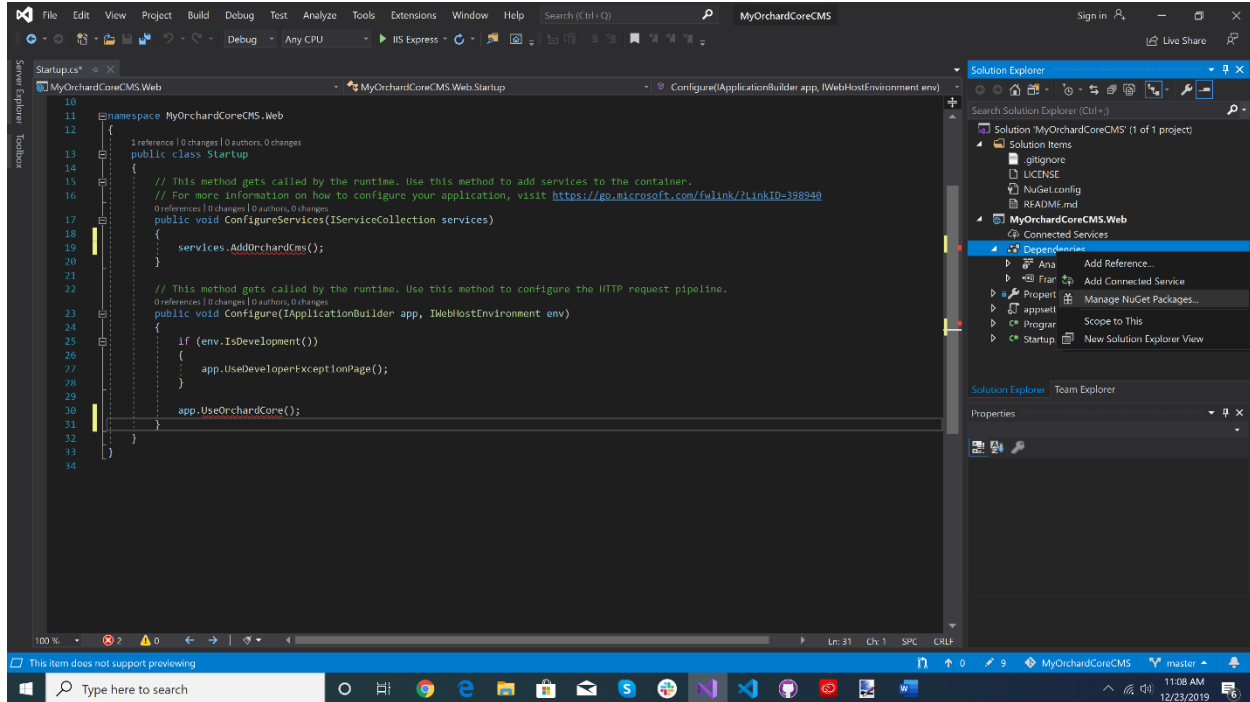
## Getting Started with Orchard Core CMS using Visual Studio



Nuget.config:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <packageSources>
    <add key="Orchard Core Preview Feed" value="https://www.myget.org/F/orchardcore-
preview/api/v3/index.json" />
  </packageSources>
</configuration>
```

## Modify Startup Code and Add the Orchard Core Preview Feed to your NuGet Package Manager Package Sources



Update code block to add Orchard Core CMS.

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

namespace MyOrchardCoreCMS.Web
{
    public class Startup
    {
        // This method gets called by the runtime. Use this method to add services to the
        // container.
        // For more information on how to configure your application, visit
        https://go.microsoft.com/fwlink/?LinkID=398940
        public void ConfigureServices(IServiceCollection services)
        {
            services.AddOrchardCms();
        }

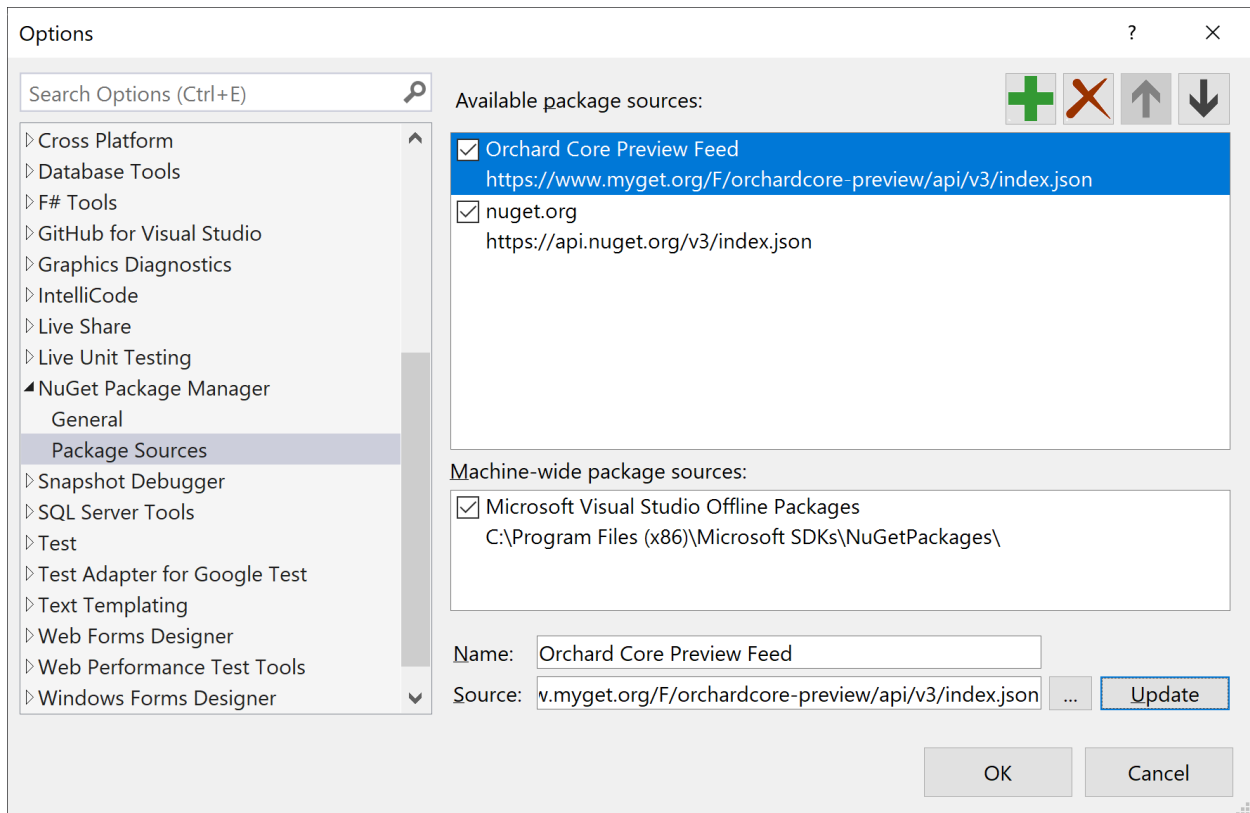
        // This method gets called by the runtime. Use this method to configure the HTTP
        // request pipeline.
        public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
        {

```

## Getting Started with Orchard Core CMS using Visual Studio

```
    if (env.IsDevelopment())  
    {  
        app.UseDeveloperExceptionPage();  
    }  
  
    app.UseOrchardCore();  
}  
}
```

Select the Include prerelease check box. Select the gear icon on the left side Package Source



Verify the Orchard Core Preview Feed package source is the first entry and that the URL is correct.

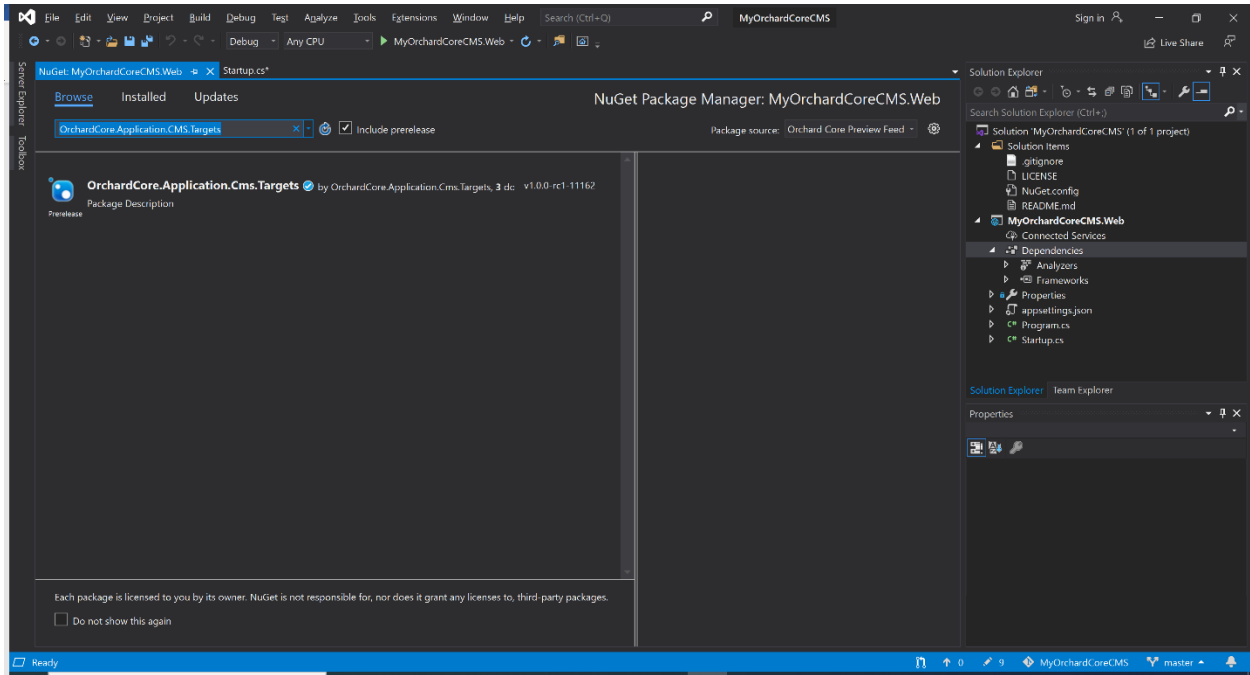
Name: Orchard Core Preview Feed

Source: <https://www.myget.org/F/orchardcore-preview/api/v3/index.json>

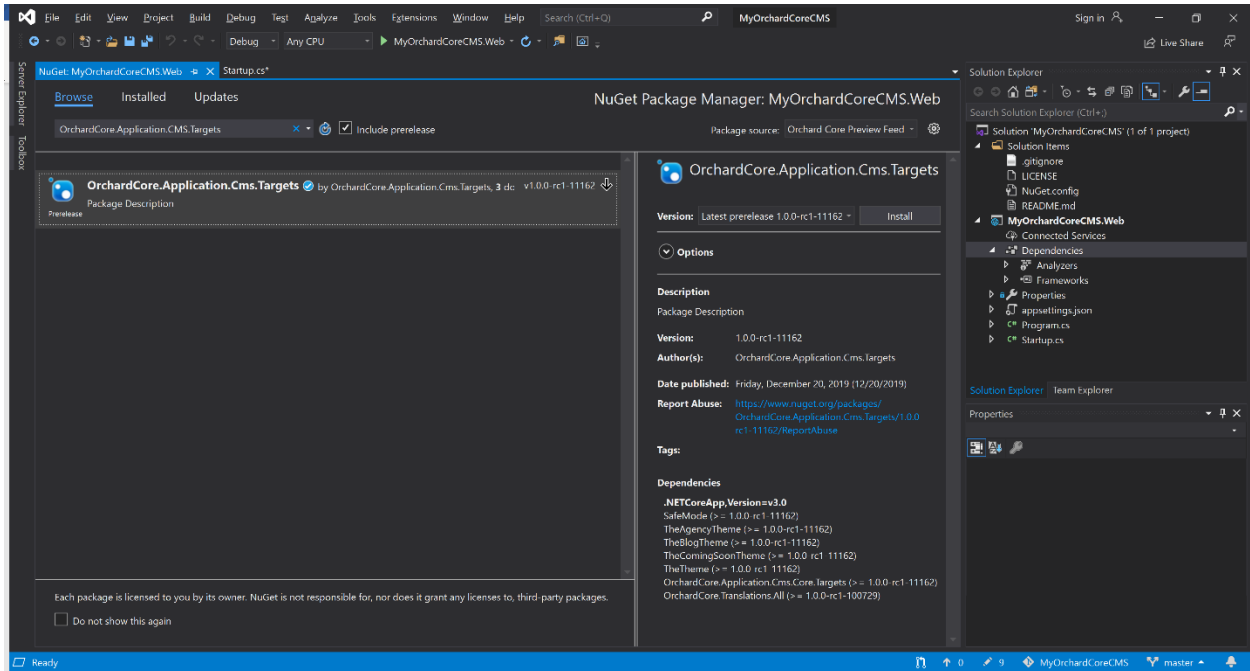


## Getting Started with Orchard Core CMS using Visual Studio

### Install OrchardCore.Application.CMS.Targets

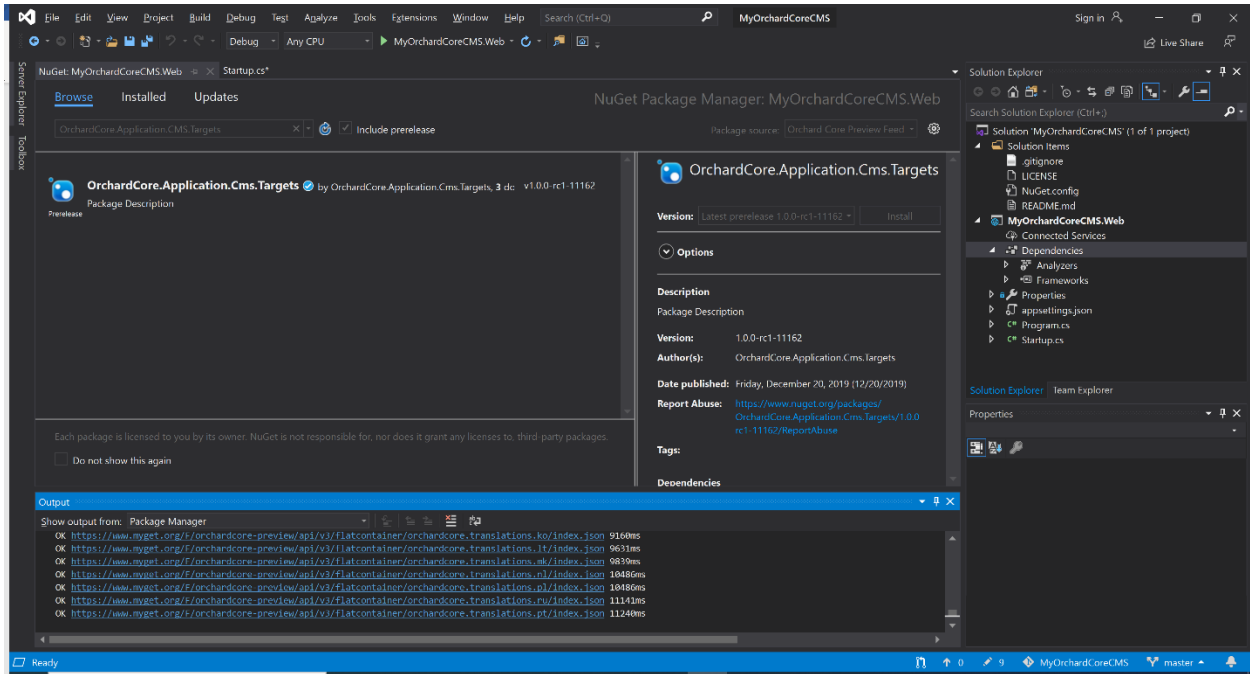


Select the “Browse” tab in the NuGet Package Manager Window. Enter “OrchardCore.Application.CMS.Targets” in the Search edit box. Make sure the check box “Include prereleases is checked.

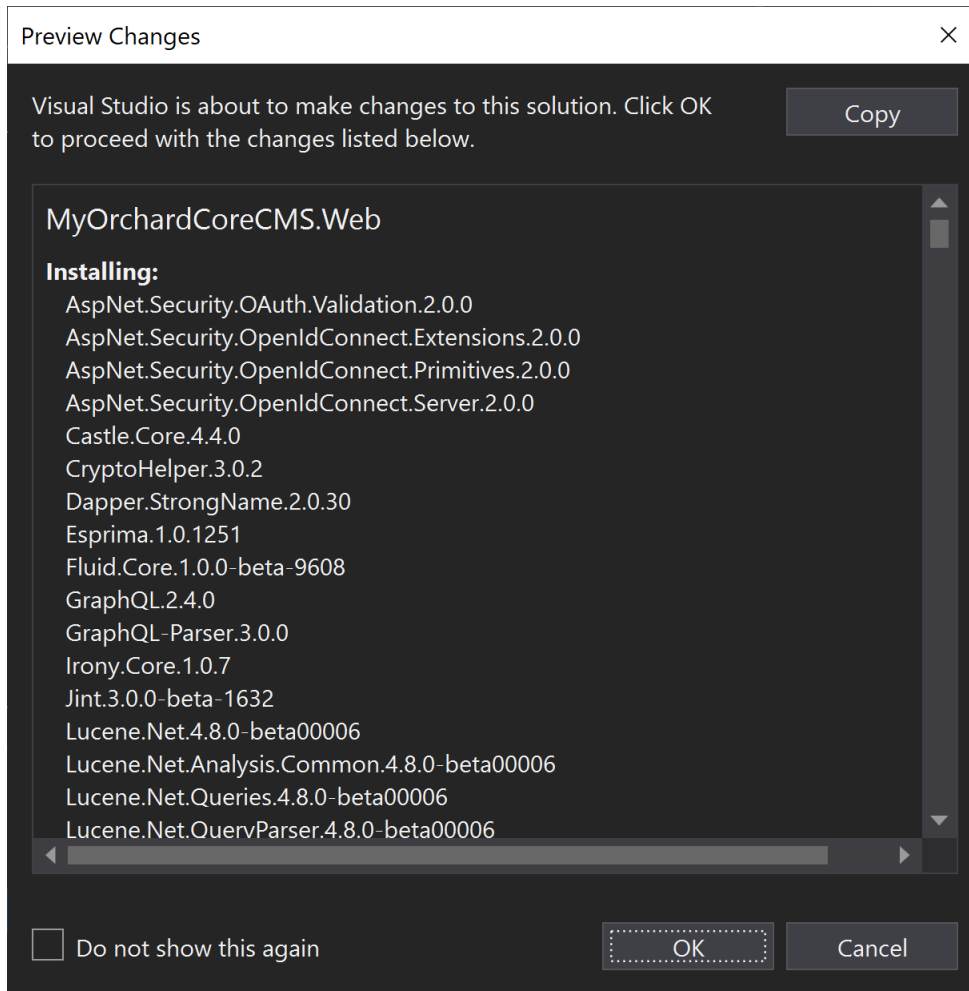


Select the package “OrchardCore.Application.CMS.Targets” and press the “install” button.

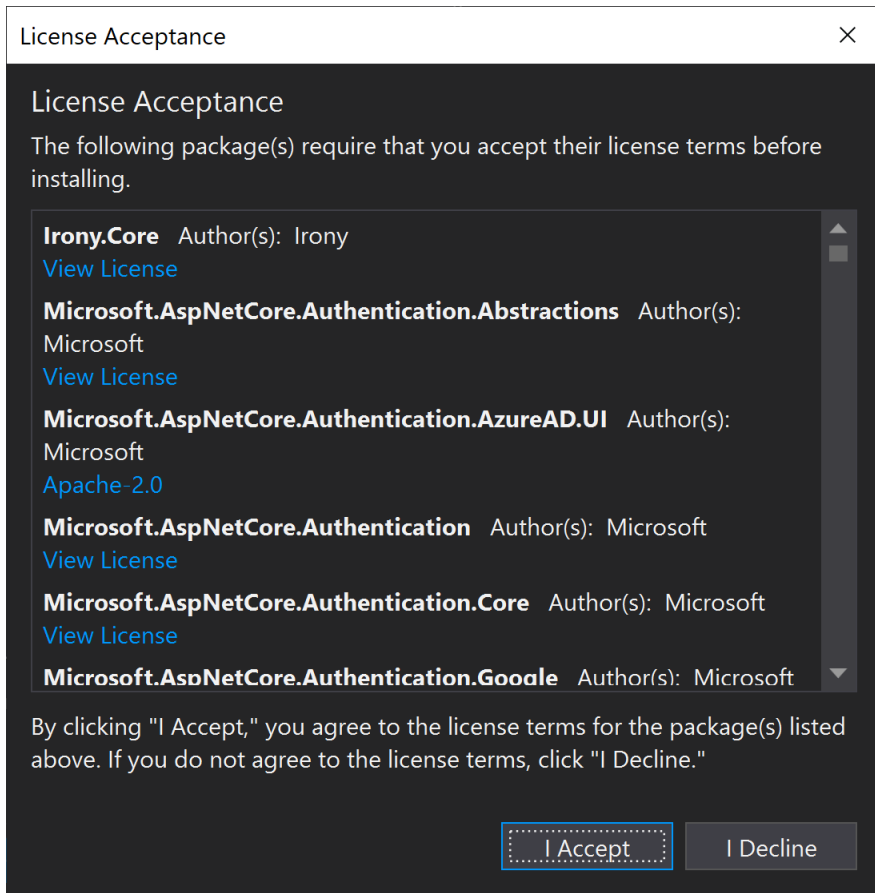
## Getting Started with Orchard Core CMS using Visual Studio



Wait until all the packages are installed.

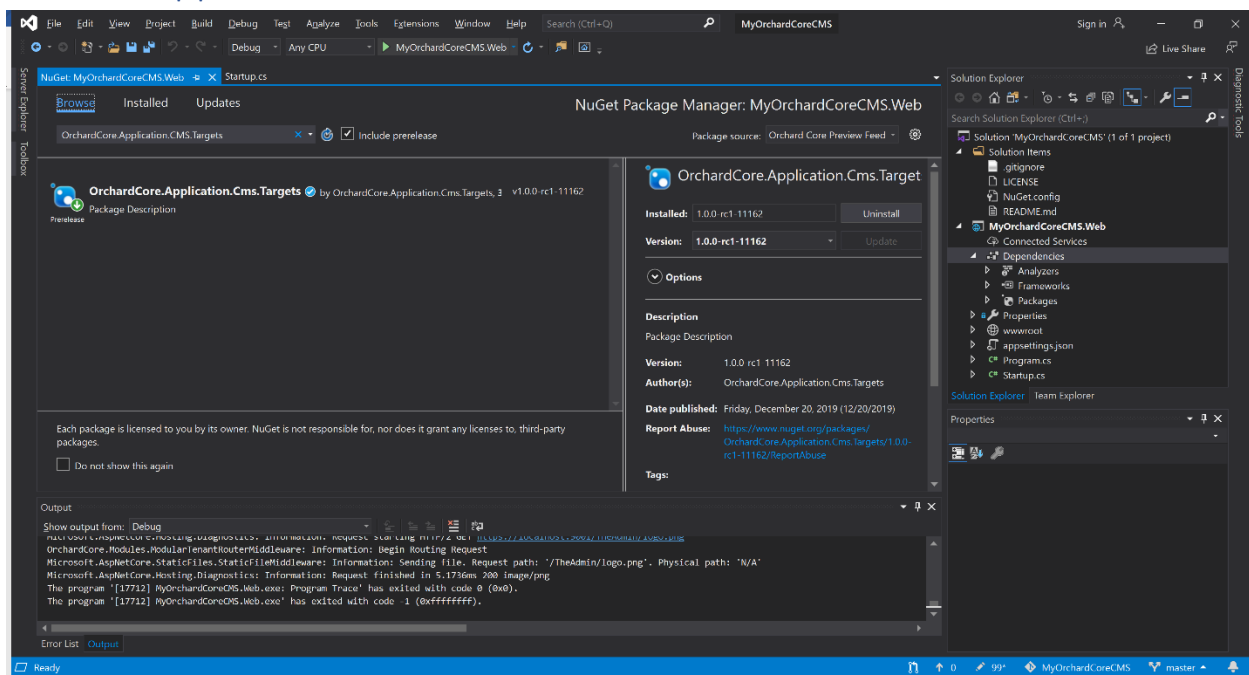


Allow Visual Studio to make changes by selecting the “OK” button.

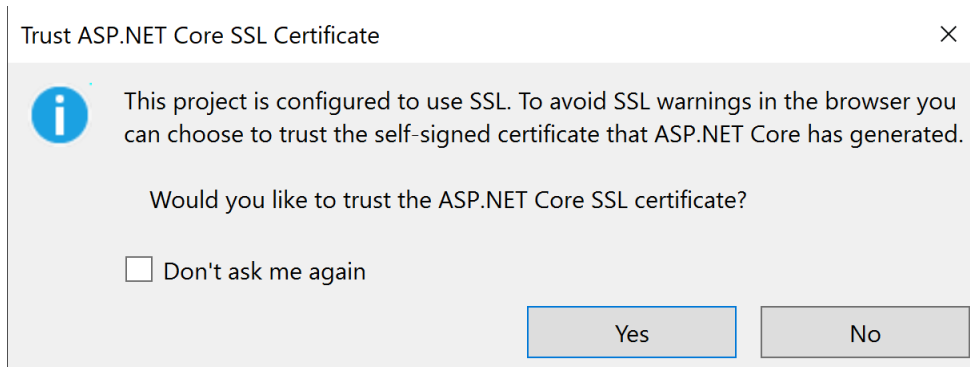


Select the "I Accept" button to accept the license agreement.

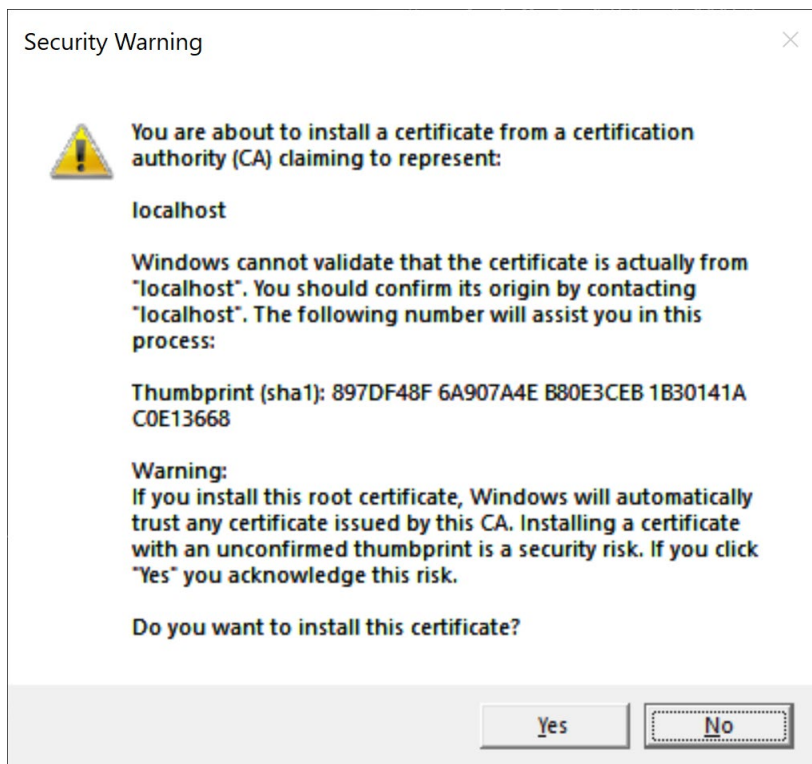
## Run the Application



Select "MyOrchardCoreCMS.Web" and click on the green play button to run the application.



Select "Yes" button to trust the self-signed certificate.

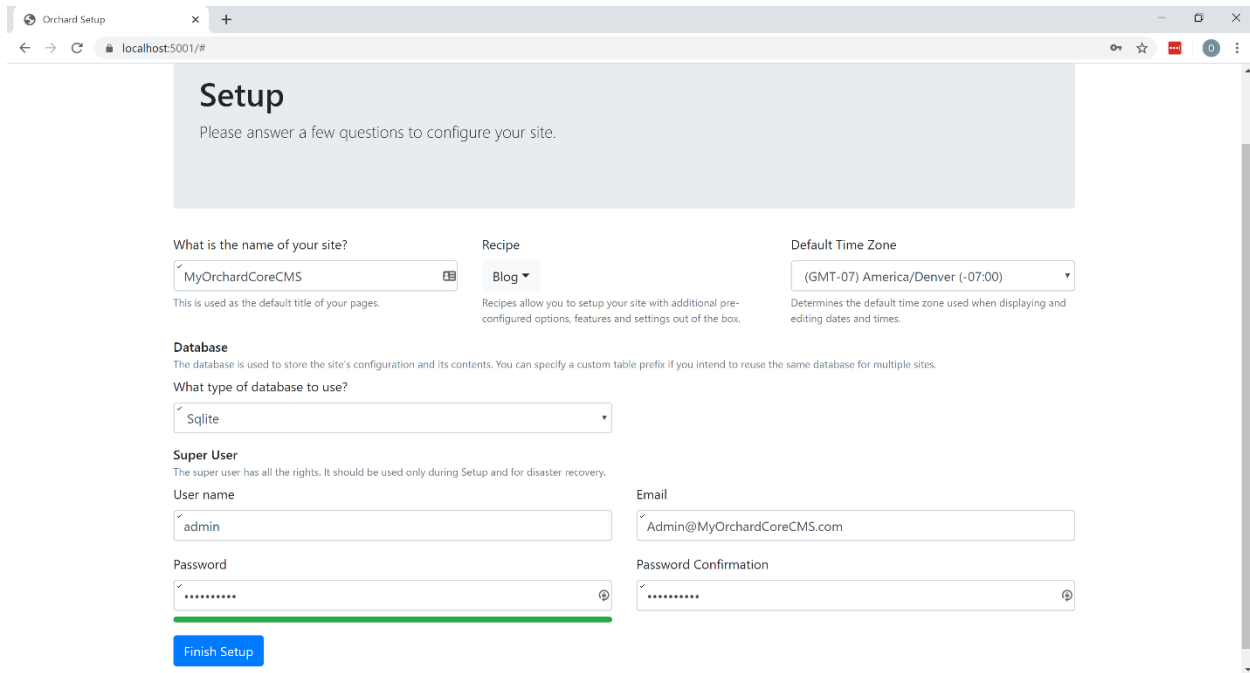


Select the "Yes" button to install the certificate.

## Getting Started with Orchard Core CMS using Visual Studio

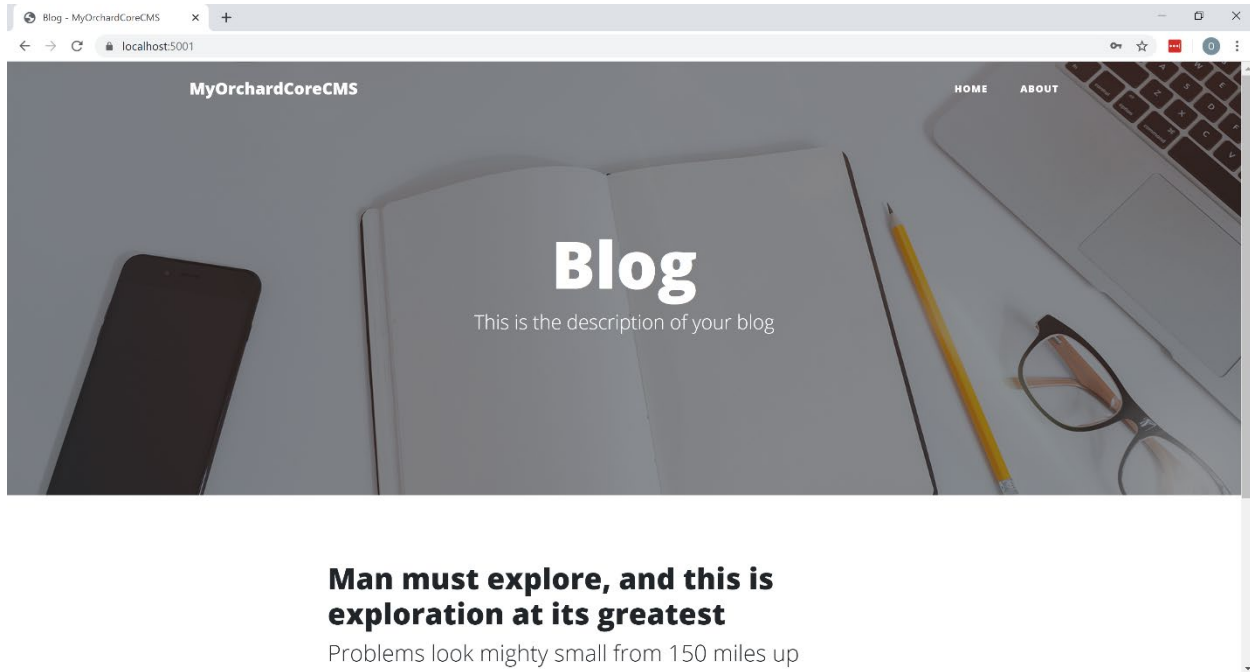
```
C:\Users\sales_73fxcvj\Documents\GitHub\MyOrchardCoreCMS\src\MyOrchardCoreCMS.Web\bin\Debug\netcoreapp3.0\MyOrchardCoreCMS...
Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
Content root path: C:\Users\sales_73fxcvj\Documents\GitHub\MyOrchardCoreCMS\src\MyOrchardCoreCMS.Web
info: Microsoft.AspNetCore.Hosting.Diagnostics[1]
Request starting HTTP/2 GET https://localhost:5001/
info: OrchardCore.Environment.Shell.ShellHost[0]
Start creation of shells
dbug: OrchardCore.Environment.Shell.ShellHost[0]
Creating shell context for root setup.
dbug: OrchardCore.Environment.Shell.Builders.ShellContextFactory[0]
No shell settings available. Creating shell context for setup
dbug: OrchardCore.Environment.Shell.Builders.ShellContextFactory[0]
Creating described context for tenant 'Default'
dbug: OrchardCore.Environment.Shell.Builders.CompositionStrategy[0]
Composing blueprint
dbug: OrchardCore.Environment.Shell.Builders.CompositionStrategy[0]
Done composing blueprint
dbug: OrchardCore.Environment.Shell.ShellHost[0]
Registering shell context for tenant 'Default'
info: OrchardCore.Environment.Shell.ShellHost[0]
Done pre-creating and registering shells
info: OrchardCore.Modules.ModularTenantRouterMiddleware[0]
Begin Routing Request
```

A logging window will appear displaying debug information.

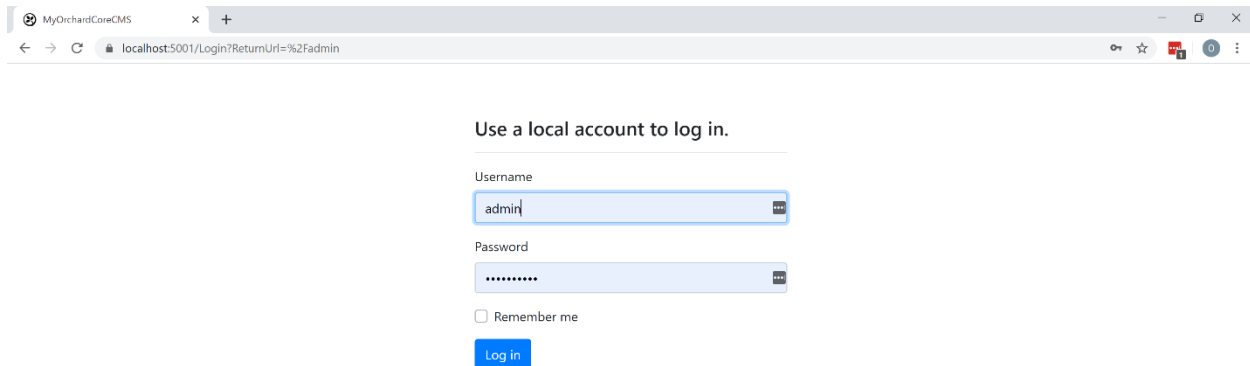


Your Orchard Core application will then display a Setup page. Enter the “name of your site”, “Blog” for Recipe, your “default time zone”, “Sqlite” for the database, a “user name” and password. Press the “Finish Setup” button to complete the process.

## Getting Started with Orchard Core CMS using Visual Studio

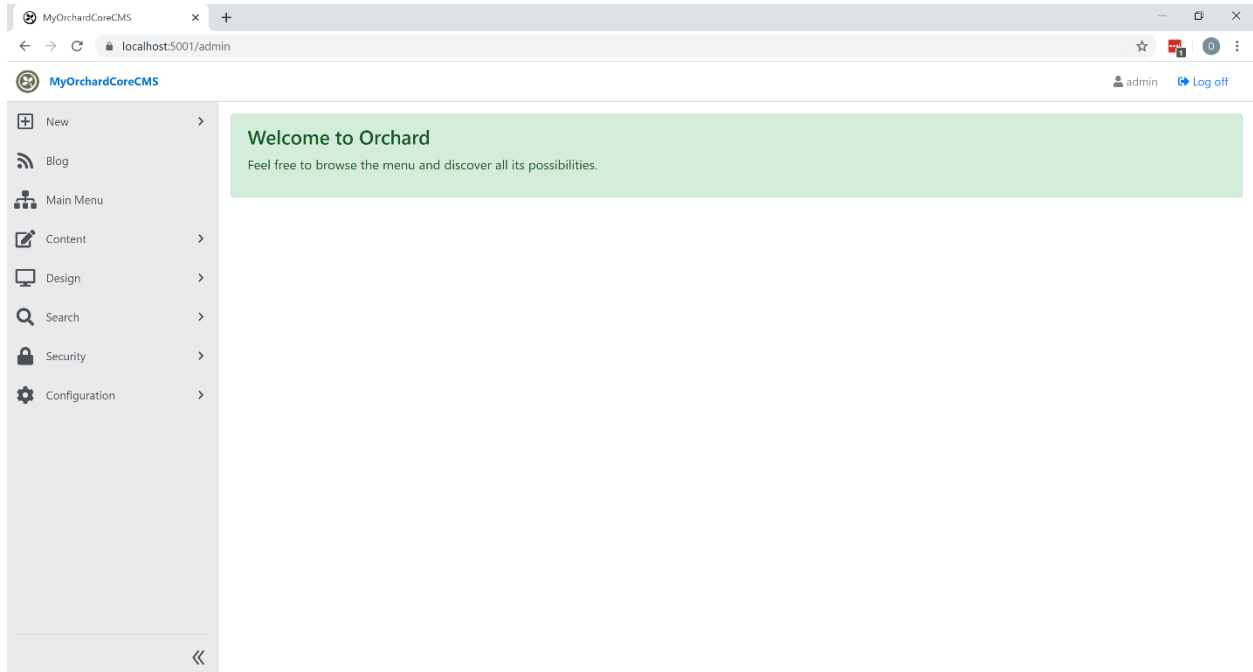


Congratulations! Your Orchard Core Web Application will now be running in the default web browser.



Navigate to the admin dashboard page by specifying “/admin” at the end of the URL. i.e. <https://localhost:5001/admin>. Login with your credentials.

## Getting Started with Orchard Core CMS using Visual Studio



You are now ready to configure and create content for your Orchard Core Web Application.

## Conclusion

With just a new few lines of code and some configuration, it is easy to create a Content Management System with ASP.NET Core and Orchard Core CMS.