https://www.pluralsight.com/guides/asp-net-mvc-creating-solutions-with-separate-projects-for-entities-data-access-and-website-functionality

To create a new web application using ASP.NET Core and Angular, in Visual Studio, click on File, New, and select Project. In the New Project dialogbox, select ASP.NET Core Web Application, and then in the New ASP.NET Core Web Application dialogbox, make sure .NET Core is selected and ASP.NET Core 2.0. Select Angular from the selection and click on the OK button. Name this something appropriate, such as keenComm.Web

You should now be able to click on F5 to start debugging the new Angular ASP.NET Core application.

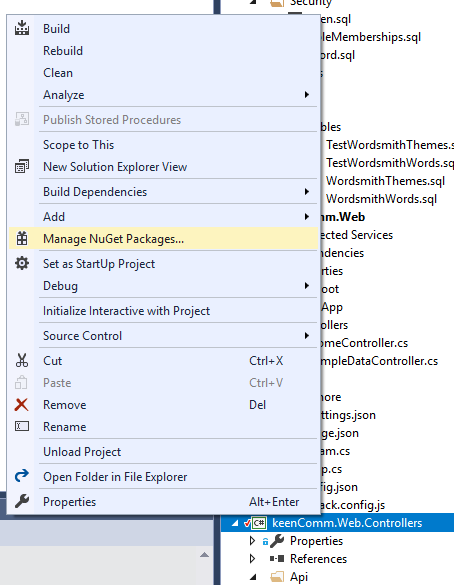
You can add a controller project to your solution by right clicking on the Solution in Solution Explorer and choosing Add, New Project. Under Visual C#, select Class Library (.NET Framework) and a name, such as keenComm

.Web.Controllers. Under Framework, ensure .NET Framework 4.6.2 is selected and click on the OK button.

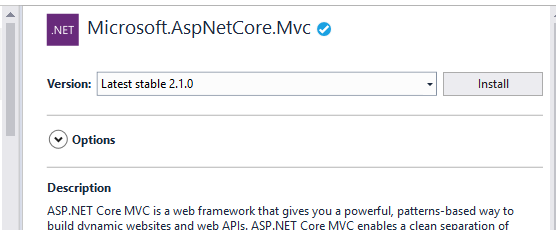
To create a new database project to house our data, right click on the Solution in Solution Explorer and choose Add, New Project. In the Add New Project dialogbox, under Installed on the left hand side of the dialogbox, select SQL Server and the select SQL Server Database Project. Specify an appropriate name, such as keenproj.db and click on the OK button. You can now create the folders for your database project that will create your tables and view. Create Scripts, Tables, and Views folders.

To import an existing database in my scenario, right click on the keenComm.Db project and select Import, and then select Database. Click on the Select Connection button and then select your database credentials. If you have this setup already, the existing connection should be listed in the list of available connections. You can also click on the Show Connection Properties and complete the information to create a new connection to the database you wish.

When setting up a new controller, you must reference the Microsoft.AspNetCore.Mvc. You can do this by right clicking on the keenComm.Web.Controllers project and choosing Manage NuGet Packages…



Next under Browse, enter in Microsoft.AspNetCore.Mvc in the search box, select the appropriate item, and then click on the Install button that enables in the right hand pane.



You can also do this for the following packages:

Microsoft.AspNetCore.Authentication.Cookies

Microsoft.AspNetCore.Mvc.Core

Right click on the keenComm.Web project and choose Manage NuGet Packages. Under Browse, search for and install the following packages:

Microsoft.AspNetCore.All

Microsoft.EntityFrameworkCore.SqlServer

Microsoft.EntityFrameworkCore.Tools

Microsoft.Extensions.DependencyInjection

Microsoft.Extensions.DependencyInjection.Abstraction

Microsoft.NetCore.App

Microsoft.VisualStudio.Web.CodeGeneration.Tools

NLog.Web.AspNetCore

Swashbuckle.AspNetCore

Note: After installing the packages, you may notice the Updates. Click on this option to update any installed packages.

// Skip for now

Right click on the Controllers folder and choose Add, Controller. Select API Controller – Empty and click the Add button. Name the Controller BaseAPIController and click on the OK button.

// End Skip for now

Right click on the keenComm.Web.Controllers project and choose Add, Class. In the Add New Item dialogbox, select Claass and name the class WordSmithController. At the top of this file, enter in the following line:

using Microsoft.AspNetCore.Mvc;

using Newtonsoft.Json;

Enter in : Controller after the public class WordSmithController.

public class WordSmithController : Controller

You will need to add a reference to this project in keenComm.Web project by right clicking on the Dependencies and choosing Add Reference. In the Add Reference dialog box, click on the Solution under Projects and select keen.Comm.Web.Controllers.