Build an EF and ASP.NET Core 2.1 App HOL

Welcome to the Build an Entity Framework Core and ASP.NET Core 2.1 Application in a Day Hands-on Lab. This lab walks you through creating the projects and adding/updating the NuGet packages.

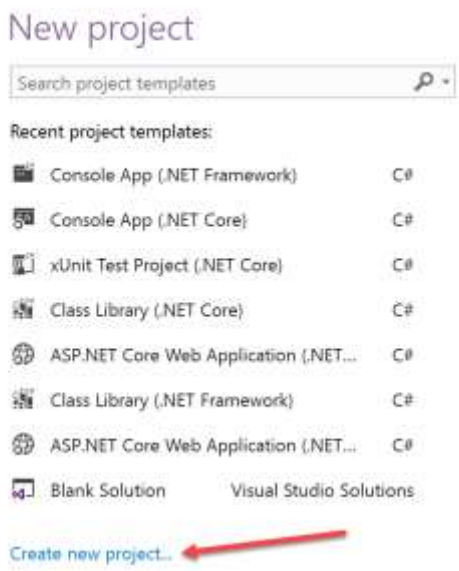
Prior to starting this lab, you must have completed Lab 0, Installing the Prerequisites.

All labs and files are available at https://github.com/skimedic/dotnetcore_hol.

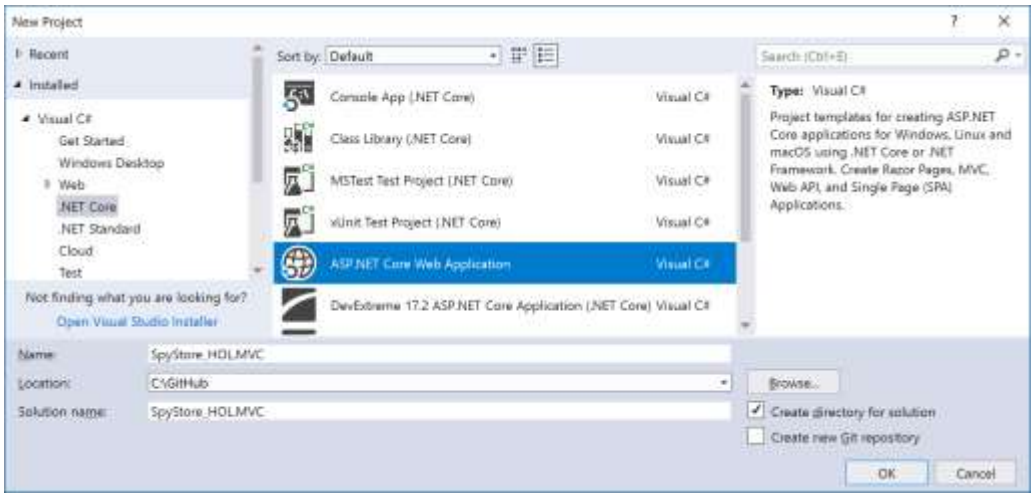
Part 1: Creating the Solution and Projects

Step 1: Create the ASP.NET Core project and solution

1) From the Start Page, select Create New Project



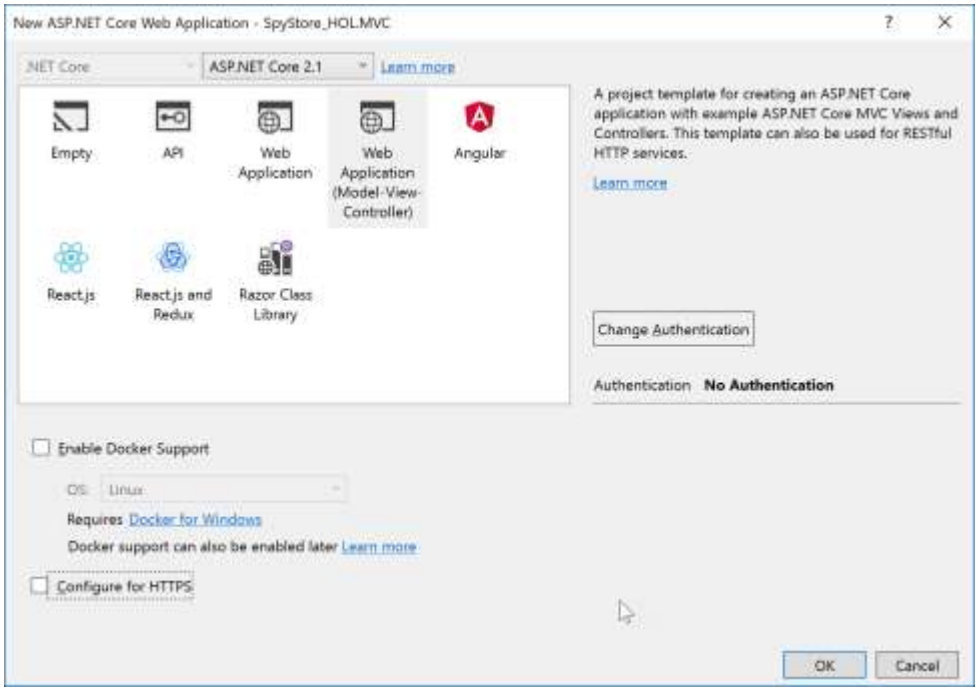
2) Select .NET Core from the templates from the left rail and ASP.NET Core Web Application from the center section and name it SpyStore_HOL.MVC:



3) On the next screen, **make sure the ASP.NET Core 2.1 templates are selected**, then select Web Application (Model-View-Controller). Leave the authentication as "No Authentication" and leave the Enable Docker Support unchecked. Uncheck the configure for HTTPS check box.

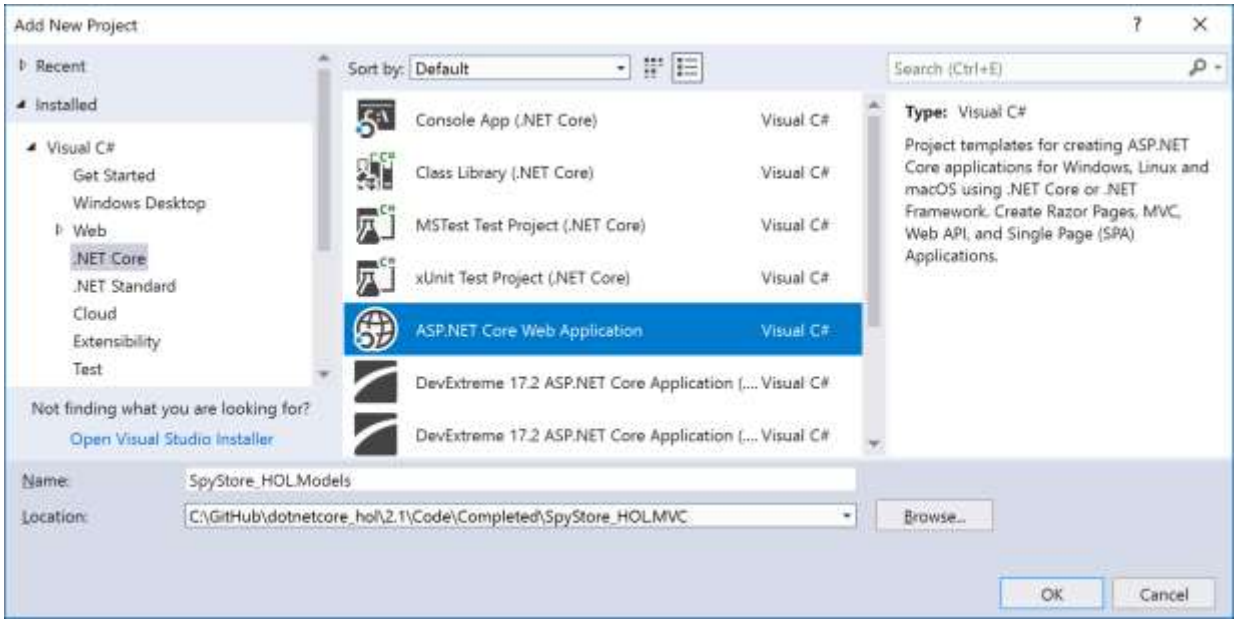
   a) Note: ASP.NET Core 2.1 added in significant support for GDPR and secure browsing. This is beyond the scope of this workshop. To find out more on this subject, go to https://docs.microsoft.com/en-us/aspnet/core/release-notes/aspnetcore-2.1?view=aspnetcore-2.1#https

   b) Note: If you see a message stating that the .NET Core SDK is installed in multiple locations, it means that you have x86 and x64 versions install. VS ignores the x86 versions but still nags you with the message in the new project dialog.



Step 2: Create the Models project

1) Right click on the solution, and select Add -> New Project
2) Select .NET Core from the templates from the left rail and Class Library (.NET Core) from the center section and name it **SpyStore_HOL.Models**:



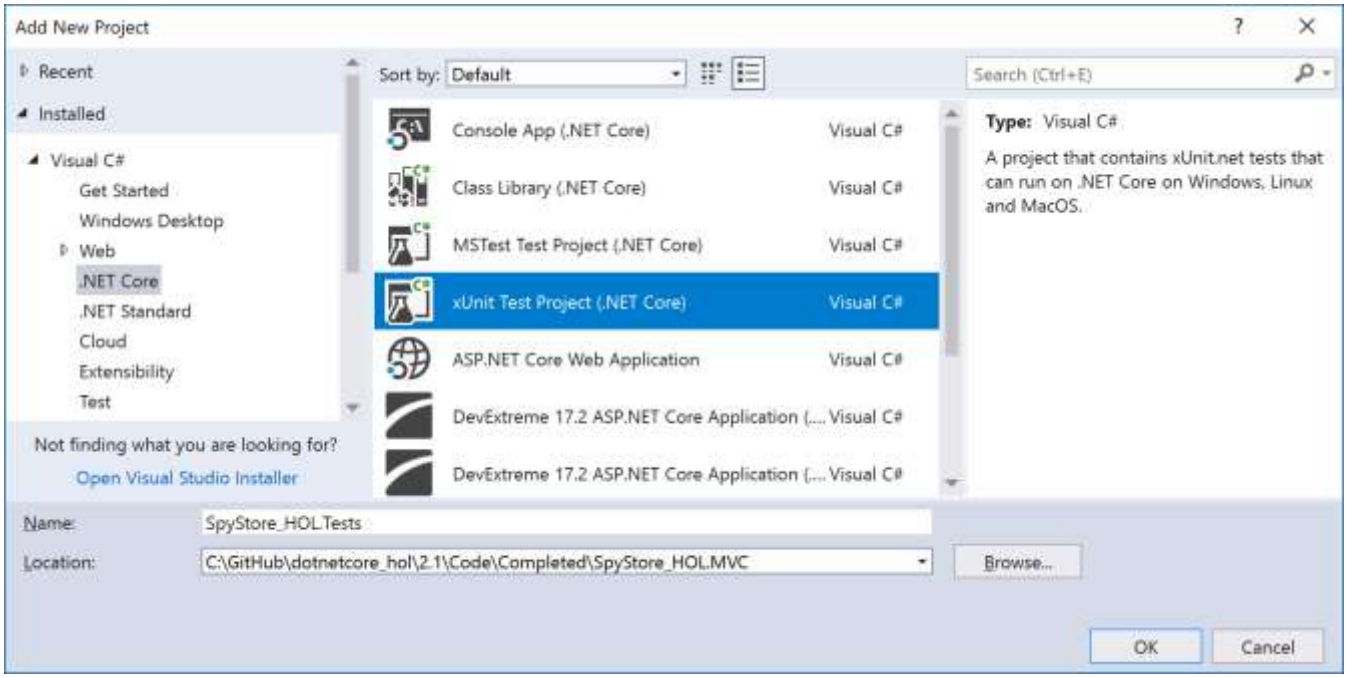
3) Delete the Class.cs file.

Step 3: Create the DAL project

1) Repeat the process and add a new Class Library (.NET Core) named **SpyStore_HOL.DAL**:
2) Delete the Class.cs file

Step 4: Create the Unit Test project [Optional]

1) Right click on the solution, and select Add -> New Project
2) Select .NET Core from the templates from the left rail and xUnit Test Project (.NET Core) from the center section and name it **SpyStore_HOL.Tests**:



Part 2: Add/Update the NuGet packages

.NET Core is composed of a series of NuGet packages that sometimes update faster than the VS2017 templates, so in addition to adding new packages for our solution, the existing packages might need to be updated.

NOTE: Make sure you don't have the "Include prerelease" checked.

Step 1: Update the MVC Project

1) Right click on the SpyStore_HOL.MVC project, and select Manage NuGet Packages.
2) Click Updates at the top, and update all packages that need updating (there might not be any).
3) Examine the two installed NuGet packages:
    a) Microsoft.AspNetCore.App
    b) Microsoft.NETCore.App
4) The Microsoft.AspNetCore.App package contains everything you need to build and run an ASP.NET MVC Core application, including Entity Framework Core
5) Add the AutoMapper package by clicking Browse, then entering AutoMapper in the search box. Select AutoMapper in the left rail, and click install in the right rail.

Step 2: Update the Models Project

1) Right click on the SpyStore_HOL.Models project, and select Manage NuGet Packages.
2) Click Updates at the top, and update all packages that need updating (there might not be any).
3) Add the Entity Framework Core abstraction layer package by clicking Browse, then entering Microsoft.EntityFrameworkCore.Abstractions in the search box. Select and install: Microsoft.EntityFrameworkCore.Abstractions 2.1.x
    a) This enables configuring EF Core entities without having to install EF Core into the project.

Step 3: Update the DAL Project

1) Right click on the SpyStore_HOL.DAL project, and select Manage NuGet Packages.
2) Click Updates at the top, and update all packages that need updating (there might not be any).
3) Install the following packages:
   Microsoft.EntityFrameworkCore 2.1.x
   Microsoft.EntityFrameworkCore.Design 2.1.x
   Microsoft.EntityFrameworkCore.Relational 2.1.x
   Microsoft.EntityFrameworkCore.SqlServer 2.1.x

Step 4: Update the Unit Test Project

1) Right click on the SpyStore_HOL.Tests project, and select Manage NuGet Packages.
2) Click Updates at the top, and update all packages that need updating (there might not be any).

3) Add the Entity Framework Core packages by clicking Browse, then entering Microsoft.EntityFrameworkCore in the search box. Select and install
   Microsoft.EntityFrameworkCore 2.1.x
   Microsoft.EntityFrameworkCore.SqlServer 2.1.x
4) Add the xUnit CLI package by editing the SpyStore_HOL.Tests project file. Add a new ItemGroup as follows, then save the project file:

```
<ItemGroup>
  <DotNetCliToolReference Include="dotnet-xunit" Version="2.3.1" />
</ItemGroup>
```

Step 5: Restore the packages (Optional with .NET Core 2.1+)

With .NET Core 2.1 and later, all packages are restored on project build. If you want to manually update the packages, do one of the following:

- Open Package Manager Console (View -> Other Windows -> Package Manager Console) and enter the following command to restore all packages:

```
dotnet restore
```

- Open a command prompt, navigate to the top level folder of your solution, and enter the following command to restore all packages:

```
dotnet restore
```

Part 3: Add the Project References

Step 1: Update the MVC Project

1) Right click on the SpyStore_HOL.MVC project, and select Add -> Reference
2) Select SpyStore_HOL.DAL and SpyStore_HOL.Models

Step 2: Update the DAL Project

1) Right click on the SpyStore_HOL.DAL project, and select Add -> Reference
2) Select SpyStore_HOL.Models

Step 3: Update the Tests Project

1) Right click on the SpyStore_HOL.Tests project, and select Add -> Reference
2) Select SpyStore_HOL.DAL and SpyStore_HOL.Models

Part 4: Running the Application

1) Set the MVC project as the Startup Project.
2) Examine launchSettings.json under the Properties node in the SpyStore_HOL.MVC project.
    a) IIS Express profile controls IIS Express in Visual Studio 2017

    b) SpyStore_HOL profile controls Kestrel in Visual Studio 2017

3) Can also run from the command line by entering ‘dotnet run’ from the same directory as the SpyStore_HOL.MVC.csproj file.

Summary

This lab created all of the projects for the HOL, added the NuGet packages, and the appropriate references.

Next steps

In the next part of this tutorial series, you will start to build the data access library using Entity Framework Core.