

# 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.

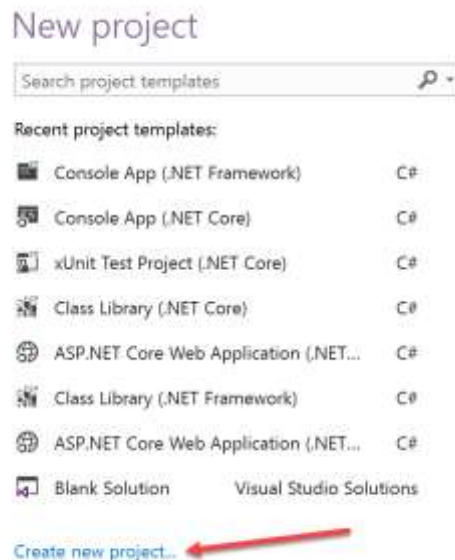
Prior to starting this lab, you must have completed Lab 0, Installing the Prerequisites.

All labs and files are available at [https://github.com/skimedid/dotnetcore\\_hol](https://github.com/skimedid/dotnetcore_hol).

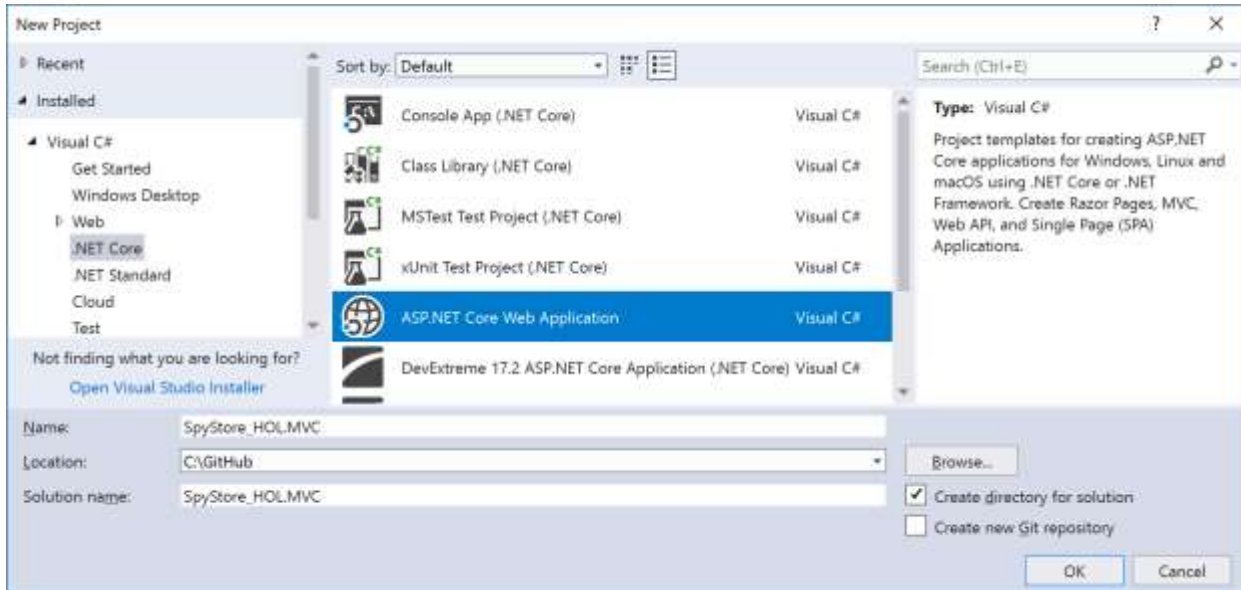
## Part 1: Creating the Solution and Projects

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

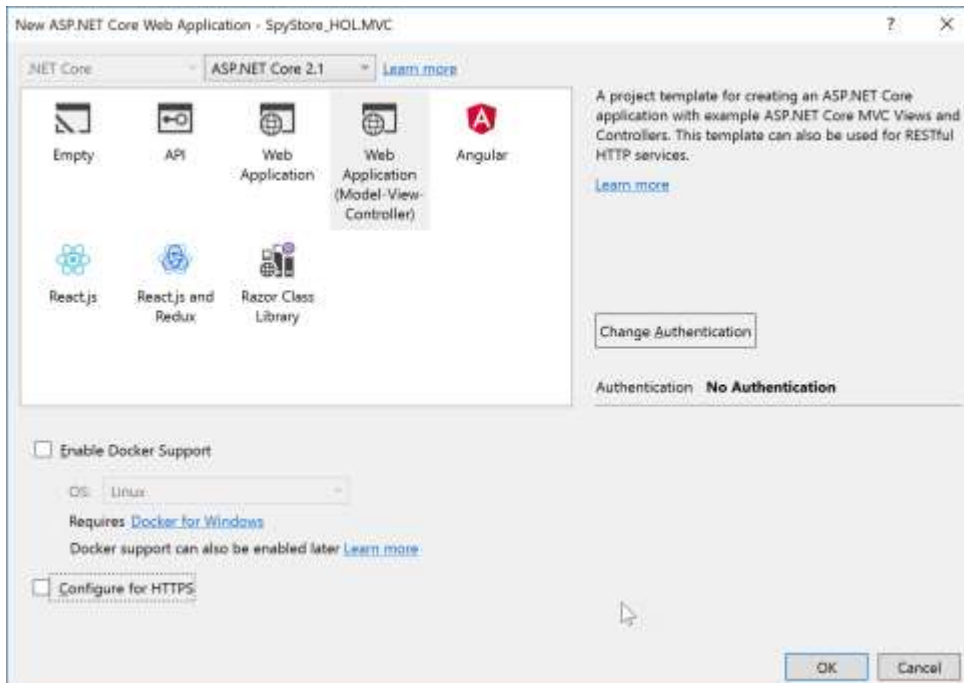
- 1) From the Start Page, select Create New Project



- 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:

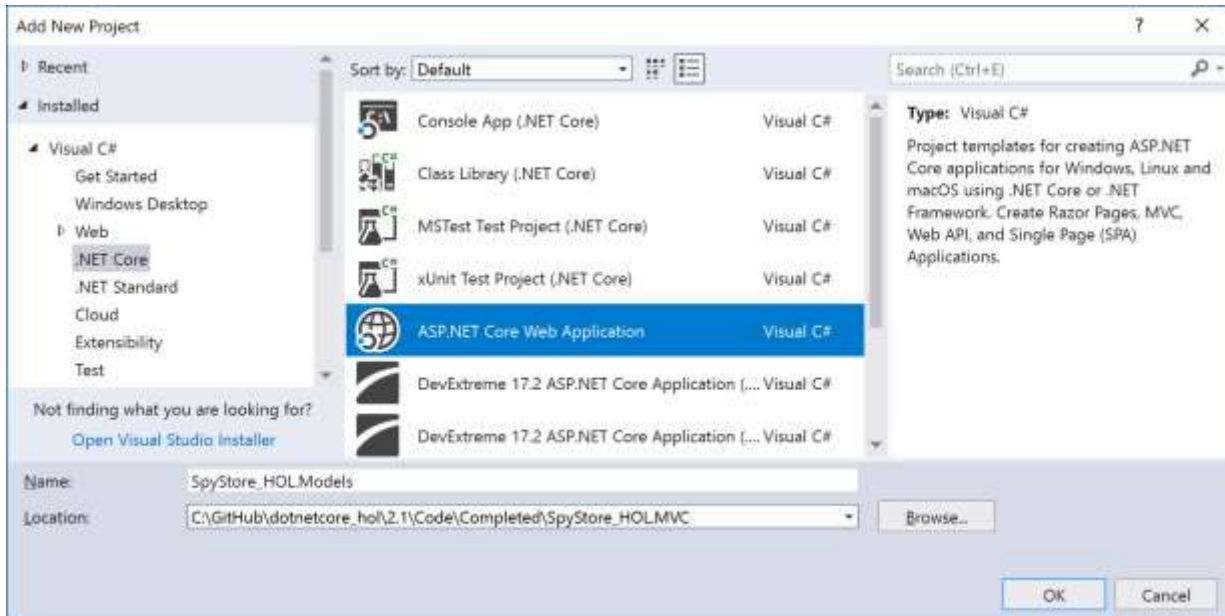


- 3) On the next screen, **make sure the ASP.NET Core 2.1 templates are selected**, then select Web Application (Model-View-Controller). Leave the authentication as “No Authentication” and leave the Enable Docker Support unchecked. Uncheck the configure for HTTPS check box.



## 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:



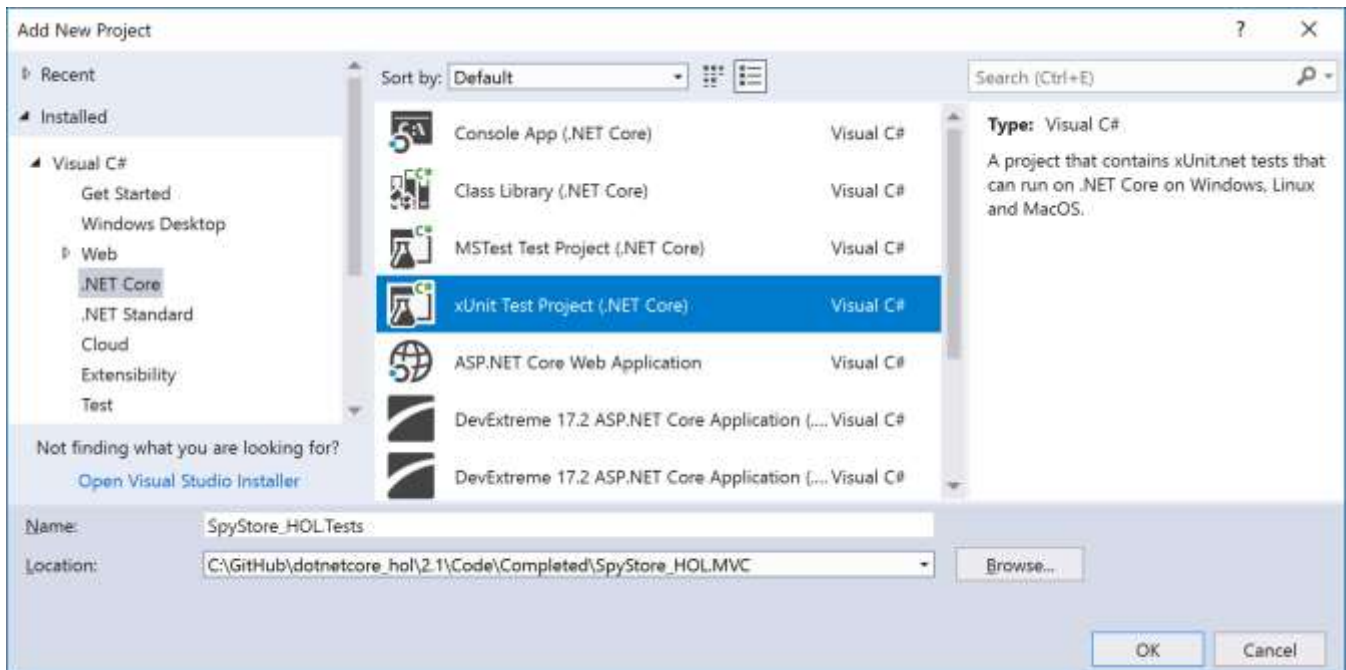
- 3) Delete the Class.cs file.

## 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

- 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 update faster than the VS2017 templates, so in addition to adding new packages, the existing packages must be updated.

**NOTE:** Please make sure not to have “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.
- 3) Examine the list of NuGet packages – on Microsoft.AspNetCore.App and Microsoft.NETCore.App
- 4) The Microsoft.AspNetCore.App package contains everything you need to build and run Asp.NET MVC Core, including EF
- 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.
- 3) Add the Entity Framework Core packages by clicking Browse, then entering Microsoft.EntityFrameworkCore.Abstractions in the search box. Select and install: Microsoft.EntityFrameworkCore.Abstractions 2.1.0

### 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.
- 3) Install the following packages:  
Microsoft.EntityFrameworkCore 2.1.0  
Microsoft.EntityFrameworkCore.Design 2.1.0  
Microsoft.EntityFrameworkCore.Relational 2.1.0  
Microsoft.EntityFrameworkCore.SqlServer 2.1.0

### 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.
- 3) Add the Entity Framework Core packages by clicking Browse, then entering Microsoft.EntityFrameworkCore in the search box. Select and install  
Microsoft.EntityFrameworkCore 2.1.0  
Microsoft.EntityFrameworkCore.SqlServer 2.1.0
- 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)

- 1) Open Package Manager Console (View -> Other Windows -> Package Manager Console) 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**

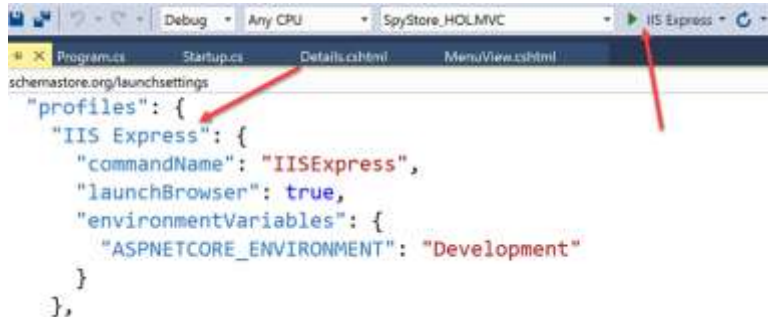
- 1) Right click on the SpyStore\_HOL.DAL project, and select Add -> Reference
- 2) Select SpyStore\_HOL.Models

### **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.