

Build an EF and ASP.NET Core 3.1 App HOL

Lab 0

Welcome to the Build an Entity Framework Core and ASP.NET Core Application in a Day Hands-On Lab. Prior to starting the rest of the workshop, you must have .NET Core 3.1+ SDK, .NET Core 3.1+ runtime, Docker Community (or SQL Server), and an appropriate IDE installed. Supported IDEs includes:

- Visual Studio 2019 16.7.0+
- Visual Studio for the Mac 8.6.6+
- Visual Studio Code 1.47+

SQL Server Management Studio (Windows) or Azure Data Studio (Windows, Mac, Linux) is recommended.

Part 0: Permissions

You must have admin permissions on your machine to complete this hands-on lab.

Part 1: Installing the Prerequisites

Step 1: Install/Confirm .NET Core Runtime and SDK

- 1) Download and install the latest 3.1 .NET Core SDK (3.1.400+) and Runtime (3.1.6+) from <http://dot.net>.
- 2) Open a command window and type:

where dotnet

- 3) After unpacking some files, it should respond with:

```
C:\Program Files\dotnet\dotnet.exe
```

- 4) You might also see the following (not required for this course but required by Visual Studio for Windows):

```
C:\Program Files (x86)\dotnet\dotnet.exe
```

- 5) Check the version of the .NET Core Runtime by entering:

```
dotnet --list-runtimes
```

- 6) The response will be (at the time of this writing). Note: More files will be listed, these are the ones needed for this workshop:

```
Microsoft.AspNetCore.App 3.1.6 [C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App]  
Microsoft.NETCore.App 3.1.6 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]
```

- a) Microsoft.AspNetCore.App leverages the ASP.NET Core shared framework. Any assets in the ASP.NET Core shared framework will not be deployed with your app, and are pre-compiled for better application startup time. Microsoft.AspNetCore.App also uses version roll-forward to work with later versions of the 3.x framework installed on the target machine.

- 7) Check the version of the .NET Core SDK by entering:

```
dotnet -list-sdks
```

- 8) The response will be (at the time of this writing):

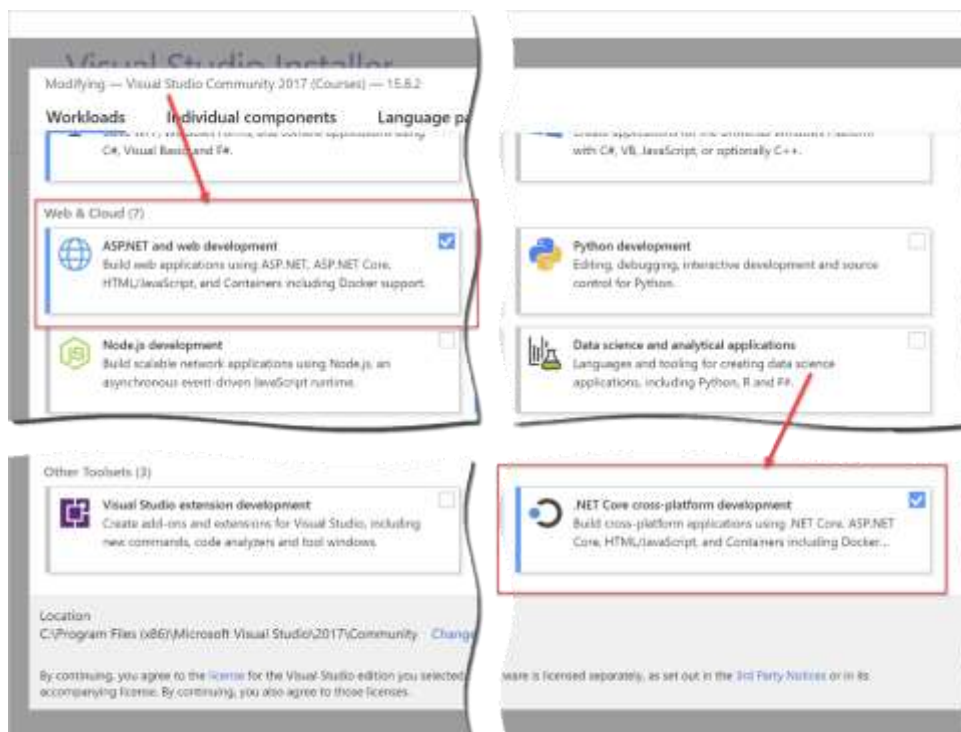
3.1.400 [C:\Program Files\dotnet\sdk]

Step 2: Install an IDE

The HOL will work with Visual Studio 2019, Visual Studio for the Mac, or Visual Studio Code.

Option 1: Download and install Visual Studio 2019

- 1) Download Visual Studio 2019 (any edition) from the Visual Studio home page:
<https://www.visualstudio.com/en-us/visual-studio-homepage-vs.aspx>
 - a) The Community Edition is free, and has everything you need to complete this Hands-On Lab
- 2) Start the installer
 - a) The new installation experience has separate workloads based on what type of work you intend to do. For this lab, select the “**ASP.NET and web development**” workload as well as the “**.NET Core cross-platform development**” workloads.

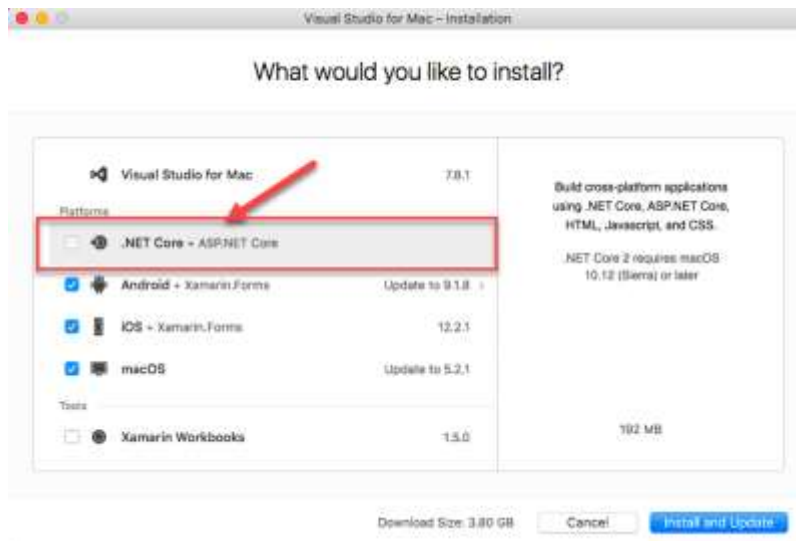


Option 2: Download and install Visual Studio Code

- 1) Download Visual Studio Code from <https://visualstudio.microsoft.com/>.
- 2) Install the “Microsoft C# extension (powered by OmniSharp)”, “.NET Core Debugger (Windows / x64)”, “Razor Language Server (Windows / x64)” extensions.

Option 3: Download and install Visual Studio for the Mac

- 1) Download Visual Studio for the Mac from <https://visualstudio.microsoft.com/>
- 2) Select .NET Core from the install screen (image from 7.8.1)



Step 2: Download and install SQL Server Tooling

Windows: Download/Install SQL Server Management Studio (SSMS)

This is not required for the workshop, but makes it easier to work with the database.

- 1) Download and install SQL Server Management Studio from <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017>
 - a) This is a free tool from Microsoft

Mac/Windows: Download/Install Azure Data Studio

This is not required for the workshop, but makes it easier to work with the database.

- 2) Download and install Azure Data Studio from <https://docs.microsoft.com/en-us/sql/azure-data-studio/download?view=sql-server-2017>
 - a) This is a free tool from Microsoft

Step 3: Install Docker Community

Docker is a containerization platform that runs on Windows, MacOS, and Linux.

- 1) Download and install Docker Desktop from <https://www.docker.com/get-started>. 2.3+ is required.
 - a) When installing, select Linux containers (and not Windows containers)
 - b) This is a free tool but requires you to have a Docker user id and password

Step 4: Pull the SQL Server Image and Create the Local Container

A Docker image is like a class definition, while a Docker Container is like an instance of that class. To run SQL Server in Docker, you must first pull the image from Docker Hub, and then create a container using that image.

- 1) Pull the SQL Server 2017 for Linux (Ubuntu 18.04) image. Enter the following command:

```
docker pull mcr.microsoft.com/mssql/server:2019-latest
```

- 2) When creating an image, there are two required environment variables, “ACCEPT_EULA” and “SA_PASSWORD”. An optional environment variable “MSSQL_PID” sets the product version. The host port mapping to the image port needs to be set, and a friendly name added. Create the container using the following command (**run as one line**):

- a) **NOTE:** On Windows, use double quotes ("). On Mac and Linux, use single quotes (').

```
docker run -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=P@ssw0rd" -e "MSSQL_PID=Express" -p 5433:1433 --name AutoLot -d mcr.microsoft.com/mssql/server:2019-latest
```

Step 5: [Optional] Download and install SQL Server 2019 Developer Edition

(Required if NOT using Docker and you don't already have a SQL Server instance installed)

- 1) Download the SQL Server 2019 Developer Edition from:
<https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

Summary

These are all of the tools you need to complete this Hands-on Lab.