.NET App Dev Hands-On Lab Prerequisites

Prior to starting the rest of the workshop, you must have the .NET (Core) SDK, .NET Runtime, access to a local SQL Server Database, an appropriate .NET development IDE, and an SQL Server IDE.

.NET/IDE Version Choices

You can do this workshop using .NET 6 or .NET 7.

.NET 6 versions:

- SDK => 6.0.100+ (6.0.400+ recommended)
- Runtime => 6.0.0+ (6.0.14 recommended)
- Visual Studio =>

.NET Runtime Version	.NET SDK Version	VS Win Version	VS Mac Version
6.0.14	6.0.1XX	2022 17.0	2022 17.4
6.0.14	6.0.3XX	2022.17.2	2022.17.4
6.0.14	6.0.4XX	2022 17.3+	2022 17.4

See https://dotnet.microsoft.com/en-us/download/dotnet/6.0 for the rest of the version correlations.

.NET 7 versions:

- SDK => 7.0.100+,
- Runtime => 7.0.0+
- Visual Studio

.NET Runtime Version	.NET SDK Version	VS Win Version	VS Mac Version
7.0.3	7.0.1XX	2022 17.4	2022 17.4
7.0.3	7.0.2XX	2022.17.5	2022.17.5

Supported .NET development IDEs include (one is required):

- [Windows] See above for version information
- [macOS] See above for version information
- [Any OS] Visual Studio Code 1.75.0+

SQL Server

SQL Server options (one is required):

- [Any OS] Docker Community with SQL Server 2019,
- [Windows] SQL Server LocalDb (installed with Visual Studio 2022), or
  [Windows] SQL Server 2019/2022 Developer Edition (or above)

Supported SQL Server IDEs include (one is recommended):

- [Windows] SQL Server Management Studio 18.11 +
- [Any OS] Azure Data Studio 1.41.0+

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 Runtime and SDK

- Download and install the latest .NET SDK, ASP.NET Core Runtime, and .NET Runtime from http://dot.net (the .NET Desktop Runtime is not used for the hands-on lab).
- Check the version of the .NET Runtime by entering:

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

- Microsoft.AspNetCore.App leverages the ASP.NET Core shared framework. Any ASP.NET Core shared framework assets 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 the later versions of the .NET Core framework installed on the target machine.
- Check the version of the .NET SDK by entering:

```
dotnet --list-sdks
```

Step 2: Install a Development IDE (one is required)

Option 1: Download and install any edition of Visual Studio 2022

- If you already have the correct version of Visual Studio 2022 installed or don't plan on using VS 2022, continue to the next step.
- Download Visual Studio 2022 (any edition) from the Visual Studio home page: https://www.visualstudio.com

a) The Community Edition is free and has everything you need to complete this Hands-On Lab

- Start the installer

b) 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 "**Data storage and processing**" workloads. The "**.NET desktop development**" workload is not used in this lab, but you can use it to experiment with the new WPF and Winforms frameworks.

Option 2: Download and install Visual Studio Code

- If you already have the latest version of VS Code installed or don't plan on using VS Code, continue to the next step.
- Download Visual Studio Code from https://visualstudio.microsoft.com/.

Install the "Microsoft C# extension (powered by OmniSharp)" extension.

Option 3: Download and install Visual Studio for the Mac

- If you already have the latest version of VS for the Mac installed or don't plan on using VS Mac, continue to the next step.
- Download Visual Studio for the Mac from https://visualstudio.microsoft.com/
- Select .NET Core from the install screen

Step 3: Install SQL Server IDE (SSMS or ADS)

Neither of these are required for the workshop but having one installed makes it easier to work with the database. You only need to install one.

Option 1: SQL Server Management Studio (Windows only)

Download/Install SQL Server Management Studio (SSMS) from https://aka.ms/ssmsfullsetup

Option 2: Azure Data Studio (Mac/Windows)

Download/Install the free Azure Data Studio from https://learn.microsoft.com/en-us/sql/azure-data-studio/download-azure-data-studio?view=sql-server-ver16

Step 4: SQL Server

You must have access to SQL Server 2019+ for this workshop. If you are on a Windows machine, the easiest option is to use LocalDb which is installed with Visual Studio 2022. If you have another edition installed on your machine, you have completed the prerequisites for this workshop.

NOTE: Check that the version of LocalDb is 2019 or greater by running

```
SELECT @@Verion
```

The command should return the following (or higher):

```
Microsoft SQL Server 2019 (RTM-CU12) (KB5004524) - 15.0.4153.1 (X64)
```

If your version is not 2019, select SQL Server Express 2019 LocalDB from the Individual Components. It will most likely be installed as (loacldb)\ProjectModels.

NOTE: This workshop is built using SQL Server 2019, but SQL Server 2022 will work as well.

Option 1: Download and install SQL Server 2022 Developer (Windows)

- ☐ Download/Install SQL Server 2022 Developer Edition: https://go.microsoft.com/fwlink/p/?linkid=2215158

Option 2: Install Docker Desktop (Windows/Mac/Linux)

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

NOTE: If you are using a Windows-based machine, Docker is optional for this workshop. If you are not on a Windows/Linux machine or can't have SQL Server 2019 installed on your laptop, Docker is required.

- ☐ Download and install Docker Desktop from https://www.docker.com/products/docker-desktop
    a) Select the edition for your operating system (Windows/Mac)
    b) During installation, when prompted about what type of containers to use, select Linux containers (and not Windows containers), even if you are on a Windows machine. This type of container will be loaded and is not related to your computer's operating system.
    c) This is a free tool but requires you to have a Docker user id and password
- ☐ 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 create a container using that image.

- ☐ Pull the SQL Server 2019 for Linux image. Enter the following command:

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


- 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:
    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" -p 5433:1433 --name AutoLot -d
mcr.microsoft.com/mssql/server:2019-latest
```

Summary

These are all the tools you need to complete this Hands-on Lab. On lab day you will get the url for the repo to clone/download.