# Steps in Visual Studio 2017

1. Open or create the ASP.NET Core Web Application (Model-View-Controller)
2. The following packages are needed. Use the NuGet package manager to install them. Ensure that they match the .NET Core version that you are using.

|  |  |
| --- | --- |
| Microsoft.EntityFrameworkCore | Entity Framework Core is a lightweight and extensible version of the popular Entity Framework data access technology. |
| Microsoft.EntityFrameworkCore.SqlServer | Microsoft SQL Server database provider for Entity Framework Core. |
| Microsoft.EntityFrameworkCore.SqlServer.Design | Design-time Entity Framework Core Functionality for Microsoft SQL Server. |
| Microsoft.EntityFrameworkCore.Tools | Entity Framework Core Package Manager Console Tools. Includes Scaffold-DbContext, Add-Migration, and Update-Database. |
| Microsoft.VisualStudio.Web.CodeGeneration.Design | Code Generation tool for ASP.NET Core. Contains the dotnet-aspnet-codegenerator command used for generating controllers and views |

1. For good measure, you should close and reopen Visual Studio

## Get the Connection String

1. Use the SQL Server Object Explorer to connect to the existing database. This assumes you have authorization. The sample database is on cscidbw.etsu.edu and uses SQL server authentication with user ‘csci3110User’ and password ‘Csci3110’.
2. After connecting, view the properties of the connection and then copy the Connection string. Here is the sample.

|  |
| --- |
| Data Source=cscidbw.etsu.edu;Integrated Security=False;User ID=csci3110user;Password=\*\*\*\*\*\*\*\*;Connect Timeout=30;Encrypt=False;TrustServerCertificate=True;ApplicationIntent=ReadWrite;MultiSubnetFailover=False |

1. If one does not exist, create the Models > Entities > Existing folder structure to the project

## Scaffold the Entities and DbContext

1. Open the Package Manager Console and then issue the Scaffold-DbContext command. Here is the sample.

|  |
| --- |
| Scaffold-DbContext "Data Source=cscidbw.etsu.edu;Integrated Security=False;User ID=csci3110user;Password=Csci3110;Connect Timeout=30;Encrypt=False;TrustServerCertificate=True;ApplicationIntent=ReadWrite;MultiSubnetFailover=False" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models/Entities/Existing |

## Moving the Connection String out of the Source Code

1. Locate the connection string literal that is used in the DbContext and then move it to appsettings.json. Here is the sample.

|  |
| --- |
| ~~protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)~~  ~~{~~  ~~optionsBuilder.UseSqlServer(@"Data Source=cscidbw.etsu.edu;Integrated Security=False;User ID=csci3110user;Password=Csci3110;Connect Timeout=30;Encrypt=False;TrustServerCertificate=True;ApplicationIntent=ReadWrite;MultiSubnetFailover=False");~~  ~~}~~ |
| {  **"ConnectionStrings": {**  **"RecommendationDb": "Data Source=cscidbw.etsu.edu;Integrated Security=False;User ID=csci3110user;Password=Csci3110;Connect Timeout=30;Encrypt=False;TrustServerCertificate=True;ApplicationIntent=ReadWrite;MultiSubnetFailover=False"**  **},**  "Logging": {  "IncludeScopes": false,  "LogLevel": {  "Default": "Warning"  }  }  } |

1. The DbContext is then configured in Startup.

|  |
| --- |
| public void ConfigureServices(IServiceCollection services)  {  // Add framework services.  services.AddMvc();  services.AddDbContext<RecommendationDbContext>(options =>  options.UseSqlServer(Configuration.GetConnectionString("RecommendationDb")));  } |

1. The DbContextOptions need to be sent to the base DbContext.

|  |
| --- |
| public RecommendationDbContext(DbContextOptions options) : base(options)  {  } |