I built this app using the EF Core 2 tutorial found at:

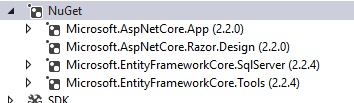
<https://docs.microsoft.com/en-us/ef/core/get-started/aspnetcore/new-db?tabs=visual-studio>

<https://www.entityframeworktutorial.net/efcore/entity-framework-core.aspx>

I created a basic (barebones) ASP.NET Core app, then added the following NuGet packages:

* Microsoft.EntityFrameworkCore.SqlServer
* Microsoft.EntityFrameworkCore.Tools
* AutoMapper
* AutoMapper.Extensions.Microsoft.DependencyInjection

Screenshot:



To configure AutoMapper, I used this:

<https://stackoverflow.com/questions/40275195/how-to-setup-automapper-in-asp-net-core>

# Create the database

The following steps use migrations to create a database.

From the main menu select: Tools > NuGet Package Manager > Package Manager Console, and run the following powershell commands:

Add-Migration InitialCreate

Update-Database

The Add-Migration command scaffolds a migration to create the initial set of tables for the model.

The Update-Database command actually runs the migration against SQL Server (i.e.: creates the database and applies the new migration to it).

**Note:** You may get an error saying that it can’t create the ProjectLists table because the table already exists.

There is already an object named 'ProjectLists' in the database.

You don’t actually need the Update-Database command, since the service will create the DB and apply the InitialCreate migration when it starts up.

You can also go into the InitialCreate class, and comment out the body for the Up() method. Then you can run Update-Database and it will succeed without problems. You can read more on [this SO question](https://stackoverflow.com/questions/43687433/update-database-command-is-not-working-in-asp-net-core-entity-framework-6-beca/43687656#43687656).

**Note:** If you get an error stating The term 'add-migration' is not recognized as the name of a cmdlet, close and reopen Visual Studio.