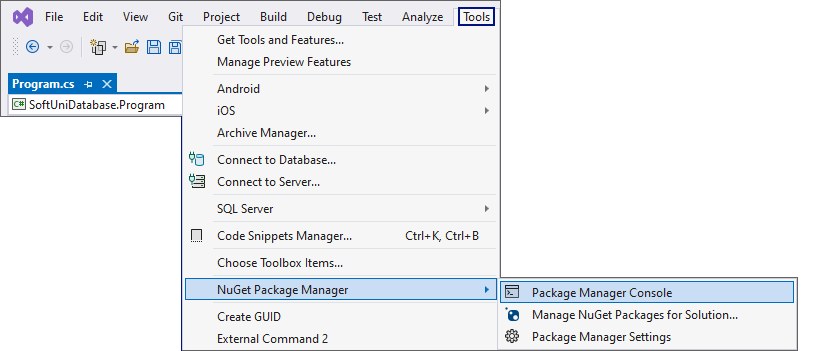
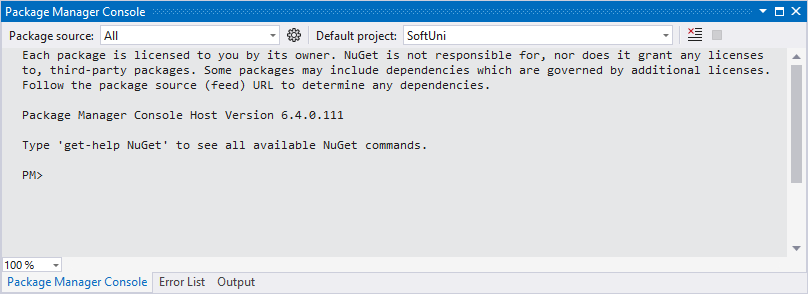
## Database First

Model the existing database by using the Database First approach.

First create a new empty **.NET Core** **Console Application** and after it is created open the **Package Manager Console**



It will look something like this:



Use it to run the following commands **one by one**:

|  |
| --- |
| Install-Package Microsoft.EntityFrameworkCore.Tools –v 6.0.1  Install-Package Microsoft.EntityFrameworkCore.SqlServer –v 6.0.1  Install-Package Microsoft.EntityFrameworkCore.Design -v 6.0.1 |

These are the **packages** you will need, in order to **scaffold** our **SoftUniContext** from the **SoftUni** **database**.

**NOTE:** If **Package Manager Console** gives you any **errors** while trying to **execute the above commands**, try using the following ones:

|  |
| --- |
| Install-Package Microsoft.EntityFrameworkCore.Tools –Version 6.0.1  Install-Package Microsoft.EntityFrameworkCore.SqlServer –Version 6.0.1  Install-Package Microsoft.EntityFrameworkCore.Design -Version 6.0.1 |

**NOTE:** For the next steps you will need **public Program class** and a **static void Main()** method.

Next, we must **execute** the **command** to **scaffold** our **context** **class**. It will consist of 4 things:

1. First, the name of the command:

|  |
| --- |
| Scaffold-DbContext |

1. Second, the connection we will be using (our connection string):

|  |
| --- |
| -Connection "Server=<ServerName>;Database=<DatabaseName>;Integrated Security=True;" |

For **ServerName**, use the name of your local MS SQL Server instance or ".".

For **DatabaseName**, use the name of the database you want to use, in this case – **SoftUni**.

1. Third, we need to declare our service provider, we'll be using **Microsoft.EntityFrameworkCore.SqlServer**:

|  |
| --- |
| -Provider Microsoft.EntityFrameworkCore.SqlServer |

1. And the fourth thing we'll do, is to give it a directory where all of our models will go (e.g. **Models**):

|  |
| --- |
| -OutputDir Data/Models |

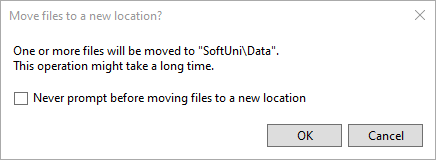
Our final command will look like this:

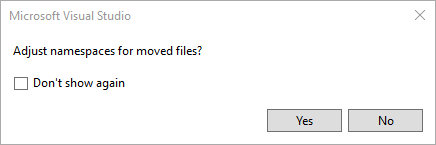
|  |
| --- |
| Scaffold-DbContext -Connection "Server=.;Database=SoftUni;Integrated Security=True;" -Provider Microsoft.EntityFrameworkCore.SqlServer -OutputDir Data/Models |

Execute the **whole command** on a **single line.**

**NOTE:** You should write only one '**\**' in the connection string. Otherwise, an **InvalidOperationException: Instance failure** will be thrown.

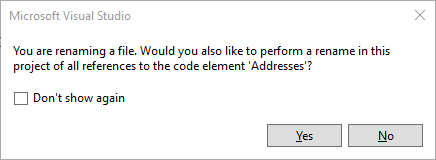
Entity Framework Core has successfully **mapped the database schema to C# classes**. Use the **Solution Explorer** in Visual Studio to move the **SoftUniContext** class out of **Models** intothe **Data** folder and move the **Models** folder out of the **Data** folder into the project's directory. Press OK on both of the pop-up windows that will be shown.



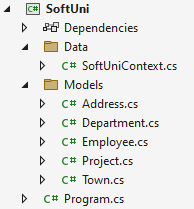


This way Visual Studio will also **rename** the **classes** **everywhere** they're used.

However, sometimes Entity Framework Core isn't good enough with names – all classes may have been pluralized. If this is your case, just rename all of our classes properly. Use **right-click → Rename** or the **F2** shortcut and press **OK** on this **pop**-**up** **window** after renaming each class:



The final result should look like this:



Don't forget to check the **SoftUniContext**'s namespace after moving it and to add a reference to the **Models** and **Data** namespaces:

**Make** **sure** that your namespaces are **exactly** the same as these:

|  |
| --- |
| SoftUni SoftUni.Data  SoftUni.Models |

Finally, we want to clean up the packages we won't be using anymore from the package manager GUI or by running these commands **one by one**:

|  |
| --- |
| Uninstall-Package Microsoft.EntityFrameworkCore.Tools -r  Uninstall-Package Microsoft.EntityFrameworkCore.Design -RemoveDependencies |