

# Guide for Enabling Entity Framework in Your Projects

## 1) Installing Entity Framework

To be able to use Entity Framework, we need to install it and then enable it with several configurations.

We will install Entity Framework by using NuGet Packages, which is defined as “a Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects that use the .NET Framework.” (<https://www.nuget.org/>).

Please right-click on your project name in the Solution Explorer and choose **Manage NuGet Packages** as seen in the following figure.

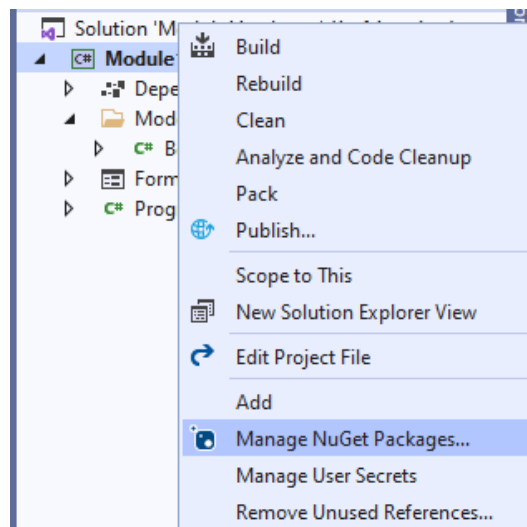


Figure 1. Manage NuGet Packages item

In the *Browse* tab, please search for **EntityFrameworkCore**. When Microsoft.EntityFrameworkCore appears in the list, please select it and click Install. See Figure 2.

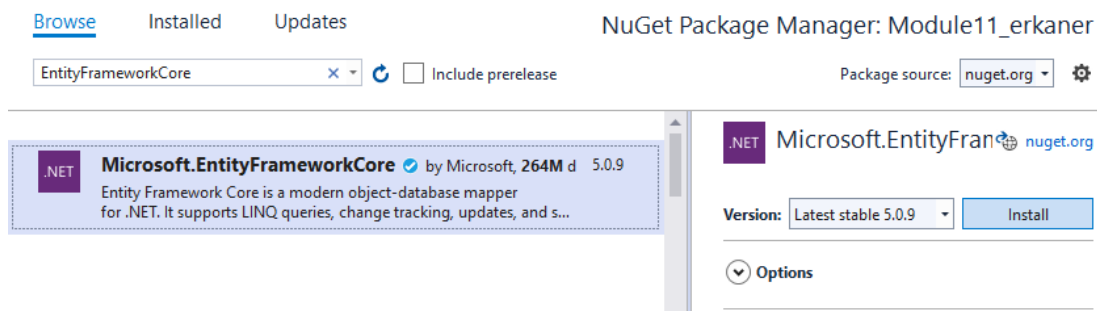


Figure 2. Installing EntityFrameworkCore

If any additional windows appear for confirmation, please click **I Agree**. Once installed, a green checkbox should appear next to Microsoft.EntityFrameworkCore, as seen in Figure 3 below.

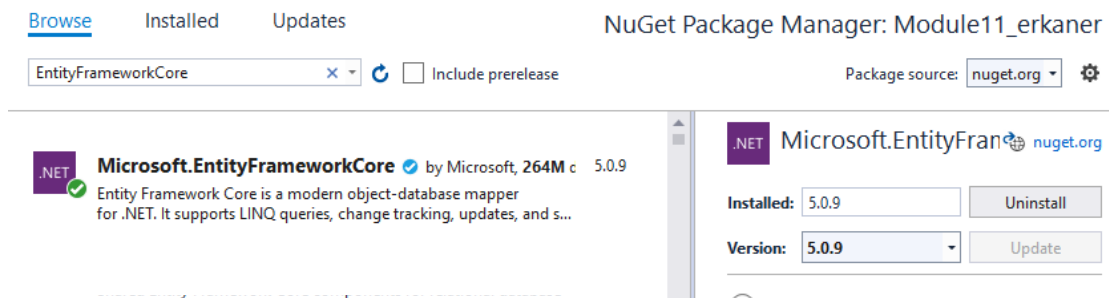


Figure 3. EntityFrameworkCore is installed.

Similarly, please install EntityFrameworkCore.Tools and EntityFrameworkCore.Design, as shown in Figure 4 below.

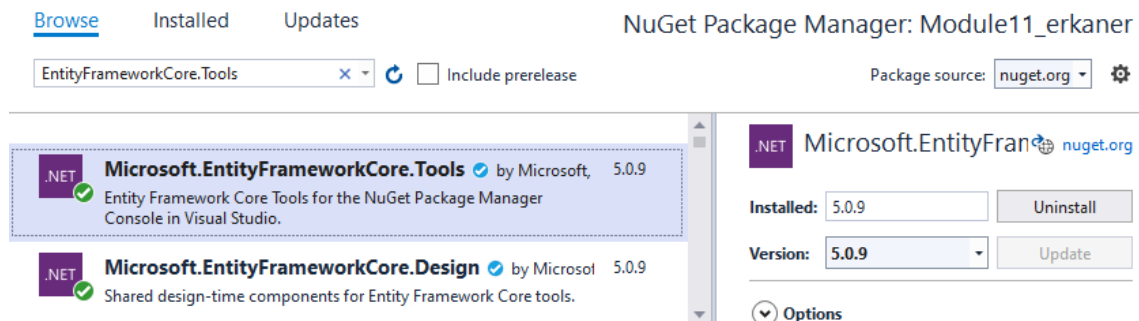


Figure 4. Installing other components of EntityFrameworkCore.

Last, you should install the Microsoft.EntityFrameworkCore.SqlServer library, as shown in Figure 5.

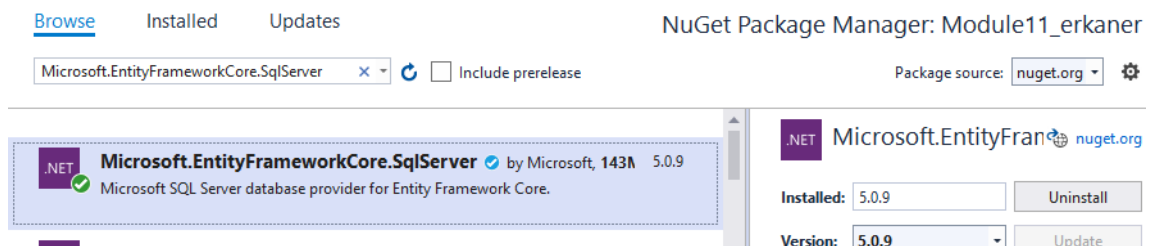


Figure 5. Installing other components of EntityFrameworkCore.SqlServer.

## 2) Creating the DbContext file

To be able to use EntityFramework, you need to configure it. To do that, you need to add a new class that inherits from **DbContext** class.

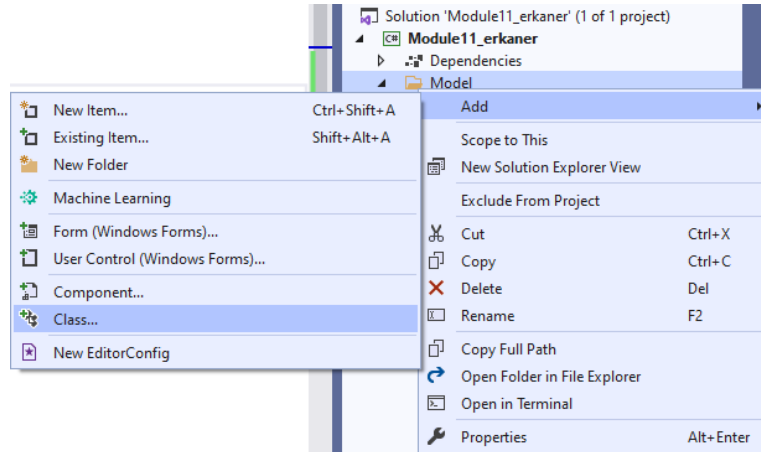


Figure 6. Adding a new class.

Please name the class file as **BookDbContext** (see Figure 7). Including the DbContext keyword at the end of the class name is a good practice.

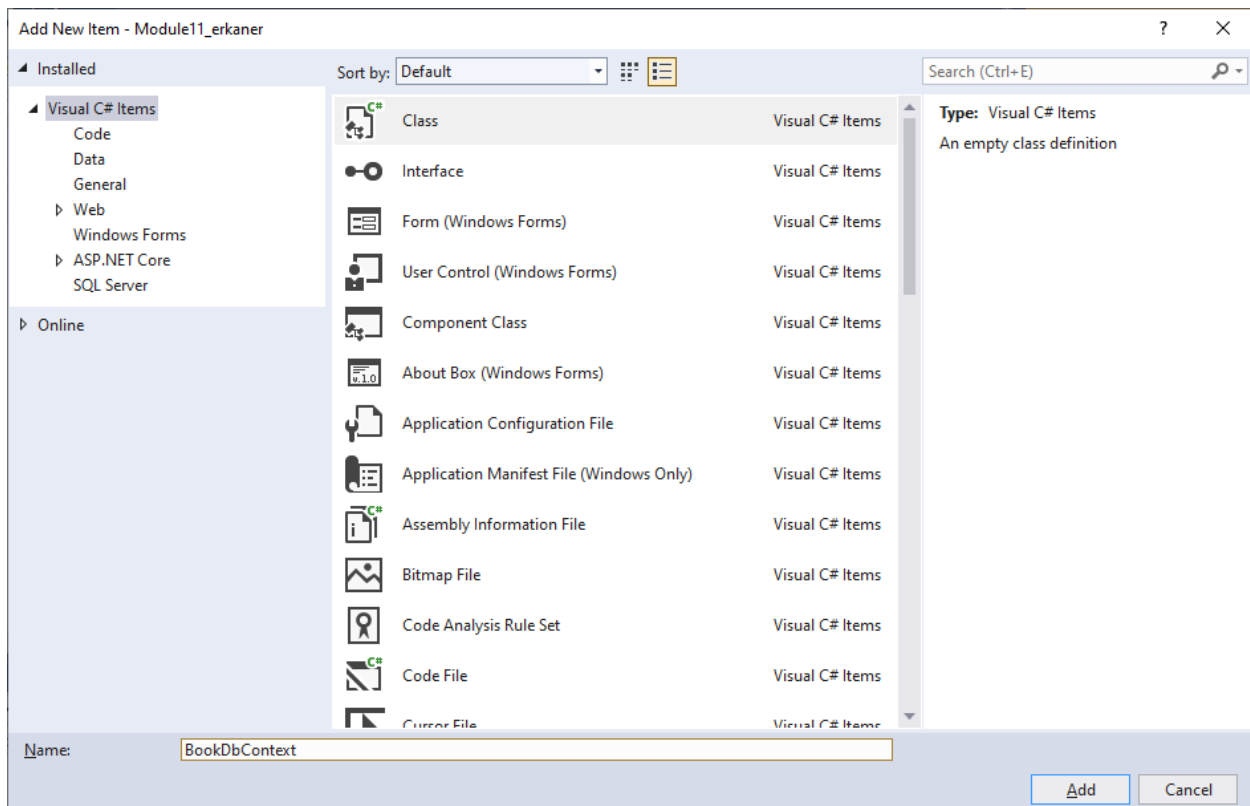
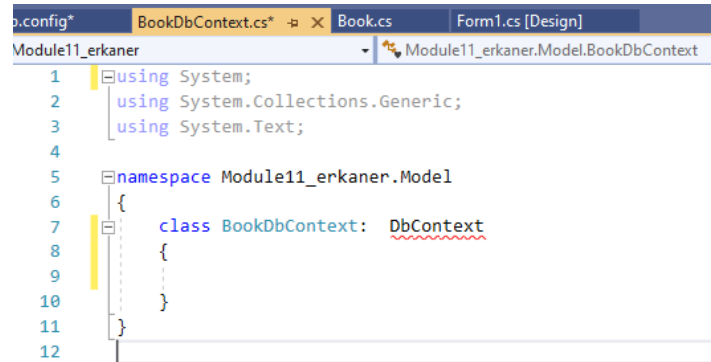


Figure 7. Creating the BookDbContext class file.

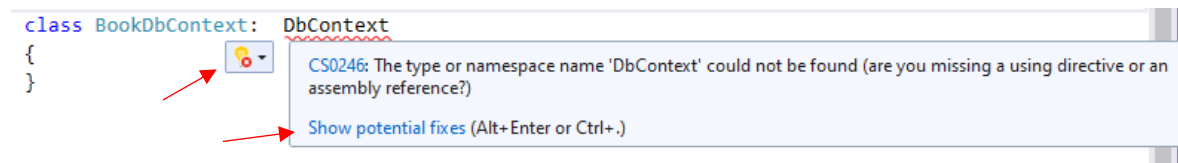
Please write the code to inherit `BookDbContext` from `DbContext` as shown below. Initially, a jagged red line should appear under `DbContext`, indicating an error. See Figure 8 below.



```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4
5 namespace Module11_erkaner.Model
6 {
7     class BookDbContext: DbContext
8     {
9     }
10 }
11
12
```

Figure 8. Inheriting from `DbContext`.

Once you mouse over it, you should see the error messages displayed in a tooltip. The error message tells that `DbContext` class is not accessible because it is not properly imported with using statements. Visual Studio can automatically fix this error. Please click one of the items indicated with arrows in Figure 9.



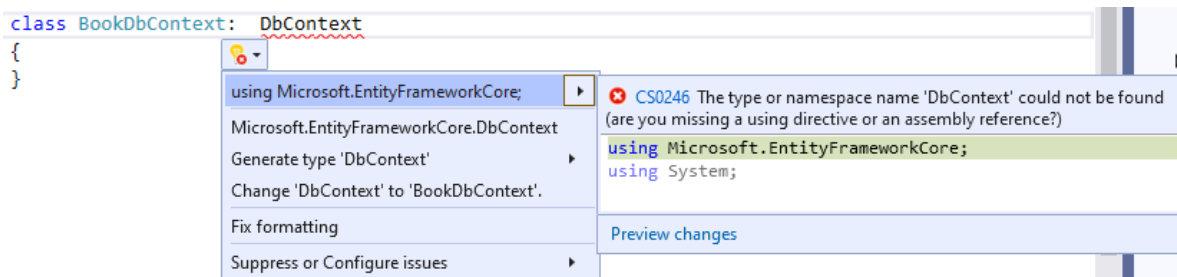
```
class BookDbContext: DbContext
{
}
```

CS0246: The type or namespace name 'DbContext' could not be found (are you missing a using directive or an assembly reference?)

Show potential fixes (Alt+Enter or Ctrl+.)

Figure 9. Displaying the potential fix window.

The potential fixes should be displayed in a small popup window as shown in Figure 10. In this window, choose the first item (**using Microsoft.EntityFrameworkCore**) and click on it twice to apply the automatic fix.



```
class BookDbContext: DbContext
{
}
```

- using Microsoft.EntityFrameworkCore;
- Microsoft.EntityFrameworkCore.DbContext
- Generate type 'DbContext'
- Change 'DbContext' to 'BookDbContext'.
- Fix formatting
- Suppress or Configure issues

CS0246 The type or namespace name 'DbContext' could not be found (are you missing a using directive or an assembly reference?)

```
using Microsoft.EntityFrameworkCore;
using System;
```

Preview changes

Figure 10. Applying the potential fix.

Please add the `OnConfiguring` method (as shown below) to your `DbContext` class. You will need to update the *Database* name in the connection string passed to the `UseSqlServer` method. For this example, the database name will be `BookManagement_v1`.

```
class BookDbContext: DbContext
{
    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        optionsBuilder.UseSqlServer(@"Server=(localdb)\mssqllocaldb;Database=BookManagement_v1");
    }
}
```

For each class for which you want a table in the database, please define a DbSet variable inside the DbContext file. In this example, **Books** variable is created as shown below.

```
class BookDbContext: DbContext
{
    public DbSet<Book> Books { get; set; } //a proxy to the Books table in the database

    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        optionsBuilder.UseSqlServer(@"Server=(localdb)\mssqllocaldb;Database=BookManagement_v1");
    }
}
```

### 3) Using migrations to create the database for the first time

Please open **Package Manager Console**. To open this console, click on *View* menu, choose *Other Windows*, and then click on *Package Manager Console* in the submenu. This is illustrated in Figure 11 below.

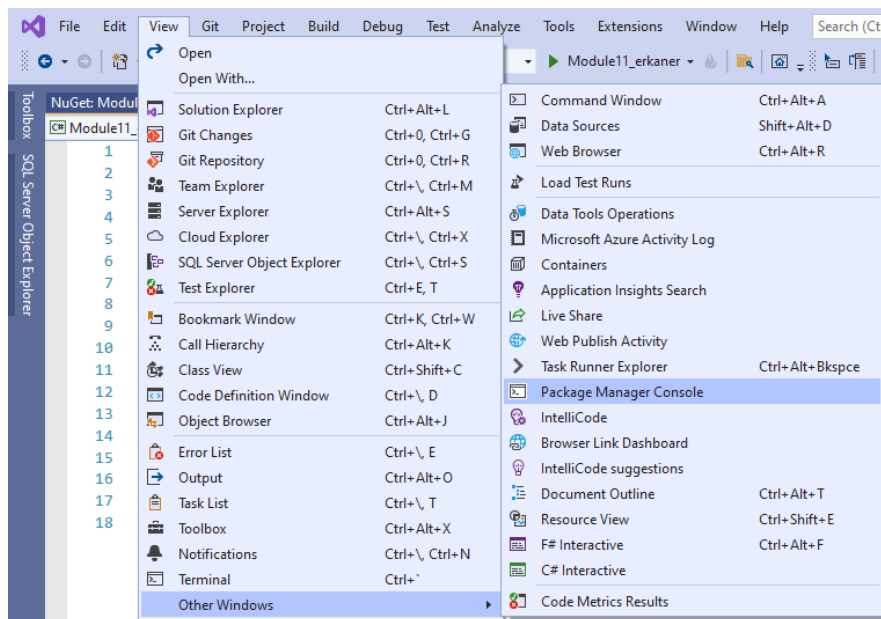


Figure 11. Opening Package Manager Console.

Please run the Add-Migration command, type a name for the migration and press enter.

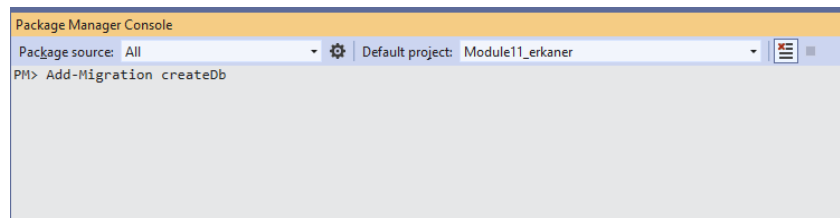


Figure 12. Adding a migration.

After running the Add-Migration command, you should see the migration file automatically opened in VS (see Figure 13). This file contains all changes to be made to the database.

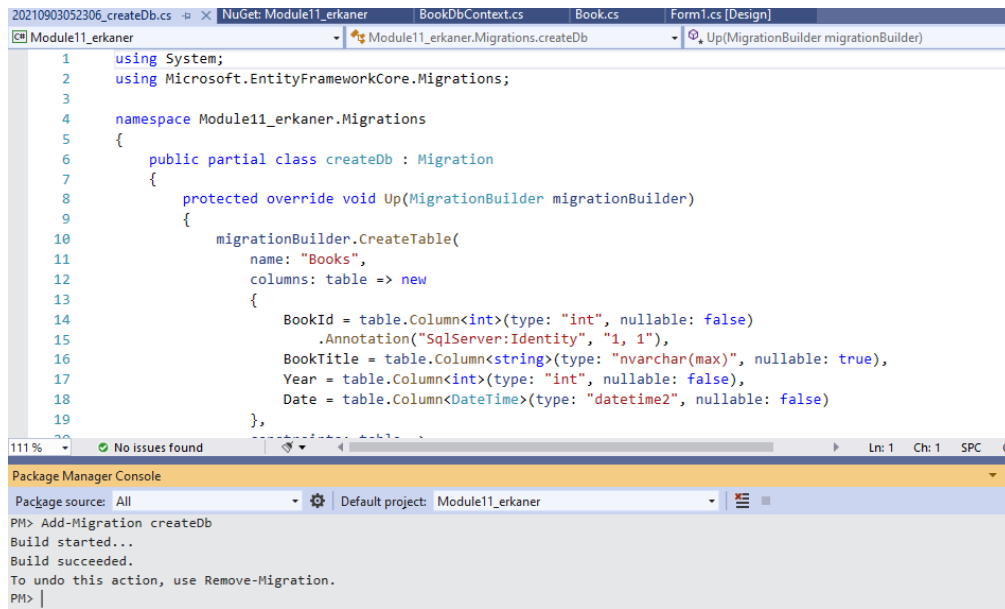


Figure 13. The using statement is added.

To apply the changes in the migration file, please run the **Update-Database** command and press enter as shown in Figure 14.

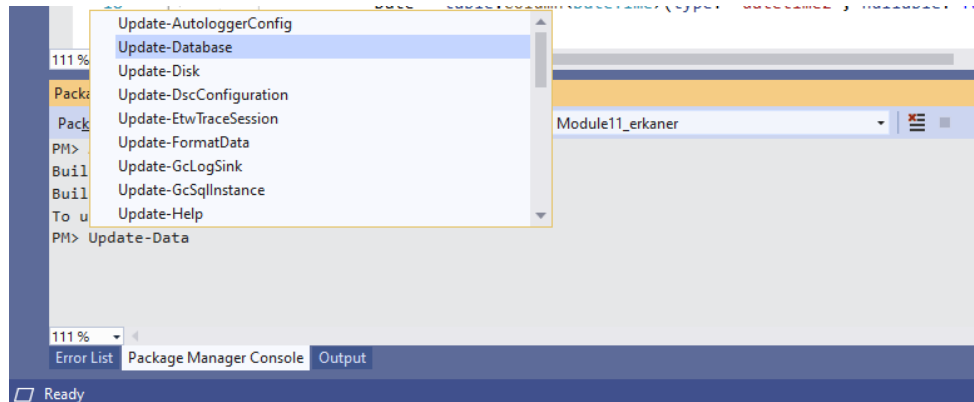


Figure 14. The using statement is added.

Once the command is executed, you should obtain the following output as shown in Figure 15. If you see Done in the output without any error messages, then the database should be created successfully.

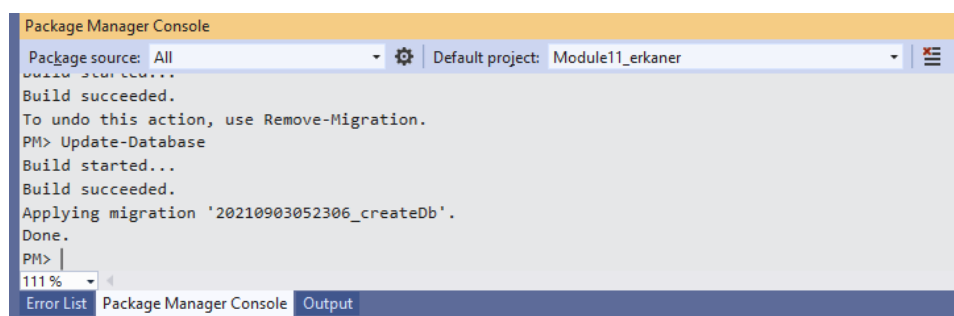


Figure 15. The using statement is added.

To see the database, please check the SQL Server Object Explorer window. The **BookManagement\_v1** database should be available under the Databases folder, and the database should have the **Books** table. See Figure 16.

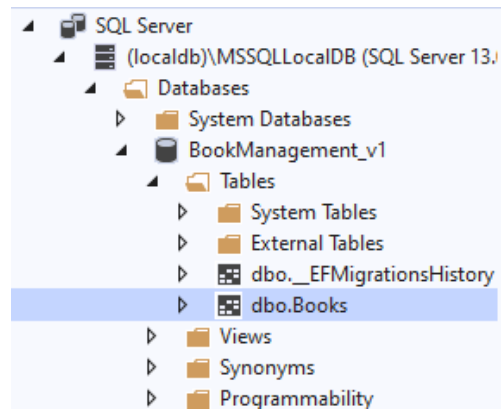


Figure 16. The using statement is added.