**how to install Entity Framework Core in Visual Studio.**  
  
  
Depending on how you have your project set up, you may have Entity Framework Core already installed.  
  
  
**Single Layer Web Application**  
  
If it's a small project, you may have your presentation layer, business layer and data access layer all in one project. So if you have created a web application project using ASP.NET Core 2.1 or higher, then in that web application project, you already have Entity Framework Core installed.  
  
**Entity Framework Core in ASP.NET Core Web Application Project**  
  
An ASP.NET Core Web application project that is created using ASP.NET Core 2.1 or higher has the following NuGet package installed.

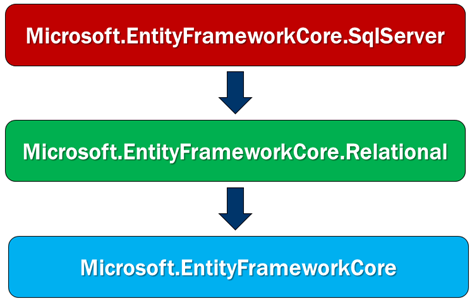
Microsoft.AspNetCore.App

This package is called - **metapackage**. A metapackage has no content of its own but is a list of dependencies (other packages). You can find this metapackage, in the Solution Explorer. When you expand the metapackage, you can find all the dependencies. In the dependencies you will find the Entity Framework Core nuget packages already installed.  
  
So the point that I am trying to make is, an asp.net core web application project that is created using ASP.NET Core Version 2.1 or later will have Entity Framework Core already installed as part of the meta package.  
  
**Multi Layer Web Application**  
  
In a large application we will usually have at least the following 3 layers

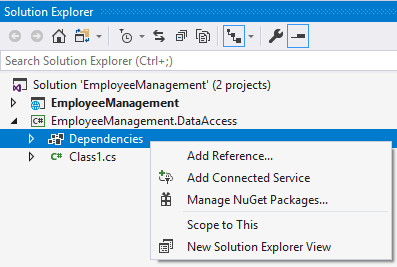
* Presentation Layer
* Business Logic Layer
* Data Access Layer

These layers are implemented as separate projects. Entity Framework Core is usually required in the Data Access Layer project. The Data Access Layer project is a class library project and does not usually have the meta package referenced. So this means, Entity Framework Core is not installed for the Data Access Layer project.  
  
To install Entity Framework Core and to be able to use SQL server as the database for your application, you need to install the following nuget packages.

|  |  |
| --- | --- |
| **Package** | **Purpose** |
| Microsoft.EntityFrameworkCore.SqlServer | This nuget package contains SQL Server specific functionality |
| Microsoft.EntityFrameworkCore.Relational | This nuget package contains functionality that is common to all relational databases |
| Microsoft.EntityFrameworkCore | This nuget package contains common entity frameowrk core functionality |



* Microsoft.EntityFrameworkCore.SqlServer has a dependency on Microsoft.EntityFrameworkCore.Relational package.
* Microsoft.EntityFrameworkCore.Relational package has a dependency on Microsoft.EntityFrameworkCore package.
* Microsoft.EntityFrameworkCore package has a dependency on several other packages.

When we install Microsoft.EntityFrameworkCore.SqlServer package, it also installs all the other dependant nuget packages automatically.   
  
**In a class library project, to install a nuget package**  
  
Right click on the *"Dependencies"* node in *"Solution Explorer"* and Select *"Manage NuGet Packages"* from the context menu.  
  
  
  
On the screen that pops up, search for the package that you want to install and follow the onscreen instructions.  
  
We want to use SQL Server as the database for our application, so we used the nuget package Microsoft.EntityFrameworkCore.SqlServer. This package is usually called Database provider package.  
  
If you want to use a different database with your application, then you will have to install that database provider specific nuget package instead of Microsoft.EntityFrameworkCore.SqlServer database provider package.  
  
For example, if you want to use mysql as your database, then install Pomelo.EntityFrameworkCore.MySql database provider package. Along the same lines, if you want to use PostgreSQL as your database, use Npgsql.EntityFrameworkCore.PostgreSQL database provider package.  
  
You can find all the provider specific NuGet packages on the following MSDN page  
<https://docs.microsoft.com/en-us/ef/core/providers/>