

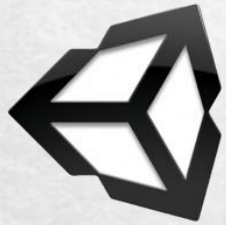
ASP.NET WEB API DEVELOPMENT

Advanced
architecture for
ASP.NET Core
Web API



Md. Hasanuzzaman Dipu
Product Development Manager
Hi-Tech (Bangla) Bangladesh Ltd.

INSTALL ALL IDE, DATABASE & TOOLS



Unity 3D

Unity 3D Game Engine



Visual Studio 2022

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft



MS SQL Server

Microsoft SQL Server is a relational database management system developed by Microsoft.

INSTALL ALL IDE, DATABASE & TOOLS



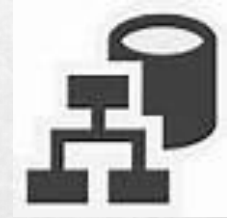
Toad for SQL Server

The Toad Object Explorer allows you to effortlessly see objects from any database on a server instance and display detailed information for each object selected



Toad Data Modeler

Toad Data Modeler is a **database design tool** allowing users to visually create, maintain, and document new or existing database systems, and to deploy changes to data structures across different platforms



EF Core Power Tools

EF Core Power Tools is a **Visual Studio extension** that exposes various EF Core design-time tasks in a simple user interface.

LET'S START

- CREATING ASP.NET CORE WEB API PROJECT
- USING VISUAL STUDIO 2022
- ASP.NET CORE WEB API PROJECT USING .NET CORE CLI

1. WIN + CMD , RUN AS ADMINISTRATOR



2. CREATE A FOLDER WHICH NAME IS CLI

3. CHANGE DRIVE D:

4. GO TO THE FOLDER CD CLI

ONCE YOU TYPE CD CLI AND PRESS THE ENTER KEY, IT WILL CHANGE THE DIRECTORY PATH TO THE CLI FOLDER AS SHOWN IN THE IMAGE.

```
Administrator: Command Prompt
C:\windows\system32>D:
```

```
Administrator: Command Prompt
C:\windows\system32>D:
D:\>CD CLI
```

```
Administrator: Command Prompt
C:\windows\system32>D:
D:\>CD CLI
D:\CLI>
```

CREATING PROJECT FOR API

FIRST, TYPE DOTNET NEW AND PRESS THE ENTER BUTTON AS SHOWN IN THE BELOW IMAGE. ASP.NET CORE WEB API

Once you type **dotnet new** and press the enter button, it will display the following. As you can see in the below image, we have the templates and the short name for the template as well as the default language for the template.

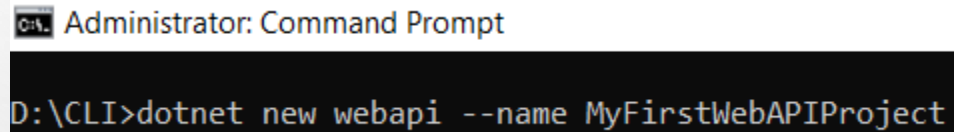
Administrator: Command Prompt

```
D:\CLI>dotnet new
```

Templates	Short Name	Language	Tags
Console Application	console	[C#],F#,VB	Common/Console
Class library	classlib	[C#],F#,VB	Common/Library
WPF Application	wpf	[C#],VB	Common/WPF
WPF Class library	wpflib	[C#],VB	Common/WPF
WPF Custom Control Library	wpfcustomcontrollib	[C#],VB	Common/WPF
WPF User Control Library	wpfusercontrollib	[C#],VB	Common/WPF
Windows Forms App	winforms	[C#],VB	Common/WinForms
Windows Forms Control Library	winformscontrollib	[C#],VB	Common/WinForms
Windows Forms Class Library	winformslib	[C#],VB	Common/WinForms
Worker Service	worker	[C#],F#	Common/Worker/Web
Unit Test Project	mstest	[C#],F#,VB	Test/MSTest
NUnit 3 Test Project	nunit	[C#],F#,VB	Test/NUnit
NUnit 3 Test Item	nunit-test	[C#],F#,VB	Test/NUnit
xUnit Test Project	xunit	[C#],F#,VB	Test/xUnit
Razor Component	razorcomponent	[C#]	Web/ASP.NET
Razor Page	page	[C#]	Web/ASP.NET
MVC ViewImports	viewimports	[C#]	Web/ASP.NET
MVC ViewStart	viewstart	[C#]	Web/ASP.NET
Blazor Server App	blazorserver	[C#]	Web/Blazor
Blazor WebAssembly App	blazorwasm	[C#]	Web/Blazor/WebAssembly
ASP.NET Core Empty	web	[C#],F#	Web/Empty
ASP.NET Core Web App (Model-View-Controller)	mvc	[C#],F#	Web/MVC
ASP.NET Core Web App	webapp	[C#]	Web/MVC/Razor Pages
ASP.NET Core with Angular	angular	[C#]	Web/MVC/SPA
ASP.NET Core with React.js	react	[C#]	Web/MVC/SPA
ASP.NET Core with React.js and Redux	reactredux	[C#]	Web/MVC/SPA
Razor Class Library	razorclasslib	[C#]	Web/Razor/Library
ASP.NET Core Web API	webapi	[C#],F#	Web/WebAPI
ASP.NET Core gRPC Service	grpc	[C#]	Web/gRPC
dotnet gitignore file	gitignore		Config
global.json file	globaljson		Config
NuGet Config	nugetconfig		Config
Dotnet local tool manifest file	tool-manifest		Config
Web Config	webconfig		Config
Solution File	sln		Solution
Protocol Buffer File	proto		Web/gRPC

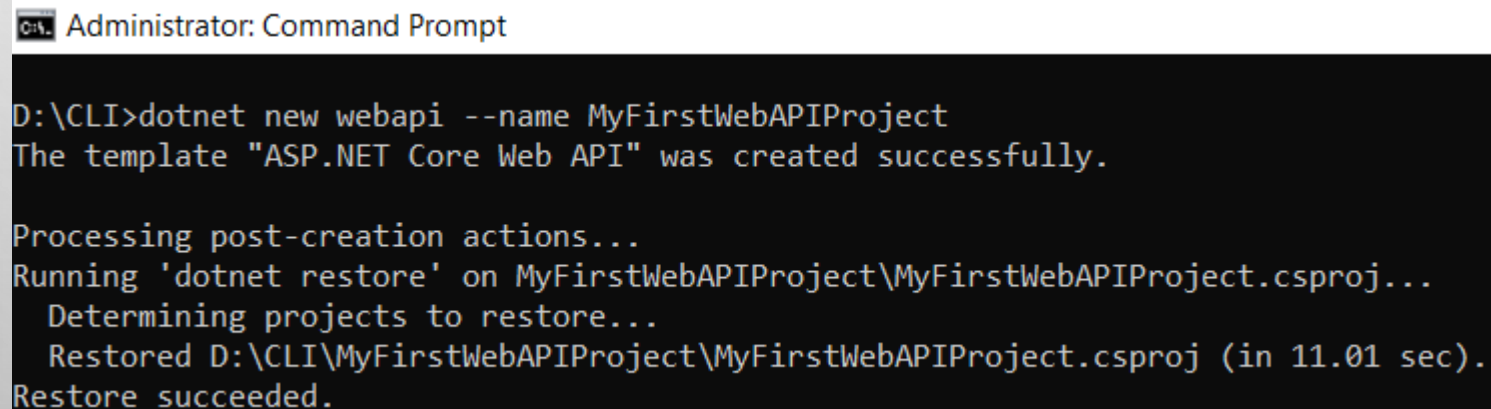
Let us create the web API project. You can create a web API project using two ways. They are as follows:

- ✓ **dotnet new webapi:** If you only type dotnet new webapi, then a new project will be created inside the CLI folder with the default name.
- ✓ **dotnet new webapi --name MyFirstWebAPIProject:** With this command, the .NET Core CLI will create a project inside the CLI folder with the name MyFirstWebAPIProject. So, let us use the second option to create the ASP.NET Core Web API Project. Type **dotnet new webapi --name MyFirstWebAPIProject** command and press enter in the command prompt as shown in the below image.



```
C:\> Administrator: Command Prompt  
D:\CLI>dotnet new webapi --name MyFirstWebAPIProject
```

- ✓ **Once you type dotnet new webapi --name MyFirstWebAPIProject and press enter, you will get the below output saying that the template ASP.NET Core Web API was created successfully.**



```
C:\> Administrator: Command Prompt  
D:\CLI>dotnet new webapi --name MyFirstWebAPIProject  
The template "ASP.NET Core Web API" was created successfully.  
  
Processing post-creation actions...  
Running 'dotnet restore' on MyFirstWebAPIProject\MyFirstWebAPIProject.csproj...  
  Determining projects to restore...  
  Restored D:\CLI\MyFirstWebAPIProject\MyFirstWebAPIProject.csproj (in 11.01 sec).  
Restore succeeded.
```

- Now, inside the CLI Folder, you can see a folder with your project name (in my case the folder name is MyFirstWebAPIProject) and inside that folder, you can see the following default project files and folders provided by the ASP.NET Core Web API template as shown in the below image.

Name	Date modified	Type	Size
bin	12/6/2021 4:25 PM	File folder	
Controllers	12/6/2021 4:25 PM	File folder	
obj	12/6/2021 4:26 PM	File folder	
Properties	12/6/2021 4:25 PM	File folder	
appsettings.Development.json	12/6/2021 4:25 PM	JSON File	1 KB
appsettings.json	12/6/2021 4:25 PM	JSON File	1 KB
Program.cs	12/6/2021 4:25 PM	C# Source File	1 KB
WeatherForecast.cs	12/6/2021 4:25 PM	C# Source File	1 KB
WebApplication1.csproj	12/6/2021 4:25 PM	C# Project file	1 KB

Opening ASP.NET Core Web API project in Visual Studio 2022:

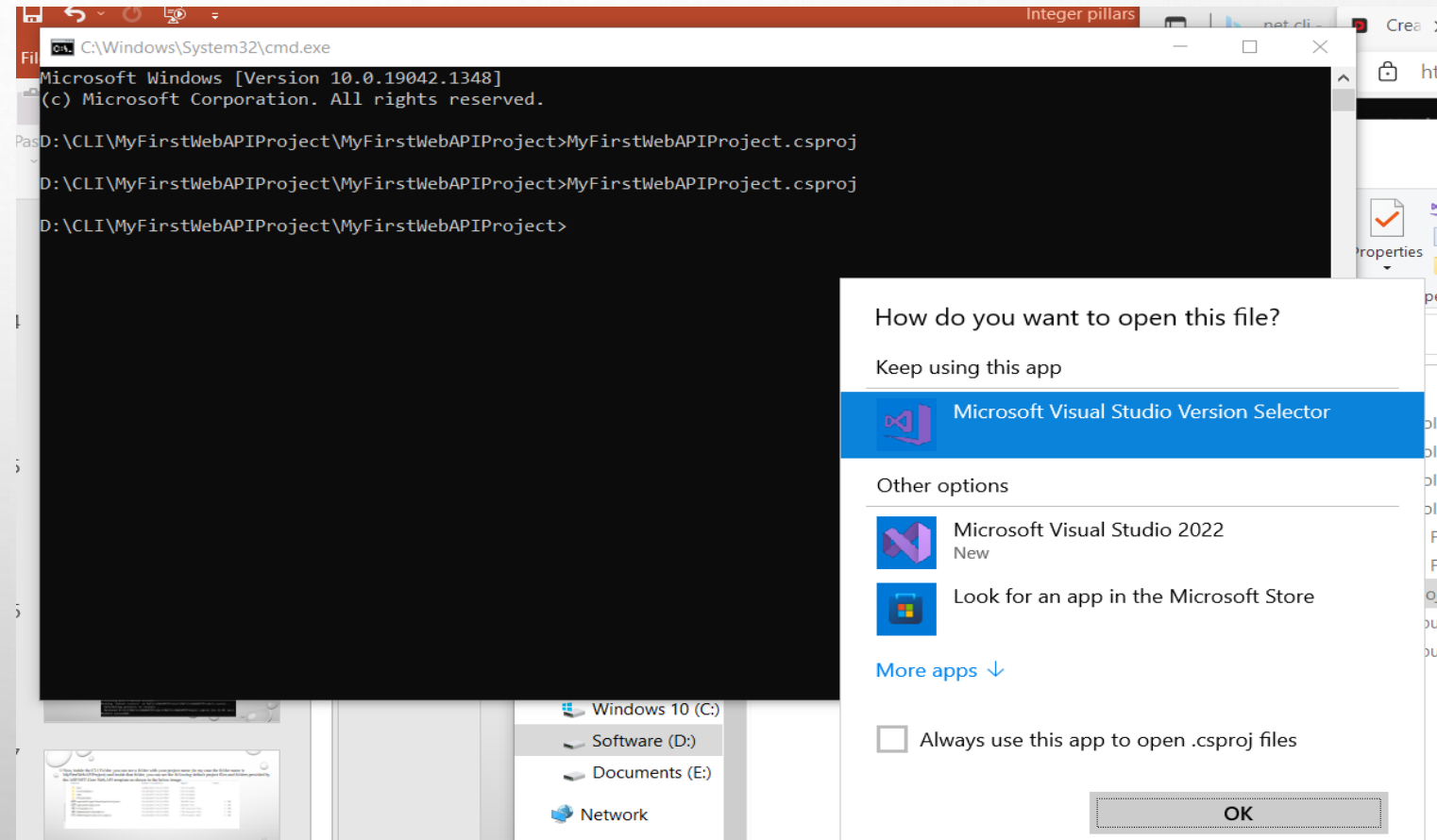
Now let us see how to open the above project in Visual Studio Code. To do so, first change the directory in the command prompt to the project directory (CD MyFirstWebAPIProject) as shown below. Now Open the Project in Visual Studio 2022 , Press Tab and select MyFirstWebAPIProject.sln or MyFirstWebAPIProject.csproj

*D:\CLI\ MyFirstWebAPIProject>MyFirstWebAPIProject.sln,*D:\CLI\ MyFirstWebAPIProject or MyFirstWebAPIProject.csproj
Press Enter Then open the project In Visual Studio 2022.

```
Administrator: Command Prompt
D:\CLI>CD MyFirstWebAPIProject
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

D:\CLI\MyFirstWebAPIProject>MyFirstWebAPIProject.sln
D:\CLI\MyFirstWebAPIProject>
```



Build ASP.Net Core Web API Project Using Command Prompt:

In the command prompt, first set the directory to your project folder and then type `dotnet build` and press the enter button as shown in the below image.

Administrator: Command Prompt

```
D:\CLI\MyFirstWebAPIProject>dotnet build
```

Once you type **dotnet build** and press the enter button, it will build your project and you will get the below message.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

D:\CLI\MyFirstWebAPIProject\MyFirstWebAPIProject>cd..

D:\CLI\MyFirstWebAPIProject>dotnet build
Microsoft (R) Build Engine version 17.0.0+c9eb9dd64 for .NET
Copyright (C) Microsoft Corporation. All rights reserved.

Determining projects to restore...
All projects are up-to-date for restore.
MyFirstWebAPIProject -> D:\CLI\MyFirstWebAPIProject\MyFirstWebAPIProject\bin\Debug\net6.0\MyFirstWebAPIProject.dll

Build succeeded.
    0 Warning(s)
    0 Error(s)

Time Elapsed 00:00:01.16
```

How to run the ASP.NET Core Web API project using .NET Core CLI?

Now let us see how to run the above ASP.NET Core Web API Project. The .NET Core CLI provides the run command to run the ASP.NET Core Web API Application. So, in the terminal type **dotnet run** and press the enter button as shown in the below image. Or **dotnet watch run**

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

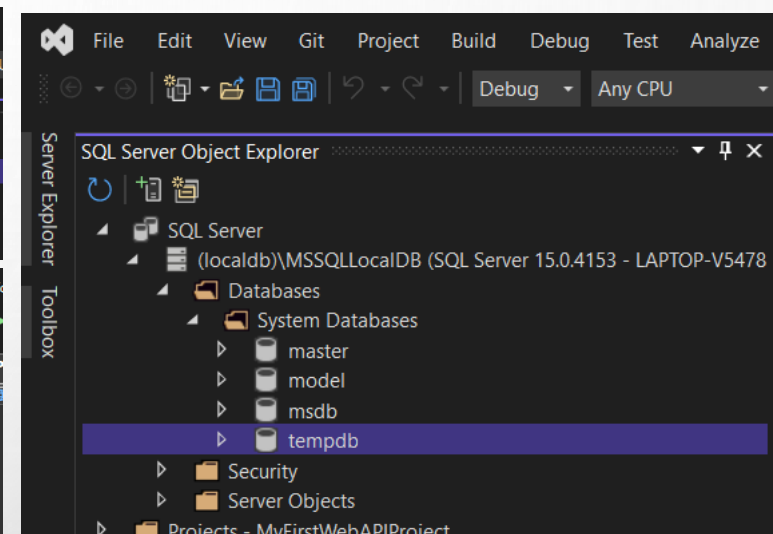
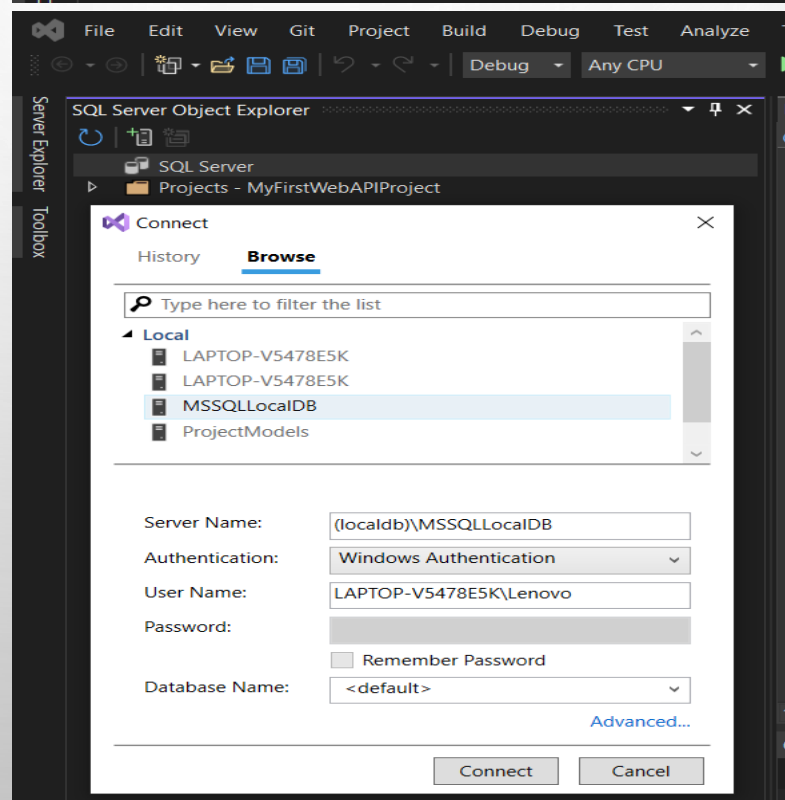
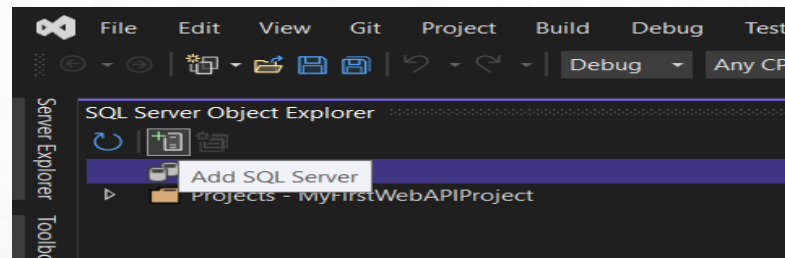
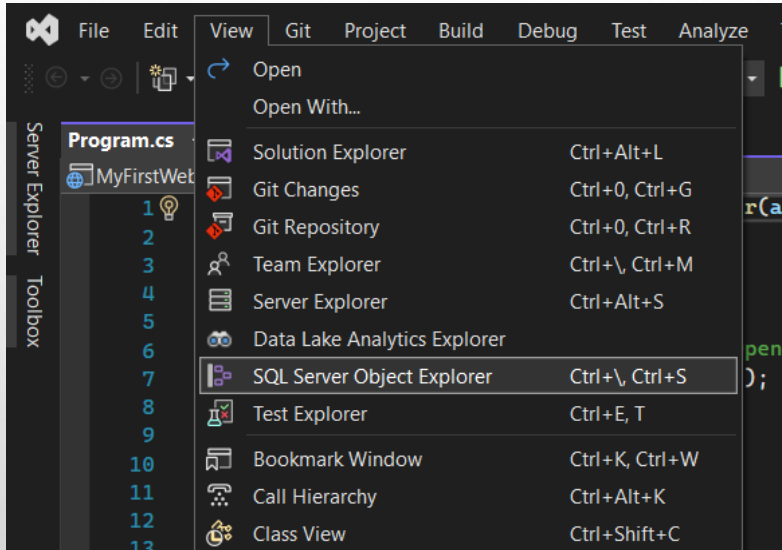
PS D:\CLI\MyFirstWebAPIProject> dotnet run

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS D:\CLI\MyFirstWebAPIProject> dotnet run
Building...
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: D:\CLI\MyFirstWebAPIProject
```

SQL Server Connection With the Application

Visual Studio > View > SQL Server Object Explorer. Add SQL Server , Select Database Name. Local > MSSQLLocalDB , Server Name > (localhost)\MSSQLLocalDB, Authentication > Windows Authentication
Now Click on Connect Button. Then Expend Databases > System Databases > Your Database Name will Show



Database Connection String Copy From Database Properties

Select Your Database and Right Click go to the properties , Now Find out the Connection String then Copy the String

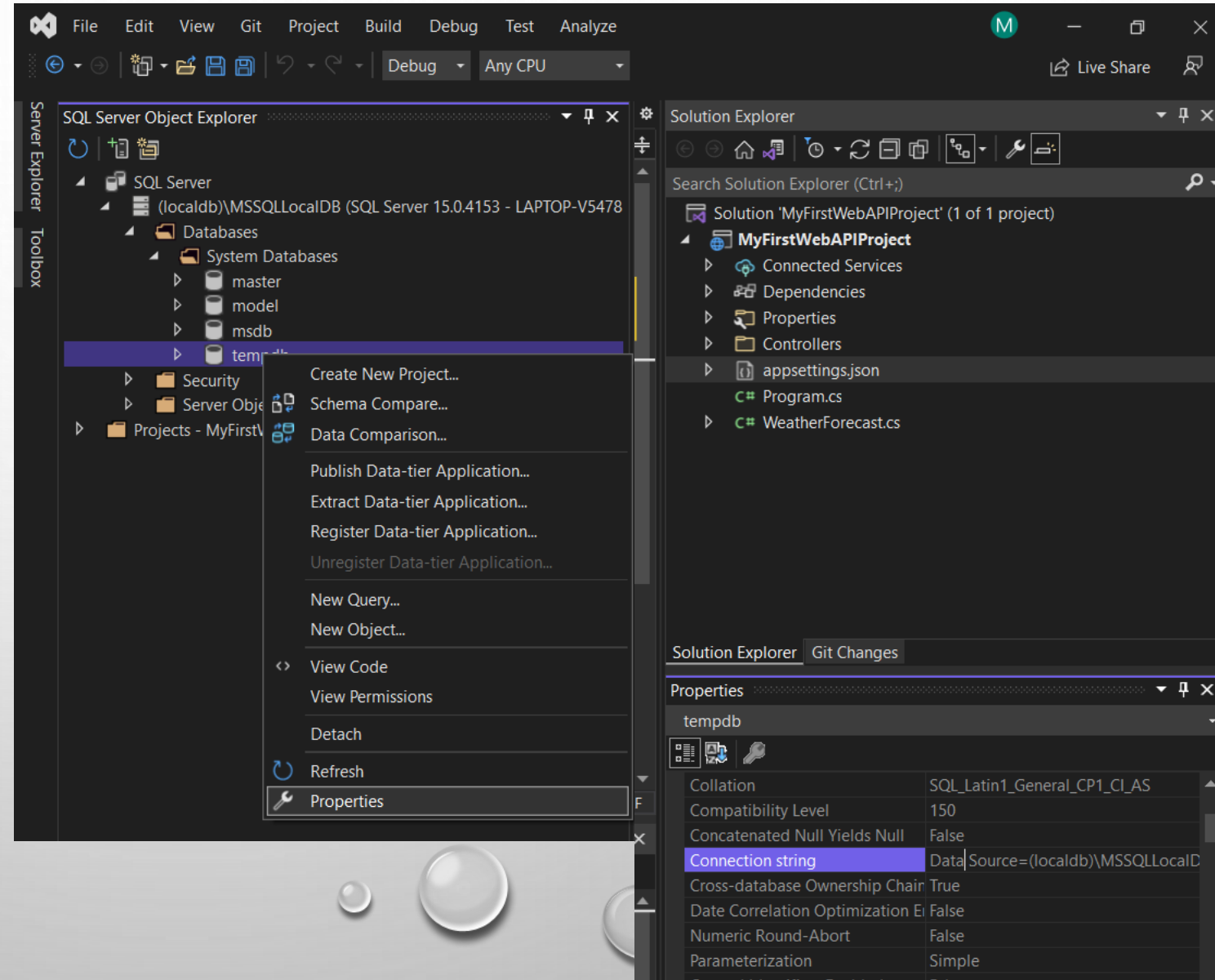
Example:

tempDB >Right click > Click on Properties >
Connection String copy the value of it

“Data Source=(localdb)\MSSQLLocalDB;Initial
Catalog=tempdb;Integrated

Security=True;Connect

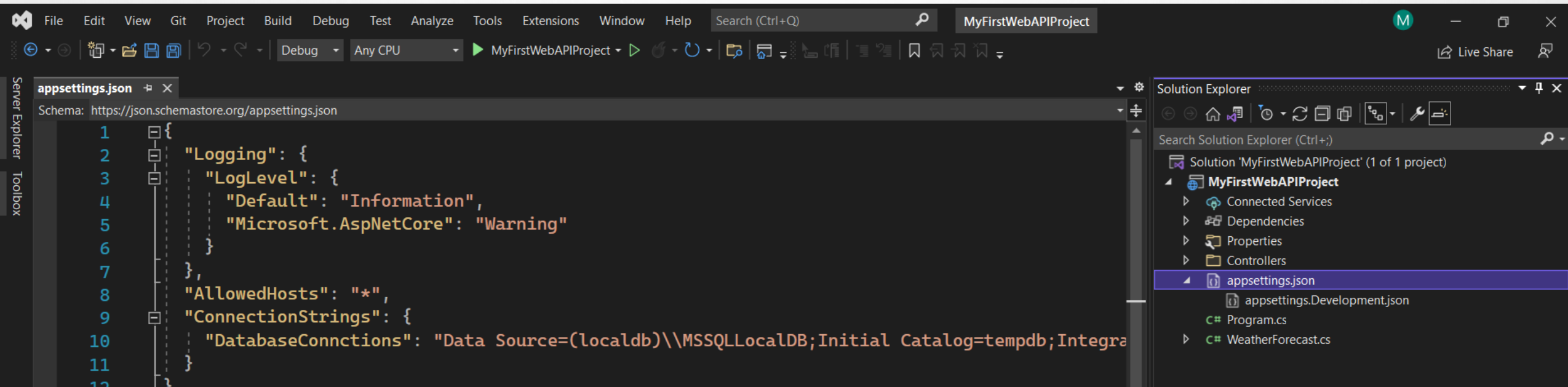
Timeout=30;Encrypt=False;TrustServerCertificate
=False;ApplicationIntent=ReadWrite;MultiSubnet
Failover=False”



Database Connection

Go to the Solutions Explorer > appsettings.json > Now Add the Code

```
"ConnectionStrings": {  
  "DatabaseConnctions": "Data Source=(localdb)\\MSSQLLocalDB;Initial Catalog=tempdb;Integrated Security=True;Connect  
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False"  
}
```



EF Core Power Tools Installing

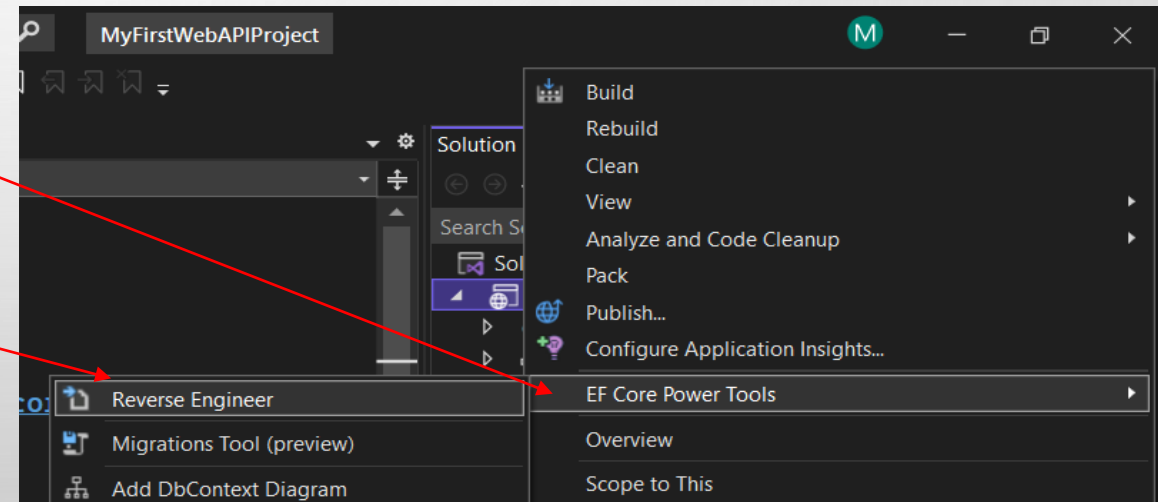
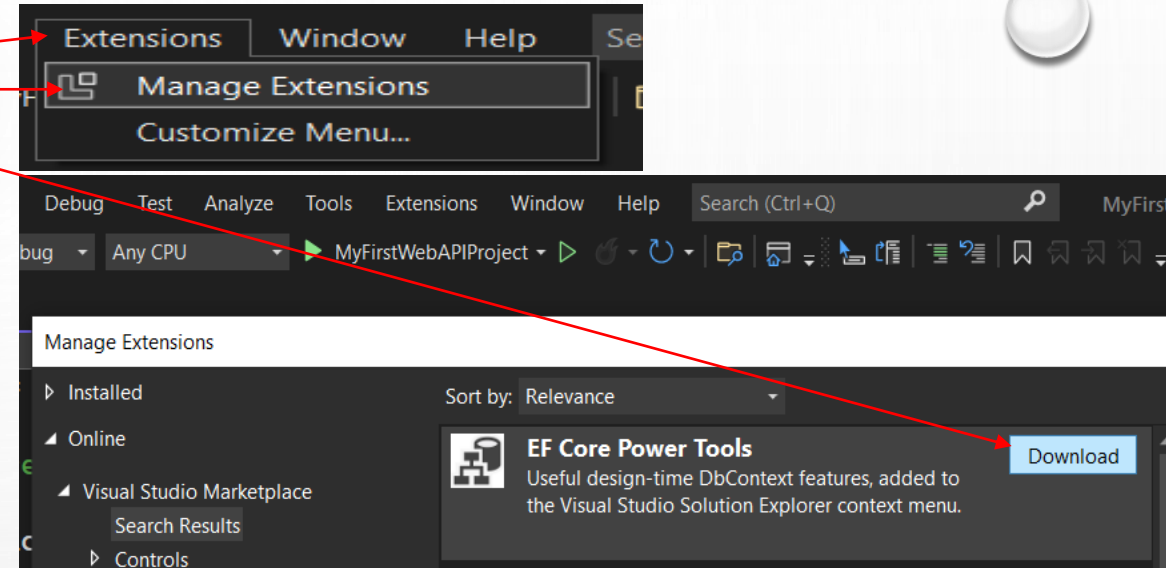
In Visual Studio > Click on Extensions > Manage Extensions > Search EF Core Power Tools > Download

Now Restart Visual Studio

Start Install EF Core Power Tools > By VSIX Installer > Click On Modify > Modification Complete > Click on Close Button Now. Open Your Visual Studio again.

Go to Solutions Explorer > Click on your Project > Right Click > EF Core Power Tools > Reverse Engineer

Now You are ready to Reverse Engineering for creating **Models** and **dependency Injection**



Click on the Link: <https://youtu.be/Vijt9z6jLWI>

Entity Framework

Core 6.0

