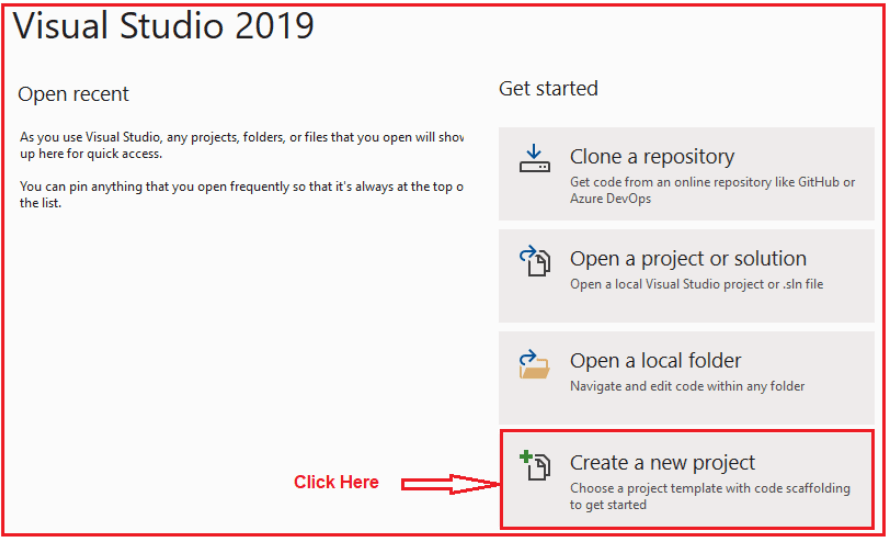
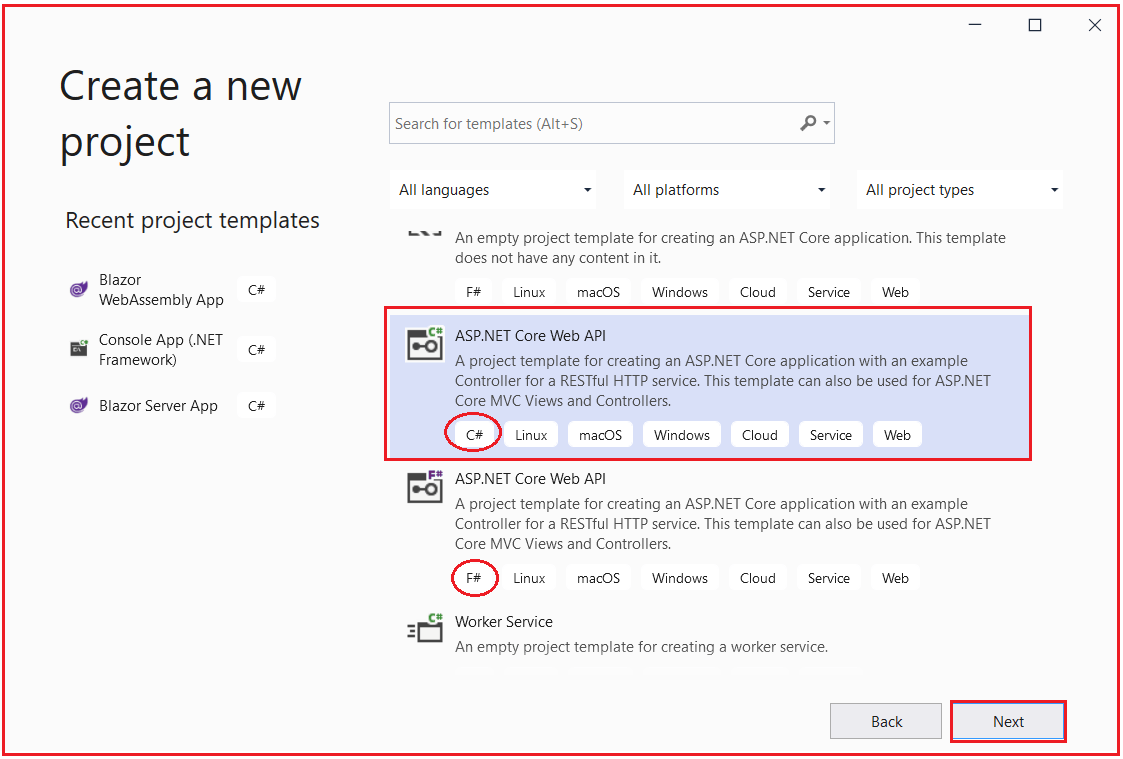
**Creating ASP.NET Core Web API Project Using Visual Studio 2019**

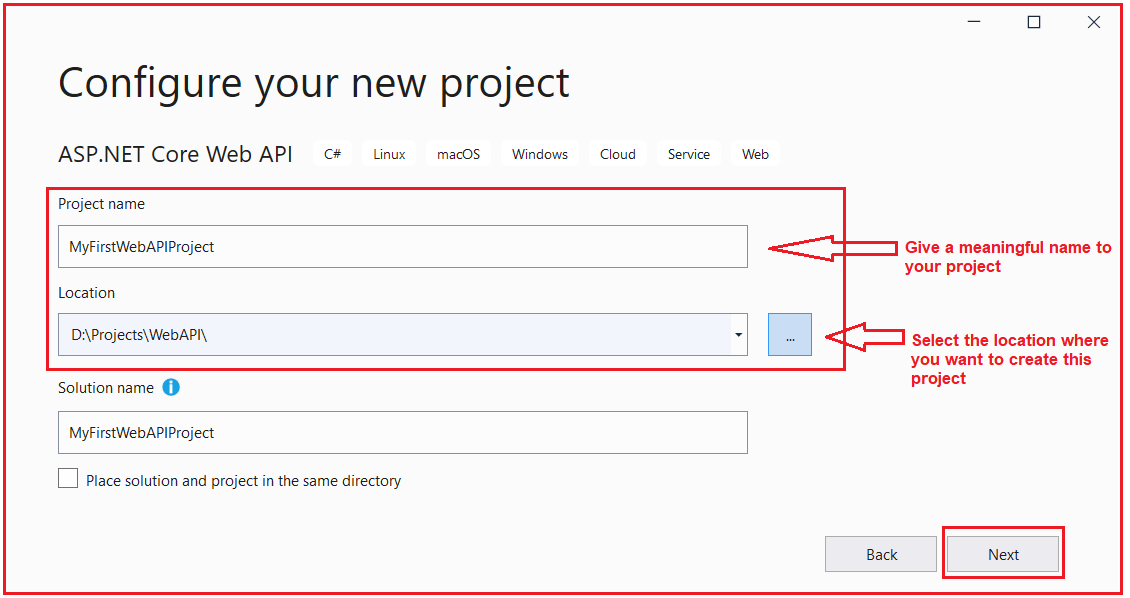
Now, we are going to create the ASP.NET Core Web API project using Visual Studio 2019. So, open Visual Studio 2019 and then click on the Create a new project option as shown in the below image.



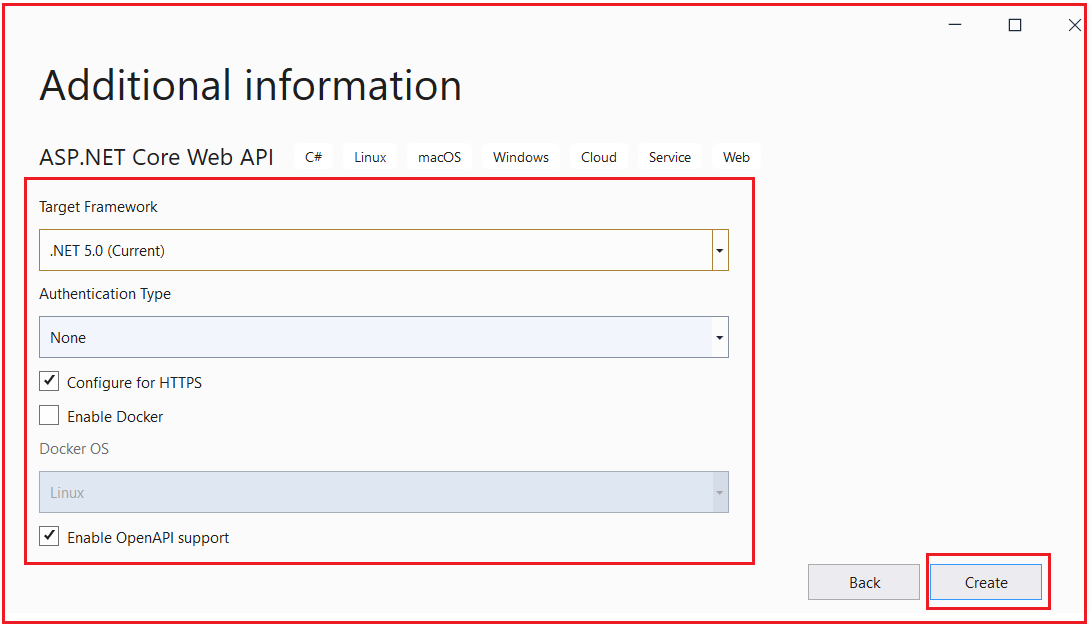
Once you click on the Create a new project option, the following Create a new project window will open. Here, you can find two projects template for creating the **ASP.NET Core Web API**project. One is using C# language and the other one is using F# language. I am going to use C# as the programming language, so I select the project template which uses C# Language as shown in the below image.



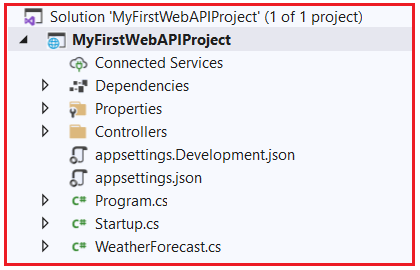
Once you click on the **Next** button, then the configure your new project window will open. Here, you need to specify the Project name (MyFirstWebAPIProject) and the location where you want to create the project. And finally, click on the **Next**button as shown in the below image.



Once you click on the Next button, it will open the Additional Information window. Here, you need to select the Target .NET Framework version. The authentication Types. Whether you want to configure HTTPS and enable Docker. Select .NET 5, select authentication type as None, check the Configure for HTTPS and uncheck the Enable Docker checkboxes and then click on the Create button as shown in the below image.

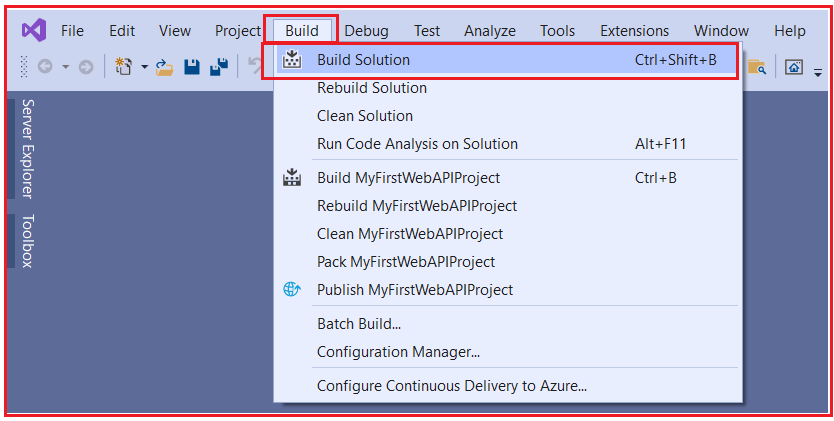


Once you click on the Create button, it will create the ASP.NET Core Web API project with the following file and folder structure. In our next article, we will discuss all these files and folders in detail.

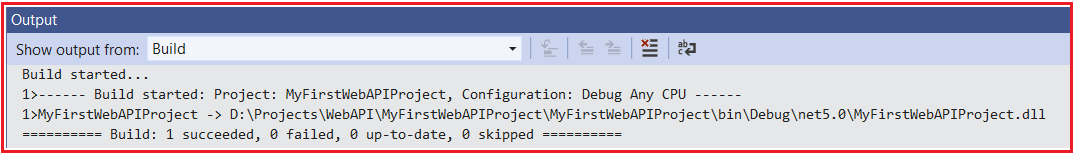


**How to Build the ASP.NET Core Web API project in Visual Studio 2019?**

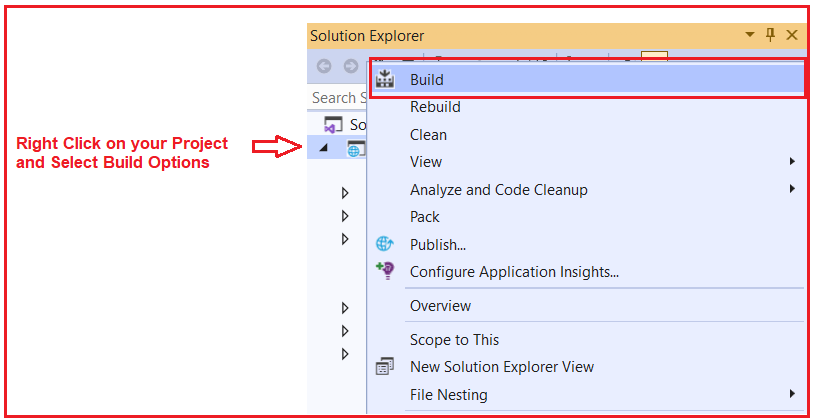
We can build the ASP.NET Core Web API Project in many ways. So, let us discuss them. From the menus, select Build => Build Solution as shown in the below image.



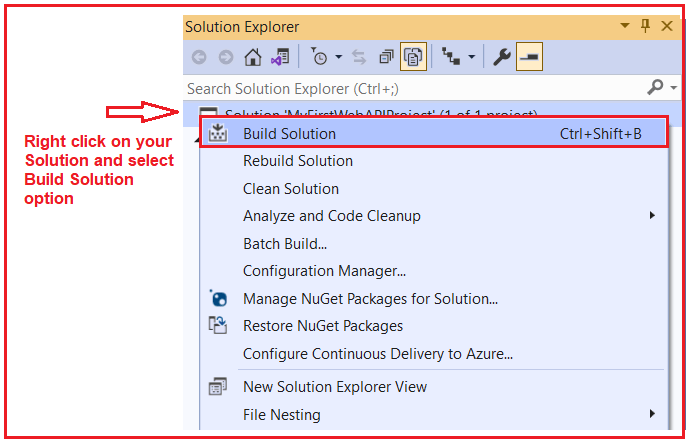
Once you select Build => Build Solution, then it will build all the projects that are there in the solution. Once the build is successful, you will get the below message in the output.



You can also build your project by right-clicking on your project and then selecting the build option from the context menus as shown in the below images.



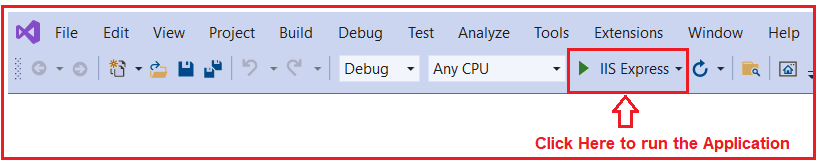
You can also right-click on your solution and select the Build Solution option to build all your projects as shown in the below image.



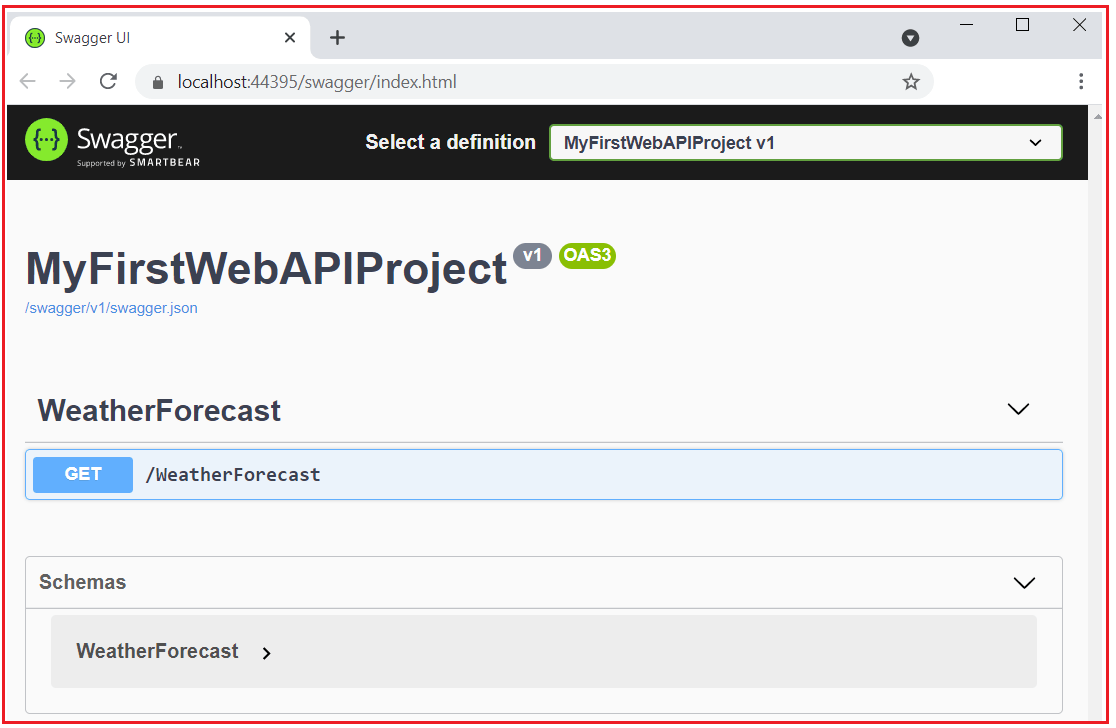
**Note:** If you use the Build Solution option, then it will build all your projects that are present inside the solution. If you want to build a particular project, then simply right-click on the Project and select the Build option which will build that project only. The last option to build projects in Visual Studio is a Keyboard shortcut. You can use **Ctrl+Shift+B** to build your solution.

**How to run the ASP.NET Core Web API Application in Visual Studio 2019?**

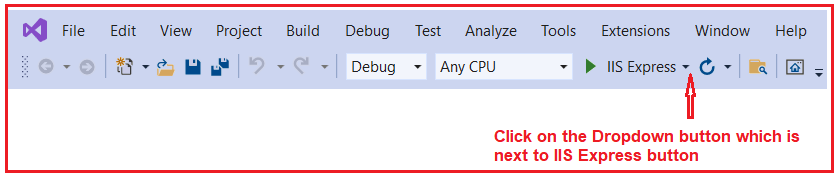
To run the application, simply click on the IIS Express green button as shown in the below image.



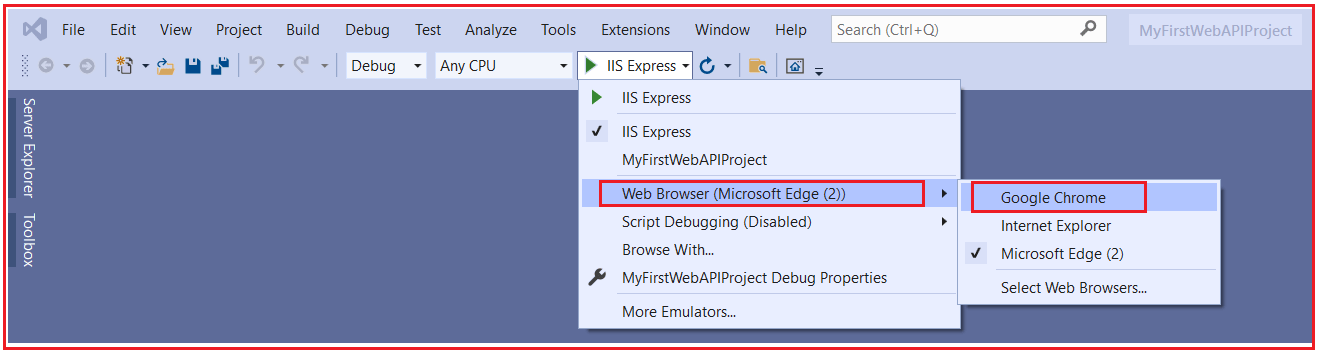
Once you click on the IIS Express button, it will run the application and you will get the following swagger page in the browser. Please have a look at the Port number (44395) on which the application is running.



Here, the above swagger opens in the default Microsoft Edge browser. If you want then you can also change the default browser to something else like Google Chrome. To do so, simply click on the dropdown button which is appeared next to the IIS Express button as shown in the below image.



Then select Google Chrome as the Web Browser as shown in the below image.



With the above changes in place, now run the application and you must the web browser as Google Chrome.

Now, have look at the port number on which the application is running. My application is running on port number 44395. Now you test the API using the same approach that we have discussed in our previous two articles. So, I am not going to discuss the same here. You just try yourself and if you face any problem, then let us know by putting a comment in the comment box. To simplify, I am just hitting the URL (https://localhost:44395/WeatherForecast) from my web browser and I am getting the result as expected as shown in the below image.

