Web Api in VS code , using .net Core

1) Install .Net core from MS site.

2) Once it is installed , can check the ver in cmd dotnet --version

3) If we are able to see the version , we need to create the folder for the application

and create the solution

dotnet new sln :- this cmd will create new sln

4)Next step was to create the web api project

dotnet new webapi -o API :- here -o is the output and "API" is name of the project

5)Next step is to add Solution to the API

dotnet sln add API

---------------------------

RUN .NET FROM CMD PROMPT

to run the .Net project we need to run the cmd as

dotnet run (Make sure we are in the project folder before this)

We need to add cert to the make the dev trust

dotnet dev-certs https --trust

-----------

SETTING THE EF IN VS CODE

in the extension of vs code intall the nuget gallery

Once installed , we can go to the nuget gallery and installe the respective entity frame work.

in this case it was sql lite so i have installed the below package .

Microsoft.EntityFrameworkCore.Sqlite :- (Note :- to add to select API.csproj, this will add the config details in the API.csproj file)

Once we did this we need to set the connection string in startup.cs file

Note :- here we can also set the dependencies injection in the ConfigureServices method.

Once we have table and connection in place , in this case we need to do the migration in or sqllite or sql .

to check the latest package go to nuget.org search for dotnet-ef , select the right version and copy the path from there.

Note in order to do the migration we need to stop running the app.

CMD for migration is dotnet ef migrations add InitialCreate -o Data/Migrations

where InitialCreate is the name.

Create DB

once we have run the above cmd we need to create the DB

run the cmd in vs terminal **dotnet ef database update**

As we are using sqlite in our case , we can go extension and add sqlite package to see the data.

once you have installed the package you can open the pallet ,

and select sqlite : open database cmd . and select dattingapp.db(Name given for creating connection string).

-----

GIT details

Refer the details on the lecture 18

---------------------

Angular details : Note installed ver of ng is ng 14.

* First step is to install the node on the pc cmd to check the node version is

**node - -version**

* Then install angular using NPM cmd

**npm install g @angular/cli  *(Note :- if you wanat tot install specific ver of ng use @12 , where 12 is ver of ng , this needs ot given after /cli ex :- npm install g @angular/cli @12)***

* check @angular version by **ng version**
* Next step is to create a new angular workspace. Cmd

Ng new [name] – strict false

**Adding Style sheets in angular**

1. **Bootstrap** is used for fast and responsive Ui design which can also be customized.

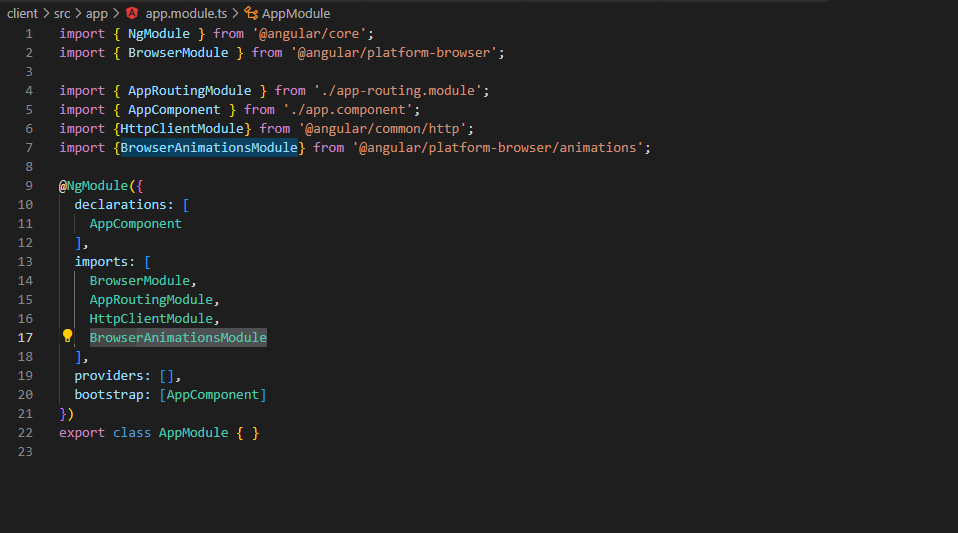
To install the Bootstrap use the below cmd.

**ng add ngx-bootstrap**

if you want install any specific bootstrap package append version at eth end like **ng add ngx-bootstrap@7.1.0**

this will update your below files

1. Package.json
2. Angular.json
3. Module.ts



In my case the module.ts was not updated which I updated it manually as shown above.

Authentication :-

In order to secure the application we are going to add to fields in user table

1. Byte[] PasswordHash
2. Byte[] PasswordSalt

Then we need to stop our dotnet app if it is running , and do the migrations in the db.

**dotnet ef migrations add UserPasswordAdded**

after running this command, you can see that migrations file is updated .

Now the next step is to update this details in DB.

**dotnet ef database update:** This will update the DB.

**Truncate DB using ef Commands.**

Stop the dotnet app.Run the below cmd

**dotnet ef database drop** *(after this you will get warning , are you sure you want to do this , can select yes)*

**dotnet ef database update**

**JWT Token.**

In order to implement the JWt token , we need to install the

**System.Identity.Model.Token.Jwt** *by Microsoft* **, *this is available in the nuget pacakage manager.***

**Check the box on the Api.csproj (name of the Api project )**

**To add the middle layer for the JWt ( means , the token generated by login and reigister , we need to preserve the token on the server and not on the DB.)**

**We need to install the package by nuget below.**

***Microsoft.Aspnetcore.authentication.jwtBearer.***

Once this is installed , for the method with header as [Authorize] , we need to add few details in starup.cs file

In the configurationService method . below details needs to be mentioned.

            services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)

            .AddJwtBearer(options=>

            {

                options.TokenValidationParameters = new TokenValidationParameters

                {

                    ValidateIssuerSigningKey = true,

                    IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(Configuration["TokenKey"])),

                    ValidateIssuer = false,

                    ValidateAudience = false,

                };

            });

Also in the configure method we need to add the Authorization , ordering in important here.

           // app.UseCors(x=> x.AllowAnyHeader().AllowAnyMethod().AllowAnyOrigin());

            app.UseCors(MyAllowSpecificOrigins);

**app.UseAuthentication();**

            app.UseAuthorization();