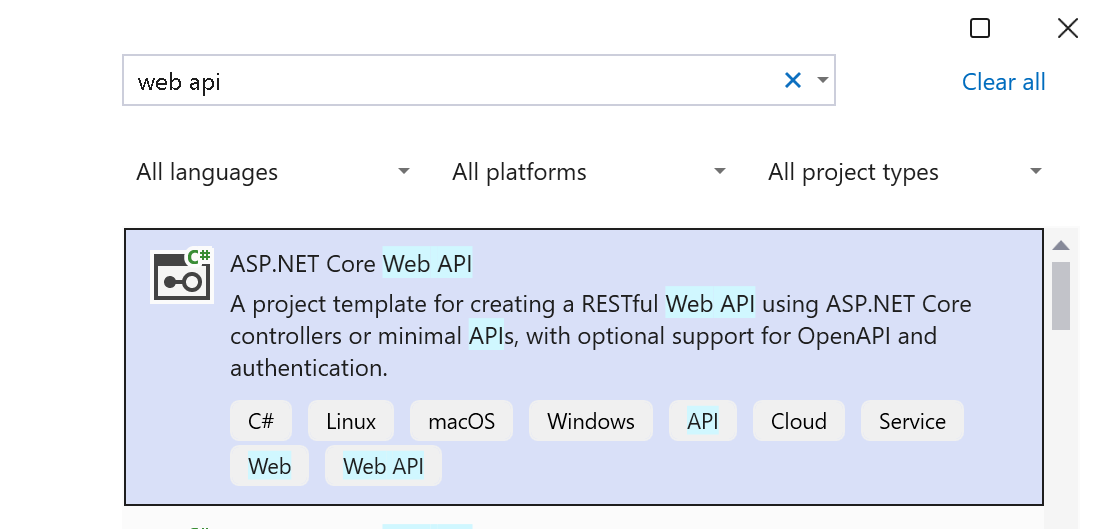
1. **Api cho webserver:**

**Bước 1:** setup bảng database // bước này sẽ tuỳ theo từng dạng bài khác nhau, như bài dũng thì gồm (id,soA,soB,tong,thoitian);

**Bước 2**: Tạo project là web api, rồi cứ bấm oke hết là được



**Bước 3:** setup api trong file : appsettings.json (tại đây setup đường dẫn cho code);

Code có sẵn khi mới tạo project xong:

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"AllowedHosts": "\*"

}

Thêm đoạn này trên logging :

"ConnectionStrings": {

"DefaultConnection": "Server=localhost;Database=demo\_websever;Trusted\_Connection=True;TrustServerCertificate=True;"

} \\ server = tuỳ demo chạy ở đâu, Database = tên của file db đã setup, không được sai tên của

\\ db, sai thì sẽ dẫn qua file db khác.

\\Trust\_connection, trustservercertificate = true ( cho phép vsc được kết nối tới db);

**Bước 4:** setup file [program.cs](http://program.cs) ( để vsc biết lấy đường dẫn ở đâu / ứng dụng db nào)

* Ở gần đầu trong file program khi mới tạo có **// Add services to the container.**
* Thêm dòng này :   
  builder.Services.AddDbContext<AppDbContext>(options=>options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")));

// builder.Services.AddDbContext (đăng ký dịch vụ kết nối Db)

//options.UseSqlServer (chọn ứng dụng db sql express hay gì đó)

//builder.Configuration.GetConnectionString("DefaultConnection") (đường dẫn là biến //DefaultConnection

**Bước 5 :** tạo folder Data, tạo [AppDbContext.cs](http://appdbcontext.cs) trong Data ( **cầu nối các thao tác với Db**)

using Microsoft.EntityFrameworkCore;

// Import namespace chứa lớp model TinhTong

using demo\_sum.Models;

// Khai báo namespace để gom nhóm các class liên quan đến dữ liệu

namespace demo\_sum.Data

{

public class AppDbContext : DbContext // kế thừa DbContext(hàm làm việc với Db)

{

// Dùng để cấu hình chuỗi kết nối, kiểu database (SQL Server, SQLite,...)

public AppDbContext(DbContextOptions<AppDbContext> options)

: base(options)

{

}

public DbSet<TinhTong> TinhTong\_DEMO { get; set; } // thực hiện các thao tác với bảng db(truy vấn, thêm , xoá );

}

}

**Bước 6 :** tạo folder Model, tạo file [TinhTong.cs](http://tinhtong.cs) ( **lưu các biến có trong bảng Db**)

using System.ComponentModel.DataAnnotations.Schema;

namespace demo\_sum.Models

{

[Table("TinhTong\_DEMO")]

public class TinhTong

{

public int Id { get; set; }

public double SoA { get; set; }

public double SoB { get; set; }

public double Tong { get; set; }

public DateTime ThoiGian { get; set; }

}

} // file này code đơn giản, chỉ là thao tác với các biến có trong bảng

**Bước 7** trong folder controller , tạo file [TinhTongController.cs](http://tinhtongcontroller.cs) ( **xử lý thao tác trong file này, như dũng là xử lý tính tổng**):

using demo\_sum.Data;

using demo\_sum.Models;

using Microsoft.AspNetCore.Mvc;

namespace demo\_sum.Controllers

{

[ApiController] //**Cho phép controller tự động kiểm tra và nhận JSON**

[Route("api/[controller]")] **// định nghĩa để URL ngắn hơn**

public class TinhTongController : ControllerBase

{

private readonly AppDbContext \_context;

public TinhTongController(AppDbContext context)  **// thao tác kêt nối tới AppDbContext**

{

\_context = context;

}

[HttpPost] **// khi client gửi request tới webserver**

public IActionResult TinhTong([FromBody] TinhTong input)

{

if (input == null)

return BadRequest("Dữ liệu không hợp lệ");

input.Tong = input.SoA + input.SoB;

input.ThoiGian = DateTime.Now;

\_context.TinhTong\_DEMO.Add(input);

\_context.SaveChanges();

return Ok(new

{

message = "Tính tổng thành công!",

a = input.SoA,

b = input.SoB,

tong = input.Tong,

thoigian = input.ThoiGian

});

}

[HttpGet]  **// gửi lại client là thành công và lưu biến vô Db**

public IActionResult GetAll()

{

var list = \_context.TinhTong\_DEMO.ToList();

return Ok(list);

}

}

}

Program.cs → cấu hình server & dịch vụ

↓

AppDbContext → đăng ký kết nối DB

↓

TinhTongController → nhận request HTTP

↓

TinhTong → map dữ liệu sang C# object

↓

AppDbContext → ghi hoặc đọc dữ liệu từ SQL

↓

Controller → trả response JSON ra cho client