# Minimal WebApi:

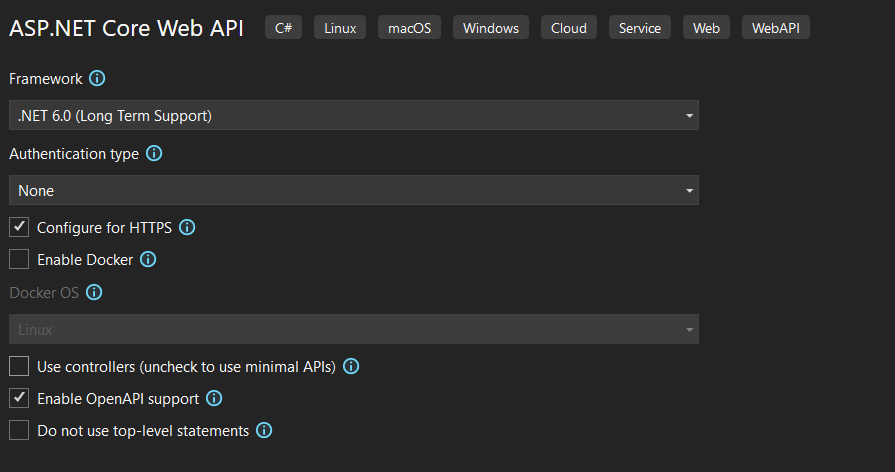
Shorter way to create web Api.

**Advantages:**

* It’s way less verbose than using traditional controllers.
* It’s easy to write and easy to read. There’s also less cognitive load associated with reading it.
* It’s faster than using traditional controllers.

**Disadvantages:**

* organizing REST API into separate controllers would make the code more readable and maintainable in an enterprise-grade application with a large number of endpoints.



app.MapGet("/weatherforecast", (NameService nameService) =>

{

app.Logger.LogInformation("/weatherforecast called");

var forecast = Enumerable.Range(1, 5).Select(index =>

new WeatherForecast

(

DateOnly.FromDateTime(DateTime.Now.AddDays(index)),

Random.Shared.Next(-20, 55),

summaries[Random.Shared.Next(summaries.Length)]

))

.ToArray();

return forecast;

})

.WithName("GetWeatherForecast")

.WithOpenApi();

//Statuscodes

app.MapGet("/Statuscodes", (bool ok) => ok ? Results.Ok("Everything is ok!") : Results.BadRequest("Bad Request!"));

//Routing

app.MapGet("/", get);

app.MapPost("/", () => "Post called");

app.MapPut("/", () => "Put called");

app.MapDelete("/", () => "Delete called");

var personHandler = new PersonHandler();

app.MapGet("/Persons", personHandler.HandleGet);

Route Parameters

app.MapGet("/Persons/{id}", personHandler.HandleGetById);

//Route parameter constraints

app.MapGet("/Persons/{id:int}", personHandler.HandleGetById);

//Parameter binding

//Person json from body to Person object automatically

app.MapPost("Persons", (Person person) => person.FirstName + ", " + person.LastName);

app.Run();

string get() => "Get called";

internal record WeatherForecast(DateOnly Date, int TemperatureC, string? Summary)

{

public int TemperatureF => 32 + (int)(TemperatureC / 0.5556);

}

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddScoped<NameService>();

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

var app = builder.Build();

// Configure the HTTP request pipeline.

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI();

}

app.UseHttpLogging();

app.UseHttpsRedirection();

var summaries = new[]

{

"Freezing", "Bracing", "Chilly", "Cool", "Mild", "Warm", "Balmy", "Hot", "Sweltering", "Scorching"

};

**Exercise**

Write A CRUD App that create all 4 functions with minimal APi, and updates of a list created in the program.cs.