ASP.NET Core 8.0 Web API

@fifth\_HostAddress = http://localhost:5293

GET {{fifth\_HostAddress}}/weatherforecast/

Accept: application/json

###

{

  "Logging": {

    "LogLevel": {

      "Default": "Information",

      "Microsoft.AspNetCore": "Warning"

    }

  },

  "AllowedHosts": "\*"

}

Microsoft Visual Studio Solution File, Format Version 12.00

# Visual Studio Version 17

VisualStudioVersion = 17.5.2.0

MinimumVisualStudioVersion = 10.0.40219.1

Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "fifth", "fifth.csproj", "{33F2C396-2702-B35A-6755-1714297C7874}"

EndProject

Global

GlobalSection(SolutionConfigurationPlatforms) = preSolution

Debug|Any CPU = Debug|Any CPU

Release|Any CPU = Release|Any CPU

EndGlobalSection

GlobalSection(ProjectConfigurationPlatforms) = postSolution

{33F2C396-2702-B35A-6755-1714297C7874}.Debug|Any CPU.ActiveCfg = Debug|Any CPU

{33F2C396-2702-B35A-6755-1714297C7874}.Debug|Any CPU.Build.0 = Debug|Any CPU

{33F2C396-2702-B35A-6755-1714297C7874}.Release|Any CPU.ActiveCfg = Release|Any CPU

{33F2C396-2702-B35A-6755-1714297C7874}.Release|Any CPU.Build.0 = Release|Any CPU

EndGlobalSection

GlobalSection(SolutionProperties) = preSolution

HideSolutionNode = FALSE

EndGlobalSection

GlobalSection(ExtensibilityGlobals) = postSolution

SolutionGuid = {54FB8659-D61E-494E-B256-F25417B414C8}

EndGlobalSection

EndGlobal

<Project Sdk="Microsoft.NET.Sdk.Web">

<PropertyGroup>

<TargetFramework>net9.0</TargetFramework>

<Nullable>enable</Nullable>

<ImplicitUsings>enable</ImplicitUsings>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="Microsoft.AspNetCore.Authentication.JwtBearer" Version="9.0.7" />

<PackageReference Include="Microsoft.AspNetCore.Cors" Version="2.3.0" />

<PackageReference Include="Microsoft.AspNetCore.OpenApi" Version="9.0.6" />

<PackageReference Include="Swashbuckle.AspNetCore" Version="9.0.3" />

<PackageReference Include="System.IdentityModel.Tokens.Jwt" Version="8.12.1" />

</ItemGroup>

</Project>

{

  "Logging": {

    "LogLevel": {

      "Default": "Information",

      "Microsoft.AspNetCore": "Warning"

    }

  }

}

using Microsoft.AspNetCore.Authentication.JwtBearer;

using Microsoft.IdentityModel.Tokens;

using System.Text;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();

// CORS Configuration

builder.Services.AddCors(options =>

{

options.AddPolicy("AllowSpecificOrigin", policy =>

{

policy.WithOrigins("http://localhost:3000", "http://localhost:4200", "http://localhost:5173")

.AllowAnyHeader()

.AllowAnyMethod()

.AllowCredentials();

});

});

// JWT Authentication Configuration

string securityKey = "mysuperdupersecret\_make\_it\_long\_for\_security\_jwt\_token";

var symmetricSecurityKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(securityKey));

builder.Services.AddAuthentication(x =>

{

x.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;

x.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;

x.DefaultSignInScheme = JwtBearerDefaults.AuthenticationScheme;

})

.AddJwtBearer(JwtBearerDefaults.AuthenticationScheme, x =>

{

x.TokenValidationParameters = new TokenValidationParameters

{

// What to validate

ValidateIssuer = true,

ValidateAudience = true,

ValidateLifetime = true,

ValidateIssuerSigningKey = true,

// Setup validate data

ValidIssuer = "mySystem",

ValidAudience = "myUsers",

IssuerSigningKey = symmetricSecurityKey

};

});

// Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

var app = builder.Build();

// Configure the HTTP request pipeline.

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI();

}

// CORS Middleware (before routing)

app.UseCors("AllowSpecificOrigin");

app.UseHttpsRedirection();

// Authentication & Authorization Middleware

app.UseAuthentication();

app.UseAuthorization();

app.MapControllers();

app.Run();





