ASP.NET Core 8.0 Web API

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}") = "EmployeeWebApi", "EmployeeWebApi\EmployeeWebApi.csproj", "{C19B173F-AB32-4B0A-4BB6-477A575CF97F}"

EndProject

Global

GlobalSection(SolutionConfigurationPlatforms) = preSolution

Debug|Any CPU = Debug|Any CPU

Release|Any CPU = Release|Any CPU

EndGlobalSection

GlobalSection(ProjectConfigurationPlatforms) = postSolution

{C19B173F-AB32-4B0A-4BB6-477A575CF97F}.Debug|Any CPU.ActiveCfg = Debug|Any CPU

{C19B173F-AB32-4B0A-4BB6-477A575CF97F}.Debug|Any CPU.Build.0 = Debug|Any CPU

{C19B173F-AB32-4B0A-4BB6-477A575CF97F}.Release|Any CPU.ActiveCfg = Release|Any CPU

{C19B173F-AB32-4B0A-4BB6-477A575CF97F}.Release|Any CPU.Build.0 = Release|Any CPU

EndGlobalSection

GlobalSection(SolutionProperties) = preSolution

HideSolutionNode = FALSE

EndGlobalSection

GlobalSection(ExtensibilityGlobals) = postSolution

SolutionGuid = {612160E3-5E79-4CDB-810C-28BB4C494913}

EndGlobalSection

EndGlobal

{

  "Logging": {

    "LogLevel": {

      "Default": "Information",

      "Microsoft.AspNetCore": "Warning"

    }

  }

}

@EmployeeWebApi\_HostAddress = http://localhost:5206

GET {{EmployeeWebApi\_HostAddress}}/weatherforecast/

Accept: application/json

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

<PropertyGroup>

<TargetFramework>net9.0</TargetFramework>

<Nullable>enable</Nullable>

<ImplicitUsings>enable</ImplicitUsings>

</PropertyGroup>

<ItemGroup>

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

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

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

</ItemGroup>

</Project>

using EmployeeWebApi.Filters;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container

builder.Services.AddControllers(options =>

{

// Add the custom exception filter globally to all controllers

options.Filters.Add<CustomExceptionFilter>();

});

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

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen(c =>

{

c.SwaggerDoc("v1", new Microsoft.OpenApi.Models.OpenApiInfo

{

Title = "Employee Web API",

Version = "v1",

Description = "A sample API for Employee Management with Custom Filters"

});

});

var app = builder.Build();

// Configure the HTTP request pipeline.

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI(c =>

{

c.SwaggerEndpoint("/swagger/v1/swagger.json", "Employee API V1");

// Set Swagger UI at apps root

c.RoutePrefix = string.Empty;

});

}

// Remove HTTPS redirection for now to fix the issue

// app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

// Add a simple redirect for debugging

app.MapGet("/", () => Results.Redirect("/swagger"));

app.Run();





