**WEEK4-6362596(Superset id)**

**MyFirstAPI**

**1.**

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

<PropertyGroup>

<TargetFramework>net9.0</TargetFramework>

<Nullable>enable</Nullable>

<ImplicitUsings>enable</ImplicitUsings>

</PropertyGroup>

<ItemGroup>

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

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

</ItemGroup>

</Project>

VisualStudioVersion = 17.5.2.0

MinimumVisualStudioVersion = 10.0.40219.1

Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "MyFirstAPI", "MyFirstAPI.csproj", "{45F75552-7C1B-A03B-D43B-1129F270063E}"

EndProject

Global

GlobalSection(SolutionConfigurationPlatforms) = preSolution

Debug|Any CPU = Debug|Any CPU

Release|Any CPU = Release|Any CPU

EndGlobalSection

GlobalSection(ProjectConfigurationPlatforms) = postSolution

{45F75552-7C1B-A03B-D43B-1129F270063E}.Debug|Any CPU.ActiveCfg = Debug|Any CPU

{45F75552-7C1B-A03B-D43B-1129F270063E}.Debug|Any CPU.Build.0 = Debug|Any CPU

{45F75552-7C1B-A03B-D43B-1129F270063E}.Release|Any CPU.ActiveCfg = Release|Any CPU

{45F75552-7C1B-A03B-D43B-1129F270063E}.Release|Any CPU.Build.0 = Release|Any CPU

EndGlobalSection

GlobalSection(SolutionProperties) = preSolution

HideSolutionNode = FALSE

EndGlobalSection

GlobalSection(ExtensibilityGlobals) = postSolution

SolutionGuid = {95EEDB1B-547F-40EA-BA75-81207E39600D}

EndGlobalSection

EndGlobal

using Microsoft.OpenApi.Models;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

var app = builder.Build();

// Configure the HTTP request pipeline.

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI();

}

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

// Original minimal API endpoint (optional)

var summaries = new[]

{

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

};

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

{

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");

app.Run();

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

{

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

}

{

"Logging": {

"LogLevel": {

"Default": "Information",

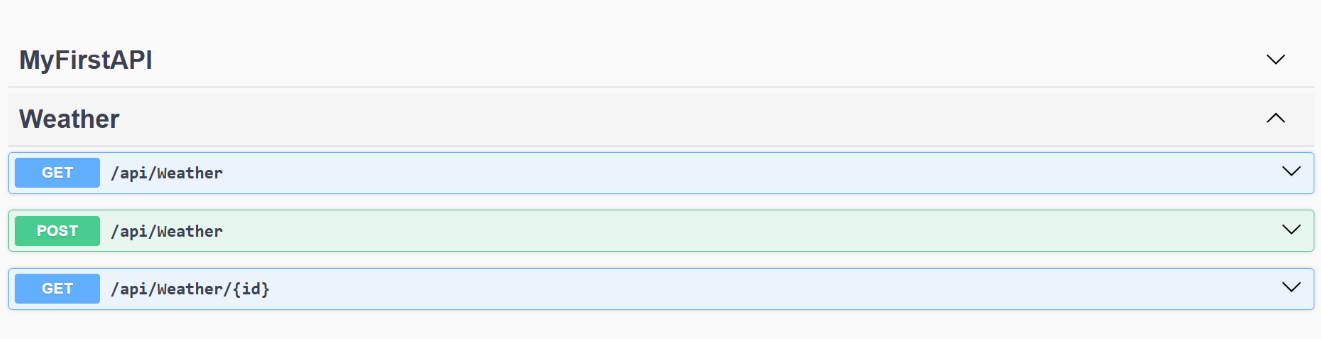
"Microsoft.AspNetCore": "Warning"

}

}

}

**OUTPUT:-**



**2.**

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

<PropertyGroup>

<TargetFramework>net9.0</TargetFramework>

<Nullable>enable</Nullable>

<ImplicitUsings>enable</ImplicitUsings>

</PropertyGroup>

<ItemGroup>

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

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

</ItemGroup>

</Project>

@MyFirstAPI\_HostAddress = http://localhost:5180

GET {{MyFirstAPI\_HostAddress}}/weatherforecast/

Accept: application/json

VisualStudioVersion = 17.5.2.0

MinimumVisualStudioVersion = 10.0.40219.1

Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "MyFirstAPI", "MyFirstAPI.csproj", "{45F75552-7C1B-A03B-D43B-1129F270063E}"

EndProject

Global

GlobalSection(SolutionConfigurationPlatforms) = preSolution

Debug|Any CPU = Debug|Any CPU

Release|Any CPU = Release|Any CPU

EndGlobalSection

GlobalSection(ProjectConfigurationPlatforms) = postSolution

{45F75552-7C1B-A03B-D43B-1129F270063E}.Debug|Any CPU.ActiveCfg = Debug|Any CPU

{45F75552-7C1B-A03B-D43B-1129F270063E}.Debug|Any CPU.Build.0 = Debug|Any CPU

{45F75552-7C1B-A03B-D43B-1129F270063E}.Release|Any CPU.ActiveCfg = Release|Any CPU

{45F75552-7C1B-A03B-D43B-1129F270063E}.Release|Any CPU.Build.0 = Release|Any CPU

EndGlobalSection

GlobalSection(SolutionProperties) = preSolution

HideSolutionNode = FALSE

EndGlobalSection

GlobalSection(ExtensibilityGlobals) = postSolution

SolutionGuid = {25D6BA75-8E96-4784-9E2E-0D183AFA142B}

EndGlobalSection

EndGlobal

using Microsoft.OpenApi.Models;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();

builder.Services.AddEndpointsApiExplorer();

// Advanced Swagger Configuration

builder.Services.AddSwaggerGen(c =>

{

c.SwaggerDoc("v1", new OpenApiInfo

{

Title = "MyFirstAPI - Advanced Demo",

Version = "v1.0",

Description = "Learning Web API with Swagger Documentation",

TermsOfService = new Uri("https://example.com/terms"),

Contact = new OpenApiContact

{

Name = "KIIT Student",

Email = "student@kiit.ac.in",

Url = new Uri("https://kiit.ac.in")

},

License = new OpenApiLicense

{

Name = "MIT License",

Url = new Uri("https://opensource.org/licenses/MIT")

}

});

});

var app = builder.Build();

// Configure the HTTP request pipeline.

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI(c =>

{

c.SwaggerEndpoint("/swagger/v1/swagger.json", "MyFirstAPI v1.0");

c.RoutePrefix = "swagger"; // Custom URL prefix

});

}

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

}

}

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

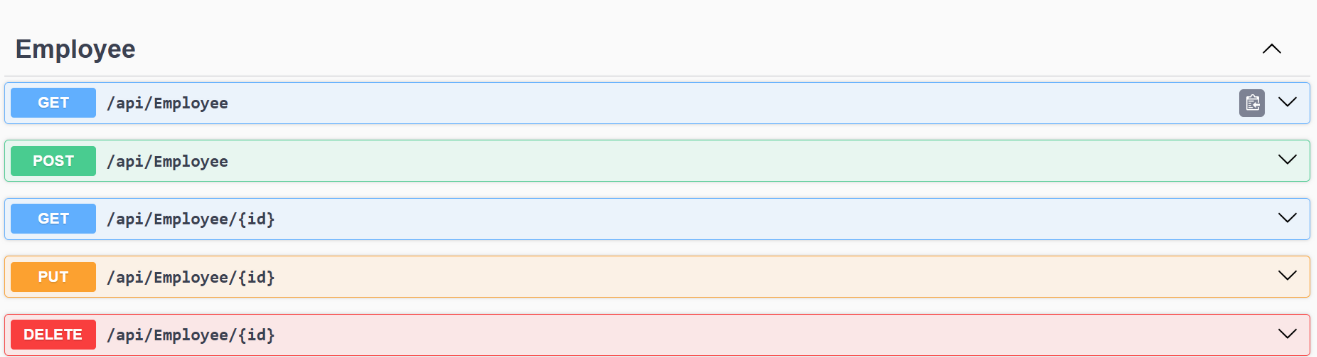
}

},

"AllowedHosts": "\*"

}

**OUTPUT:-**



**3.**

<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>

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

GET {{EmployeeWebApi\_HostAddress}}/weatherforecast/

Accept: application/json

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();

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

}

}

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

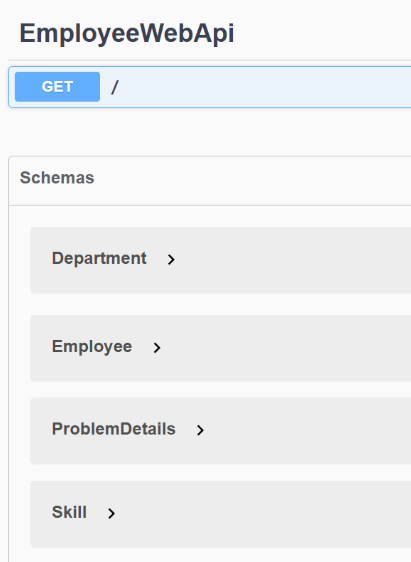
GlobalSection(ExtensibilityGlobals) = postSolution

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

EndGlobalSection

EndGlobal

**OUTPUT:-**



**4.**

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

<PropertyGroup>

<TargetFramework>net9.0</TargetFramework>

<Nullable>enable</Nullable>

<ImplicitUsings>enable</ImplicitUsings>

</PropertyGroup>

<ItemGroup>

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

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

</ItemGroup>

</Project>

@EmployeeCrudAPI\_HostAddress = http://localhost:5116

GET {{EmployeeCrudAPI\_HostAddress}}/weatherforecast/

Accept: application/json

using Microsoft.OpenApi.Models;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container

builder.Services.AddControllers();

// Add Swagger services

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen(c =>

{

c.SwaggerDoc("v1", new OpenApiInfo

{

Title = "Employee CRUD API",

Version = "v1",

Description = "API for Employee CRUD Operations"

});

});

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 CRUD API V1");

c.RoutePrefix = string.Empty; // Makes Swagger UI available at root

});

}

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

}

}

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}") = "EmployeeCrudAPI", "EmployeeCrudAPI\EmployeeCrudAPI.csproj", "{82401007-9F5E-FDEE-7F56-0BAA78E8DB3E}"

EndProject

Global

GlobalSection(SolutionConfigurationPlatforms) = preSolution

Debug|Any CPU = Debug|Any CPU

Release|Any CPU = Release|Any CPU

EndGlobalSection

GlobalSection(ProjectConfigurationPlatforms) = postSolution

{82401007-9F5E-FDEE-7F56-0BAA78E8DB3E}.Debug|Any CPU.ActiveCfg = Debug|Any CPU

{82401007-9F5E-FDEE-7F56-0BAA78E8DB3E}.Debug|Any CPU.Build.0 = Debug|Any CPU

{82401007-9F5E-FDEE-7F56-0BAA78E8DB3E}.Release|Any CPU.ActiveCfg = Release|Any CPU

{82401007-9F5E-FDEE-7F56-0BAA78E8DB3E}.Release|Any CPU.Build.0 = Release|Any CPU

EndGlobalSection

GlobalSection(SolutionProperties) = preSolution

HideSolutionNode = FALSE

EndGlobalSection

GlobalSection(ExtensibilityGlobals) = postSolution

SolutionGuid = {5492F1D9-3195-4581-8646-D991C1B2EF7C}

EndGlobalSection

EndGlobal

**OUTPUT:-**



**5.**

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();

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"AllowedHosts": "\*"

}

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

}

}

<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>

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

GET {{fifth\_HostAddress}}/weatherforecast/

Accept: application/json

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

**OUTPUT:-**

