1. WebApi Handson

Code :-

**Program.cs :**

using Microsoft.OpenApi.Models;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();

builder.Services.AddEndpointsApiExplorer();

// Custom Swagger setup

builder.Services.AddSwaggerGen(c =>

{

c.SwaggerDoc("v1", new OpenApiInfo

{

Title = "Swagger Demo",

Version = "v1",

Description = "TBD",

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

Contact = new OpenApiContact

{

Name = "John Doe",

Email = "john@xyzmail.com",

Url = new Uri("https://www.example.com")

},

License = new OpenApiLicense

{

Name = "License Terms",

Url = new Uri("https://www.example.com")

}

});

});

var app = builder.Build();

// Configure the HTTP request pipeline.

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI(c =>

{

c.SwaggerEndpoint("/swagger/v1/swagger.json", "Swagger Demo");

});

}

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();

**EmployeeController.cs**

using Microsoft.AspNetCore.Mvc;

using System.Collections.Generic;

namespace MyFirstWebAPI.Controllers

{

[ApiController]

[Route("emp")] // Custom route

public class EmployeeController : ControllerBase

{

private static List<string> employees = new List<string> { "Alice", "Bob", "Charlie" };

// GET /emp

[HttpGet]

[ProducesResponseType(200)]

public IActionResult GetEmployees()

{

return Ok(employees);

}

// POST /emp

[HttpPost]

[ProducesResponseType(200)]

public IActionResult AddEmployee([FromBody] string name)

{

employees.Add(name);

return Ok(employees);

}

// PUT /emp/{id}

[HttpPut("{id}")]

[ActionName("UpdateEmployee")]

[ProducesResponseType(200)]

[ProducesResponseType(404)]

public IActionResult UpdateEmployee(int id, [FromBody] string name)

{

if (id < 0 || id >= employees.Count)

return NotFound("Employee not found");

employees[id] = name;

return Ok(employees);

}

// DELETE /emp/{id}

[HttpDelete("{id}")]

[ActionName("RemoveEmployee")]

[ProducesResponseType(200)]

[ProducesResponseType(404)]

public IActionResult RemoveEmployee(int id)

{

if (id < 0 || id >= employees.Count)

return NotFound("Employee not found");

employees.RemoveAt(id);

return Ok(employees);

}

}

}

Output :



