****Assignment 2:****

## ****1. Objective****

* To enhance a previously created .NET 8 WebAPI by:
* Integrating Swagger for documentation
* Adding a new controller: EmployeeController
* Implementing custom routing and ActionName attributes
* Testing endpoints using Postman and Swagger UI

## ****2. Project Setup – Starting from Assignment 1****

**Modification:**

Used the existing WebAPI project FirstWebAPI created in Assignment 1.

bash

CopyEdit

cd FirstWebAPI

1. ****Installed Swagger Support****

**Modification:**

Added Swagger support by installing Swashbuckle.AspNetCore into the existing project.

bash

CopyEdit

dotnet add package Swashbuckle.AspNetCore

## ****4.** Modified Program.cs to Enable Swagger**

🛠️ **Modification:**

Updated the existing Program.cs to include Swagger services and UI configuration.

csharp

CopyEdit

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers();

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen(options =>

{

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

{

Title = "Swagger Demo",

Version = "v1",

Description = "TBD",

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

Contact = new Microsoft.OpenApi.Models.OpenApiContact

{

Name = "John Doe",

Email = "john@xyzmail.com",

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

},

License = new Microsoft.OpenApi.Models.OpenApiLicense

{

Name = "License Terms",

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

}

});

});

var app = builder.Build();

app.UseSwagger();

app.UseSwaggerUI();

app.UseHttpsRedirection();

app.MapControllers();

app.Run();

## **Created**EmployeeController.cs

**Path:** FirstWebAPI/Controllers/EmployeeController.cs

csharp

CopyEdit

using Microsoft.AspNetCore.Mvc;

using System.Collections.Generic;

namespace FirstWebAPI.Controllers

{

[ApiController]

[Route("api/emp")]

public class EmployeeController : ControllerBase

{

[HttpGet]

public ActionResult<IEnumerable<string>> GetEmployees()

{

return Ok(new List<string> { "John", "Jane", "Alice" });

}

[HttpGet("all")]

[ActionName("all")]

public ActionResult<IEnumerable<string>> GetAllEmployees()

{

return Ok(new List<string> { "John", "Jane" });

}

[HttpGet("single/{id}")]

[ActionName("single")]

public ActionResult<string> GetEmployeeById(int id)

{

return Ok($"Employee with ID: {id}");

}

}

}

1. ****Ran and Opened Swagger UI****

After running dotnet run, tested the Swagger UI with the newly added controller and endpoints.

bash

CopyEdit

dotnet run

Opened in browser:

bash

CopyEdit

<http://localhost:5249/swagger>

## ****7. Test in Postman****

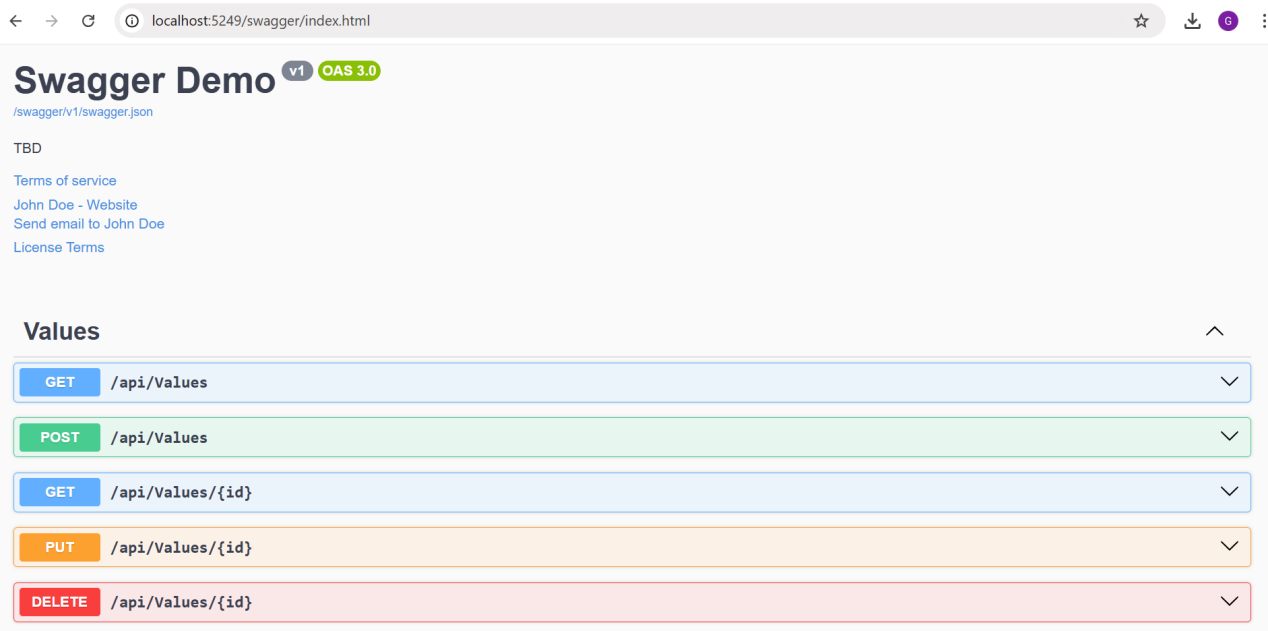
🛠️ **Modification:**

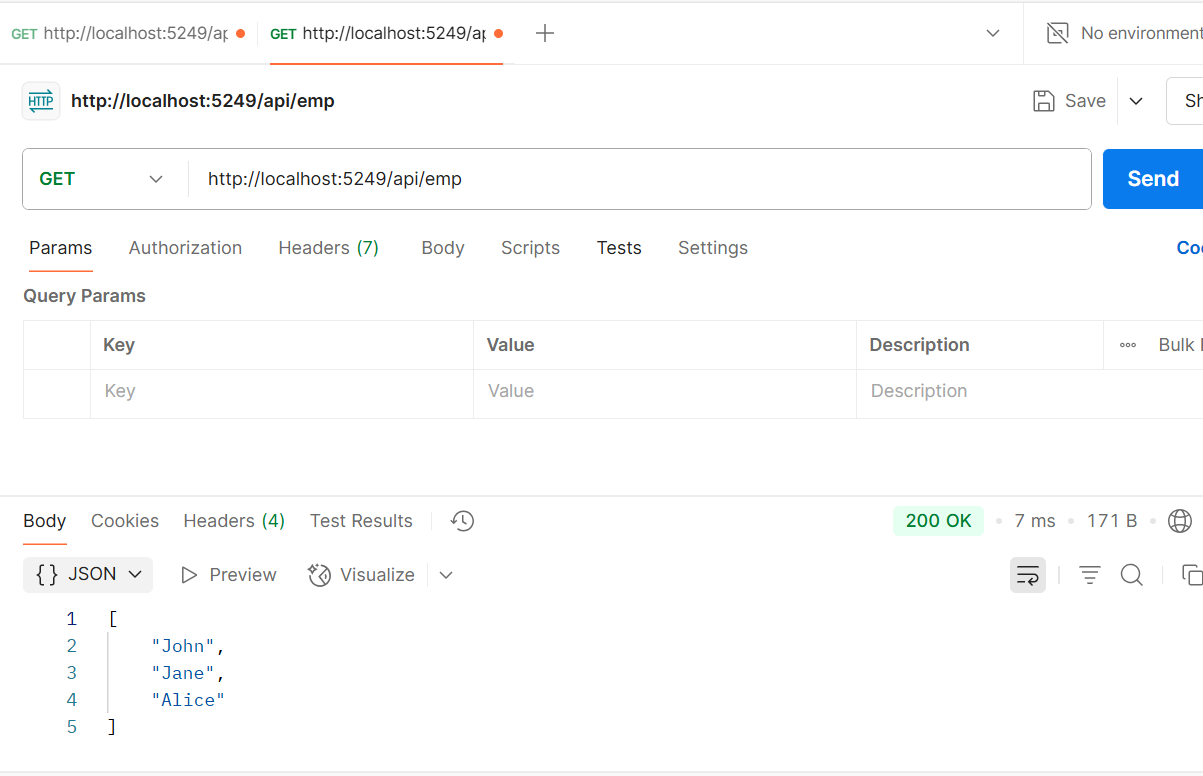
Used **Postman** to send GET requests to the new routes. No headers or body required.

| **Method** | **URL** | **Description** |
| --- | --- | --- |
| GET | http://localhost:5249/api/emp | Get default employee list |
| GET | http://localhost:5249/api/emp/all | List all employees (action) |
| GET | http://localhost:5249/api/emp/single/1 | Get employee by ID |

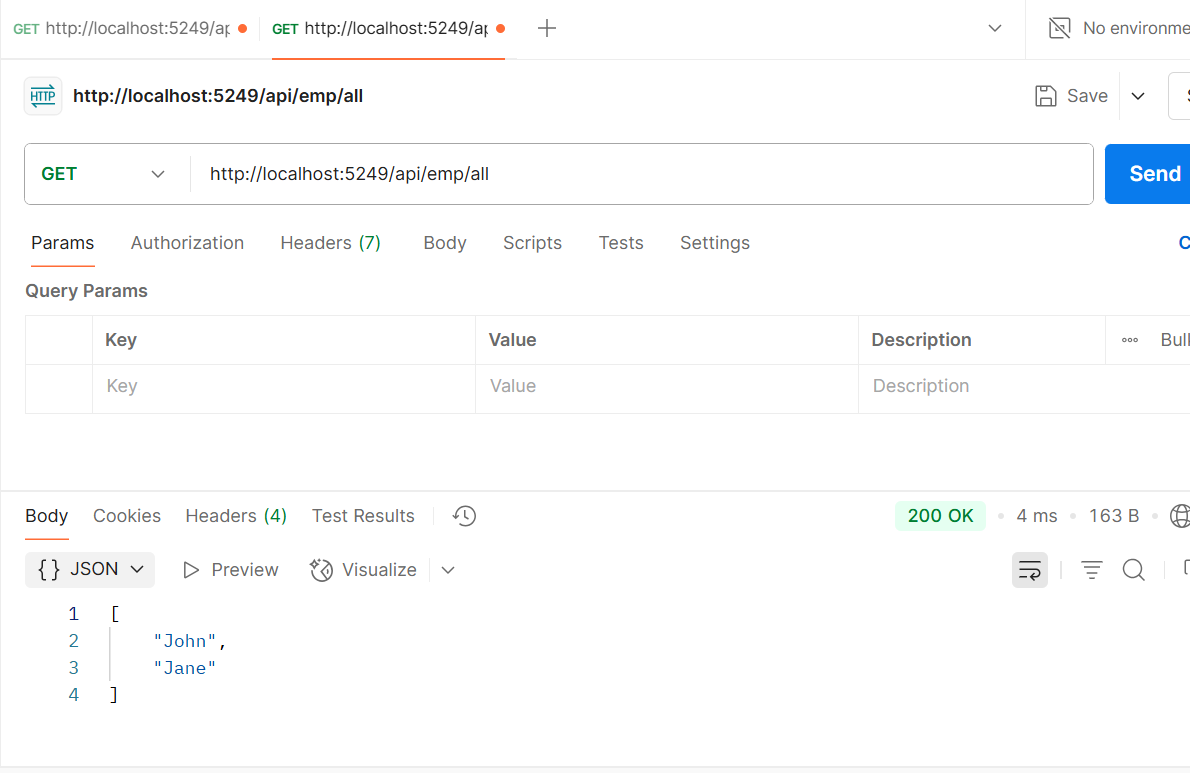
## ****Screenshots :****

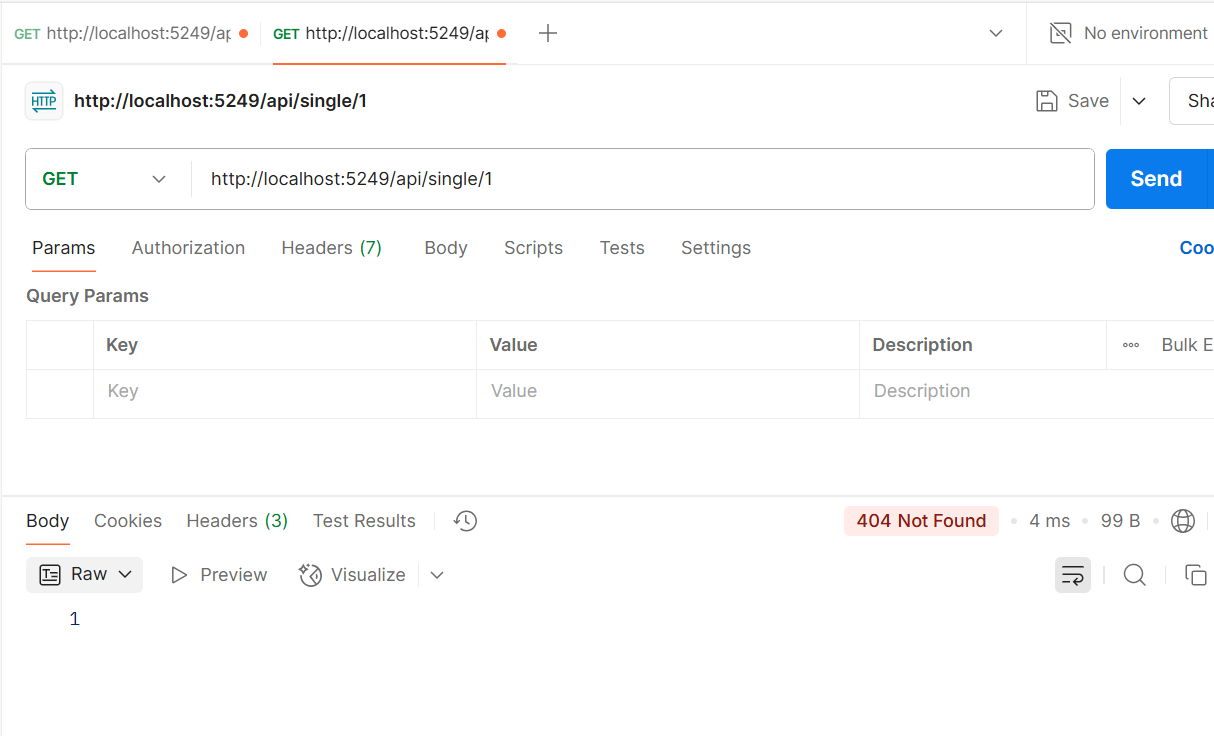
Swagger UI showing EmployeeController:



********GET /api/emp in Postman:

GET /api/emp/all in Postman:

********

GET /api/emp/single/1 in Postman********: