3 . Web Api using custom model class

Create a Custom class ‘Employee’ of the below defined structure

2. Create a Custom action filter for Authorization.

CODE:

CustomExceptionFilter.cs:

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Filters;

namespace EmployeeWebApi.Filters

{

    public class CustomExceptionFilter : IExceptionFilter

    {

        public void OnException(ExceptionContext context)

        {

            var log = $"[{DateTime.Now}] {context.Exception.Message}\n{context.Exception.StackTrace}";

            File.AppendAllText("exception\_log.txt", log + "\n\n");

            context.Result = new ObjectResult("Internal server error")

            {

                StatusCode = StatusCodes.Status500InternalServerError

            };

        }

    }

}

CustomAuthFilter.cs:  
using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Filters;

namespace EmployeeWebApi.Filters

{

    public class CustomAuthFilter : ActionFilterAttribute

    {

        public override void OnActionExecuting(ActionExecutingContext context)

        {

            var hasHeader = context.HttpContext.Request.Headers.TryGetValue("Authorization", out var token);

            if (!hasHeader)

            {

                context.Result = new BadRequestObjectResult("Invalid request - No Auth token");

                return;

            }

            if (!token.ToString().Contains("Bearer"))

            {

                context.Result = new BadRequestObjectResult("Invalid request - Token present but Bearer unavailable");

            }

        }

    }

}

EmployeeController.cs:

using EmployeeWebApi.Models;

using Microsoft.AspNetCore.Mvc;

using EmployeeWebApi.Filters;

namespace EmployeeWebApi.Controllers

{

    [ApiController]

    [Route("api/[controller]")]

    [ServiceFilter(typeof(CustomAuthFilter))]

    public class EmployeeController : ControllerBase

    {

        private static List<Employee> employeeList;

        static EmployeeController()

        {

            employeeList = GetStandardEmployeeList();

        }

        [HttpGet]

        [ProducesResponseType(StatusCodes.Status200OK)]

        [ProducesResponseType(StatusCodes.Status500InternalServerError)]

        public ActionResult<List<Employee>> Get()

        {

            // Uncomment below to simulate exception

            // throw new Exception("Simulated exception");

            return Ok(employeeList);

        }

        [HttpGet("standard")]

        public ActionResult<Employee> GetStandrad()

        {

            return Ok(GetStandardEmployeeList().FirstOrDefault());

        }

        [HttpPost]

        public IActionResult Post([FromBody] Employee emp)

        {

            employeeList.Add(emp);

            return Ok("Employee added");

        }

        [HttpPut("{id}")]

        public IActionResult Put(int id, [FromBody] Employee emp)

        {

            var existing = employeeList.FirstOrDefault(e => e.Id == id);

            if (existing == null) return NotFound();

            existing.Name = emp.Name;

            existing.Salary = emp.Salary;

            existing.Permanent = emp.Permanent;

            existing.Department = emp.Department;

            existing.Skills = emp.Skills;

            existing.DateOfBirth = emp.DateOfBirth;

            return Ok("Employee updated");

        }

        private static List<Employee> GetStandardEmployeeList()

        {

            return new List<Employee>

            {

                new Employee

                {

                    Id = 1,

                    Name = "Rahul",

                    Salary = 50000,

                    Permanent = true,

                    DateOfBirth = new DateTime(1990, 1, 1),

                    Department = new Department { Id = 101, Name = "IT" },

                    Skills = new List<Skill>

                    {

                        new Skill { Id = 1, Name = "C#" },

                        new Skill { Id = 2, Name = "ASP.NET" }

                    }

                }

            };

        }

    }

}

Empolyee.cs:

using System;

using System.Collections.Generic;

namespace EmployeeWebApi.Models

{

    public class Employee

    {

        public int Id { get; set; }

        public string Name { get; set; }

        public int Salary { get; set; }

        public bool Permanent { get; set; }

        public Department Department { get; set; }

        public List<Skill> Skills { get; set; }

        public DateTime DateOfBirth { get; set; }

    }

}

Skill.cs:  
namespace EmployeeWebApi.Models

{

    public class Skill

    {

        public int Id { get; set; }

        public string Name { get; set; }

    }

}

Department.cs:  
namespace EmployeeWebApi.Models

{

    public class Department

    {

        public int Id { get; set; }

        public string Name { get; set; }

    }

}

Program.cs:  
using EmployeeWebApi.Filters;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers(options =>

{

    options.Filters.Add<CustomExceptionFilter>(); // Register exception filter globally

});

builder.Services.AddScoped<CustomAuthFilter>(); // Register custom auth filter

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

var app = builder.Build();

app.UseSwagger();

app.UseSwaggerUI();

app.UseAuthorization();

app.MapControllers();

app.Run();

OUTPUT:

C:\Users\rahul\FirstWebApi> info: Microsoft.Hosting.Lifetime[14]

>> Now listening on: http://localhost:5283

>> info: Microsoft.Hosting.Lifetime[0]

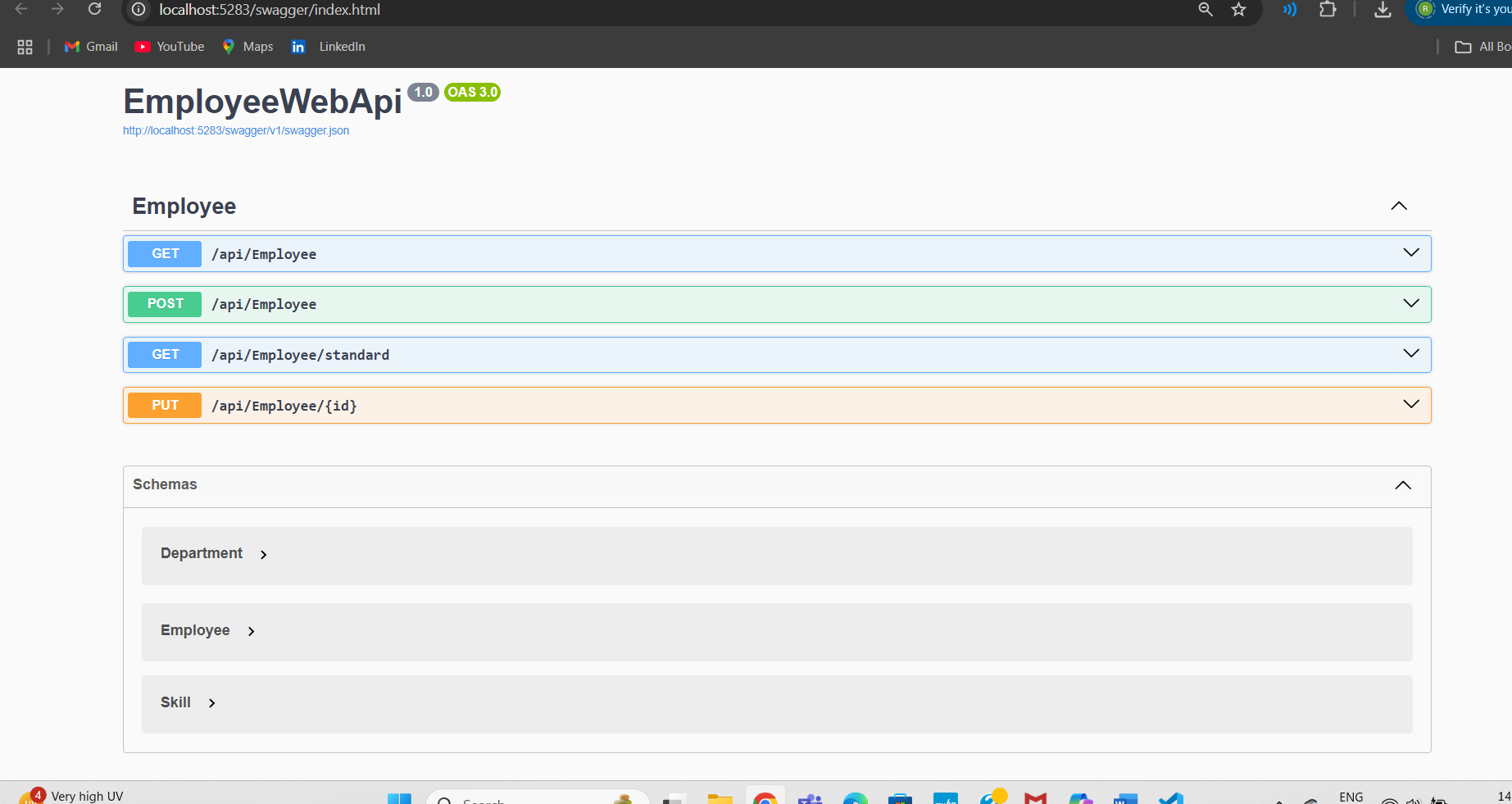
>> Application started. Press Ctrl+C to shut down.

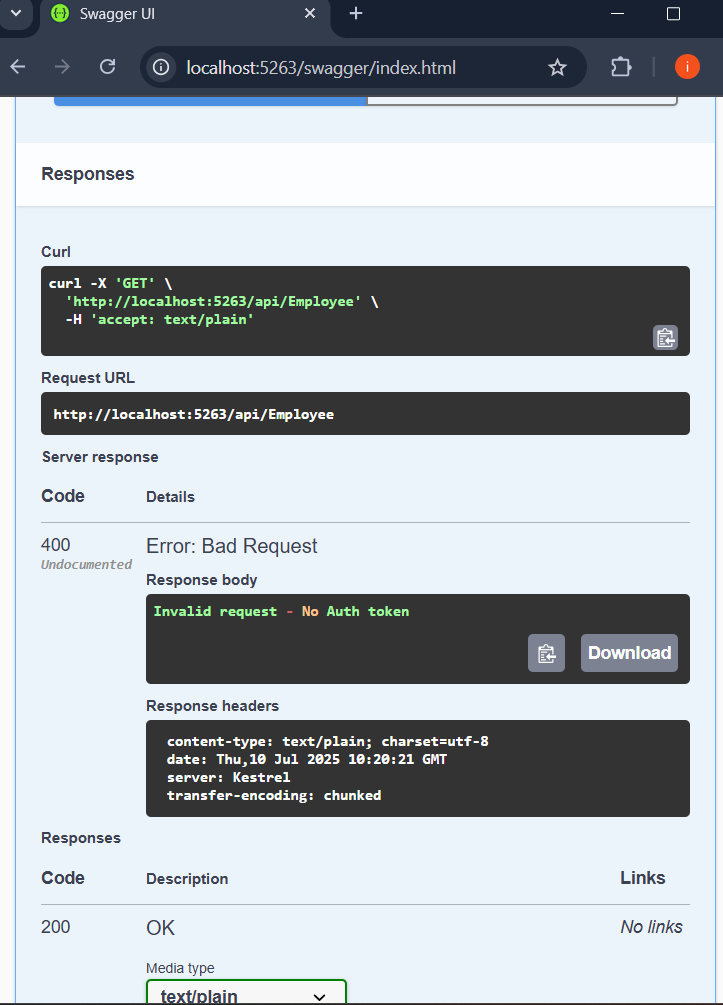
>> info: Microsoft.Hosting.Lifetime[0]

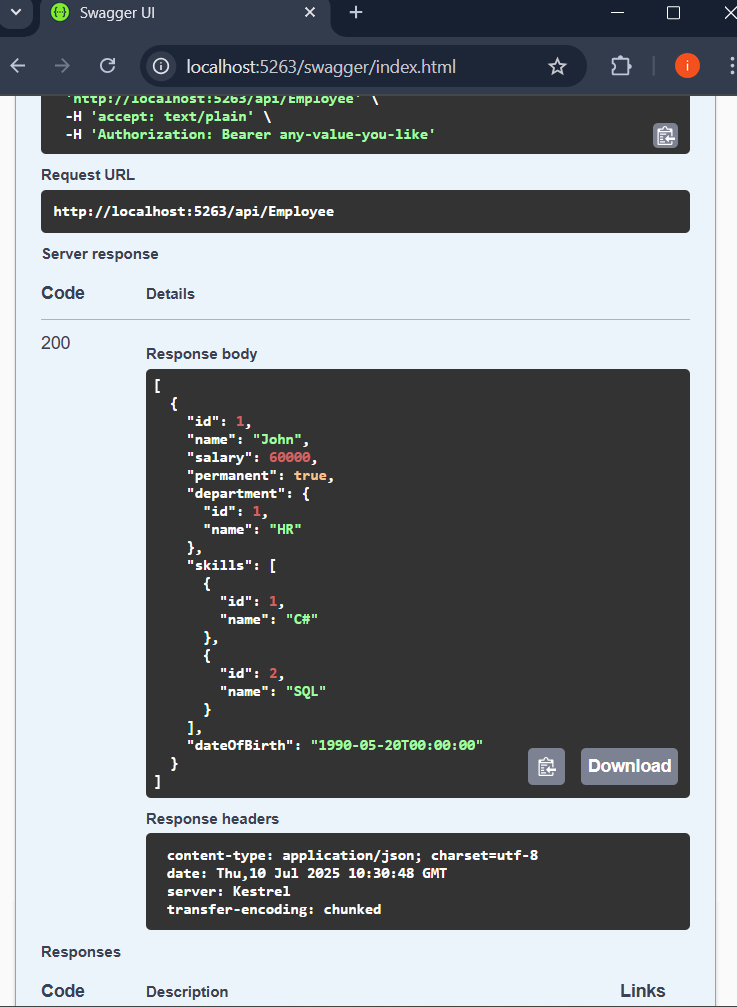
>> Hosting environment: Development

>> info: Microsoft.Hosting.Lifetime[0]

>> Content root path: C:\Users\rahul\FirstWebApi\EmployeeWebApi



****

****