Name:-Adnan Hasan

GitHub Link:-

[https://github.com/adnan-984/Cognizant\_6364530\_Adnan](#_top)CognizantDotNet

**-----------------------------------------------------------------------------------------**

**Web API (Handson 04)**EmployeeController.cs  
  
using System;

using System.Collections.Generic;

using System.Linq;

using Microsoft.AspNetCore.Mvc;

using SwaggerDemoAPI.Models;

using SwaggerDemoAPI.Filters;

namespace SwaggerDemoAPI.Controllers

{

[ApiController]

[Route("api/[controller]")]

[ServiceFilter(typeof(CustomAuthFilter))]

public class EmployeeController : ControllerBase

{

private static List<Employee> employees = new List<Employee>

{

new Employee

{

Id = 1,

Name = "Ritesh",

Salary = 50000,

Permanent = true,

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

Skills = new List<Skill>

{

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

new Skill { Id = 2, Name = "SQL" }

},

DateOfBirth = new DateTime(2003, 12, 01)

}

};

[HttpGet("GetStandard")]

[ProducesResponseType(StatusCodes.Status500InternalServerError)]

public ActionResult<List<Employee>> GetStandard()

{

throw new Exception("Deliberate error for testing");

}

[HttpPost]

[ProducesResponseType(StatusCodes.Status200OK)]

public IActionResult AddEmployee([FromBody] Employee employee)

{

employees.Add(employee);

return Ok("Employee Added");

}

[HttpPut]

[ProducesResponseType(StatusCodes.Status200OK)]

[ProducesResponseType(StatusCodes.Status400BadRequest)]

public ActionResult<Employee> UpdateEmployee([FromBody] Employee updatedEmployee)

{

if (updatedEmployee.Id <= 0)

{

return BadRequest("Invalid employee id");

}

var employee = employees.FirstOrDefault(e => e.Id == updatedEmployee.Id);

if (employee == null)

{

return BadRequest("Invalid employee id");

}

employee.Name = updatedEmployee.Name;

employee.Salary = updatedEmployee.Salary;

employee.Permanent = updatedEmployee.Permanent;

employee.Department = updatedEmployee.Department;

employee.Skills = updatedEmployee.Skills;

employee.DateOfBirth = updatedEmployee.DateOfBirth;

return Ok(employee);

}

}

}  
  
**Program.cs**  
  
using Microsoft.AspNetCore.Builder;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

using SwaggerDemoAPI.Filters;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddScoped<CustomAuthFilter>();

builder.Services.AddScoped<CustomExceptionFilter>();

builder.Services.AddControllers(options =>

{

options.Filters.Add<CustomExceptionFilter>();

});

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen(c =>

{

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

{

Title = "Swagger Demo",

Version = "v1",

Description = "TBD",

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

Contact = new Microsoft.OpenApi.Models.OpenApiContact

{

Name = "John Doe",

Email = "john@xyzmail.com",

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

},

License = new Microsoft.OpenApi.Models.OpenApiLicense

{

Name = "License Terms",

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

}

});

});

var app = builder.Build();

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();

## 