Name:-Ritesh Kumar Sahoo

GitHub Link:-[https://github.com/Riteshksahoo/Cognizant\_6364581\_Ritesh](#_top)CognizantDotNet

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

**Web API (Handson 03)  
  
1. Web Api using custom model class**

**Models--> Department.cs**namespace SwaggerDemoAPI.Models

{

public class Department

{

public int Id { get; set; }

public string Name { get; set; }

}

}  
  
**Models--> Employee.cs**using System;

using System.Collections.Generic;

using SwaggerDemoAPI.Models;

namespace SwaggerDemoAPI.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; }

}

}  
  
**Models--> Skill.cs**namespace SwaggerDemoAPI.Models

{

public class Skill

{

public int Id { get; set; }

public string Name { get; set; }

}

}  
  
**Controllers-->EmployeeController.cs**  
  
using System.Collections.Generic;

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

using SwaggerDemoAPI.Models;

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

[ApiController]

public class EmployeeController : ControllerBase

{

private List<Employee> employees;

public EmployeeController()

{

employees = GetStandardEmployeeList();

}

private List<Employee> GetStandardEmployeeList()

{

return 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(1998, 05, 21)

}

};

}

[HttpGet("GetStandard")]

[ProducesResponseType(StatusCodes.Status200OK)]

[AllowAnonymous]

public ActionResult<List<Employee>> GetStandard()

{

return Ok(employees);

}

[HttpPost]

[ProducesResponseType(StatusCodes.Status200OK)]

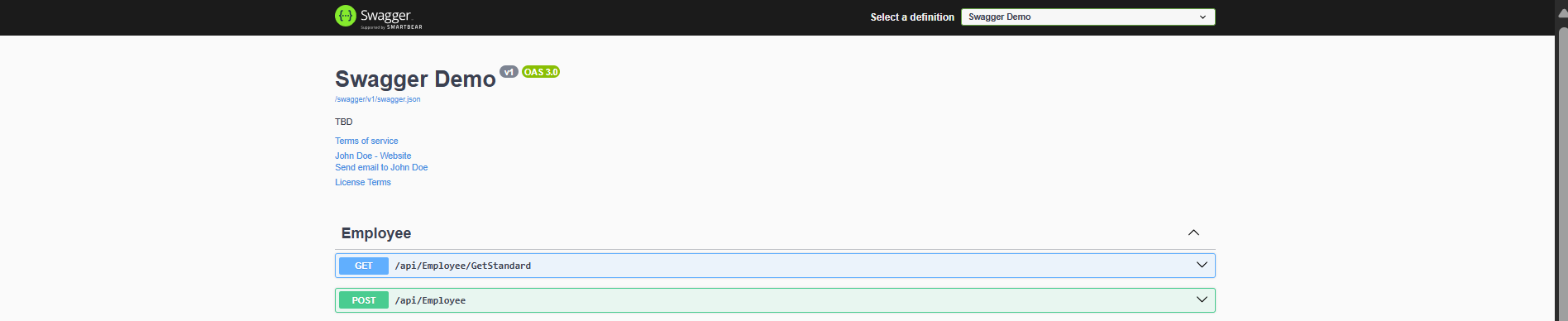
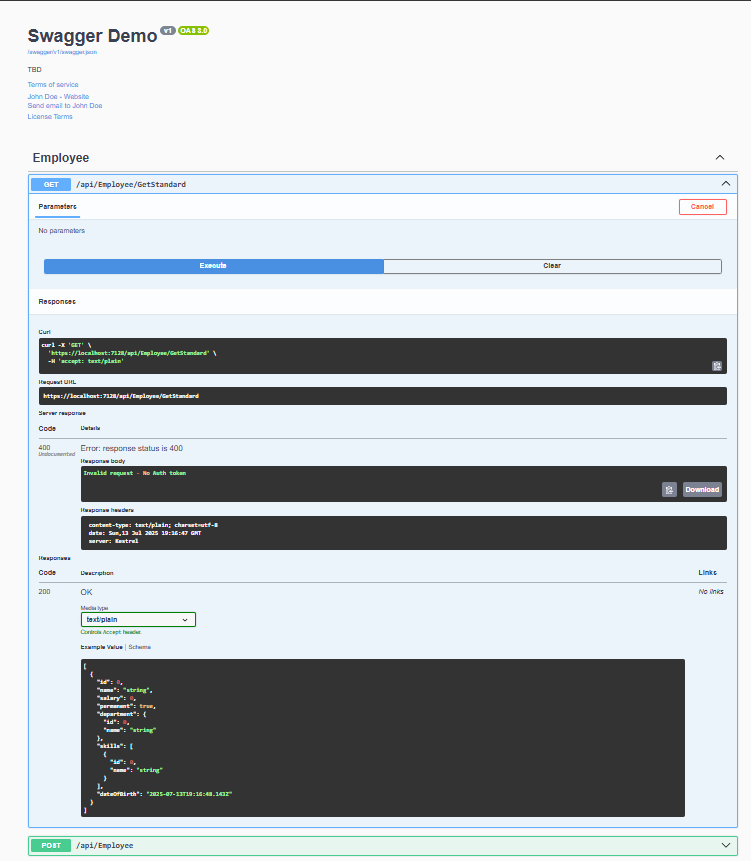
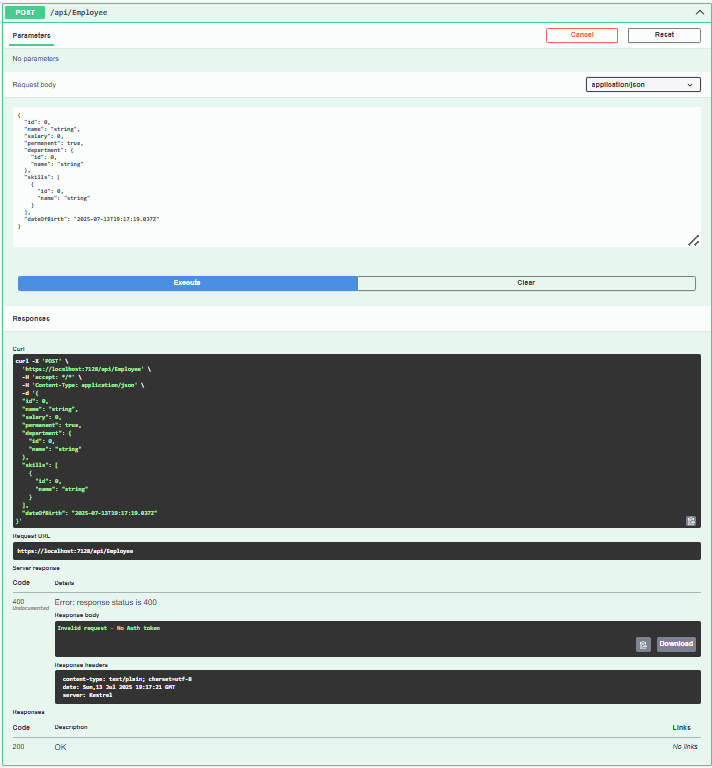
public IActionResult AddEmployee([FromBody] Employee employee)

{

employees.Add(employee);

return Ok("Employee Added");

}

}  
  
  
  


**2.Create a Custom action filter for Authorization.**using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Filters;

using System.Linq;

public class CustomAuthFilter : ActionFilterAttribute

{

public override void OnActionExecuting(ActionExecutingContext context)

{

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

if (!hasAuth)

{

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

return;

}

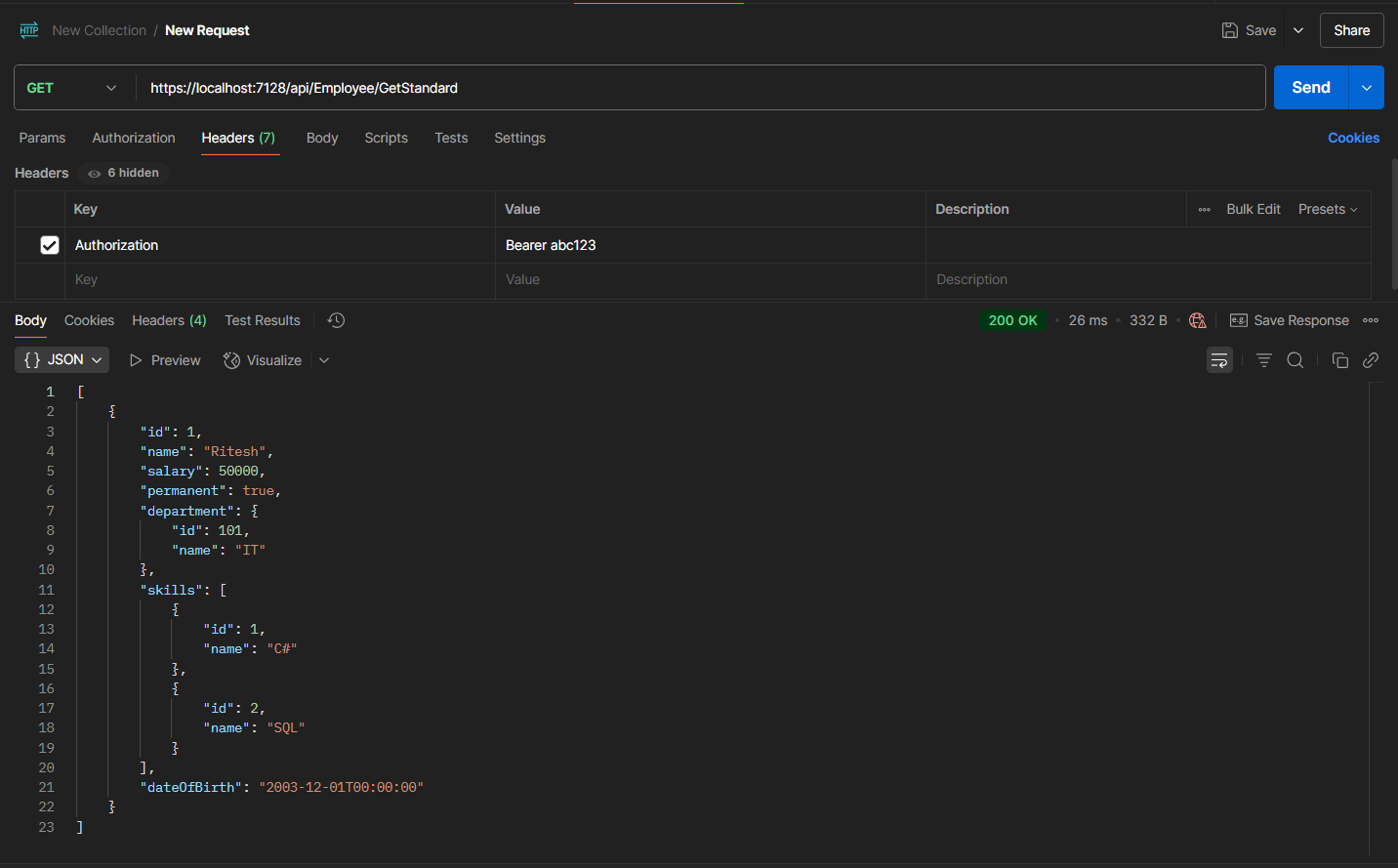
if (!token.ToString().Contains("Bearer"))

{

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

}

}

}  
  


**3.Custom Exception filter  
  
CustomExceptionFilter.cs**using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Filters;

using System;

using System.IO;

namespace SwaggerDemoAPI.Filters

{

public class CustomExceptionFilter : IExceptionFilter

{

public void OnException(ExceptionContext context)

{

var exception = context.Exception;

var message = $"Exception: {exception.Message} | StackTrace: {exception.StackTrace}";

File.AppendAllText("C:\\Logs\\error\_log.txt", message + Environment.NewLine);

context.Result = new ObjectResult("Internal Server Error. Please contact admin.")

{

StatusCode = 500

};

}

}

}

**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<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();

**EmployeeController**using System.Collections.Generic;

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

using SwaggerDemoAPI.Models;

[ServiceFilter(typeof(CustomAuthFilter))]

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

[ApiController]

public class EmployeeController : ControllerBase

{

private List<Employee> employees;

public EmployeeController()

{

employees = GetStandardEmployeeList();

}

private List<Employee> GetStandardEmployeeList()

{

return 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");

}

}

