**Handson 3: Web API**

Superset id: 6363535

[**Program.cs**](http://program.cs)

using EmployeeApiDemo.Filters;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();

// Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

builder.Services.AddScoped<CustomAuthFilter>();

var app = builder.Build();

// Configure the HTTP request pipeline.

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI();

}

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();

**Controller-**

[**EmployeeController.cs**](http://employeecontroller.cs)

using Microsoft.AspNetCore.Mvc;

using EmployeeApiDemo.Models;

using EmployeeApiDemo.Filters;

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

[ApiController]

[ServiceFilter(typeof(CustomAuthFilter))]

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 = "Alice",

Salary = 50000,

Permanent = true,

Department = new Department { Id = 1, Name = "HR" },

Skills = new List<Skill>

{

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

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

},

DateOfBirth = new DateTime(1990, 1, 1)

}

};

}

[HttpGet]

[ProducesResponseType(StatusCodes.Status200OK)]

public ActionResult<List<Employee>> Get()

{

return Ok(employees);

}

[HttpPost]

public IActionResult Post([FromBody] Employee emp)

{

employees.Add(emp);

return Ok(emp);

}

}

**Filters-**

[**CustomAuthFilter.cs**](http://customauthfilter.cs)

namespace EmployeeApiDemo.Filters;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.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");

return;

}

base.OnActionExecuting(context);

}

}

[**CustomExceptionFilter.cs**](http://customexceptionfilter.cs)

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Filters;

namespace EmployeeApiDemo.Filters

{

public class CustomExceptionFilter : IExceptionFilter

{

public void OnException(ExceptionContext context)

{

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

File.AppendAllText("errors.txt", log);

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

{

StatusCode = 500

};

}

}

}

**Models-**

[**Department.cs**](http://department.cs)

namespace EmployeeApiDemo.Models

{

public class Department

{

public int Id { get; set; }

public string Name { get; set; }

}

}

**Skill**[**.cs**](http://department.cs)

namespace EmployeeApiDemo.Models

{

public class Skill

{

public int Id { get; set; }

public string Name { get; set; }

}

}

**Employee.cs**

namespace EmployeeApiDemo.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; }

}

}









