using System;

using System.Collections.Generic;

public class Department { public int Id { get; set; } public string Name { get; set; } }

public class Skill { public int Id { get; set; } public string Name { get; set; } }

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; }

}

public class CustomAuthFilter {

public static void CheckAuth(Dictionary<string, string> headers) {

if (!headers.ContainsKey("Authorization"))

throw new Exception("Invalid request - No Auth token");

if (!headers["Authorization"].Contains("Bearer"))

throw new Exception("Invalid request - Token present but Bearer unavailable");

}

}

public class CustomExceptionFilter {

public static void LogException(Exception ex) {

Console.WriteLine("===== Exception Caught by Filter =====");

Console.WriteLine($"Message: {ex.Message}");

Console.WriteLine($"StackTrace: {ex.StackTrace}");

Console.WriteLine("500 Internal Server Error - Exception handled gracefully.");

}

}

public class EmployeeController {

private List<Employee> \_employees;

public EmployeeController() { \_employees = GetStandardEmployeeList(); }

private List<Employee> GetStandardEmployeeList() {

return new List<Employee> {

new Employee {

Id = 1,

Name = "John",

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)

}

};

}

public List<Employee> Get(Dictionary<string, string> headers) {

CustomAuthFilter.CheckAuth(headers);

Console.WriteLine("200 OK - Returning Employee List");

return \_employees;

}public string Post(Employee employee, Dictionary<string, string> headers) {

CustomAuthFilter.CheckAuth(headers);

\_employees.Add(employee);

return $"200 OK - Employee {employee.Name} added.";}

public List<Employee> GetWithException(Dictionary<string, string> headers) {

CustomAuthFilter.CheckAuth(headers);

throw new Exception("Simulated exception in GET");}}public class Program {

public static void Main() {

var headersWithAuth = new Dictionary<string, string> { { "Authorization", "Bearer sample-token" } };

var controller = new EmployeeController();try {

Console.WriteLine("=== GET Employees ===");

var result = controller.Get(headersWithAuth);

foreach (var emp in result) Console.WriteLine($"Id: {emp.Id}, Name: {emp.Name}, Department: {emp.Department.Name}");

Console.WriteLine("\n=== POST Employee ===");var newEmp = new Employee {

Id = 2,

Name = "Alice",

Salary = 60000,

Permanent = false,

Department = new Department { Id = 2, Name = "IT" },

Skills = new List<Skill> { new Skill { Id = 3, Name = "Python" } },

DateOfBirth = new DateTime(1992, 2, 2)};

var postResult = controller.Post(newEmp, headersWithAuth);

Console.WriteLine(postResult);

Console.WriteLine("\n=== GET with Exception ===");

controller.GetWithException(headersWithAuth);} catch (Exception ex) {

CustomExceptionFilter.LogException(ex);}}}

