1. **WebApi\_Handson**

**CODE:**

**#ValuesController.cs**

using Microsoft.AspNetCore.Mvc;

using System.Collections.Generic;

namespace MyFirstWebAPI.Controllers

{

// Wrapper model class for string input

public class ValueWrapper

{

public string Value { get; set; }

}

[ApiController]

[Route("[controller]")]

public class ValuesController : ControllerBase

{

private static List<string> data = new() { "value1", "value2" };

[HttpGet]

[ProducesResponseType(StatusCodes.Status200OK)]

public IActionResult Get()

{

return Ok(data); // Returns 200 OK

}

[HttpPost]

[ProducesResponseType(StatusCodes.Status200OK)]

[ProducesResponseType(StatusCodes.Status400BadRequest)]

public IActionResult Post([FromBody] ValueWrapper wrapper)

{

if (wrapper == null || string.IsNullOrWhiteSpace(wrapper.Value))

return BadRequest("Invalid input");

data.Add(wrapper.Value);

return Ok("Added");

}

[HttpPut("{index}")]

[ProducesResponseType(StatusCodes.Status200OK)]

[ProducesResponseType(StatusCodes.Status400BadRequest)]

public IActionResult Put(int index, [FromBody] ValueWrapper wrapper)

{

if (index < 0 || index >= data.Count)

return BadRequest("Invalid Index");

if (wrapper == null || string.IsNullOrWhiteSpace(wrapper.Value))

return BadRequest("Invalid input");

data[index] = wrapper.Value;

return Ok("Updated");

}

[HttpDelete("{index}")]

[ProducesResponseType(StatusCodes.Status200OK)]

[ProducesResponseType(StatusCodes.Status400BadRequest)]

public IActionResult Delete(int index)

{

if (index < 0 || index >= data.Count)

return BadRequest("Invalid Index");

data.RemoveAt(index);

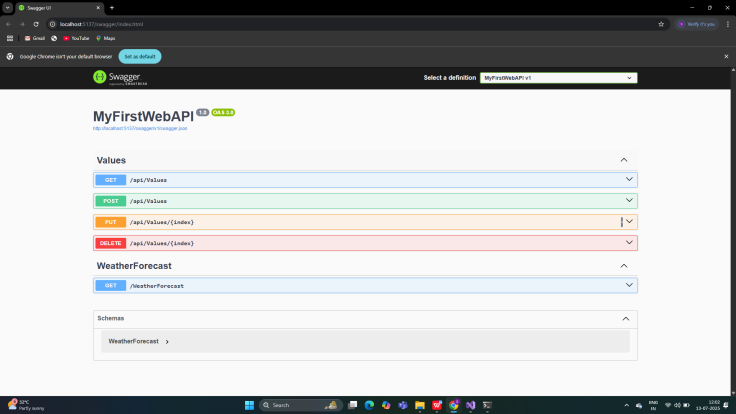
return Ok("Deleted");

}

}

}

**OUTPUT:**

****

1. **WebApi\_Handson**

**CODE:**

**#EmployeeController.cs**

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using System.Collections.Generic;

namespace MyFirstWebAPI.Controllers

{

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

//[ApiController]

[ApiController]

[Route("api/Emp")] // Changed from "api/Employee" to "api/Emp"

public class EmployeeController : ControllerBase

{

private static List<object> employees = new()

{

new { Id = 1, Name = "John", Designation = "Manager" },

new { Id = 2, Name = "Sara", Designation = "Developer" }

};

[HttpGet]

[ProducesResponseType(StatusCodes.Status200OK)]

public IActionResult GetEmployees()

{

return Ok(employees); // Return the list of employees

}

}

}

**#program.cs**

using Microsoft.OpenApi.Models;

var builder = WebApplication.CreateBuilder(args);

// Add services

builder.Services.AddControllers();

builder.Services.AddEndpointsApiExplorer();

// Add Swagger configuration

builder.Services.AddSwaggerGen(c =>

{

c.SwaggerDoc("v1", new OpenApiInfo

{

Title = "Swagger Demo",

Version = "v1",

Description = "TBD",

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

Contact = new OpenApiContact

{

Name = "John Doe",

Email = "john@xyzmail.com",

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

},

License = new OpenApiLicense

{

Name = "License Terms",

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

}

});

});

var app = builder.Build();

// Configure the HTTP request pipeline

app.UseSwagger();

app.UseSwaggerUI(c =>

{

c.SwaggerEndpoint("/swagger/v1/swagger.json", "Swagger Demo");

});

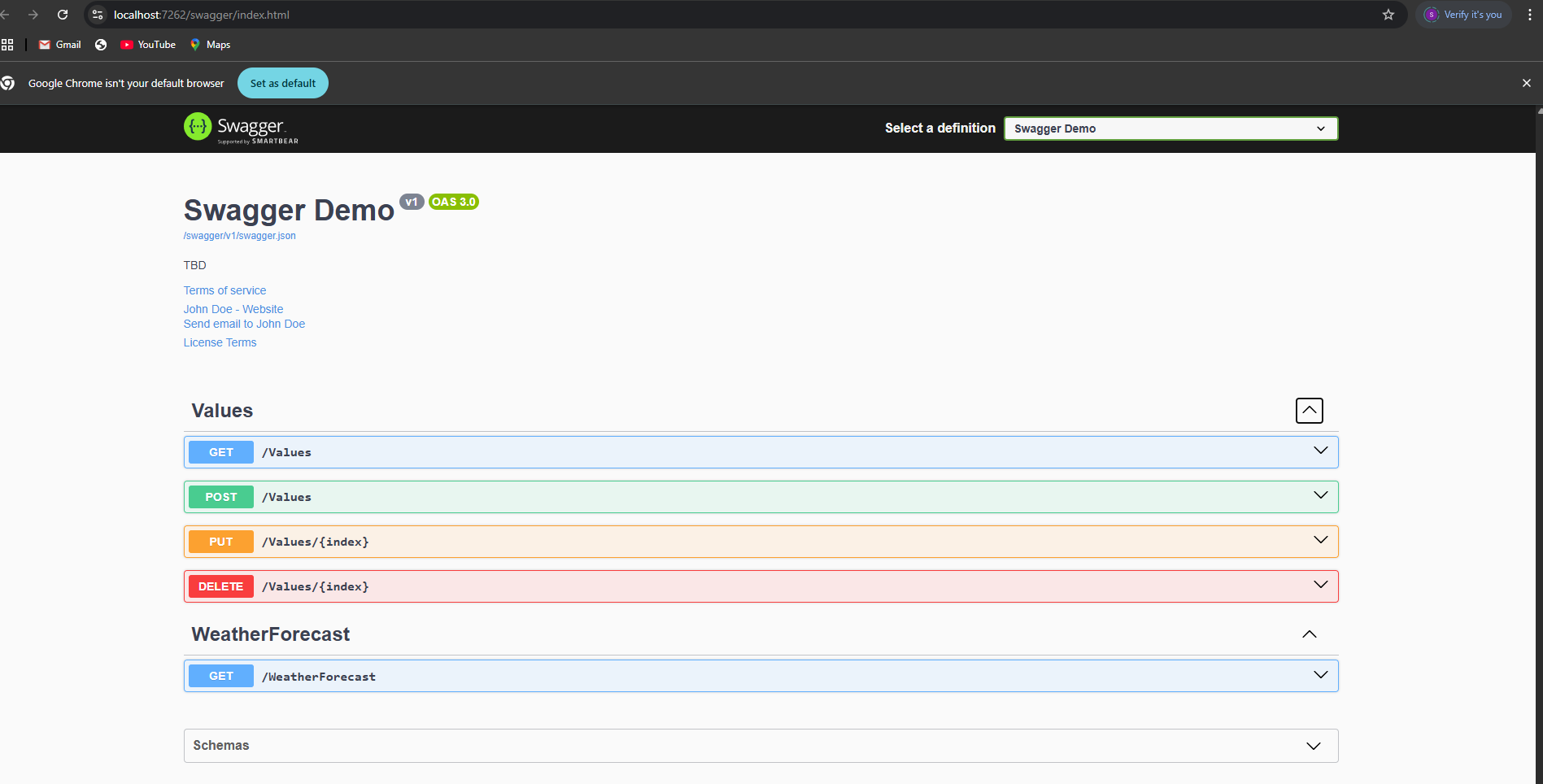
app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

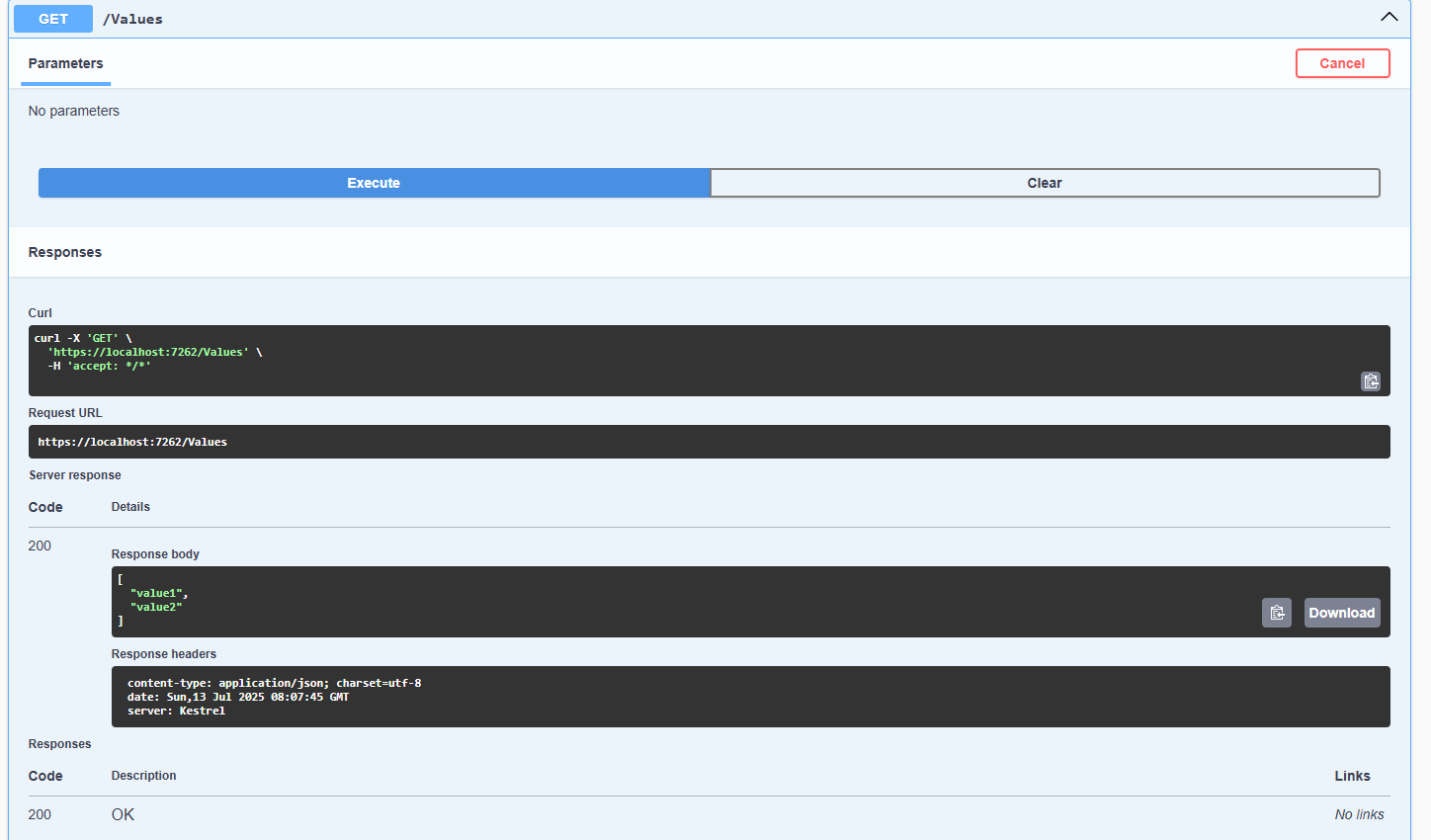
app.Run();

**OUTPUT:**

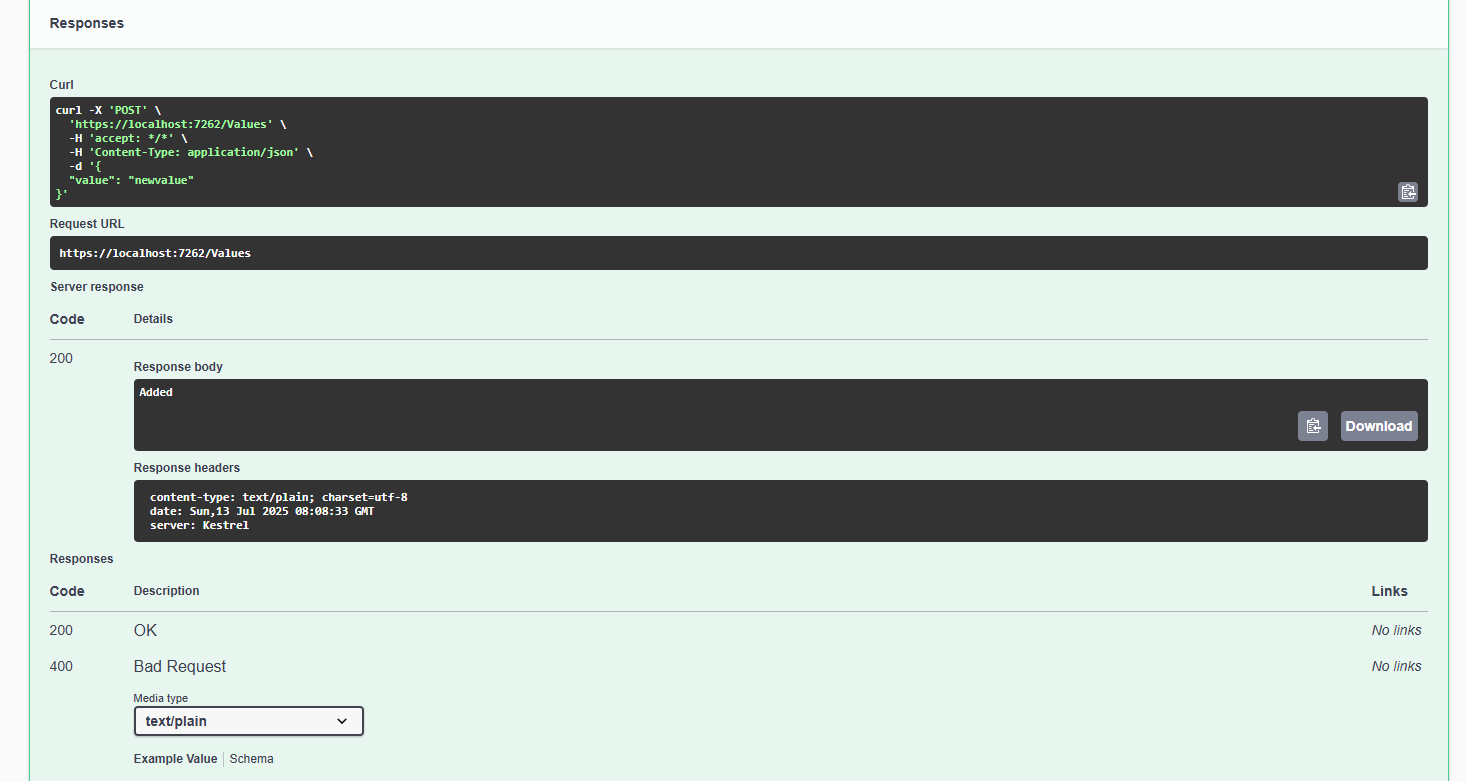
****

**Working of**  **Action methods :**

**#GET:**

****

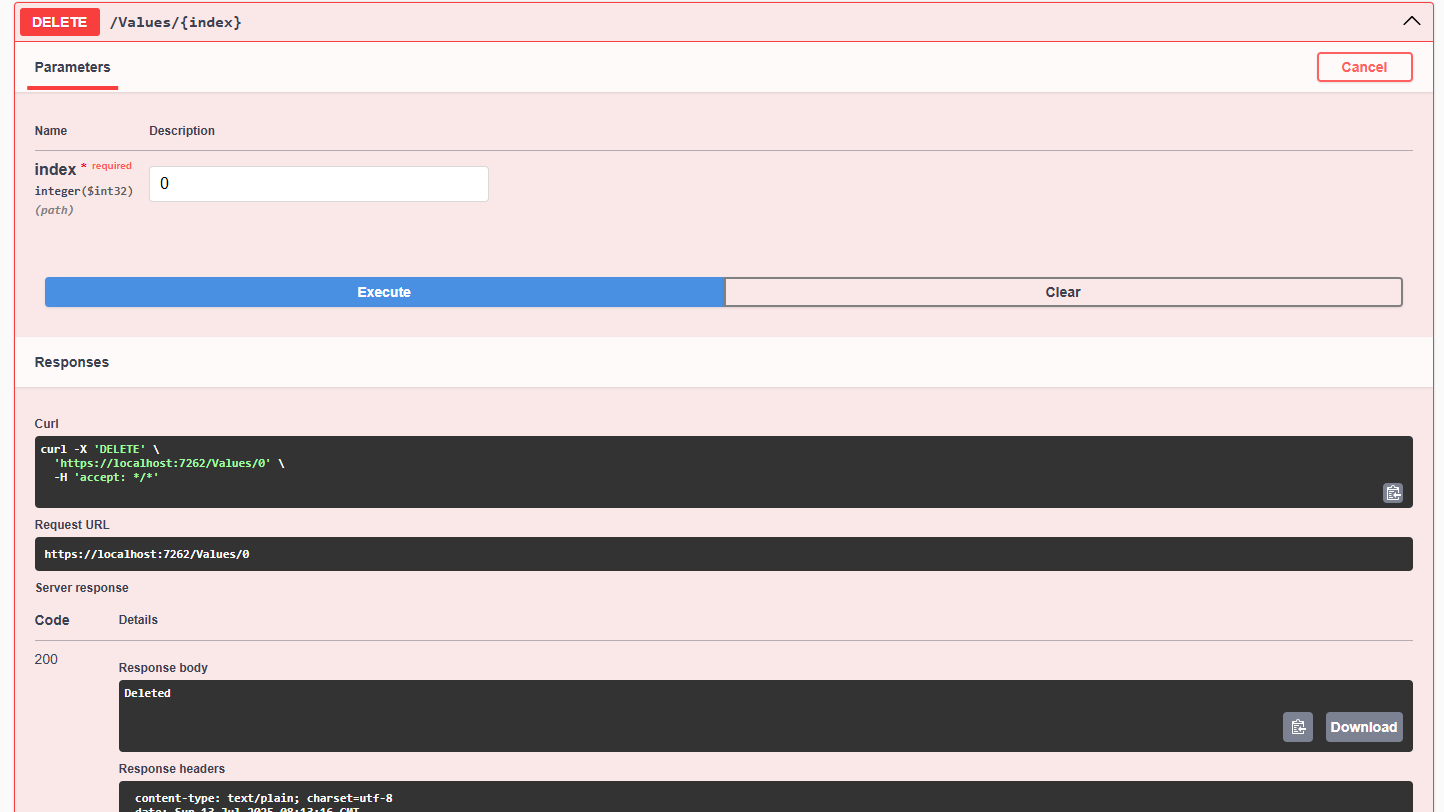
**#POST**

****

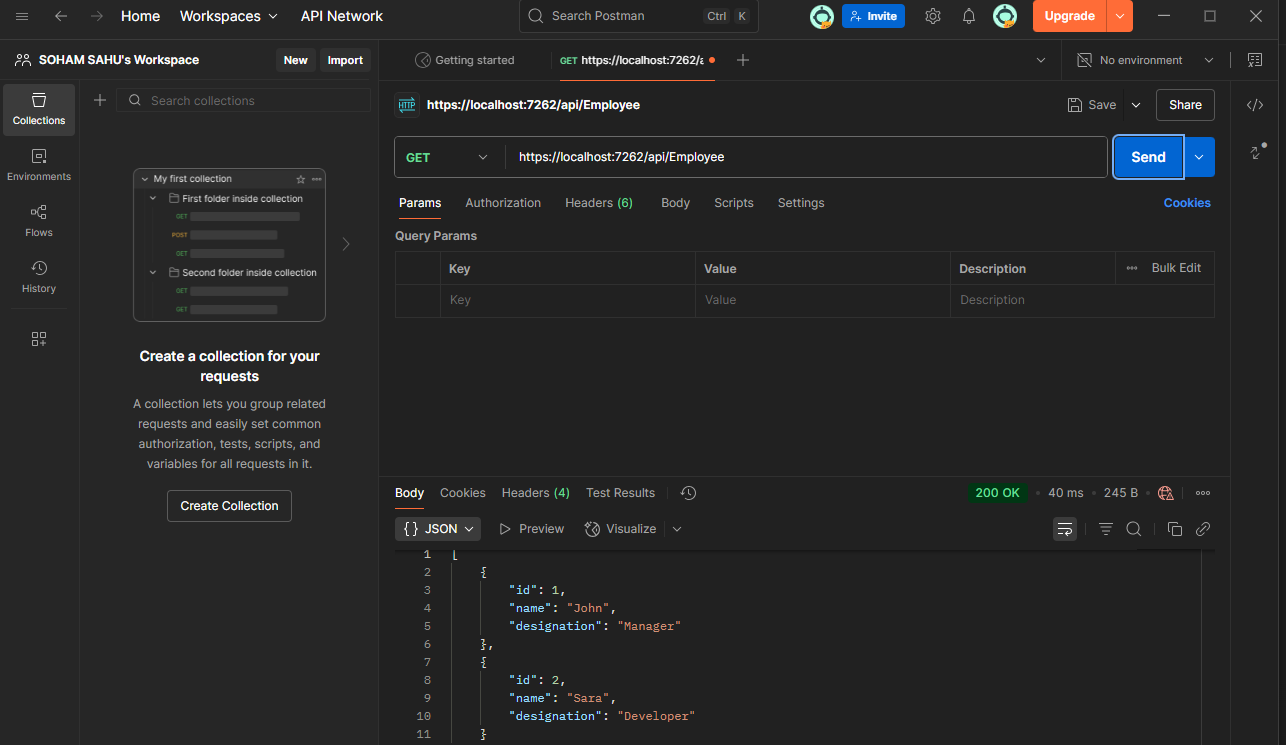
**#PUT:**

****

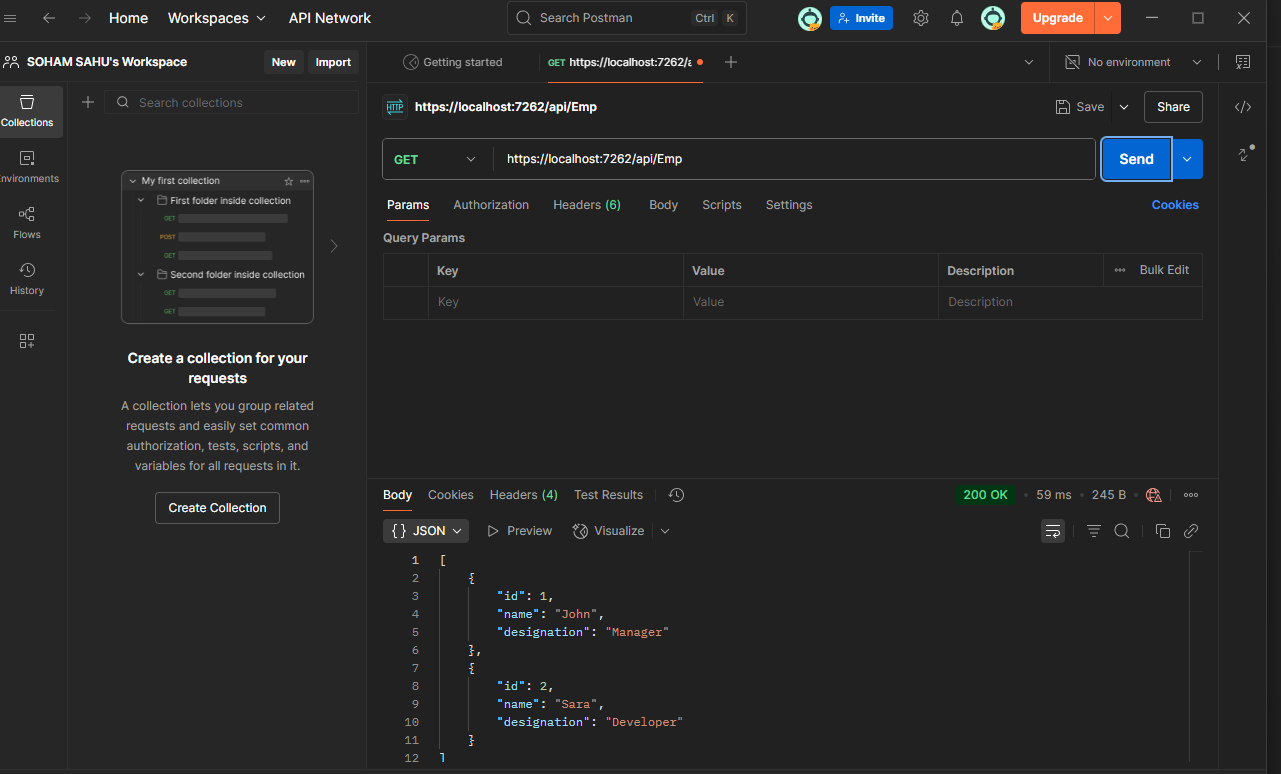
**#DELETE:**

****

**#POSTMAN \_ Employee:**

****

**#POSTMAN\_Emp:**

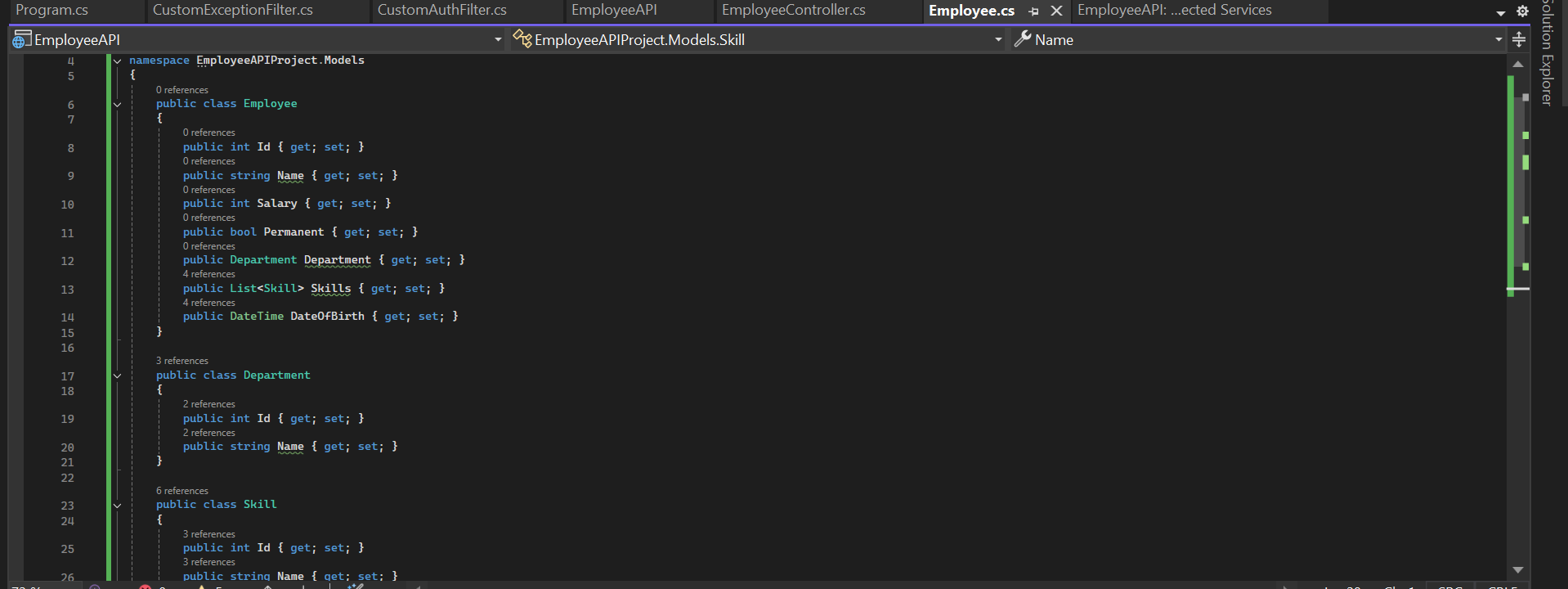
****

1. **WebApi\_Handson:**

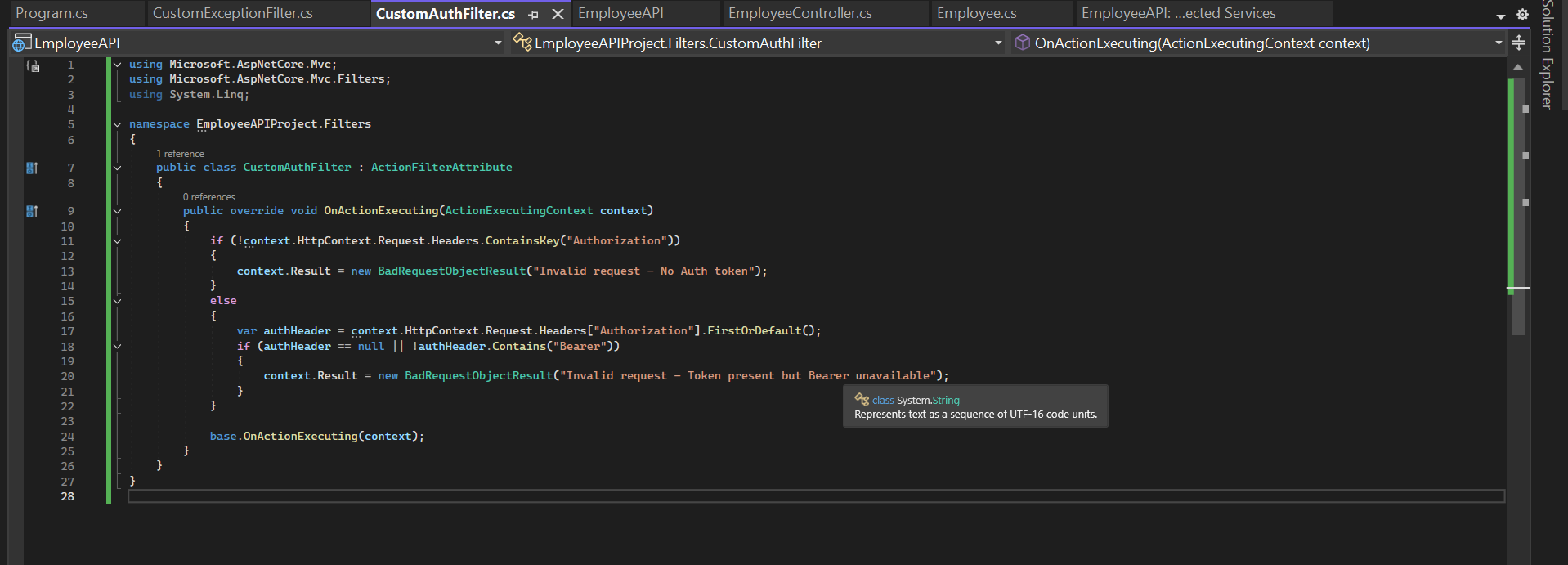
**#CODE:**

**Web Api using custom model class**

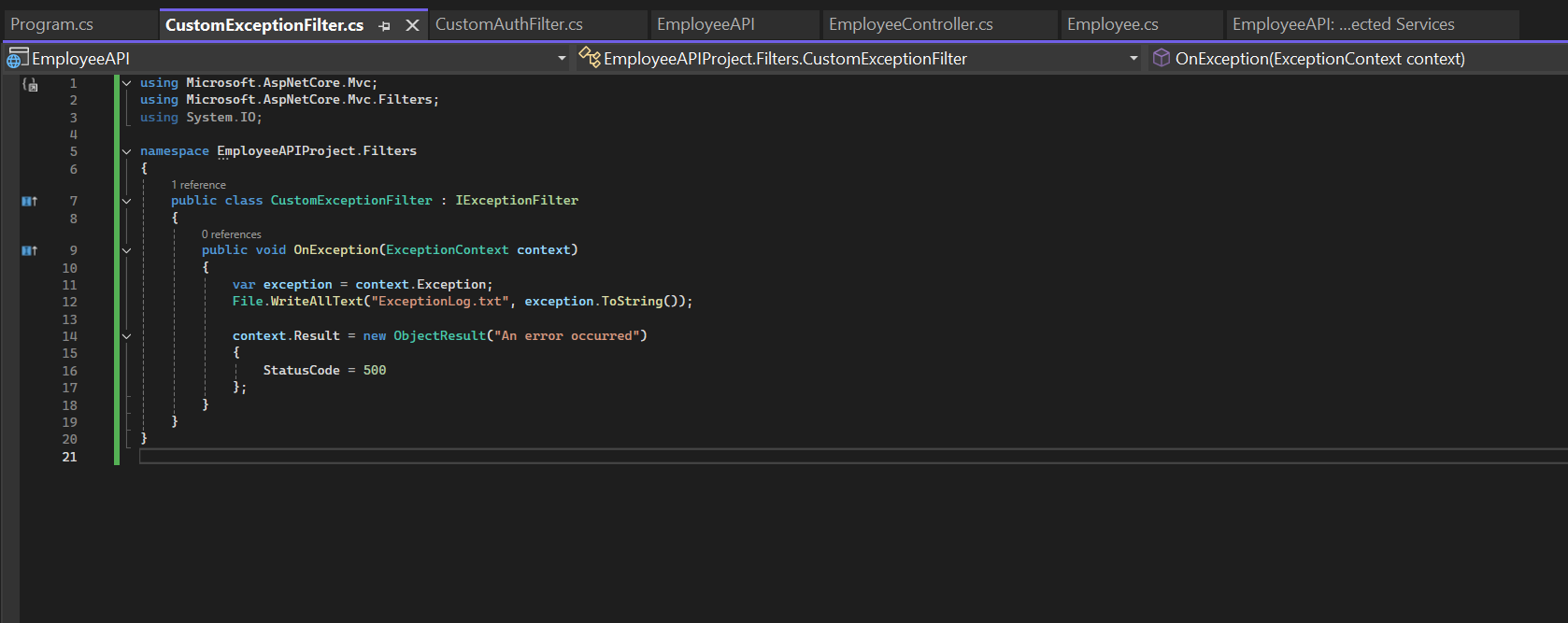
Create a Custom class **‘Employee’** of the below defined structure



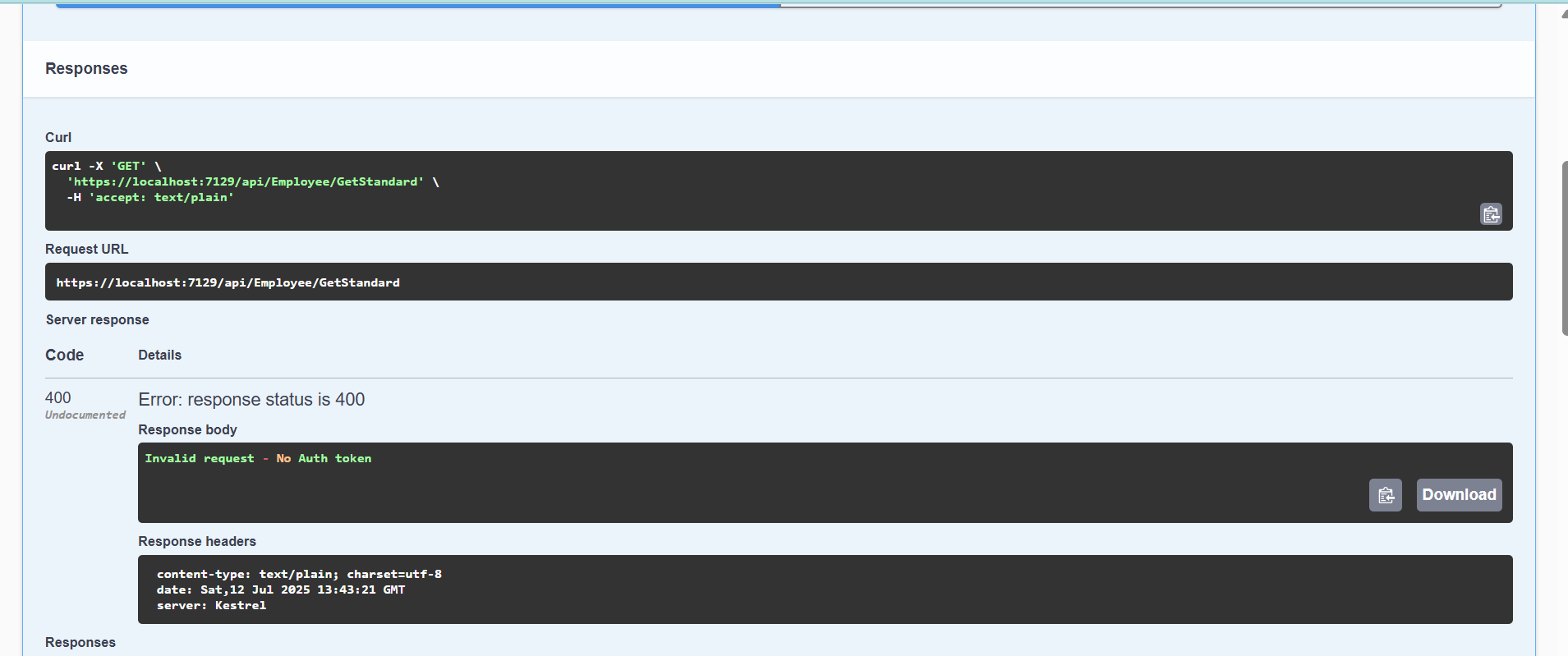
**Create a Custom action filter for Authorization.**

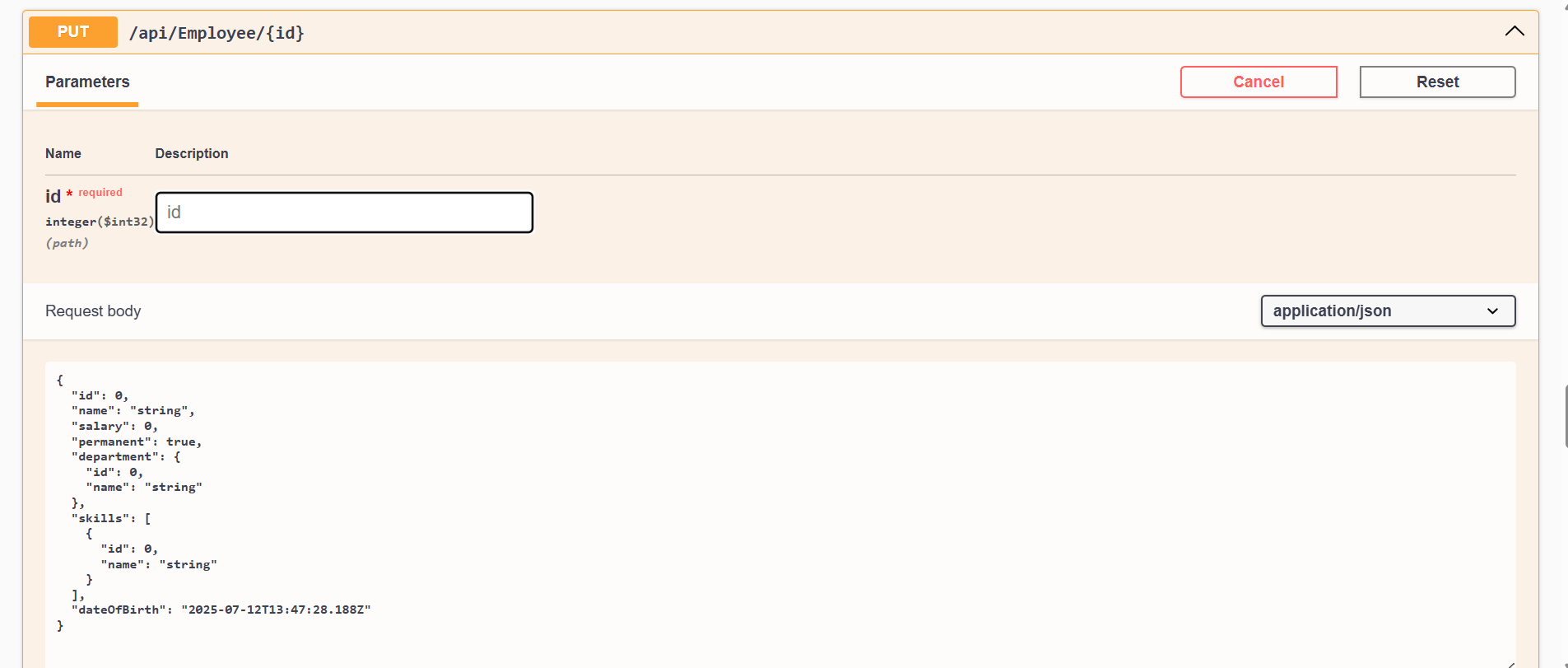


**CustomExceptionfilter:**



**OUTPUT:**





1. **WebApi\_Handson:**

**CODE:**

**#EmployeeController.cs**

using Microsoft.AspNetCore.Mvc;

using System.Collections.Generic;

using System.Linq;

using WebApplication1.Models;

namespace WebApplication1.Controllers

{

[ApiController]

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

public class EmployeeController : ControllerBase

{

// Hardcoded employee list

private static List<Employee> employees = new()

{

new Employee

{

Id = 1,

Name = "John Doe",

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

Skills = new List<Skill>

{

new Skill { Id = 1, Name = "Communication" }

}

},

new Employee

{

Id = 2,

Name = "Jane Smith",

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

Skills = new List<Skill>

{

new Skill { Id = 2, Name = "C#" },

new Skill { Id = 3, Name = "SQL" }

}

}

};

// PUT: api/employee/{id}

[HttpPut("{id}")]

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

{

if (id <= 0)

{

return BadRequest("Invalid employee id");

}

var existingEmployee = employees.FirstOrDefault(e => e.Id == id);

if (existingEmployee == null)

{

return BadRequest("Invalid employee id");

}

// Update existing employee

existingEmployee.Name = updatedEmployee.Name;

existingEmployee.Department = updatedEmployee.Department;

existingEmployee.Skills = updatedEmployee.Skills;

return Ok(existingEmployee);

}

}

}

**#Skill.cs**

namespace WebApplication1.Models

{

public class Skill

{

public int Id { get; set; }

public string Name { get; set; }

}

}

**#Department.cs**

namespace WebApplication1.Models

{

public class Department

{

public int Id { get; set; }

public string DeptName { get; set; }

}

}

**#Employee.cs**

using System.Collections.Generic;

namespace WebApplication1.Models

{

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public Department Department { get; set; }

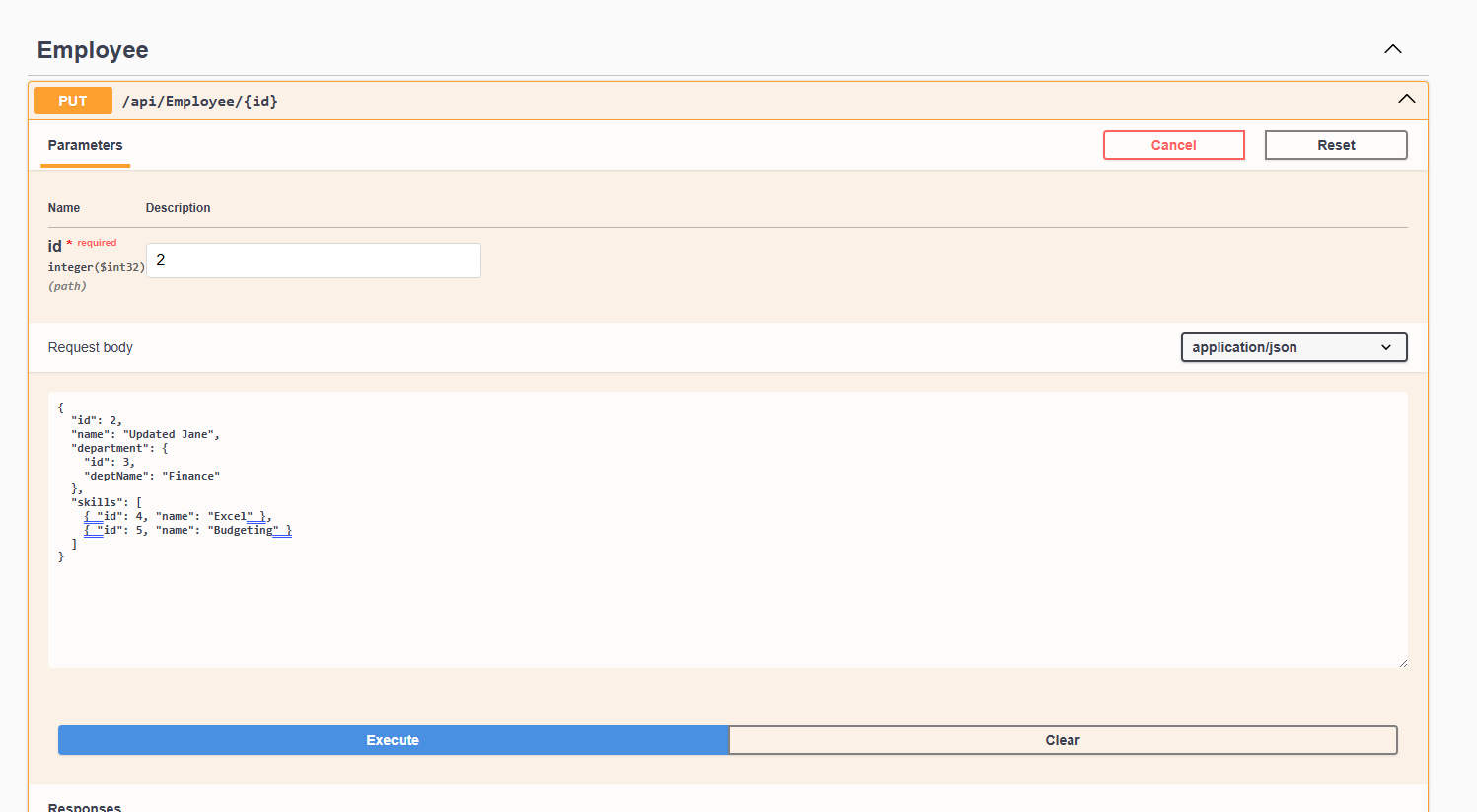
public List<Skill> Skills { get; set; }

}

}

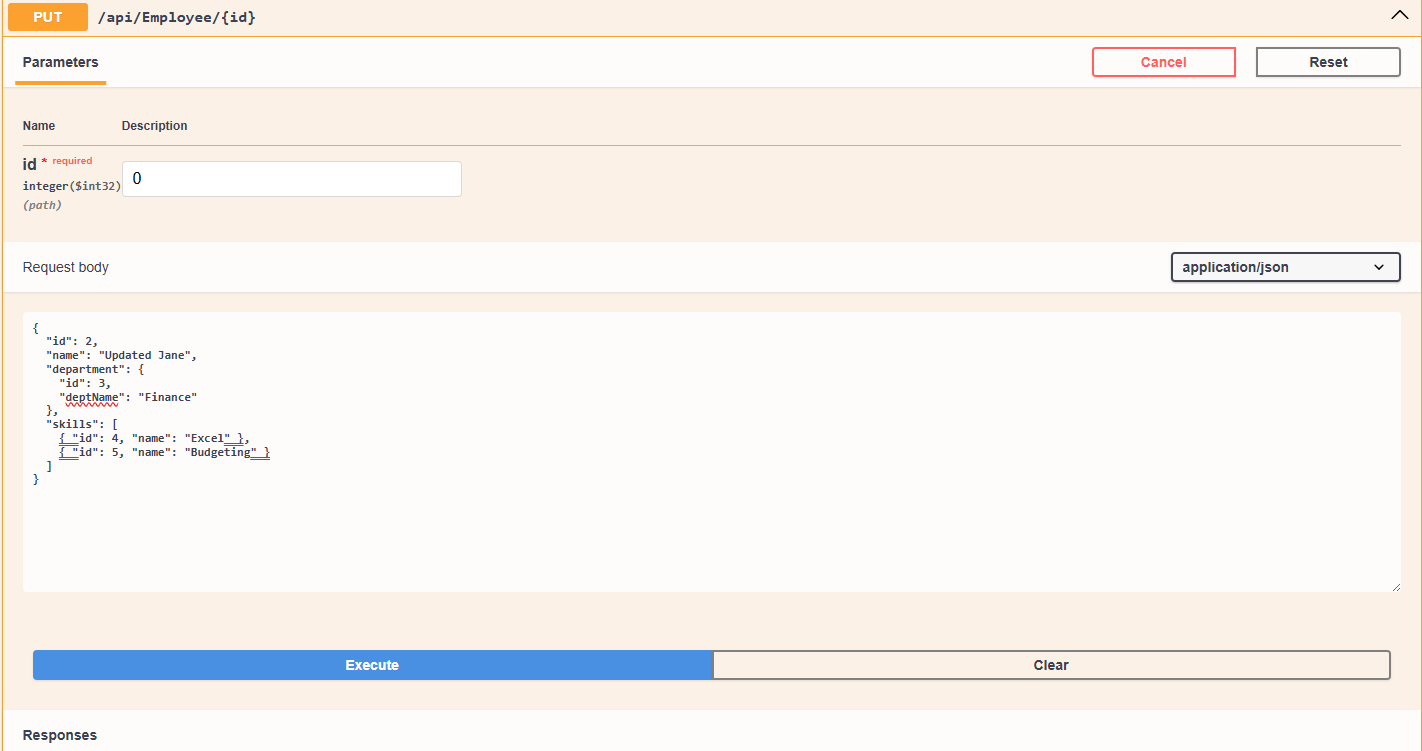
**#OUTPUT**

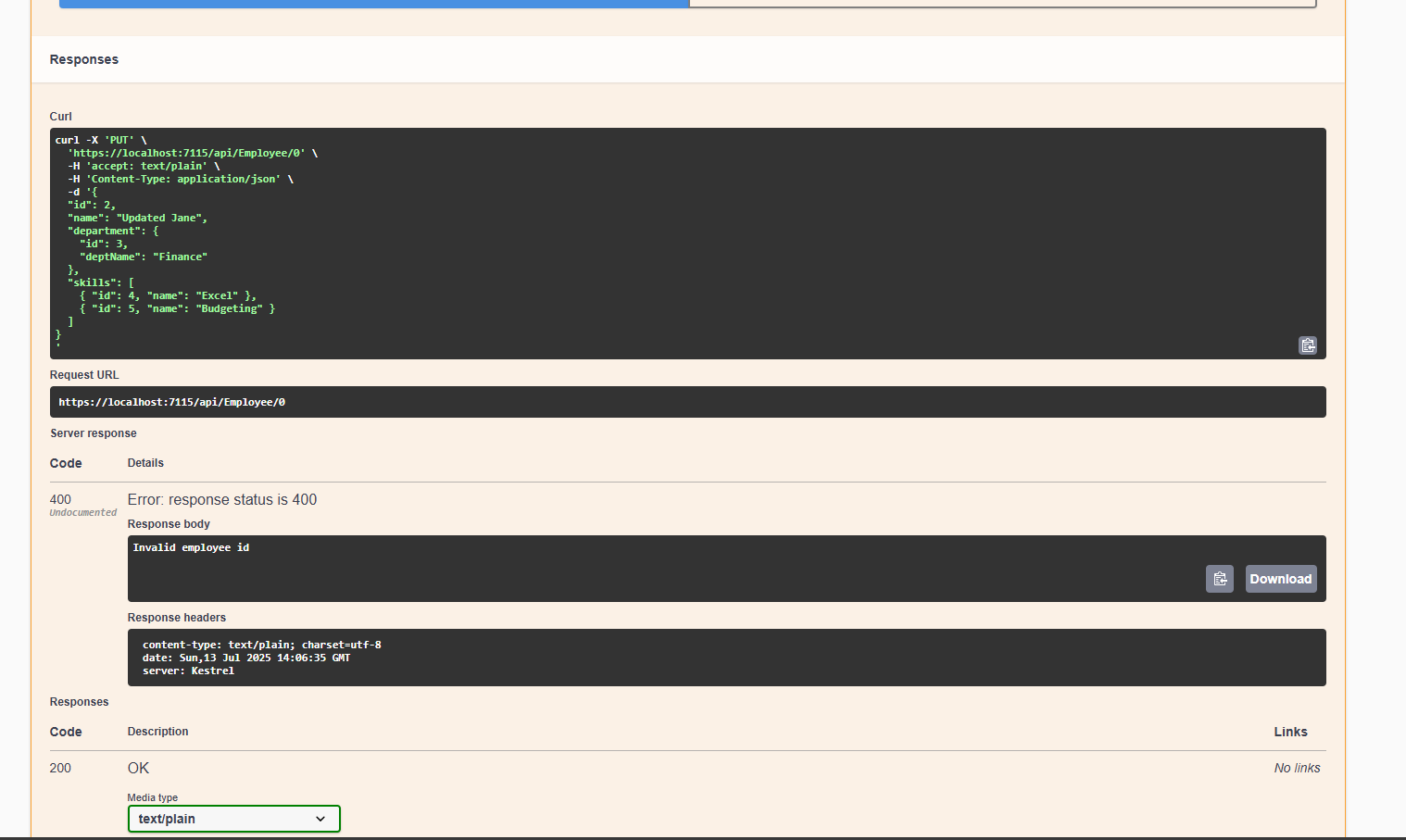
**#Valid Output: ID>0**

****

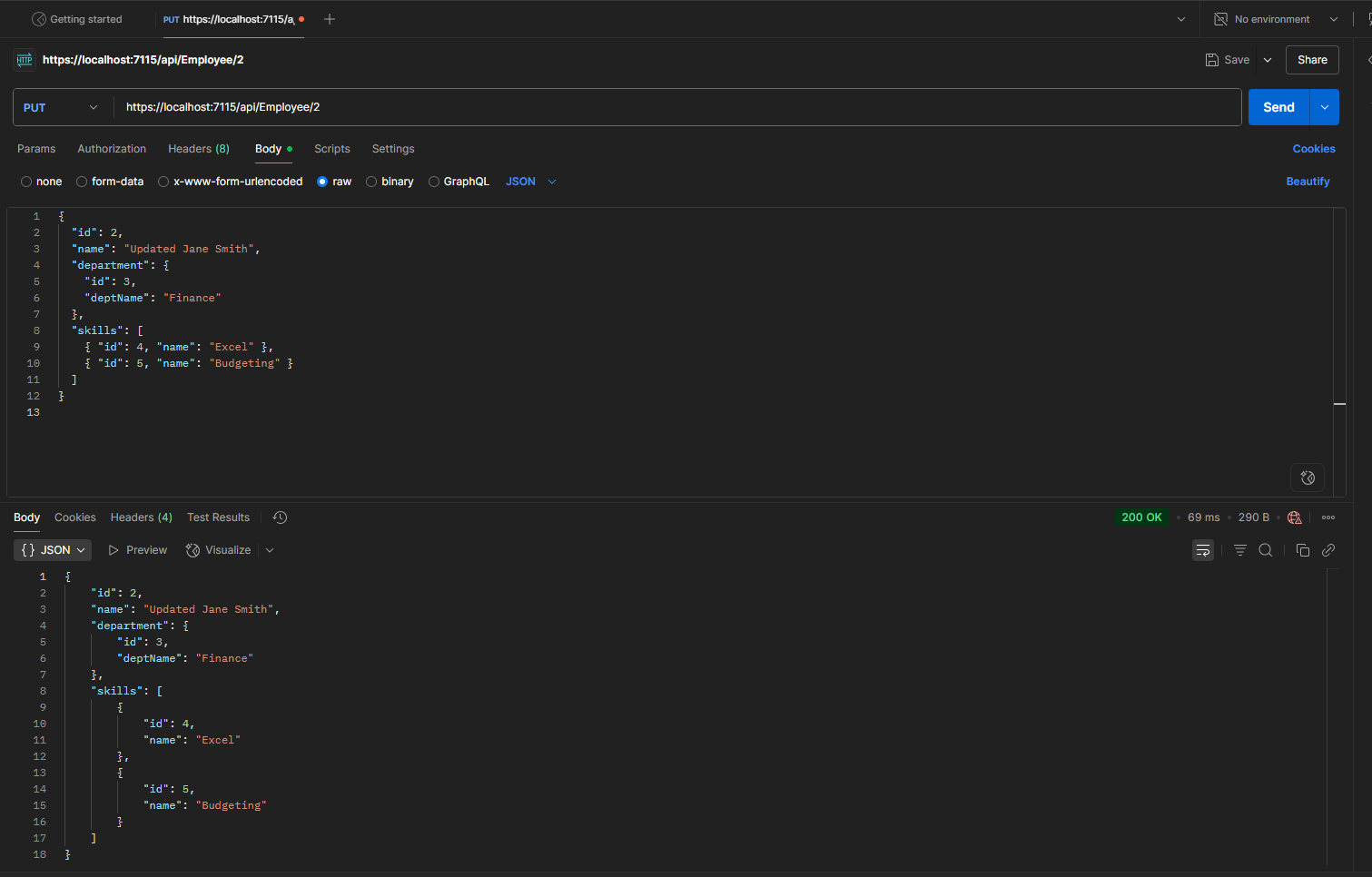
****

**#Invalid Output: ID<=0**

****

****

**#POSTMAN OUTPUT:**

****

1. **WebApi\_Handson:**

**CODE:**

**#EmployeeController.cs**

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

using System.Collections.Generic;

using WebAPI2.Models;

namespace WebAPI2.Controllers

{

[ApiController]

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

[Authorize(Roles = "Admin,POC")]

public class EmployeeController : ControllerBase

{

private static List<Employee> employees = new()

{

new Employee { Id = 1, Name = "John", Department = new Department { Id = 1, DeptName = "HR" }, Skills = new List<Skill> { new Skill { Id = 1, Name = "Excel" } } },

new Employee { Id = 2, Name = "Jane", Department = new Department { Id = 2, DeptName = "IT" }, Skills = new List<Skill> { new Skill { Id = 2, Name = "C#" } } }

};

[HttpGet]

public IActionResult GetEmployees()

{

return Ok(employees);

}

}

}

**#AuthController.cs**

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

using Microsoft.IdentityModel.Tokens;

using System.IdentityModel.Tokens.Jwt;

using System.Security.Claims;

using System.Text;

namespace WebAPI2.Controllers

{

[ApiController]

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

[AllowAnonymous]

public class AuthController : ControllerBase

{

[HttpGet("token")]

public IActionResult GetToken()

{

var token = GenerateJSONWebToken(101, "Admin"); change the role

return Ok(token);

}

private string GenerateJSONWebToken(int userId, string userRole)

{

var securityKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes("mysuperdupersecretkey1234567890!!"));

var credentials = new SigningCredentials(securityKey, SecurityAlgorithms.HmacSha256);

var claims = new List<Claim>

{

new Claim(ClaimTypes.Role, userRole),

new Claim("UserId", userId.ToString())

};

var token = new JwtSecurityToken(

issuer: "mySystem",

audience: "myUsers",

claims: claims,

expires: DateTime.Now.AddMinutes(2),

signingCredentials: credentials);

return new JwtSecurityTokenHandler().WriteToken(token);

}

}

}

**#Skill.cs**

namespace WebApplication1.Models

{

public class Skill

{

public int Id { get; set; }

public string Name { get; set; }

}

}

**#Department.cs**

namespace WebApplication1.Models

{

public class Department

{

public int Id { get; set; }

public string DeptName { get; set; }

}

}

**#Employee.cs**

using System.Collections.Generic;

namespace WebApplication1.Models

{

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public Department Department { get; set; }

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

}

}

**#Program.cs**

using Microsoft.AspNetCore.Authentication.JwtBearer;

using Microsoft.IdentityModel.Tokens;

using System.Text;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container

builder.Services.AddControllers();

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

// Enable CORS

builder.Services.AddCors(options =>

{

options.AddPolicy("AllowAll", policy =>

{

policy.AllowAnyOrigin().AllowAnyHeader().AllowAnyMethod();

});

});

// ?? JWT Authentication Configuration

string securityKey = "mysuperdupersecretkey1234567890!!"; // ? 32+ characters

var symmetricSecurityKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(securityKey));

builder.Services.AddAuthentication(options =>

{

options.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;

options.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;

}).AddJwtBearer(options =>

{

options.TokenValidationParameters = new TokenValidationParameters

{

ValidateIssuer = true,

ValidateAudience = true,

ValidateLifetime = true,

ValidateIssuerSigningKey = true,

ValidIssuer = "mySystem",

ValidAudience = "myUsers",

IssuerSigningKey = symmetricSecurityKey

};

});

var app = builder.Build();

app.UseSwagger();

app.UseSwaggerUI();

app.UseCors("AllowAll");

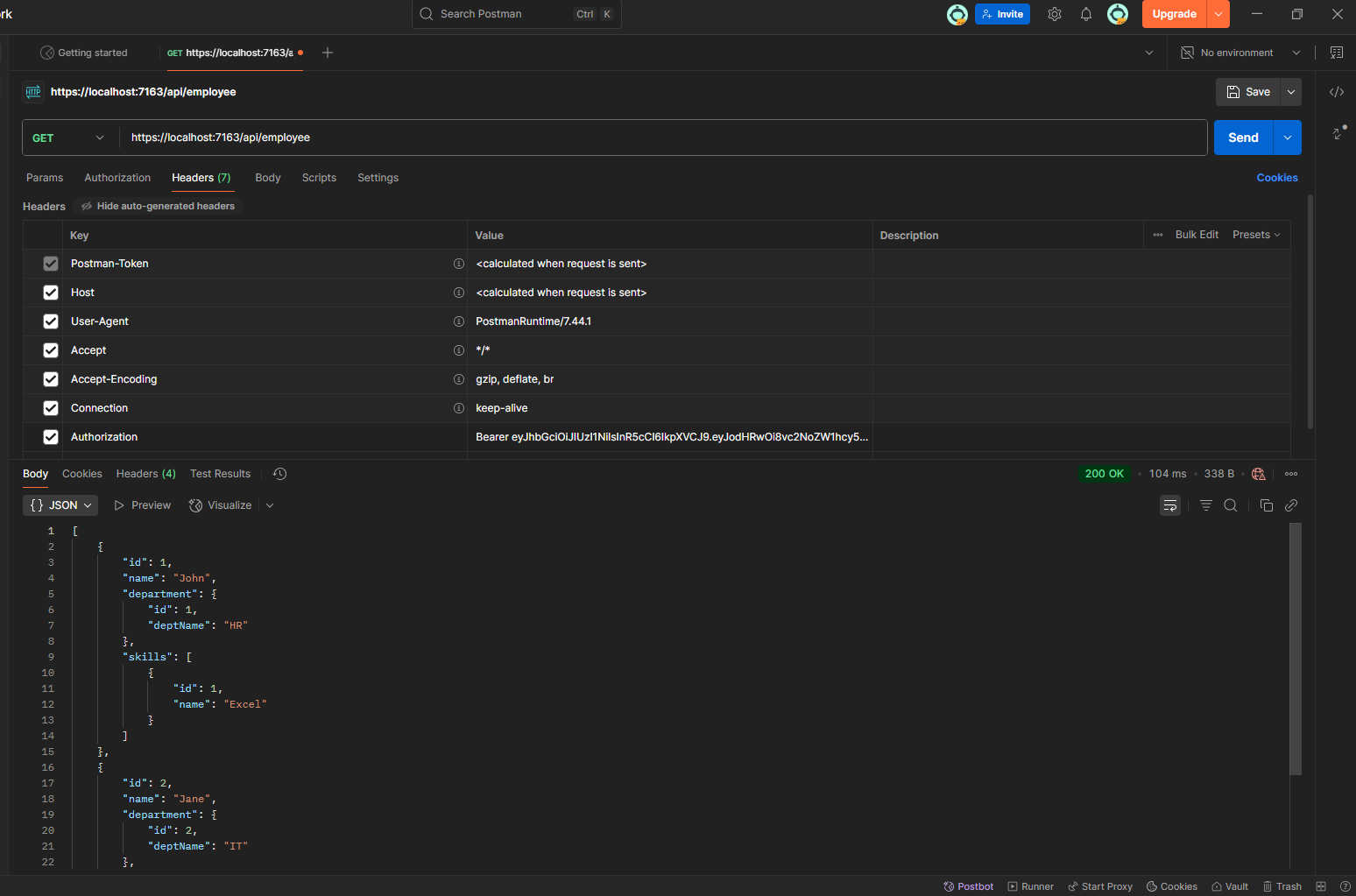
app.UseAuthentication();

app.UseAuthorization();

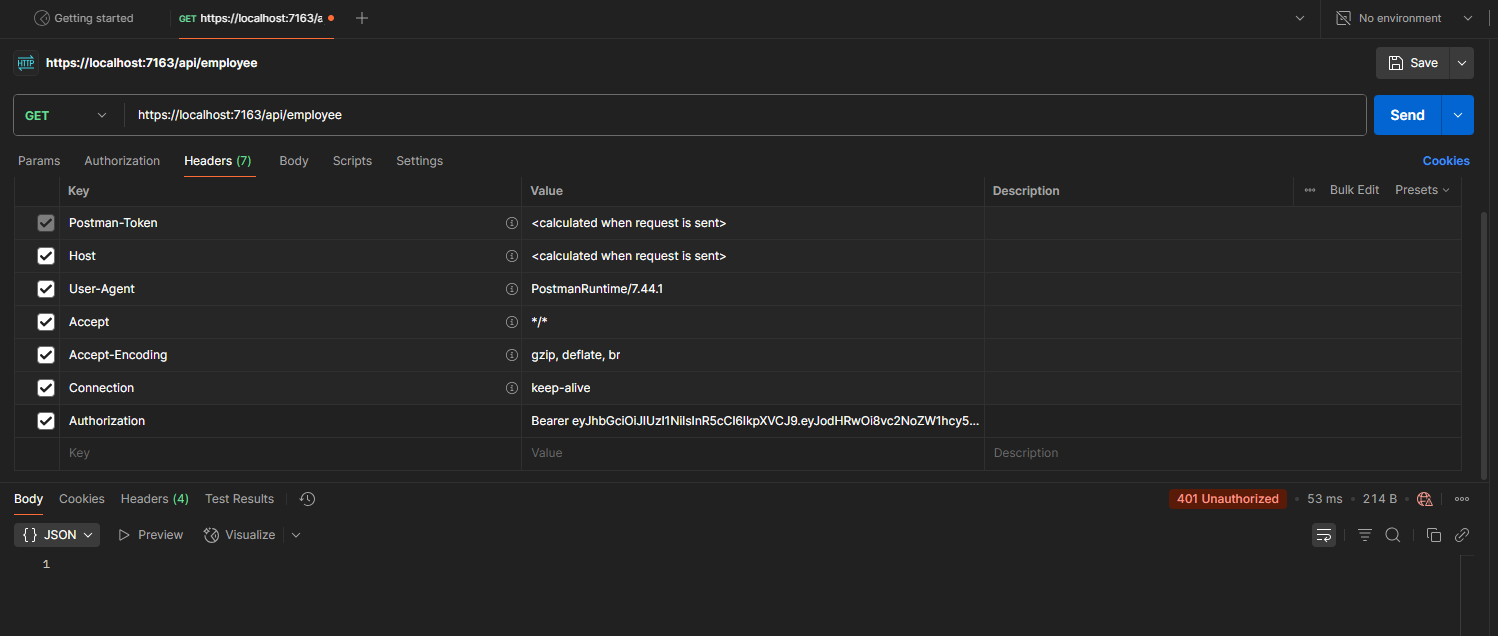
app.MapControllers();

app.Run();

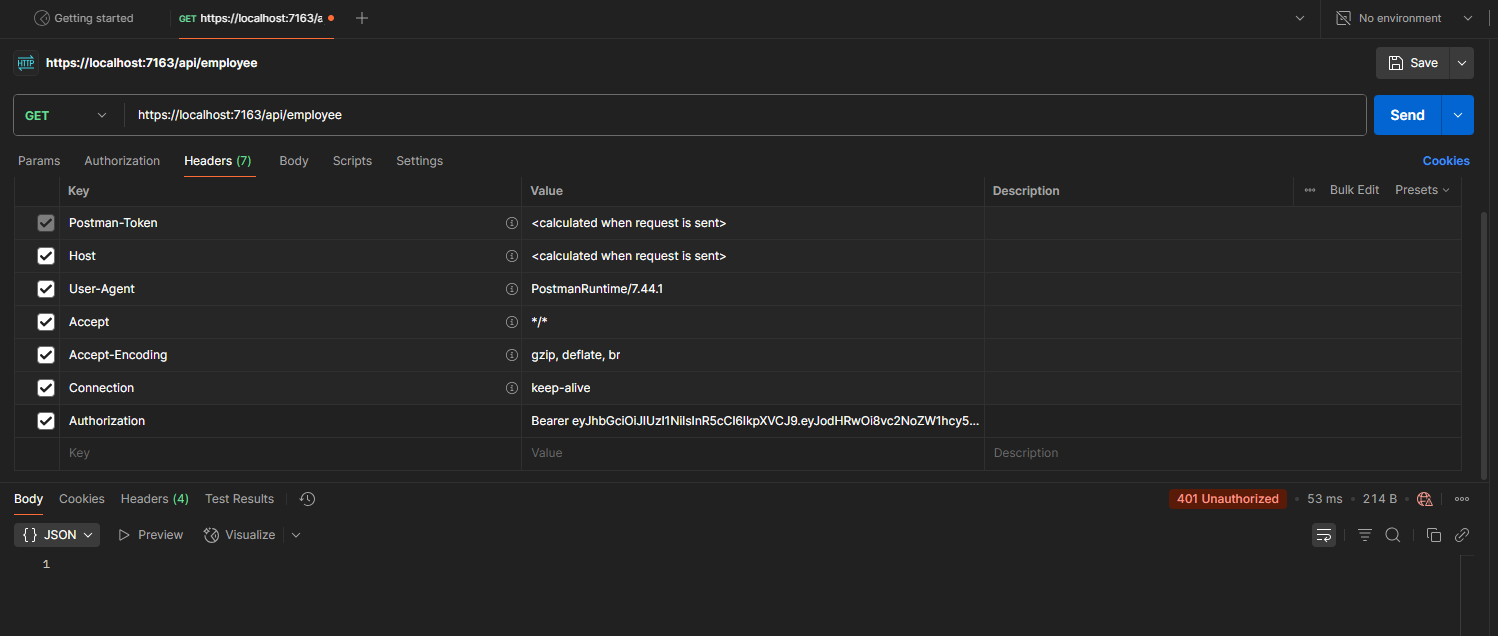
**#OUTPUT:**

****

**#Check for JWT expiration :**

****

**#Only POC:**

****