**ASP .NET CORE 8.0 WEB API**

1. **First Web Api using .Net Core**

Create a .Net core web application with API template. Use the option to create controller with Read Write permissions. Notice the ValuesController creation with Action methods corresponding to the Action verbs.

**ValueController.cs**

using Microsoft.AspNetCore.Mvc;

namespace MyWebApiApp.Controllers

{

    [ApiController]

    [Route("[controller]")]

    public class ValuesController : ControllerBase

    {

        [HttpGet]

        public IActionResult Get()

        {

*return* Ok(new[] { "value1", "value2" });

        }

    }

}

**Program.cs**

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers();

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

var app = builder.Build();

app.UseSwagger();

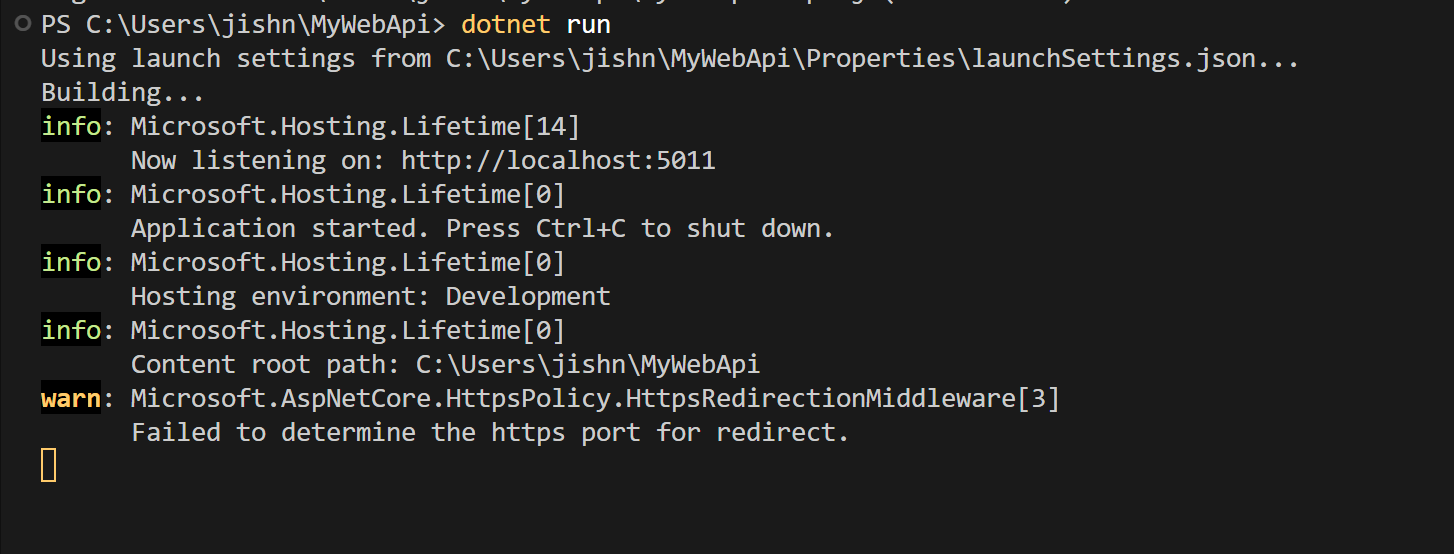
app.UseSwaggerUI();

app.UseHttpsRedirection();

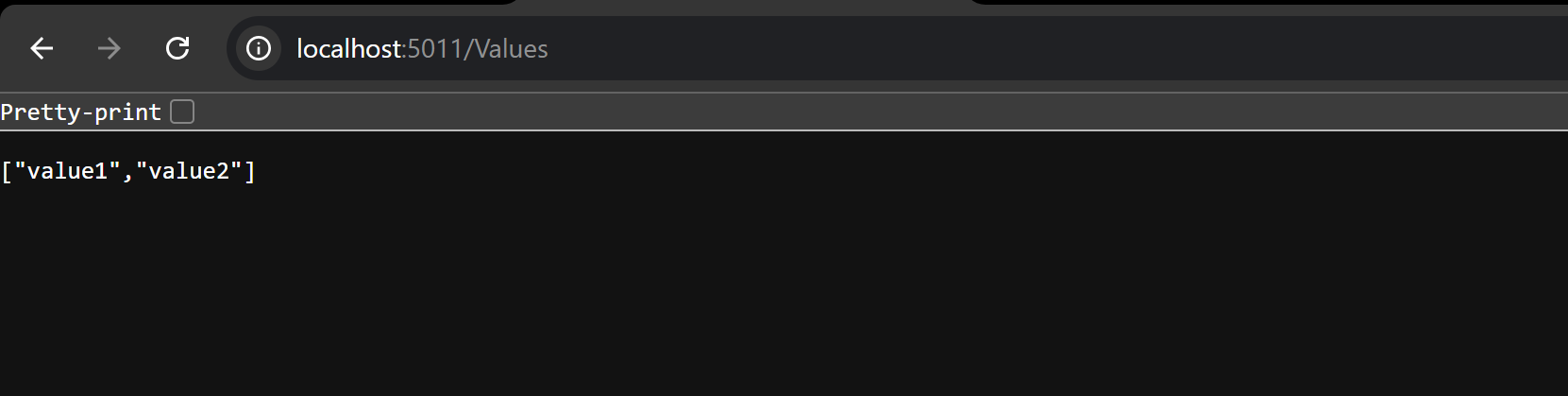
app.UseAuthorization();

app.MapControllers();

app.Run();



**Output:**

****

1. **Web Api using Swagger**

**EmployeeController.cs**

using Microsoft.AspNetCore.Mvc;

namespace MyWebApiApp.Controllers

{

    [ApiController]

    [Route("Emp")]

    public class EmployeeController : ControllerBase

    {

        [HttpGet]

        public IActionResult Get()

        {

            var employees = new[] { "John", "Jane", "Bob" };

*return* Ok(employees);

        }

        [HttpPost]

        public IActionResult Post([FromBody] string employee)

        {

*return* Ok($"Added: {employee}");

        }

    }

}

**Program.cs**

var builder = WebApplication.CreateBuilder(args);

*// Add services*

builder.Services.AddControllers();

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

*// Build app*

var app = builder.Build();

app.UseSwagger();

app.UseSwaggerUI();

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();

builder.Services.AddSwaggerGen(c =>

{

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

    {

        Title = "Swagger Demo",

        Version = "v1",

        Description = "Demo for Web API hands-on",

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

        Contact = new Microsoft.OpenApi.Models.OpenApiContact

        {

            Name = "John Doe",

            Email = "john@xyzmail.com",

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

        },

        License = new Microsoft.OpenApi.Models.OpenApiLicense

        {

            Name = "License Terms",

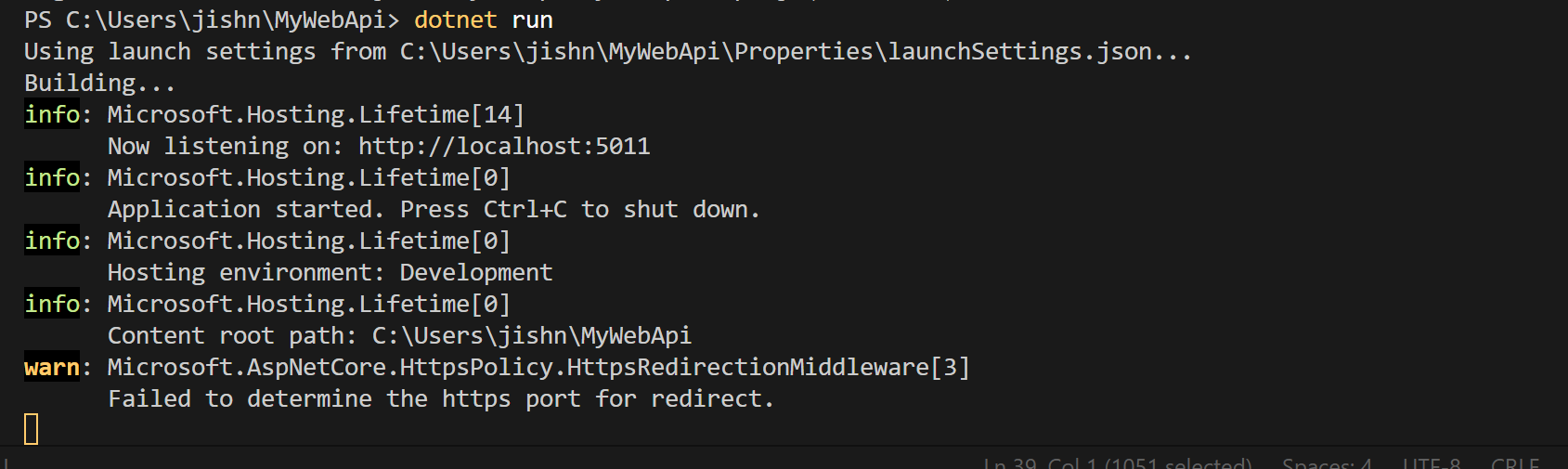
            Url = new Uri("https://www.example.com")

        }

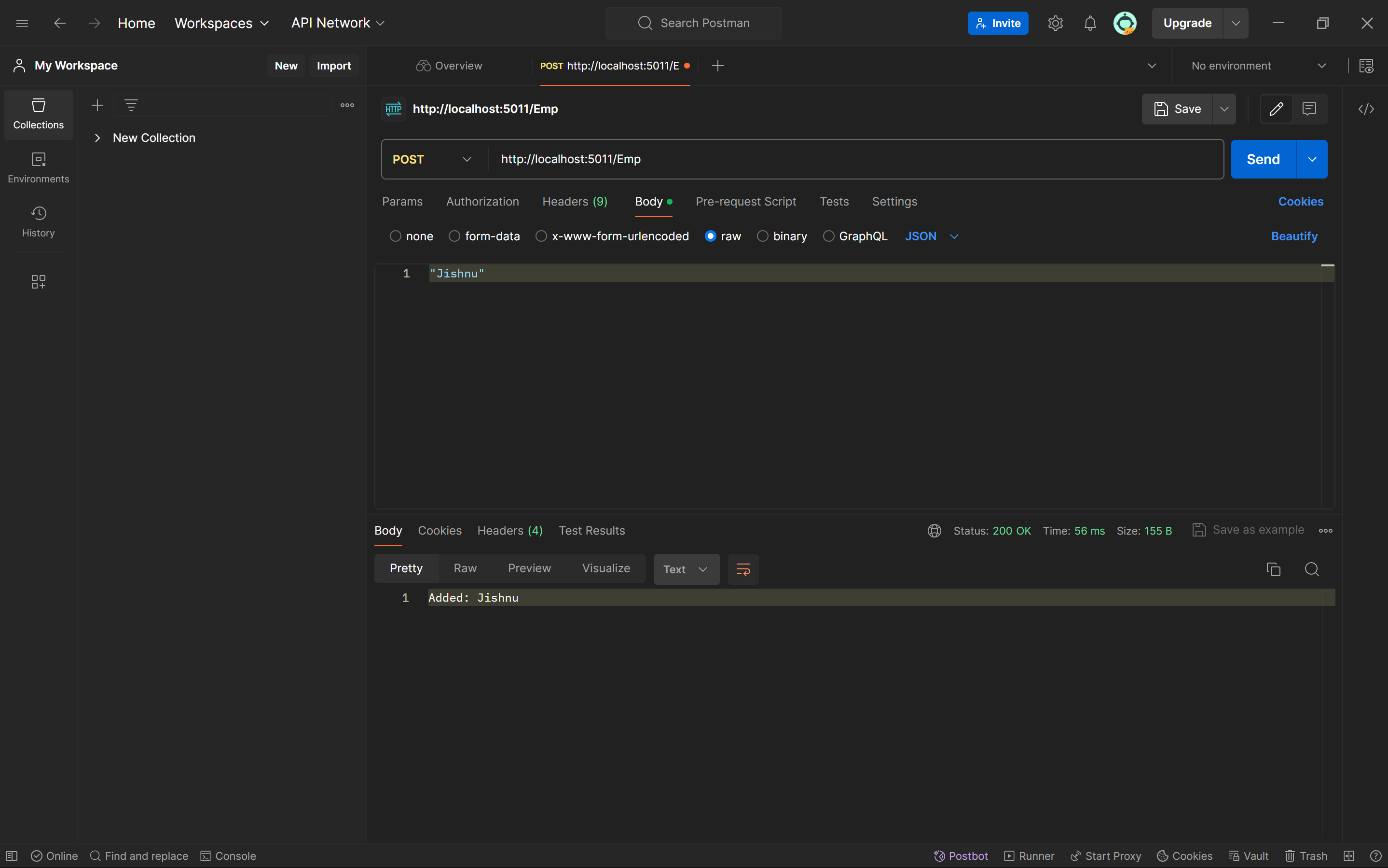
    });

});

**Output**

****

Adding a new Employee



1. **Web Api using Custom Model Class**

**EmployeeController.cs**

using Microsoft.AspNetCore.Mvc;

using MyWebApiApp.Filters;

using MyWebApiApp.Models;

namespace MyWebApiApp.Controllers;

[ApiController]

[Route("[controller]")]

[CustomAuthFilter]

[ServiceFilter(typeof(CustomExceptionFilter))]

public class EmployeeController : ControllerBase

{

    private readonly 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 = "IT" },

                Skills = new List<Skill> { new Skill { Id = 1, Name = "C#" } },

                DateOfBirth = new DateTime(1990, 1, 1)

            }

        };

    }

    [HttpGet("GetStandard")]

    [ProducesResponseType(StatusCodes.Status200OK)]

    [ProducesResponseType(StatusCodes.Status500InternalServerError)]

    public ActionResult<List<Employee>> GetStandard()

    {

*throw* new Exception("Simulated server error!");

*// return Ok(\_employees);*

    }

    [HttpPost("AddEmployee")]

    [ProducesResponseType(StatusCodes.Status200OK)]

    public ActionResult AddEmployee([FromBody] Employee emp)

    {

*return* Ok($"Employee {emp.Name} added successfully.");

    }

}

**CustomAuthFilter.cs**

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Filters;

namespace MyWebApiApp.Filters;

public class CustomAuthFilter : ActionFilterAttribute

{

    public override void OnActionExecuting(ActionExecutingContext context)

    {

        var headers = context.HttpContext.Request.Headers;

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

        {

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

        }

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

        {

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

        }

    }

}

**CustomExceptionFilter.cs**

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Filters;

namespace MyWebApiApp.Filters;

public class CustomExceptionFilter : IExceptionFilter

{

    public void OnException(ExceptionContext context)

    {

        File.AppendAllText("errors.txt", $"{DateTime.Now} - {context.Exception.Message}\n");

        context.Result = new ObjectResult("Something went wrong")

        {

            StatusCode = 500

        };

    }

}

**Employee.cs**

namespace MyWebApiApp.Models;

public class Skill

{

    public int Id { get; set; }

    public string Name { get; set; }

}

public class Department

{

    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; }

}

**Program.cs**

using MyWebApiApp.Filters;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers();

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

builder.Services.AddScoped<CustomExceptionFilter>();

var app = builder.Build();

app.UseSwagger();

app.UseSwaggerUI();

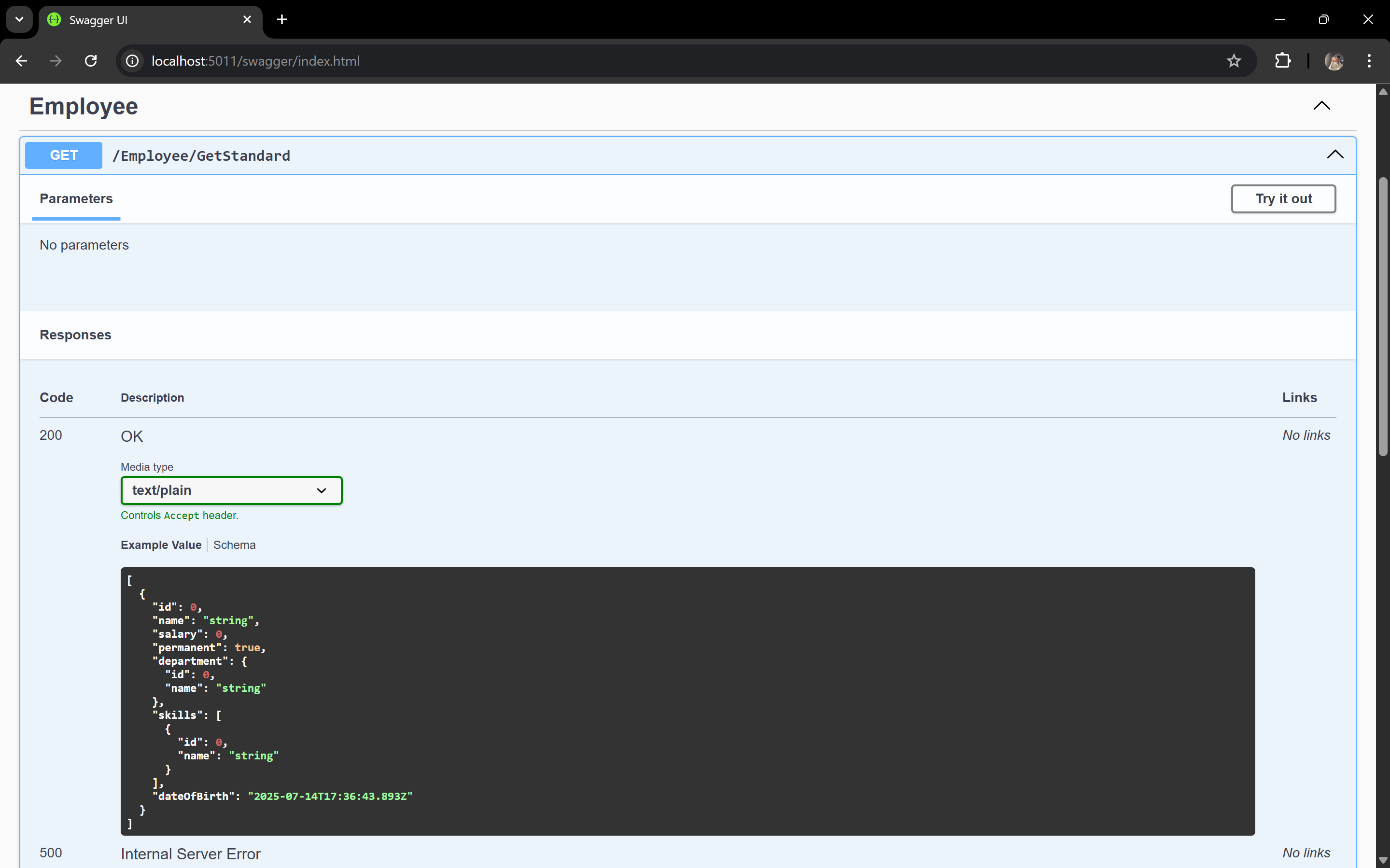
app.UseHttpsRedirection();

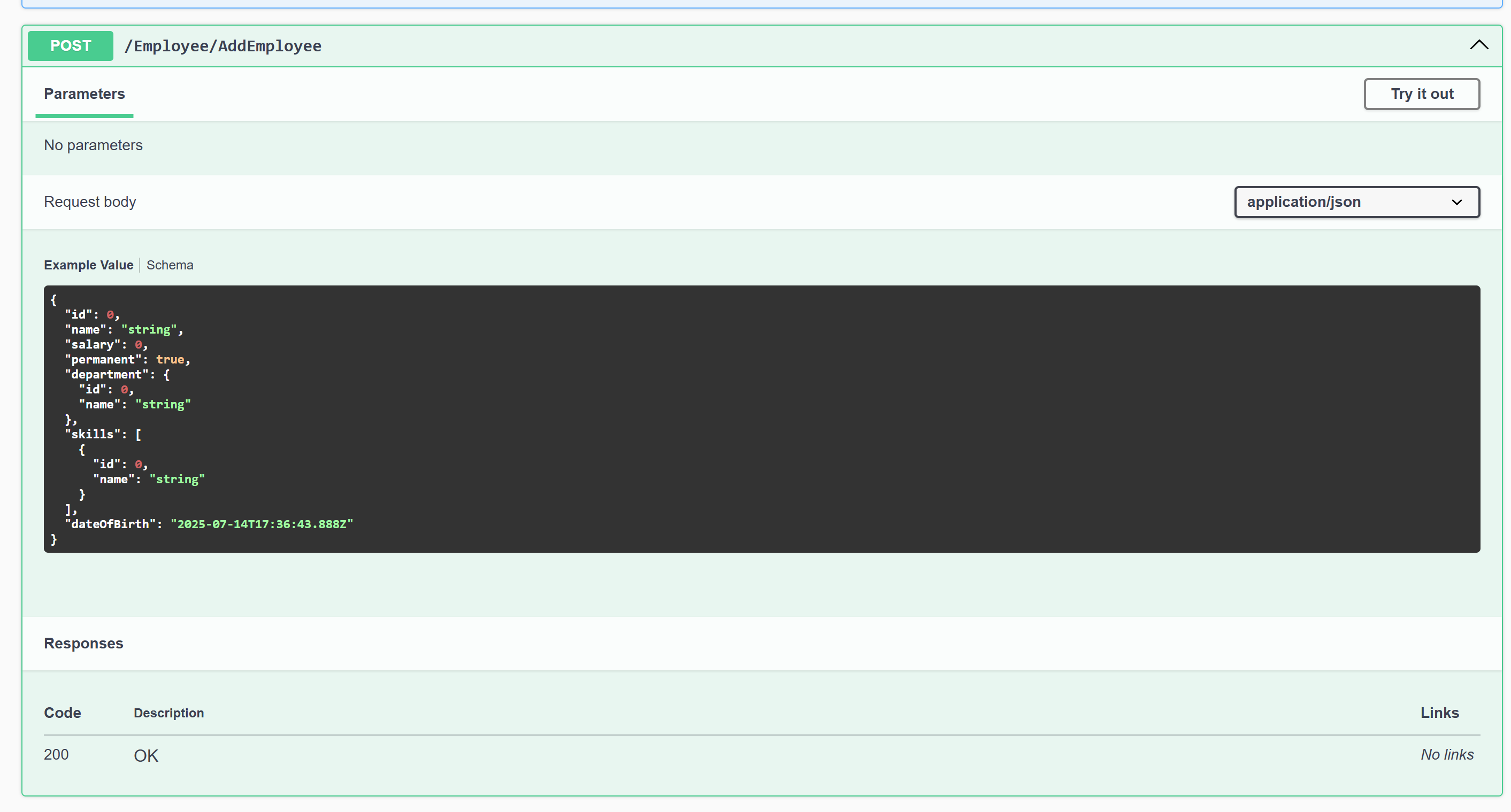
app.UseAuthorization();

app.MapControllers();

app.Run();

**Output**

****

****

1. **Web Api CRUD Operations**

**EmployeeController.cs**

using Microsoft.AspNetCore.Mvc;

using MyWebApiApp.Models;

namespace MyWebApiApp.Controllers;

[ApiController]

[Route("[controller]")]

public class EmployeeController : ControllerBase

{

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

    {

        new Employee

        {

            Id = 1,

            Name = "John",

            Salary = 50000,

            Permanent = true,

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

            Skills = new List<Skill> { new Skill { Id = 1, Name = "C#" } },

            DateOfBirth = new DateTime(1990, 1, 1)

        },

        new Employee

        {

            Id = 2,

            Name = "Jane",

            Salary = 60000,

            Permanent = true,

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

            Skills = new List<Skill> { new Skill { Id = 2, Name = "Communication" } },

            DateOfBirth = new DateTime(1992, 6, 15)

        }

    };

    [HttpPut("UpdateEmployee/{id}")]

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

    {

*if* (id <= 0)

*return* BadRequest("Invalid employee id");

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

*if* (emp == null)

*return* BadRequest("Invalid employee id");

        emp.Name = updatedEmployee.Name;

        emp.Salary = updatedEmployee.Salary;

        emp.Permanent = updatedEmployee.Permanent;

        emp.Department = updatedEmployee.Department;

        emp.Skills = updatedEmployee.Skills;

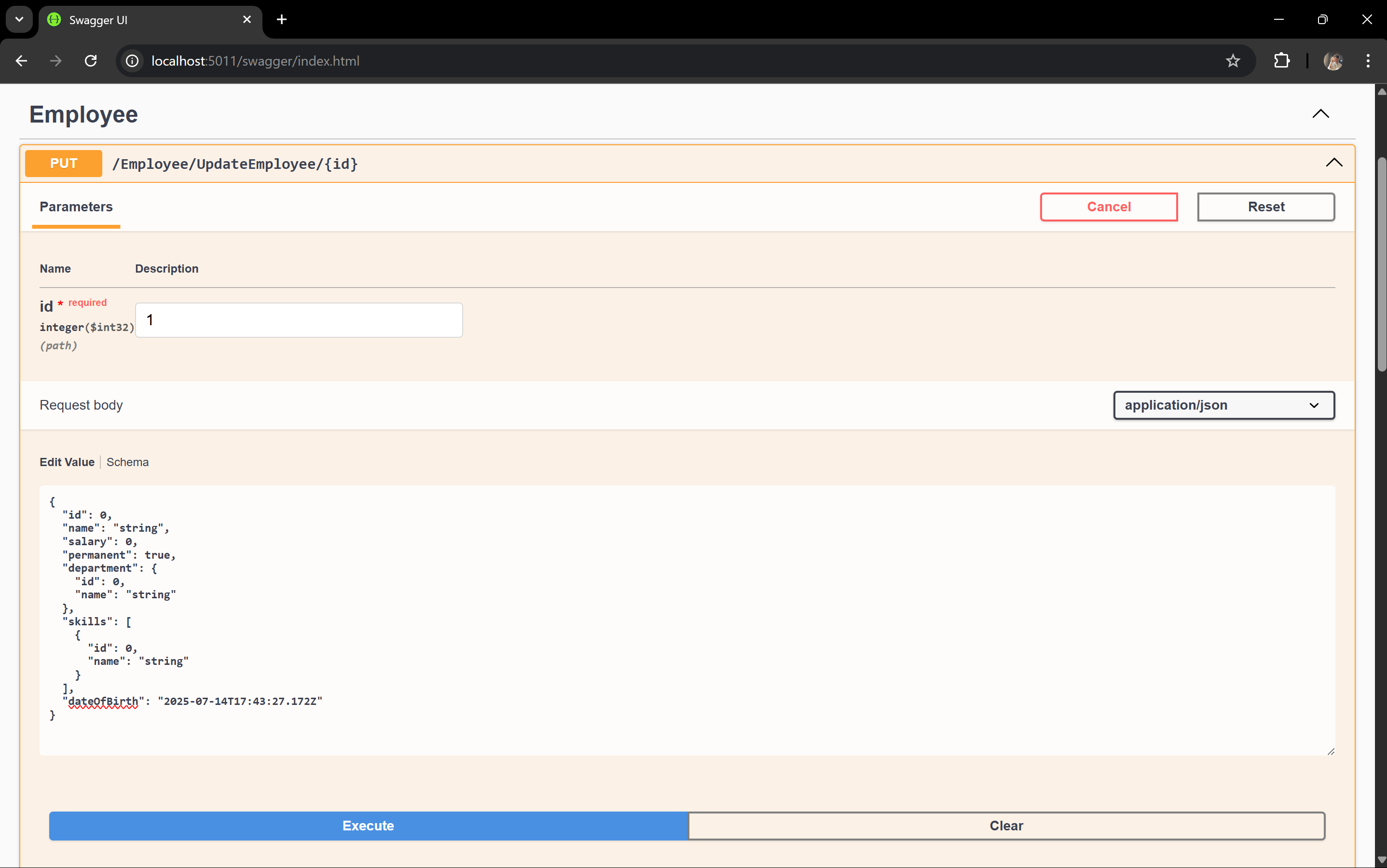
        emp.DateOfBirth = updatedEmployee.DateOfBirth;

*return* Ok(emp);

    }

}

**Output**

****

