**1.WebApi**



**2.WebApi**

**Web Api using .Net core with Swagger**

**Program.cs**

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers();

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://www.example.com"),

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")

}

});

});

var app = builder.Build();

if (app.Environment.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

app.UseSwagger();

app.UseSwaggerUI(c =>

{

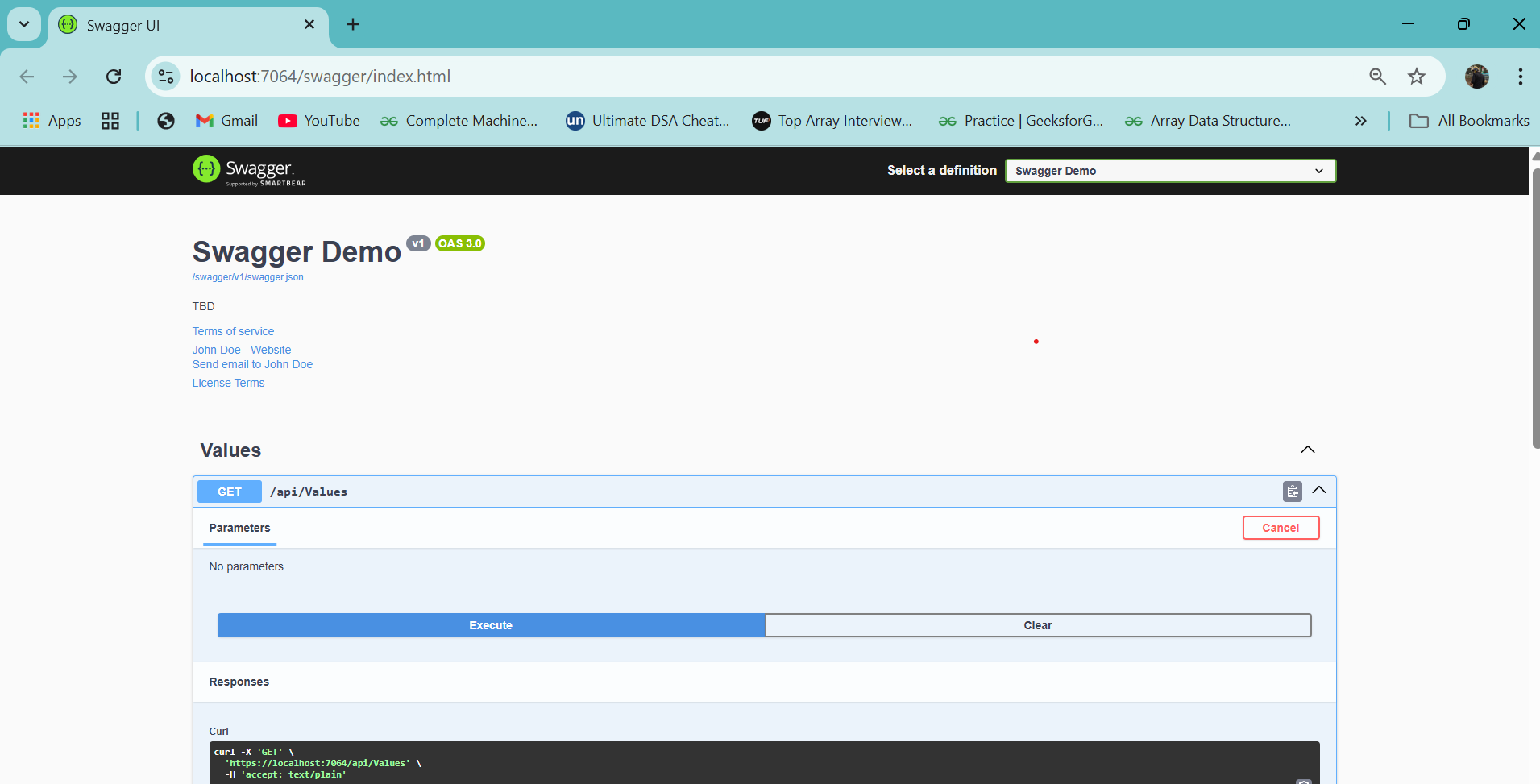
c.SwaggerEndpoint("/swagger/v1/swagger.json", "Swagger Demo");

});

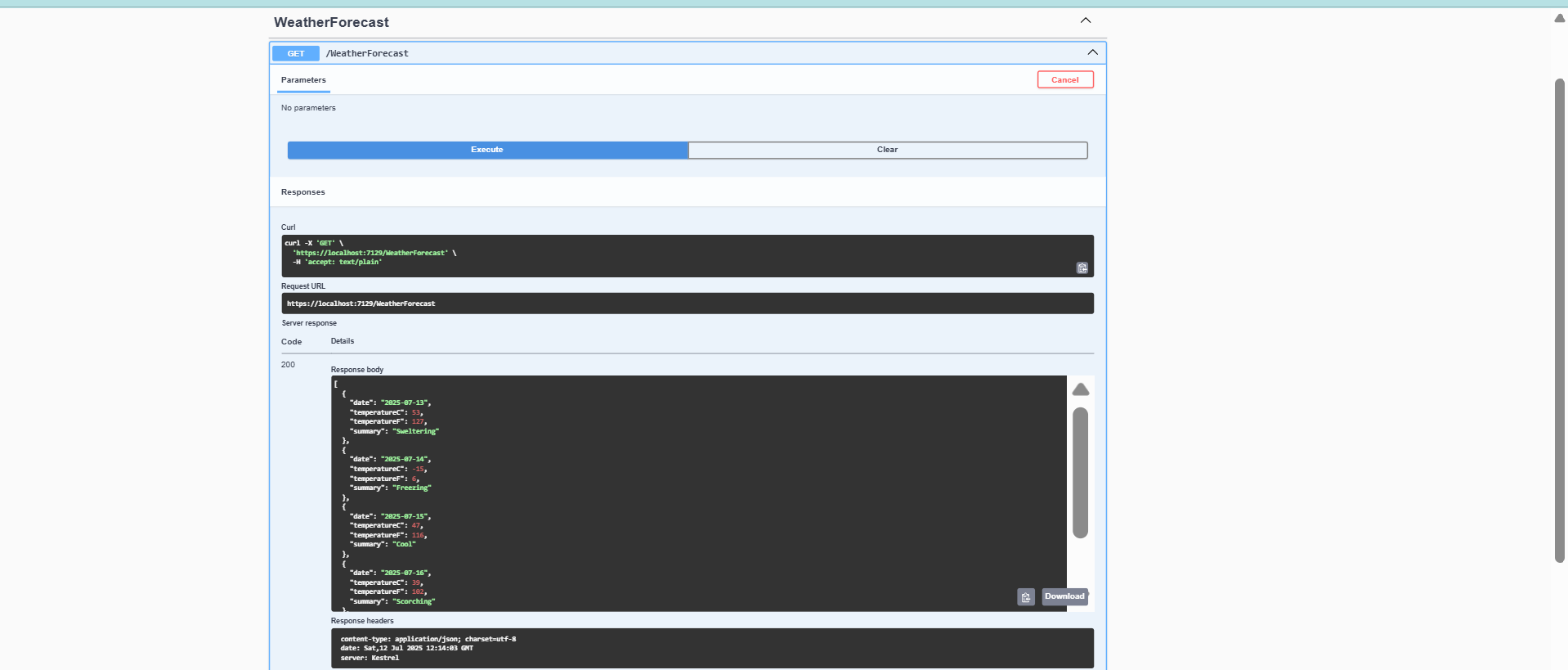
app.UseAuthorization();

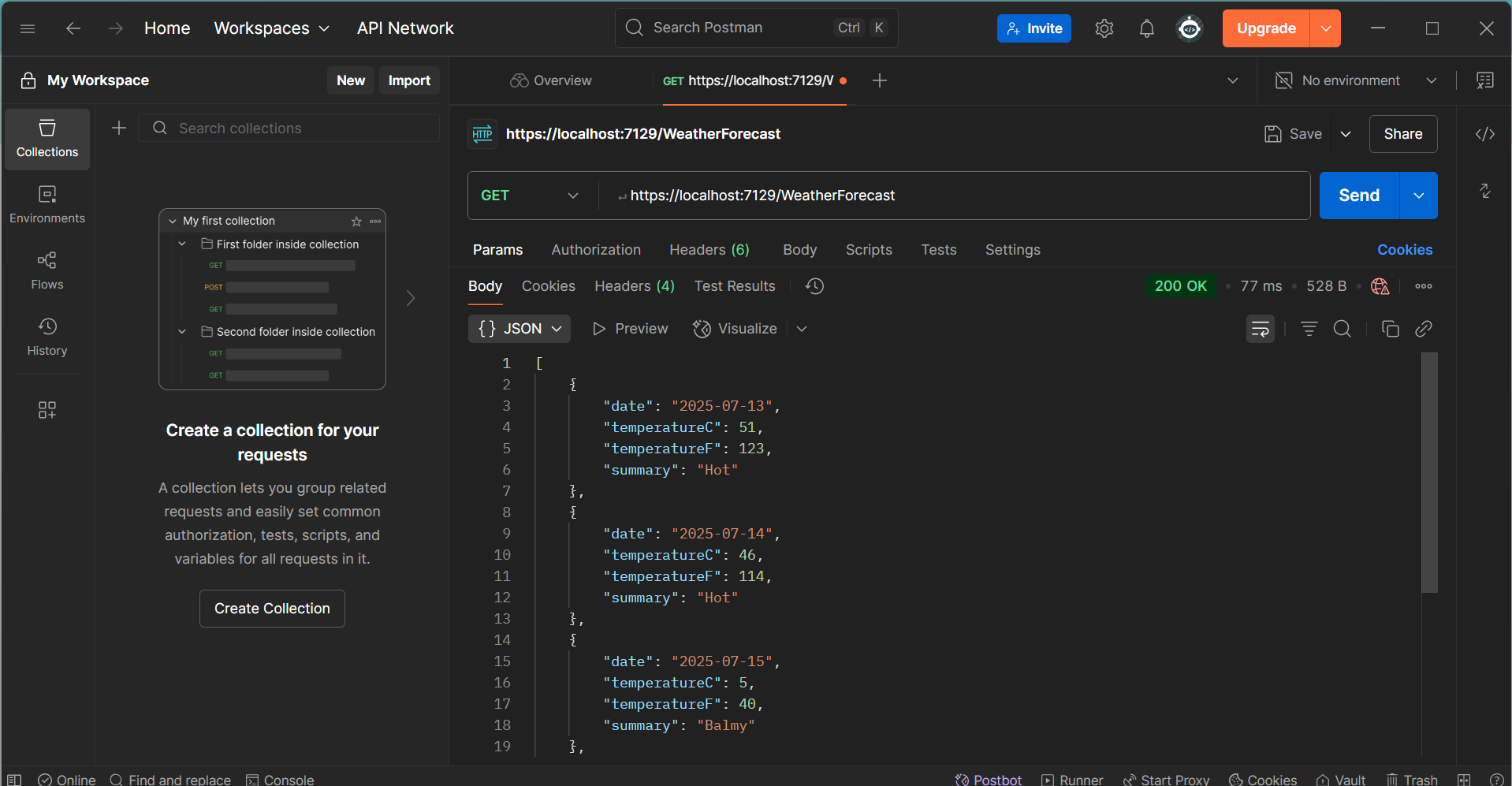
app.MapControllers();

app.Run();

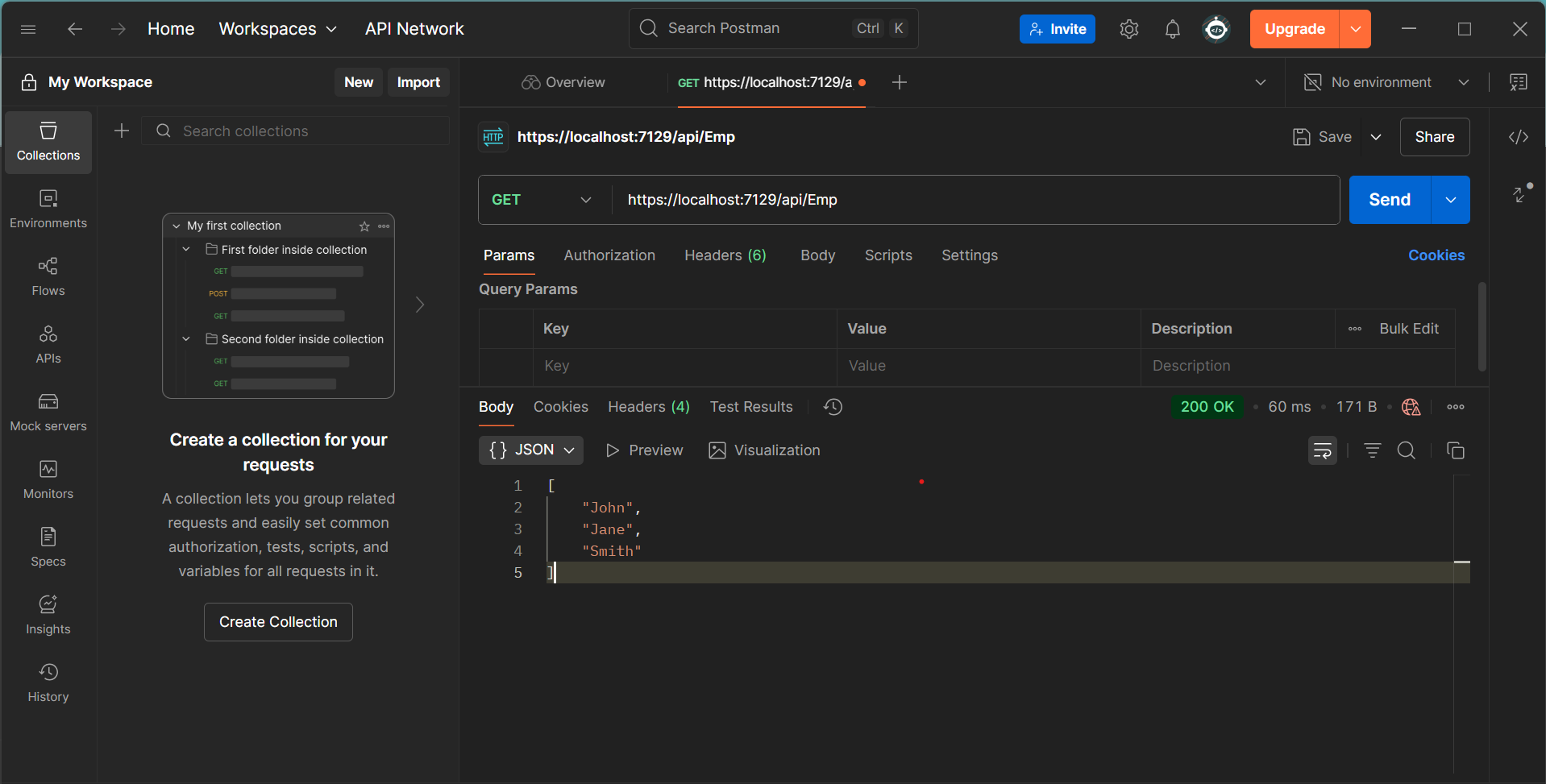


**Use POSTMAN tool, to point to the local Web API that was created with Employee controller. Test the GET action method using POSTMAN.Verify the output if the List of employees are listed in the ‘Body’ part of the GET window on POSTMAN tool.Verify the Status on the right side of the output pane on POSTMAN tool.**





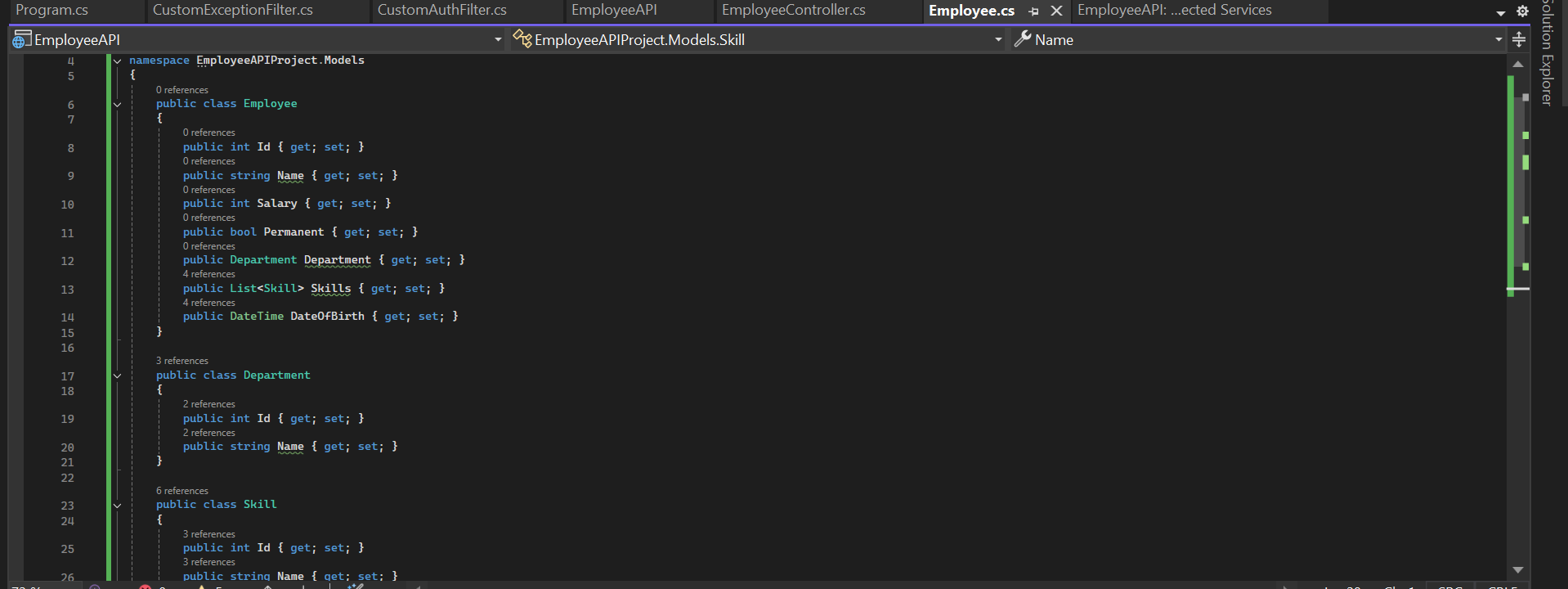
**Modify the Controller name in the Route attribute of the Employee controller to ‘Emp’ and check its access thru POSTMAN**



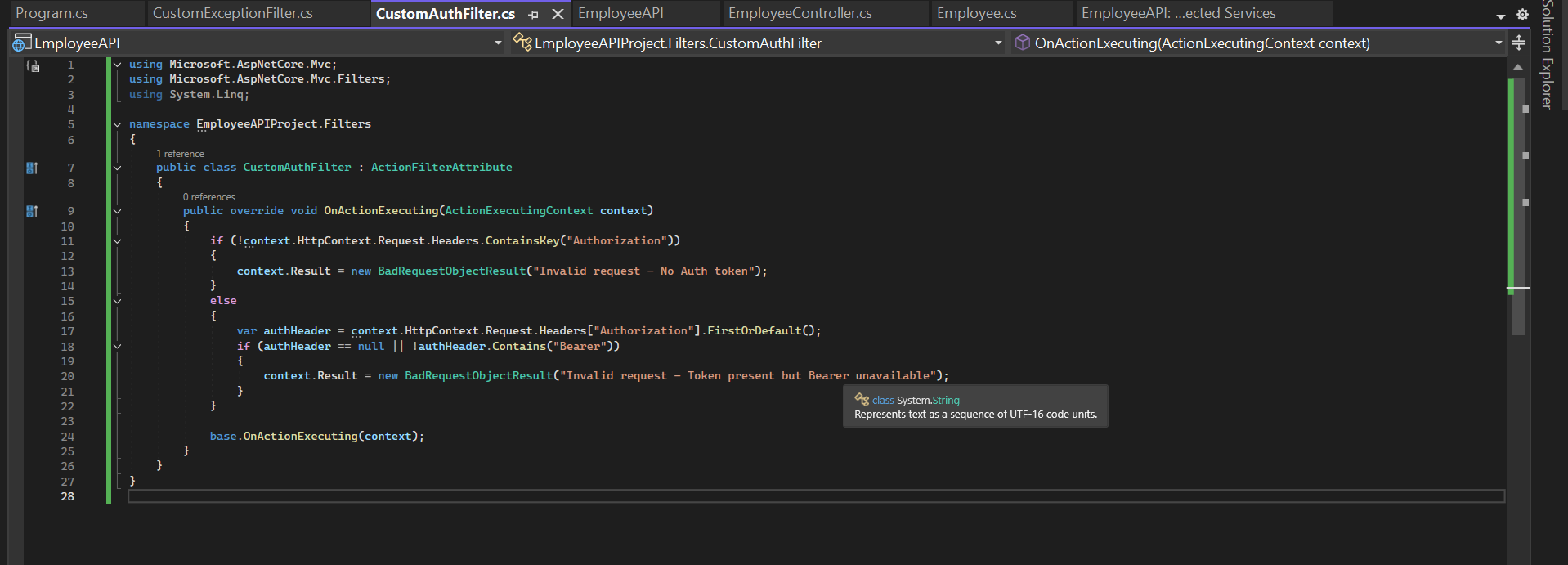
**3.WebApi**

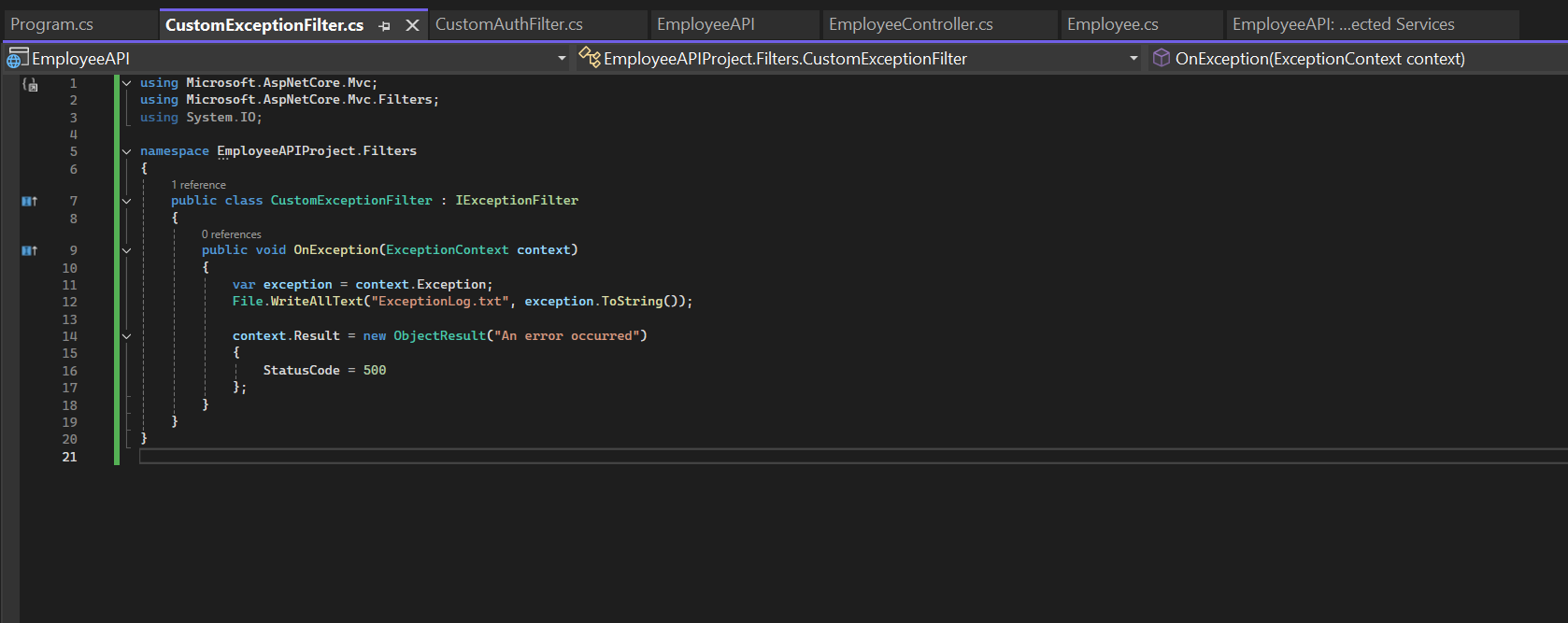
**Web Api using custom model class**

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

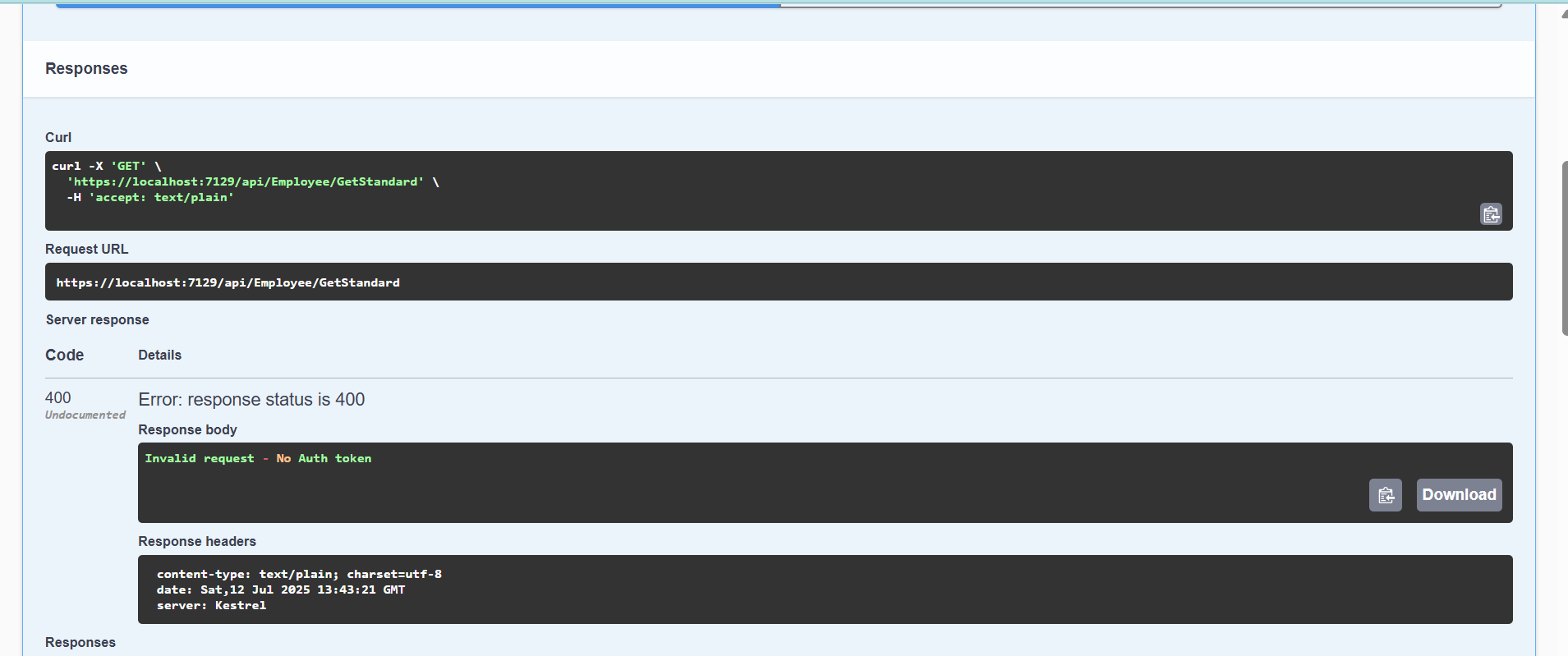


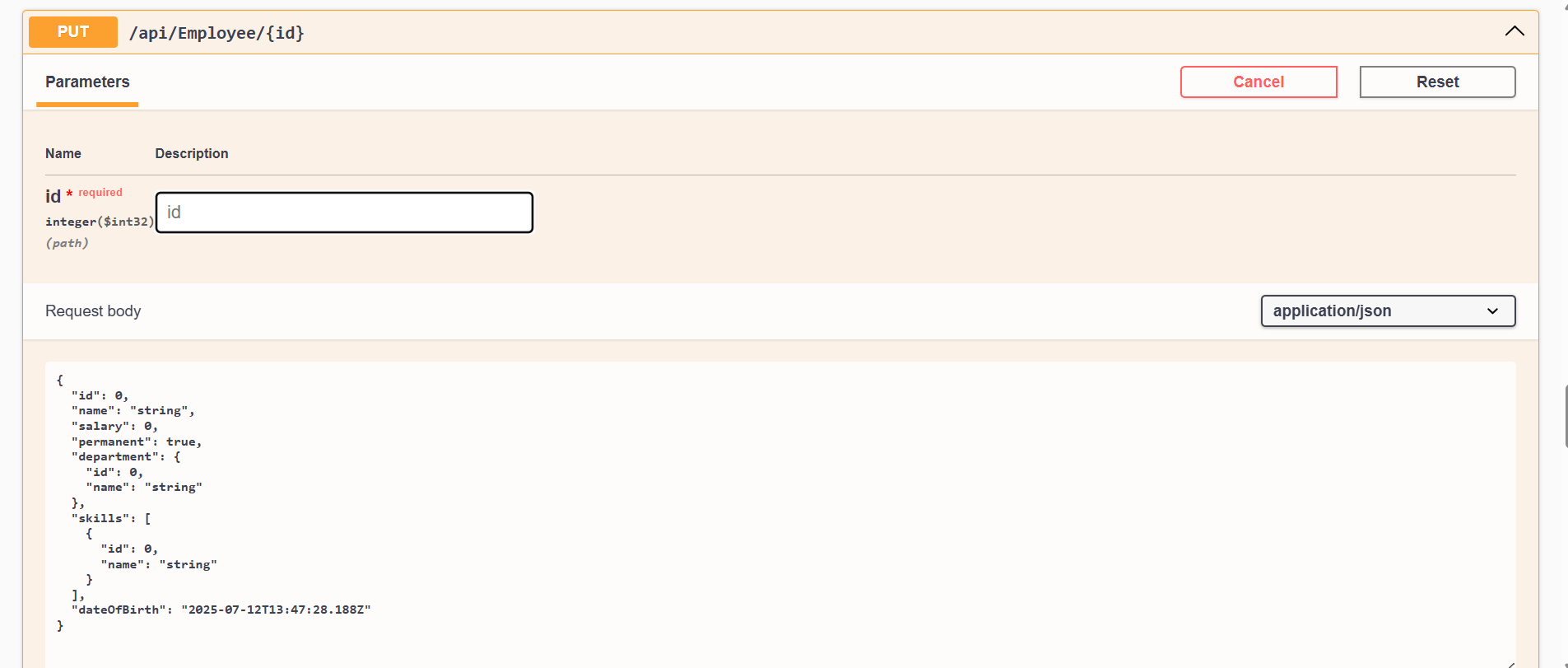
**Create a Custom action filter for Authorization.**



**Custom Exception filter**

RESULTS





**4.WebApi**

**Web Api CRUD operation**

Employeecontroller.cs

using EmployeeAPIProject.Models;

using Microsoft.AspNetCore.Mvc;

namespace EmployeeAPI.Controllers

{

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

[ApiController]

public class EmployeeController : ControllerBase

{

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

{

new Employee { Id=1, Name="John", Salary=50000, Permanent=true },

new Employee { Id=2, Name="Jane", Salary=60000, Permanent=false }

};

[HttpPut("{id}")]

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

{

if (id <= 0)

return BadRequest("Invalid employee id");

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

if (existingEmployee == null)

return BadRequest("Invalid employee id");

existingEmployee.Name = emp.Name;

existingEmployee.Salary = emp.Salary;

existingEmployee.Permanent = emp.Permanent;

existingEmployee.Department = emp.Department;

existingEmployee.Skills = emp.Skills;

existingEmployee.DateOfBirth = emp.DateOfBirth;

return Ok(existingEmployee);

}

}

}

Program.cs

using EmployeeAPIProject.Filters;

using Microsoft.OpenApi.Models;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers(options =>

{

options.Filters.Add<CustomExceptionFilter>();

});

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen(c =>

{

c.SwaggerDoc("v1", new OpenApiInfo { Title = "My API", Version = "v1" });

c.AddSecurityDefinition("Bearer", new OpenApiSecurityScheme

{

In = ParameterLocation.Header,

Description = "Please enter token",

Name = "Authorization",

Type = SecuritySchemeType.ApiKey

});

c.AddSecurityRequirement(new OpenApiSecurityRequirement

{

{

new OpenApiSecurityScheme

{

Reference = new OpenApiReference { Type = ReferenceType.SecurityScheme, Id = "Bearer" }

},

new string[] { }

}

});

});

var app = builder.Build();

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI();

}

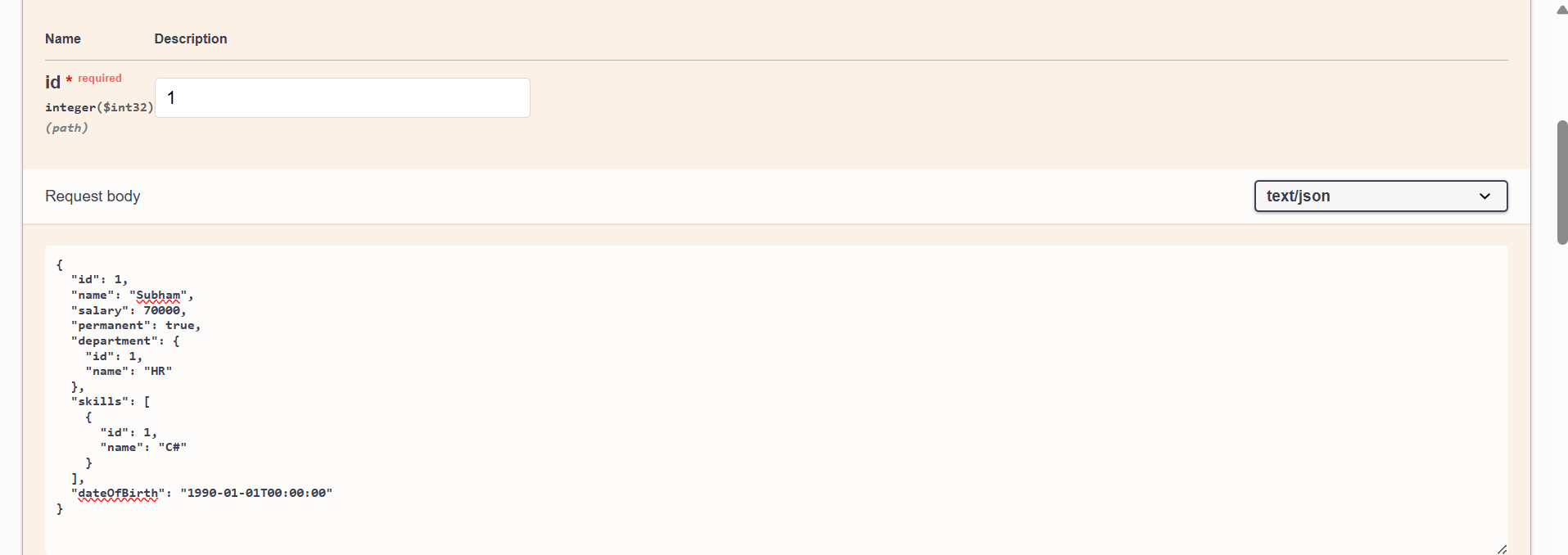
app.UseHttpsRedirection();

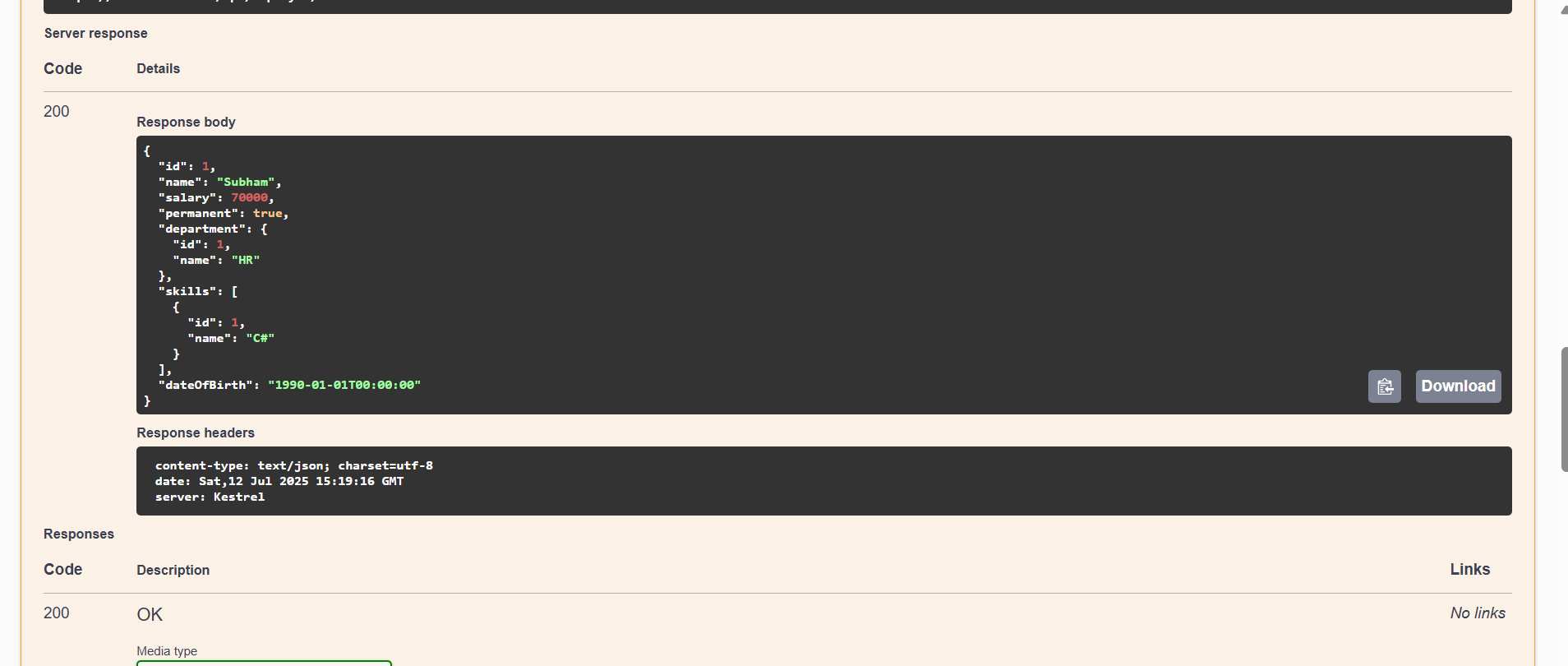
app.UseAuthorization();

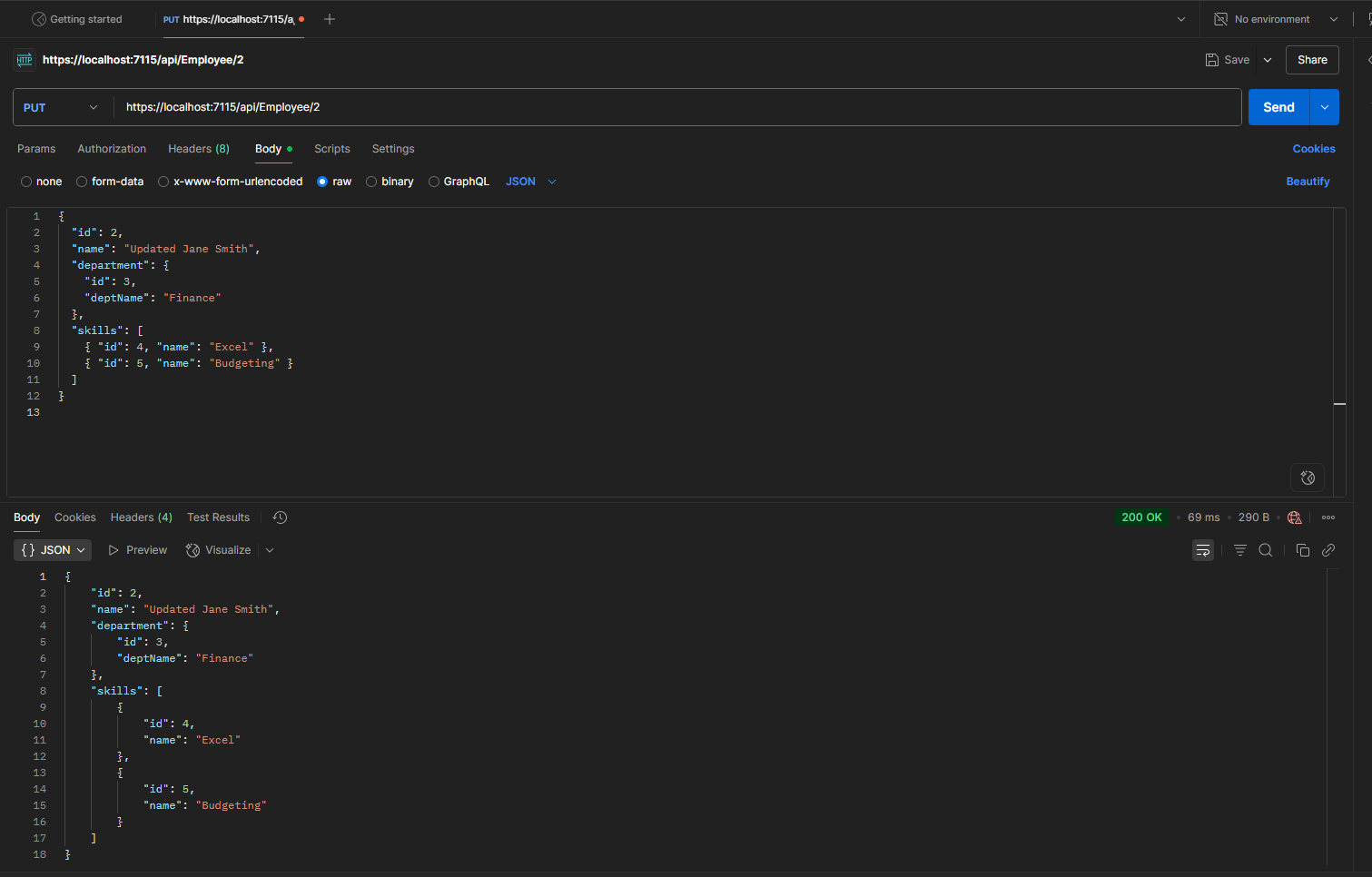
app.MapControllers();

app.Run();

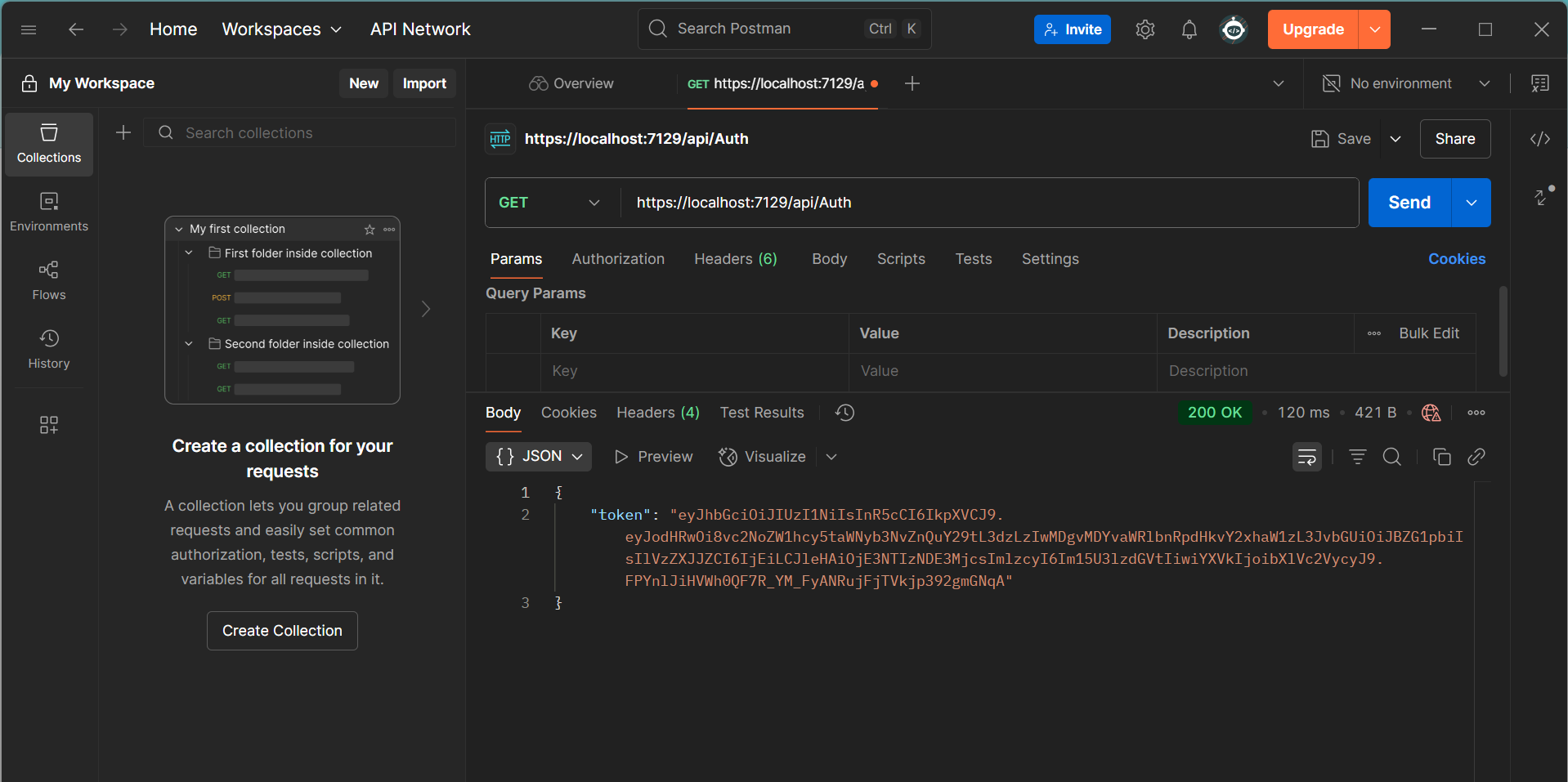
**Result**



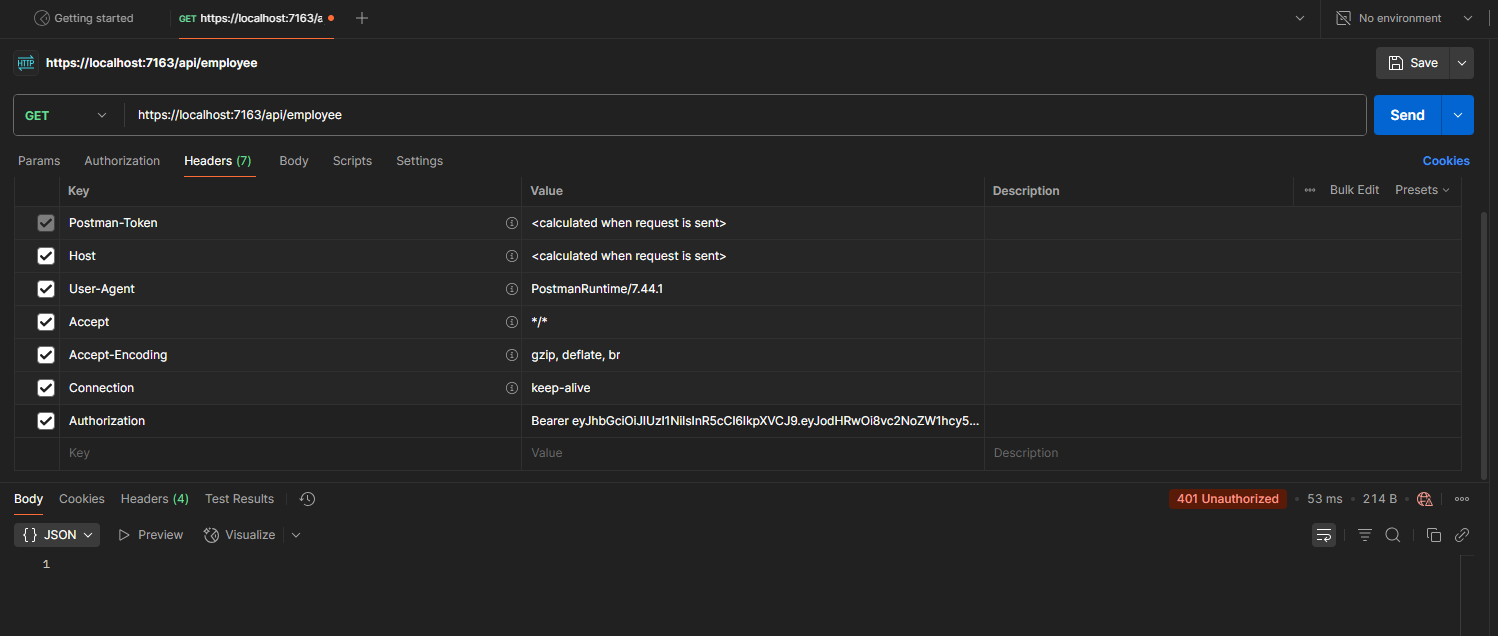




**5.WebApi**



**#Check for JWT expiration :**



**#Only POC:**

