# Module 6: ASP.NET Core 8.0 Web API – Learning Objectives

## 1. ****Concept of RESTful Web Service, Web API & Microservice****

### ****RESTful Web Service****

REST (Representational State Transfer) is an architectural style for designing networked applications. It uses standard HTTP methods (GET, POST, PUT, DELETE) and stateless communication.

**Key Features of REST:**

**Stateless:** Each request from client to server must contain all the information to understand and process the request.

**Representation:** Data is represented in formats like JSON or XML.

**Uniform Interface:** Uses standard HTTP methods and URIs.

**Resource-Based:** Everything is treated as a resource (e.g., /products/5).

### ****Web API****

Web API is a framework for building HTTP services in ASP.NET Core that can be accessed by a broad range of clients, including browsers, mobile apps, and other servers.

**Key Points:**

Built using REST principles

Supports JSON/XML (not restricted to XML)

Lightweight and ideal for mobile/web apps

### ****Microservice****

A Microservice is a small, independently deployable service that handles a specific business functionality. Web APIs are commonly used to implement microservices.

**Benefits of Microservices:**

Scalability

Independent deployment

Technology flexibility

## 2. ****What is HttpRequest & HttpResponse****

### ****HttpRequest:****

Represents the incoming HTTP request. It includes:

Request headers

URL and query strings

HTTP method (GET, POST, etc.)

Request body (for POST/PUT)

### ****HttpResponse:****

Represents the response sent from the server to the client. It includes:

Status code (e.g., 200 OK)

Headers

Response body (e.g., JSON data)

## 3. ****Types of Action Verbs****

### 🔹 ****HTTP Methods Used in Web API:****

1. GET  
   2. POST  
   3. PUT  
   4. DELETE  
     
   4. **Types of HTTP Status Codes Used in Web API  
     
   1.**200 OK  
   2.400 BadRequest  
   3.401 Unauthorized  
   4.500 InternalServerError  
     
   5. **Demonstration: Simple Web API with Read/Write Actions :  
     
   Controller Structure:**

CODE:

[ApiController]

[Route("[controller]")]

public class SampleController : ControllerBase

{

private static List<string> items = new List<string> { "Item1", "Item2" };

[HttpGet]

public IActionResult GetItems()

{

return Ok(items); }

[HttpPost]

public IActionResult AddItem([FromBody] string item)

{

items.Add(item);

return Ok(items);

}

## } 6. ****Configuration Files in ASP.NET Core Web API****

### Program.cs & Startup.cs (ASP.NET Core <=5.0)

Registers services using **Dependency Injection.**

Configures Middleware (e.g., routing, authentication, exception handling)

public void ConfigureServices(IServiceCollection services)

{

services.AddControllers();

}

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

app.UseRouting();

app.UseEndpoints(endpoints => { endpoints.MapControllers(); });

}

appsettings.json:  
  
{

"ConnectionStrings": {

"Default": "Data Source=app.db"

},

"Jwt": {

"Key": "SuperSecretKey123"

}

}

launchSettings.json:  
  
"profiles": {

"FirstWebAPI": {

"commandName": "Project",

"launchBrowser": true,

"applicationUrl": "https://localhost:5001",

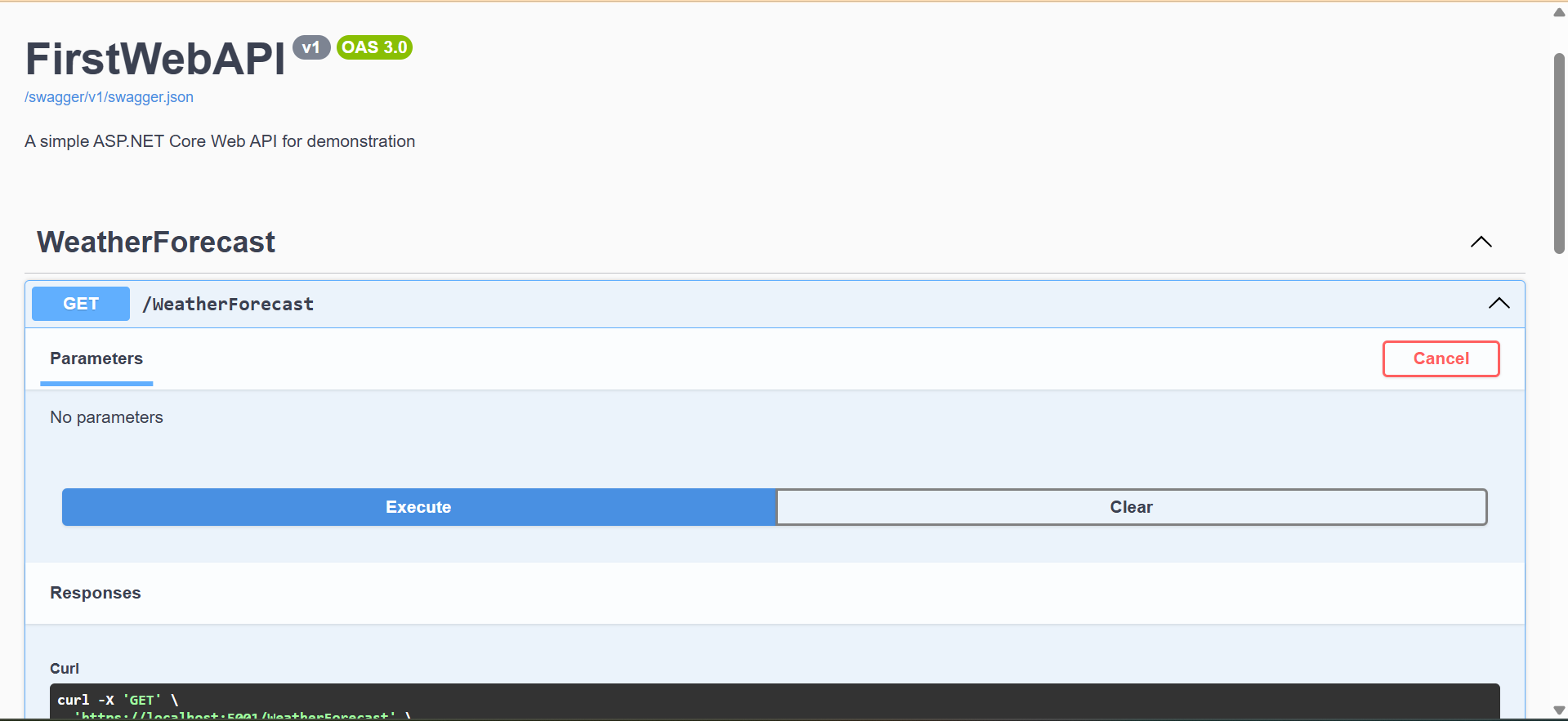
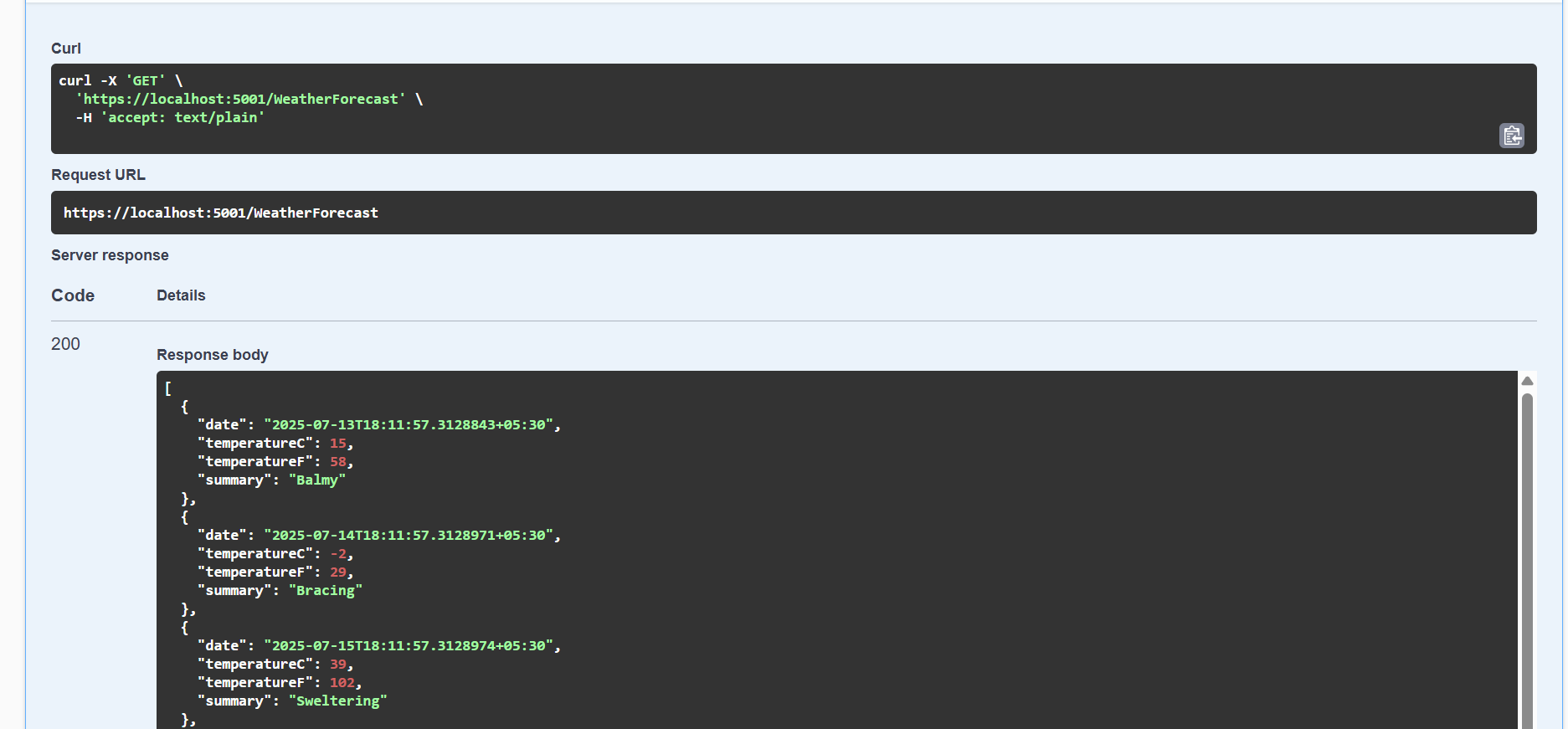
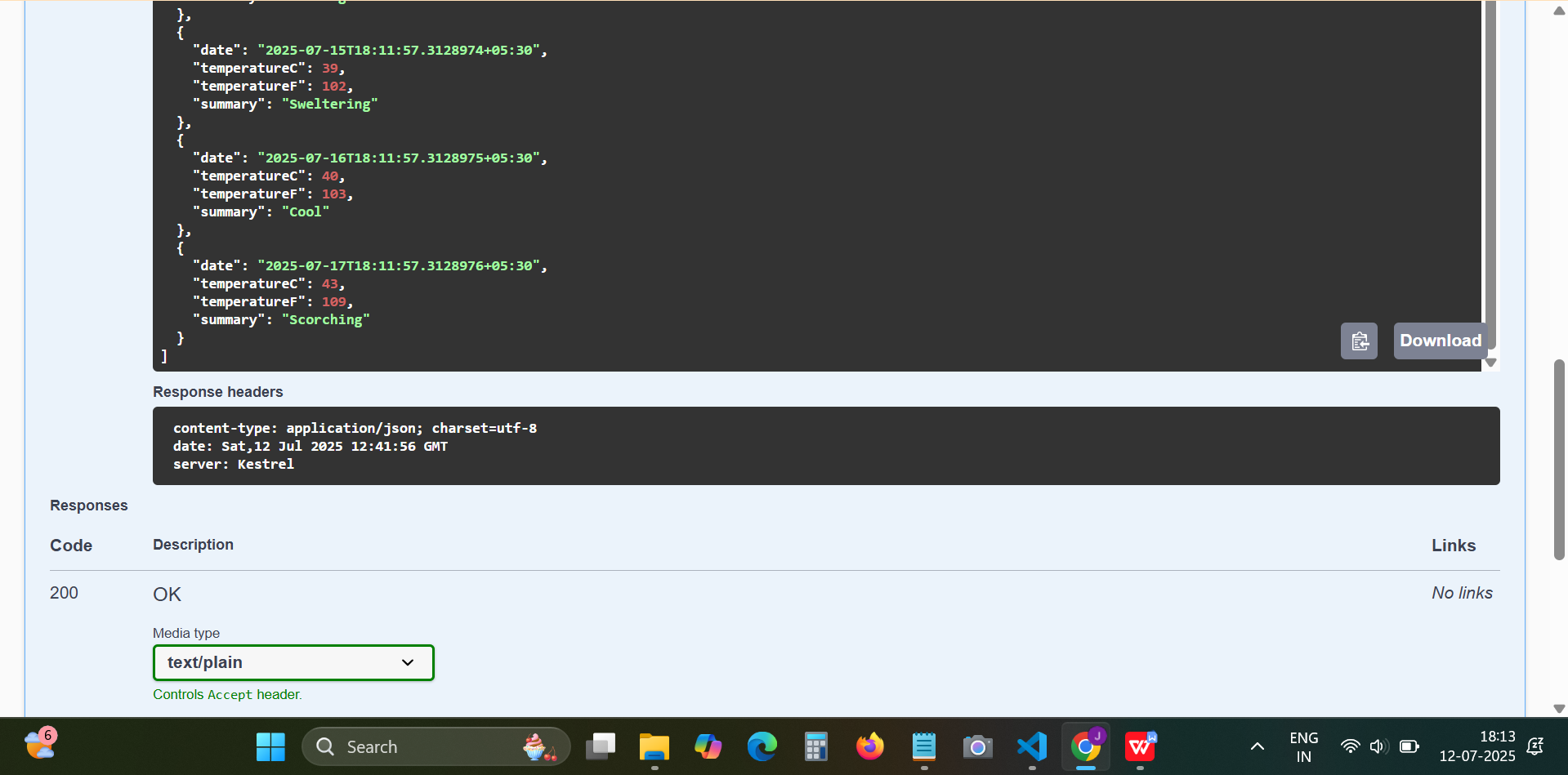
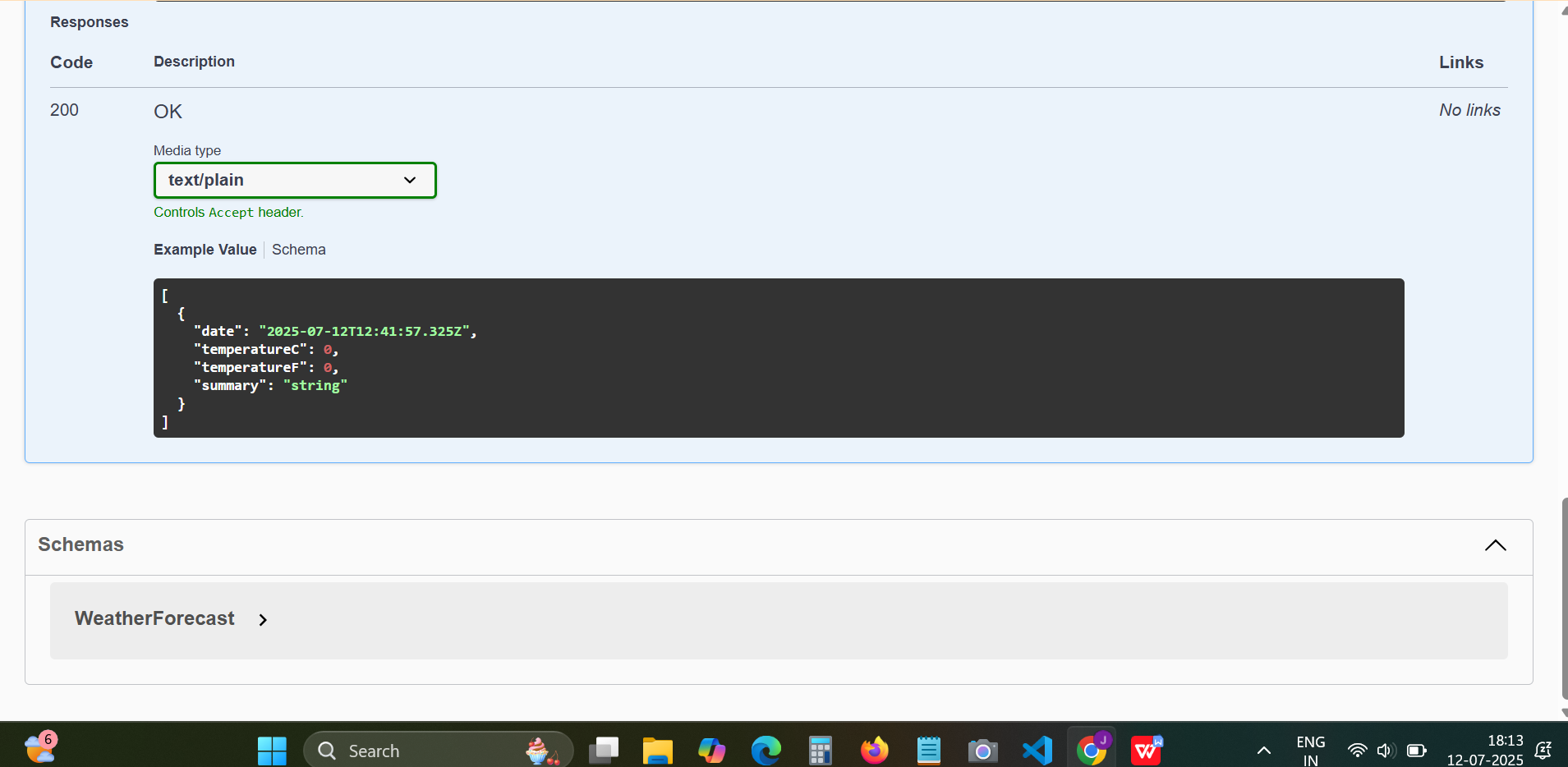
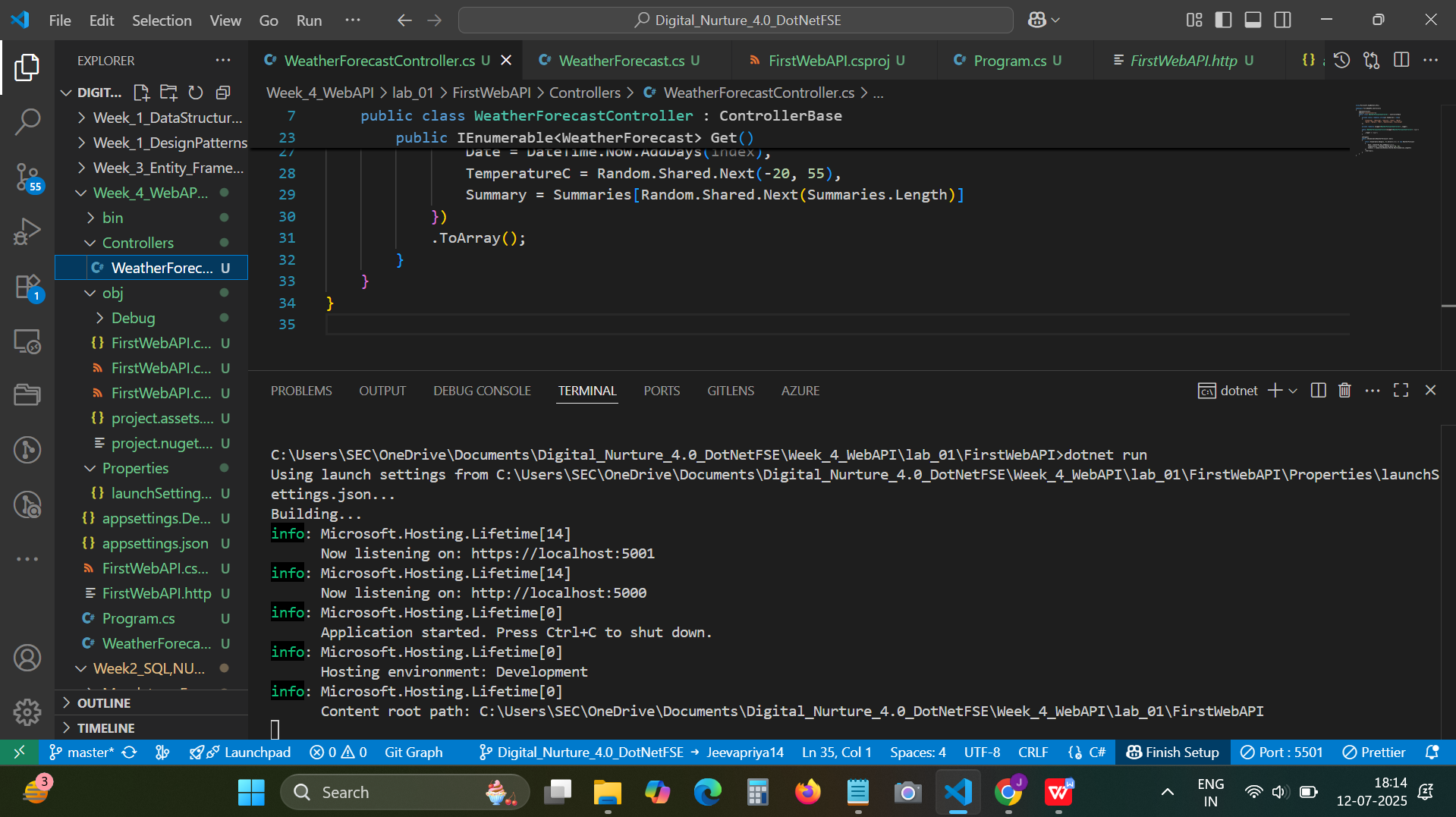
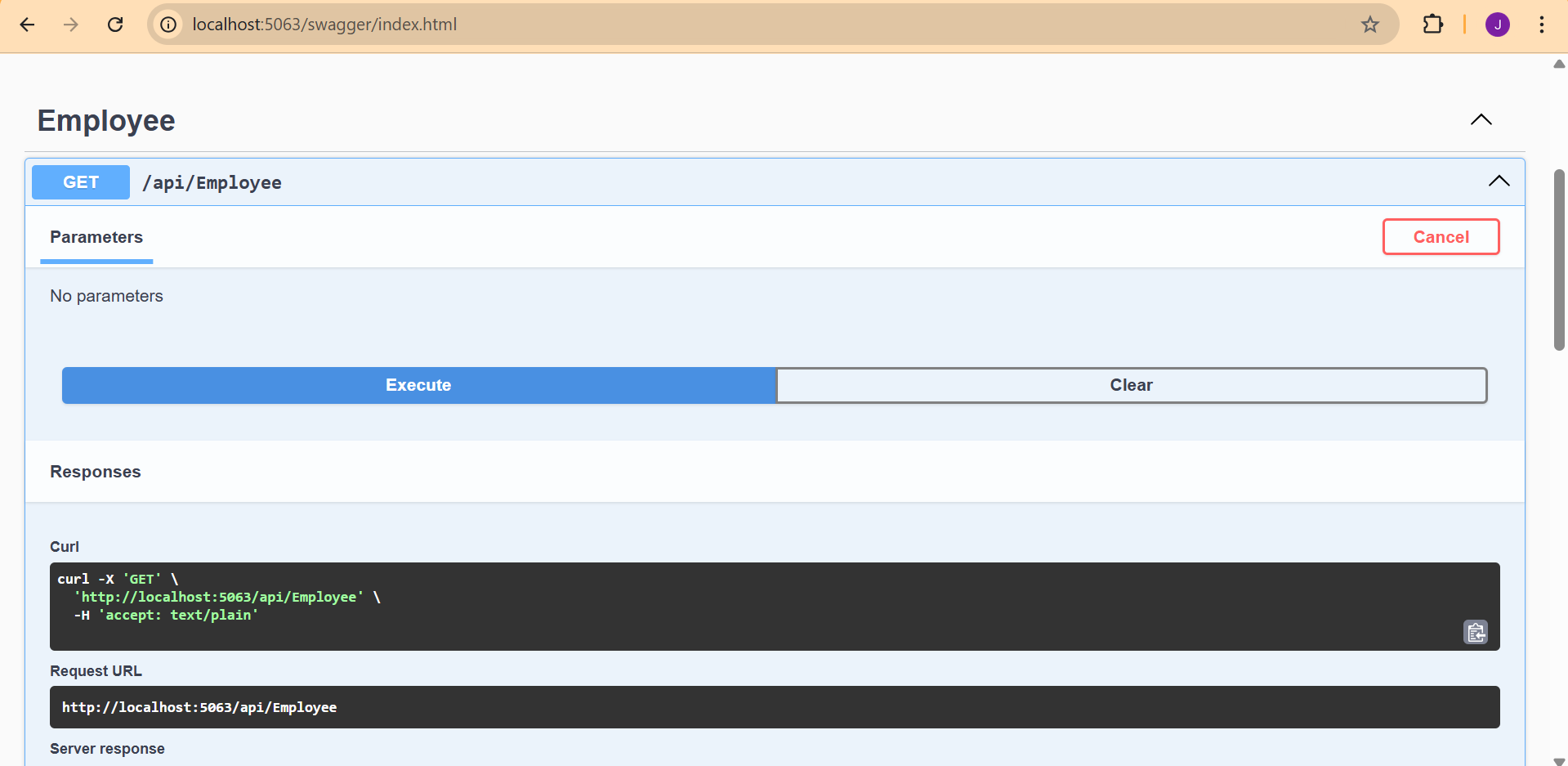
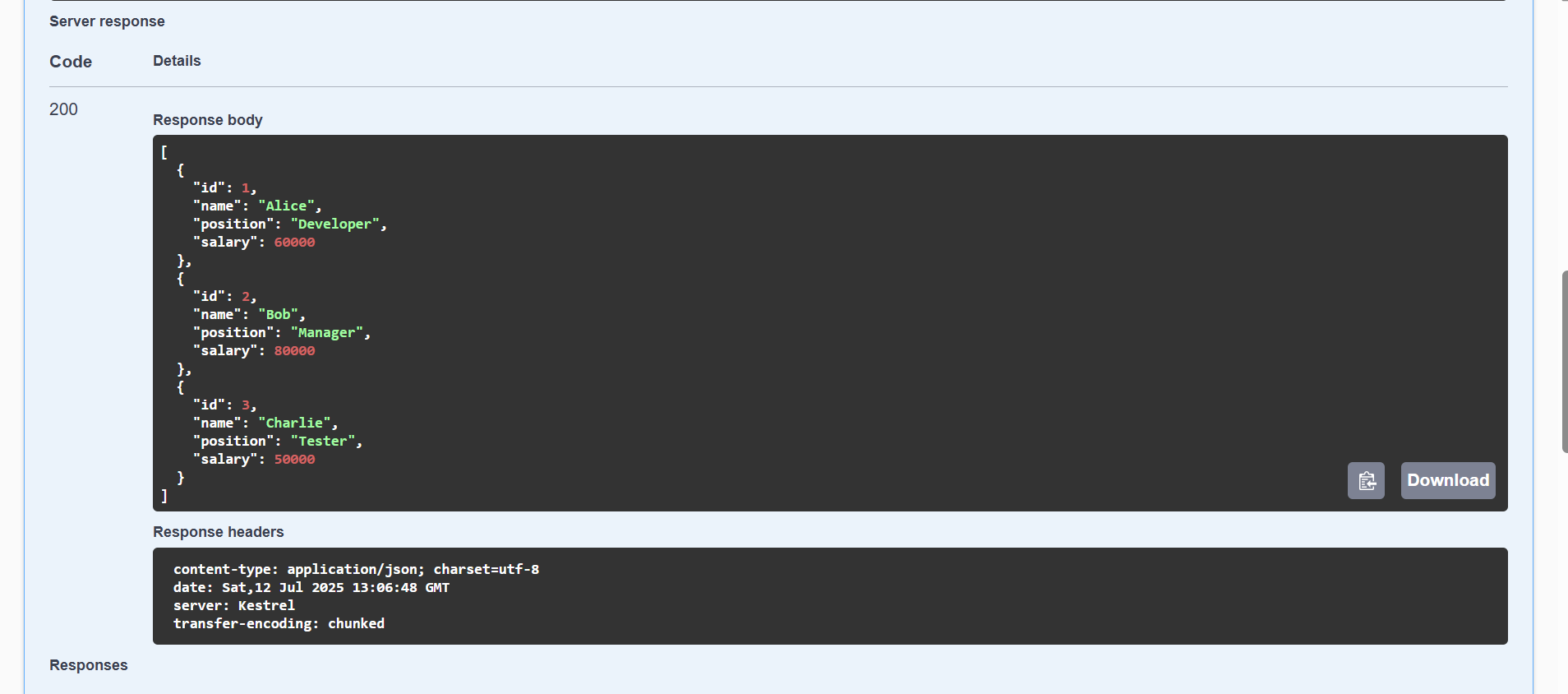
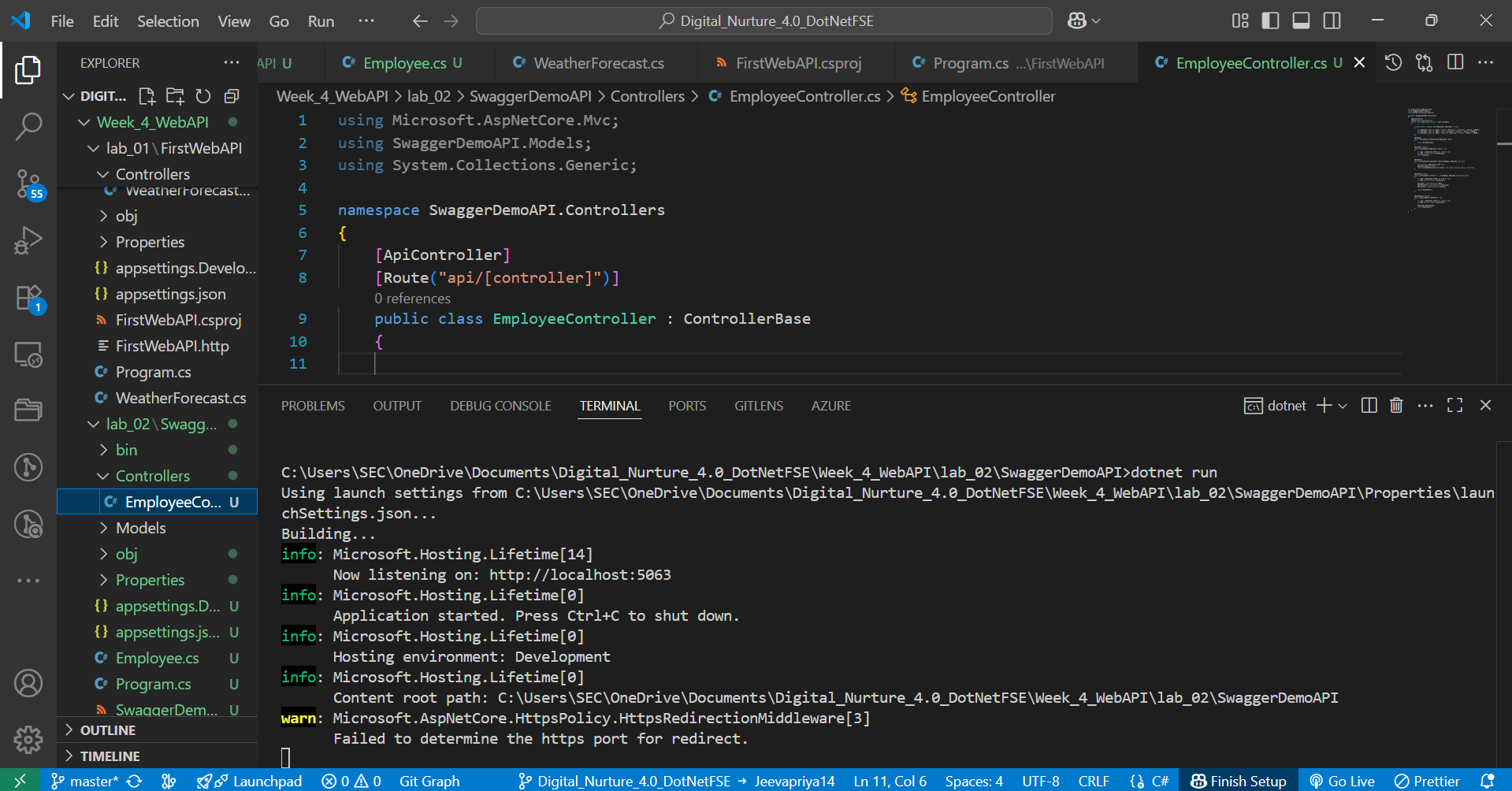
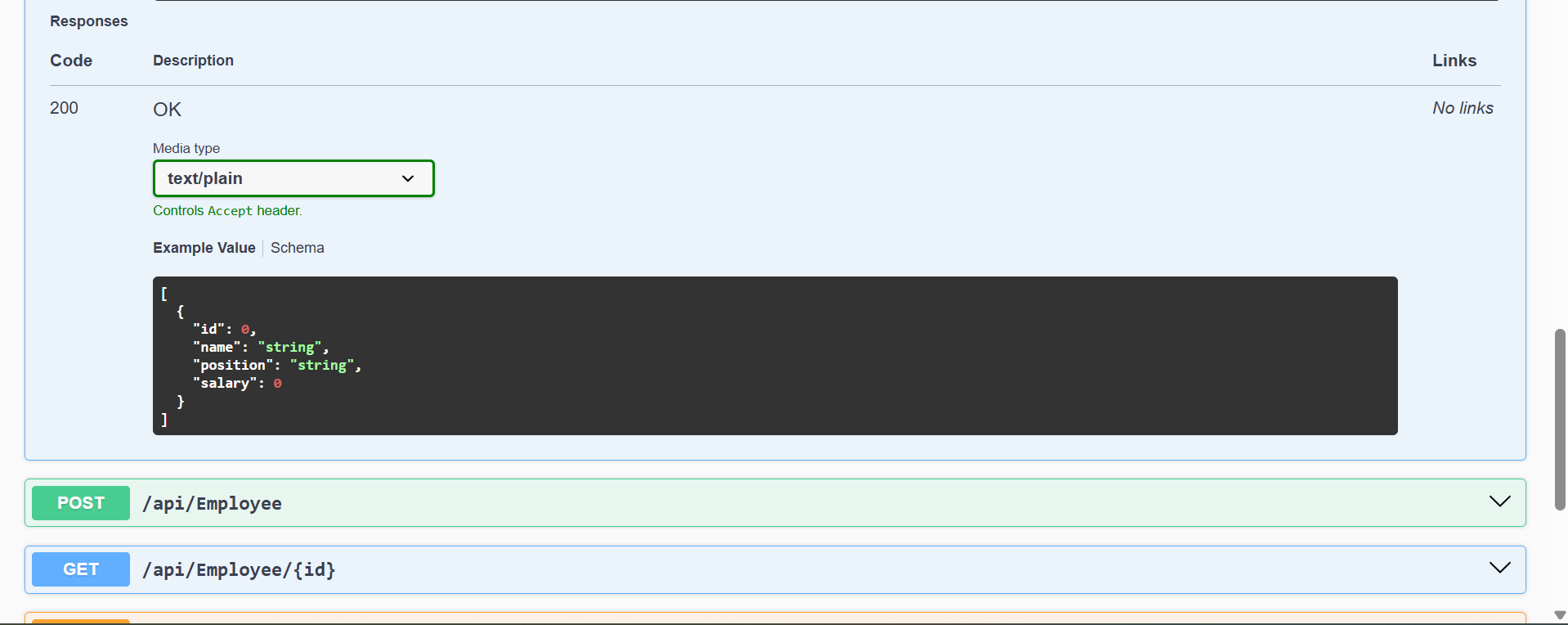
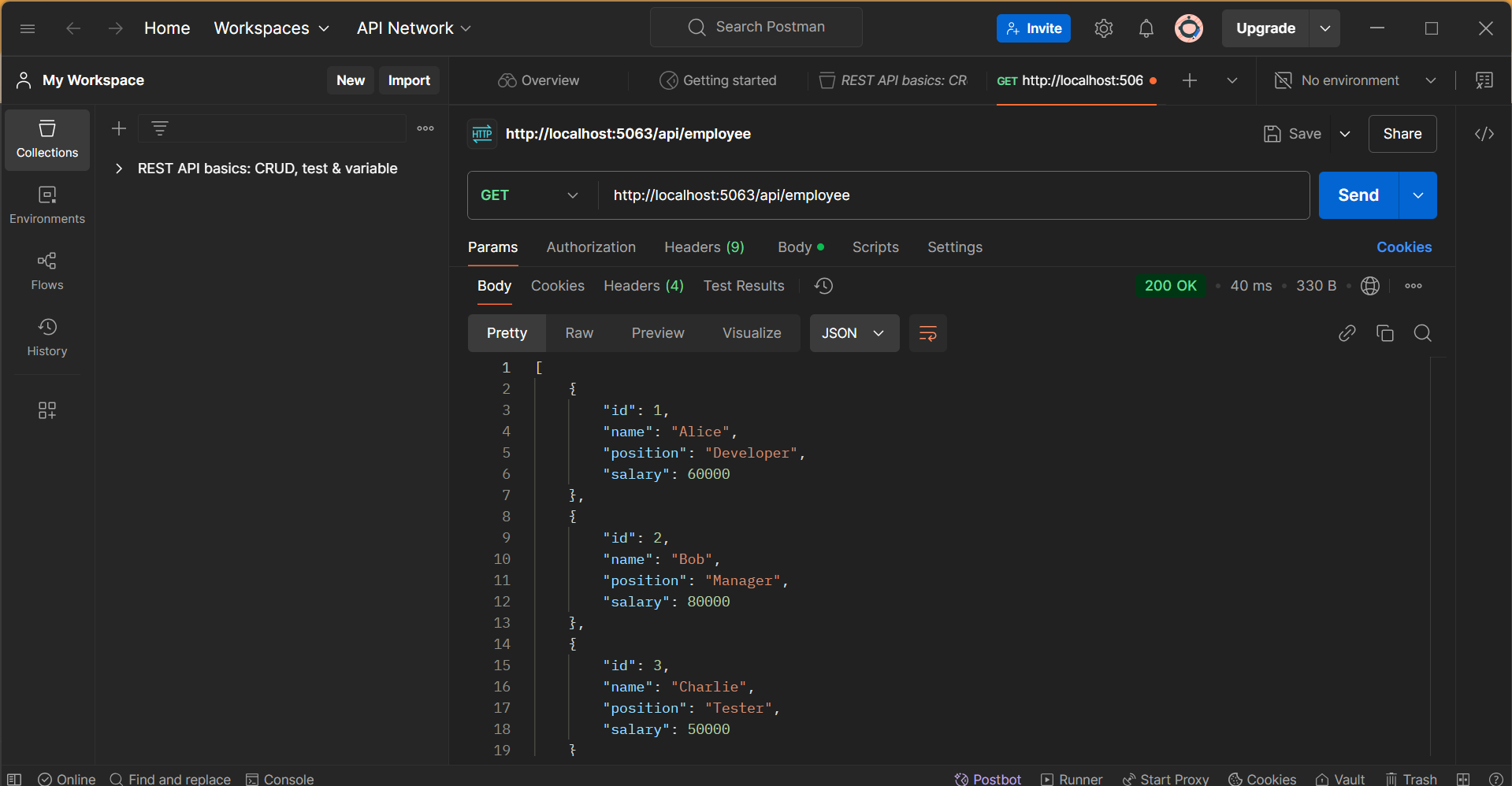
"environmentVariables": {

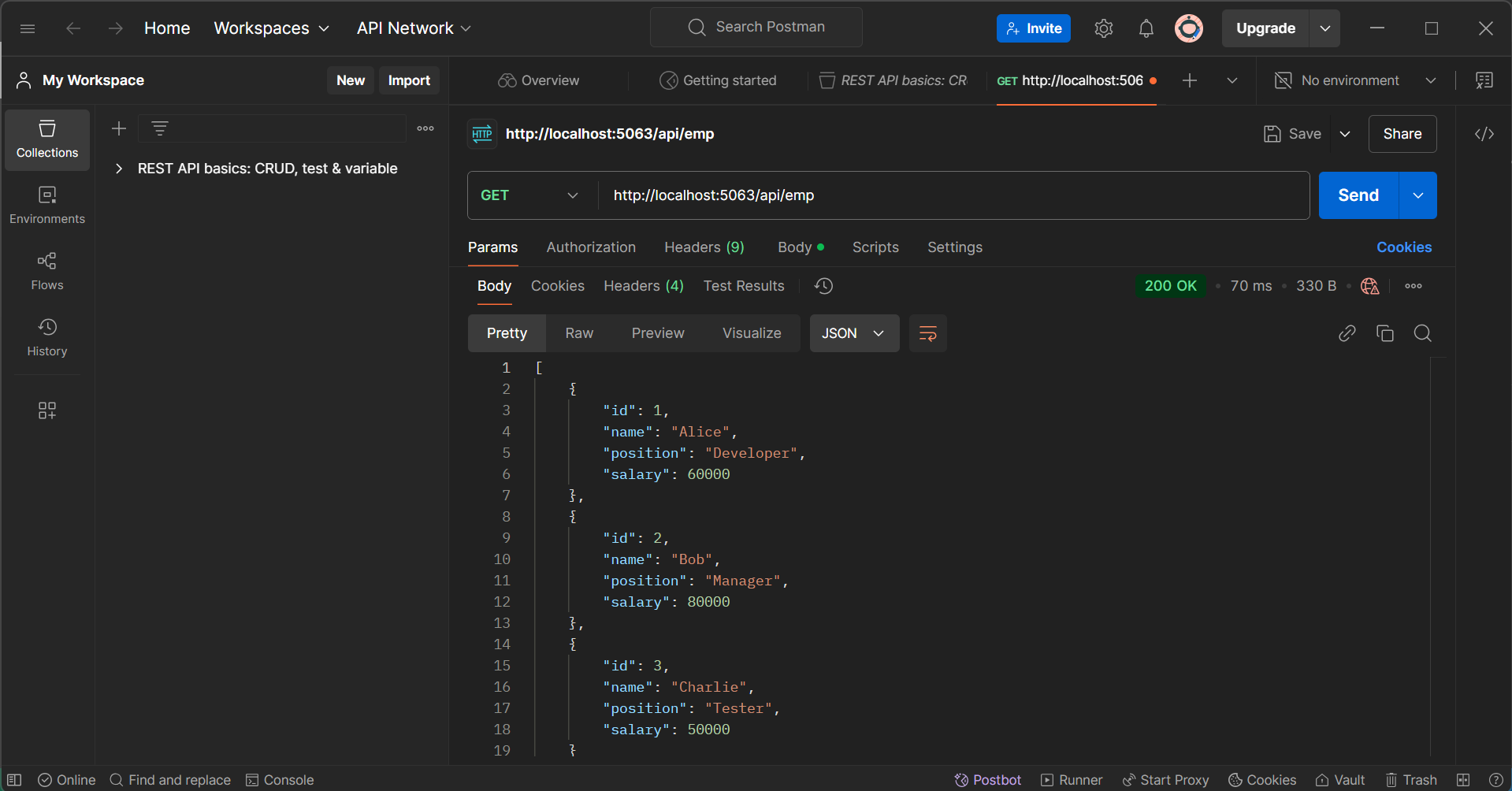
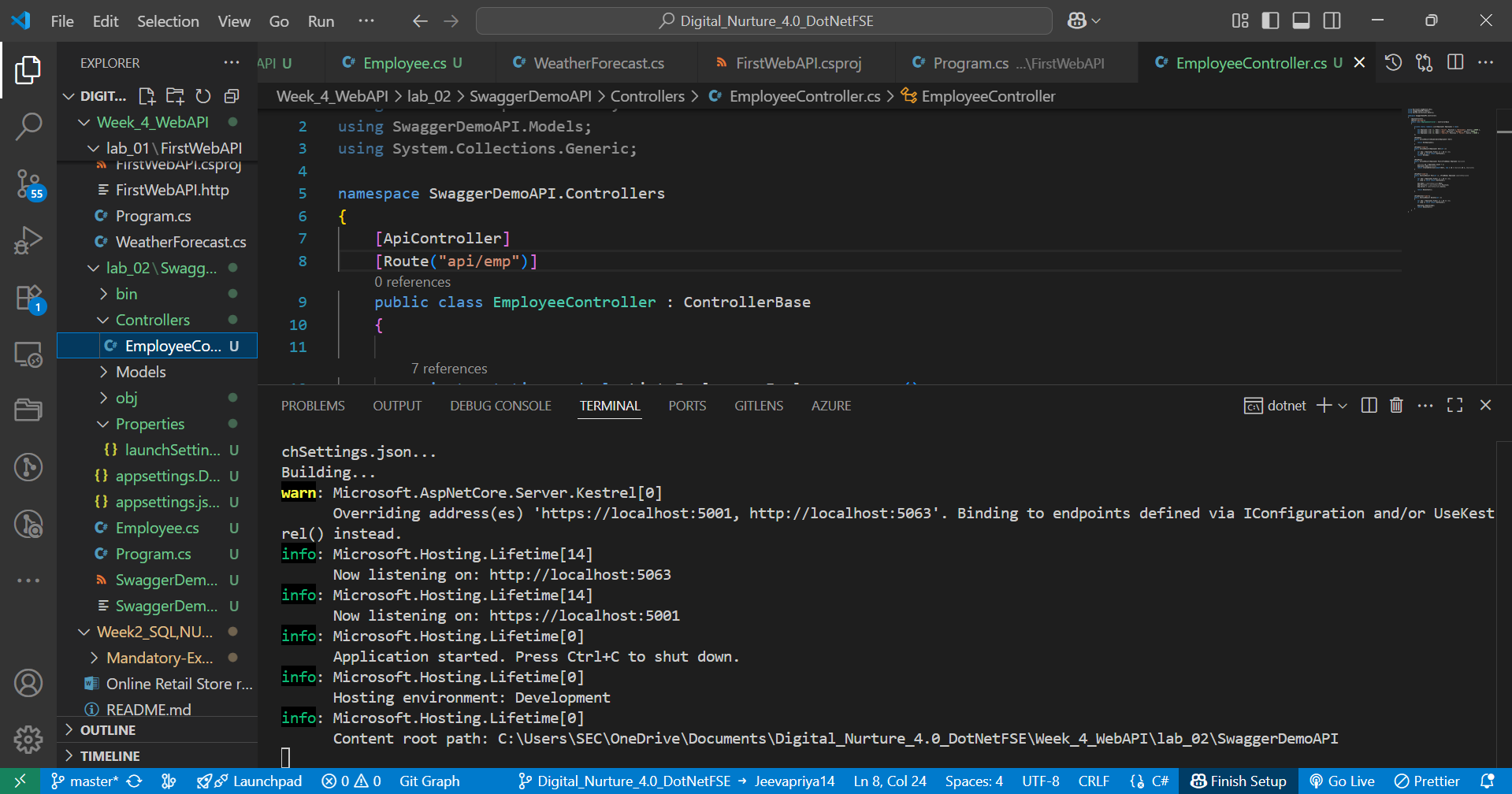
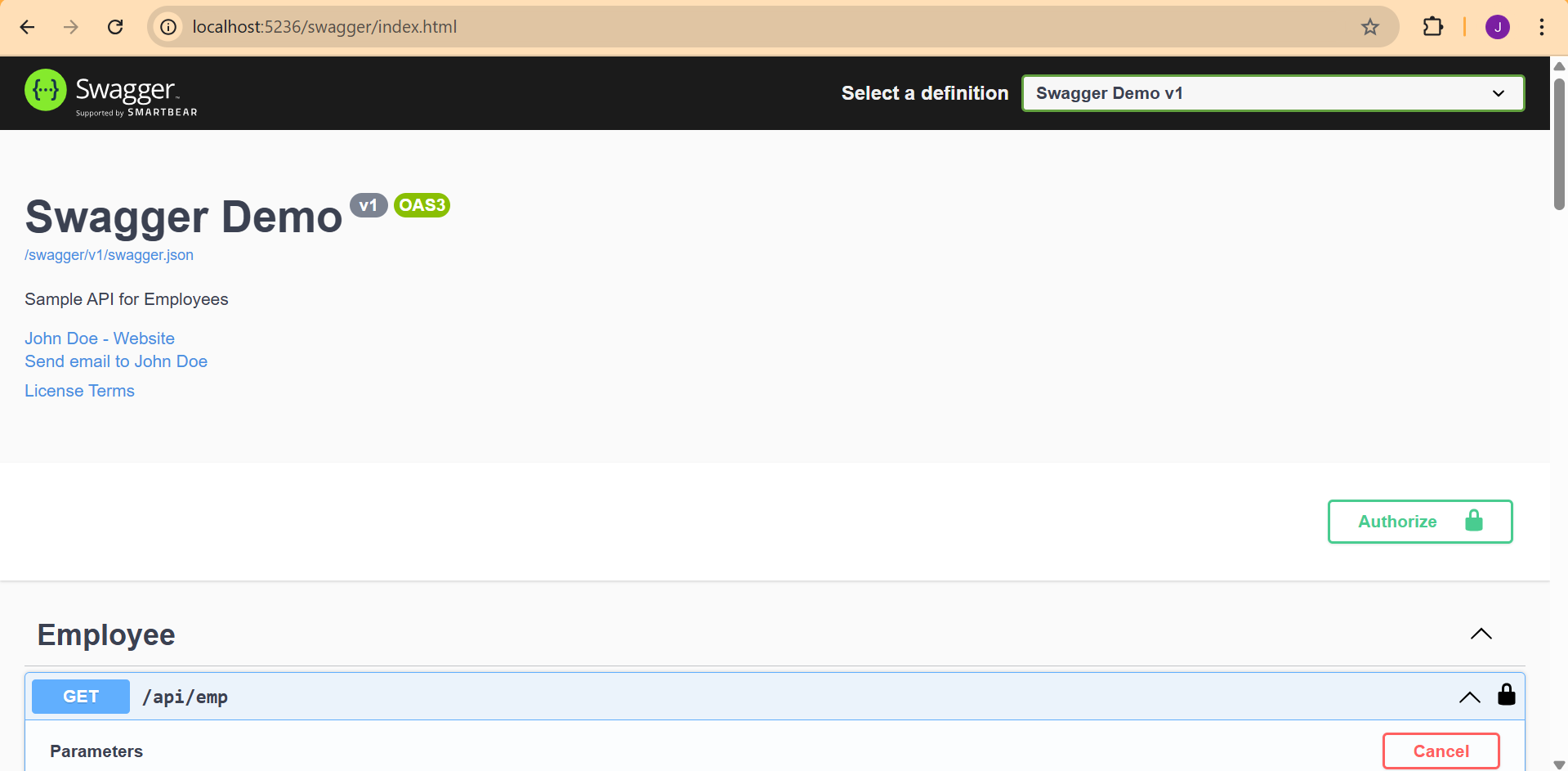
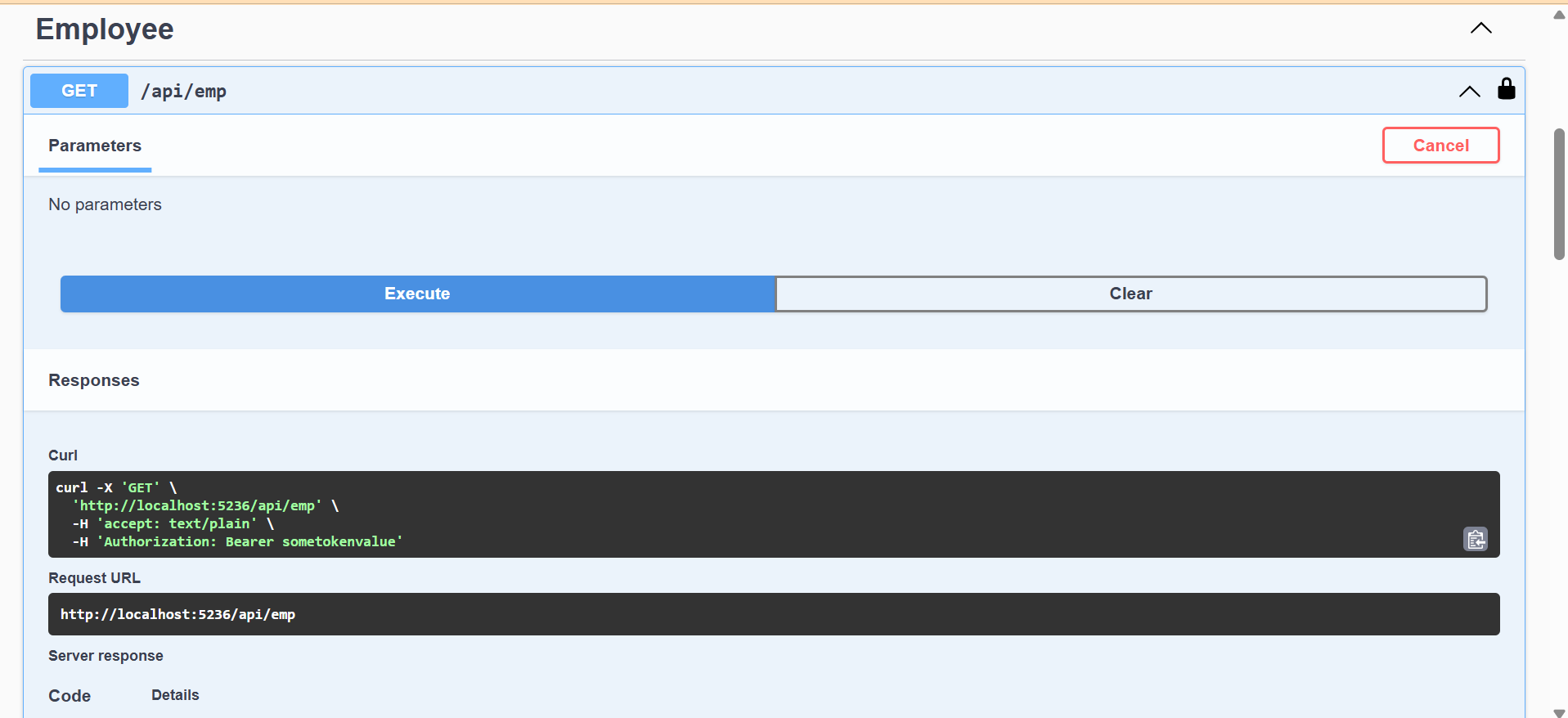
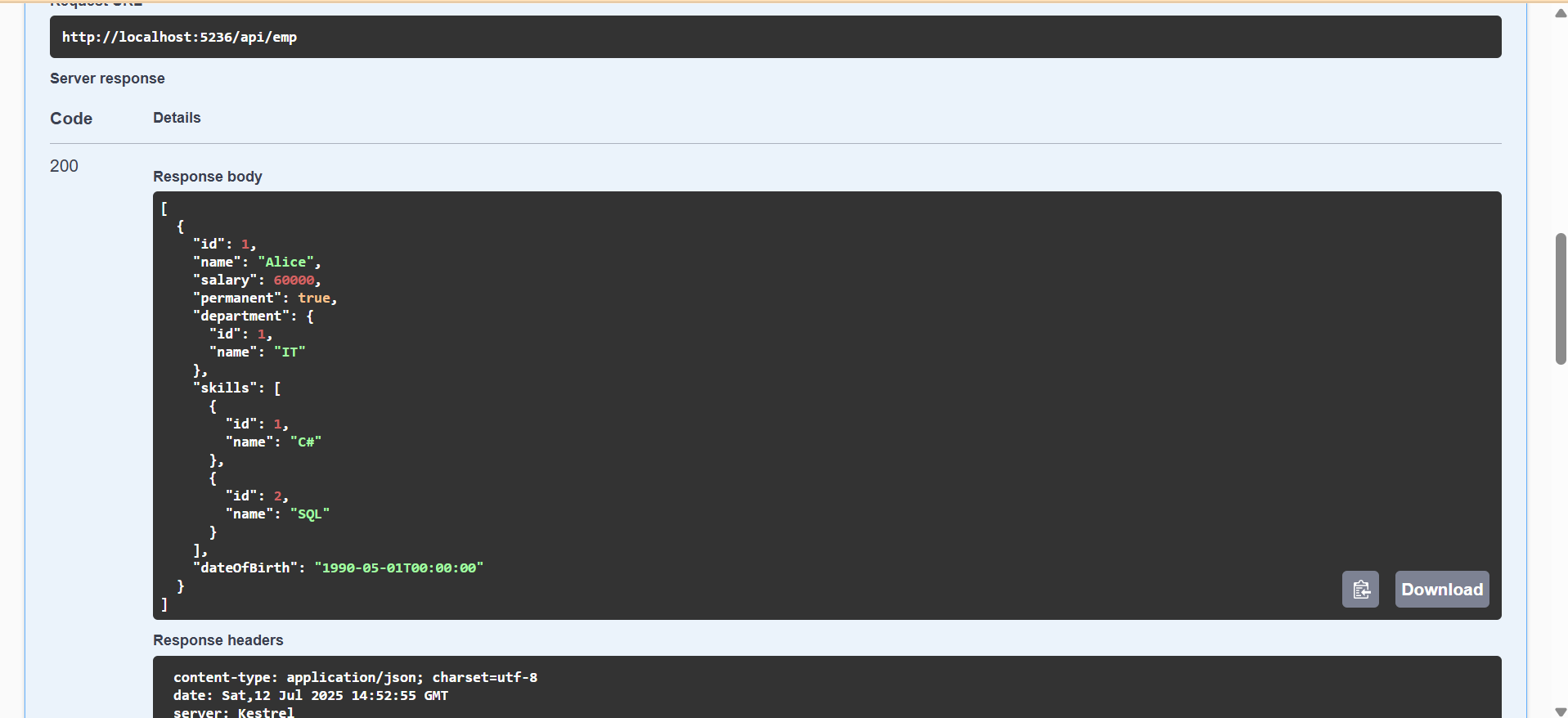
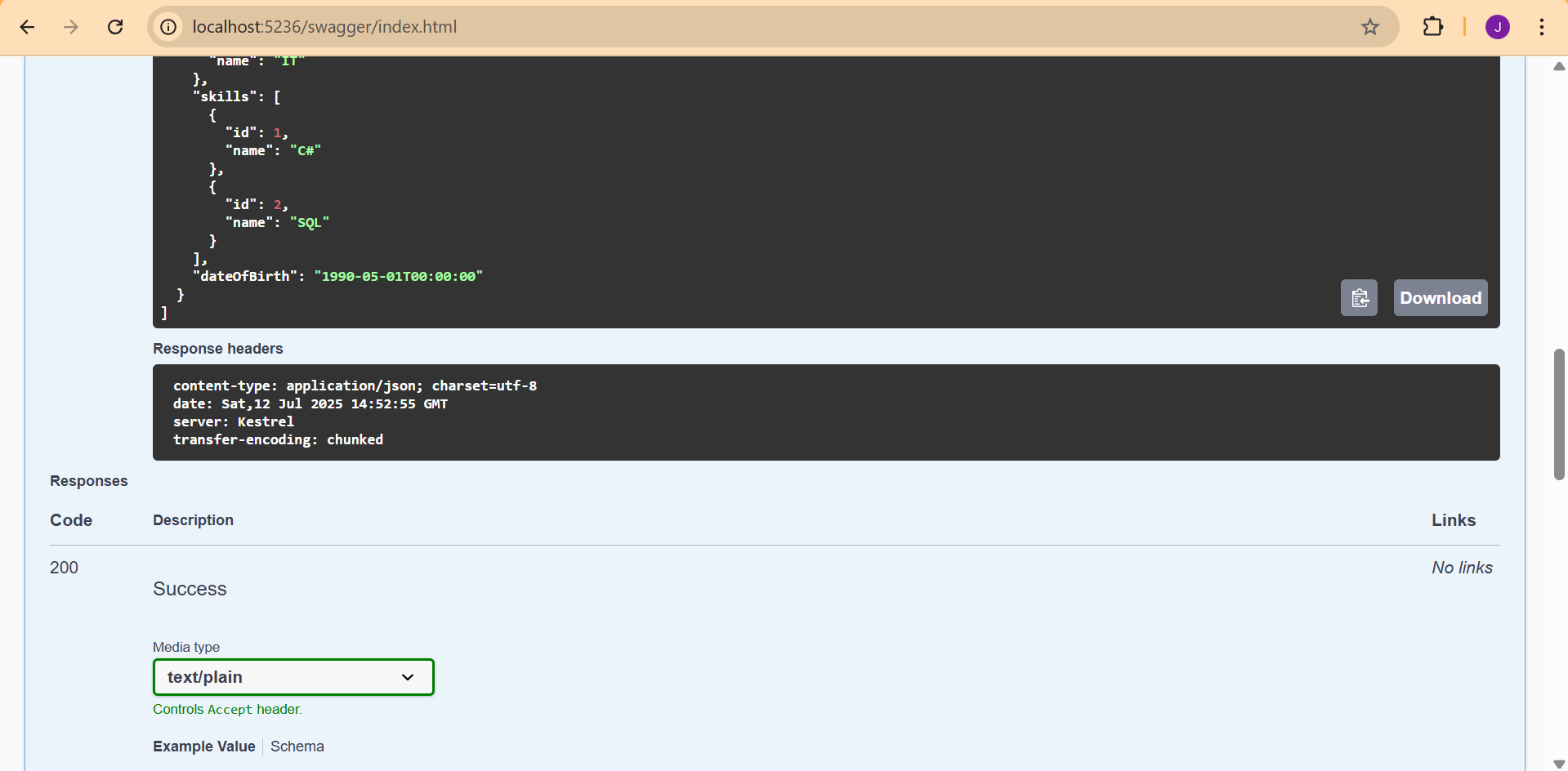
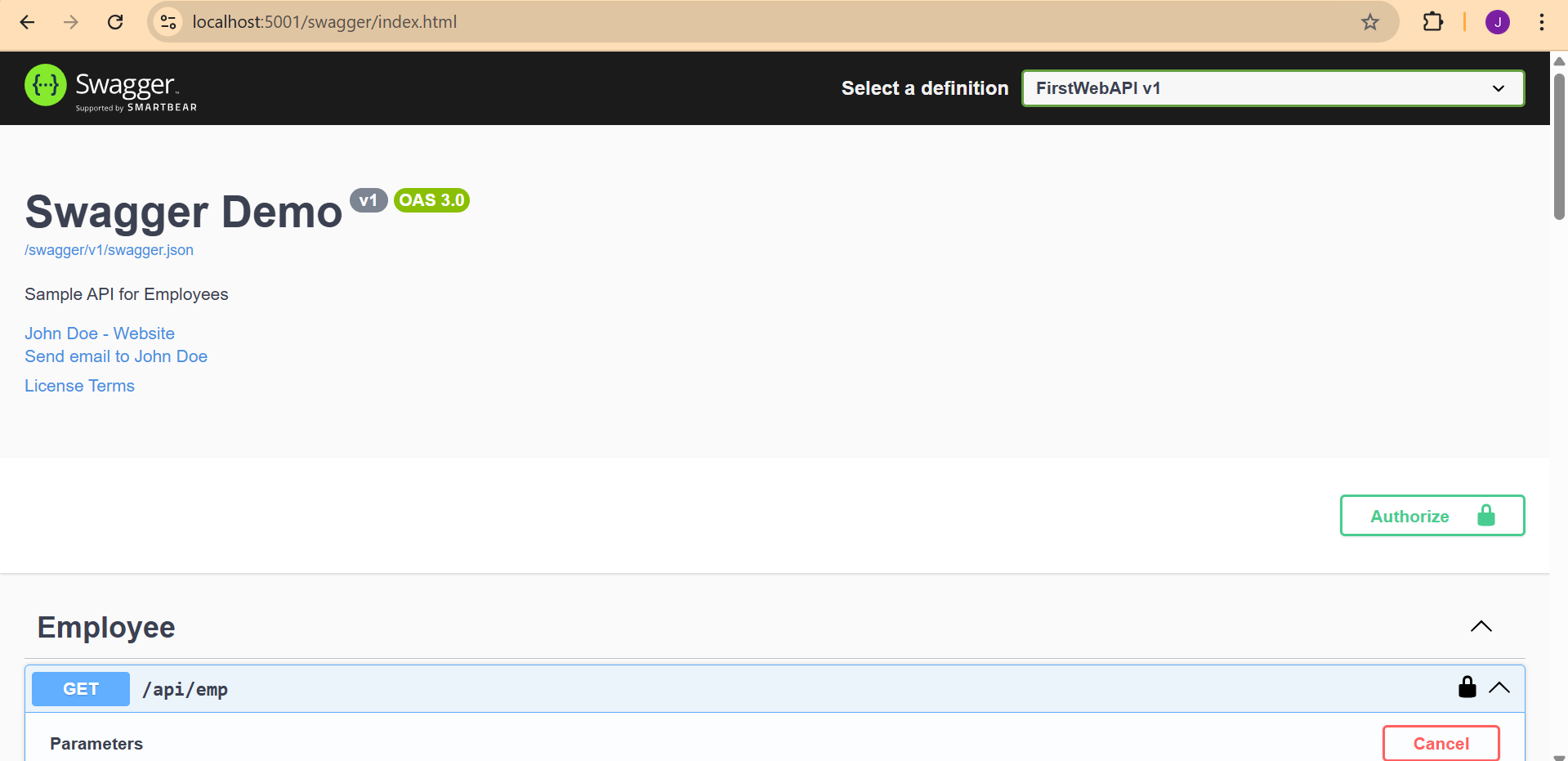
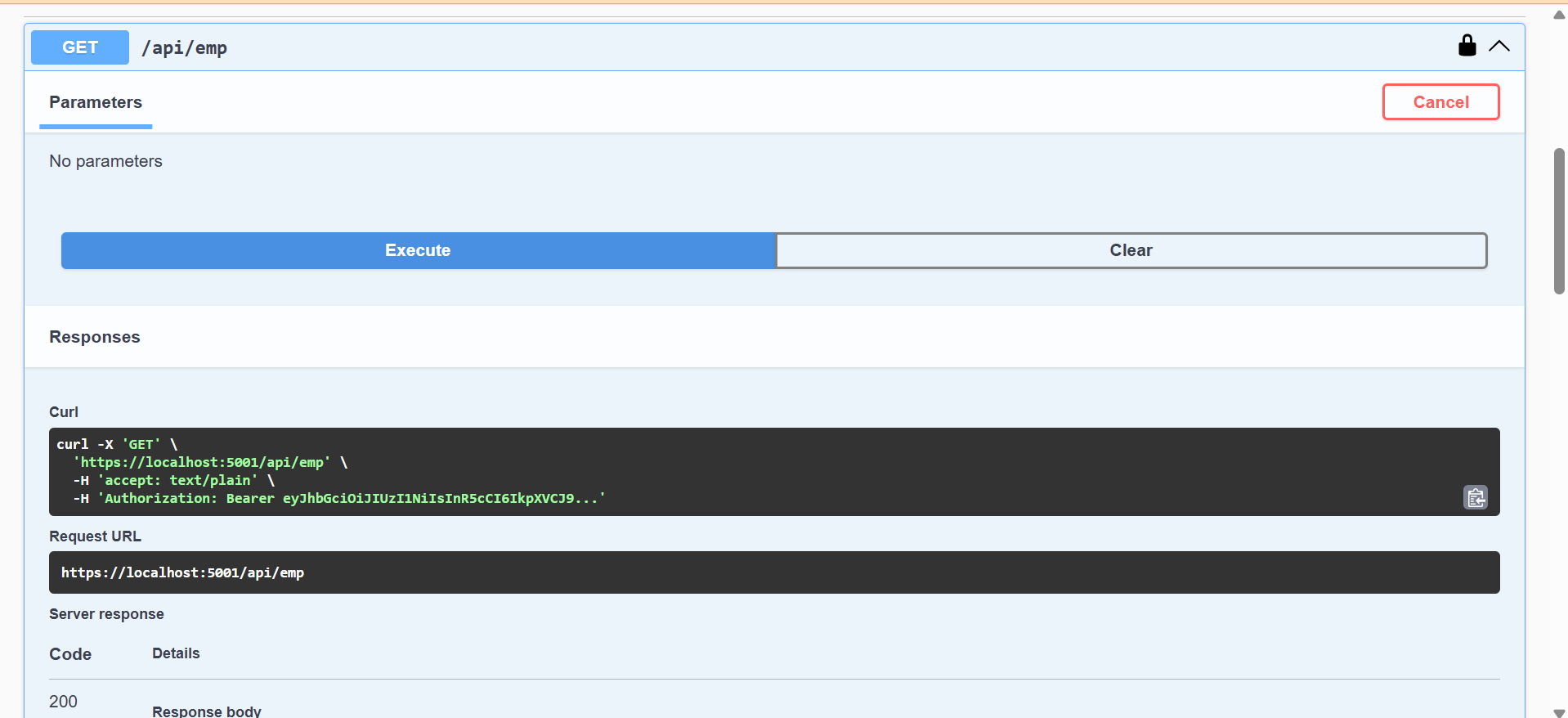
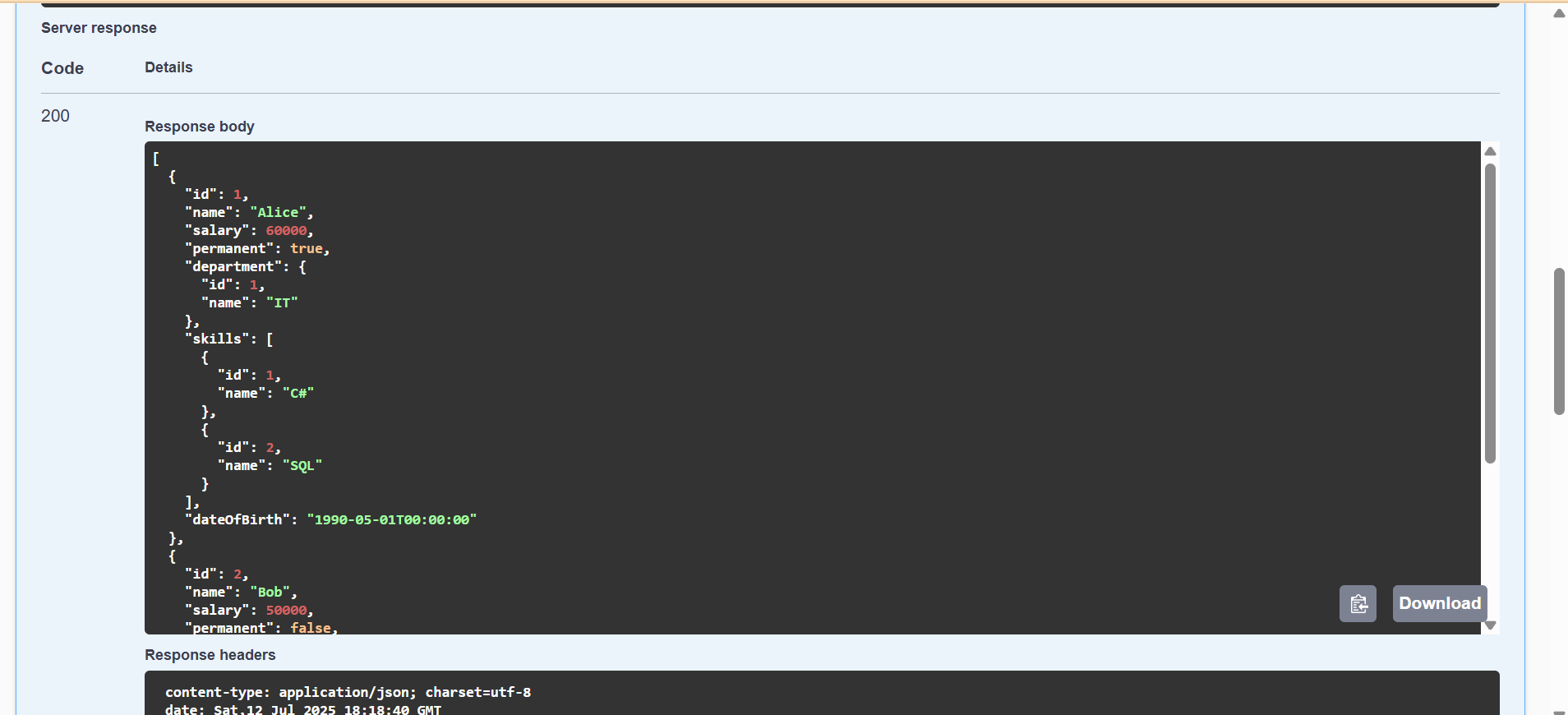
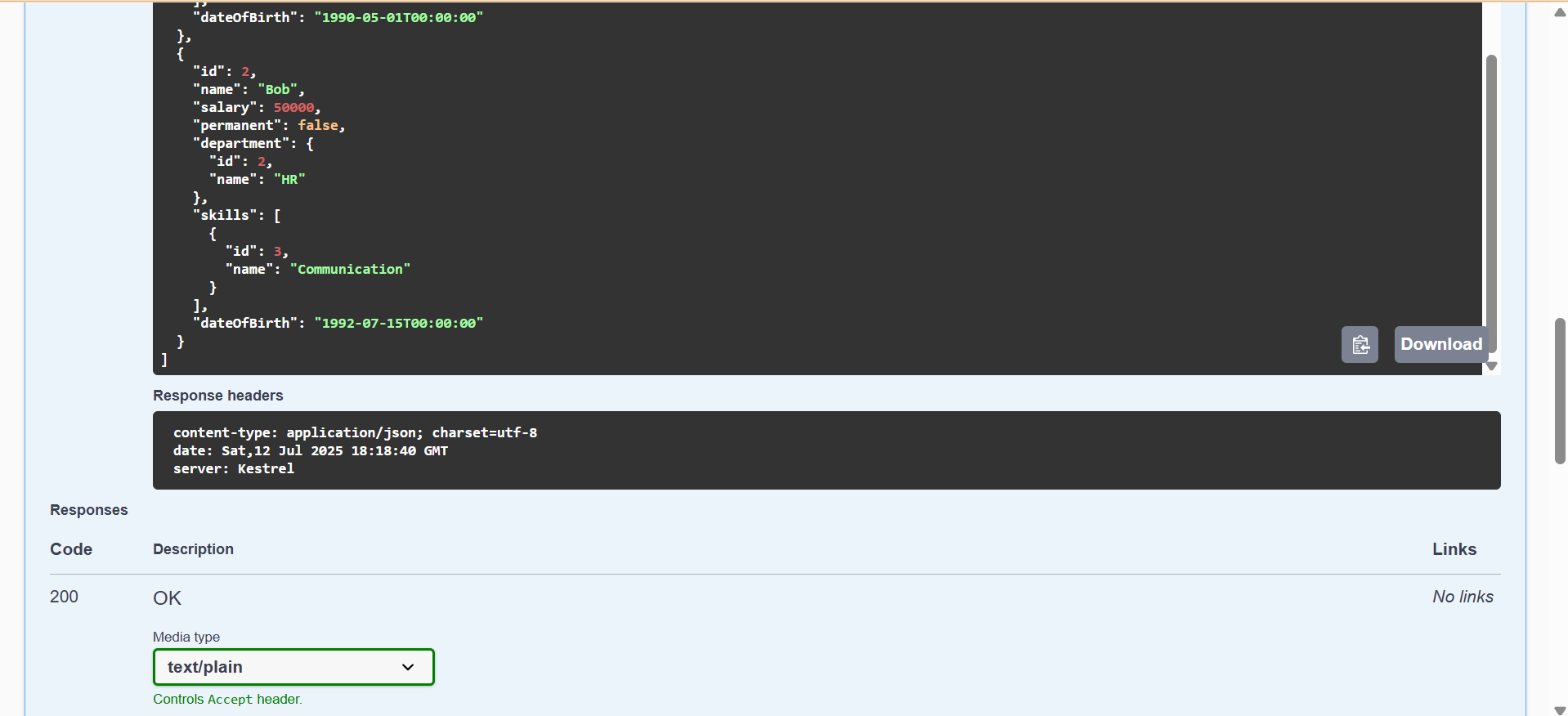
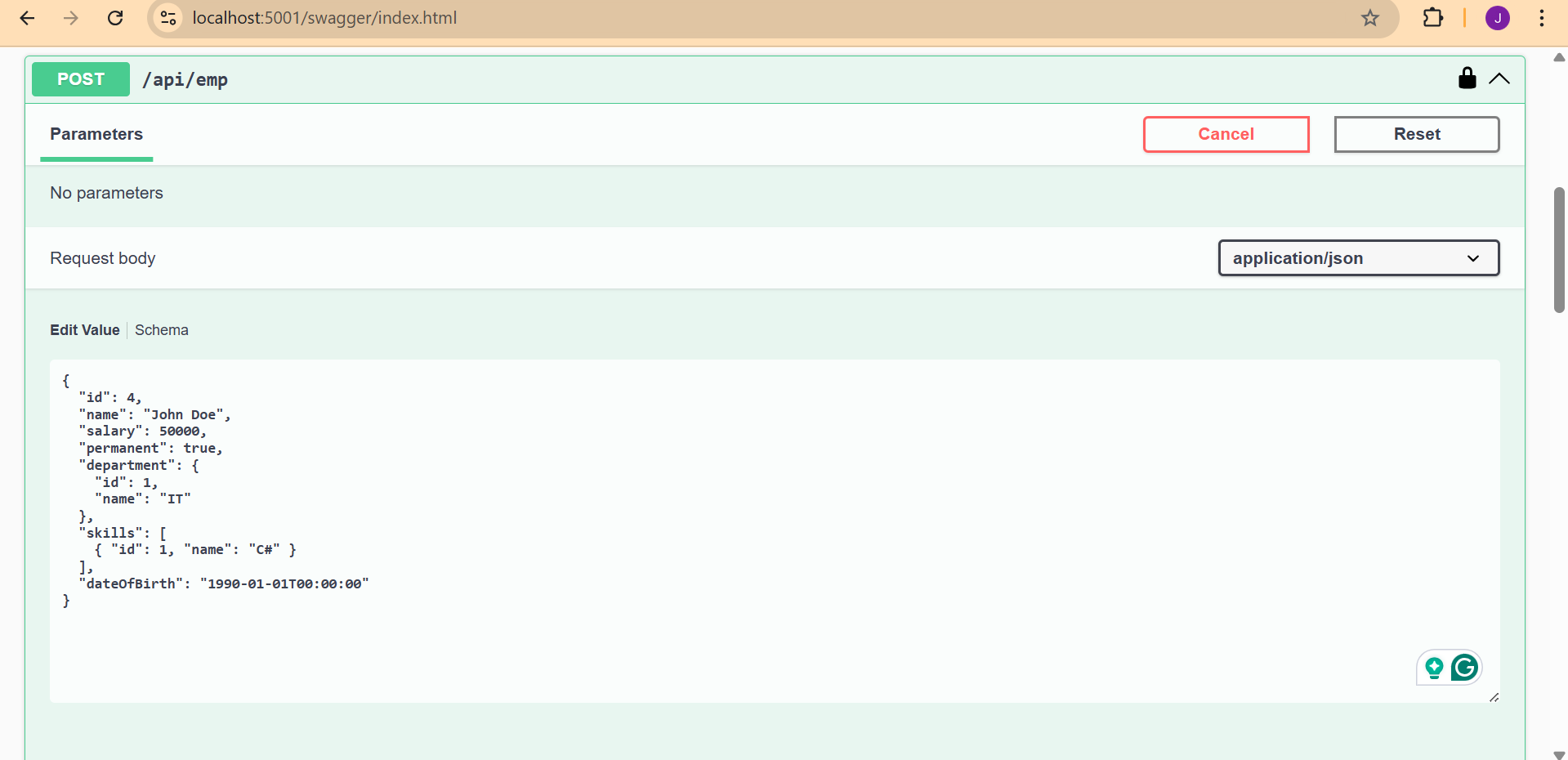
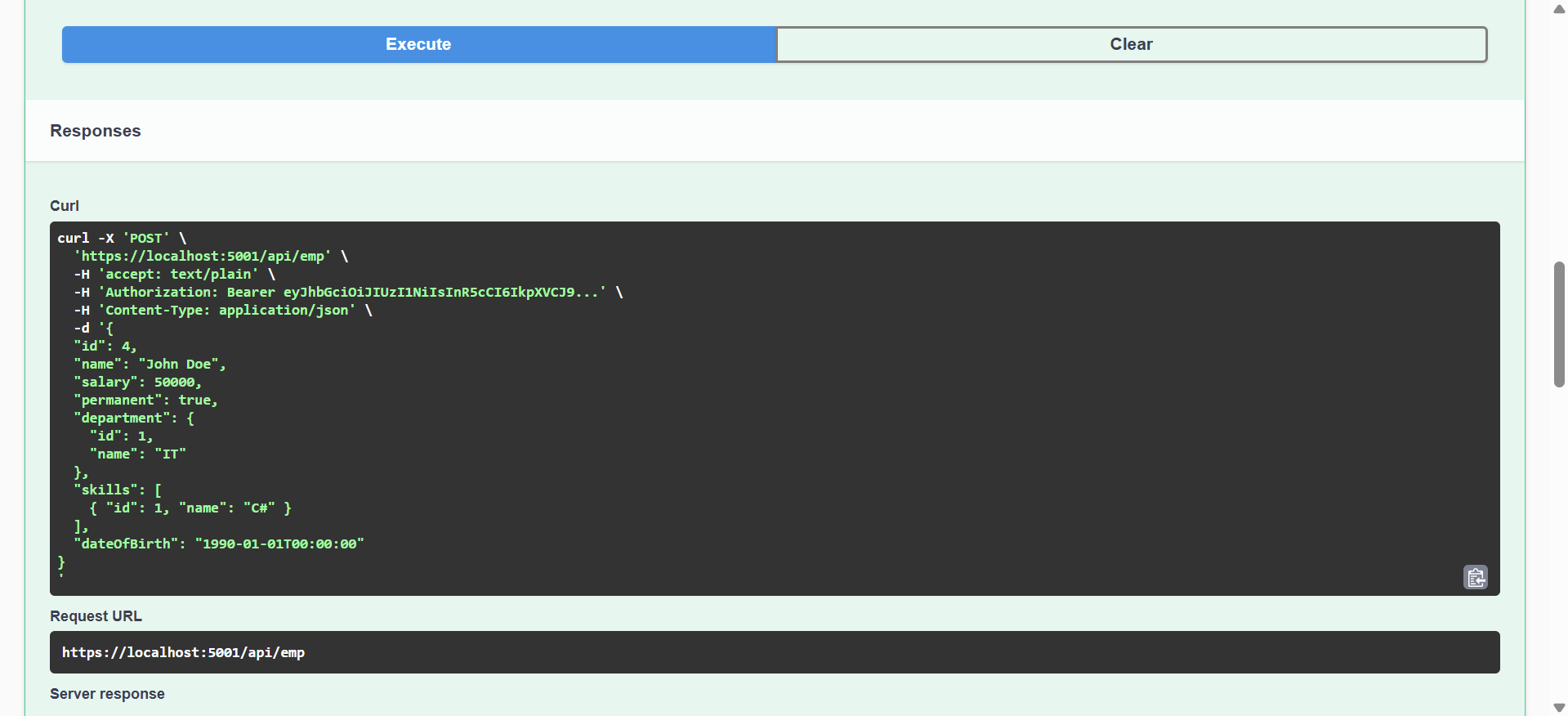
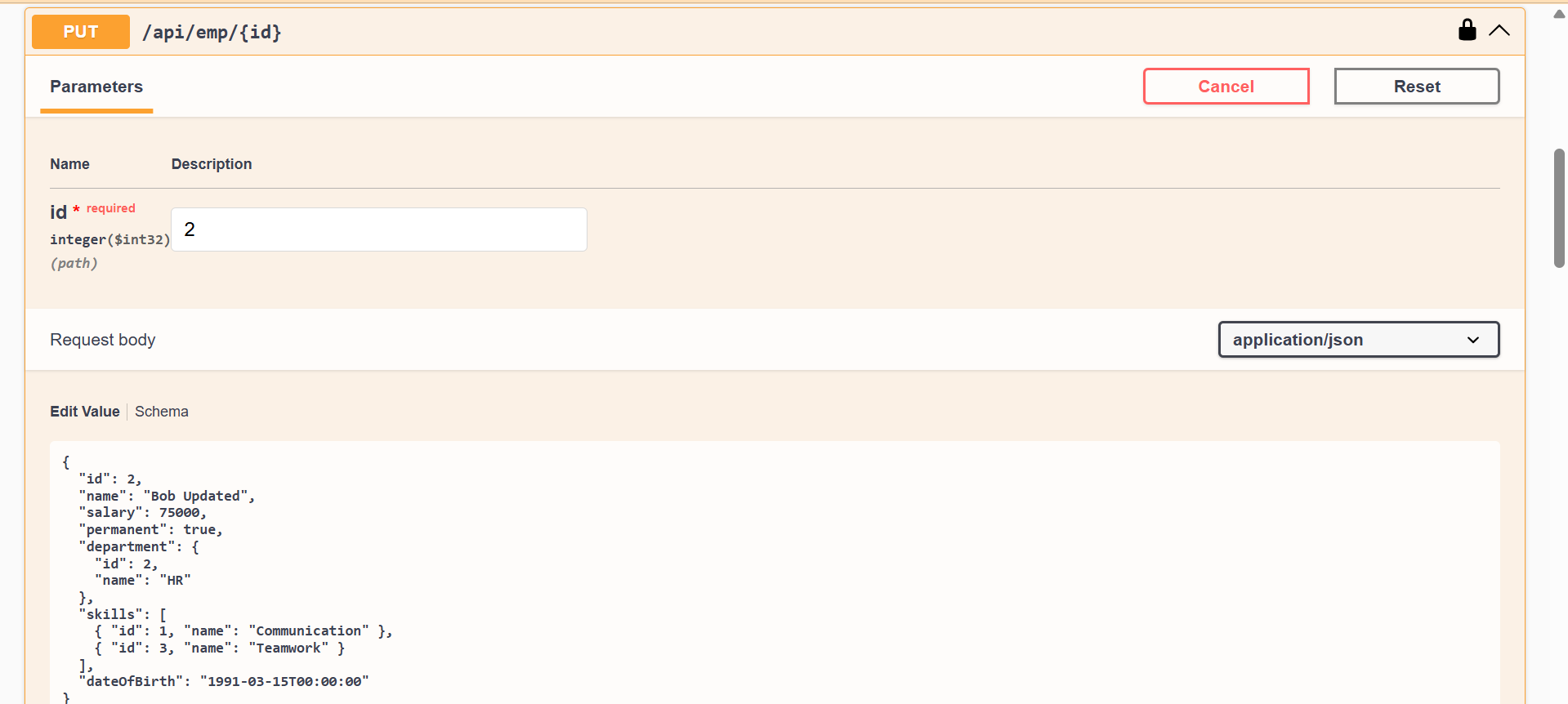
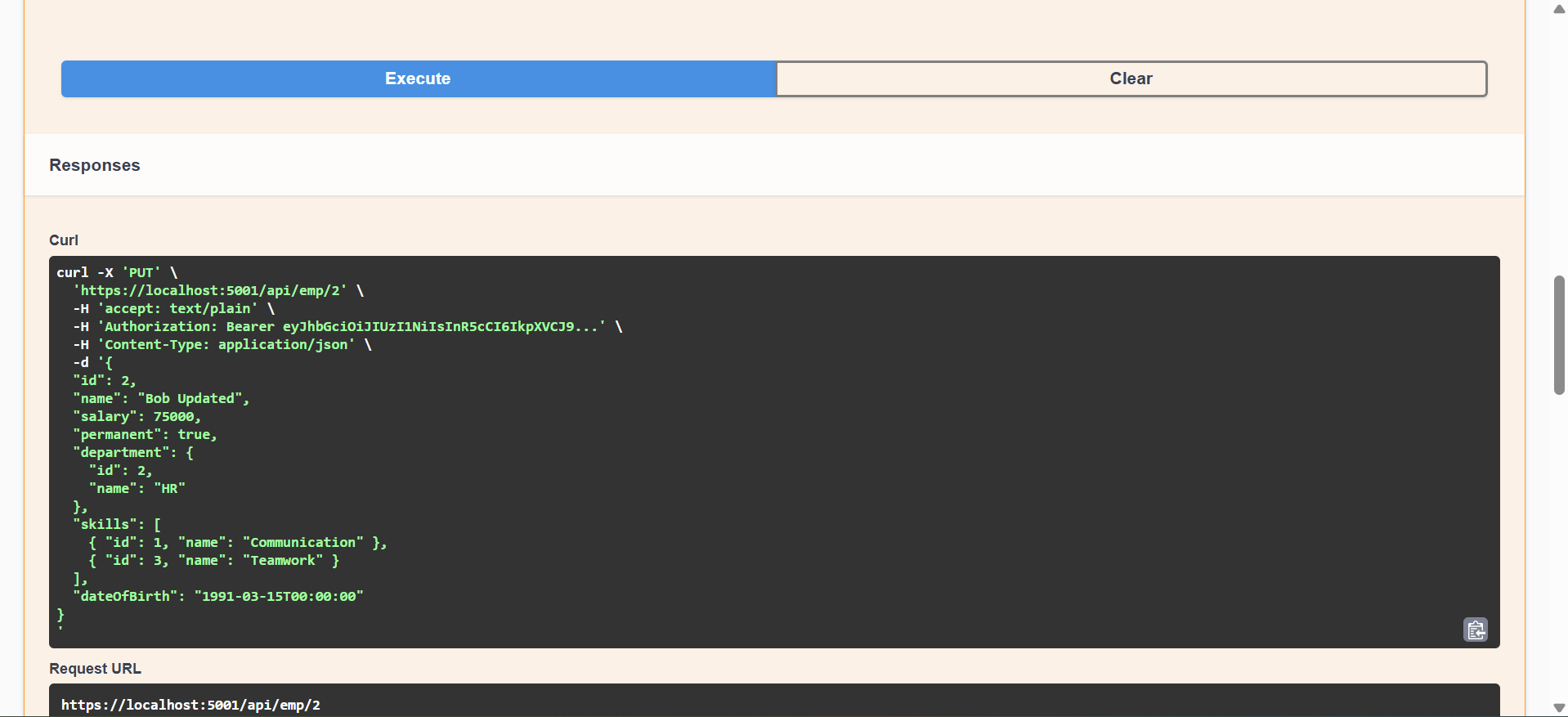
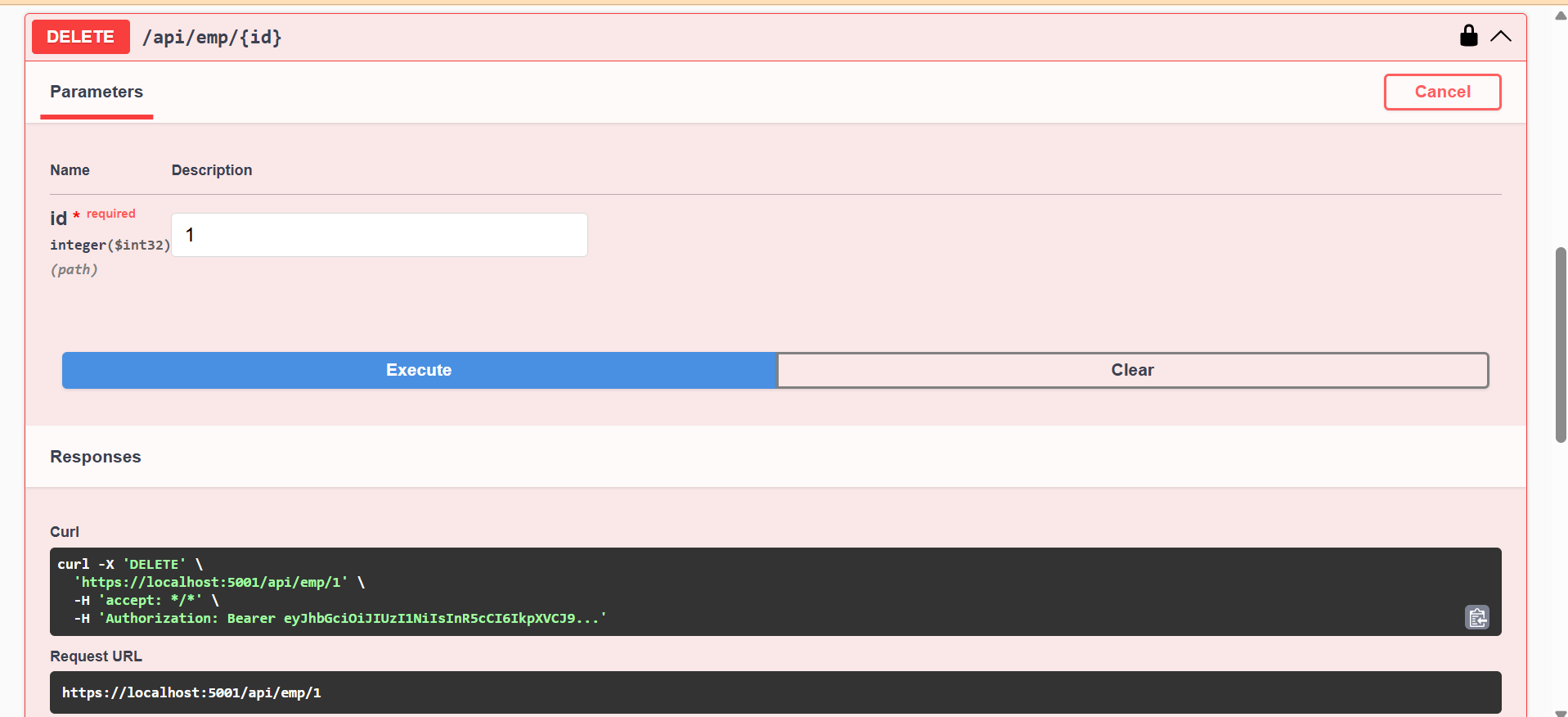
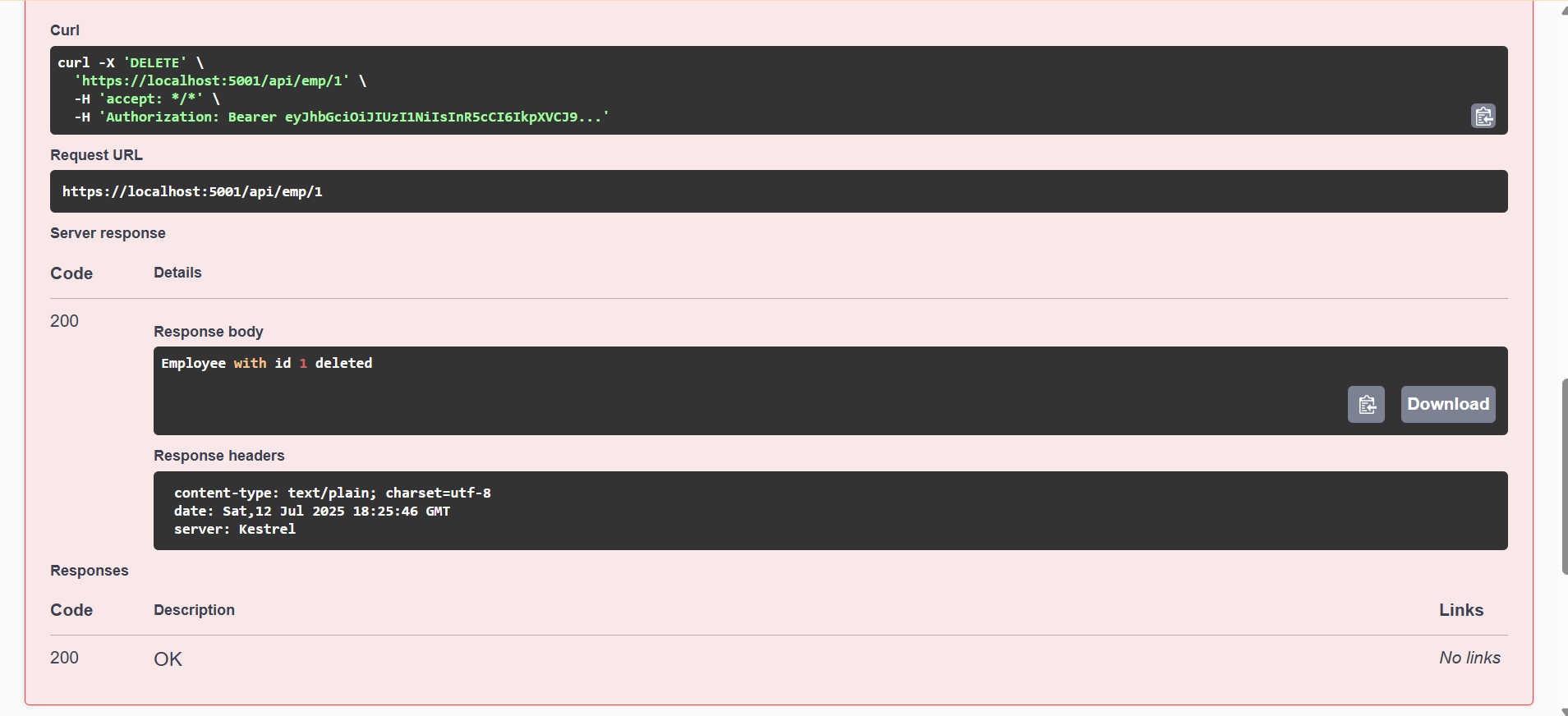
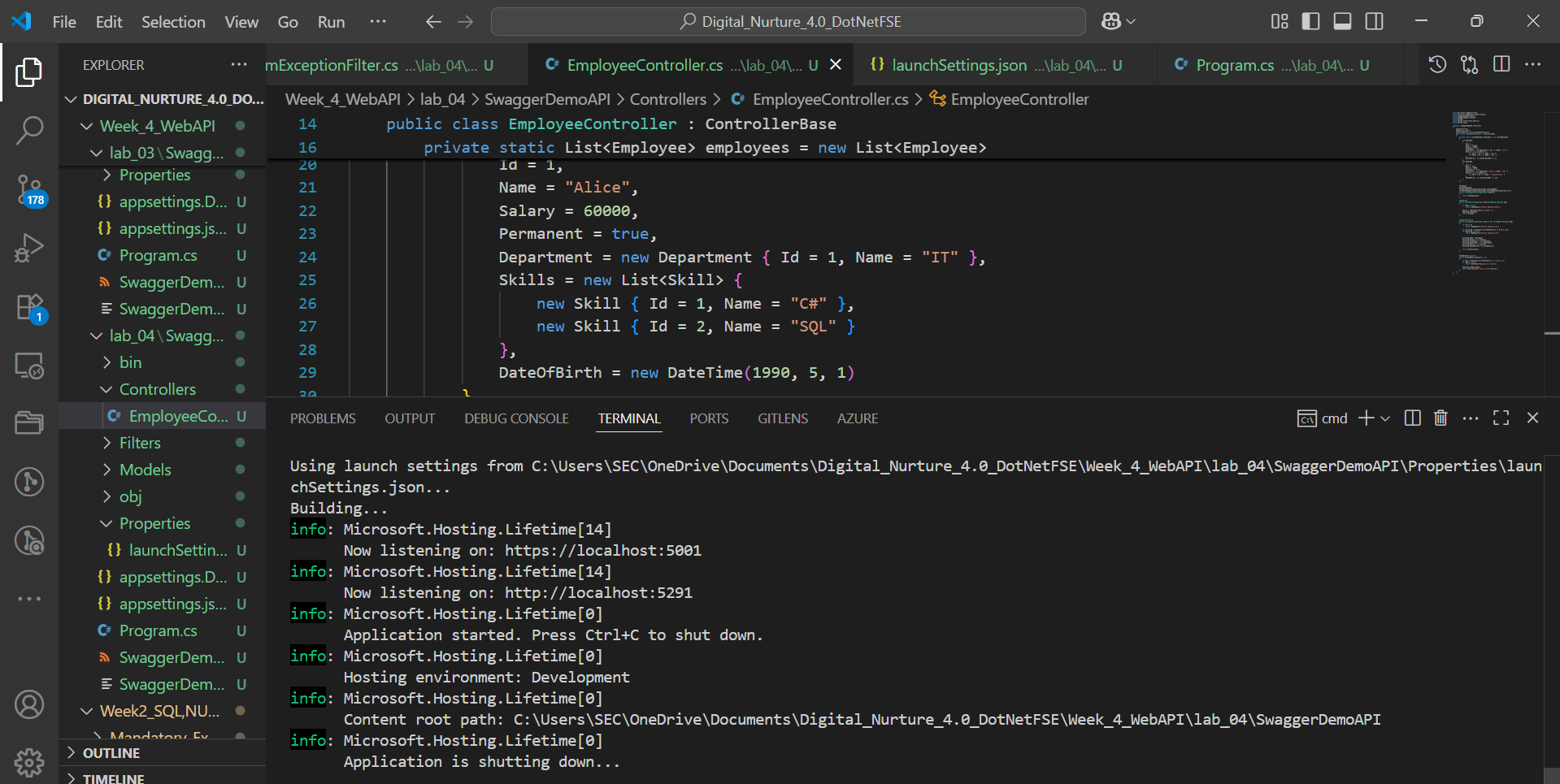
"ASPNETCORE\_ENVIRONMENT": "Development"

}

}

}

1. OUTPUT:  
     
     
     
     
     
     
     
     
     
     
   **2.LAB\_02  
     
   OUTPUT:**  
     
     
     
     
     
     
   **3.Modify the Controller name in the Route attribute of the Employee controller to ‘Emp’ and check its access thru POSTMAN**

  
  
  
**LAB 03:  
  
OUTPUT:**  
  
  
  
  
  
  
  
**LAB 04:  
  
OUTPUT:**  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**LAB 05:  
  
OUTPUT:**