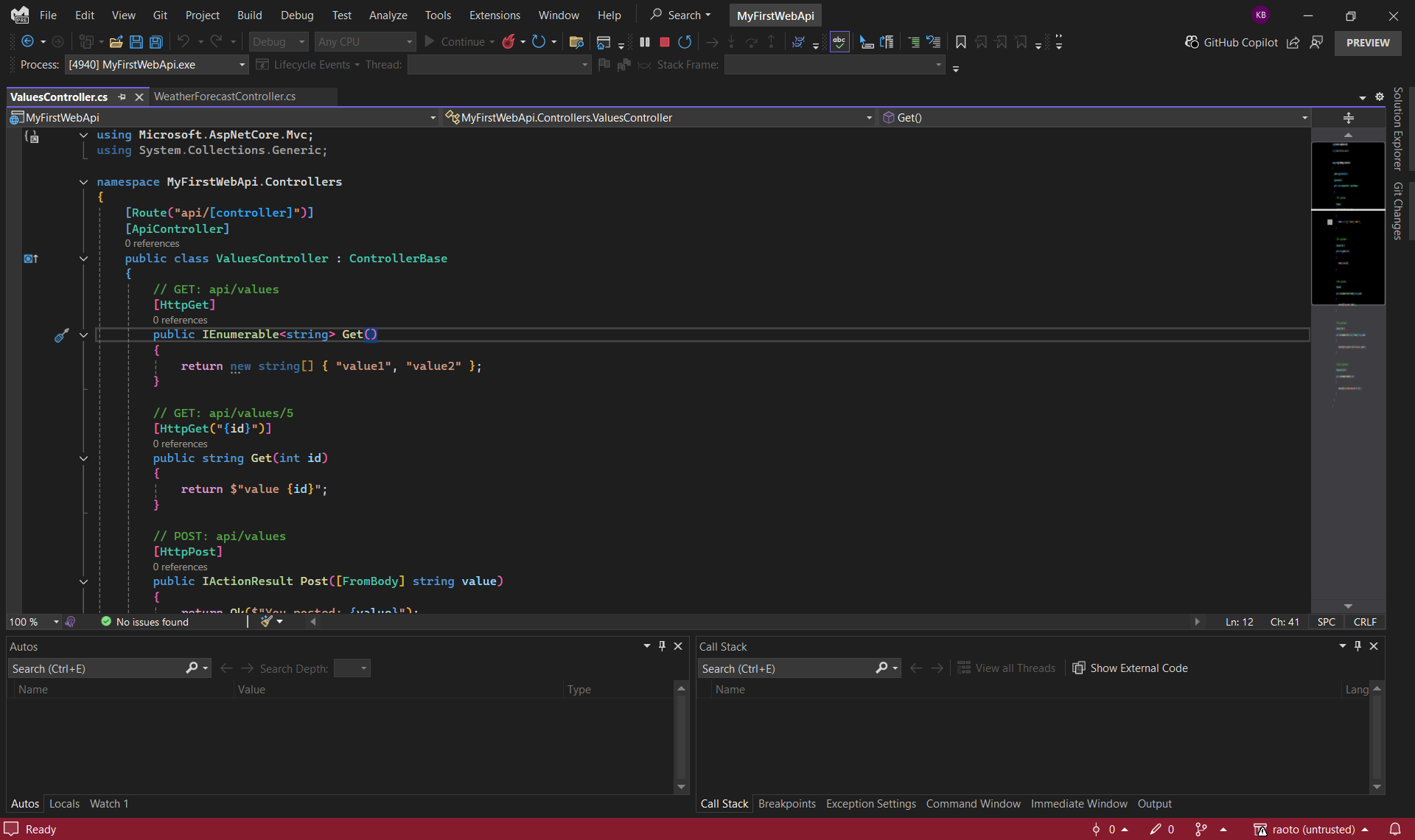
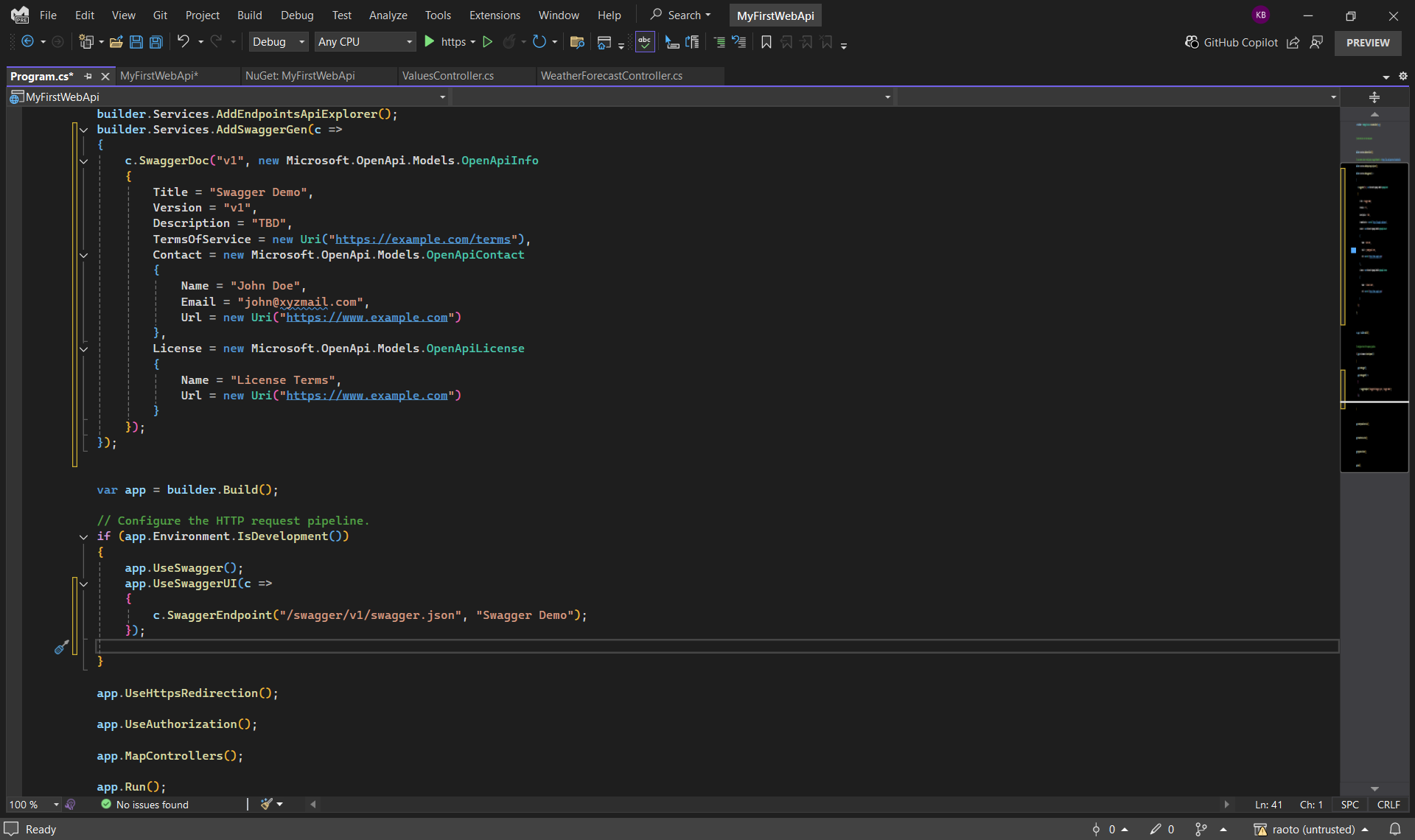
# **Web API and Kafka Hands-On Lab**

# **Document 1 Swagger and Web API Setup**

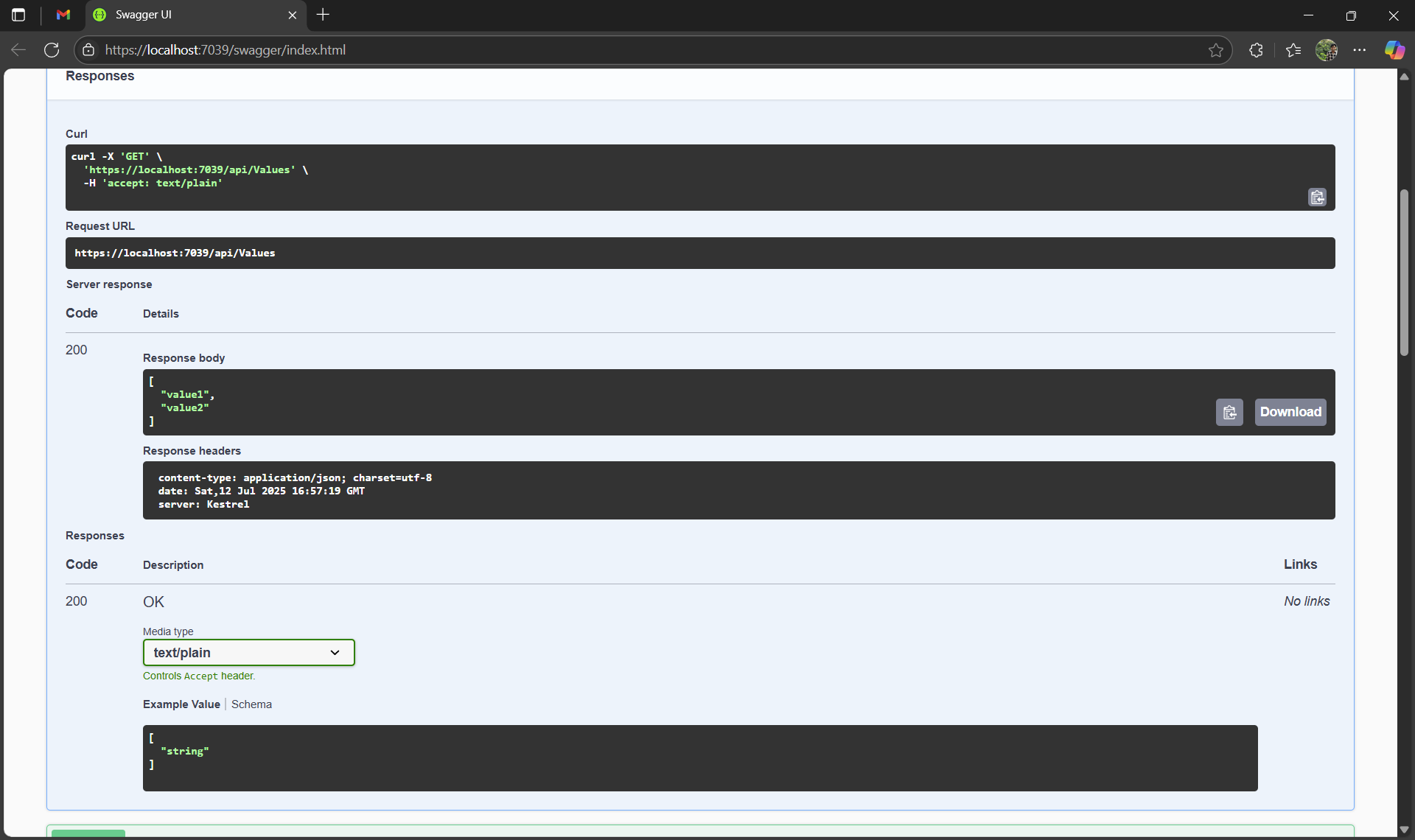
ValuesController code :



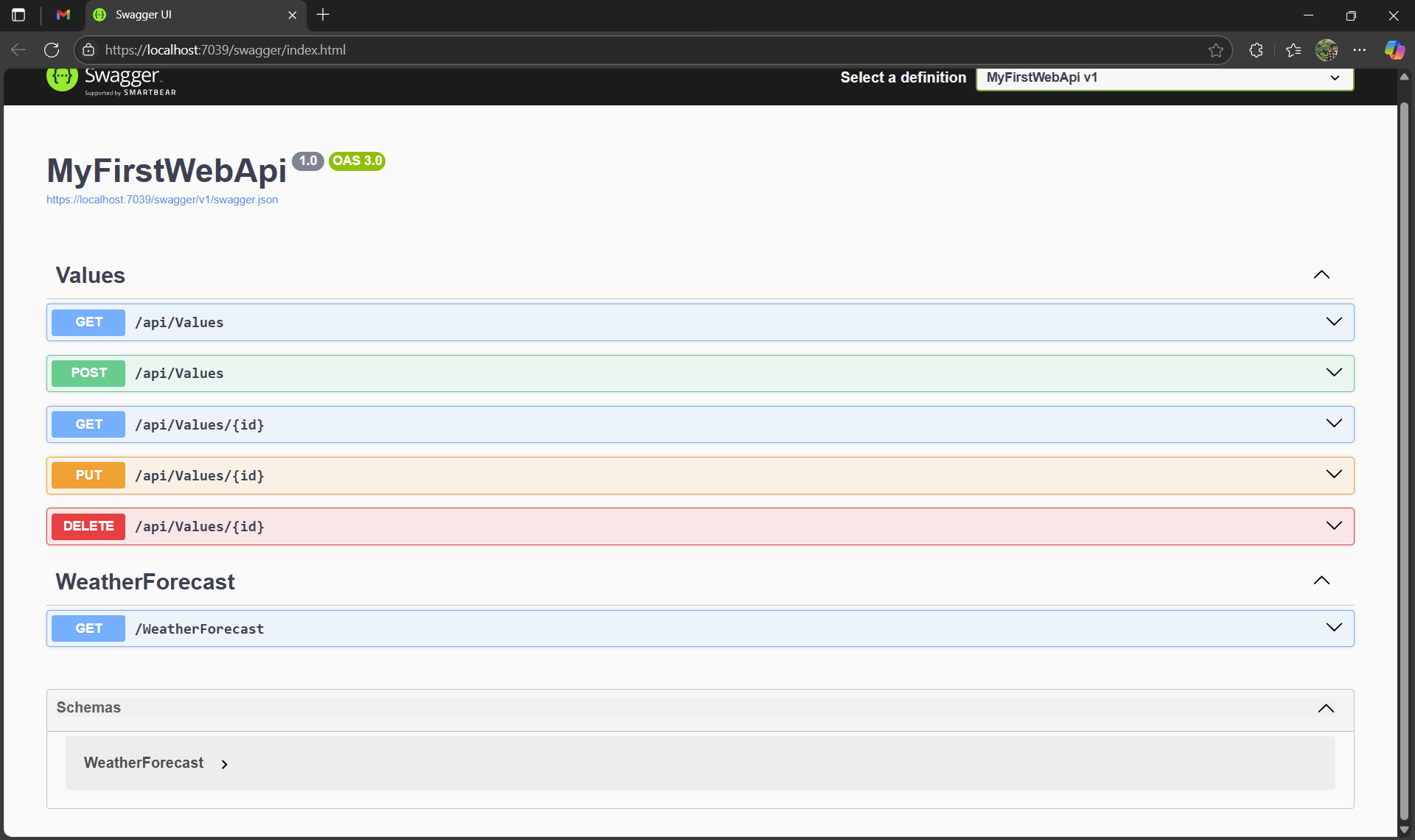
Program.cs code:



**Swagger - GET Method Response**



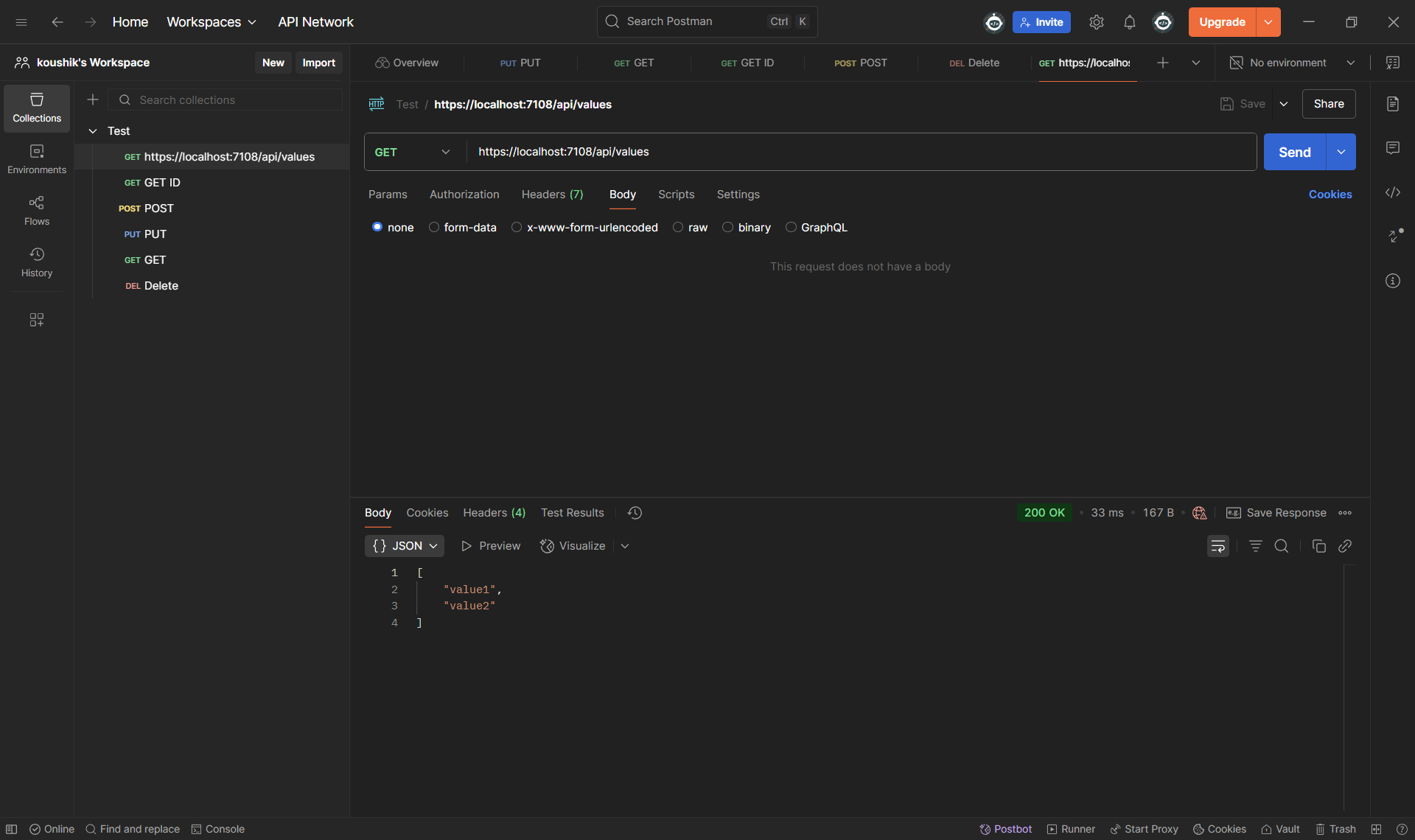
**Swagger UI showing the ValuesController listed**



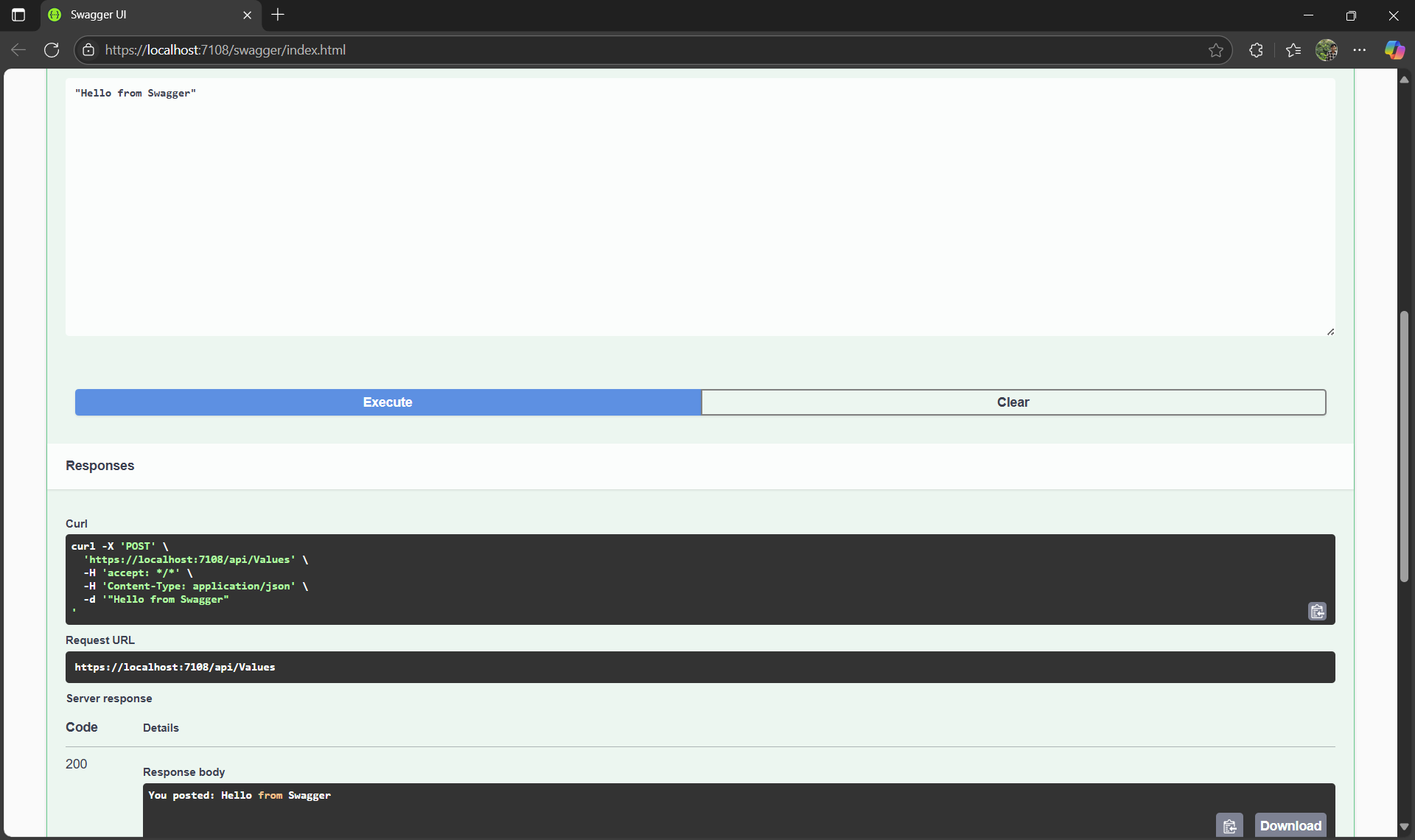
## 

## **Document 2: Testing API with Postman**

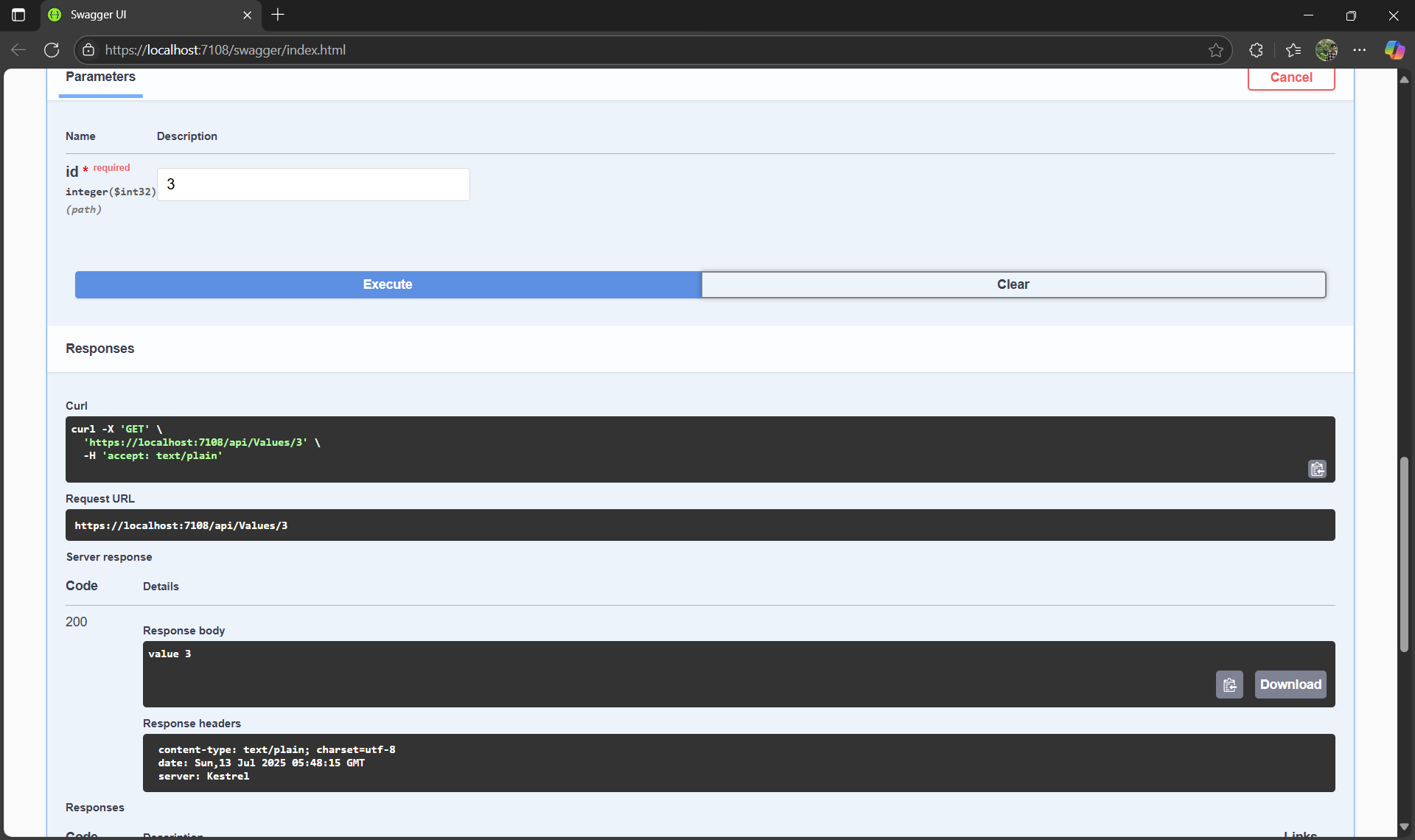
Postman - GET Request and Response



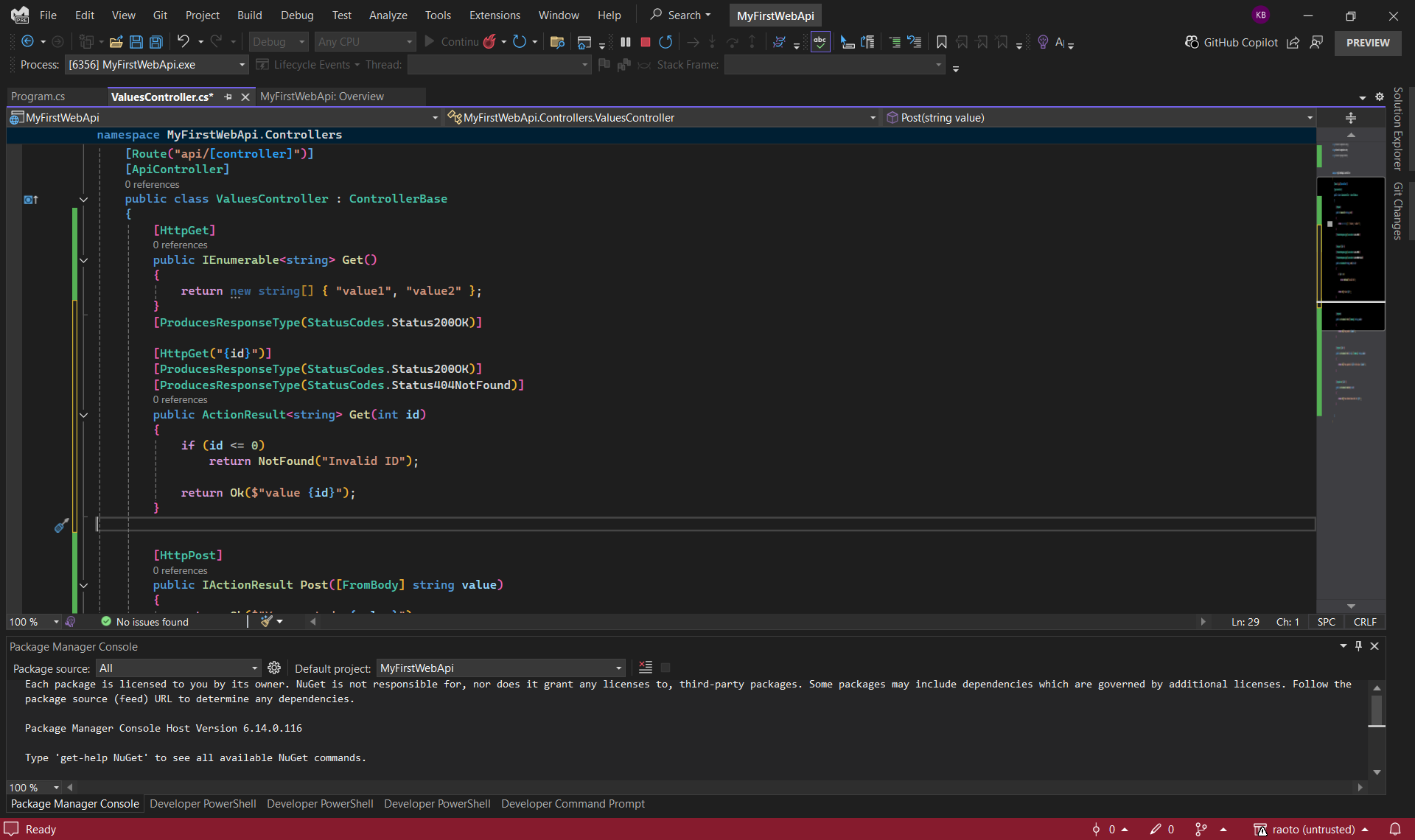
Swagger - POST Request to /api/Values with Response:



Swagger - GET Request to /api/Values/3:

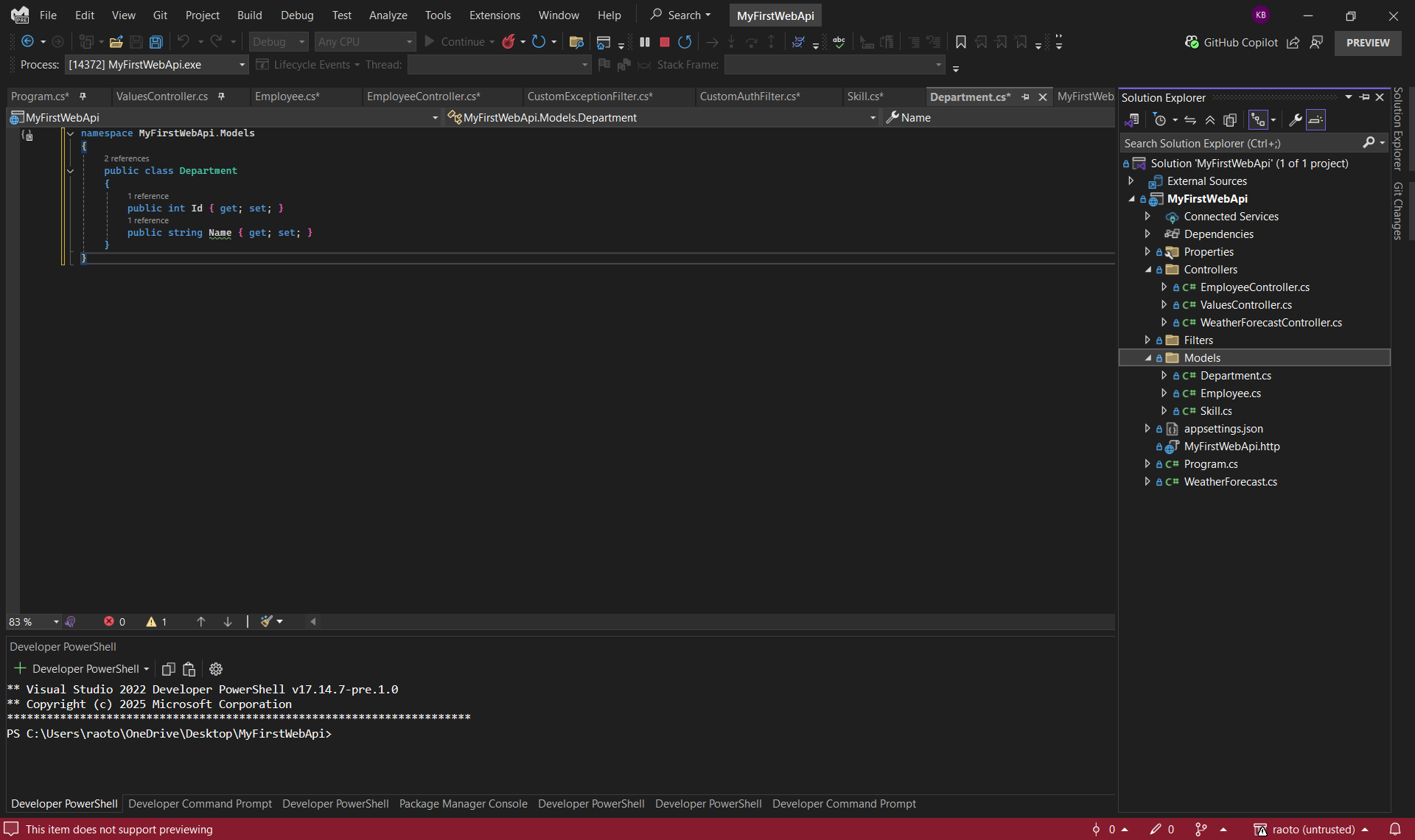


Visual Studio - ValuesController.cs Code with CRUD Methods:

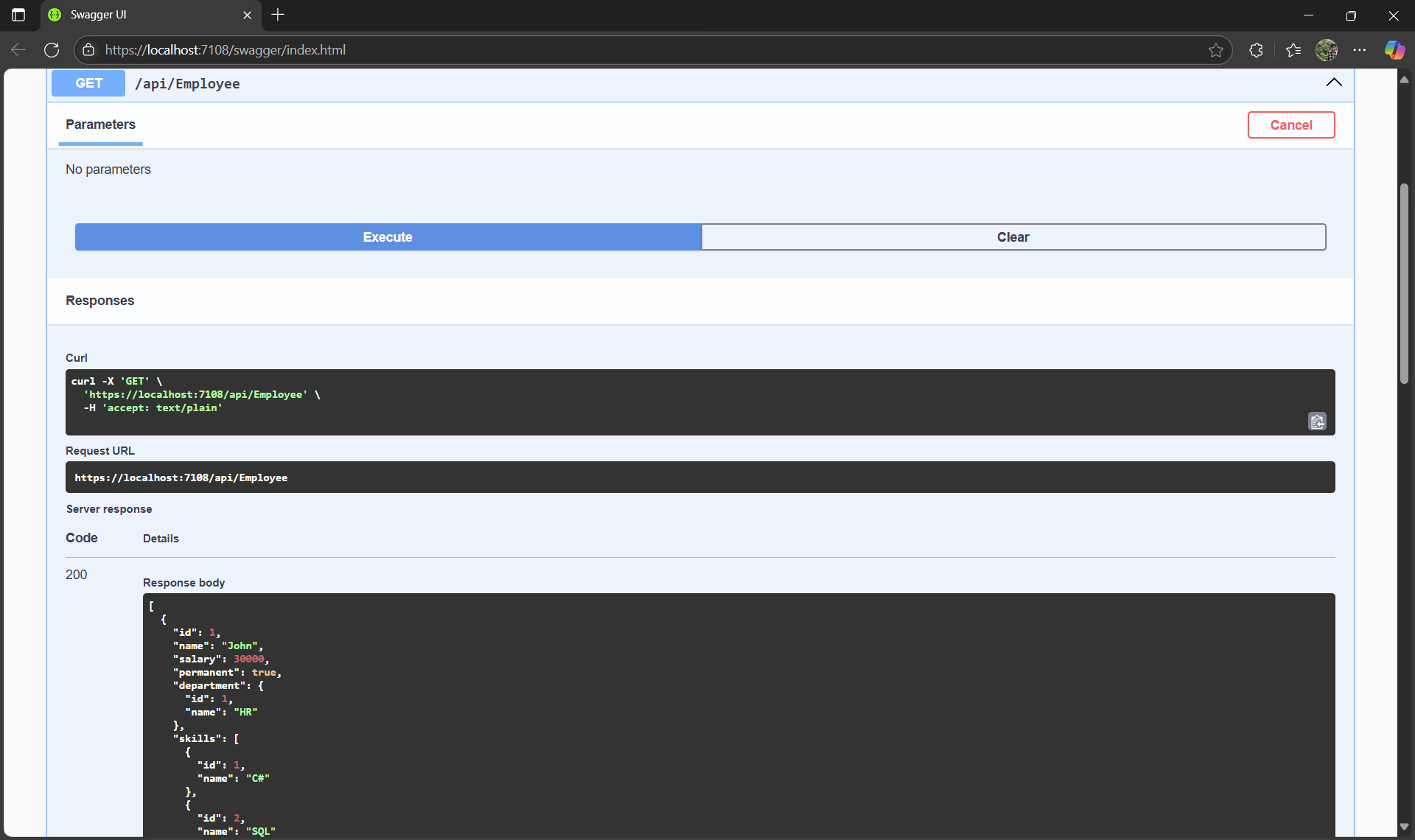


## Document 3: Action Filter and Exception Handling

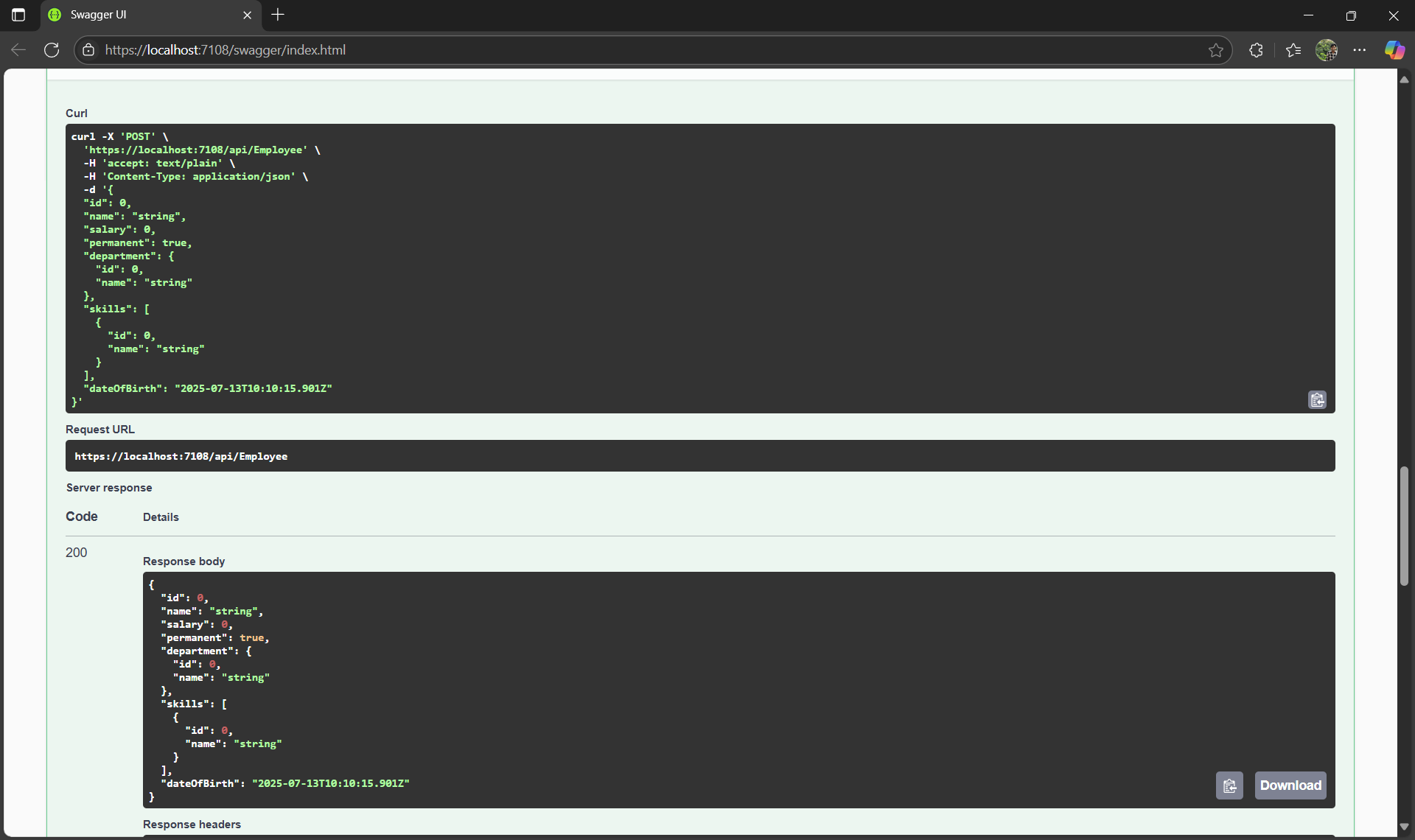
Created a class named Employee.cs with properties:



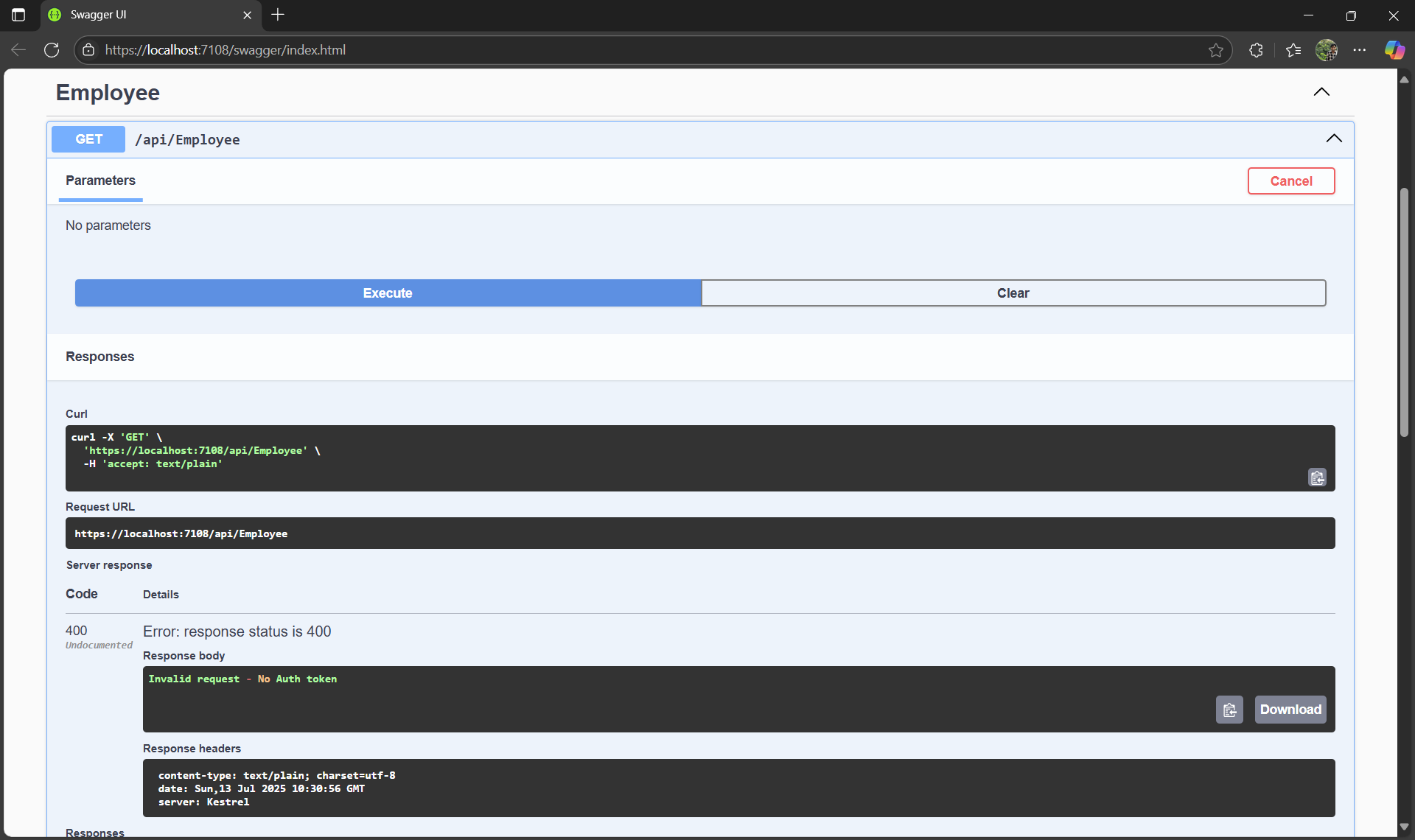
## Swagger - GET /api/Employee Showing Employee List:



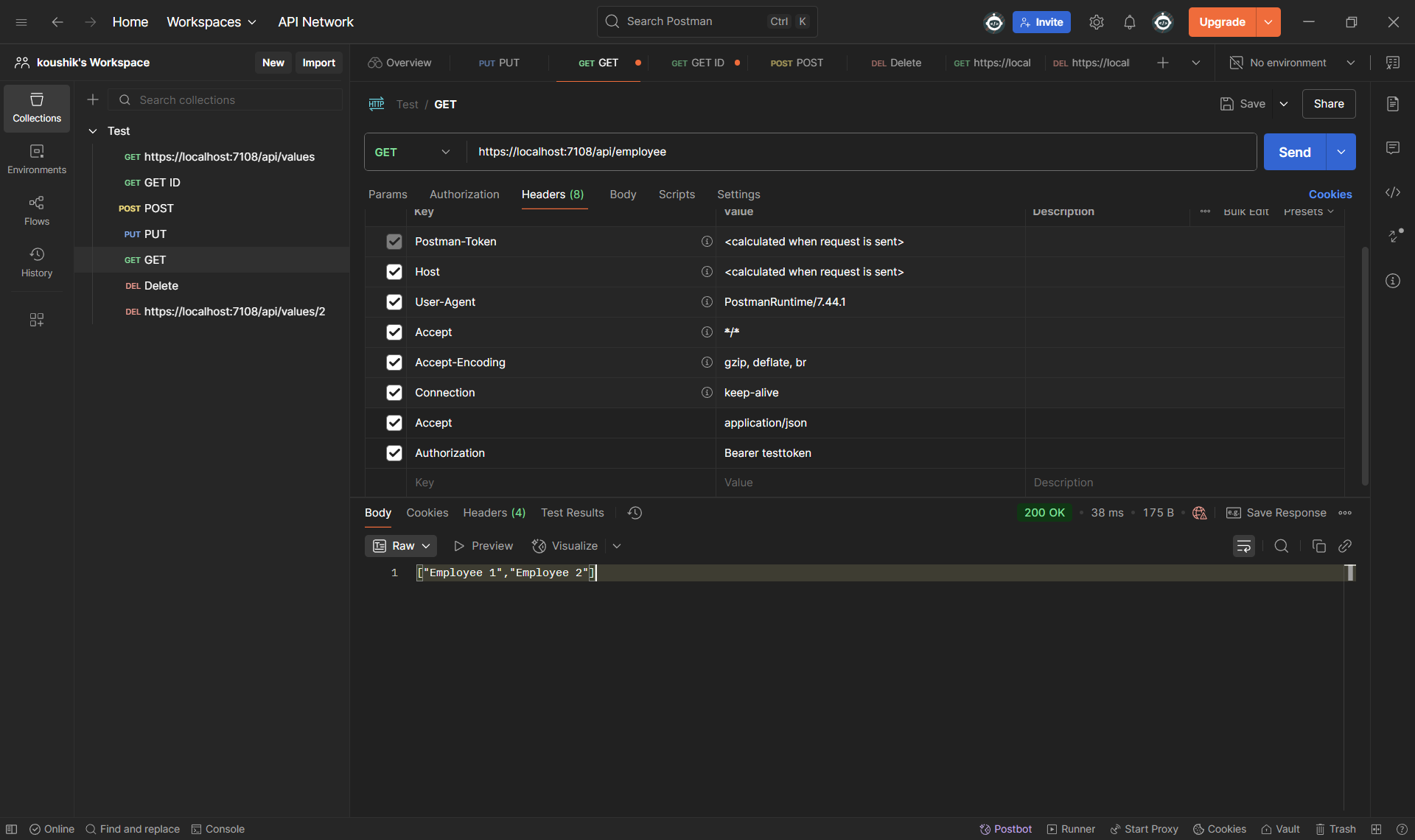
**Swagger - post /api/Employee:**



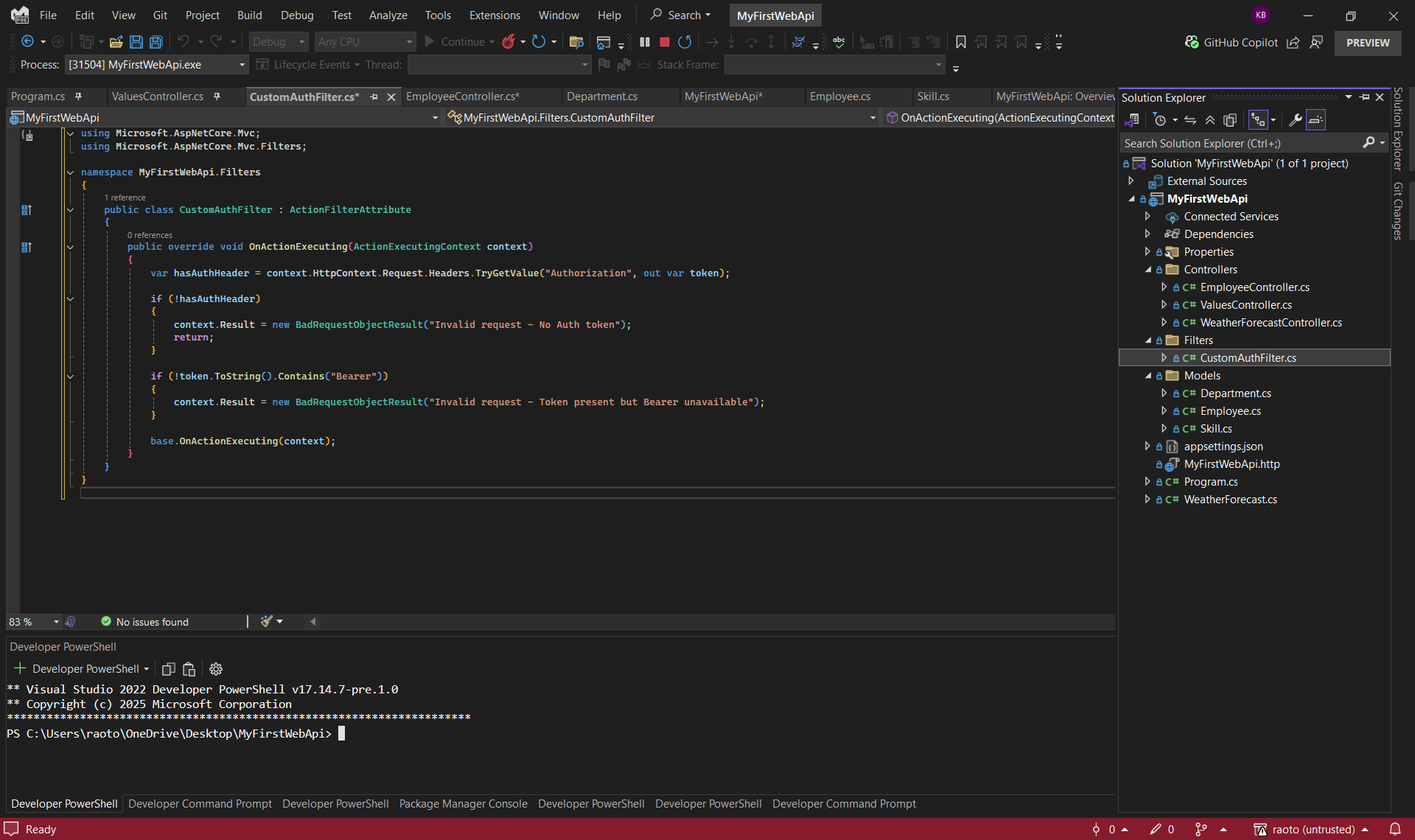
## Swagger - Authorization Header Missing (CustomAuthFilter):



## Postman - GET /api/Employee Returning List of Employee Names:

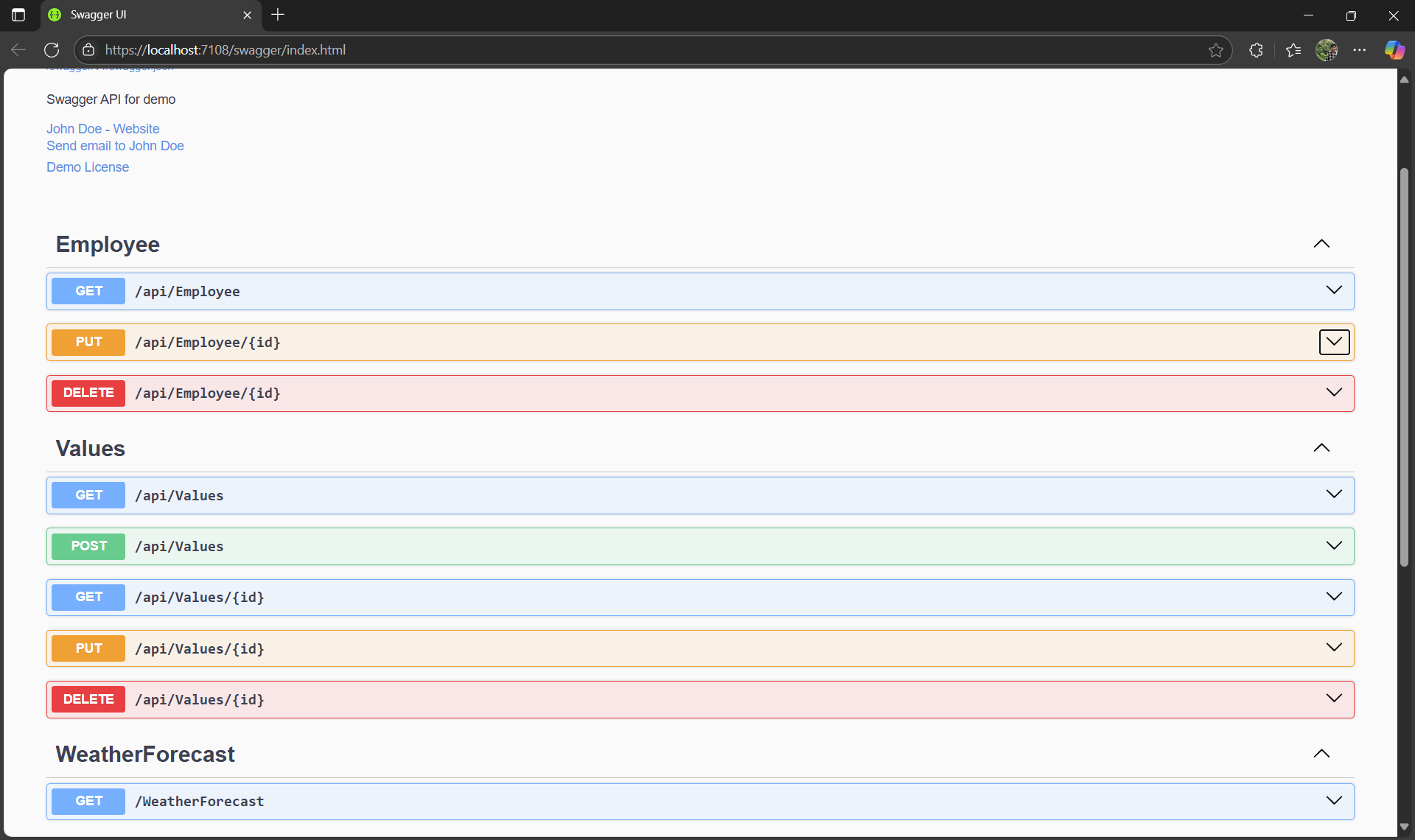


## Visual Studio - CustomAuthFilter.cs Code Implementation:

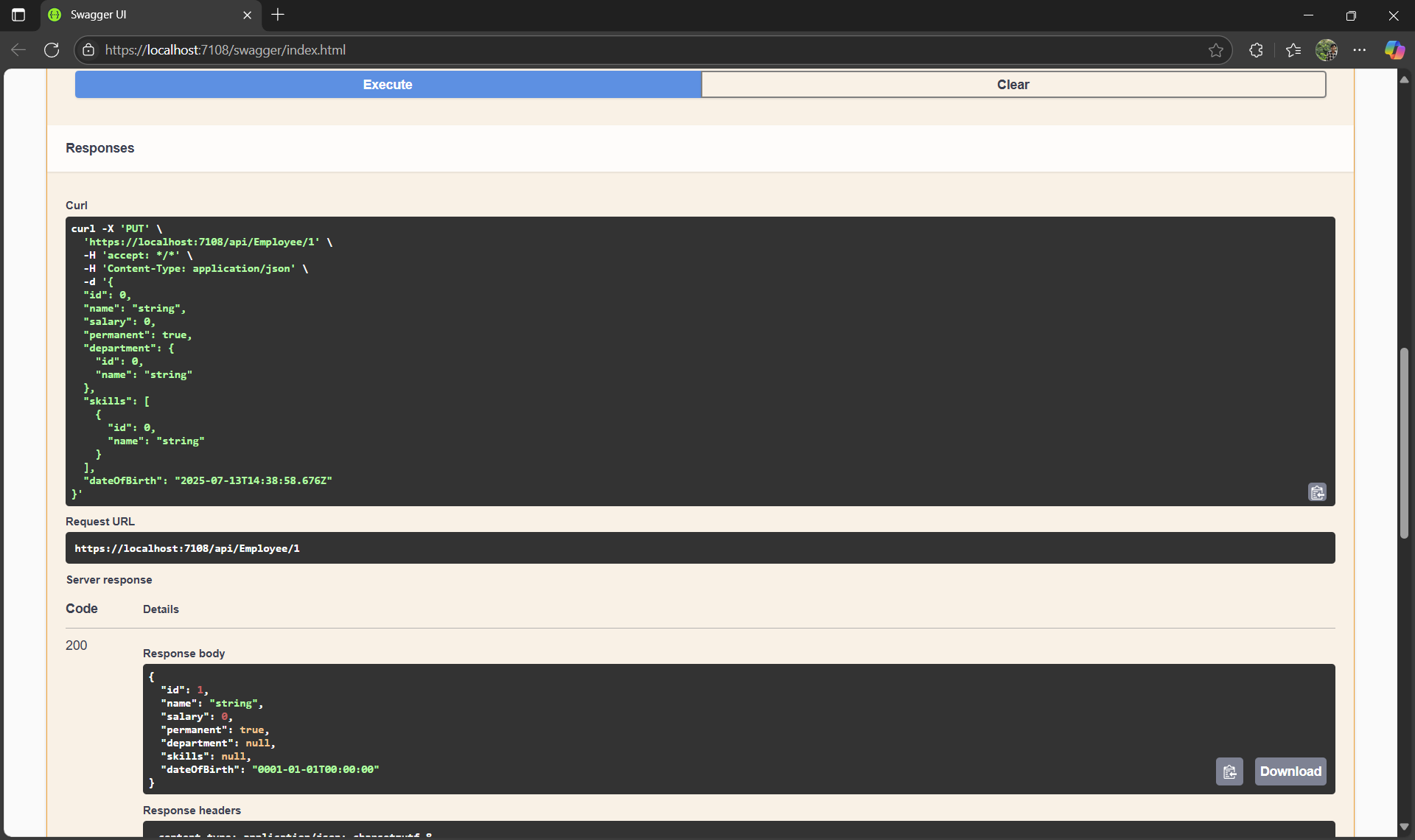


## Document 4: Web API CRUD Operation – Put and Delete

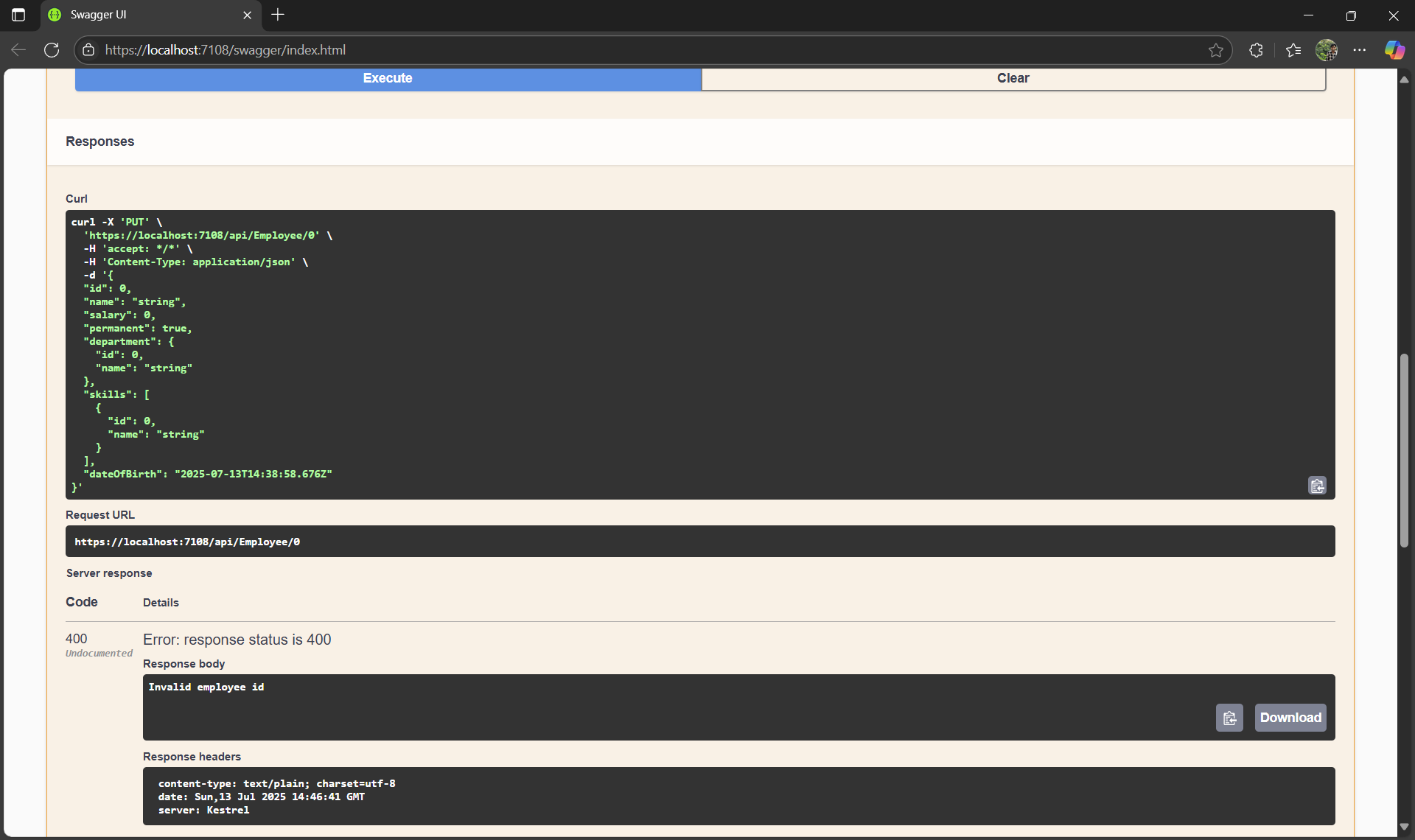
Swagger UI - CRUD Endpoints for Employee and Values Controllers (GET, POST, PUT, DELETE)



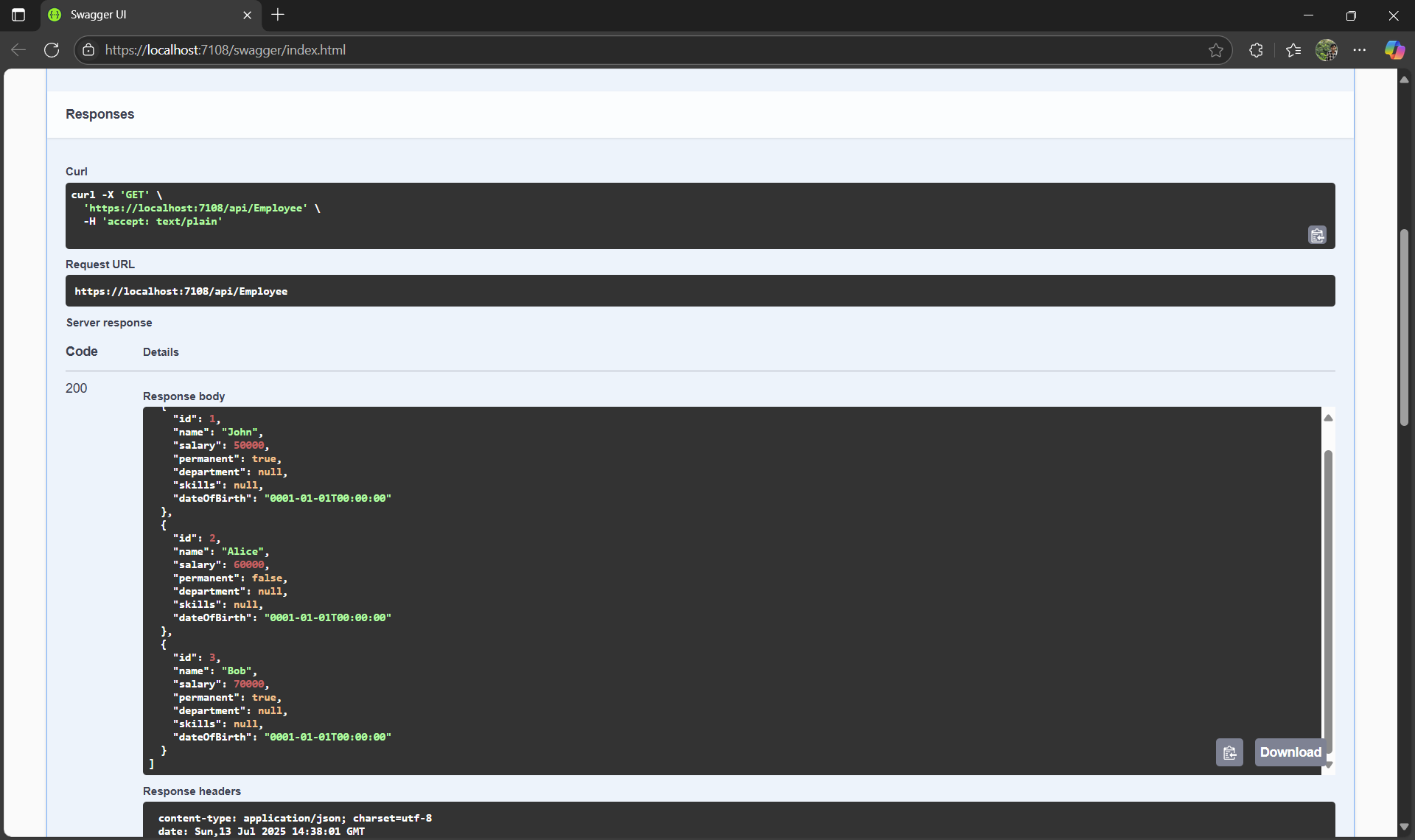
Swagger - PUT Employee with Valid ID (200 OK)



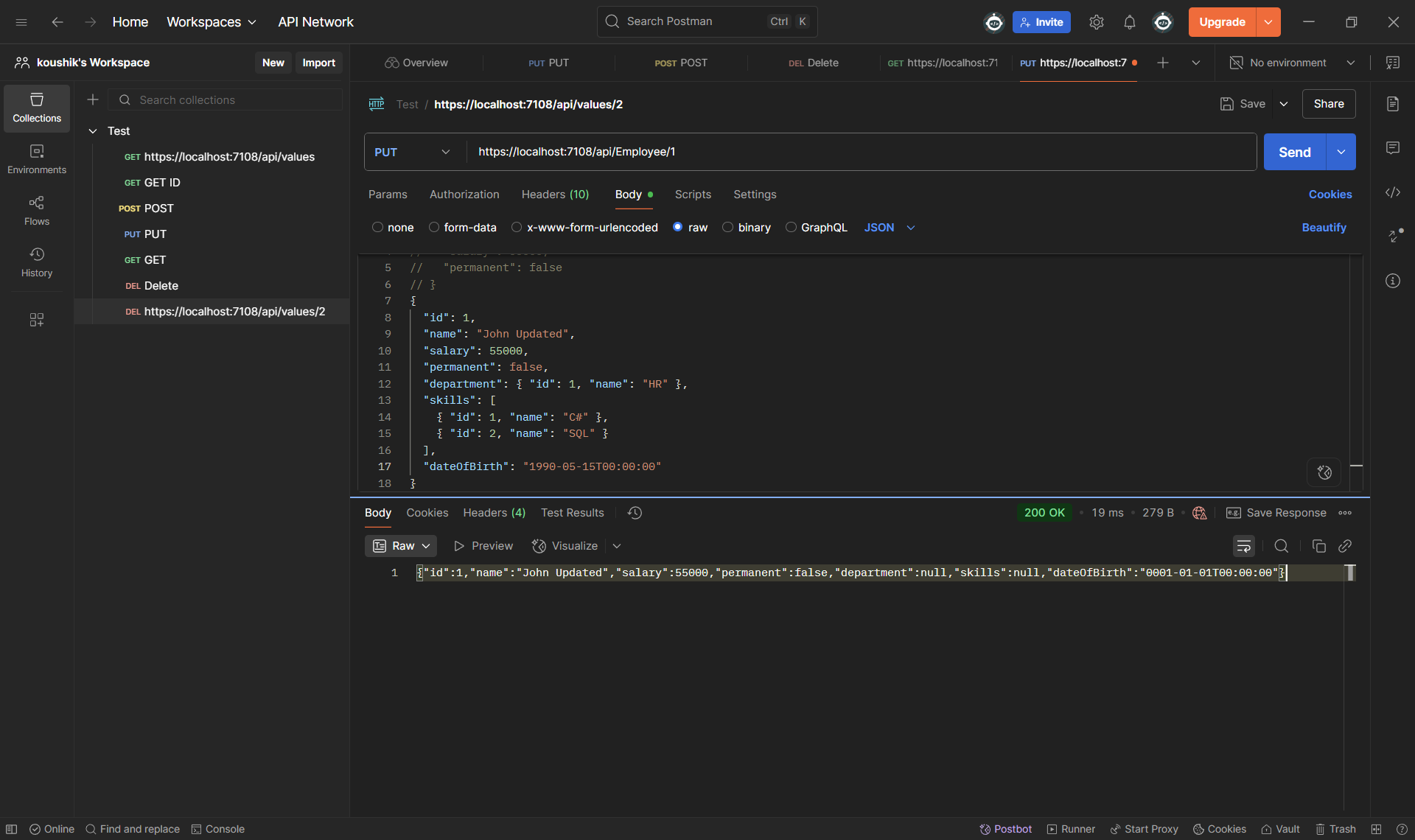
Swagger - PUT Employee with Invalid ID (400 BadRequest):



Swagger - GET /api/Employee Showing Updated List:



Postman - PUT Employee with Valid ID (200 OK)

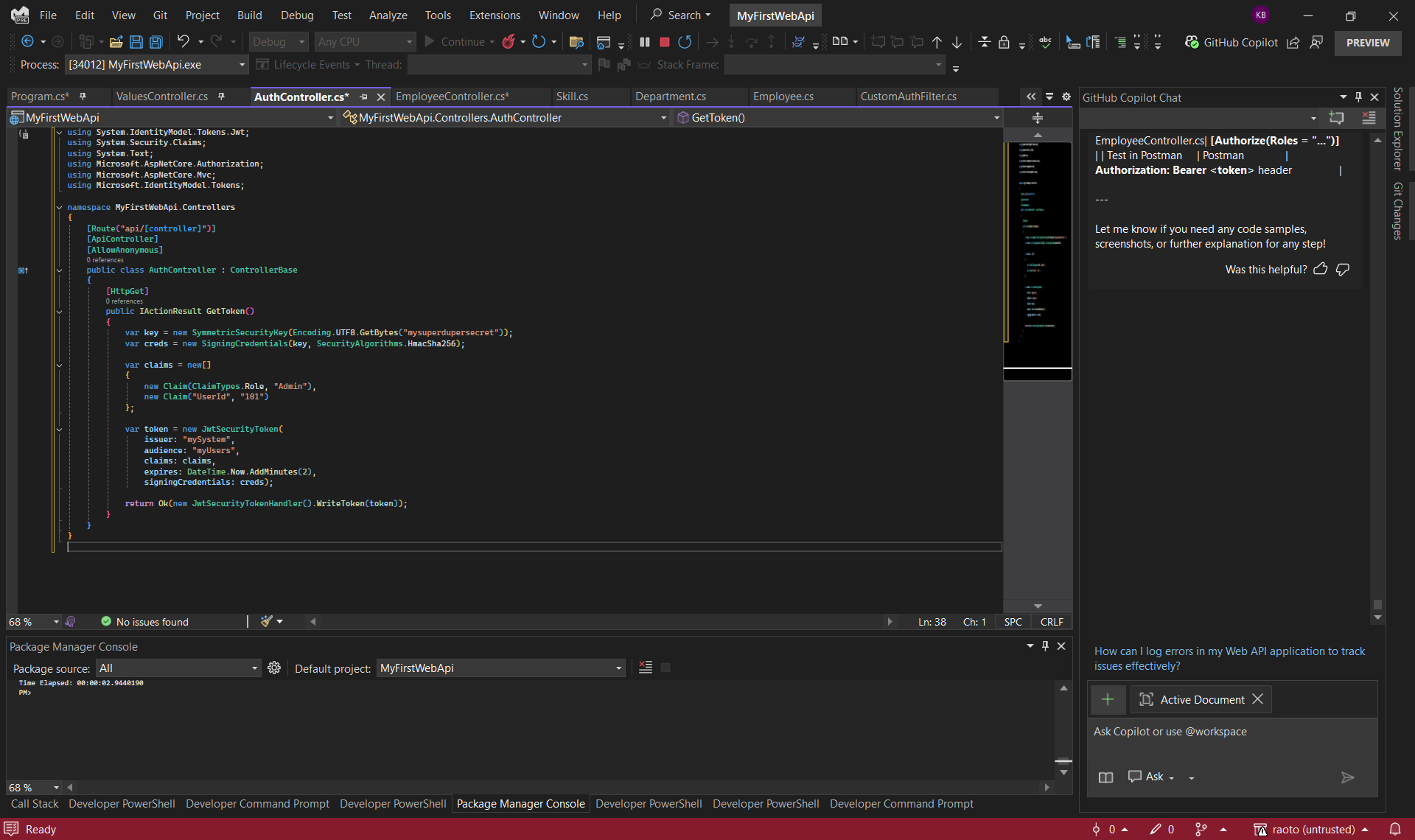


## Document 5: Securing API using JWT Token

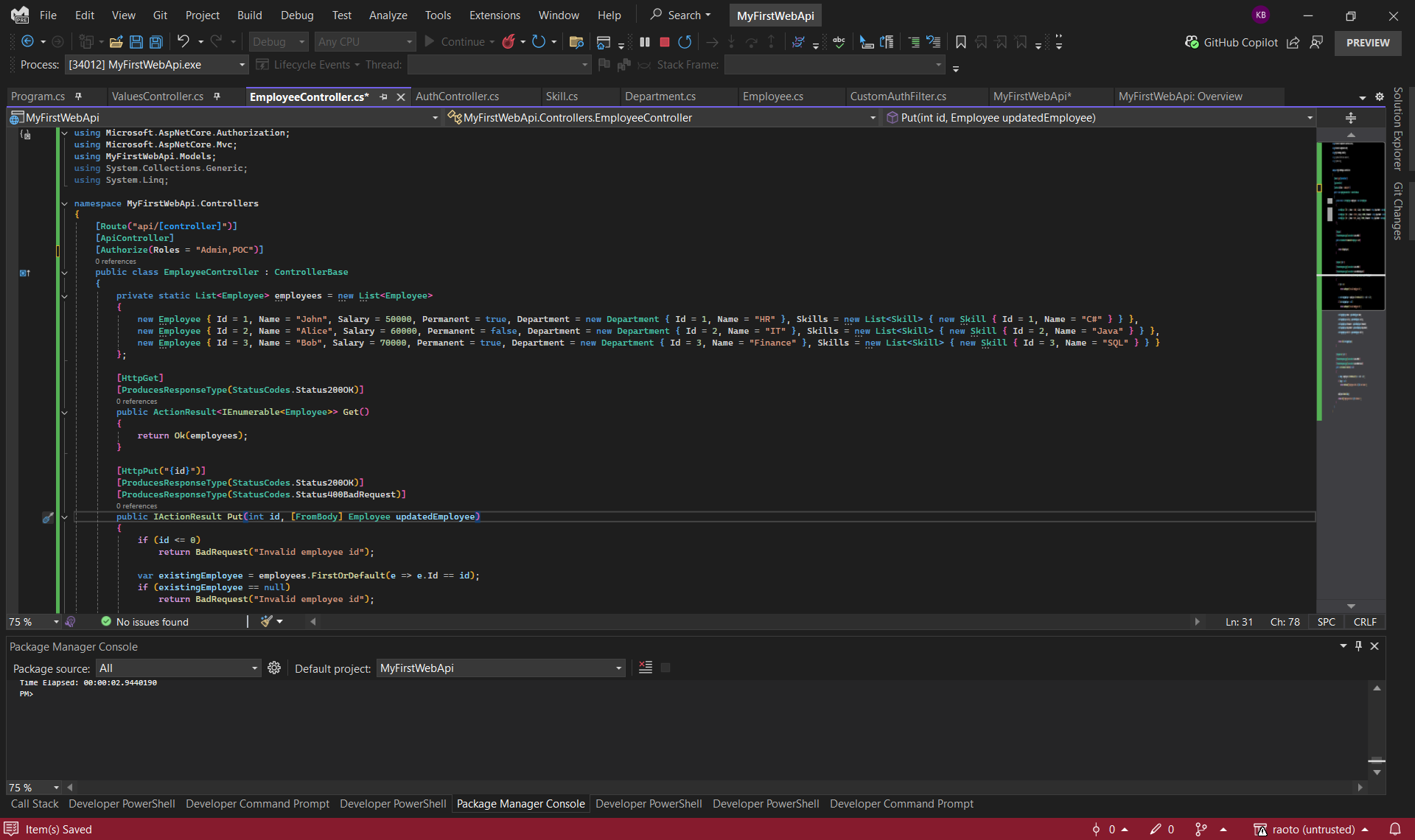
## NuGet Package Installation for JWT Authentication :

## C:\week3\week 4\doc5.png

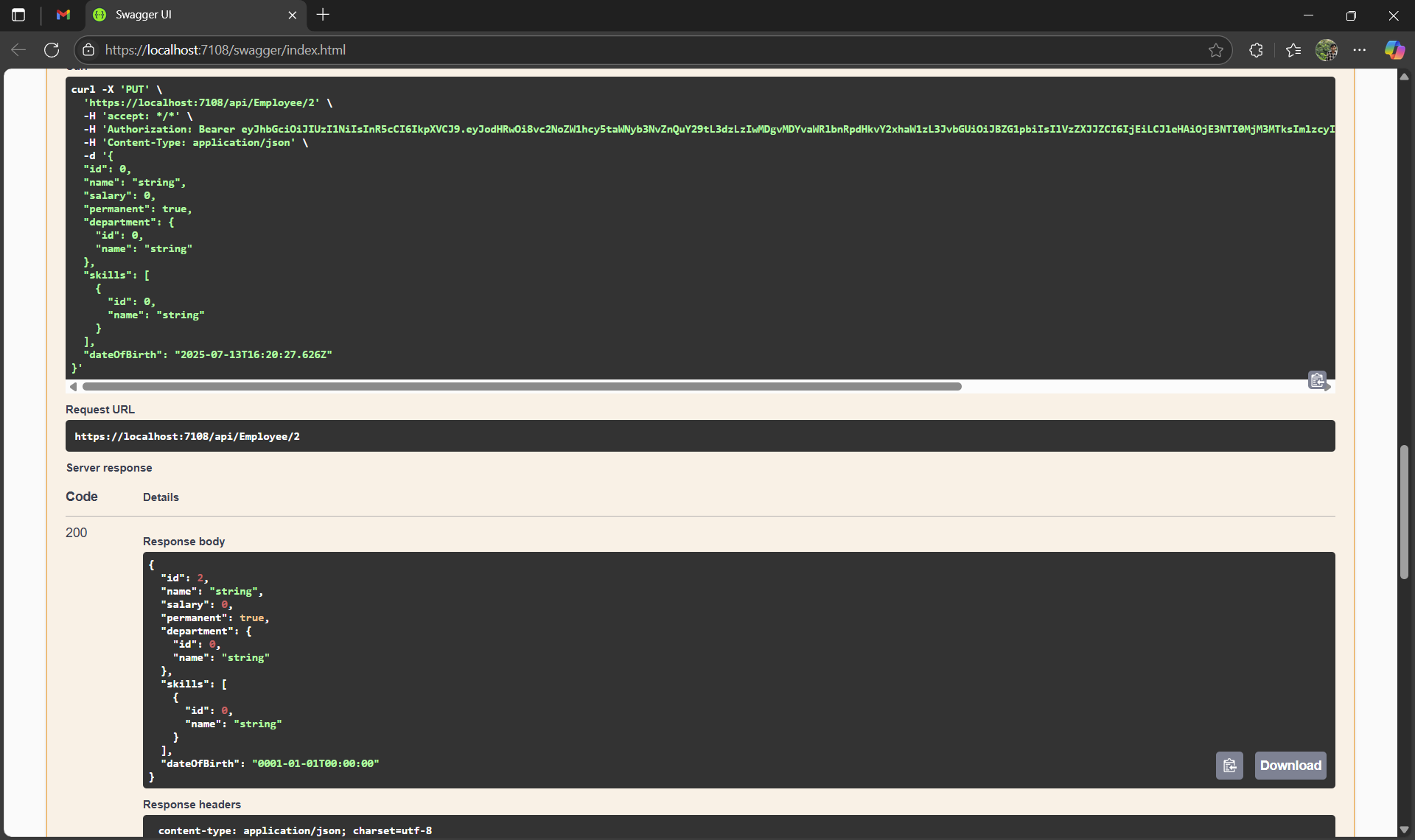
AuthController Implementation for JWT Token Generation:



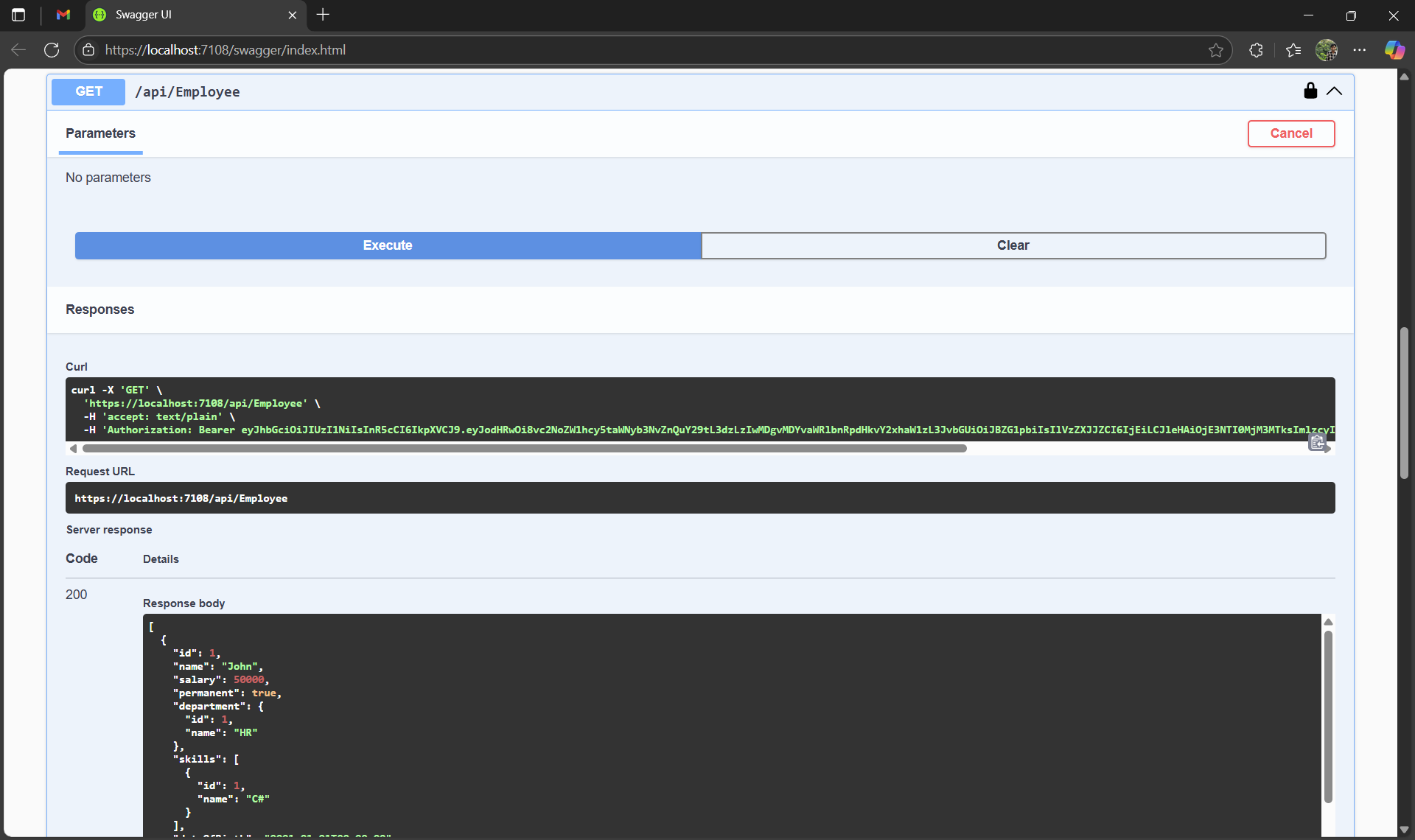
Securing API with Authorize Attribute:



Executing PUT /api/Employee with JWT Bearer Token:

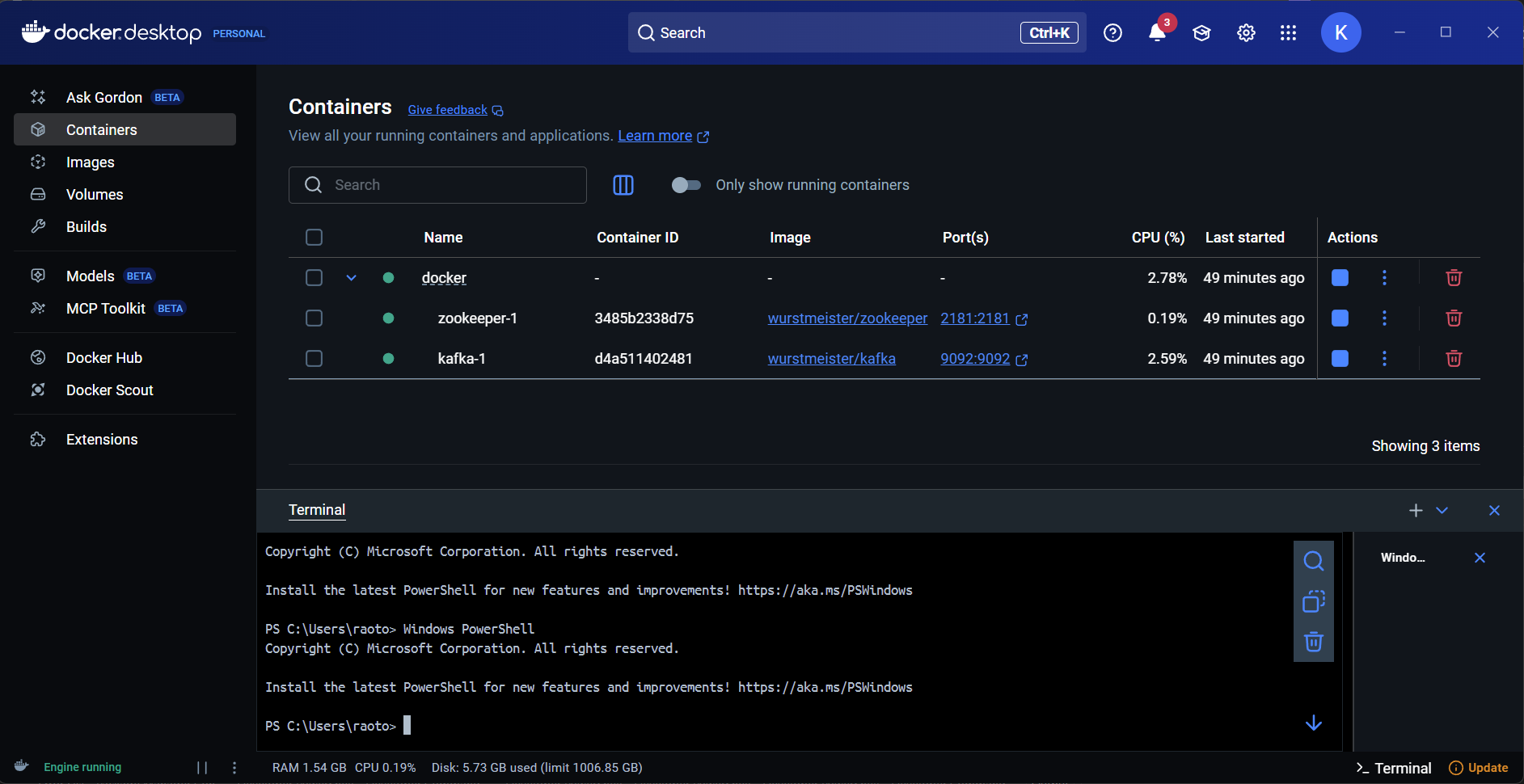


Executing GET /api/Employee/{id} with JWT Token:

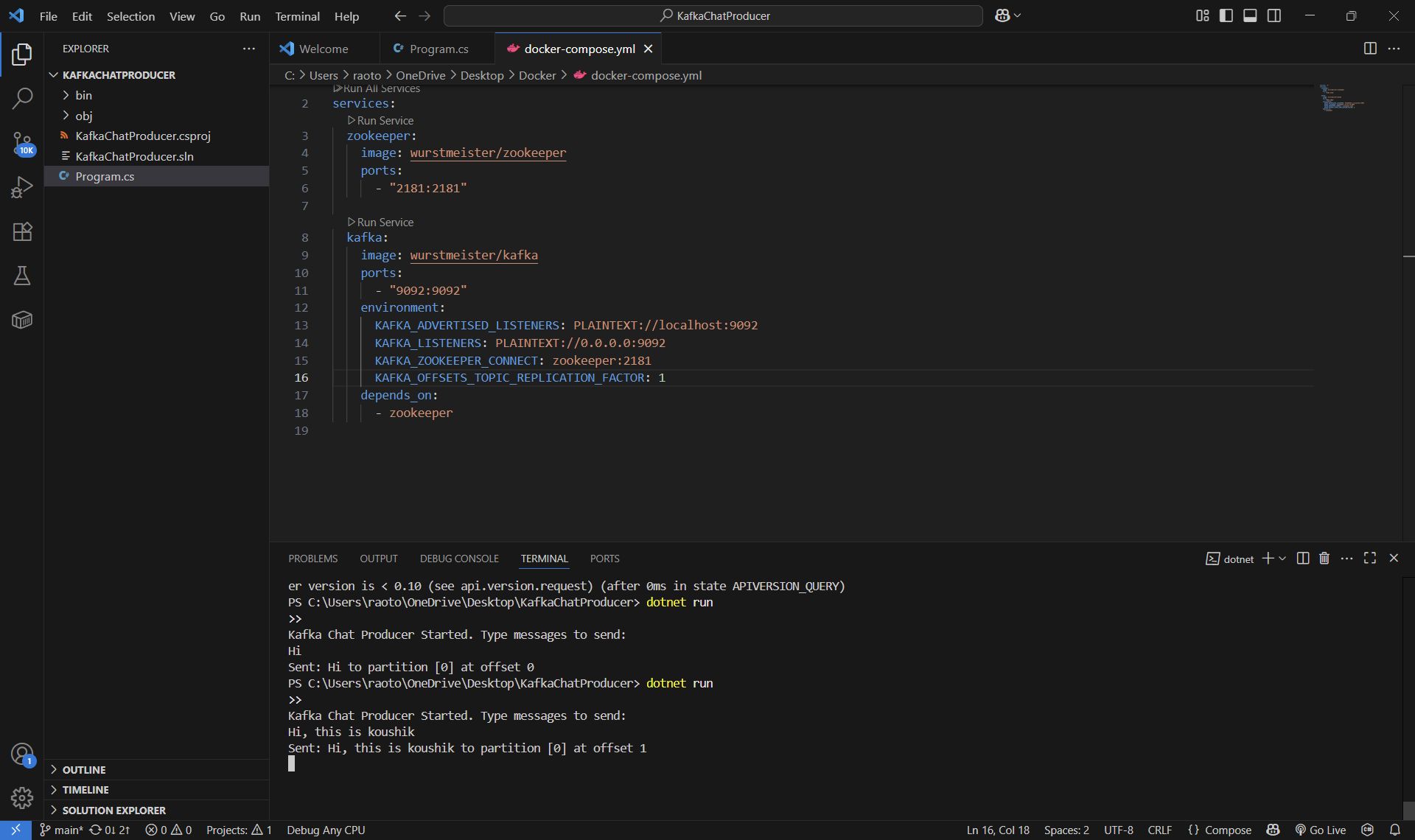


## Document 6: Kafka Integration with C# Chat App

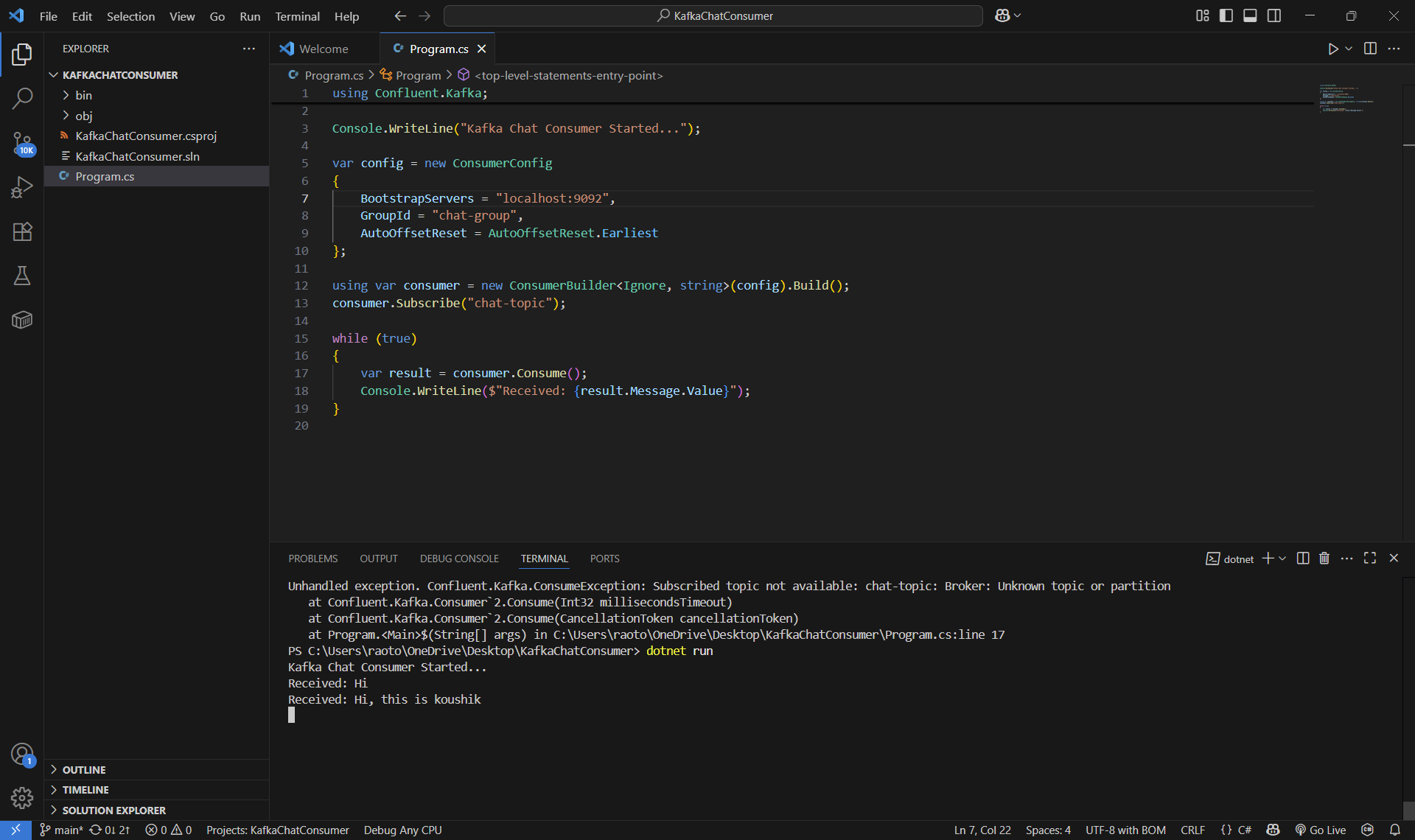
Docker Desktop Showing Zookeeper and Kafka Containers Running:



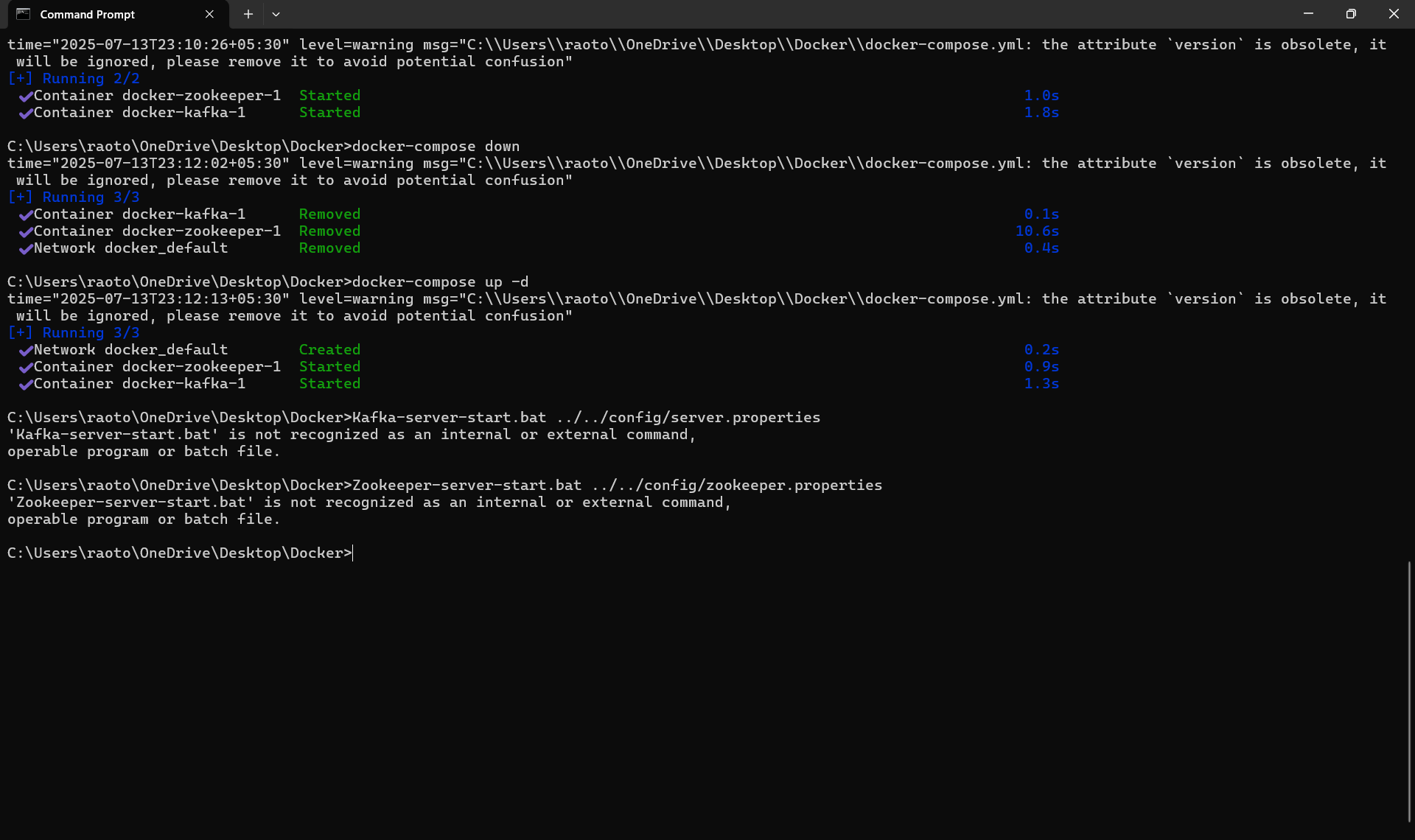
Kafka Chat Console Producer - Sending Messages:



Kafka Console or WinForms Consumer Receiving Messages:



Command Prompt Running Kafka Server Logs:



Two Windows Forms Chat Clients Communicating via Kafka:

