**Microservices Architecture using ASP.NET Core Web API**

LoginModel.cs

namespace JwtAuthDemo.Models

{

public class LoginModel

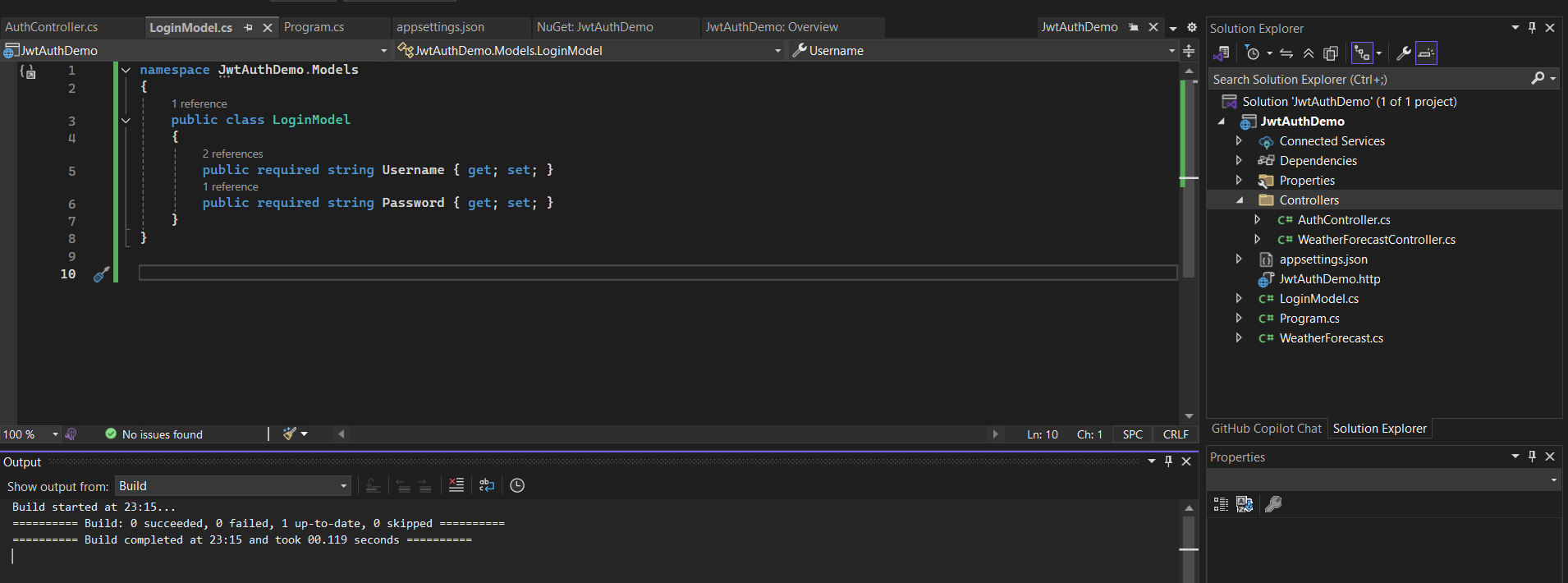
{

public required string Username { get; set; }

public required string Password { get; set; }

}

}



appsettings.json

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"AllowedHosts": "\*",

"Jwt": {

"Key": "ThisIsASecretKeyForJwtTokenWith256BitLengthToAvoidError",

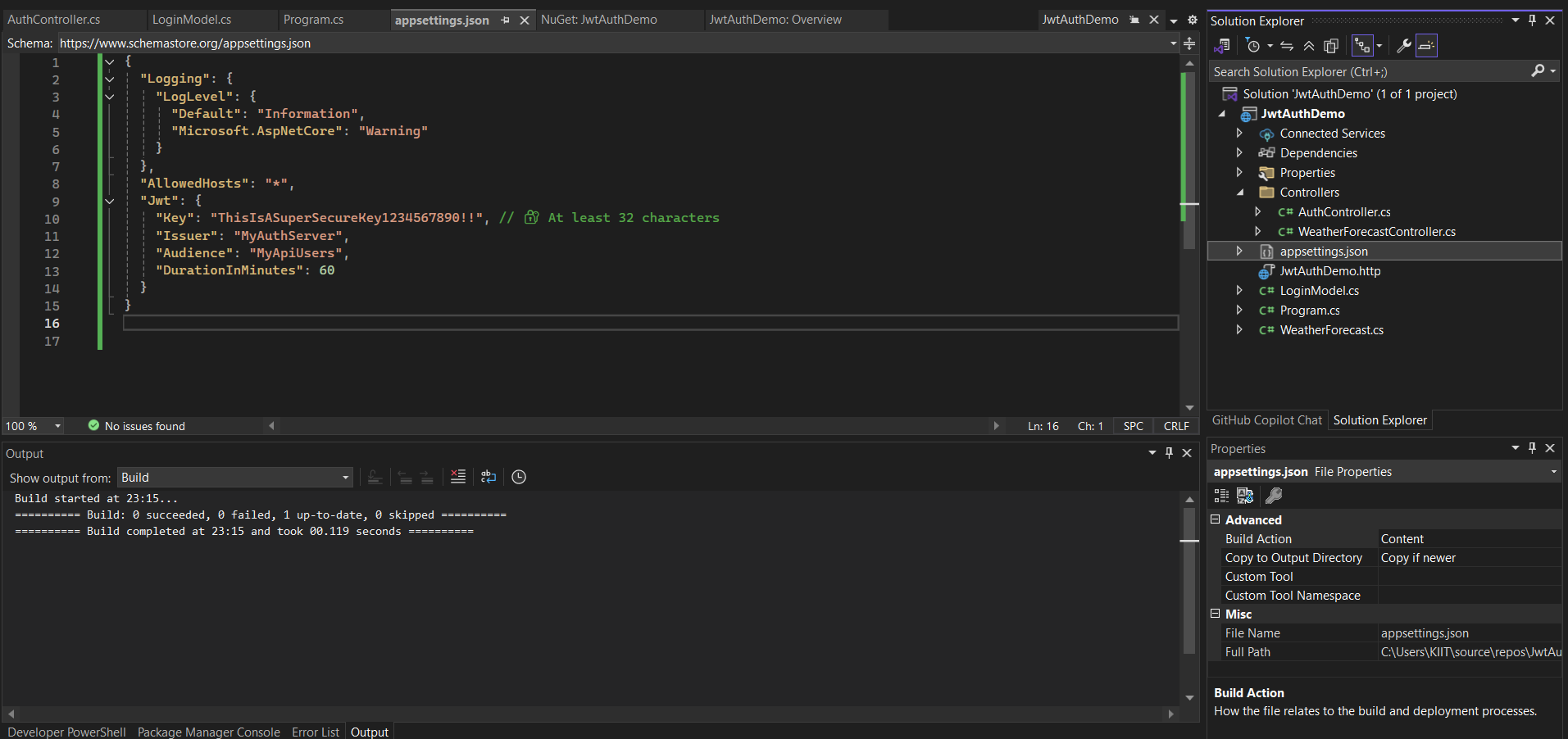
"Issuer": "MyAuthServer",

"Audience": "MyApiUsers",

"DurationInMinutes": 60

}

}



AuthController.cs

using Microsoft.AspNetCore.Mvc;

using Microsoft.IdentityModel.Tokens;

using System.IdentityModel.Tokens.Jwt;

using System.Security.Claims;

using System.Text;

using JwtAuthDemo.Models;

namespace JwtAuthDemo.Controllers

{

[ApiController]

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

public class AuthController : ControllerBase

{

private readonly IConfiguration \_configuration;

public AuthController(IConfiguration configuration)

{

\_configuration = configuration;

}

[HttpPost("login")]

public IActionResult Login([FromBody] LoginModel model)

{

if (model.Username == "admin" && model.Password == "password")

{

var claims = new[]

{

new Claim(ClaimTypes.Name, model.Username),

new Claim(ClaimTypes.Role, "Admin")

};

var secretKey = \_configuration["Jwt:Key"];

var key = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(secretKey));

var creds = new SigningCredentials(key, SecurityAlgorithms.HmacSha256);

var token = new JwtSecurityToken(

issuer: \_configuration["Jwt:Issuer"],

audience: \_configuration["Jwt:Audience"],

claims: claims,

expires: DateTime.Now.AddMinutes(30),

signingCredentials: creds

);

return Ok(new

{

token = new JwtSecurityTokenHandler().WriteToken(token)

});

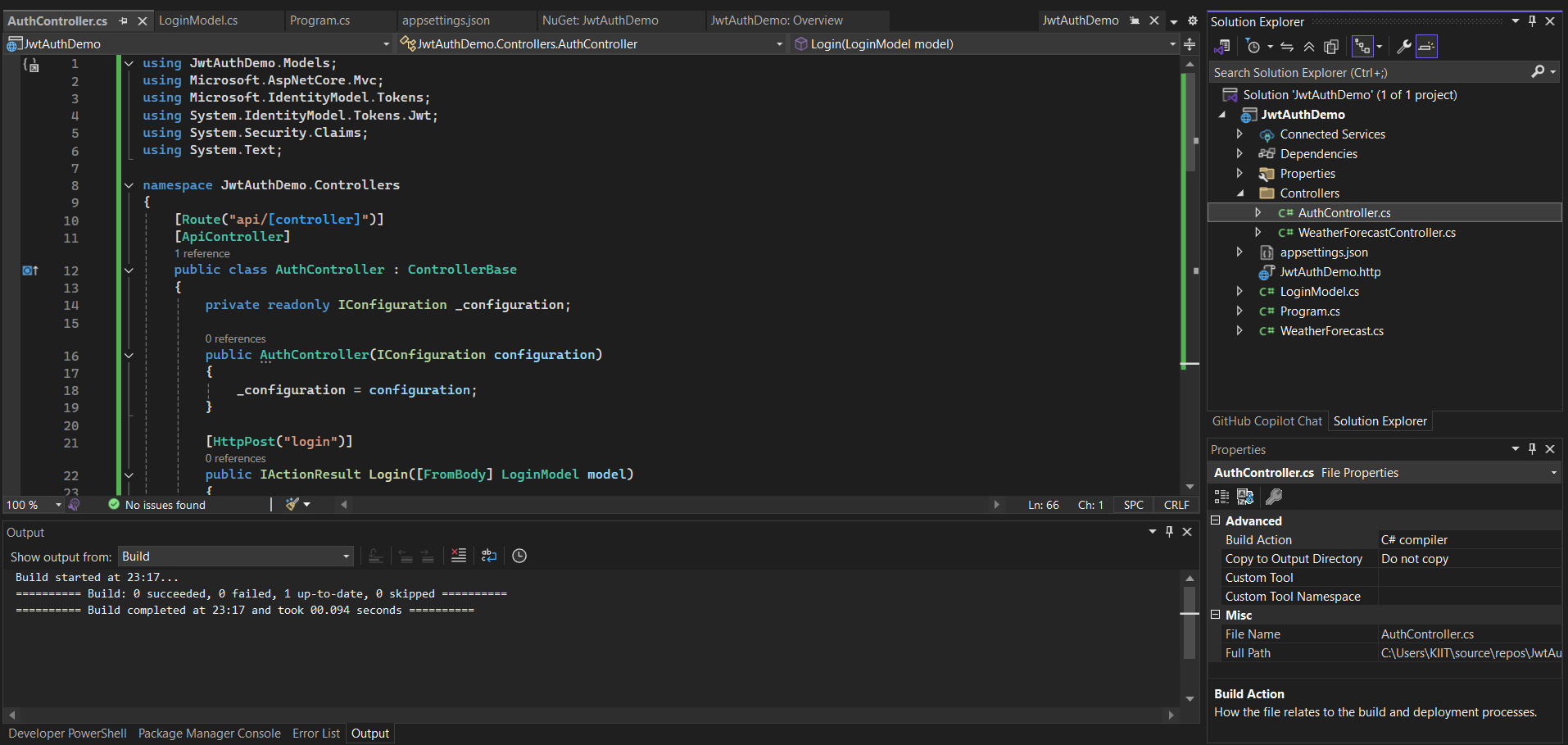
}

return Unauthorized("Invalid credentials.");

}

}

}



**Postman APP results and inputs**

URL: <https://localhost:7199/swagger/index.html>

Method: POST

Body

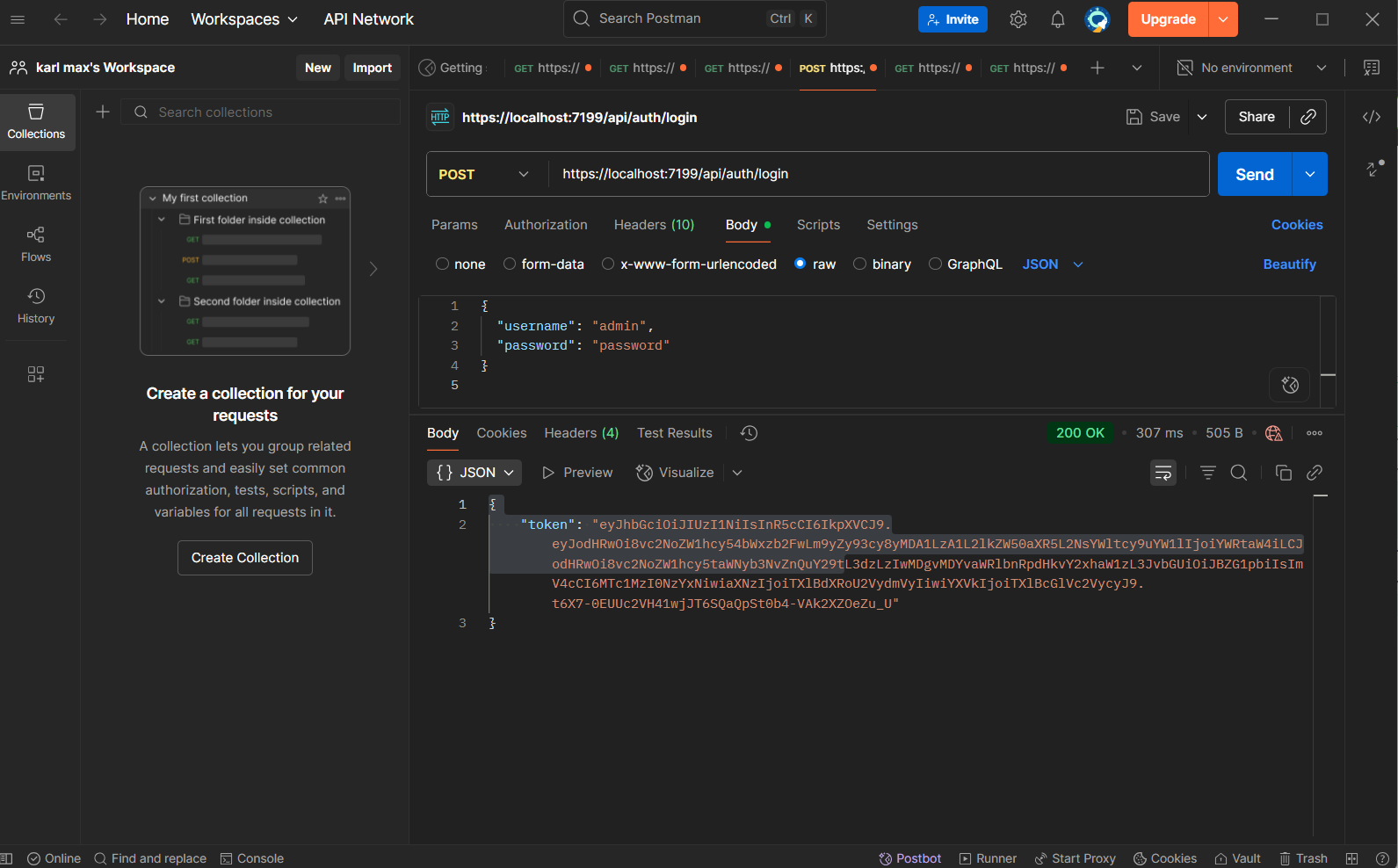
{

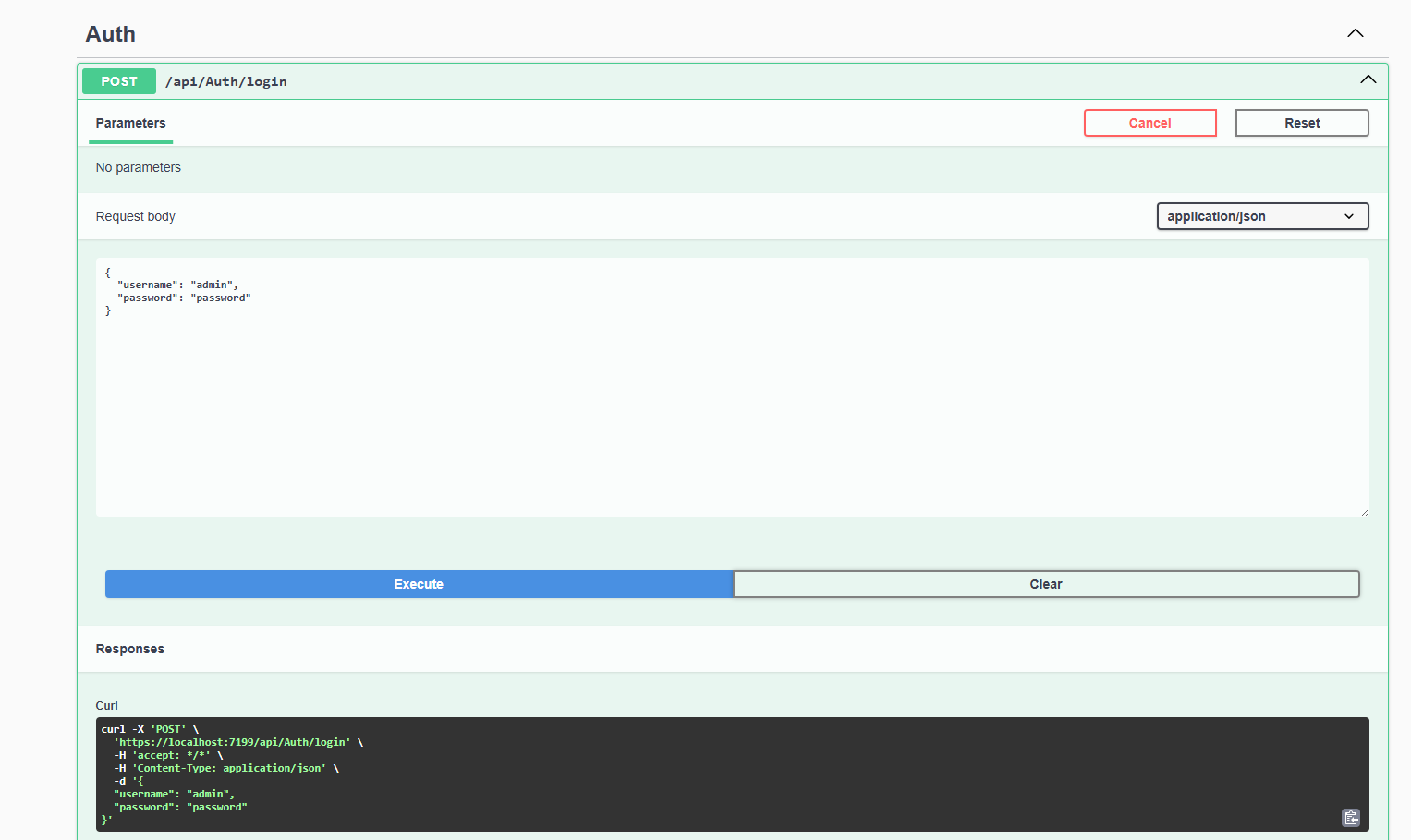
"username": "admin",

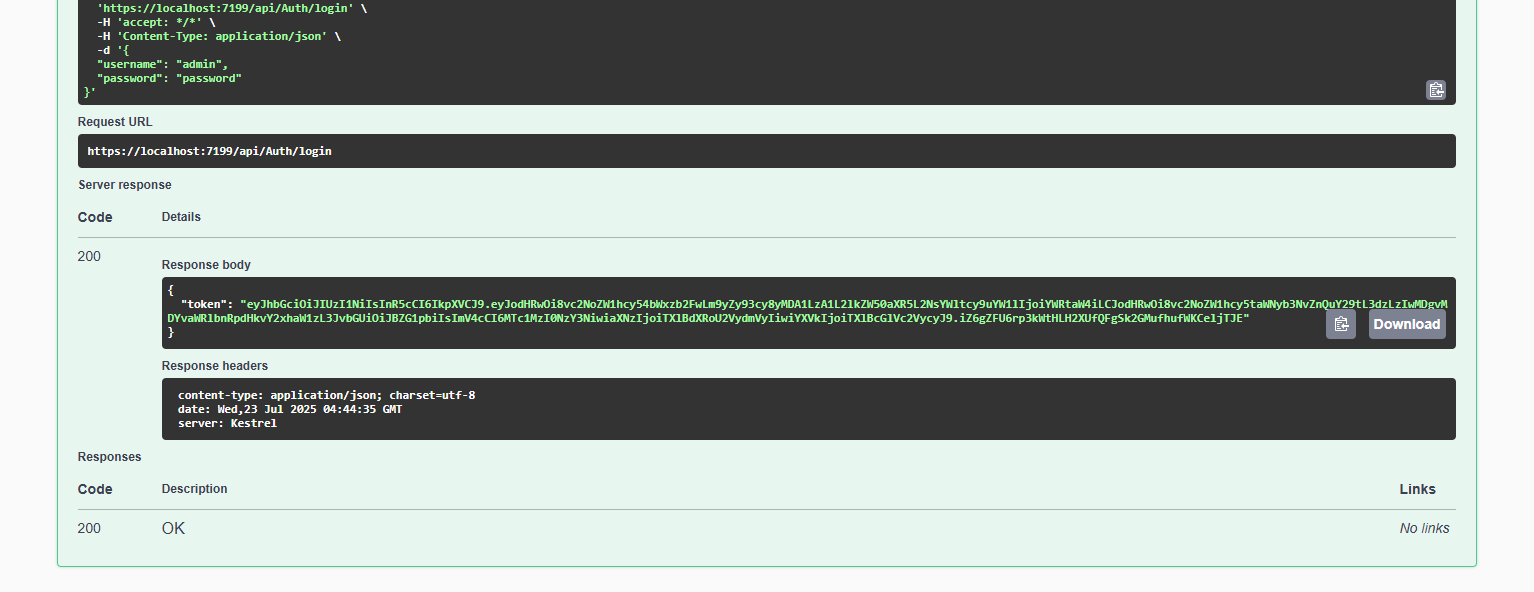
"password": "password"

}

**OUTPUT**







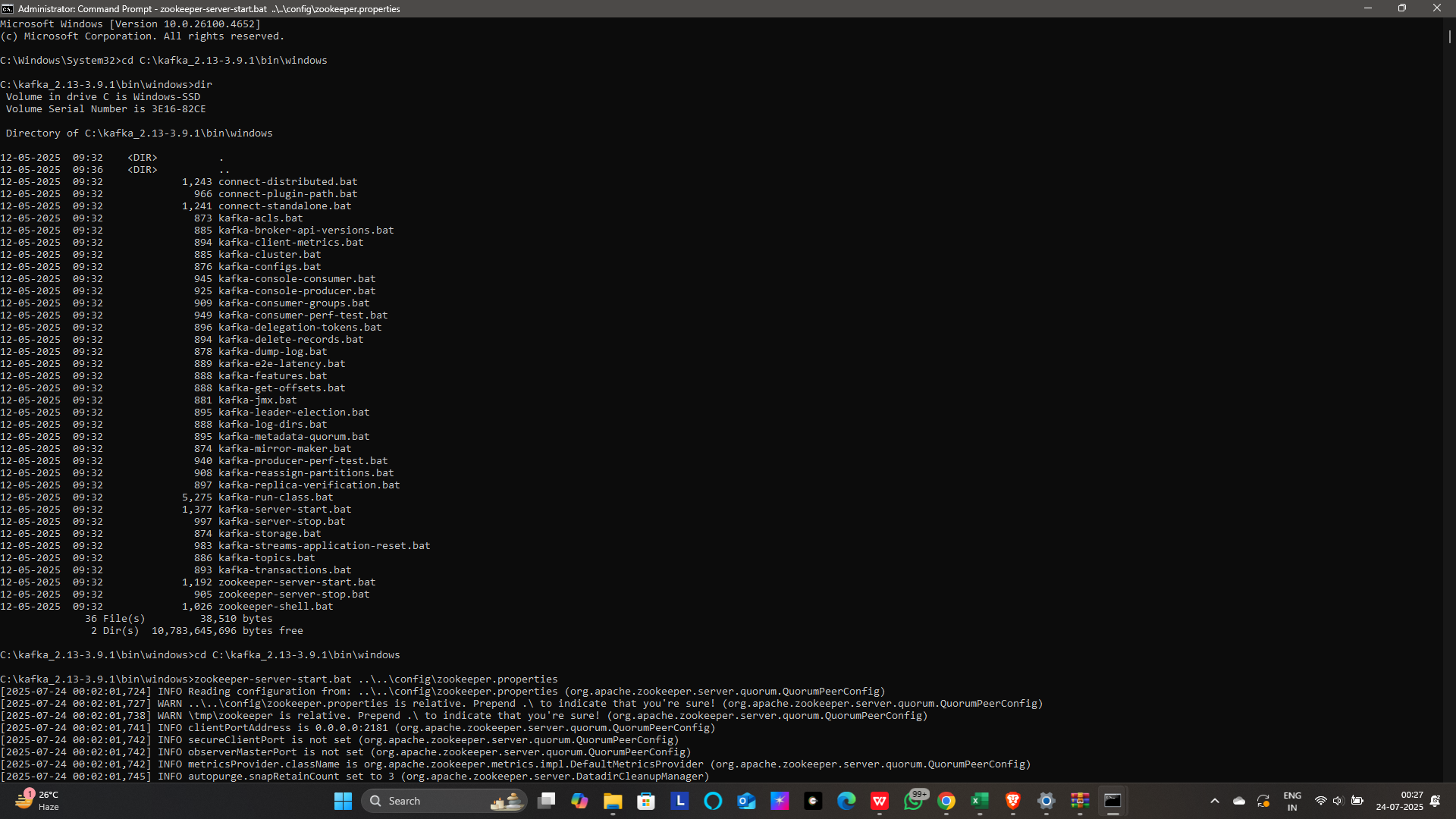
**ASP.NET Core 8.0 Web API**

Kafka Integration with C#:

ZooKeeper:

cd C:\kafka\_2.13-3.9.1\bin\windows

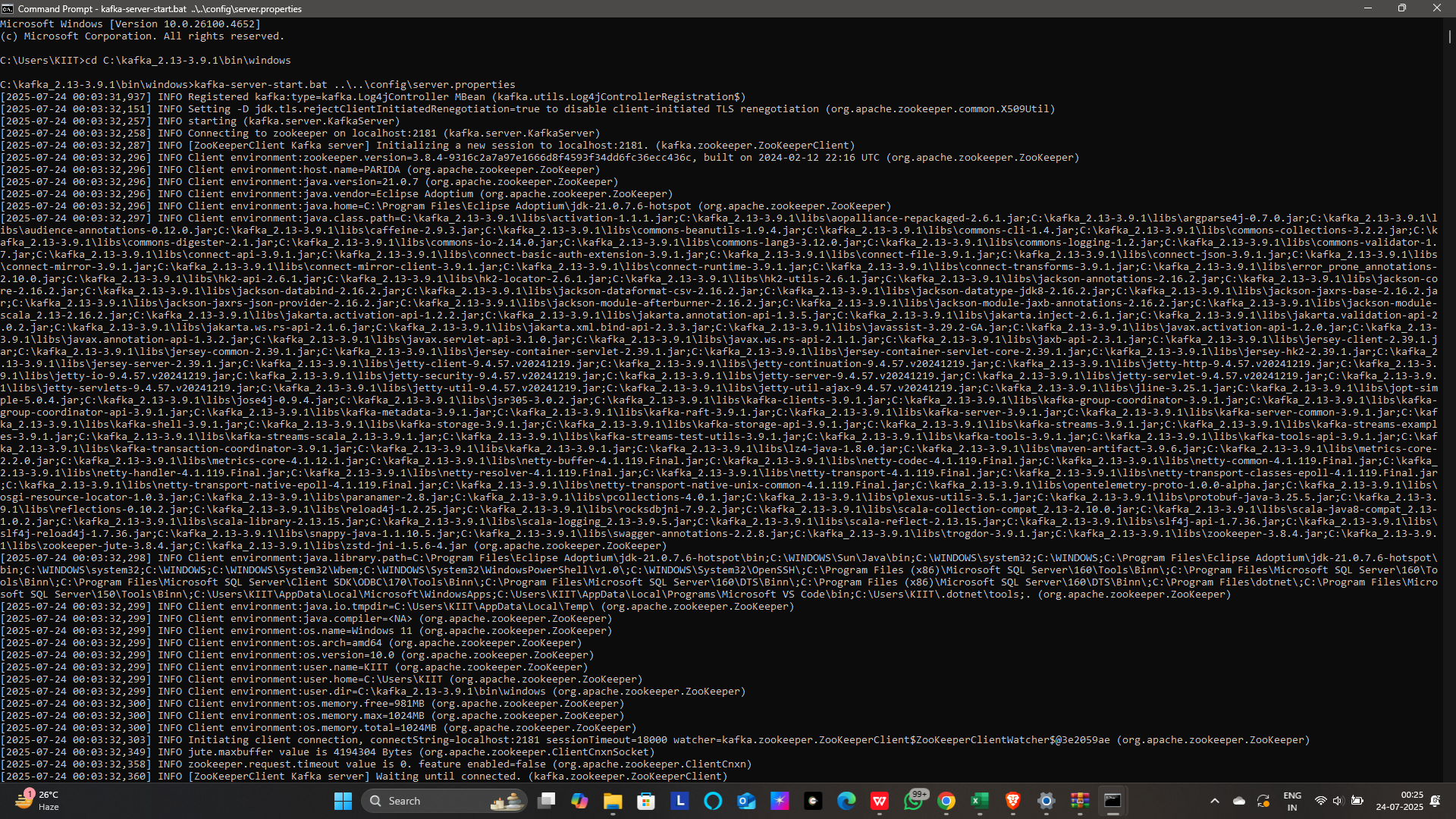
zookeeper-server-start.bat ..\..\config\zookeeper.properties



Kafka Broker:

cd C:\kafka\_2.13-3.9.1\bin\windows

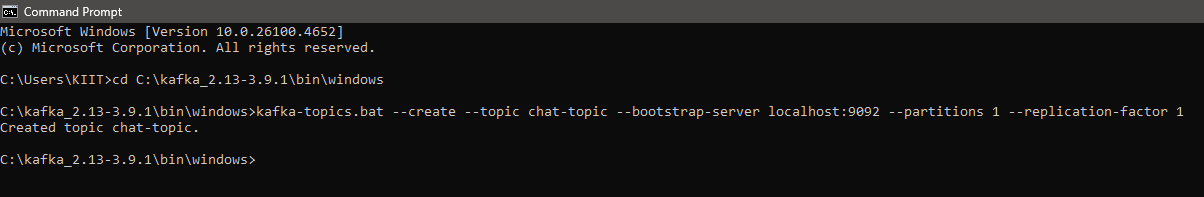
kafka-server-start.bat ..\..\config\server.properties



Topic Creation:

cd C:\kafka\_2.13-3.9.1\bin\windows

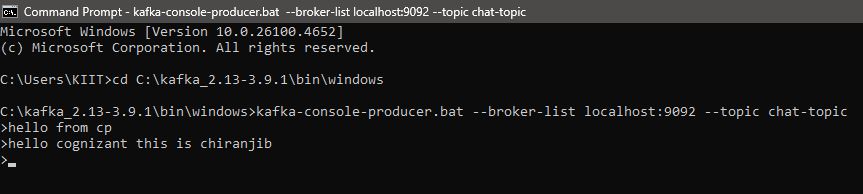
kafka-topics.bat --create --topic chat-topic --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1



Start Producer:

cd C:\kafka\_2.13-3.9.1\bin\windows

kafka-console-producer.bat --broker-list localhost:9092 --topic chat-topic



Start Consumer:

cd C:\kafka\_2.13-3.9.1\bin\windows

kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic chat-topic --from-beginning

