Hands on:Authentication & Authorization in ASP.NET Core Web API

This document summarizes everything I learnt while working on hands-on exercises related to Authentication and Authorization in ASP.NET Core Web API using JWT. It includes code, configuration, and placeholders for screenshots.

# Exercise 1: JWT Authentication Setup

We created a simple AuthController that generates a JWT token based on username/password.

## Code - AuthController.cs

[HttpPost("login")]  
public IActionResult Login([FromBody] LoginRequest request)  
{  
 if (request.Username == "admin" && request.Password == "password")  
 {  
 var token = GenerateJwtToken(request.Username);  
 return Ok(new { token });  
 }  
 return Unauthorized();  
}  
  
private string GenerateJwtToken(string username)  
{  
 var claims = new[] {  
 new Claim(JwtRegisteredClaimNames.Sub, username),  
 new Claim(JwtRegisteredClaimNames.Jti, Guid.NewGuid().ToString())  
 };  
 var key = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(\_config["Jwt:Key"]));  
 var creds = new SigningCredentials(key, SecurityAlgorithms.HmacSha256);  
  
 var token = new JwtSecurityToken(  
 issuer: \_config["Jwt:Issuer"],  
 audience: \_config["Jwt:Audience"],  
 claims: claims,  
 expires: DateTime.UtcNow.AddMinutes(Convert.ToDouble(\_config["Jwt:DurationInMinutes"])),  
 signingCredentials: creds);  
  
 return new JwtSecurityTokenHandler().WriteToken(token);  
}

# Screenshot 2025-07-20 171350

# Exercise 2: SecureController with [Authorize]

We created a SecureController that required the JWT token to access.

## Code - SecureController.cs

[Authorize]  
[ApiController]  
[Route("api/[controller]")]  
public class SecureController : ControllerBase  
{  
 [HttpGet("secret")]  
 public IActionResult GetSecret()  
 {  
 return Ok("This is a protected endpoint.");  
 }  
}

# Screenshot 2025-07-20 172836

# Exercise 3: AdminController with Role-Based Authorization

We set up role-based authorization using roles embedded in the JWT token.

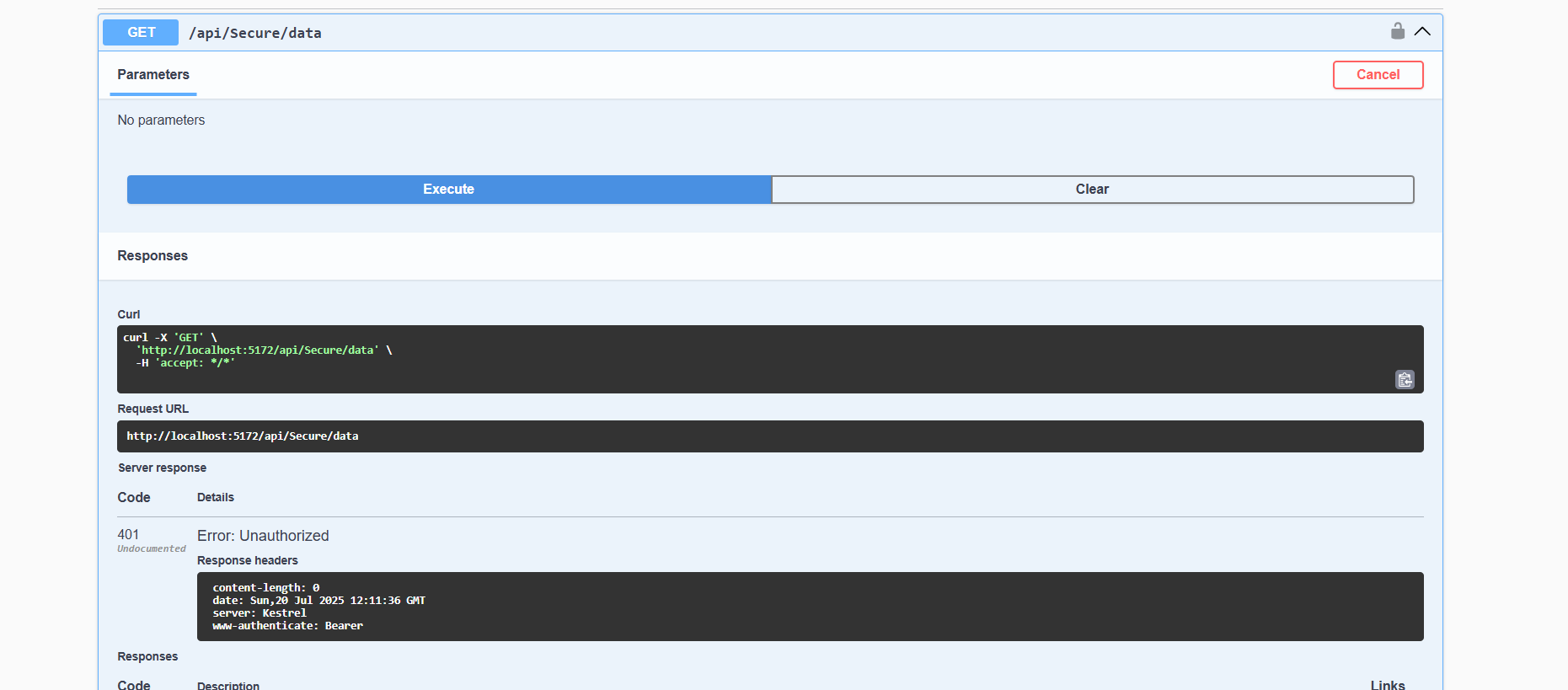
## Code - AdminController.cs

[Authorize(Roles = "Admin")]  
[ApiController]  
[Route("api/[controller]")]  
public class AdminController : ControllerBase  
{  
 [HttpGet("dashboard")]  
 public IActionResult Dashboard()  
 {  
 return Ok("Welcome Admin!");  
 }  
}

# Screenshot 2025-07-20 172808

# Unauthorized access

# 



# Configuration - appsettings.json

"Jwt": {  
 "Key": "YourSuperSecretKey123456",  
 "Issuer": "yourapi.com",  
 "Audience": "yourapi.com",  
 "DurationInMinutes": 30  
}

# Program.cs

builder.Services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)  
 .AddJwtBearer(options =>  
 {  
 options.TokenValidationParameters = new TokenValidationParameters  
 {  
 ValidateIssuer = true,  
 ValidateAudience = true,  
 ValidateLifetime = true,  
 ValidateIssuerSigningKey = true,  
 ValidIssuer = builder.Configuration["Jwt:Issuer"],  
 ValidAudience = builder.Configuration["Jwt:Audience"],  
 IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(builder.Configuration["Jwt:Key"]))  
 };  
 });

# Postman screenshots Screenshot 2025-07-20 173512

