**Week-5 Hands-on**

[**JWTAuthentication**](https://github.com/SurendarNehru/Digital-Nurture-4.0-DotNetFSE/tree/main/week5/%20JWTAuthentication)

[**AdminController.cs**](https://github.com/SurendarNehru/Digital-Nurture-4.0-DotNetFSE/blob/main/week5/%20JWTAuthentication/AdminController.cs)**:**

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

namespace JwtAuthDemo.Controllers

{

[ApiController]

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

public class AdminController : ControllerBase

{

[HttpGet("dashboard")]

[Authorize(Roles = "Admin")]

public IActionResult GetAdminDashboard()

{

return Ok("Welcome to the admin dashboard.");

}

}

}

**SecureController.cs:**

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

namespace JwtAuthDemo.Controllers

{

    [ApiController]

    [Route("api/[controller]")]

    public class SecureController : ControllerBase

    {

        [HttpGet("data")]

        [Authorize]

        public IActionResult GetSecureData()

        {

            return Ok("This is protected data");

        }

    }

}

**AuthenticateController.cs:**

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Configuration;

using Microsoft.IdentityModel.Tokens;

using System.IdentityModel.Tokens.Jwt;

using System.Security.Claims;

using System.Text;

namespace Jwtapi.Controllers

{

[ApiController]

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

public class AuthenticateController : ControllerBase

{

private readonly IConfiguration \_configuration;

public AuthenticateController(IConfiguration configuration)

{

\_configuration = configuration;

}

[HttpPost("login")]

public IActionResult Login([FromBody] Login model)

{

if (IsValidUser(model))

{

var token = GenerateJwtToken(model.Username!);

return Ok(new { Token = token });

}

return Unauthorized();

}

private bool IsValidUser(Login model)

{

return model.Username == "admin" && model.Password == "password";

}

private string GenerateJwtToken(string username)

{

var claims = new[]

{

new Claim(ClaimTypes.Name, username),

new Claim(ClaimTypes.Role, "Admin")

};

var key = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(\_configuration["Jwt:Key"]!));

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(Convert.ToDouble(\_configuration["Jwt:DurationInMinutes"])),

signingCredentials: creds

);

return new JwtSecurityTokenHandler().WriteToken(token);

}

}

}

**Program.cs:**

using Microsoft.AspNetCore.Authentication.JwtBearer;

using Microsoft.IdentityModel.Tokens;

using System.IdentityModel.Tokens.Jwt;

using System.Security.Claims;

using System.Text;

using Microsoft.OpenApi.Models;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddAuthentication("Bearer")

.AddJwtBearer("Bearer", 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"] ?? "FallbackSecretKey123"))

};

options.Events = new JwtBearerEvents

{

OnAuthenticationFailed = context =>

{

if (context.Exception.GetType() == typeof(SecurityTokenExpiredException))

{

context.Response.Headers.Append("Token-Expired", "true");

}

return Task.CompletedTask;

}

};

});

builder.Services.AddAuthorization();

builder.Services.AddControllers();

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

var app = builder.Build();

app.UseSwagger();

app.UseSwaggerUI();

app.UseHttpsRedirection();

app.UseAuthentication();

app.UseAuthorization();

app.MapControllers();

app.Run();

**Login.cs:**

public class Login

{

public string? Username { get; set; }

public string? Password { get; set; }

}

**Jwtapi.csproj:**

<Project Sdk="Microsoft.NET.Sdk.Web">

<PropertyGroup>

<TargetFramework>net9.0</TargetFramework>

<Nullable>enable</Nullable>

<ImplicitUsings>enable</ImplicitUsings>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="Microsoft.AspNetCore.Authentication.JwtBearer" Version="9.0.7" />

<PackageReference Include="Microsoft.AspNetCore.OpenApi" Version="9.0.6" />

<PackageReference Include="Swashbuckle.AspNetCore" Version="9.0.3" />

</ItemGroup>

</Project>

**appsettings.Development.json:**

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

}

}

**appsettings.json:**

{

"Jwt": {

"Key": "this\_is\_a\_very\_strong\_secret\_key\_1234567890",

"Issuer": "YourIssuer",

"Audience": "YourAudience",

"DurationInMinutes": 60

}

}

