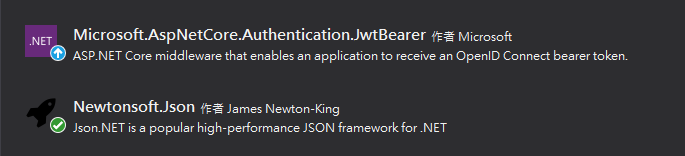
1. 安裝JwtBearer
2. 安裝 Newtonsoft.Json



1. appsettings.json新增Jwt參數

"JwtSettings": {

"Issuer": "JwtAuthDemo",

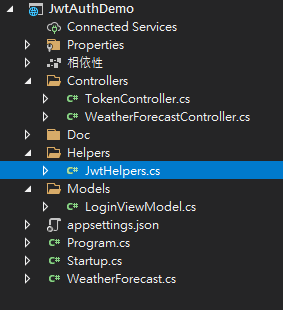
"SignKey": "JwtAuthDemoSignKey"

}



SignKey 至少16個字以上

1. 建立JwtHelpers



using Microsoft.Extensions.Configuration;

using Microsoft.IdentityModel.Tokens;

using System;

using System.Collections.Generic;

using System.Security.Claims;

using System.Text;

using System.IdentityModel.Tokens.Jwt;

namespace JwtAuthDemo.Helpers

{

public class JwtHelpers

{

private readonly IConfiguration Configuration;

public JwtHelpers(IConfiguration configuration)

{

this.Configuration = configuration;

}

public string GenerateToken(string userName, int expireMinutes = 30)

{

var issuer = Configuration.GetValue<string>("JwtSettings:Issuer");

var signKey = Configuration.GetValue<string>("JwtSettings:SignKey");

// 設定要加入到 JWT Token 中的聲明資訊(Claims)

var claims = new List<Claim>();

claims.Add(new Claim(JwtRegisteredClaimNames.Sub, userName)); // User.Identity.Name

claims.Add(new Claim(JwtRegisteredClaimNames.Jti, Guid.NewGuid().ToString())); // JWT ID

// 你可以自行擴充 "roles" 加入登入者該有的角色

claims.Add(new Claim("roles", "Admin"));

//claims.Add(new Claim("roles", "Users"));

var userClaimsIdentity = new ClaimsIdentity(claims);

// 建立一組對稱式加密的金鑰，主要用於 JWT 簽章之用

var securityKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(signKey));

// HmacSha256 有要求必須要大於 128 bits，所以 key 不能太短，至少要 16 字元以上

// https://stackoverflow.com/questions/47279947/idx10603-the-algorithm-hs256-requires-the-securitykey-keysize-to-be-greater

var signingCredentials = new SigningCredentials(securityKey, SecurityAlgorithms.HmacSha256Signature);

// 建立 SecurityTokenDescriptor

var tokenDescriptor = new SecurityTokenDescriptor

{

Issuer = issuer,

Subject = userClaimsIdentity,

Expires = DateTime.Now.AddMinutes(expireMinutes),

SigningCredentials = signingCredentials

};

// 產出所需要的 JWT securityToken 物件，並取得序列化後的 Token 結果(字串格式)

var tokenHandler = new JwtSecurityTokenHandler();

var securityToken = tokenHandler.CreateToken(tokenDescriptor);

var serializeToken = tokenHandler.WriteToken(securityToken);

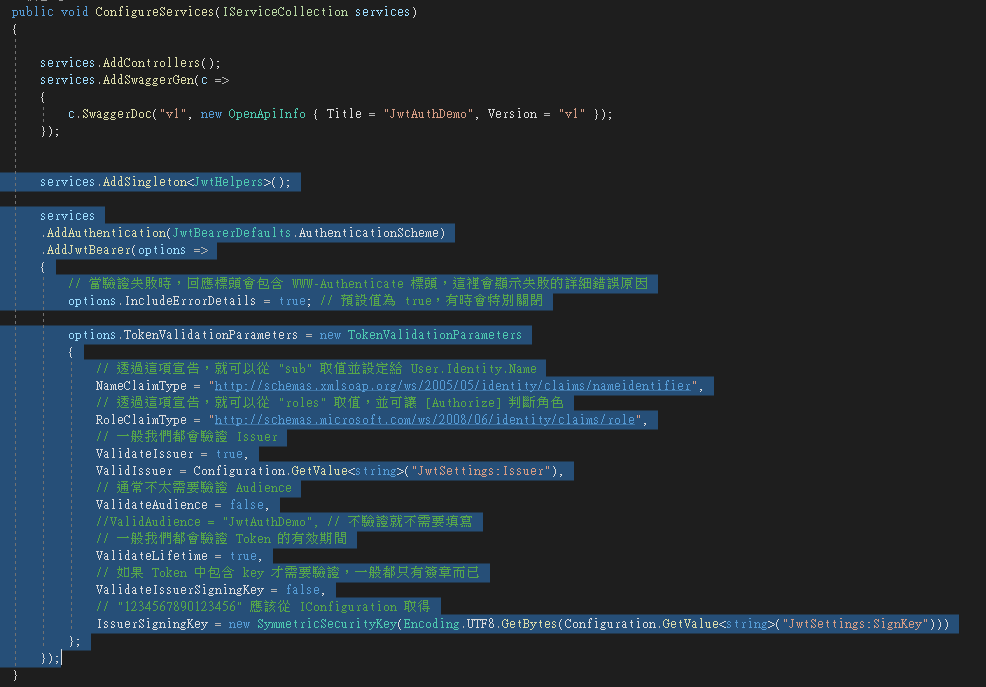
return serializeToken;

}

}

}

1. 設定Startup.cs



services.AddSingleton<JwtHelpers>();

services

.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)

.AddJwtBearer(options =>

{

// 當驗證失敗時，回應標頭會包含 WWW-Authenticate 標頭，這裡會顯示失敗的詳細錯誤原因

options.IncludeErrorDetails = true; // 預設值為 true，有時會特別關閉

options.TokenValidationParameters = new TokenValidationParameters

{

// 透過這項宣告，就可以從 "sub" 取值並設定給 User.Identity.Name

NameClaimType = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier",

// 透過這項宣告，就可以從 "roles" 取值，並可讓 [Authorize] 判斷角色

RoleClaimType = "http://schemas.microsoft.com/ws/2008/06/identity/claims/role",

// 一般我們都會驗證 Issuer

ValidateIssuer = true,

ValidIssuer = Configuration.GetValue<string>("JwtSettings:Issuer"),

// 通常不太需要驗證 Audience

ValidateAudience = false,

//ValidAudience = "JwtAuthDemo", // 不驗證就不需要填寫

// 一般我們都會驗證 Token 的有效期間

ValidateLifetime = true,

// 如果 Token 中包含 key 才需要驗證，一般都只有簽章而已

ValidateIssuerSigningKey = false,

// "1234567890123456" 應該從 IConfiguration 取得

IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(Configuration.GetValue<string>("JwtSettings:SignKey")))

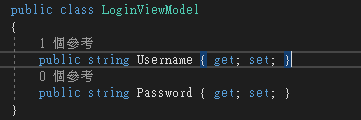
};

});



app.UseAuthentication();

1. 建立Model



public class LoginViewModel

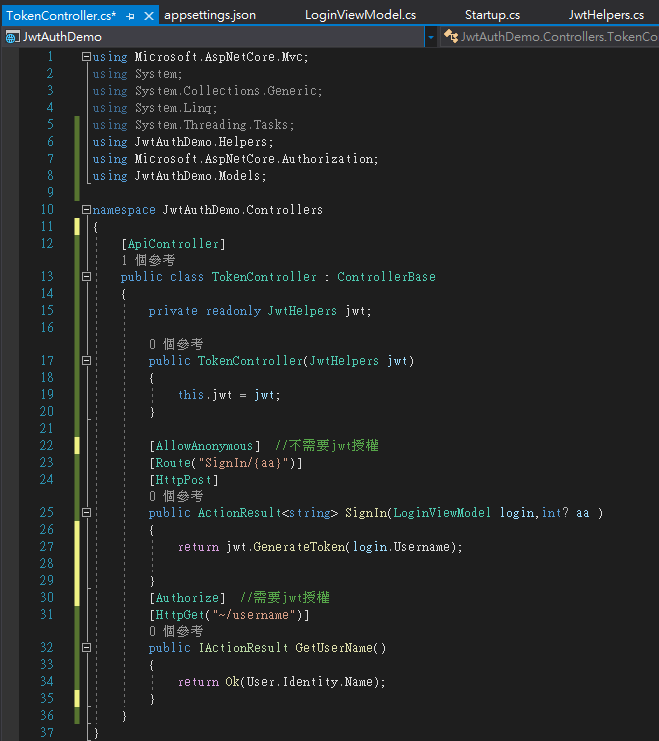
{

public string Username { get; set; }

public string Password { get; set; }

}

1. 建立 Controller



using Microsoft.AspNetCore.Mvc;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using JwtAuthDemo.Helpers;

using Microsoft.AspNetCore.Authorization;

using JwtAuthDemo.Models;

namespace JwtAuthDemo.Controllers

{

[ApiController]

public class TokenController : ControllerBase

{

private readonly JwtHelpers jwt;

public TokenController(JwtHelpers jwt)

{

this.jwt = jwt;

}

[AllowAnonymous] //不需要jwt授權

[Route("SignIn/{aa}")]

[HttpPost]

public ActionResult<string> SignIn(LoginViewModel login,int? aa )

{

return jwt.GenerateToken(login.Username);

}

[Authorize] //需要jwt授權

[HttpGet("~/username")]

public IActionResult GetUserName()

{

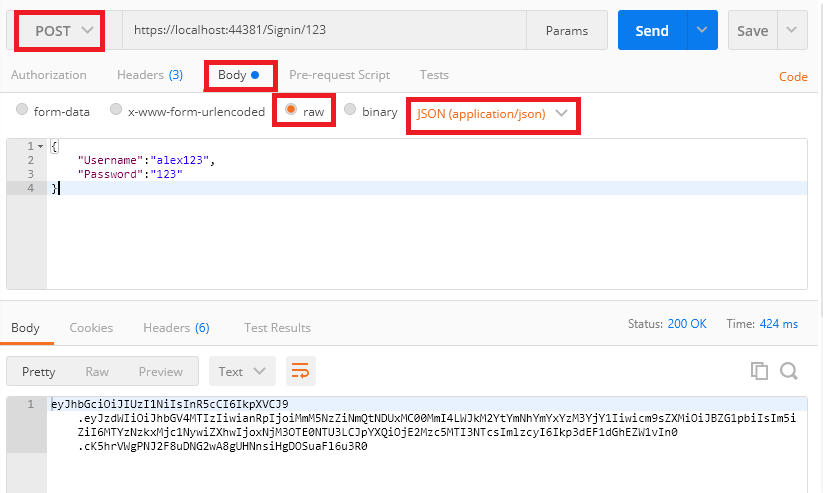
return Ok(User.Identity.Name);

}

}

}

1. 測試取得token



1. 測試取得user name

