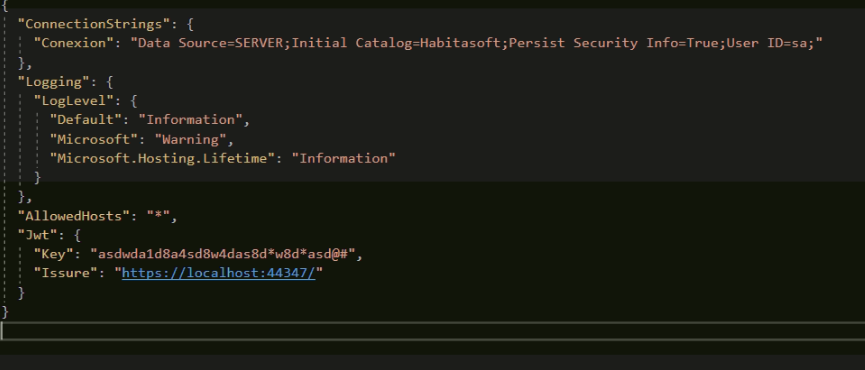
Documento que contiene información de configuración para que funcione correctamente la opción de generar un token en la API.

Proyecto plantilla: ApiComCuota.

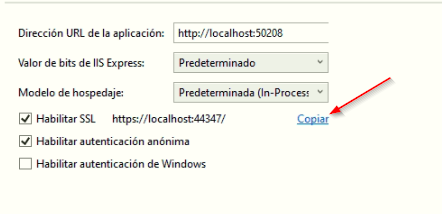
Para generar el token utilizaremos el servicio JWT.

1. Agregar en el fichero appsettings.json el siguiente código:

"Jwt": {"Key": "asdwda1d8a4sd8w4das8d\*w8d\*asd@#", "Issure": "https://localhost:44347/"}



La url que aparece en la propiedad “Issure” se obtiene: Botón derecho sobre el proyecto api – propiedades – pestaña Depurar y copiar la url que aparece al lado del check Habilitar SSL.



1. Fichero Starup.cs
   1. Añadir el siguiente código en public void ConfigureServices(IServiceCollection services)

// Agregamos servicio de intercambio de recursos, estamos permitiendo cualquier origen, cualquier header y cualquier metodos.

// Esto deberá ser controlado…

services.AddCors(options =>

{

options.AddPolicy("Todos", builder => builder.WithOrigins("\*").WithHeaders("\*").WithMethods("\*"));

});

var key = Encoding.ASCII.GetBytes(Configuration.GetValue<string>("Jwt:key"));

services.AddAuthentication(x =>

{

x.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;

x.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;

}).AddJwtBearer(x =>

{

x.RequireHttpsMetadata = false;

x.SaveToken = true;

x.TokenValidationParameters = new TokenValidationParameters

{

ValidateIssuerSigningKey = true,

IssuerSigningKey = new SymmetricSecurityKey(key),

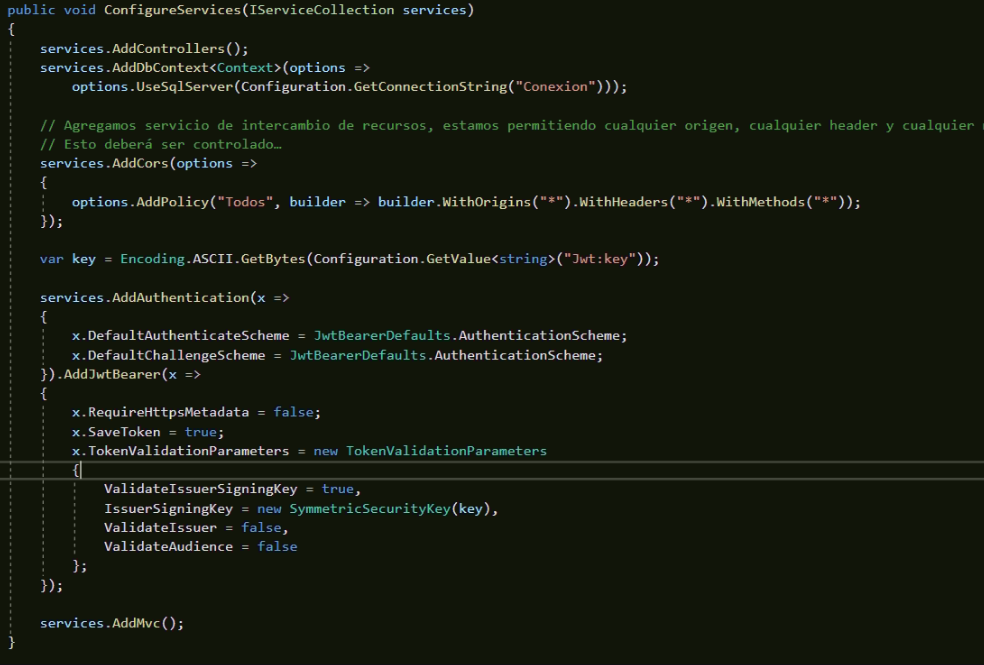
ValidateIssuer = false,

ValidateAudience = false

};

});

services.AddMvc();



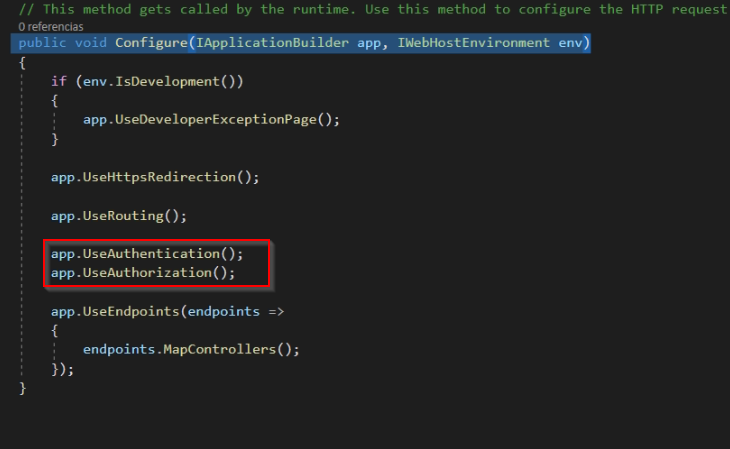
1. Fichero Starup.cs

Agregando app.UseAuthentication(); app.UseAuthorization(); en la función public void Configure(IApplicationBuilder app, IWebHostEnvironment env).

IMPORTANTE(Agregar el orden de autorizaciones como se ve en el cogió de la imagen.)

app.UseAuthentication();

app.UseAuthorization();



1. Crear un controlador de autorización como el de la plantilla. AuthController