# JWT Authentication

Definition:

JWT is used as a method to transmit information between different parties as a JSON object, this information is trustworthy because it is digitally signed using a secret public/private key pair through the RSA and ECDSA.

# Why would you use this?

JWT is used for authorising, where the user can access services after logging in with a specific token assigned to a specific authorisation level.

It is also used for information exchange between parties, which the user can be sure through that the data being received is by the sender that is who they say they are, and that the data has not been tampered with.

The advantages of using such a mechanism are that the server can find out who the user is and what permissions they have for the present services, what this basically does is eliminate the need for a session store.

# When would you use this?

JWT is used for authorising, where the user can access services after logging in with a specific token assigned to a specific authorisation level, and it is a particularly useful technology for API authentication and server to server authorization.

# How would you implement this?

Step 1:

Install the required nuget packages:

* Microsoft.EntityFrameworkCore.SqlServer
* Microsoft.EntityFrameworkCore.Tools
* Microsoft.IdentityModel.Tokens
* System.IdentityModel.Tokens.Jwt
* Microsoft.AspNetCore.Authentication.JwtBearer

Step 2:

Create a new class and name it as a AuthHelper that will host the GenerateToken() method and the AddJWtAuth method that allows the server to add the authentication options as well as the bearer options.

Step 3:

Create another Dto class that allows you to get the Email, Password and refresh token of the user for the request.

Step 4:

Create another Dto class that allows you to get the Refresh token, Message of the response, time of creation of token, Access token and time of expiration.

Step 5:

Create a class for the auth controller which includes methods for posting the authentication by using other defined methods for retrieving the required email and password corresponding to the inserted form details.

Diagram showing the authentication of JWT:

Diagram

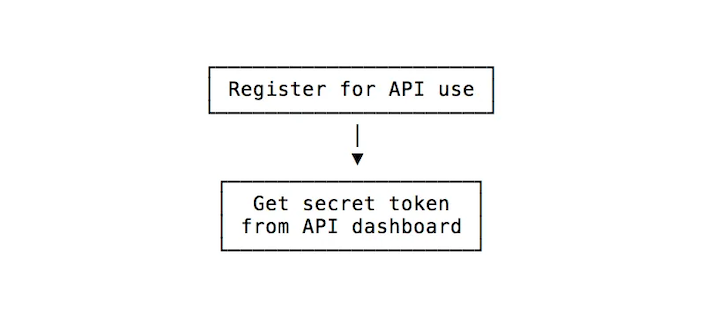
Description automatically generated

# Can you use JWT for Authentication?

A very common use for JWT, the only good one, is as an API authentication mechanism.

JWT technology is so popular and widely used that Google uses it to let you authenticate to its APIs.

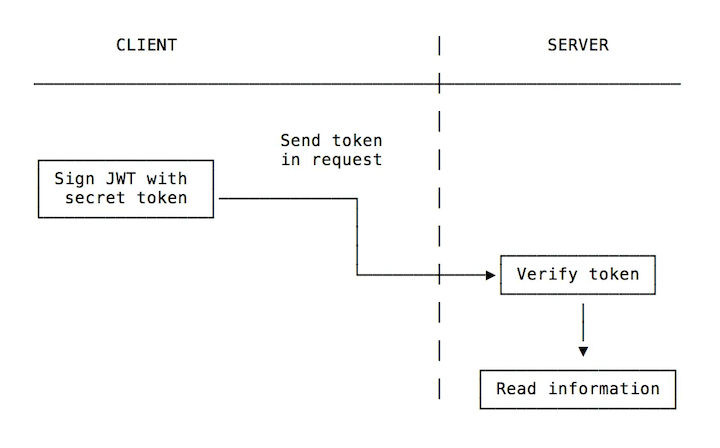
First you get a secret token from the service when you set up your Api:



.

Then the token can be created on the client side where it can be used as a unique identifier that can be passed into an Api request that allows you to read information specific to that user.

Diagram depicting the interaction between the Client and the server:



# Conclusion:

JWT is a good and trusted standard for trusting requests and using signatures, it is also used for securely exchanging information between parties that have the required authorization to access that information.