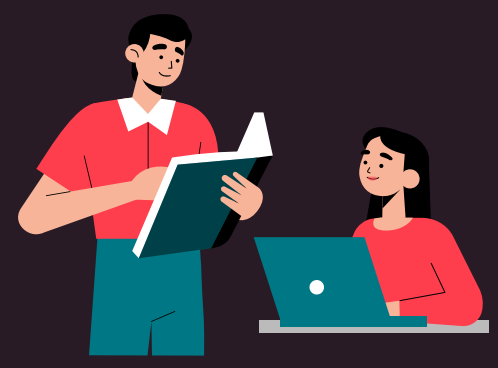
<Coding𝄞onata />

JWT Authentication vs OAuth 2.0

Aram Tchekrekjian

@AramT87



- What is JWT
- JWT Structure
- JWT Authentication
- OAuth 2.0
- OAuth 2.0 vs JWT Authentication
- JWT Authentication in ASP.NET Core Web API

1

What is JWT

JWT or Json Web Token is a **self-contained**, **digitally signed** and **base64UrlEncoded** format to exchange data between two entities

A JWT can contain key information (**claims**) about token itself, the user and the permissions (or roles) to be authorized to access your APIs

A JWT contains three parts separated by dots:
{Header}.{Payload}.{Signature}

2

JWT Structure

Header
Includes the algorithm used to sign/encrypt the JWT, usually it is HMAC SHA-256 or HS256

Payload
Contains the claims which are key,value pairs of info about the resource owner and some standard information like issuer, audience, subject, issued at and others

Signature
Encoding of the header and payload hashed using the header's algorithm with a secret

3

JWT Authentication

Using JWT Authentication you can **secure your API** using **JWT formatted bearer tokens**

The authentication starts by the client exchanging the user's login credentials with a JSON Web Token (JWT)

Then the client can access protected APIs by providing the JWT via the HTTP Authorization Header as Bearer Token

The resource server will have to validate the JWT itself

4

OAuth 2.0

OAuth 2.0 is an **authorization protocol** that provides secure delegated access to the clients to allow users to access protected data on the resource server

OAuth 2.0 can be used for **multi-sites and apps** authorization

There are several use cases to authenticate the user and the client, each use case is represented by an **Authorization Flow**

OAuth 2.0 defines 2 major types of clients:

- **Public Clients**
- **Confidential Clients**

5

JWT Authentication vs OAuth 2.0

You might wonder what is the difference between JWT authentication and OAuth 2.0?

Both provide secure delegated access for clients to allow users access protected data on the resource server

Both ways the client has to provide the Bearer Token (Access Token) via the Authorization Header

Both ways can use JWT as the bearer or access token format

6

JWT Authentication vs OAuth 2.0

The main difference is **in the way the client would authenticate the user and retrieve the access token**

7

JWT Authentication vs OAuth 2.0

JWT Authentication **doesn't constrain you for how to authenticate the client/user**, since the implementation can be done and customized per need

JWT Authentication allows you to **run your own authorization server** and generate JWT access tokens and applying the claims for your users

JWT Authentication is mainly used for **simpler implementations for authentication scenarios**

8

JWT Authentication in ASP.NET Core Web API

In ASP.NET Core, using the **JWTBearer library**, you can easily build and host a simple authorization server and secure your Web APIs without having to rely on setting up a full suited 3rd-party authorization server or identity provider

With **policy-based authorization**, you can implement fine grained authorization for your users to access your APIs

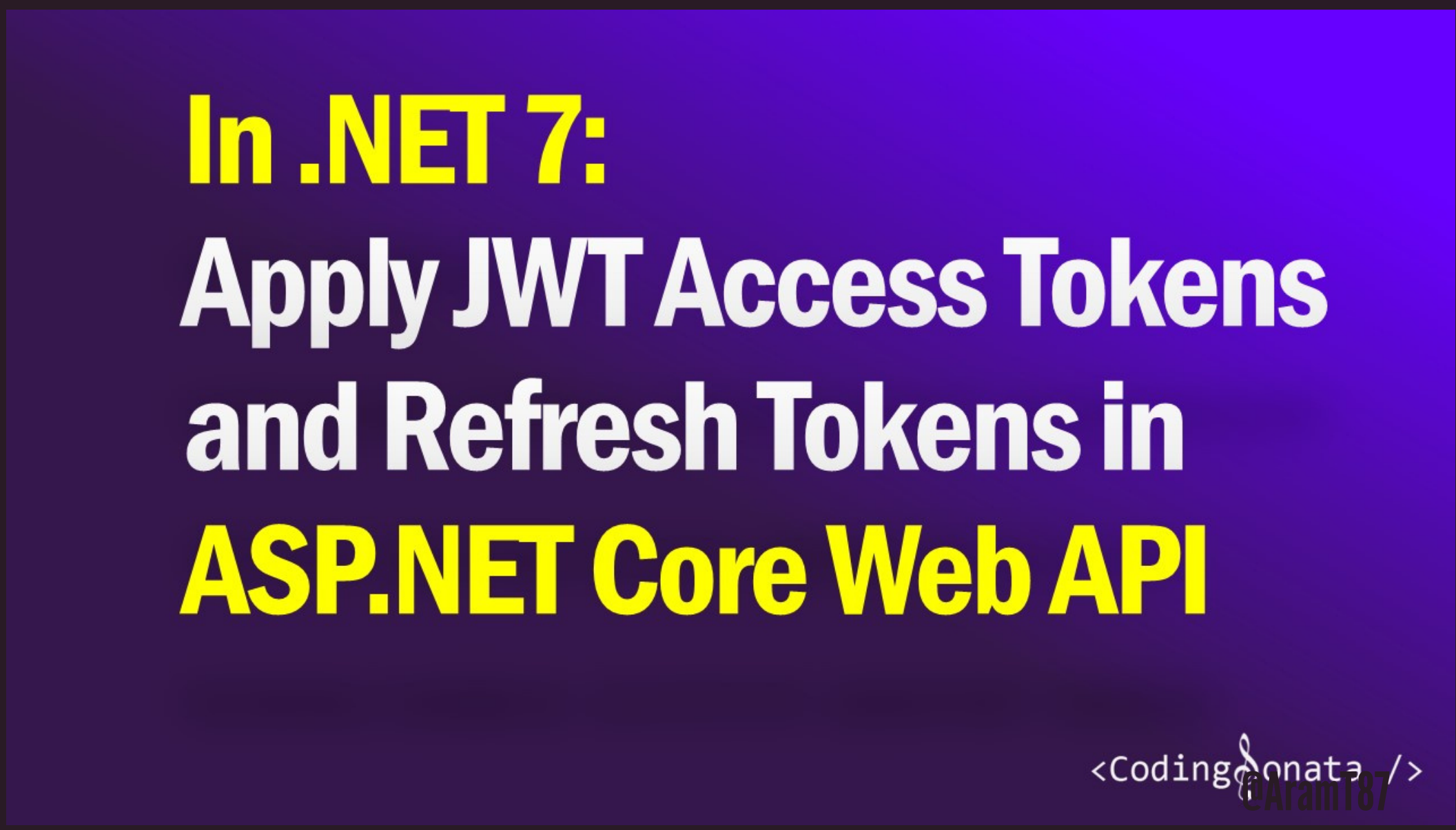
A policy is a set of **requirements** that can be either **predefined roles** or **customized requirements** that you can override

9

JWT Authentication in ASP.NET Core Web API

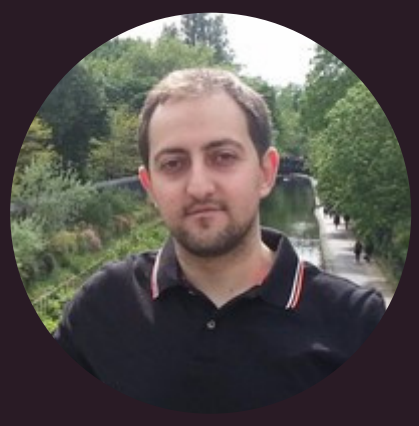
Bonus: Learn how to apply JWT Access Tokens & Refresh Tokens in **ASP.NET Core Web API 7**

Tutorial Link in the comments



Thank You

Follow me for more content



Aram Tchekrekjian

AramT87

CodingSonata.com/newsletters

<Coding𝄞onata />