JSON Web Token(JWT)

A JSON Web Token (JWT) is an open standard for securely transmitting information as a JSON object. The information can be verified and trusted because it is digitally signed. JWTs use either a secret key (HMAC algorithm) or a public/private key pair (RSA or ECDSA) for signing.

JWTs can also be encrypted for secrecy, but this explanation focuses on signed tokens. Signed tokens ensure the integrity of claims, while encrypted tokens conceal those claims.

Scenarios where JSON Web Tokens are useful:

- **Authorization**: This is the most common scenario for using JWT.
- **Information Exchange**: Securely transmit data between parties.

JWT Structure:

In its compact form, JSON Web Tokens consist of three parts separated by dots (.), which are:

1. Header
2. Payload
3. Signature

Header:

The header specifies the token type (JWT) and the signing algorithm (e.g., HMAC SHA256 or RSA).

For Example:

```
{
 "alg": "HS256",
 "typ": "JWT"
}
```

Payload:

The payload contains claims, which are statements about the user or other data.
Claims can be:
Registered Claims: Predefined fields (e.g., iss, exp).
Public Claims: Custom claims defined by the user.
Private Claims: Agreed upon between parties.
An example payload could be:

```
{
 "sub": "1234567890",
 "name": "John Doe",
 "admin": true
}
```

Signature:
The signature is generated by combining:
Encoded header.
Encoded payload.
A secret key.
The specified algorithm.

For example if you want to use the HMAC SHA256 algorithm, the signature will be created in the following way:

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret)
```