JWT-Based Authentication

(Implementation in Node.js)

Strategies for authentication

- **Session-based**: Information of logged in user maintained on the server. A cookie is set to identify a logged-in user. Every request from client has the cookie which is used to identify the user.

- **JWT-based**: A token is generated and passed to client. The token has “claims” which identify what the user is authorized to do. Every request from client has the token. The token has a server-side encrypted form of the claims – so it is not possible to tamper with it.

- **Oauth-based**: Uses third-party authentication providers like Google, Facebook, GitHub etc.

Strategies for authentication

- **Session-based**: Information of logged in user maintained on the server. A cookie is set to identify a logged-in user. Every request from client has the cookie which is used to identify the user.

- **JWT-based**: A token is generated and passed to client. The token has “claims” which identify what the user is authorized to do. Every request from client has the token. The token has a server-side encrypted form of the claims – so it is not possible to tamper with it.

- **Oauth-based**: Uses third-party authentication providers like Google, Facebook, GitHub etc.

JWT (Token)

- JSON Web Token (JWT) is a standard

- A JWT (token) has 3 parts
  - **Header**: Information about encryption algorithm (HS256, RS256) etc.
  - **Payload (Claims)**: Usually a JSON-formatted string with details of what user is authorized to do
  - **Signature:** The first 2 parts which are encrypted using a private key stored on the server

- All 3 parts are base64 encoded (a string encoding that uses 64 characters).

Header

Information about encryption algorithm (HS256, RS256) etc.

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

base64 encoded form*:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

* Sample only, not actual converted string

Payload (Claims)

Usually a JSON-formatted string with details of what user is authorized to do


```
{
   "sub": "1234567890",
   "name": "John Doe",
   "iat": 1516239022,
   "isAdmin": true
}
```


base64 encoded form*:
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ

* Sample only, not actual converted string

Signature

The first 2 parts (header.payload) are concatenated and encrypted using a private key stored on the server

base64 encoded form*:
0QsiCvdZOPsSkn4NJoPGAEtKOLptJXhWo1Js8zmO834

* Sample only, not actual converted string

The JWT (token)

- The 3 parts are combined with dot (.) as delimiter

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.0QsiCvdZOPsSkn4NJoPGAEtKOLptJXhWo1Js8zmO834

Further reading

- https://jwt.io/
- https://flaviocopes.com/jwt/
- https://www.npmjs.com/package/jsonwebtoken
- https://medium.com/devgorilla/how-to-log-out-when-using-jwt-a8c7823e8a6