6주차 04 - JWT

쿠키

- 장점 : 서버가 저장을 안 함, Stateless
- 단점 : 보안에 취약

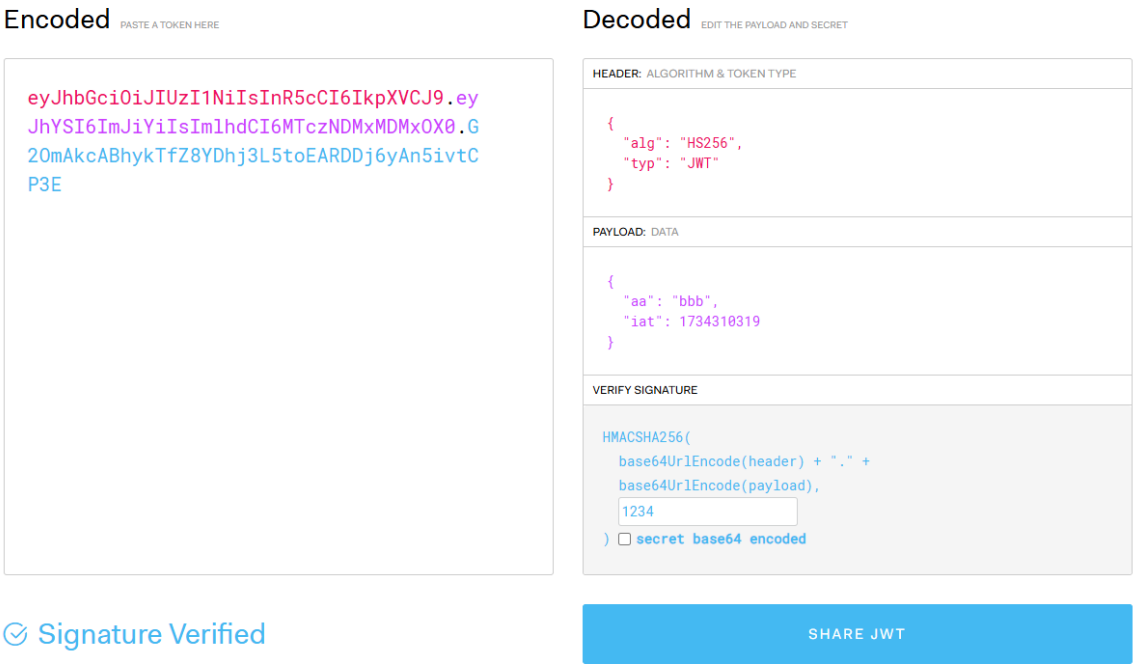
세션

- 장점 : 보안이 비교적 좋다
- 단점 : 서버가 저장, Stateless하지 않음

JWT

- Json Web Token
- 개념 : JSON 형태의 데이터를 안전하게 전송하기 위한 토큰
  - 토큰 : 인증용, 인가용
- **장점** :
  - 보안에 강하다 → 암호화가 되어 있다.
  - HTTP 특징을 잘 따랐다. → 서버가 상태를 저장하지 않는다.
  - 서버 부담을 줄여줄 수 있다.
  - 토큰을 발행하는 서버를 따로 만들 수 가 있다.
- **구조** :
  - HEADER
    - 알고리즘 : 토큰을 암호화하는데 어떤 알고리즘을 사용하였는가
    - 타입 : 토큰의 타입(주로 JWT)
  - PAYLOAD
    - 전달하고자 하는 데이터 내용
  - VERIFY SIGNATURE

    - 만약 페이로드 값이 바뀌면 서명값 또한 바뀌기 때문에 보안을 더 확실히 해준다.
- 실습



로그인 절차

1. 로그인
2. 서버에서 jwt 토큰 발행
3. 그 후로는 토큰으로 본인인지 확인

.env 파일

- 개발을 하다가 포트넘버, 데이터베이스 계정, 암호키 등등 외부에 유출 되면 안되는 환경변수를 따로 관리하기 위한 파일
- PRIVATE_KEY = '1234'; 이런 형식
    - 카멜 케이스 사용
- '#' 뒤에 주석
- 사용 방법

```
var dotenv = require('dotenv');

dotenv.config();

var token = jwt.sign({aa : 'bbb'},process.env.PRIVATE_KEY)
```

토큰 사용 실습

```
const token = jwt.sign({
                        userEmail : loginInfo.userEmail,
                        name : userInfo.name
                    },process.env.PRIVATE_KEY,{
                        expiresIn : '30m',
                        issuer : '1234'
                    });

                    res.cookie("token",token,{
                        httpOnly : true
                    });
```

jwt.sign(페이로드,암호화키,{만료기간, 발행자})