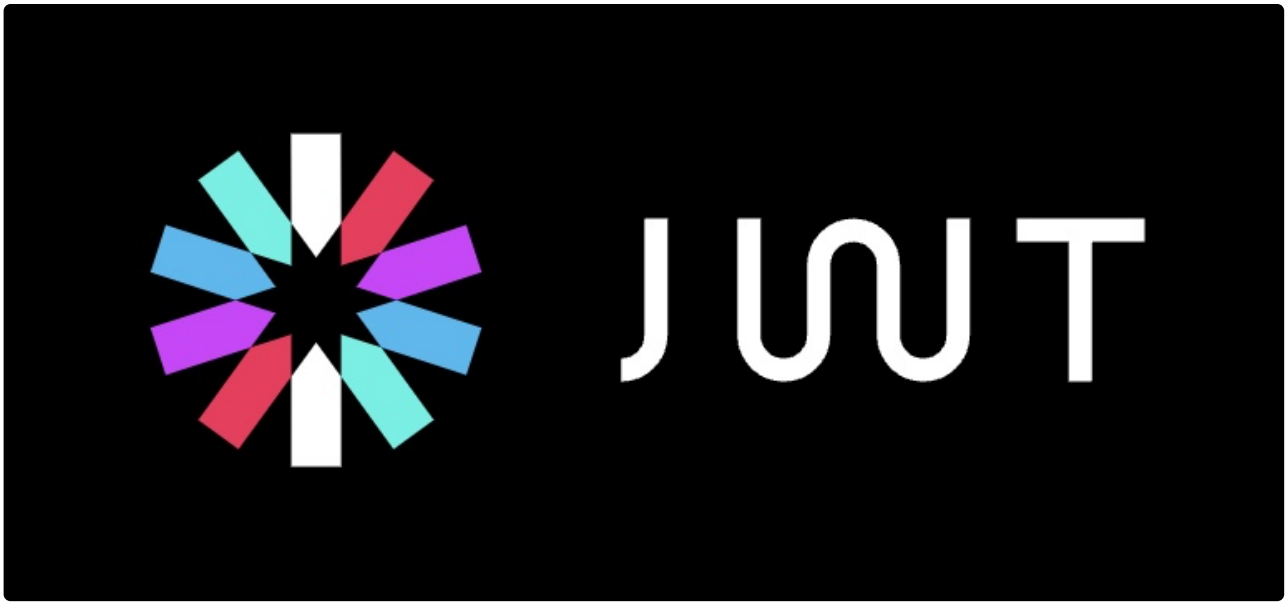
JWT(JSON Web Token)



JWT(JSON Web Token) 란?

JSON Web Token은 두 개체 간에 안전하게 클레임을 나타내기 위한 업계 표준인 RFC 7519 방법입니다.

JSON(JavaScript Object Notation) 형식으로 인코딩된 정보를 사용하여 발행자(Publisher), 제목(Header), 청구(Claim) 및 서명(Signature)으로 이루어진 토큰입니다.

JWT의 장단점

장점

1. 무상태(Stateless): JWT는 서버 측에서 인증 정보를 유지할 필요가 없습니다. 따라서 서버에 부하를 줄이고, 확장성을 높일 수 있습니다.
2. 분산 시스템에 적합: JWT는 분산 시스템에서 인증 정보를 안전하게 전송하는 데 적합합니다. 각각의 서버는 JWT의 서명을 검증하여 인증 정보를 확인할 수 있습니다.
3. 보안: JWT는 서명을 사용하여 안전하게 인증 정보를 전송합니다. 또한 JWT의 발행자, 제목 및 청구 항목을 검증하여 위조를 방지할 수 있습니다.
4. 표준 기술: JWT는 표준 기술로, 다양한 프로그래밍 언어와 플랫폼에서 지원됩니다. 따라서 JWT를 사용하면 다양한 환경에서 인증 정보를 안전하게 전송할 수 있습니다.

단점

1. JWT의 크기: JWT는 정보를 Base64로 인코딩하여 전송합니다. 따라서 JWT의 크기가 커질 수 있습니다. 이는 대량의 트래픽이 발생하는 시스템에서 부하를 초래할 수 있습니다.
2. 보안: JWT는 서명을 사용하여 인증 정보를 보호하지만, 서명 키가 유출되면 JWT가 위조될 수 있습니다. 따라서 서명 키를 안전하게 보관해야 합니다.

JWT의 구성요소

JWT는 크게 세 개의 구성요소로 이루어져 있습니다: Header, Payload, Signature입니다.

1. Header: JWT가 어떤 알고리즘을 사용하여 서명되었는지와 어떤 타입의 토큰인지를 나타냅니다. Header는 JSON 객체로 이루어져 있으며, 다음과 같은 형식을 가집니다.

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

"alg"는 서명 알고리즘을 의미하며, "typ"은 토큰의 타입을 의미합니다.

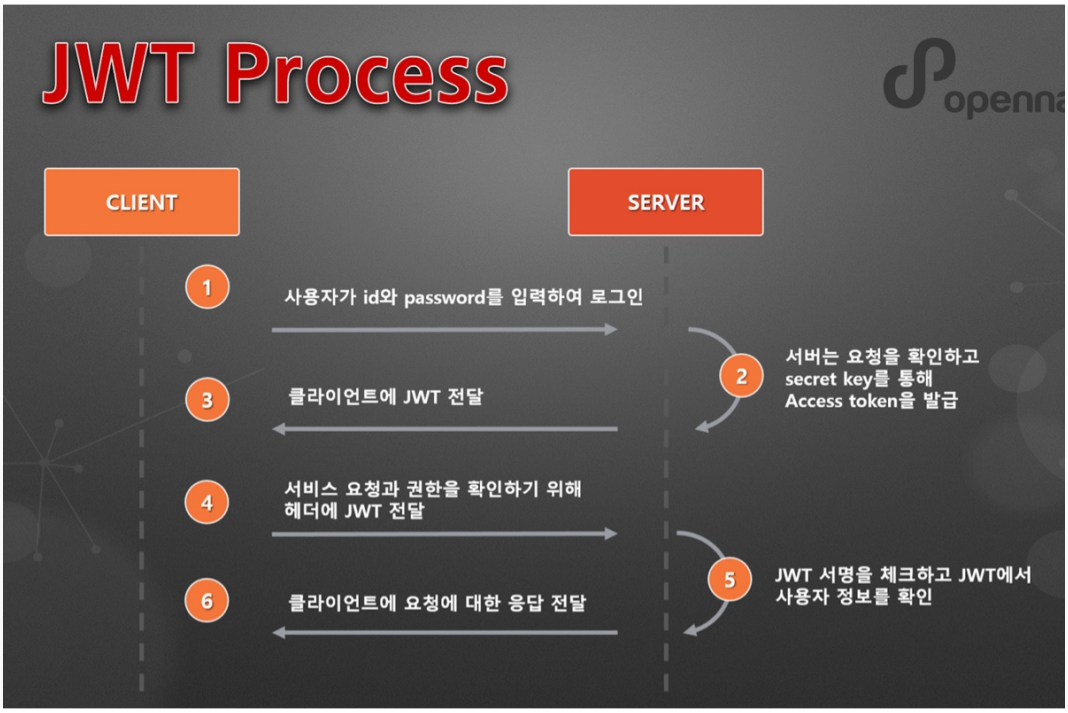
2. Payload: JWT에 포함된 정보를 담고 있습니다. 이는 사용자의 ID나 권한 등의 정보를 포함할 수 있습니다. Payload 역시 JSON 객체로 이루어져 있으며, 다음과 같은 형식을 가집니다.

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

"sub"는 주제(Subject)를 의미하며, "name"은 사용자의 이름을 의미합니다. "iat"은 발행된 시간을 의미합니다.

3. Signature: Header와 Payload의 내용을 조합한 후, 서버에서 발급된 비밀키를 사용하여 서명됩니다. 이 서명을 통해 JWT의 무결성을 검증할 수 있습니다. Signature는 Header와 Payload를 Base64 인코딩한 문자열과 비밀키를 사용하여 생성됩니다.

JWT의 작동 방식



1. 사용자가 인증을 시도
2. 서버는 사용자 정보를 확인 후 secret key를 통해 Access token을 발급
3. 서버는 사용자가 인증되었다는 정보와 함께 JWT를 발행합니다. JWT에는 사용자가 인증된 것을 증명하는 정보가 포함됩니다.
4. 서버는 JWT를 클라이언트에게 전송합니다. 클라이언트는 요청을 보낼 때마다 JWT를 포함하여 서버에 전송합니다.
5. 서버는 JWT에 포함된 정보를 검증
6. 요청이 유효한 것으로 확인되면 요청을 처리합니다.