토큰이란?

블록체인 암호화폐도 토큰이라는 말을 씁니다만 여기에서 다룰 것은 블록체인 토큰이 아니고 로그인을 하거나 사이트를 이용할 때 토큰입니다.

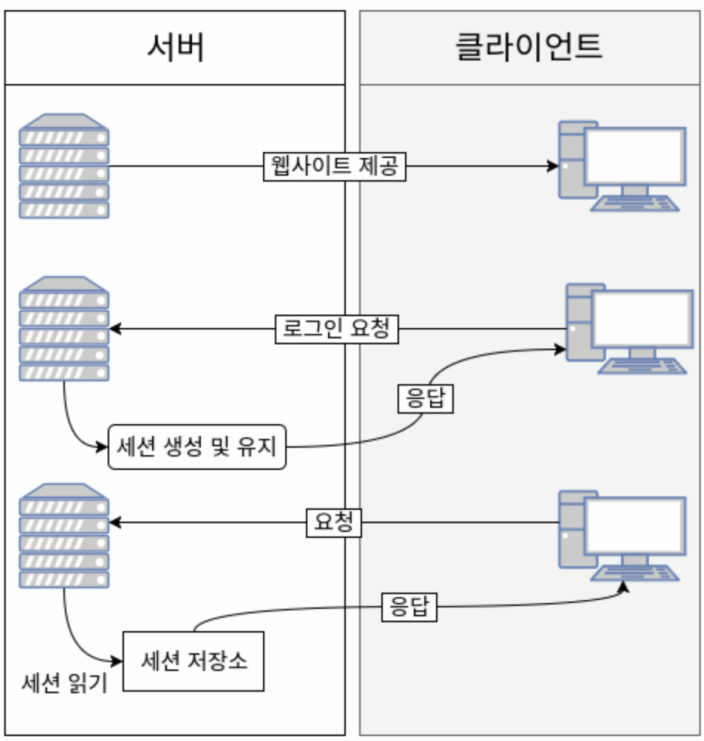
토큰은 말 그대로 동전이란 뜻입니다. 일종에 버스카드나 교통카드라고 생각하면 됩니다.
그런데 동전하고는 조금 다릅니다. 동전 처럼 생겼지만 시장에 가서 물건을 사거나 할 수는 없고 버스를 탈 때 동전 대신 넣듯이 토큰은 특정한 목적에만 사용할 수 있습니다. 돈 보다는 바우처하고 비슷합니다.
토큰은 일종에 권리를 주는 것입니다. 버스를 탈 수 있는 권리, 지하철을 탈 수 있는 권리를 부여 해주는것입니다. 이 권리를 받는 방법은 토큰의 경우 돈을 냈느냐 안냈느냐 입니다. 돈을 내면 토큰을 받고 이 토큰을 내고 버스에 탈 수 있습니다.
토큰은 이미 없어진지 오래 되었고 동전도 점점 없어지고 있어서 토큰의 의미를 직관적으로 이해하기가 어렵게 되었지만 의미 자체는 '권리'입니다.
점점 이 토큰이라는 것이 우리 눈에 보이지 않게 클라이언트와 서버 단에 코드로 왔다갔다 하고 있는게 지금 기술의 발전 단계입니.

로그인 할 때 토큰이란?

저는 서버 개발을 하고 있습니다. 서버에서는 사용자가 결제한 돈도 왔다갔다 합니다. 그렇기 때문에 '권한'이라는 것이 중요합니다. 인터넷에 사이트를 올리면 전 세계에서 접근 할 수 있기 때문에 사용자나 해커나 이 사이트에 접근을 할 수 있습니다.
그래서 사용자들의 돈을 보호 하기 위해 본인만 돈을 넣고 빼고 주문을 넣고 결제를 할 수 있게 만들어야 합니다. 이 토큰은 본인 확인 수단입니다. ① 로그인을 할 때 id와 pw를 넣고 로그인을 하면 서버가 그것을 확인 해서 id와 pw가 맞으면 이 사용자가 유효한 사용자라는 토큰을 발행 해줍니다. ← 서버는 한번 사용자의 정보를 "메모리"에 저장하고 있을 필요가 없다!!
② 그러면 사용자들은 이 토큰을 가지고 해당 사이트의 여러 버스에 해당하는 서비스들을 이용 할 수 있는 것입니다.
↑ 토큰은 '열쇠'의 역할.

[서버 기반 인증 시스템이란?]

기존의 인증 시스템은 서버 기반의 인증 방식으로, 서버 측에서 사용자들의 정보를 기억하고 있어야 한다. 사용자들의 정보를 기억하기 위해서는 세션을 유지해야 하는데, 메모리나 디스크 또는 데이터베이스 등을 통해 관리한다. 서버 기반의 인증 시스템은 클라이언트로부터 요청을 받으면, 클라이언트의 상태를 계속해서 유지하고 이 정보를 서비스에 이용하는데, 이러한 서버를 Sateful 서버라고 한다. 예를 들어 사용자가 로그인을 하면, 세션에 사용자 정보를 저장해두고 서비스를 제공할 때 사용하곤 한다. 이러한 서버 기반의 시스템은 다음과 같은 흐름을 갖는다.



이러한 인증 장식은 소규모 시스템에서는 아직 많이 사용되고 있지만, 웹/앱 어플리케이션이 발달하게 되면서 서버를 확장하기가 어렵다는 등 다음과 같은 문제점을 보이기 시작했다.

1. 세션
   사용자가 인증을 할 때, 서버는 이러한 정보를 저장해야 하고 이를 세션(Session)이라고 부른다. 대부분의 경우에는 메모리에 저장하는데, 로그인 중인 사용자가 늘어날 경우에는 서버의 RAM에 부하가 걸리게 된다. 이를 피하기 위해 데이터베이스에 저장을 하기도 하는데, 이러한 방식 역시 데이터베이스에 무리를 줄 수 있다.
2. 확장성
   사용자가 늘어나게 되면 더 많은 트래픽을 처리하기 위해 여러 프로세스를 돌리거나 컴퓨터를 추가하는 등 서버를 확장해야 한다. 세션을 사용한다면 세션을 분산시키는 시스템을 설계해야 하지만 이러한 과정은 매우 어렵고 복잡한다.
3. CORS(Cross-Origin Resource Sharing)
   웹 어플리케이션에서 세션을 관리할 때 자주 사용되는 쿠키는 단일 도메인 및 서브 도메인에서만 작동하도록 설계되어 있다. 따라서 쿠키를 여러 도메인에서 관리하는 것은 번거롭다.

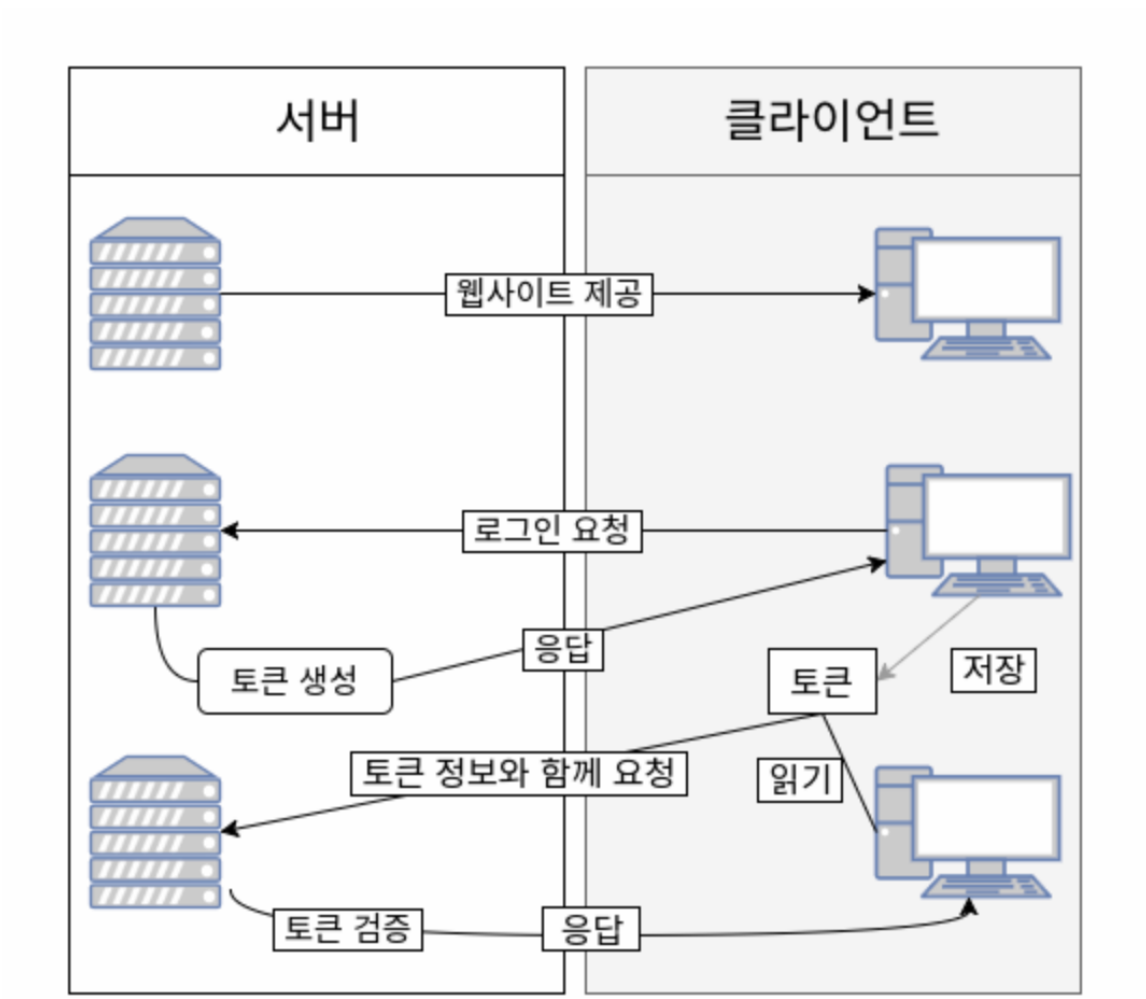
이러한 문제들 때문에 도큰 기반의 인증 시스템을 사용하게 되었다.

[토큰 기반 인증 시스템이란?]

토큰 기반의 인증 시스템은 인증받은 사용자들에게 토큰을 발급하고, 서버에 요청을 할 때 헤더에 토큰을 함께 보내도록 하여 유효성 검사를 한다. 이러한 시스템에서는 더이상 사용자의 인증 정보를 서버나 세션에 유지하지 않고 클라이언트 측에서 들어오는 요청만으로 작업을 처리한다. 즉, 서버 기반의 인증 시스템과 달리 상태를 유지하지 않으므로 Stateless한 구조를 갖는다. 이러한 토큰 기반의 인증 방식을 통해 수많은 문제점들을 해결할 수 있는데, 대표적으로 사용자가 로그인이 되어있는지 안되어있는지 신경쓰지 않고 손쉽게 시스템을 확장할 수 있다.

이러한 토큰 기반의 인증 시스템은 아래와 같은 과정으로 작동한다.

1. 사용자가 아이디와 비밀번호로 로그인을 한다.
2. 서버 측에서 해당 정보를 검증한다.
3. 정보가 정확하다면 서버 측에서 사용자에게 Signed 토큰을 발급한다. (Signed는 해당 토큰이 서버에서 정상적으로 발급된 토큰임을 증명하는 Signature를 가지고 있다는 것)
4. 클라이언트 측에서 전달받은 토큰을 저장해두고, 서버에 요청을 할 때마다 해당 토큰을 서버에 함께 전달한다. 이때 Http 요청 헤더에 토큰을 포함시킨다.
5. 서버는 토큰을 검증하고, 요청에 응답한다.



[토큰 기반 인증 시스템의 이점]

1. **무상태성(Stateless) & 확장성(Scalability)**
   토큰은 클라이언트 측에 저장되기 때문에 서버는 완전히 Stateless하며, 클라이언트와 서버의 연결고리가 없기 때문에 확장하기에 매우 적합하다. 만약 사용자 정보가 서버 측 세션에 저장된 경우에 서버를 확장하여 분산처리 한다면, 해당 사용자는 처음 로그인 했었던 서버에만 요청을 받도록 설정을 해주어야 한다. 하지만 토큰을 사용한다면 어떠한 서버로 요청이 와도 상관이 없다.
2. **보안성**
   클라이언트가 서버로 요청을 보낼 때 더 이상 쿠키를 전달하지 않으므로, 쿠키 사용에 의한 취약점이 사라지게 된다. 하지만 토큰 환경의 취약점이 존재할 수 있으므로 이에 대비해야 한다.
3. **확장성(Extensibility)**
   시스템의 확장성을 의미하는 Scalability와 달리 Extensibility는 로그인 정보가 사용되는 분야의 확정을 의미한다. 토큰 기반의 인증 시스템에서는 토큰에 선택적인 권한만 부여하여 발급할 수 있으며 OAuth의 경우 Facebook, Google 등과 같은 소셜 계정을 이용하여 다른 웹서비스에서도 로그인을 할 수 있다.
4. **여러 플랫폼 및 도메인**
   서버 기반 인증 시스템의 문제점 중 하나인 CORS를 해결할 수 있는데, 애플리케이션과 서비스의 규모가 커지면 여러 디바이스를 호환시키고 더 많은 종류의 서비스를 제공하게 된다. 토큰을 사용한다면 어떤 디바이스, 어떤 도메인에서도 토큰의 유효성 검사를 진행한 후에 요청을 처리할 수 있다. 이런 구조를 통해 assests 파일(Image, html, css, js 등)은 모두 CDN에서 제공하고, 서버 측에서는 API만 다루도록 설계할 수 있다.

최근에는 Json 포맷을 이용하는 JWT(Json Web Token)을 주로 사용한다. JWT 토큰에 대한 내용은 아래의 포스팅을 참고하면 된다.