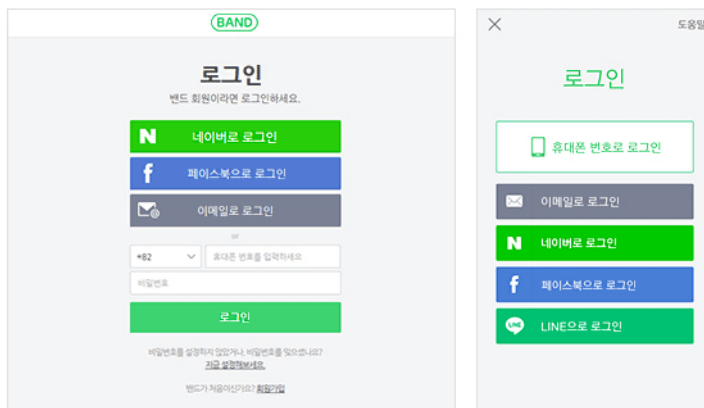


OAuth란 무엇일까

정리중

아래와 같은 로그인 창을 보셨을 것입니다. 별도의 회원가입 없이 로그인을 제공하는 플랫폼의 아이디만 있으면 서비스를 이용 할 수 있습니다. 외부 서비스에서도 인증을 가능하게 하고 그 서비스의 API를 이용하게 해주는 것, 이것을 바로 OAuth라고 합니다.



출처: naver developer

OAuth란

OAuth 2.0은 다양한 플랫폼 환경에서 권한 부여를 위한 산업 표준 프로토콜입니다.

간단하게 인증(Authentication)과 권한(Authorization)을 획득하는 것으로 볼 수 있습니다.

- 인증은 시스템 접근을 확인하는 것 (로그인) -> 인증만 하는 것은 openID
- 권한은 행위의 권리를 검증하는 것

OAuth의 배경

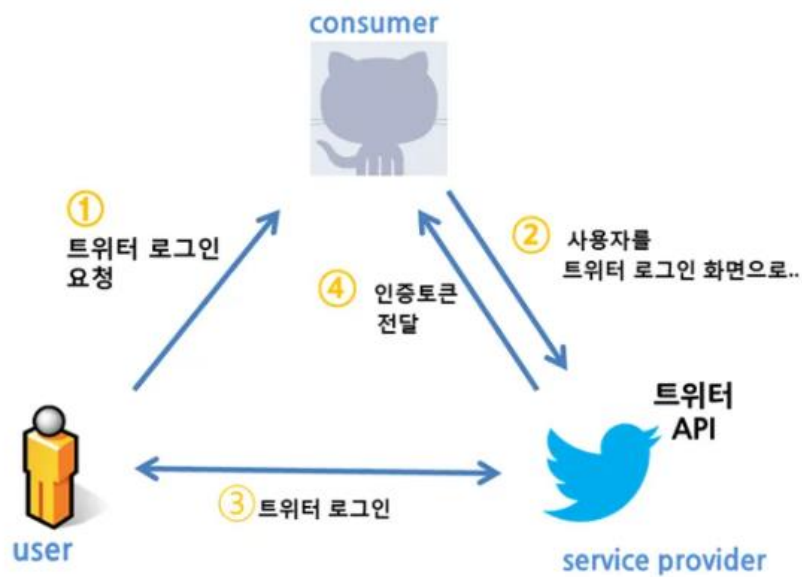
third party Application에 아이디와 비밀번호를 제공하고 싶지 않은 요구가 첫번째입니다. 개인정보를 여러 곳에 입력하면서 피싱에 둔감해지고 무엇보다 Application이 안전하다는 보장이 없기 때문에 보안에 취약했습니다.

당시에는 인증과 권한을 부여하는 요구를 만족 시킬 수 있는 인증방식이 없어서 Twitter의 주도로 OAuth 1.0이 탄생하였습니다.

비밀번호 인증방식의 문제

- 신뢰: 사용자가 애플리케이션에 ID/PW를 제공하기 꺼려함
- 피싱에 둔감해짐: 각 종 애플리케이션들에 ID/PW 를 계속 제공하는 경우
- 접근범위가 늘어남에 따른 위험 부담: ID/PW를 모두 알고 있는 애플리케이션은 모든 권한을 가짐
- 신뢰성의 제한: PW 를 변경한다면 애플리케이션은 동작을 하지 못하게 됨
- 폐기문제: 권한을 폐기할 수 있는 유일한 방법이 PW를 변경하는 것,

OAuth 구성 (1.0a)



3-legged-auth

유저(user), 컨슈머(consumer), 서비스 프로바이더(service provider)

그런데 OAuth1.0은..

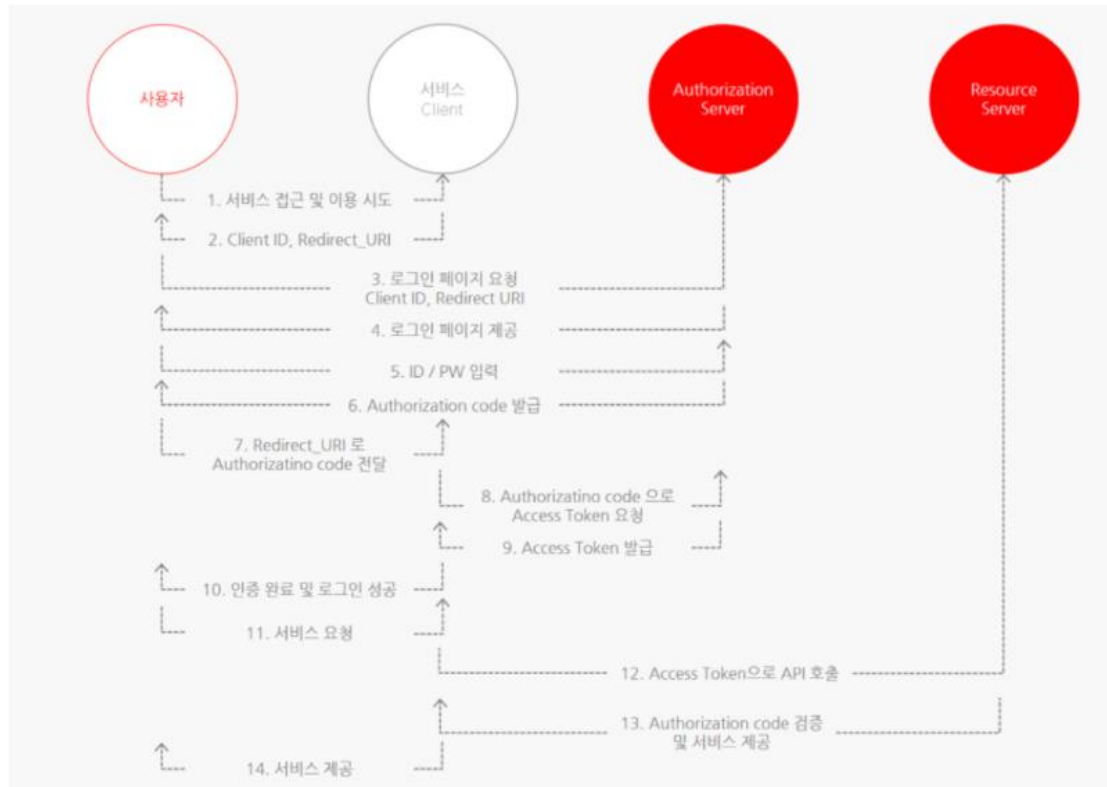
구현이 복잡하고 웹이 아닌 어플리케이션에서의 지원이 부족하였습니다. HMAC을 통해 암호화를 하는 번거로운 과정을 겪습니다. 또한 인증토큰(access token)이 만료되지 않습니다

구성은 이렇게 4가지로 존재한다.

OAuth의 원리와 과정

- 1) Resource Owner(사용자)
- 2) Authorization Server(인증 서버)
- 3) Resource Server(REST API)
- 4) Client

2. Oauth 동작방식



OAuth의 주체는 3가지로 이루어져 있다.

Resource owner(사용자) - client(Wep/App) - resource server(제공자 : google, facebook, kakao, naver etc)

[구글 API를 통해서 배우는 인증 \(oauth 2.0\) - 생활코딩 \(opentutorials.org\)](#)

이거 참조해서, 어떻게 개발되는지 확인하셈.

3. OAuth 특징과 기능

- 보안이 철저하게 보장된다.
- 대신 그 절차가 복잡하다.(이용하기 어렵다.)

웹 애플리케이션이 아닌 애플리케이션 지원 강화

암호화가 필요 없음 HTTPS를 사용하고 HMAC을 사용하지 않는다.

Signature 단순화 정렬과 URL 인코딩이 필요 없다.

Access Token 갱신 OAuth 1.0에서 Access Token을 받으면 Access Token을 계속 사용할 수 있었다. 트위터의 경우에는 Access Token을 만료시키지 않는다. OAuth 2.0에서는 보안 강화를 위해 Access Token의 Life-time을 지정할 수 있도록 했다.

• • •

Firebase란 구글에서 인수한 모바일 및 웹 애플리케이션 개발 플랫폼입니다.

이를 사용하면 서버 관리를 구글에서 해주기 때문에 편리하고, Firebase 하나만으로도 앱 하나를 만들 수 있습니다.

이러한 Firebase에서 제공되는 Firebase 서비스 중 하나인 Authentication는 앱에서 사용자 인증 시 필요한 백엔드 서비스와 사용하기 쉬운 SDK, 기성 UI를 제공하는 라이브러리입니다.

예를 들어, Firebase Authentication를 이용해 Firebase를 안드로이드 스튜디오에 연결하고 Google 로그인을 통해 앱에서 사용자의 신원 정보 등을 얻을 수 있습니다.