Programming

쉽게 알아보는 서버 인증 3편(SNS 로그인, OAuth 2.0)

이호연 자유로운 오랑우탄 | 2018. 8. 11. 22:57

안녕하세요 여러분. 오늘은 "쉽게 알아보는 인증"의 마지막 편인 SNS 로그인 과 OAuth에 대해 써보려고 합니다.

만일 SNS 로그인을 한 번이라도 구현해보려고 하셨다면, OAuth을 한 번은 들어보셨을 겁니다. 그래서 SNS 로그인 = OAuth라고 생각하시는 분이 계시는데 이는 잘못된 생각입니다. OAuth 프로토콜의 기능 중 하나로 SNS 로그인이 있는겁니다!

지금부터 OAuth 의 정의를 시작으로 어떤 방식으로 SNS 로그인이 작동되는지 차근차근 알아보도록 하겠습니다.

참고 포스팅

> http://tansfil.tistory.com/58 (세션/쿠키, JWT를 이용한 인증)
> http://tansfil.tistory.com/59 (Access Token + Refresh Token을 이용한 인증)

Oauth

Oauth 는 외부서비스의 인증 및 권한부여를 관리하는 범용적인 프로토콜입니다.

* 권한 : OAuth는 인증뿐만 아니라 권한도 관리합니다. 사용자의 권한에 따라 접근할 수 있는 데이터가 다르도록 설정이 가능합니다.

* 프로토콜 : 특정한 프로그램을 지칭하는게 아니라 일종의 규격입니다. Facebook, Google, Naver 등은 OAuth라는 규격에 맞춰 인증 및 권한을 대행관리 해줍니다

* 외부서비스 : 우리가 만들고 있는 서비스를 이야기합니다. 외부 서비스를 위한 서비스인 OAuth는 우리 서비스의 인증 및 권한부여를 관리를 대행해줍니다.

한 가지 명심해야할 점은, 우리가 배웠던 (사용자 <-> 어플리케이션 서버) 인증 절차였던 세션/쿠키, 토큰 기반 인증 방식을 완전히 대체하는게 아니라는 점입니다. 즉 SNS 로그인 기능을 넣더라도 결국은 세션/쿠키 방식이나 토큰을 활용해 인증을 거쳐야 합니다. 이 부분은 아래에서 더 상세하게 다루겠습니다.

OAuth 2.0

현재 대다수가 사용하고 있는 Oauth는 2.0 버전입니다.

2007년 처음으로 Oauth 1.0의 초안이 발표되었고 그 뒤로는 사람들에게 많이 알려지게 되었습니다. 그러나 점점 커져가는 네트워크 시장에서 한계가 나타나기 시작했고 2012년 Oauth 2.0을 새롭게 제시하였습니다. 그리고 현재 우리가 사용하고 있구요.

Oauth 2.0에서 크게 바뀐 점은 다음과 같습니다.

1. 모바일 어플리케이션에서도 사용이 용이해짐.

2. 반드시 HTTPS를 사용하기에 보안이 강화됨.

3. Access Token 의 만료기간이 생김.

OAuth 2.0의 인증 방식은 크게 4가지 입니다.

1. Authorization Code Grant

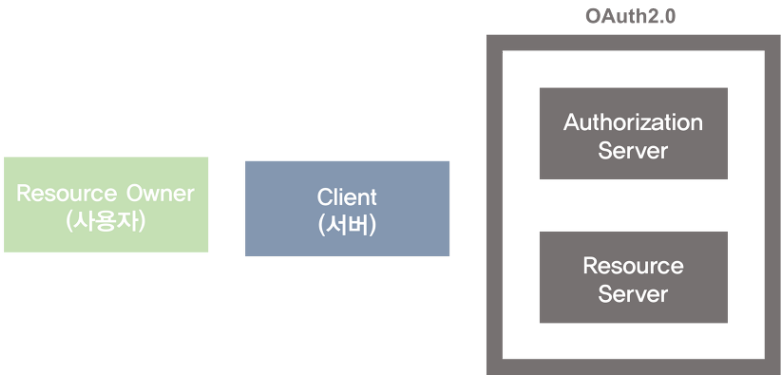
2. Implicit Grant

3. Resource Owner Password Credentials Grant

4. Client Credentials Grant

각 인증 방식에는 장단점이 존재합니다. 저는 가장 많이 쓰이는 Authorization Code Grant 방식을 예로 들어 동
작 순서를 적도록 하겠습니다.

OAuth 2.0 의 동작순서



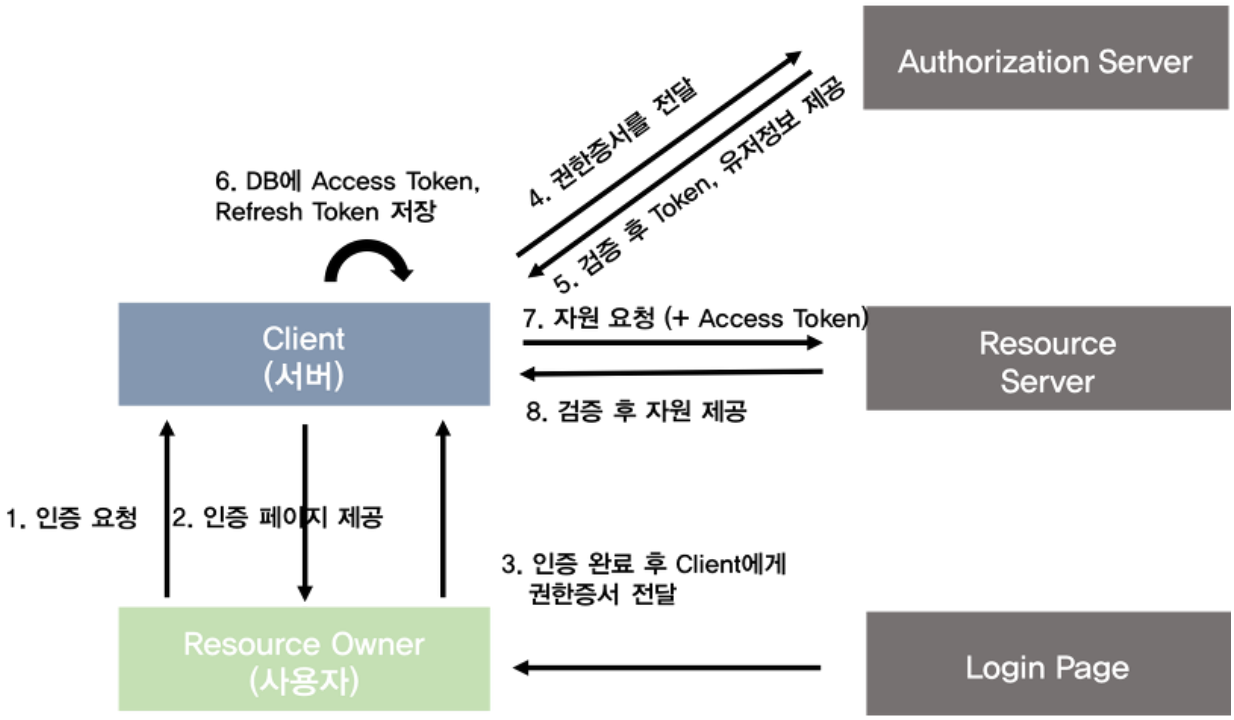
사전 개념을 미리 정리하겠습니다.

Resource Owner : User, 즉 일반 사용자를 칭합니다.

Client : 우리가 관리하는 어플리케이션 서버(User와 혼동될 수 있는데 아닙니다!)

Authorization Server : 권한을 관리하는 서버입니다. Access Token, Refresh Token을 발급, 재발급 해주는 역할을 합니다.

Resource Server : OAuth2.0을 관리하는 서버(Google, Facebook, Naver 등) 의 자원을 관리하는 서버입니다. 주의할 점은 우리가 만드는 서버의 자원을 관리하는 곳이 아닙니다. Oauth 2.0 관리 서버의 자체 API를 의미합니다.



1. Resource Owner(사용자)가 Client(우리 서버)에게 인증 요청을 합니다.

2. Client는 Authorization Request를 통해 Resource Owner에게 인증할 수단(ex Facebook, Google 로그인 url)을 보냅니다.

3. Resource Owner는 해당 Request를 통해 인증을 진행하고 인증을 완료했다는 신호로 Authorization Grant를 url에 실어 Client에게 보냅니다.

4. Client는 해당 권한증서(Authorization Grant)를 Authorization Server에 보냅니다.

5. Authorization Server는 권한증서를 확인 후, 유저가 맞다면 Client에게 Access Token, Refresh Token, 그리고 유저의 프로필 정보(id 포함) 등을 발급해줍니다.

6. Client는 해당 Access Token을 DB에 저장하거나 Resource Owner에게 넘깁니다.

7. Resource Owner(사용자)가 Resource Server에 자원이 필요하면, Client는 Access Token을 담아 Resource Server에 요청합니다.

8. Resource Server는 Access Token이 유효한지 확인 후, Client에게 자원을 보냅니다.

9. 만일 Access Token이 만료됐거나 위조되었다면, Client는 Authorization Server에 Refresh Token을 보내 Access Token을 재발급 받습니다.

10. 그 후 다시 Resource Server에 자원을 요청합니다.

11. 만일 Refresh token도 만료되었을 경우, Resource Owner는 새로운 Authorization Grant를 Client에게 넘겨야합니다. (이는 다시 사용자가 다시 로그인 하라는 말입니다.)

** 여기서 2편의 Access Token + Refresh Token 편을 보신 독자분들이라면 인증 과정이 유사하다는 것을 알 수 있습니다. Access Token, Refresh Token을 이용한 인증 방식은 한 서버에서 모두 관리하는 반면, 여기 OAuth에서는 Authorization Server에서 인증+권한 관리를 하고 Resource Server에서는 자원에 대한 관리만 합니다.

** 9~11의 과정은 2편을 보신 분이라면 쉽게 이해하실 수 있을겁니다. http://tansfil.tistory.com/59

** 다시 한 번 강조하지만 Oauth 2.0 은 우리가 이전에 봤던 (사용자-서버) 구조가 아닌 (사용자 - 서버 - Oauth 서버) 입니다. 우리가 만들 서비스들의 인증을 돕기 위한 서비스가 바로 OAuth입니다. Resource Server는 우리의 서버가 아닌 Oauth를 관리하는 서버의 일부임을 명심하세요.

자 여기까지라면 왜 OAuth를 알아야하는지 궁금하실 겁니다. 나는 SNS 로그인 원리를 알고싶어서 왔는데 왜!!
왜냐하면 SNS 로그인을 제공하는 Google, Facebook, Naver 등은 모두 OAuth2.0 프레임워크를 통해 로그인 API를 제공하기 때문입니다!

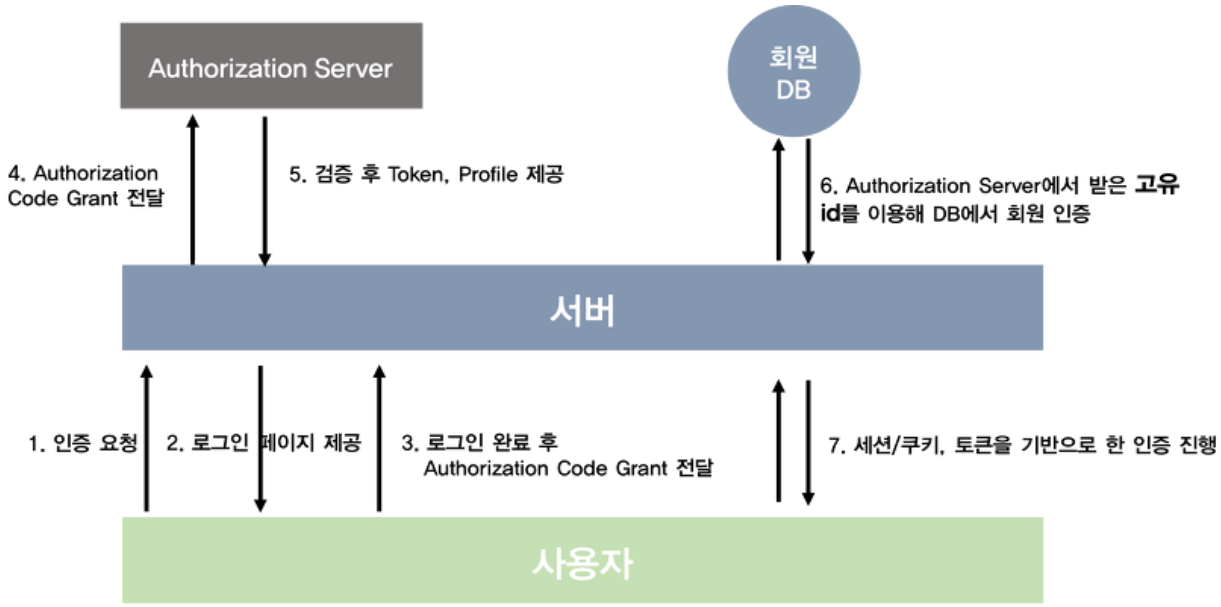
SNS 로그인

<위시캣 FaceBook 로그인 화면>

오래 기다리셨습니다. 이제 SNS 로그인 동작방식에 대해 알아보도록 하겠습니다.

SNS 로그인은 간단하게 봤을 때 OAuth2.0 + 서버 인증(세션/쿠키 , 토큰기반 인증)으로 구성됩니다.

본 설명은 페이스북 로그인을 예로 들겠습니다. 또한 OAuth2.0에 사용되는 명칭은 이해하기 쉽게 바꿔서 설명하도록 하겠습니다.



1. 사용자(Resource Owner)가 서버에게 로그인을 요청합니다.

2. 서버는 사용자에게 특정 쿼리들을 붙인 페이스북 로그인 URL을 사용자에게 보냅니다.

3. 사용자는 해당 URL로 접근하여 로그인을 진행한 후 권한증서(code)를 담아 서버에게 보냅니다.

4. 서버는 해당 권한 증서를 Facebook의 Authorization Server로 요청합니다.

5. 서버는 권한 증서를 확인 후, Access Token, Refresh Token, 유저의 정보(고유 id 포함) 등을 돌려줍니다.

** 여기서 프로필 이미지나 이메일 주소, 이름 등을 얻을 수도 있는데 이는 초기에 관리자가 권한 설정을 어디까지 하느냐에 따라 다릅니다. 페이스북 이름에 대해서만 접근할 수 있는 권한을 설정하면 이름 값만 Authorization Server에서 돌려줄 것입니다.

6. 받은 고유 id를 key값으로 해서 DB에 유저가 있다면 로그인, 없다면 회원가입을 진행합니다.

7. 로그인이 완료되었다면 세션/쿠키 , 토큰기반 인증 방식을 통해 사용자의 인증을 처리합니다.

** 우리가 만들 서버에서 OAuth를 이용하기 위해서는 사전에 OAuth에 등록하는 과정이 필요합니다. 등록 후 APP_ID와 CLIENT_ID 등을 보내야 OAuth 에서는 어느 서비스인지를 알 수 있습니다.

** 페이스북 로그인을 인증을 이용하는 경우, 대부분은 **Resource Server**(페이스북 자체 API)를 사용하지 않습니다. 따라서 Access Token, Refresh Token은 실제로 쓰이지 않습니다. 우리의 서버에서 access token을 검증할 수도 없을 뿐더러 인증의 수단으로 활용하기엔 부족한 점이 많습니다. 따라서 보통 7번 절차처럼 **Authorization Server**로 부터 얻는 고유 id값을 활용해서 DB에 회원관리를 진행합니다.

SNS 로그인의 장점

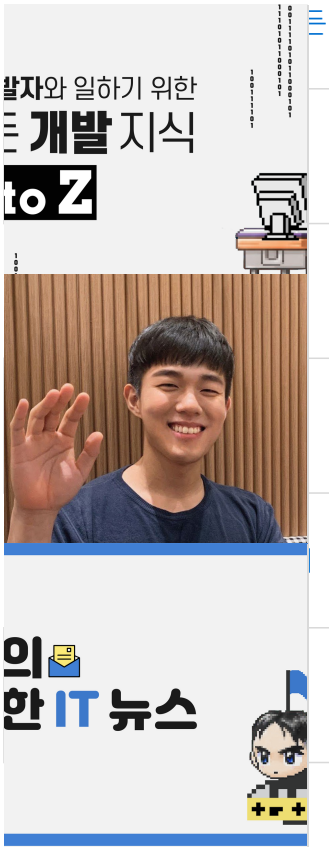
1. 회원가입이라는 귀찮은 절차를 없애고, 사용자가 빠르게 회원가입을 할 수 있다.

2. 접근하고 싶은 정보들은 사용자들이 미리 권한 내용을 확인하고 허락하기에 쉽게 접근할 수 있다.

본 포스팅을 끝으로 세션/쿠키 , 토큰기반 인증, SNS 로그인(feat. OAuth) 에 대해 알아보았습니다. 최대한 상세하게 동작 방식을 풀어내서 포스팅을 작성하였습니다. 이해가 안되시거나 잘못된 정보 혹은 오타들 모두 적극적으로 수용하겠습니다. 많은 댓글 부탁드리겠습니다. 감사합니다.

[참고]

현재 그랩이라는 닉네임으로 크리에이터 활동을 하고 있습니다. 많은 관심 부탁드립니다 :)



든 개발지식 A to Z 자료집

[IT 개발자와 일할 때 필요한 모든 개발지식] A to ...

장담하건대 이 내용들만 알고 계시면 IT 개발의 전체적인 흐름은 전부 파악한다고 보셔도 무방합니다.

www.notion.so

그랩의 IT 열차

IT 트렌드 & 지식을 재밌게 전달해주는 그랩입니다👍

www.youtube.com

그랩의 IT 뉴스레터 ARCHIVE

평일 아침 8시. 하루 3분. 새로운 IT 뉴스와 IT 지식을 전달해드립니다. 개발, IT 지식은 그랩의 IT뉴스로 끝! 딱, 3분만 투자하세요!

www.notion.so

'Programming' 카테고리의 다른 글