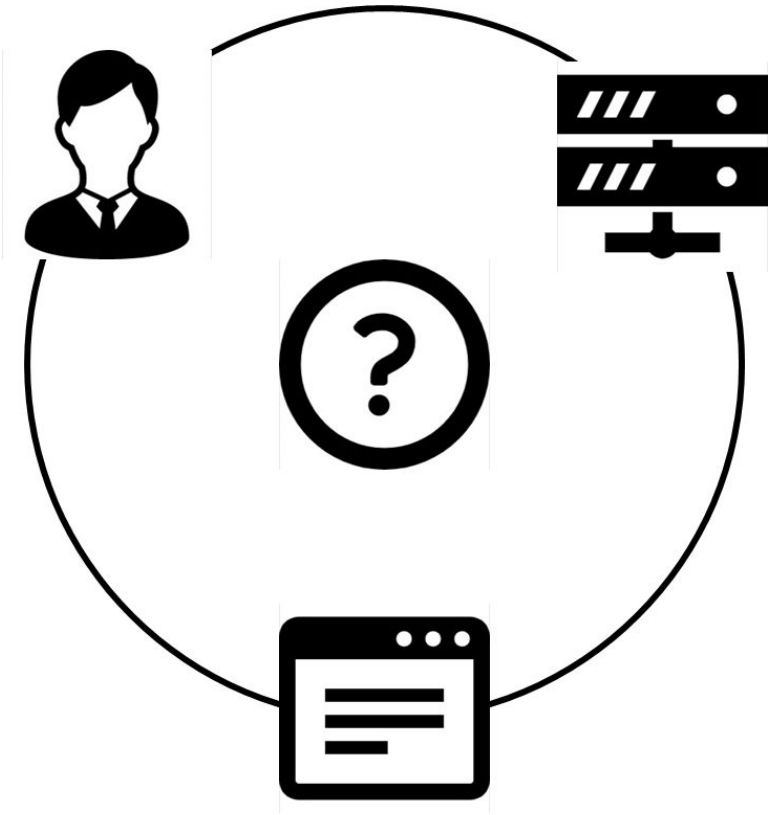
OAuth2.0

※이 문서는 Youtube 생활코딩의 OAuth 강의를 기반으로 작성되었습니다.
보다 자세한 강의가 필요하다면 강의를 들어보는것을 추천드립니다.



위 그림은 OAuth 2.0의 주체들을 표현 및 도식화한 그림입니다.

1) 왼쪽의, 서비스를 이용하고자 하는 사람.
2) 하단의, 외부 Service를 이용하여 사용자에게 서비스를 제공할 어플리케이션
3) 우측의, 외부 Service를 제공하는 외부 Server입니다.

이제부터 1번을 Resource Owner, 2번을 Client, 3번을 Resource Server라고 부르겠습니다.
설명은 진행하면서 하는거로…

쉽게 말하면 OAuth는 를 Resource Owner(이하, Owner)를 대신해서 Resource Server에서 제공하는 서비스를, 사용자의 정보를 토대로 접근할 수 있도록 허가를 받는것을 의미합니다.

가장 쉬운 예로는 어플리케이션에서 Google/Facebook과의 연동을 예로 들 수 있습니다.
이러한 서비스의 경우에는 Client의 서비스는 아니지만, 내가 Resource Server의 서비스를 사용자의 인증을 통하여 제공 할 수 있도록 합니다.

Why? 왜 OAuth를 사용하나요?
Resource Server의 서비스를 제공하는 가장 쉬운 방법은 Owber의 ID/PASSWORD를 받는것입니다.
가장 단순하면서도 모든 정보에 접근할 수 있기 때문에 가장 강력합니다.

하지만, 문제가 발생합니다.
Owner의 입장에서는 자신의 ID/PASSWORD를 처음보는 Client에 맡겨야 되는 상황이며,
Resource Server의 입장에서는 Owner가 아닌 제3자에게 Owner의 정보를 줘야되는 상황,
그리고 Client의 입장에서는 혹시 모를 정보의 유실로부터의 부담 혹은 고초가 예상됩니다.

이러한 이유로 Owner의 ID/PASSWORD를 그대로 Client에 위임하는것은 옳지 못합니다.
그래서, OAuth를 사용하게 되었습니다.

OAuth는 외부 서비스 접근/사용을 위해 사용자의 인증을 대신하도록 해주는 일종의 표준화된 기술이라고 생각하시면 될 것 같습니다.
OAuth를 사용하여 Facebook, Google, Naver등 다양한 Resource Server의 서비스를 사용 할 수 있게 됩니다.

Access Token.
OAuth에서는 Access Token을 통해서 Owner의 인증을 통한 Resource Server의 서비스를 접근하며, 모든 기능이 아닌 필요한 서비스에 국한된 제한된 서비스를 제공할 수 있도록 해줍니다.
제한된 서비스제공을 통해서 정보의 유출시에 제한된 수준의 정보만이 유출되는, 일종의 보안 효과를 이뤄냅니다.

과정.
OAuth를 사용하기 위해서는 우선, Client에서는 일종의 행정적인 작업을 필요로 합니다.
물론, 의미없는 작업은 아닙니다.
Client는 ResourceServer에 우리의 어플리케이션에서 ResourceServer의 어떤 기능을 사용할것인지를 ResourceServer에 등록합니다.
그리고 당연한 말이지만, 나중에 Access Token을 발급받아도 여기서 지정한 기능에 대한 기능만을 사용 할 수 있게 됩니다.

다시 돌아와서, 등록이 끝나고 나면 ResourceServer는 Client에게 ClientID와 Client Secret 두개의 값을 발급합니다.
이는 해당 Client가 ResourceServer로부터 이러한 기능에 대하여 사용을 요청/허가했다는 것을 의미하며, Client는 이 ID값과 Secret값을 DB 혹은 File로 잘 가지고 있어야 합니다.
이때, 무엇보다도 Secret값은 노출되도록 해서는 안됩니다.

인증의 과정은 이렇습니다.
Client에서 자신의 서비스를 포함하여 ResourceServer의 서비스를 이용하여 부가 가치를 창출하여 Owner에게 제공하는 서비스를 제작하였습니다.
그리고 Owner는 우리의 서비스를 사용하기 시작합니다.
Client는 Owner에게 기능 사용을 위해서는 Resource Server의 이러이러한 Service를 사용하며, 그렇기 때문에 이런 정보를 사용하겠다는 것을 허락/인증 받아주세요. 라는 부탁을 하게됩니다.

Client의 부탁을 Owner가 승낙/승인하게 됩니다.
그렇게 되면 ResourceServer의 Login항목이 나타나게되며, 이곳에서는 Client가 당신의 ResouceServer의 정보를 사용하려고 하는데 동의 하느냐라는 문구를 볼 수 있습니다.

승낙시 Owner는 미리 지정된 Redirection페이지로 이동하게 되며, Client는 정보 사용을 위한 Code를 ResoureServer로부터 발급받게됩니다.

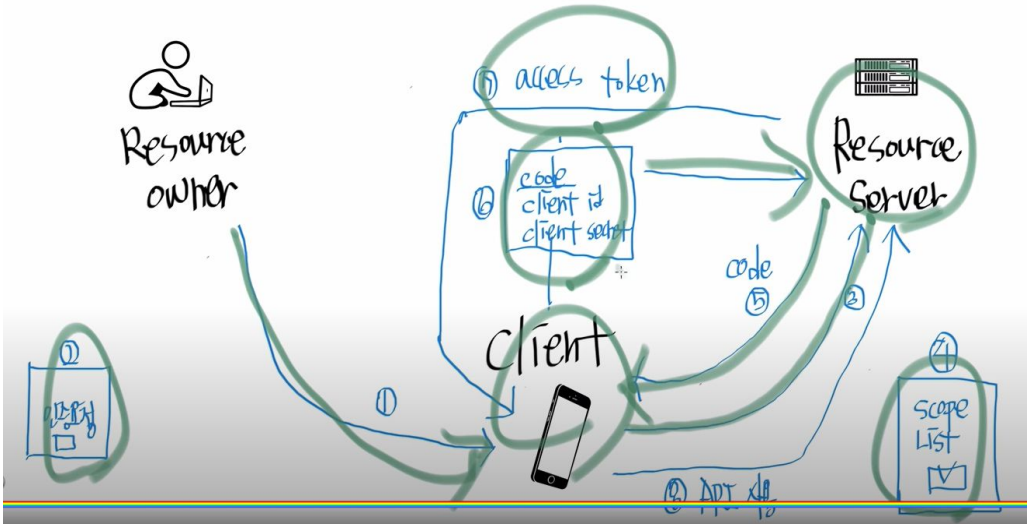
끝난것 같지만 아직 끝나지 않았습니다.
Code는 사용자의 정보에 대한 승낙을,
이전의 Client Id와 Client Secret은 ResourceServer의 서비스에 대하여 Client의 사용을 의미합니다.
즉, 사용자의 정보를 가지고 ResourceServer의 서비스를 이용하여 정보를 얻어내기 위해서는 아직 한가지 과정을 더 거쳐야 합니다.

Client는 Code값과 Client Id, Client Secret 3가지 값들을 ResourceServer에 보냅니다.
ResourceServer는 이 값들을 비교하여 현재 요청한 client가 자신들이 인증한 client가 맞는지를 판단하며 그제서야 중요한 정보를 주기 시작합니다.
이 과정의 결과로 위에서 한번 언급되었던 Access Token을 획득 할 수 있습니다.

진짜 비밀번호라고 생각하면 됩니다.
Access Token을 통해서 실제 ResourceServer에 자원을 요청하게 됩니다.
ResourceServer는 자신이 발급했던 토큰임을 알아채곤 데이터를 제공하게 됩니다.
Client는 받은 데이터를 가공하여 Owner에게 서비스를 제공하게 됩니다.
이때,AccessToken은 잃어버리지 않도록 주의해야 하며, 일정 시간동안만 존재하기 때문에 주의해야 합니다.
→ 이는 이후 Refresh Token을 통하여 갱신시킬 수 있게됩니다.

정리하면 아래와 같습니다.

1) Owner의 Client접속
2) ResourceServer에 인증요청.
3) ResourceServer 로그인.
4) ScopesList전시. Client에서 이러한 기능을 요청중에있다. 승낙할것인지?/
5) 승낙.
6) ResourceServer에서 Client에게 Code를 반환.
   이러한 정보를 볼 수 있도록 하겠다의 의미를 지님.
   Client가 이를 직접 사용하는것은 아님. AccessToken을 얻는데 사용함.
7) AuthorizeCode와 ClientID, Client Secret 3가지 정보를 묶어 ResourceServer전송.
8) 이 Code에 해당하는 행위인지, 그리고 인증된 Client가 맞는지 Id/Secret 확인
9) AccessToken을 제공.
10) AccessToken을 이용하여 ResourceServer의 API사용.



https://console.developers.google.com/apis

우리는 Client를 제작하게되며, 일단 목표는 Youtube Data API사용을 목표로 하고있습니다.
음.. 직접 실행을 해보려고 하는데 이 부분은 기술스택의 Youtube API란에서 진행하도록 하겠습니다.