# Oauth

# Oauth

- Oauth
  - 비밀번호 없이 사이트의 정보를 제 3자에게 제공해주는 인증하는 표준 프로토콜
- History
  - OpenID : 아이디, 패스워드 없이 인증하는 방법 (피싱 공격에 취약)
  - 2007년 10월 트위터에 OpenID 작업을 위해 Oauth 1.0 표준 초안 작성
  - 2008년 11월 IEFT(Internet Engineering Task Force)에서 Oauth 워킹 그룹 생성
  - 2012년 10월 Oauth 2.0 표준 초안 작성

# Oauth 1.0

- **User**
  - 서비스를 사용하는 사용자
  - Service Provider에 계정이 있으면서 Consumer 서비스를 이용하여는 사람

- **Service Provider**
  - User의 정보를 가지고 있는 서비스
  - Oauth 인증을 이용하여 Open API로 Consumer에게 User의 정보를 제공

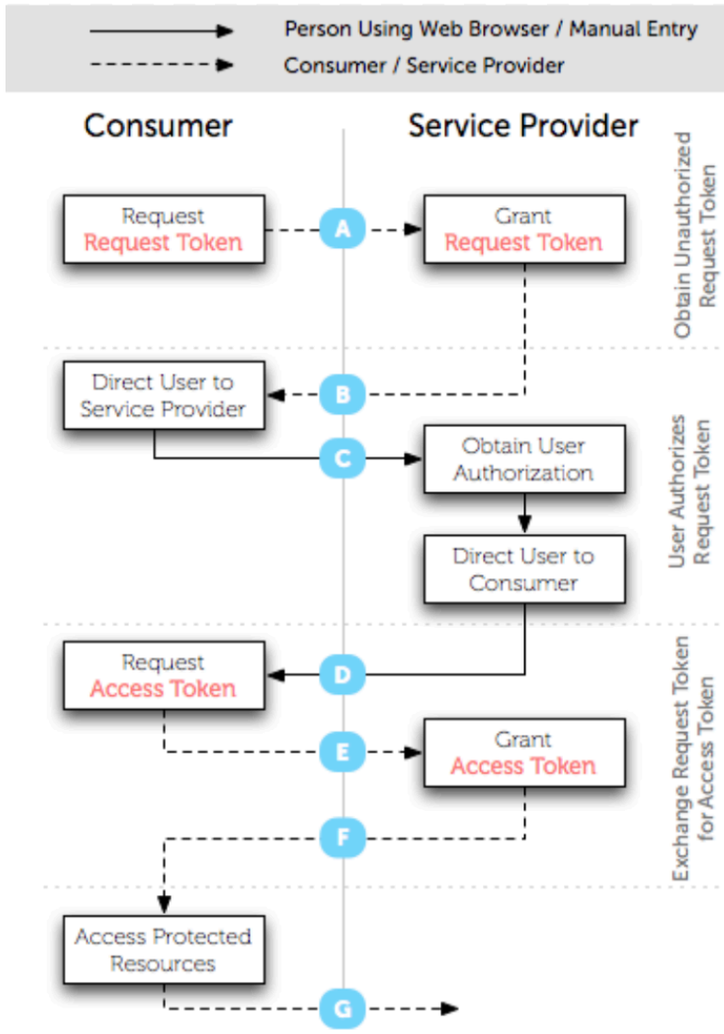
- **Consumer**
  - Oauth 인증을 이용하여 User의 정보를 Service Provider로 부터 제공 받는 서비스

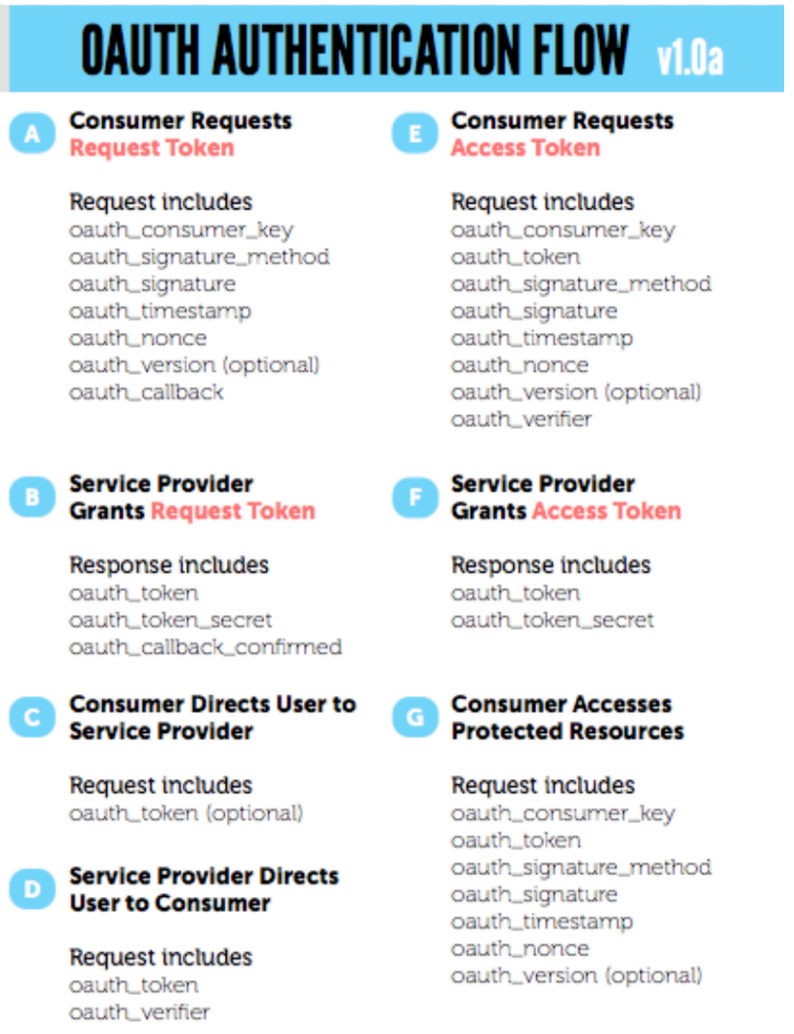
- **Request Token**
  - Consumer가 Service Provider에게 접근 권한을 인증 받기 위한 토큰
  - Request Token으로 Access Token을 전달 받음

- **Access Token**
  - Consumer가 Service Provider에게 정보를 요청할때 사용하는 토큰

# Oauth 1.0

- consumer 가 service provider 에게 request token 을 요청
- service provider 가 consumer 에게 request token 발급
- consumer 가 user 를 service provider 로 이동시켜 인증 수행
- service provider 가 user 를 consumer 로 이동 consumer 가 access token 을 요청
- service provider 가 consumer 에게 access token 발급
- cunsumer가 발급받은 access token 을 이용하여 service provider에 있는 user의 정보에 접근



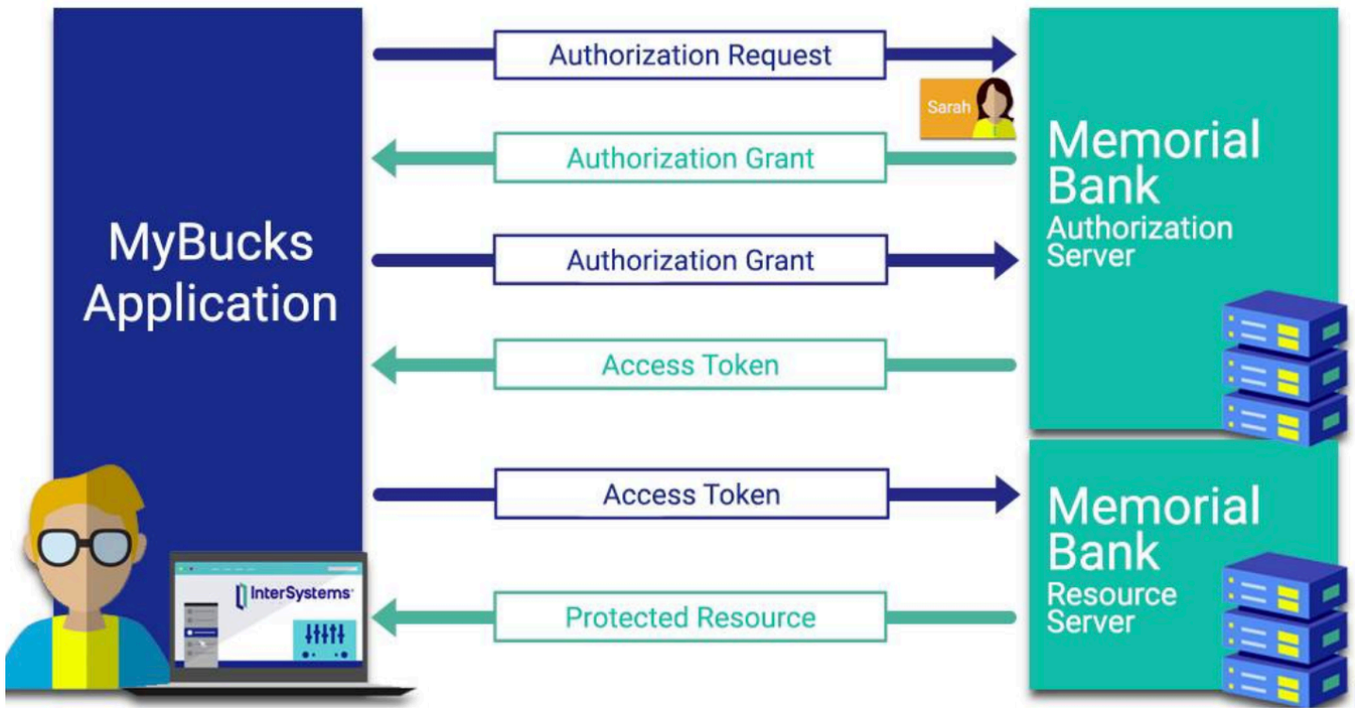



# Oauth 2.0

- User
- Service Provider
  - Authorization Server
  - Resource Server
- Client
- Request Token
- Access Token

# Oauth 2.0

- Client가 Authorization Server로 Request Token 요청
- Authorization Server가 Client에게 Request Token 발급
- Client가 Authorization Server에게 Request Token으로 Access Token 발급 요청 (Resource Owner 의 허락 필요)
- Authorization Server가 Client에게 Access Token 발급
- Client에게 Access Token을 이용하여 Resource Server에게 Resource 요청
- Resource Server가 Client에게 Resource 제공



# Oauth 1.0과 2.0의 차이

- Oauth 1.0
    - RFC 5849
- Oauth 2.0
    - RFC 6749
    - Oauth 1.0과 호환성을 제공하지 않음
    - Resource Server와 Authorization Server의 분리
    - 인증절차의 간소화 (디지털 서명 -> https 이용)
    - 다양한 인증방식 추가 ( Refresh Token)