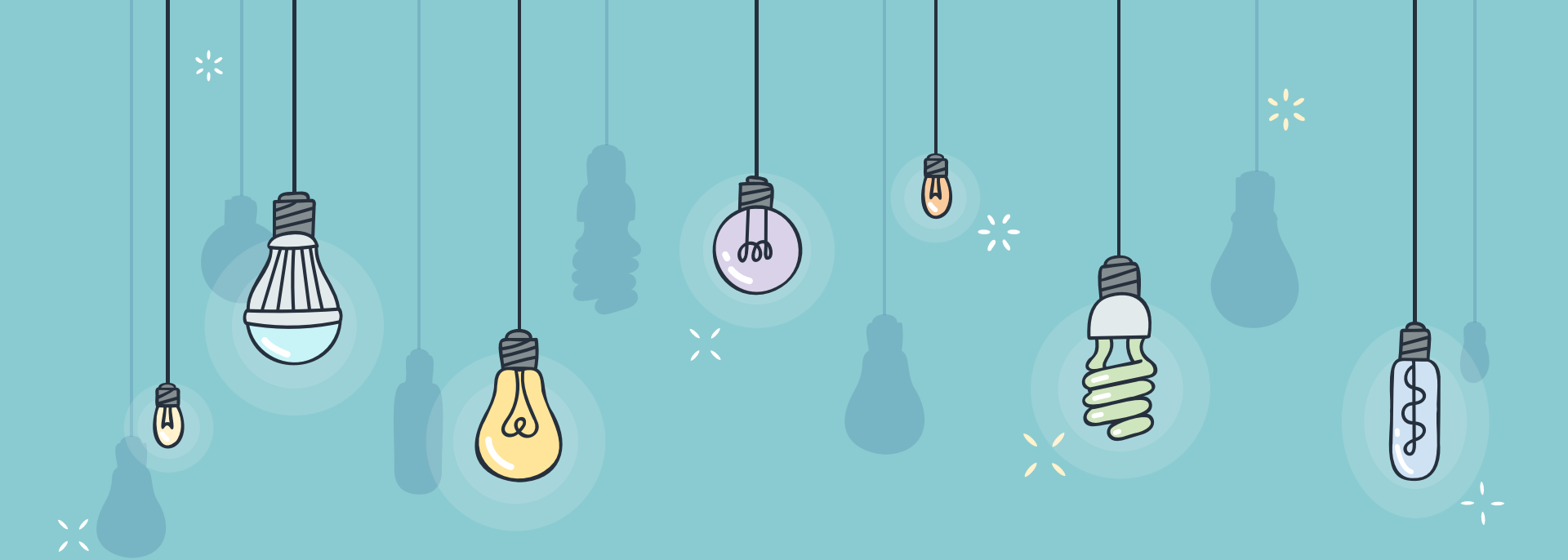
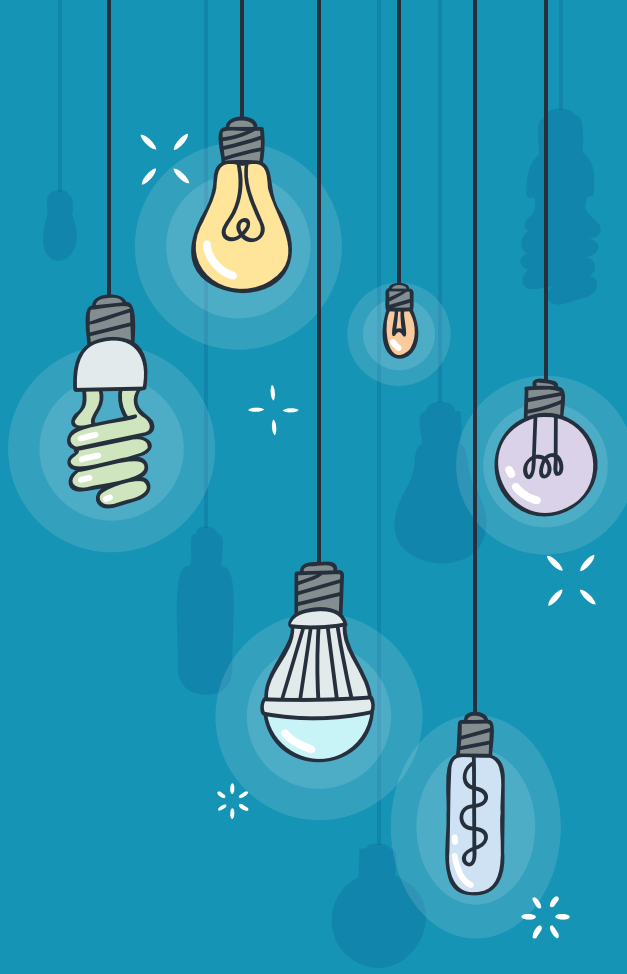
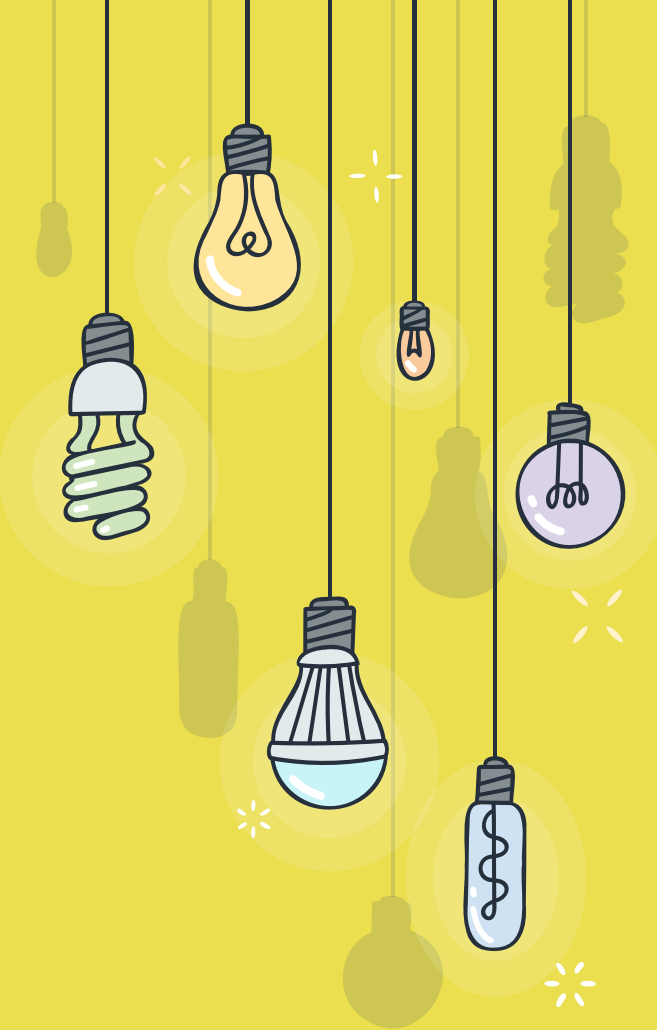


Spring boot

1. JWT의 개요



1 JWT의 개요



인증

- 세션기반 인증
  - 사용자의 인증 정보가 서버의 세션 저장소에 저장
- 토큰기반 인증
  - 클라이언트가 ID/pass를 제공하고 서버에 로그인하면 서버가 토큰을 만들어 클라이언트에게 제공
    - 인증 정보는 토큰의 형태로 브라우저의 로컬 스토리지(혹은 쿠키)에 저장
  - 서버에 접근시 토큰을 제시하고 접근
    - 사례: 페이스북, 트위터, 구글, github 등 한번 로그인하면 일정기간동안 패스워드를 묻지 않고 로그인된 상태로 사용 가능

인증

세션 기반 vs 토큰 기반 인증

	세션 기반 인증	토큰 기반 인증
트래픽	ID만 실어 보내면 되므로 트래픽을 적게 사용	많은 네트워크 트래픽을 사용(인증 정보, 발급시각, 만료시각, 토큰의 ID등)
보안	- 서버에 저장되므로 안전 - Payload가 암호화 되어있지 않음 (누구나 내용 확인가능)	- 해커에게 탈취되면 해당 토큰이 만료되기 전까지 보안이 뚫림 - 저장할 수 있는 데이터 제한 없음
확장성	여러대의 서버가 요청을 처리할 때 세션 불일치 문제	세션 불일치 문제로부터 자유롭다
서버의 부담	다량의 사용자 있는 경우 서버 성능에 부하 발생	서버의 부담이 증가하지 않음

JWT개요

- JWP(Json Web Token) 개요
  - 디지털 서명(Digitally Signed)을 통해 서로 다른 시스템간의 정보를 보안 상 문제가 없도록 JSON 객체 형태로 주고받는 오픈 스탠다드
  - Http Request 헤더에 포함되어 있는 토큰을 통해 인증
  - Claim Base Token
    - Claim : 토큰 주체에 대한 사실 정보를 담고 있는 키/값 쌍.
      ex) 사용자 정보, 권한 등

JWT의 구조

JWT의 구조

헤더(Header) 내용(Payload) 서명(Signature)

x x x x x . y y y y y . z z z z z

```
{
  "alg" : "HS256"
  "typ" : "JWT"
}
```

```
{
  "iss" : "토큰 발급자"
  "sub" : "토큰 제목"
  "iat" : "발급 시간"
  "exp" : "만료 시간"
  "jti" : "JWT의 고유식별자"
  "mydata" : {
    "id" : "사용자id"
  }
}
```

```
HMACSHA256(
  base64URL(header)
  +          •
  + base64URL(payload)
    , my_secret_key_키값
)
```

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJqd3RTZWNyZXQiLCJuYW1lIjoiaml3b24ga2ltIiwiaWF0IjoxNTE2MjM5MDIyfQ.6Xeyam3KL3dpcJAKsN8vKB_LgeFwg9lE4ySmRM3Gw
```

JWT의 장단점

- 장점
  - 서버는 세션정보를 사용하지 않아 사용자 관리가 편리
  - 타 사이트에도 인증 제공 가능
    - 사례: 페이스북/카카오톡으로 로그인하기
  - 모든 종류의 디바이스, 어플리케이션에 토큰이 있으면 접근 허용
- 단점
  - 서버에서 강제로 Revoke 하기 어려움

Thanks!

- Any questions?