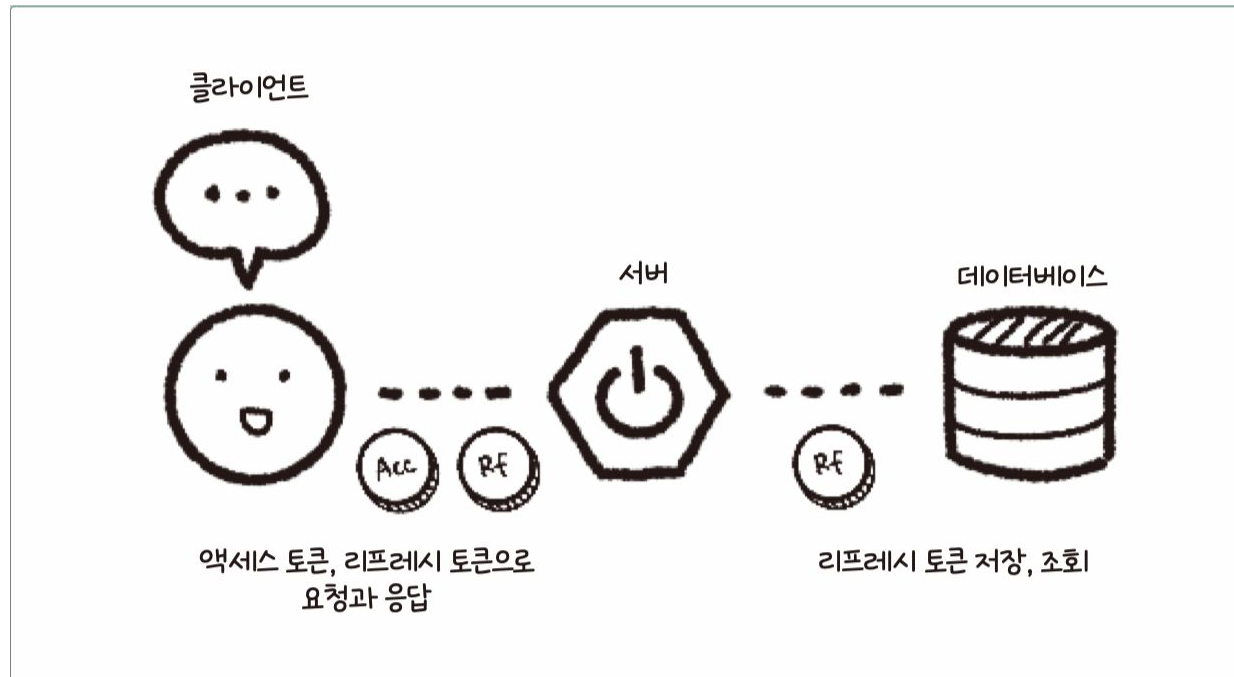


# 스프링 부트 3 백엔드 개발자 되기 (2판)

JPA+OAuth2+JWT+AWS와 배우는 스프링 부트 3  
Java 백엔드 입문자를 위한 풀 패키지

## 09 JWT로 로그인/로그아웃 구현하기

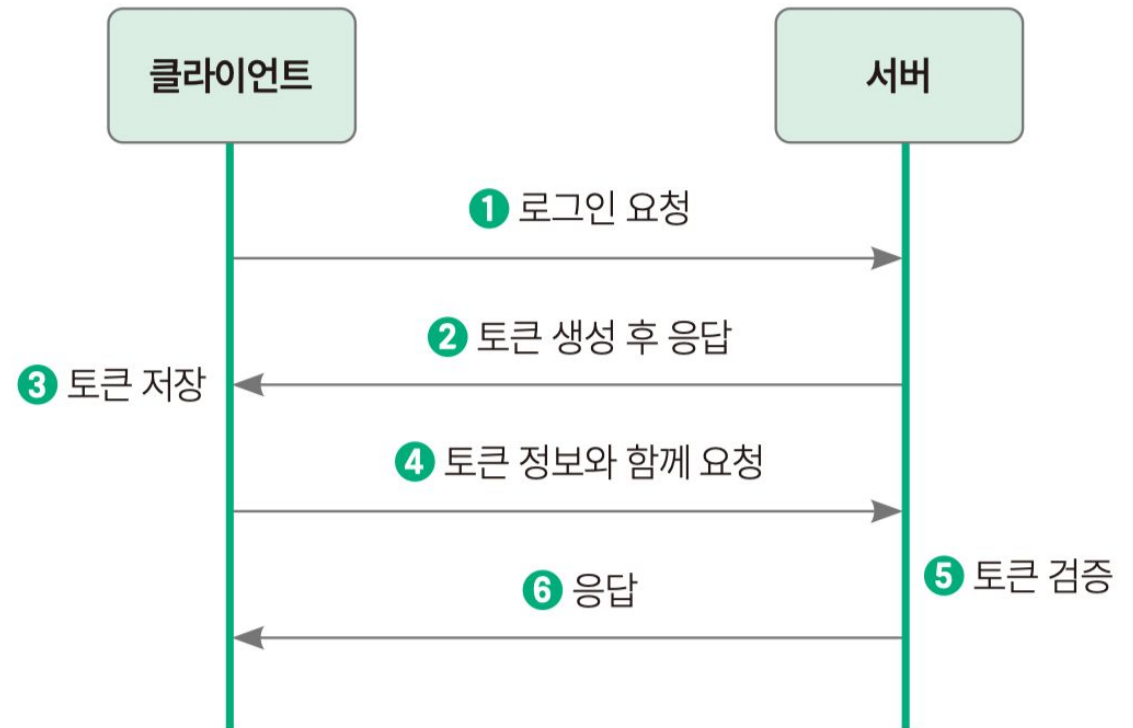
# 프로젝트 구성



# 9.1 사전 지식 : 토큰 기반 인증

## • 토큰

- 서버에서 클라이언트를 구분하기 위한 유일한 값
- 서버가 토큰 생성 →
- 클라이언트에게 제공 →
- 클라이언트는 토큰을 소지 →
- 클라이언트는 토큰과 함께 서버에 요청
- 서버는 토큰으로 유효한 사용자인지 검증



# 9.1 사전 지식 : 토큰 기반 인증(cont.)

- 토큰 기반 인증의 특징
  - 무상태성
    - 토큰을 클라이언트가 소지 → 서버의 자원 절약
    - 서버가 클라이언트의 인증 정보를 저장하거나 유지할 필요가 없음
  - 확장성
    - 하나의 토큰으로 다양한 서버에 요청을 보낼 수 있음
    - 페이스북 로그인이나 구글 로그인으로 다른 서비스에 로그인하는 것이 그 예
  - 무결성
    - 토큰 발급 이후 변경하면 서버에서 감지

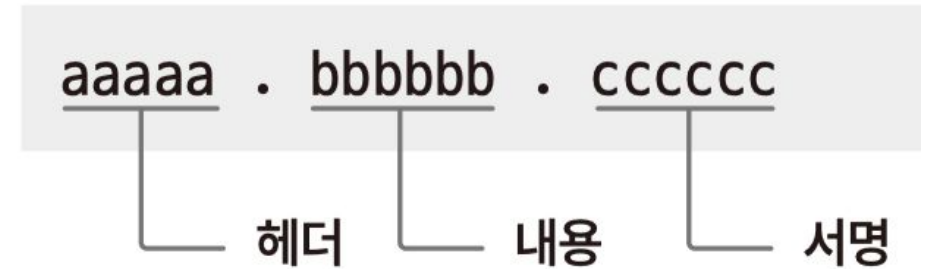
# 9.1 사전 지식 : 토큰 기반 인증(cont.)

- JWT

- HTTP 요청 헤더 중 Authorization 키값에 Bearer + JWT 토큰값을 보냄

- JWT의 구조

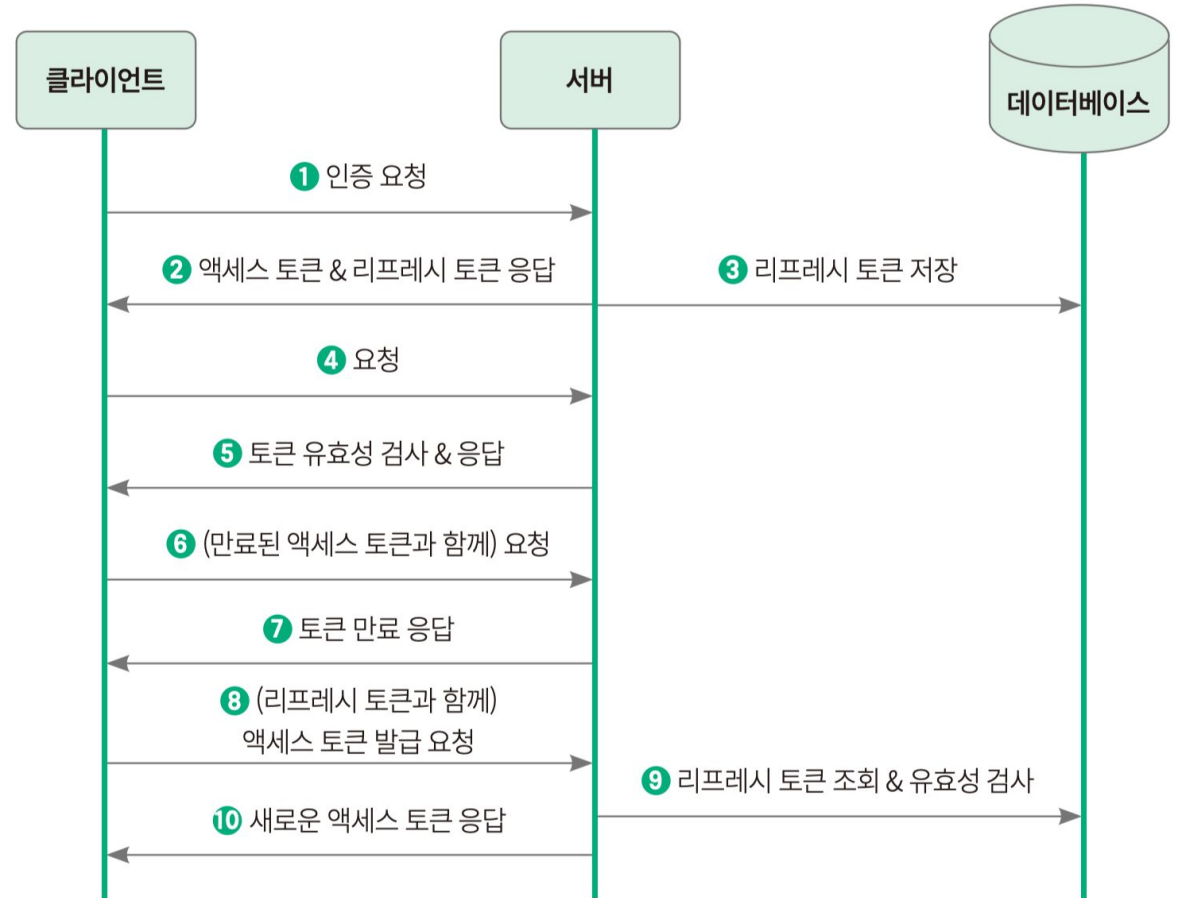
- 헤더 : 토큰 타입 + 해싱 알고리즘 정보
  - 내용 : 토큰과 관련된 정보
    - 클레임 덩어리로 구성됨
    - 등록된 클레임
    - 공개 클레임
    - 비공개 클레임
  - 서명 : 토큰이 조작되었거나 변경되었는지 확인하기 위한 정보
    - 헤더의 인코딩값 + 내용의 인코딩값 → 비밀키로 해시값 생성



# 9.1 사전 지식 : 토큰 기반 인증(cont.)

- 토큰 유효기간

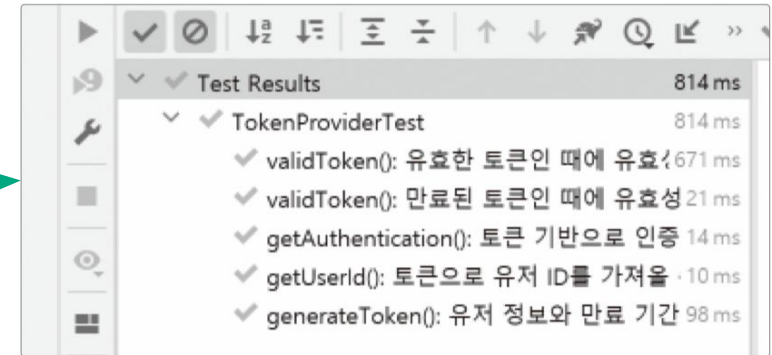
- 토큰 자체가 노출되었을 시 유효기간으로 보호



## 9.2 JWT 서비스 구현하기

- 실습 과정(233쪽부터 참고)

- build.gradle
- application.yml
- /config/jwt/JwtProperties.java
- TokenProvider.java
- JwtFactory.java
- TokenProviderTest.java
- RefreshToken.java
- RefreshTokenRepository.java
- TokenAuthenticationFilter.java



getAuthentication( ) 메서드로 유저 객체 반환

```
context.getAuthentication()
result = {UsernamePasswordAuthenticationToken@12318} "UsernamePasswordAuthenticationToken [Principal=org.springframework.security.core.userdetails.User@12335, Password=[PROTECTED], Enabled=true]"
principal = {User@12335} "org.springframework.security.core.userdetails.User [Username=ajufresh@gmail.com, Password=[PROTECTED], Enabled=true]"
password = ""
username = "ajufresh@gmail.com"
authorities = {Collections$UnmodifiableSet@12350} size = 1
0 = {SimpleGrantedAuthority@12339} "ROLE_USER"
```



## 9.3 토큰 API 구현하기

- 실습 과정(246쪽부터 참고)
  - UserService.java
  - RefreshTokenService.java
  - TokenService.java
  - CreateAccessTokenRequests.java
  - CreateAccessTokenResponse.java
  - TokenApiController.java
  - TokenApiControllerTest.java

