김병훈

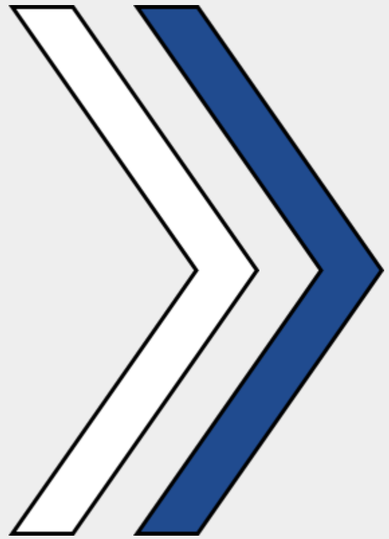
“ 토큰과 세션 ”

INDEX

목차

01

JWT 인증 방식의 장단점



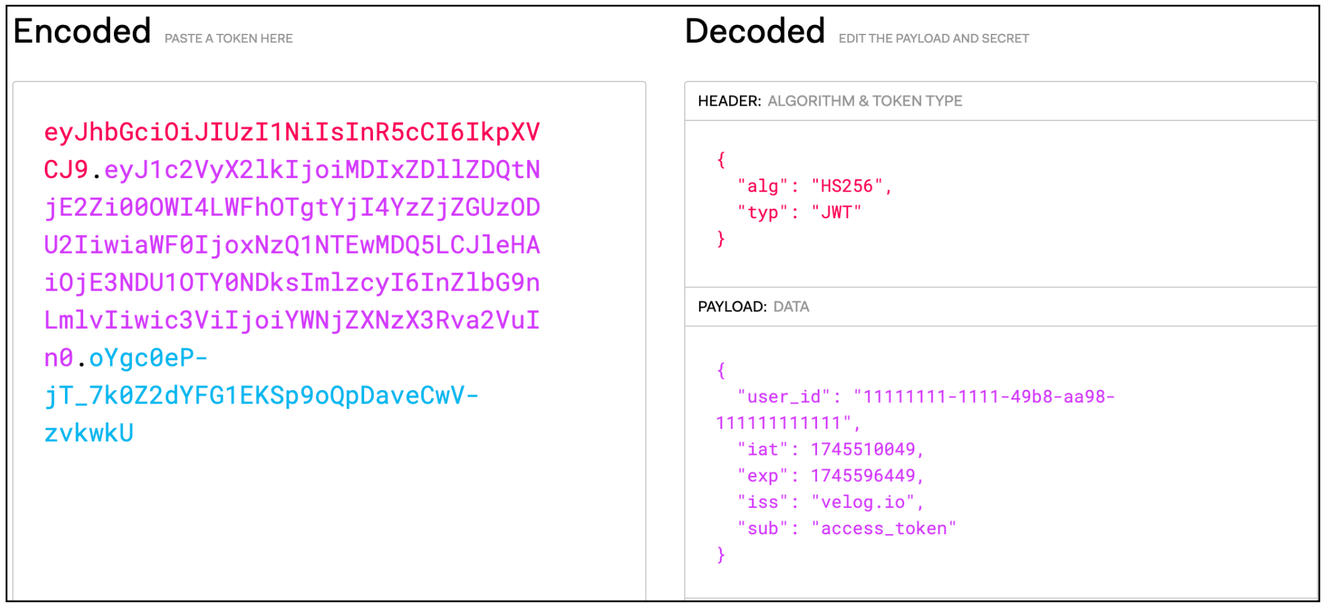
01 JWT (JSON Web Token)의 장단점

이름	값	Do...	Path	Exp...	크기	Htt...	Sec...	Sa...	Part...	Cro...	Prio...
access_token	eyJhbGciOiJIUzI1NiIsInR5cCI6Ik...	.vel...	/	202...	256	✓					Me...
refresh_token	eyJhbGciOiJIUzI1NiIsInR5cCI6Ik...	.vel...	/	202...	325	✓					Me...

JWT의 장점

- 1. 무상태(Stateless)이다 → 서버가 클라이언트의 상태를 보존하지 않음
- 2. 분산 시스템에 적합하다 → 클라이언트가 갖고 있기 때문에 MSA 아키텍쳐 경우 유리
- 3. 클라이언트 중심 인증 → 클라이언트가 직접 토큰을 보관하고 전송 → RESTful API와 잘 어울림
- 4. 토큰에 정보 포함 가능 → 사용자 권한 등 포함 가능 → 추가 조회 없이 처리 가능

01 JWT (JSON Web Token)의 장단점

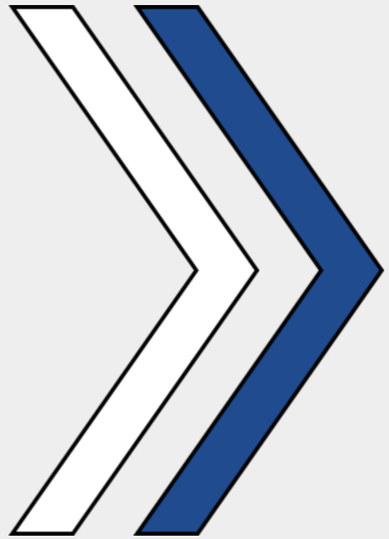


“ JWT의 단점 ”

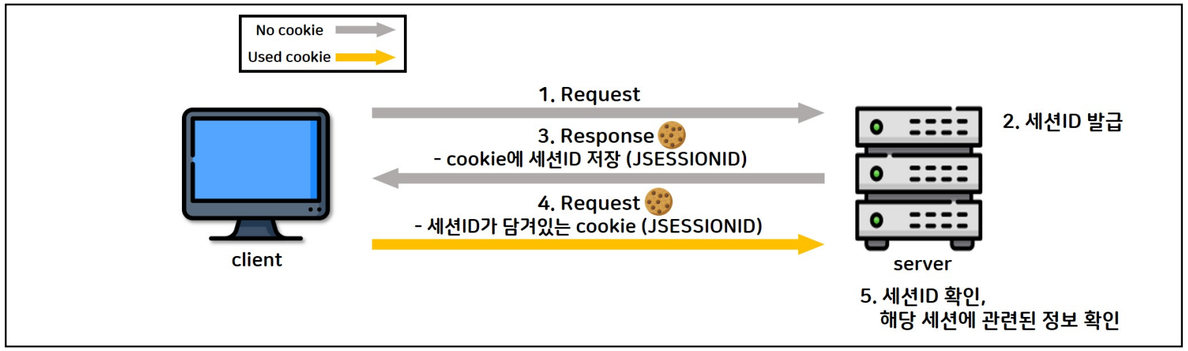
1. Payload 노출 가능성이 있다.
   → 서명되어 있지만 디코딩은 가능 → 민감 정보 저장 금지
2. 토큰 탈취 시 위험 큼
   → 탈취되면 만료 전까지 막을 수 없음
3. 실시간 사용자 상태 반영 어려움

02

세션 인증 방식의 장단점



02 세션(Session)의 장단점



세션의 장점

1. 서버에서 상태 관리 가능
   → 사용자 상태 변경을 즉시 반영 가능 (예: 강제 로그아웃, 권한 변경 등)
2. 보안에 유리
   → 민감 정보는 서버에만 저장, 클라이언트에는 세션 ID만 전달
3. 로그아웃 처리 용이
   → 서버에서 세션 무효화로 처리 가능
4. CSRF 방어가 용이
   → 서버에서 Referer, Origin 검사로 비교적 쉽게 방어 가능

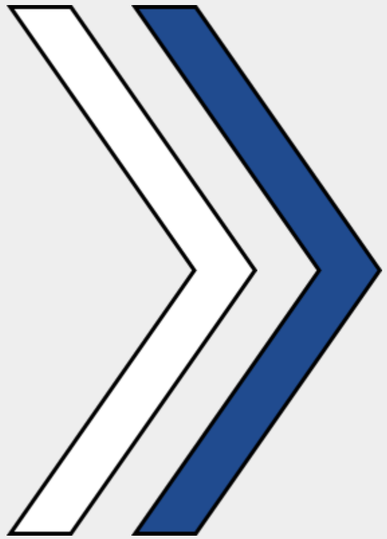
02 세션(Session)의 장단점

“ 세션의 단점 ”

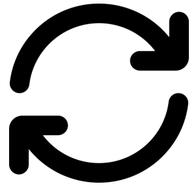
1. 서버에 저장 공간 필요하다. (상태 유지) -> 상태성이기 때문
    1. 사용자 수가 많을수록 메모리 부담 증가
2. 수평 확장 어려움
    1. 로드 밸런서 환경에서는 세션 공유를 위한 Redis 등 별도 저장소 필요 (Redis Session Clustering 등)
3. 스케일 아웃 구조에서는 부적합할 수 있음
    1. 중앙 저장소가 병목이 될 수 있음

03

JWT에서 세션 방식으로 리팩토링한 이유

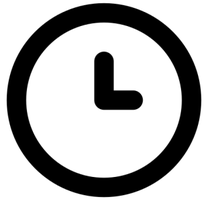


03 JWT에서 세션 방식으로 리팩토링한 이유



Refresh Token 관리 복잡성

JWT 기반 인증에서는 Refresh Token의 저장 및 갱신 로직이 복잡하며, 보안 문제로 인해 별도 저장소가 필요



실시간 사용자 상태 반영 어려움

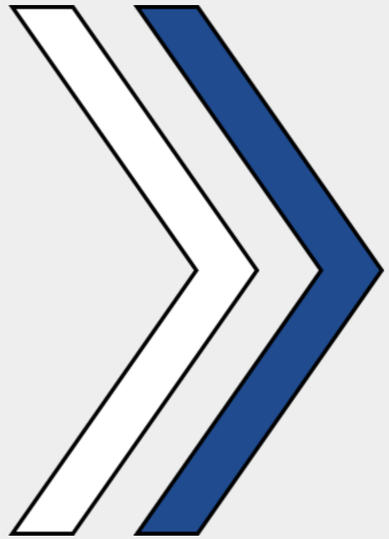
JWT는 Stateless 구조로 인해 토큰 내 정보 갱신이 어려워, 권한 변경 등 사용자 상태 반영이 지연됨

보안성 및 유지보수 개선 필요

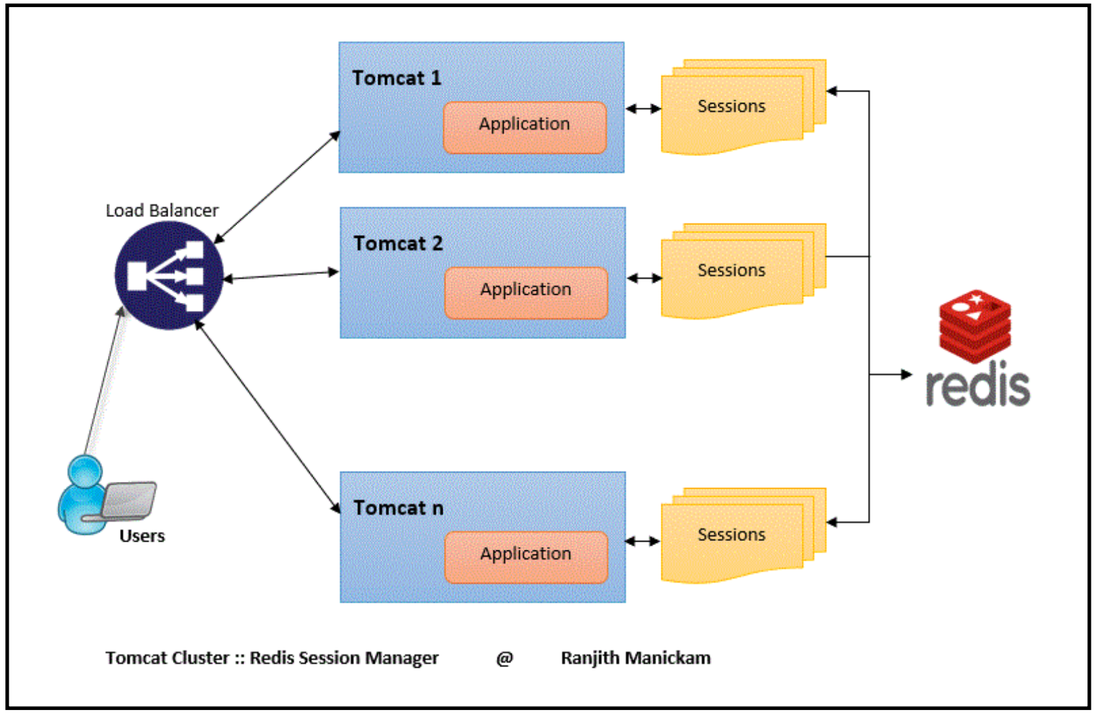
토큰 탈취 시 서버에서 토큰 폐기가 불가능해 보안에 취약하며, 인증 로직 유지보수가 어려움

04

스프링 시큐리티 세션 기반 인증 구조(리팩토링 이후)



04 스프링 시큐리티 세션 기반 인증 구조



“ 리팩토링 후 구조 ”

1. 로그인 요청
   - 사용자가 ID/PW를 입력하고 로그인 요청
2. 인증(Authentication) 처리
   - 자격 증명 확인, 인증 성공 시 Authentication 객체 생성
3. 세션(Session) 생성
   - 인증 성공 시 SecurityContext에 인증 정보를 저장
   - 해당 SecurityContext를 HttpSession 바인딩 -> 서버 저장
4. 이후 요청 처리
   - 클라이언트는 JSESSIONID라는 세션 ID 쿠키를 함께 전송
   - 서버는 세션 ID를 통해 인증 정보 확인 이후 보안 처리를 수행
5. 로그아웃
   - 세션을 무효화(invalidate())하면 서버에서 인증 정보 삭제 → 즉시 로그아웃 가능

05

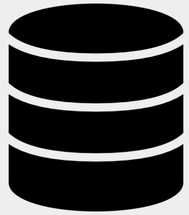
결론

05 JWT vs 세션 기반 인증 요약

JWT 인증

Stateless 구조로 확장성 높음

토큰 갱신 및 보안 측면에서 복잡성 존재



세션 인증

상태 유지로 사용자 관리 용이

Redis 등과 결합 시 확장성 보완 가능

선택 기준

시스템 요구사항, 확장성, 보안 정책 등을 고려한 방식 선택 필요