

2025년 상반기 K-디지털 트레이닝

# member 테이블 (심화 1)

---

[KB] IT's Your Life

## member 테이블

 다음과 같이 사용자 정보 테이블을 생성하세요.

```
-- 사용자 정보 테이블
drop table if exists tbl_member;
```

```
create table tbl_member
```

```
(
    username          varchar(50) primary key,          -- 사용자 id
    password          varchar(128) not null,             -- 암호화된 비밀번호
    email             varchar(50) not null,
    reg_date          datetime default now(),
    update_date        datetime default now()
);
```

```
-- 사용자 권한 테이블
drop table if exists tbl_member_auth;
```

```
create table tbl_member_auth
```

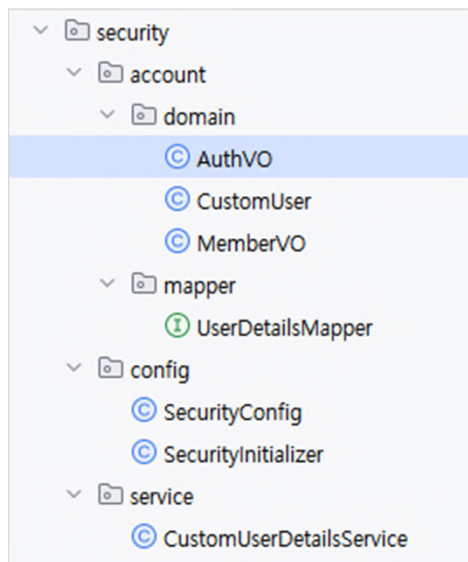
```
(
    username          varchar(50) not null,              -- 사용자 id
    auth              varchar(50) not null,              -- 권한 문자열 ROLE_ADMIN, ROLE_MANAGER, ROLE_ME
    primary key(username, auth),                          -- 복합키
    constraint fk_authorities_users foreign key (username) references tbl_member(username)
);
```

MEMBER 등

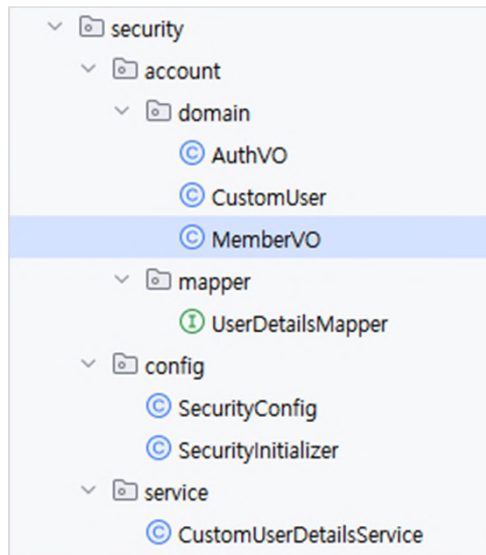
- ✓ 앞에서 만든 사용자 정보 테이블과 권한 테이블에 데이터를 추가하세요.

### ✓ 권한 테이블에 대응하는 AuthVO 클래스를 정의하세요.

#### ○ GrantedAuthority 인터페이스 구현



- ✓ 사용자 정보 테이블에 대응하는 MemberVO를 정의하세요.



## member 테이블

✓ UserDetailsMapper 인터페이스와 UserDetailsMapper.xml을 작성하세요.

○ 메서드

- MemberVO get(String username)

## member 테이블

- ✓ SecurityConfig에 UserDetailsMapper가 검색되도록 설정을 추가하고, 단위 테스트를 통해 Mapper의 동작을 확인하세요.

2025년 상반기 K-디지털 트레이닝

# UserDetails 사용하기 (심화 2)

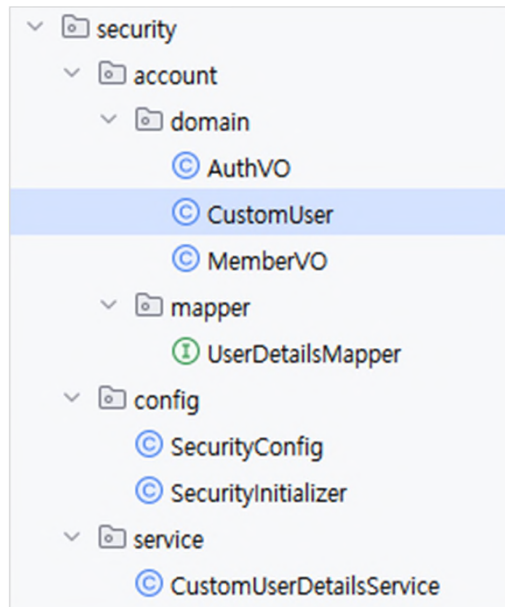
---

[KB] IT's Your Life



## UserDetails 사용하기

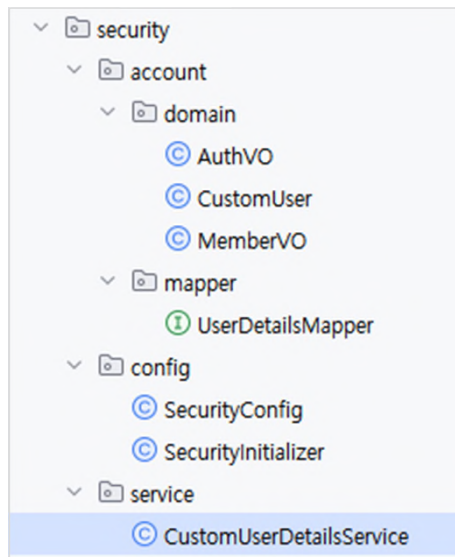
- ✓ CustomUser 클래스를 정의하세요.
  - User 클래스 상속



## UserDetails 사용하기

### ✓ CustomUserDetailsService 클래스를 정의하세요.

- UserDetailsService 인터페이스 구현



## UserDetails 사용하기

- ✓ AuthenticationManager에서 인메모리 사용자 정보 대신 UserDetailsService를 이용하도록 수정하세요.

## UserDetails 사용하기

- ✓ 컨트롤러에 UserDetails을 얻는 방법 사용하여 확인하세요.
  - Principal 주입
  - Authentication 주입
  - @AuthenticationPrincipal 주입

## UserDetails 사용하기

✓ index.jsp에서 로그인 여부에 따라 화면을 재구성하세요.

- 로그인 한 경우
  - username 출력
  - 로그아웃 링크 출력
  
- 로그인 안 한 경우
  - 로그인 링크 출력