

2024년 상반기 K-디지털 트레이닝

# member 테이블

---

[KB] IT's Your Life

## 테이블 생성

-- 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_MEMBER 등
  primary key(username, auth),      -- 복합키
  constraint fk_authorities_users foreign key (username) references tbl_member(username)
);
```

## 데이터 추가

-- 테스트 사용자 추가

```
insert into tbl_member(username, password, email)
```

```
values
```

```
  ('admin', '$2a$10$EsIMfxbJ6NuvwX7MDj4Wq0YFzLU9U/lddCyn0nic5dFo3VfJYrXYC', 'admin@galapgos.org'),  
  ('user0', '$2a$10$EsIMfxbJ6NuvwX7MDj4Wq0YFzLU9U/lddCyn0nic5dFo3VfJYrXYC', 'user0@galapgos.org'),  
  ('user1', '$2a$10$EsIMfxbJ6NuvwX7MDj4Wq0YFzLU9U/lddCyn0nic5dFo3VfJYrXYC', 'user1@galapgos.org'),  
  ('user2', '$2a$10$EsIMfxbJ6NuvwX7MDj4Wq0YFzLU9U/lddCyn0nic5dFo3VfJYrXYC', 'user2@galapgos.org'),  
  ('user3', '$2a$10$EsIMfxbJ6NuvwX7MDj4Wq0YFzLU9U/lddCyn0nic5dFo3VfJYrXYC', 'user3@galapgos.org'),  
  ('user4', '$2a$10$EsIMfxbJ6NuvwX7MDj4Wq0YFzLU9U/lddCyn0nic5dFo3VfJYrXYC', 'user4@galapgos.org');
```

```
select * from tbl_member;
```

## 권한 데이터 추가

```
insert into tbl_member_auth(username, auth)
values
  ('admin', 'ROLE_ADMIN'),
  ('admin', 'ROLE_MANAGER'),
  ('admin', 'ROLE_MEMBER'),
  ('user0', 'ROLE_MANAGER'),
  ('user0', 'ROLE_MEMBER'),
  ('user1', 'ROLE_MEMBER'),
  ('user2', 'ROLE_MEMBER'),
  ('user3', 'ROLE_MEMBER'),
  ('user4', 'ROLE_MEMBER');

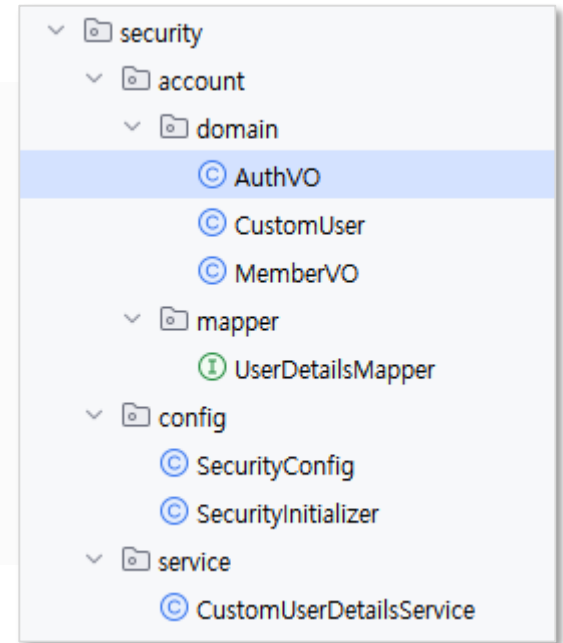
select * from tbl_member_auth order by auth;
```

- ADMIN(최고 관리자)
  - ROLE\_ADMIN, ROLE\_MANAGER, ROLE\_MEMBER
- MANAGER(일반 관리자)
  - ROLE\_MANAGER, ROLE\_MEMBER
- MEMBER(일반 회원)
  - ROLE\_MEMBER

## AuthVO.java

```
@Data
public class AuthVO implements GrantedAuthority {
    private String username;
    private String auth;

    @Override
    public String getAuthority() {
        return auth;
    }
}
```



### ○ GrantedAuthority 인터페이스

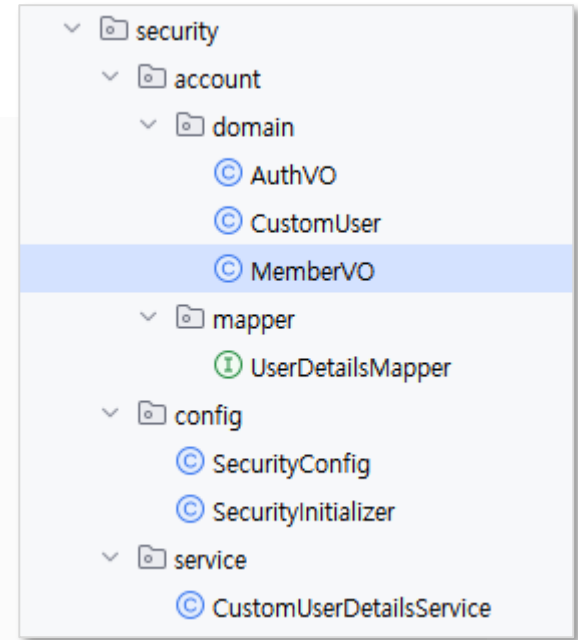
#### ■ 권한 정보 추출

```
public interface GrantedAuthority extends Serializable {
    String getAuthority();
}
```

## MemberVO.java

```
@Data
@NoArgsConstructor
@AllArgsConstructor
public class MemberVO {
    private String username;
    private String password;
    private String email;
    private Date regDate;
    private Date updateDate;

    private List<AuthVO> authList; // 권한 목록, join 처리 필요
}
```

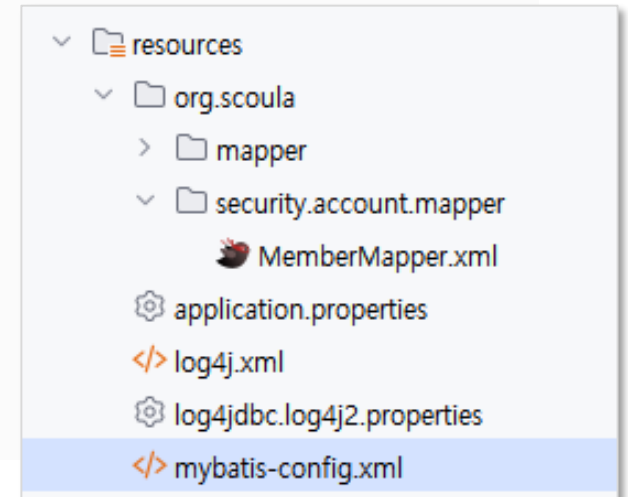


## resources::mybatis-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN" "http://mybatis.org/dtd/mybatis-3-config.dtd">

<configuration>
  <settings>
    <setting name="mapUnderscoreToCamelCase" value="true" />
  </settings>

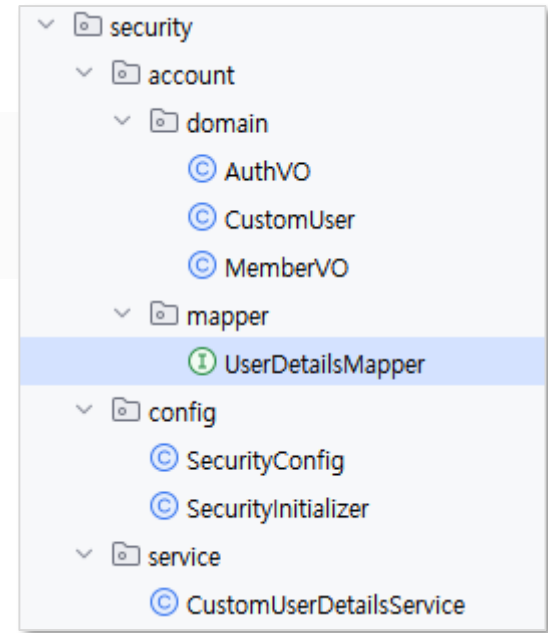
  <typeAliases>
    <package name="org.scoula.security.account.domain" />
  </typeAliases>
</configuration>
```



## UserDetailsMapper.java

```
public interface UserDetailsMapper {  
    public MemberVO get(String username);  
}
```

- tbl\_membe와 tbl\_member\_auth를 Join처리
  - MemberVO의 authList를 설정
  - MyBatis의 resultMap으로 Join 쿼리 결과를 MemberVO로 맵핑





## ✓ MyBatis의 <resultMap>

```
select m.username, password, email, reg_date, update_date, auth
from
  tbl_member m left outer join tbl_member_auth a
    on m.username = a.username
where m.username = 'admin';
```

List<AuthVO>로 구성

	username	password	email	reg_date	update_date	auth
1	admin	\$2a\$10\$EsIMfxbJ6NuvwX7M	admin@galapgos.org	2023-11-24 17:24:04	2023-11-24 17:24:04	ROLE_ADMIN
2	admin	\$2a\$10\$EsIMfxbJ6NuvwX7M	admin@galapgos.org	2023-11-24 17:24:04	2023-11-24 17:24:04	ROLE_MANAGER
3	admin	\$2a\$10\$EsIMfxbJ6NuvwX7M	admin@galapgos.org	2023-11-24 17:24:04	2023-11-24 17:24:04	ROLE_USER

1개의 MemberVO 인스턴스로 구성

```
@Data
public class MemberVO {
    private String username;
    private String password;
    private String email;
    private Date regDate;
    private Date updateDate;

    private List<AuthVO> authList;
}
```

## ✓ resultMap

### ○ select 쿼리 결과를 VO객체로 매핑하는 규칙 정의

```
<ResultMap type="VO 객체 클래스명" id="식별자">  
  <id property="VO 속성명" column="컬럼명"/>  
  <result property="VO 속성명" column="컬럼명"/>  
  ...  
</ResultMap>
```

### ○ 1:N의 Join 결과 매핑

```
<ResultMap type="VO 객체 클래스명" id="식별자">  
  <id property="VO 속성명" column="컬럼명"/>  
  <result property="VO 속성명" column="컬럼명"/>  
  ...  
  <collection property="부모객체의 컬렉션 속성명" resultMap="자식 테이블 매핑 resultmap ID">  
</ResultMap>
```

## ✓ MyBatis의 <resultMap>

```
<resultMap id="memberMap" type="MemberVO">
  <id property="username" column="username" />
  <result property="password" column="password" />
  <result property="email" column="email" />
  <result property="regDate" column="reg_date" />
  <result property="updateDate" column="update_date" />
```

```
  <collection property="authList" resultMap="authMap" />
</resultMap>
```

List<AuthVO>로 구성해줌

```
<resultMap id="authMap" type="AuthVO">
  <result property="username" column="username" />
  <result property="auth" column="auth" />
</resultMap>
```

## resources :: UserDetailsMapper.xml

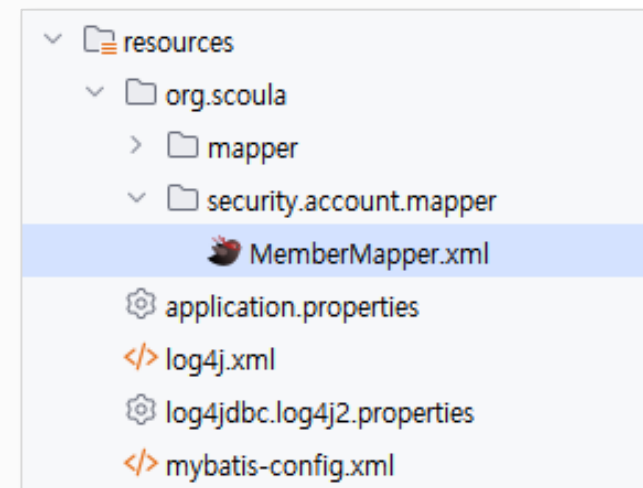
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="org.scoula.security.account.mapper.UserDetailsMapper">

    <resultMap id="authMap" type="AuthVO">
        <result property="username" column="username" />
        <result property="auth" column="auth" />
    </resultMap>

    <resultMap id="memberMap" type="MemberVO">
        <id property="username" column="username" />
        <result property="username" column="username" />
        <result property="password" column="password" />
        <result property="email" column="email" />
        <result property="regDate" column="reg_date" />
        <result property="updateDate" column="update_date" />

        <collection property="authList" resultMap="authMap" />
    </resultMap>
```



## resources :: UserDetailsMapper.xml

```
<select id="get" resultMap="memberMap">
  select m.username, password, email, reg_date, update_date, auth
  from
    tbl_member m left outer join tbl_member_auth a
      on m.username = a.username
  where m.username = #{username}
</select>
</mapper>
```

## SecurityConfig.java

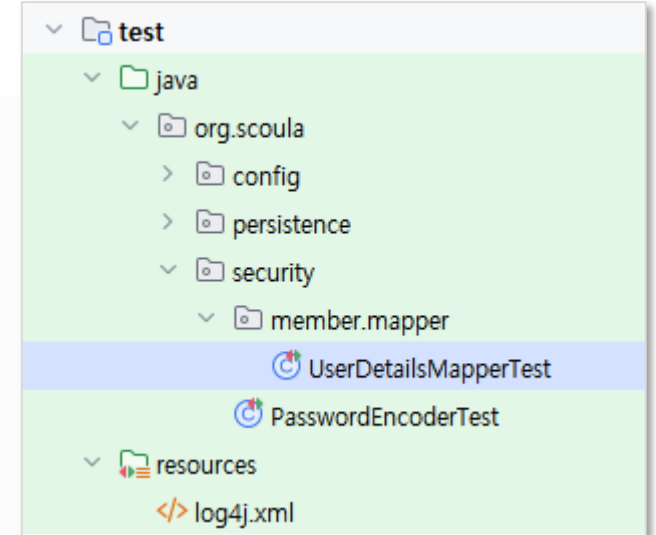
```
@Configuration
@EnableWebSecurity
@Log4j
@MapperScan(basePackages = {"org.scoula.security.account.mapper"})
public class SecurityConfig extends WebSecurityConfigurerAdapter {
    ...
}
```

## test :: UserDetailsMapperTest.java

```
@ExtendWith(SpringExtension.class)
@ContextConfiguration(classes = { RootConfig.class, SecurityConfig.class })
@Log4j
public class UserDetailsMapperTest {
    @Autowired
    private UserDetailsMapper mapper;

    @Test
    public void testGet() {
        MemberVO member = mapper.get("admin");
        log.info(member);

        for(AuthVO auth : member.getAuthList()) {
            log.info(auth);
        }
    }
}
```



```
INFO : org.scoula.security.account.mapper.UserDetailsMapperTest - MemberVO(username=admin,
password=$2a$10$EsIMfxbJ6NuvwX7MDj4Wq0YFzLU9U/lddCyn0nic5dFo3VfJYrXYC, email=admin@galappgos.org, regDate=Wed Jul 24
12:01:52 KST 2024, updateDate=Wed Jul 24 12:01:52 KST 2024, authList=[AuthVO(username=admin, auth=ROLE_ADMIN),
AuthVO(username=admin, auth=ROLE_MANAGER), AuthVO(username=admin, auth=ROLE_MEMBER)])
INFO : org.scoula.security.account.mapper.UserDetailsMapperTest - AuthVO(username=admin, auth=ROLE_ADMIN)
INFO : org.scoula.security.account.mapper.UserDetailsMapperTest - AuthVO(username=admin, auth=ROLE_MANAGER)
INFO : org.scoula.security.account.mapper.UserDetailsMapperTest - AuthVO(username=admin, auth=ROLE_MEMBER)
```