KB금융그룹 | 국민의 평생 금융파트너

인메모리말고 디비로
인증, 로그인 처리해보자

2025년 상반기 K-디지털 트레이닝

member 테이블

[KB] IT's Your Life

KB국민은행

member

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
```

권한 테이블을 따로 운영해야함 1:N관계!

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

member

데이터 추가

```
-- 테스트 사용자 추가
insert into tbl_member(username, password, email)
values
        ('admin', '$2a$10$EsIMfxbJ6NuvwX7MDj4WqOYFzLU9U/lddCyn0nic5dFo3VfJYrXYC', 'admin@galapgos.org'),
        ('user0', '$2a$10$EsIMfxbJ6NuvwX7MDj4WqOYFzLU9U/lddCyn0nic5dFo3VfJYrXYC', 'user0@galapgos.org'),
        ('user1', '$2a$10$EsIMfxbJ6NuvwX7MDj4WqOYFzLU9U/lddCyn0nic5dFo3VfJYrXYC', 'user1@galapgos.org'),
        ('user2', '$2a$10$EsIMfxbJ6NuvwX7MDj4WqOYFzLU9U/lddCyn0nic5dFo3VfJYrXYC', 'user2@galapgos.org'),
        ('user3', '$2a$10$EsIMfxbJ6NuvwX7MDj4WqOYFzLU9U/lddCyn0nic5dFo3VfJYrXYC', 'user3@galapgos.org'),
        ('user4', '$2a$10$EsIMfxbJ6NuvwX7MDj4WqOYFzLU9U/lddCyn0nic5dFo3VfJYrXYC', 'user4@galapgos.org');


select * from tbl_member;
```

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

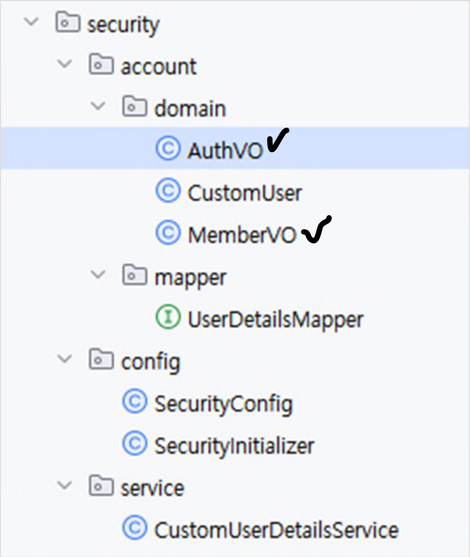
member

security.account.domain.AuthVO.java

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

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

너의 권한이 뭐니



```
security
    account
        domain
            AuthVO ✓
            CustomUser
            MemberVO ✓
        mapper
            UserDetailsMapper
    config
        SecurityConfig
        SecurityInitializer
    service
        CustomUserDetailsService
```

- GrantedAuthority 인터페이스
  - 권한 정보 추출

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

member

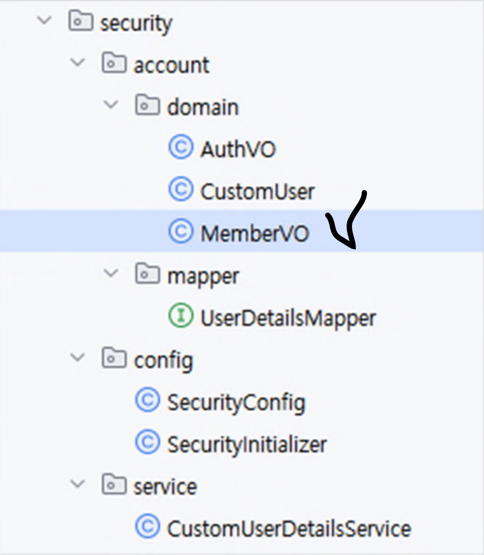
security.account.domain.MemberVO.java

```
@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

}
```

resultMap으로 mybatis



```
security
  account
    domain
      AuthVO
      CustomUser
      MemberVO
    mapper
      UserDetailsMapper
  config
    SecurityConfig
    SecurityInitializer
  service
    CustomUserDetailsService
```

유저는 한개이상의 권한을 반드시 갖고 있어야 한다.

member

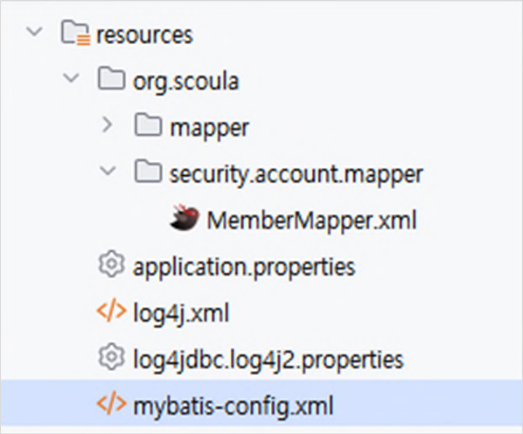
resources::mybatis-config.xml

```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>
```



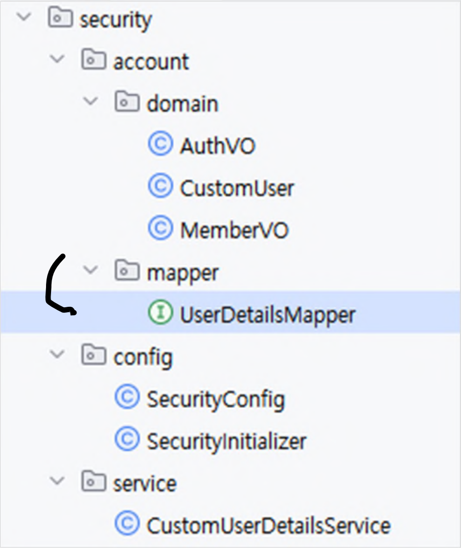
```
resources
    org.scoula
        mapper
        security.account.mapper
            MemberMapper.xml
    application.properties
    log4j.xml
    log4jdbc.log4j2.properties
    mybatis-config.xml
```

member

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

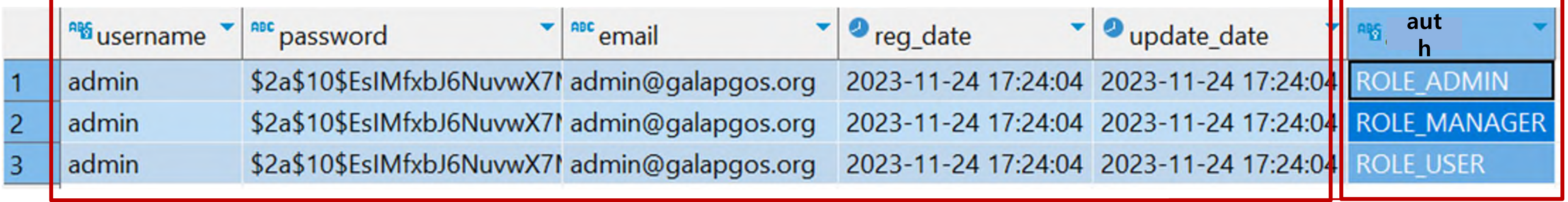


```
security
  account
    domain
      AuthVO
      CustomUser
      MemberVO
    mapper
      UserDetailsMapper
  config
    SecurityConfig
    SecurityInitializer
  service
    CustomUserDetailsService
```

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$10$EsIMfxbJ6NuvwX7I	admin@galapgos.org	2023-11-24 17:24:04	2023-11-24 17:24:04	ROLE_ADMIN
2	admin	$2a$10$EsIMfxbJ6NuvwX7I	admin@galapgos.org	2023-11-24 17:24:04	2023-11-24 17:24:04	ROLE_MANAGER
3	admin	$2a$10$EsIMfxbJ6NuvwX7I	admin@galapgos.org	2023-11-24 17:24:04	2023-11-24 17:24:04	ROLE_USER

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

            private List<AuthVO> authList
;
}
```

1개의 MemerVO 인스턴스로 구성

member

ResultMap

- select 쿼리 결과를 VO객체로 맵핑하는 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 맵핑

```
<ResultMap type="VO 객체 클래스명" id="식별자">
   <id property="VO 속성명" column="컬럼명"/>
   <result  property="VO 속성명" column="컬럼명"/>
    ...
   <collection property="부모객체의 콜렉션 속성명" resultMap="자식 테이블 맵핑  resutlmap 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>

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

List<AuthVO>로 구성해줌

resources :: UserDetailsMapper.xml

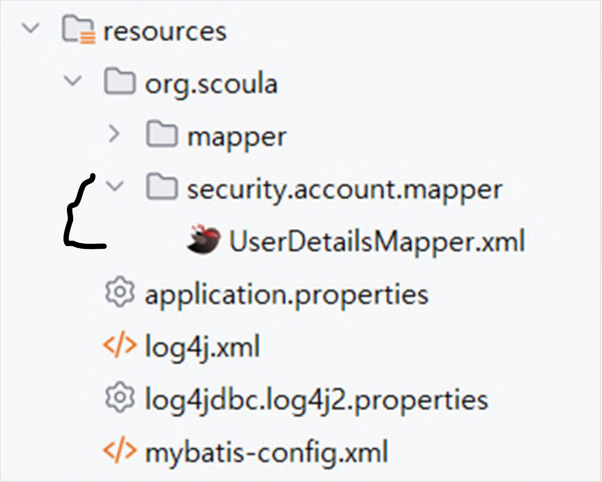
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<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
  org.scoula
    mapper
    security.account.mapper
      UserDetailsMapper.xml
  application.properties
  log4j.xml
  log4jdbc.log4j2.properties
  mybatis-config.xml
```

member

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>
```

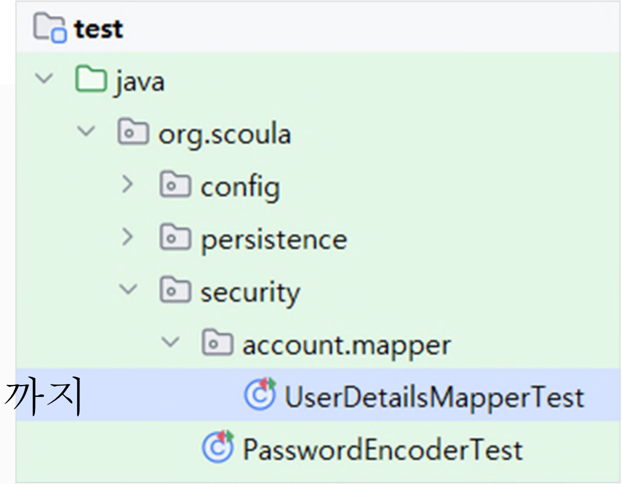
member

SecurityConfig.java

```
@Configuration
@EnableWebSecurity
@Log4j2
@MapperScan(basePackages  = {"org.scoula.security.account.mapper"})
public class SecurityConfig extends WebSecurityConfigurerAdapter {
...
}
```

✓ 사용할 매퍼 스캔

test :: UserDetailsMapperTest.java



```
@ExtendWith(SpringExtension.class)
@ContextConfiguration(classes = { RootConfig.class, SecurityConfig.class })
@Log4j2
public class UserDetailsMapperTest {
            @Autowired
            private UserDetailsMapper mapper;

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

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

DB 설정 매퍼 사용 V

원래는 서비스에서
vo를 매퍼를 통해 받고
dto로 변환시키고 반환하는 거 까지

V

이번예시에서는
DTO로 변환X

spring security
에서 발동시켜보자

```
 INFO  jdbc.sqlonly(sqlOccurred:228) - 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'

 INFO  org.scoula.security.account.mapper.UserDetailsMapperTest(get:24) - MemberVO(username=admin, password=$2a$10$EsIMfxbJ6NuvwX7MDj4WqOYFzLU9U/Id
dCyn0nic5dFo3VfJYrXYC, email=admin@galapgos.org, regDate=Fri Jan 17 13:12:21 KST 2025, updateDate=Fri Jan 17 13:12:21 KST 2025, 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(get:27) - AuthVO(username=admin, auth=ROLE_ADMIN)
 INFO  org.scoula.security.account.mapper.UserDetailsMapperTest(get:27) - AuthVO(username=admin, auth=ROLE_MANAGER)
 INFO  org.scoula.security.account.mapper.UserDetailsMapperTest(get:27) - AuthVO(username=admin, auth=ROLE_MEMBER)
```