

스프링 공부 4주차

모각소 5차(02.24)

박수인

프로젝트 세팅

- Spring Initializr 이용
- Dependenceise : Spring Boot DevTools, Lombok, Thymeleaf, H2 Database, spring web, mybatis framework 추가
- Project : gradle project

Project

☒ Gradle - Groovy

Language

☒ Java ☐ Kotlin ☐ Groovy

Gradle - Kotlin

Maven

Spring Boot

☐ 3.0.3 (SNAPSHOT) ☒ 3.0.2 ☐ 2.7.9 (SNAPSHOT) ☐ 2.7.8

Project Metadata

Group

com.example

Artifact

bulletin-board

Name

bulletin-board

Description

Demo project for Spring Boot

Package name

com.example.bulletin-board

Packaging

☒ Jar ☐ War

Java

☐ 19 ☐ 17 ☒ 11 ☐ 8

Dependencies

ADD DEPENDENCIES... ⌘ + B

Spring Boot DevTools

DEVELOPER TOOLS

Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

—

Lombok

DEVELOPER TOOLS

Java annotation library which helps to reduce boilerplate code.

—

Thymeleaf

TEMPLATE ENGINES

A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

—

H2 Database

SQL

Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.

—

Spring Web

WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

—

MyBatis Framework

SQL

Persistence framework with support for custom SQL, stored procedures and advanced mappings. MyBatis couples objects with stored procedures or SQL statements using a XML descriptor or annotations.

—

프로젝트 세팅

- 데이터베이스 - H2DB 이용
 - **인메모리 DB** : 서버(Server)모드 & 임베디드(Embedded)모드 지원
- 별도 설치 과정 X

English Preferences Tools Help

Login

Saved Settings: Generic H2 (Embedded)

Setting Name: Generic H2 (Embedded) Save Remove

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:mem:testdb

User Name: sa

Password:

Connect Test Connection

Auto commit

Max rows: 1000

Auto complete Off

jdbc:h2:file:/Users/suin/spring/bo

Run Run Selected Auto complete Clear

SQL statement:

TBL_BOARD

TBL_MEMBER

INFORMATION_SCHEMA

Sequences

Users

H2 1.4.200 (2019-10-14)

select * from tbl_board

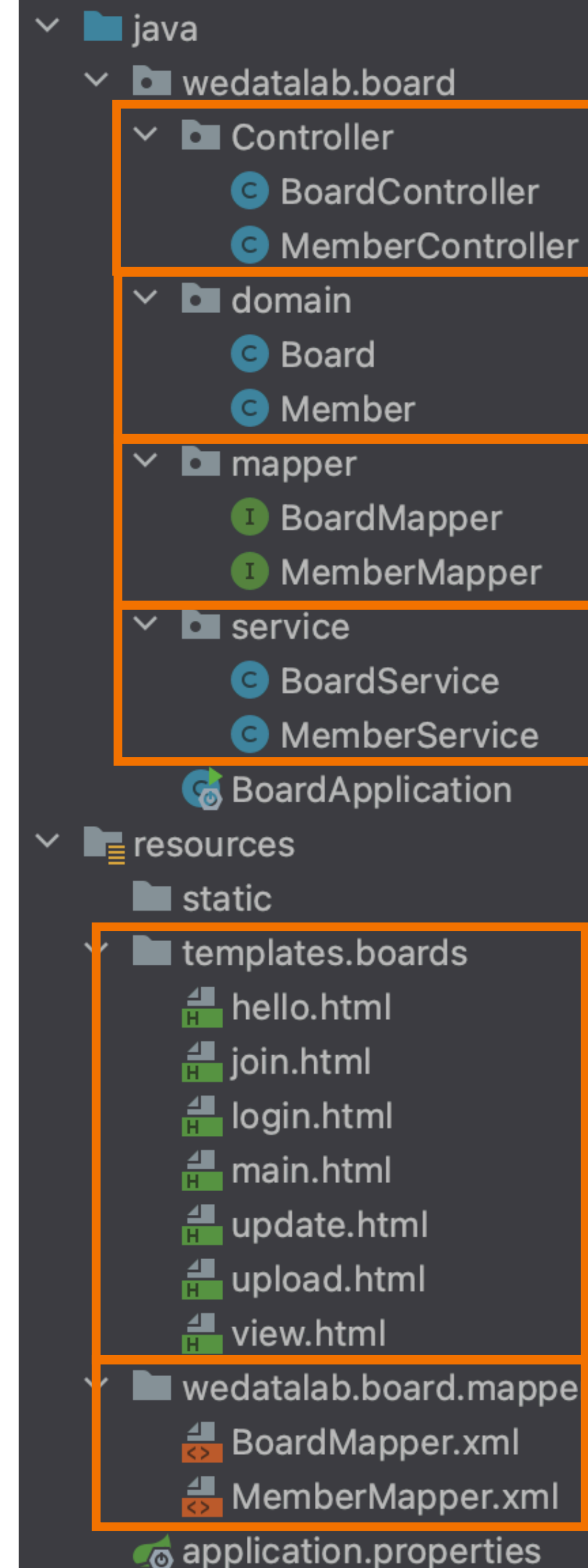
select * from tbl_board;

BOARDID	TITLE	CONTENT	NAME	READ
1	제목	내용	이름	0
2	제목2	내용22	이름2	1
3	제목3	내용3	이름3	0
4	제목4	내용4	이름4	0
5	제목1	내용1	이름1	1
6	제목2	내용2	이름2	1
7	제목3	내용3	이름3	0
8	제목4	내용4	이름4	0
9	제목5	내용5	이름5	1
10	제목6	내용6	이름6	0
11	제목7	내용7	이름7	0
12	제목8	내용8	이름8	0
13	제목9	내용9	이름9	0
14	ㅎㅎㅎ	ㅋㅋㅎㅎ	이름10	1
15	수정	수정 테스트	박수인	0
16	조회수 테스트	내용	박수인	2

(16 rows, 28 ms)

프로젝트 구조

- bulletin-board/src/main 구조 ➡



데이터베이스 구조

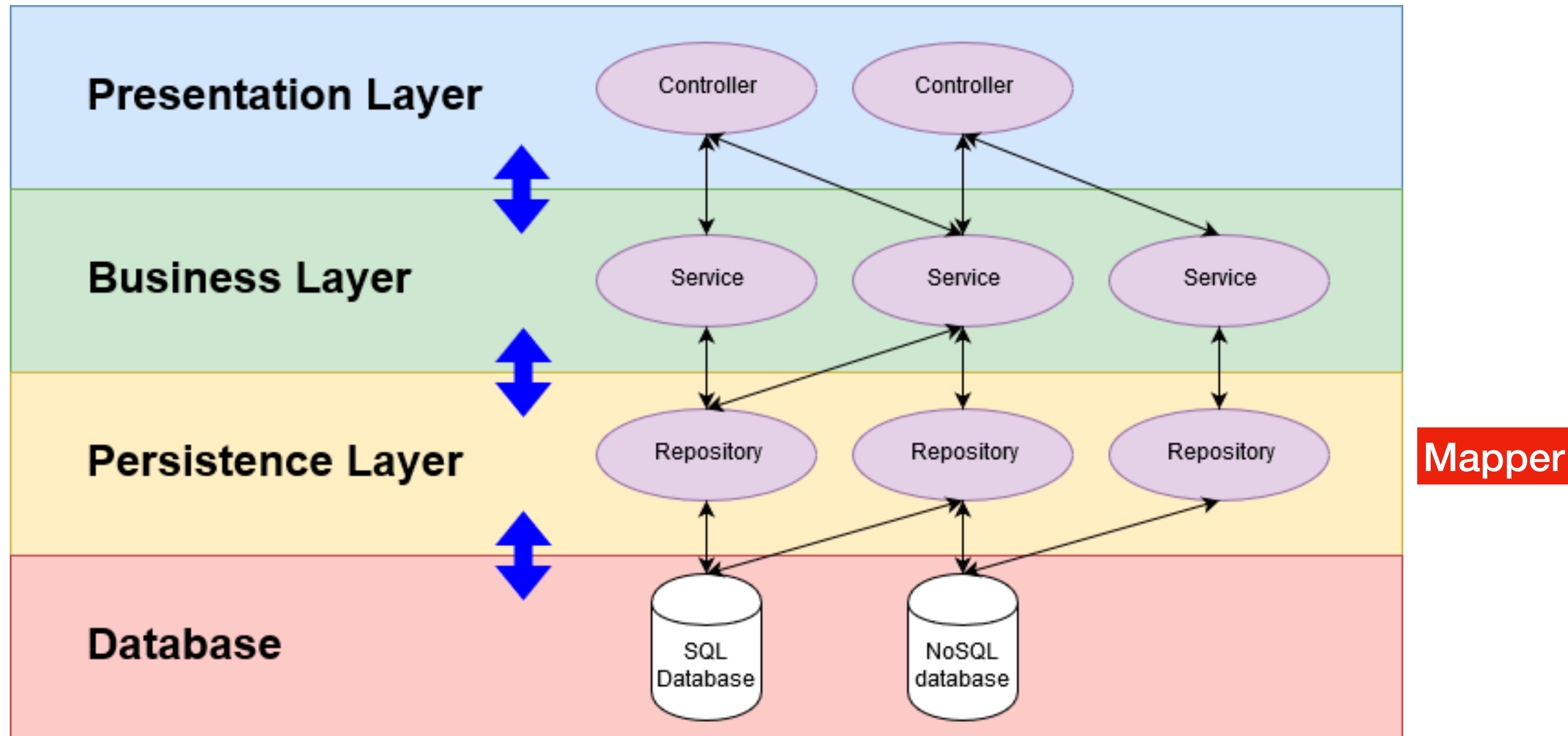
Member

```
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class Member {
    no usages
    private Long memberId; PK
    no usages
    private String name;
    no usages
    private String password;
}
```

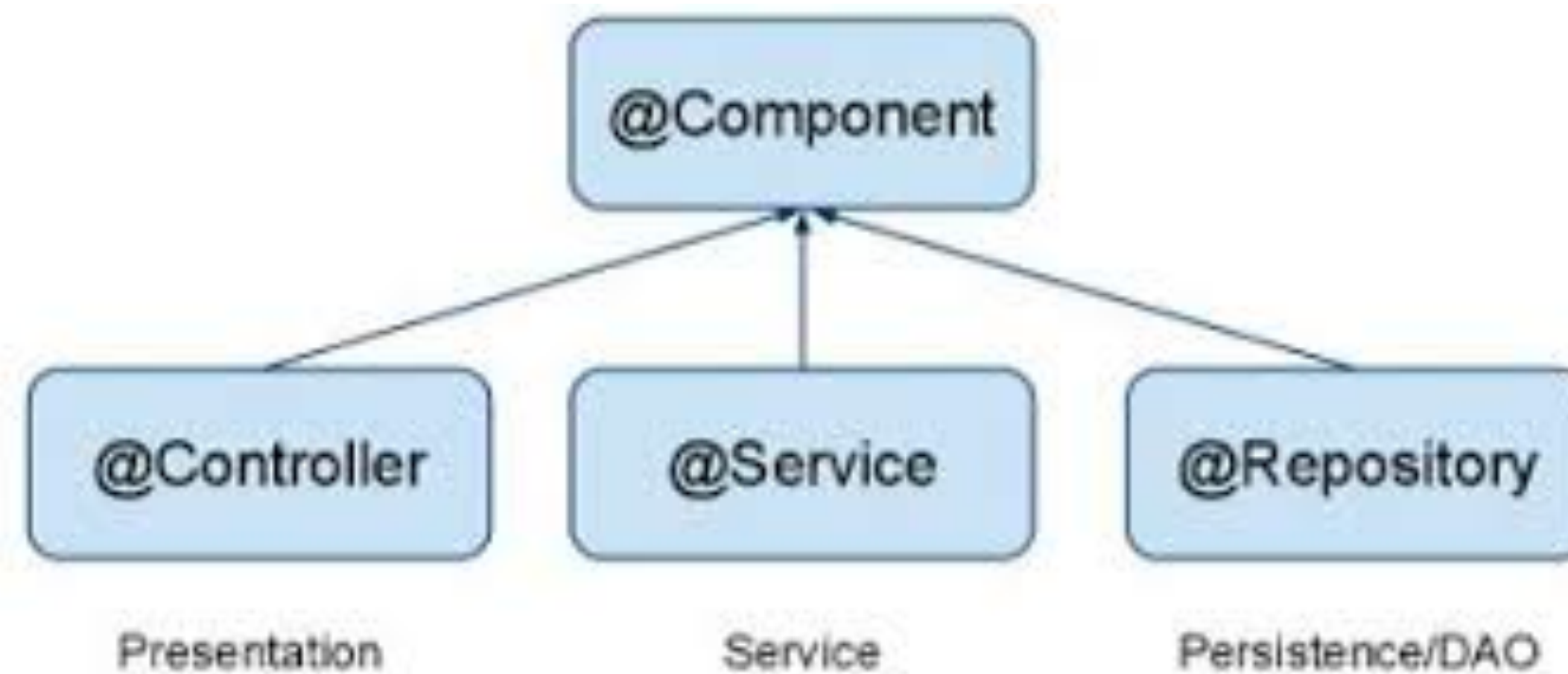
Board

```
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class Board {
    no usages
    private Long boardId; PK
    no usages
    private String title;
    no usages
    private String content;
    no usages
    private int read;
    no usages
    private Long memberId; FK
}
```


데이터 접근 구조



@Component 애노테이션 구조



@Controller

```
@RequestMapping("/board/**")
```

```
@RequiredArgsConstructor
```

```
public class BoardController {
```

9 usages

```
private final BoardService service;
```

no usages suin park

```
@GetMapping("/hello")
```

```
public String Hello() { return "/boards/hello"; }
```

@Service

```
@RequiredArgsConstructor
```

```
@Transactional(readonly = true)
```

```
public class BoardService {
```

7 usages

```
private final BoardMapper boardMapper;
```

@Repository

```
public interface BoardMapper {
```

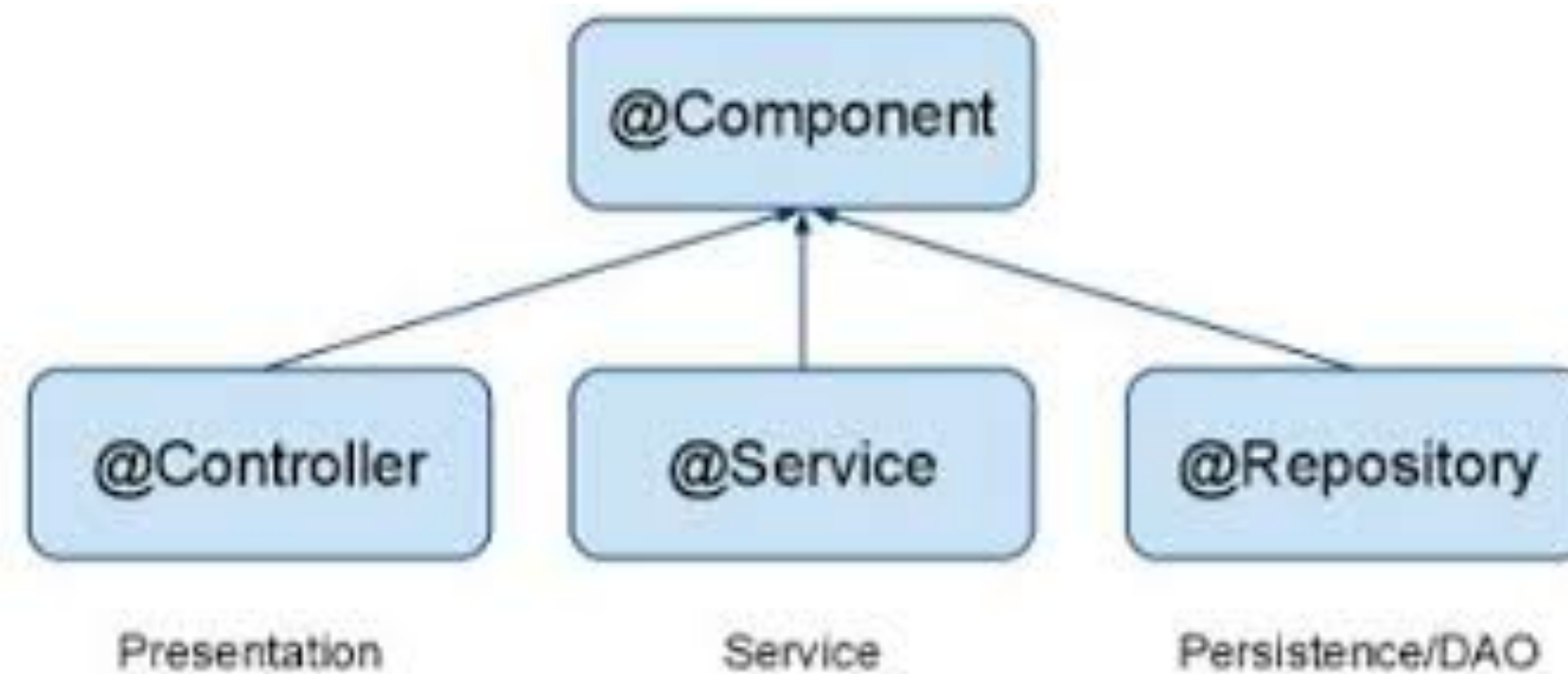
1 usage new *

```
int boardCount();
```

1 usage suin park

```
Board getBoard(Long boardId)
```

@Component 애노테이션 구조



@Controller

```
@RequestMapping("/board/**")
```

```
@RequiredArgsConstructor
```

```
public class BoardController {
```

9 usages

```
private final BoardService service;
```

no usages suin park

```
@GetMapping("/hello")
```

```
public String Hello() { return "/boards/hello"; }
```

@Service

```
@RequiredArgsConstructor
```

```
@Transactional(readonly = true)
```

```
public class BoardService {
```

7 usages

```
private final BoardMapper boardMapper;
```

@Repository

```
public interface BoardMapper {
```

1 usage new *

```
int boardCount();
```

1 usage suin park

```
Board getBoard(Long boardId)
```


Mapper Interface

BoardMapper 인터페이스

- SQL문을 호출

BoardMapper.xml 파일

@Repository

public interface BoardMapper {

1 usage new *

int boardCount();

1 usage suin park

Board getBoard(Long boardId);

```
<?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="wedatalab.board.mapper.BoardMapper">

    <select id="boardCount" resultType="int">
        SELECT count(boardId) AS cnt FROM tbl_board
    </select>

    <select id="getBoard" parameterType="Long" resultType="wedatalab.board.domain.Board">
        SELECT
            *
        FROM tbl_board where boardId=#{boardId};
    </select>

</mapper>
```