

# 61. Spring Framework 8.0 출시!

- ① 'spring-webmvc' 확장 버전을 출시한 이후

✓ ( spring 6.x → Jakarta EE 11 & 9.x      jakarta.\*      Tomcat 10.x  
      spring 5.x → JavaEE (8.x) = Jakarta EE (8.1)      jaxws.\*      Tomcat 9.x )

- ② 스트링 애플리케이션으로 뷰를

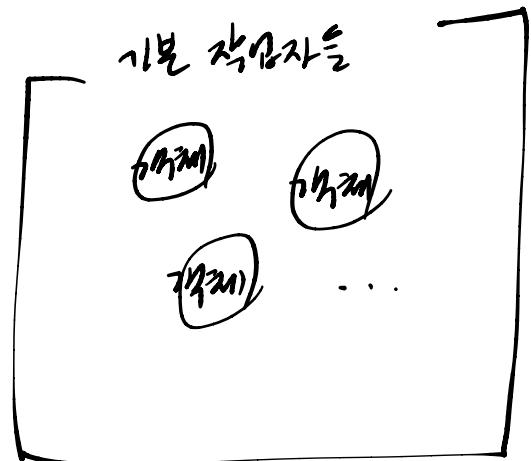
- ③    " 웹으로 뷰를

- ④    " Application 뷰로 AppConfig 123

- ⑤    " 콘트롤러로 뷰를

- ⑥    " IoC 컨테이너로 뷰를

# \* Spring Framework



이런 기능을 처리할 때  
적용자가 등록되어 있으려면  
(인증)  
그 권한을 실행(액션호출)하는데  
처리된다.  
없으면 예외를 던져서 기능을 무시한다.

기능 추가?

그 기능을 수행할 적용자를 등록  
(액션호출)



- ① 적용자를 사용해서 등록
- ② (액션호출) 설정을 통해  
(XML)

\* 요청 자세히 봐서 요청 헤더의 자세히 봐

<form>

```
<input name="email"/>  
<input name="password"/>  
<button>로그인</button>
```

POST로

</form>

요청 헤더

요청 자세히 봐

email=aaa@test.com & password=1111

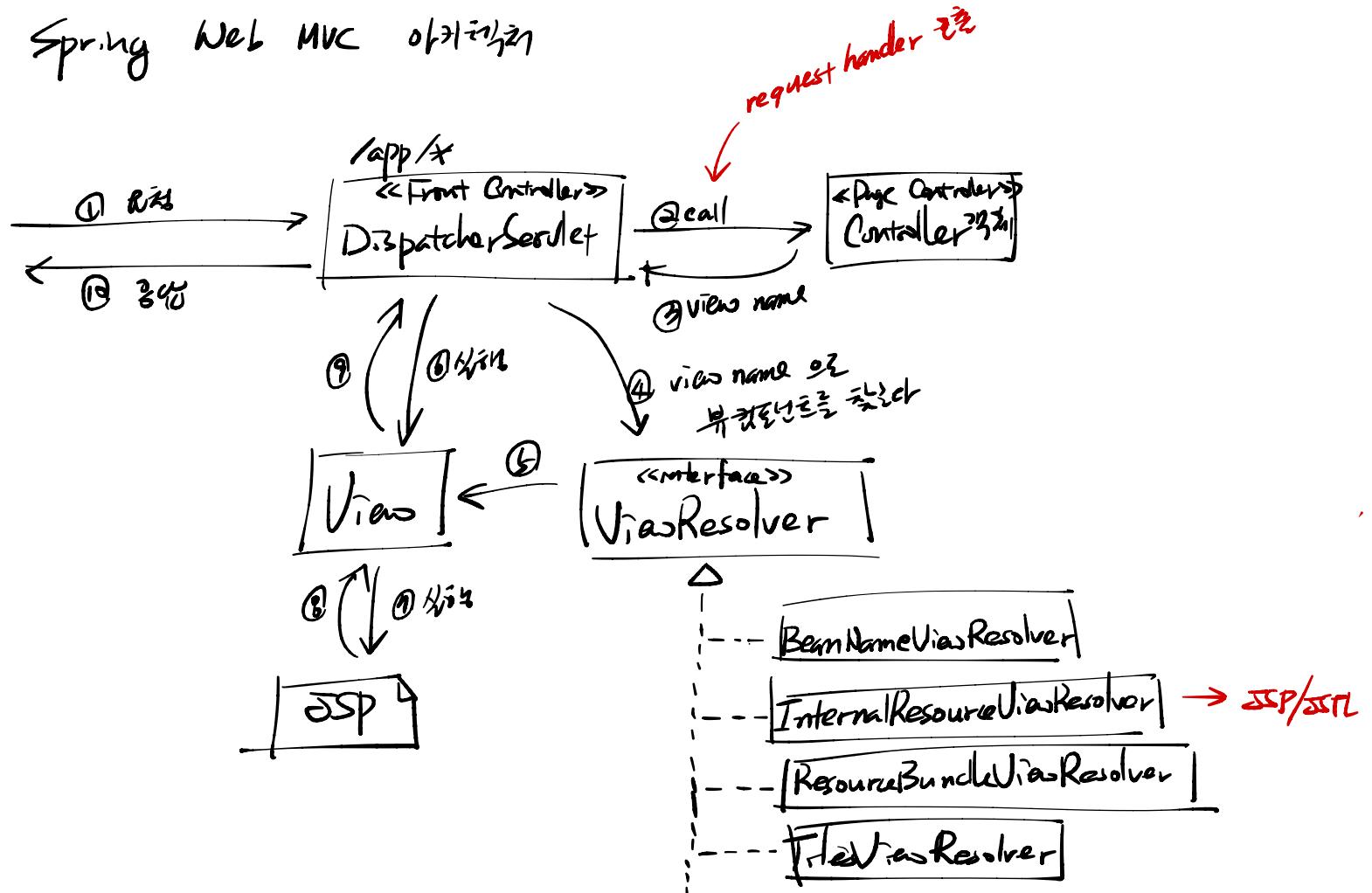
String

login(String email, String password) {

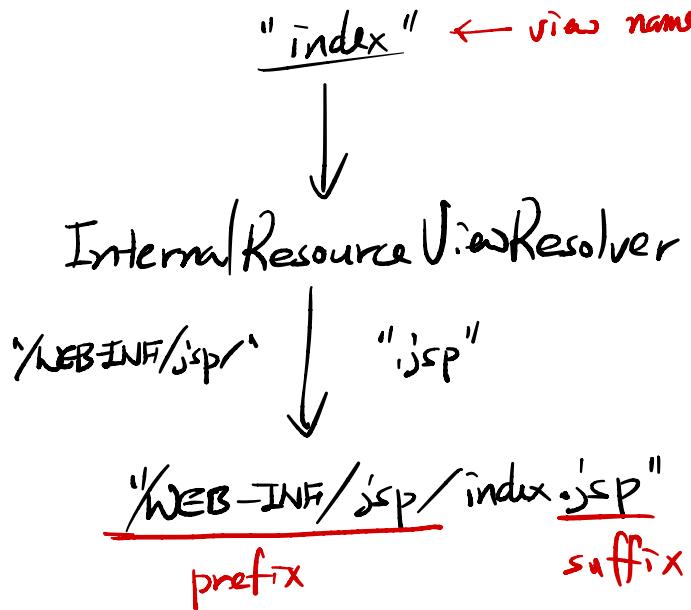
요청 헤더의 자세히 봐

}

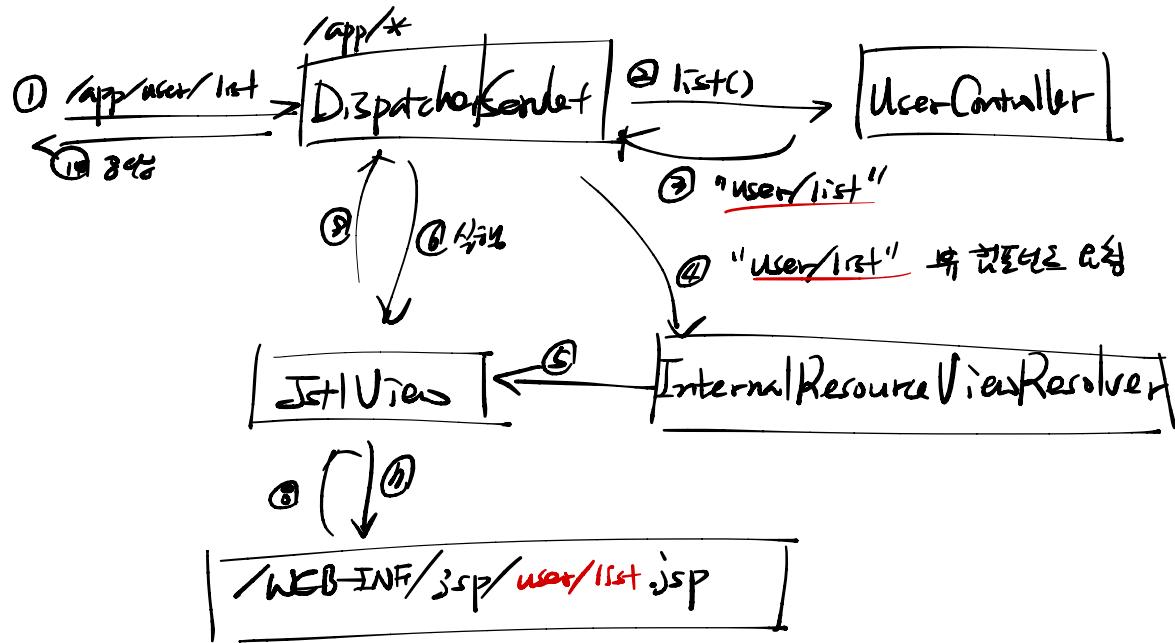
## \* Spring Web MVC 07/27/21



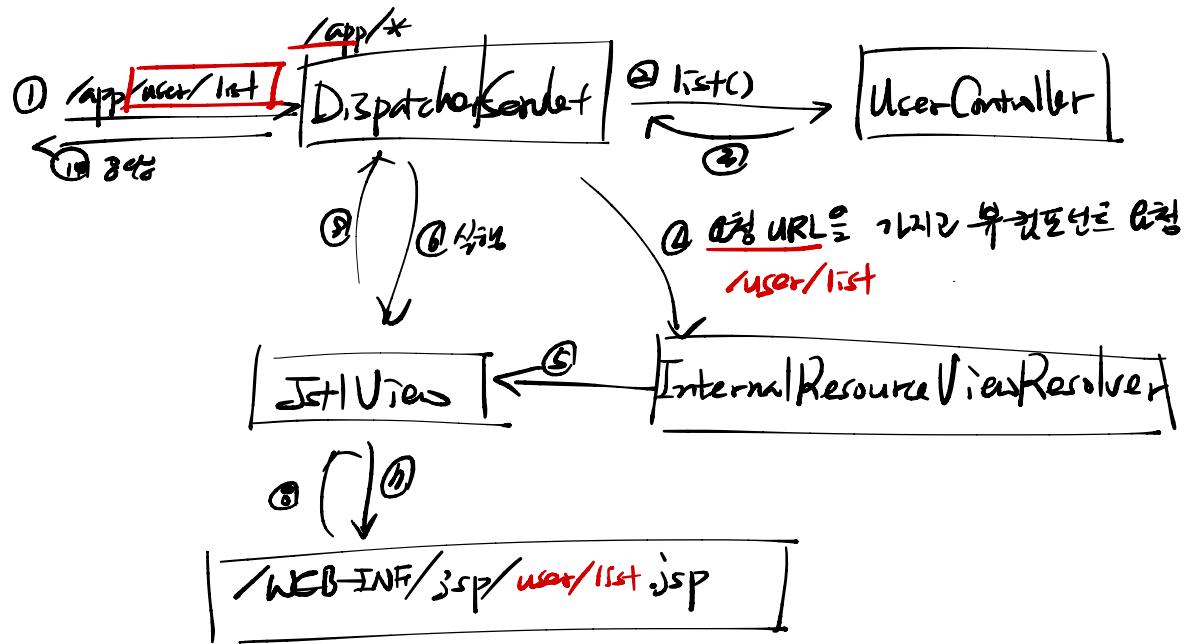
- \* Internal/Resource ViewResolver



\* view name %\*% can

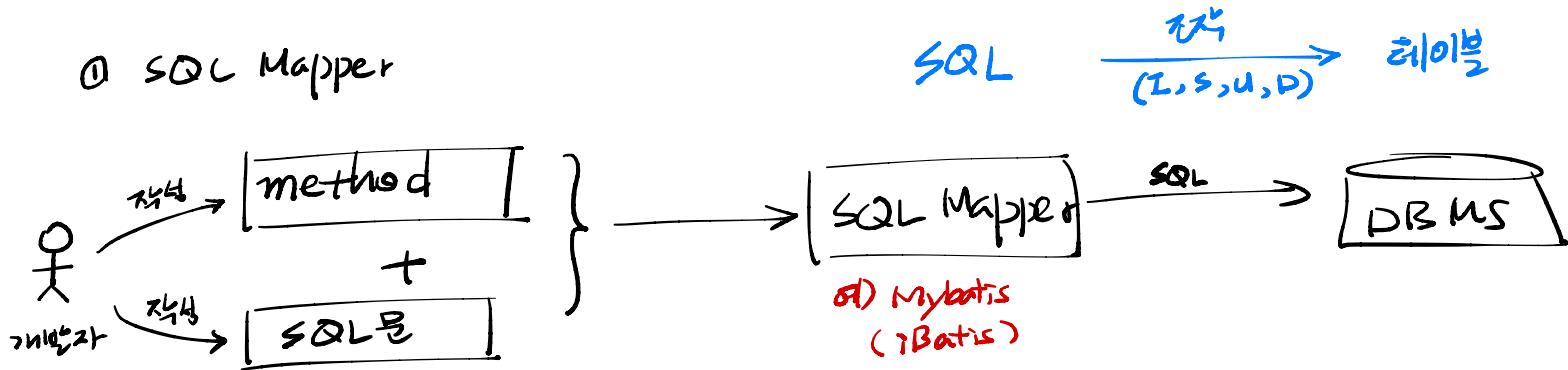


\* view name  $\stackrel{\text{보통}}{\rightarrow}$  controller

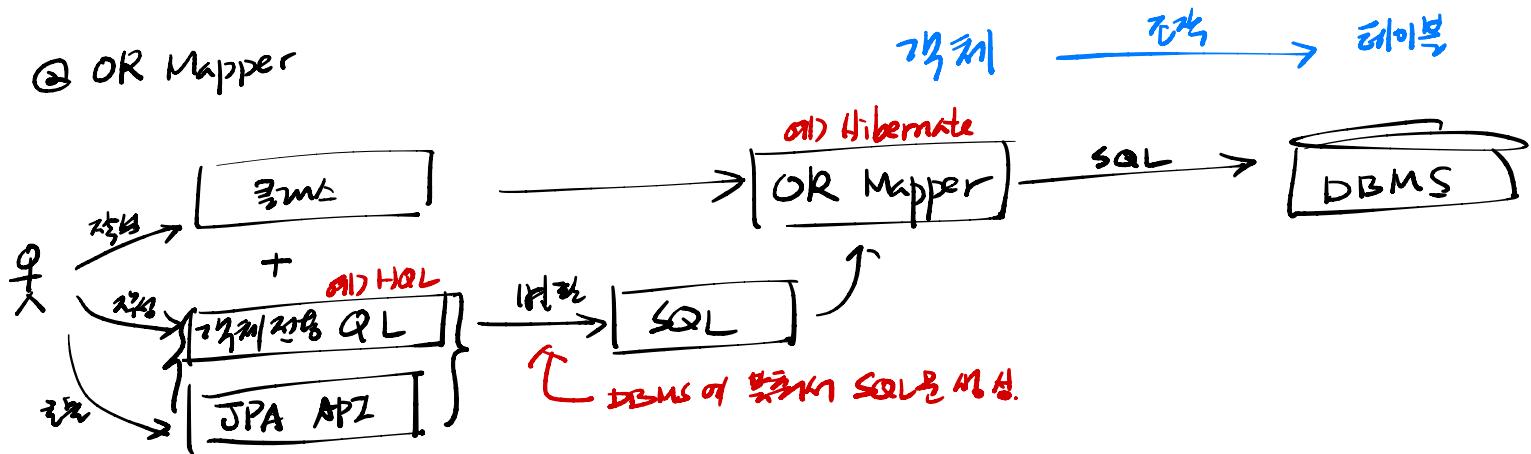


## \* SQL Mapper vs OR Mapper

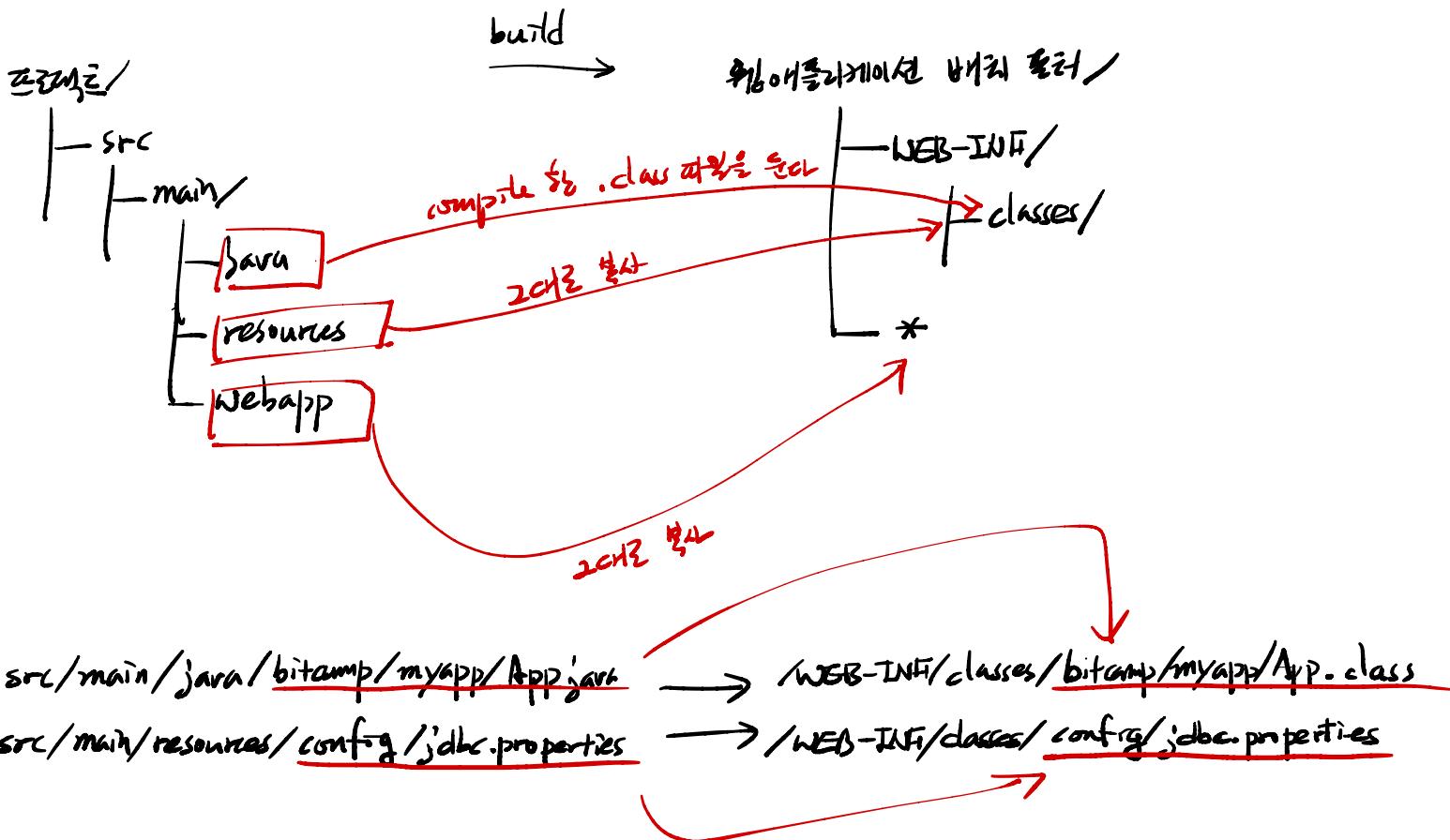
### ① SQL Mapper



### ② OR Mapper



## \* build et classpath



\* DaoFactory

↑  
생성  
DAO

SqISessionTemplate

↑  
Mybatis  
가져온  
DAO

[BoardDao]

insert()

[DaoFactory]

↳ sqISession.insert("BoardDao.insert", -);

[SqISessionTemplate]

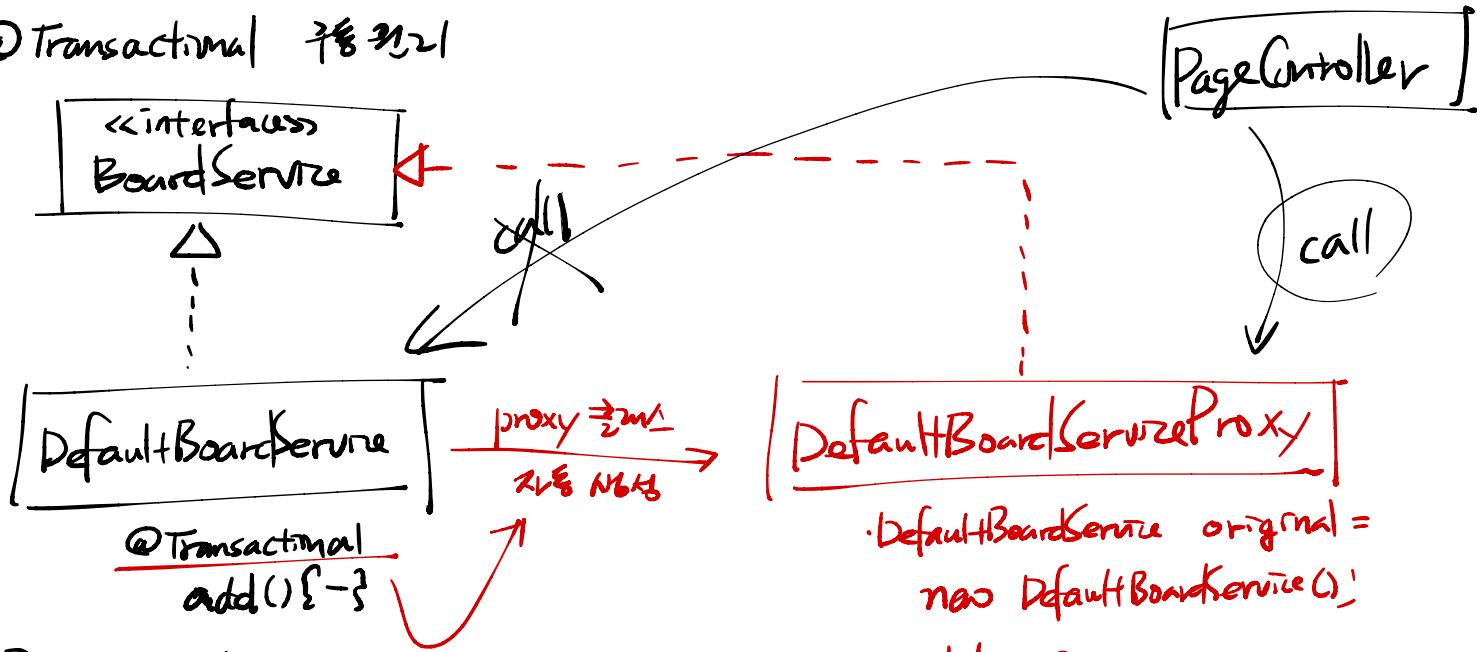
↳ sqISession.insert(

"bifrap-myapp.dao.BoardDao.insert",  
- );

SQL  
실행  
하기위한  
SQL문  
생성  
하기위한  
Template

MyBatis  
Template.

\* @Transactional 78 31, 21



. DefaultBoardService original =  
new DefaultBoardService();

. add () {

```

    try {
        original.add();
        txManager.commit();
    } catch {
        txManager.rollback();
    }
}
  
```

\* AOP의 특징, 예제

↳ 기존 코드를 대체하지 않고  
추가 기능을 적용하는 방법

↳ proxy 구현은 어렵지만 ("proxy를 쓰면 어렵다")

이 애플리케이션  
을 만들면서  
있을 경우

\* VPC (Virtual Private Cloud) - 퍼블릭 네트워크 대신에 내부망을 사용하는 네트워크

