

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 스프링 아�플리케이션으로 뷰를

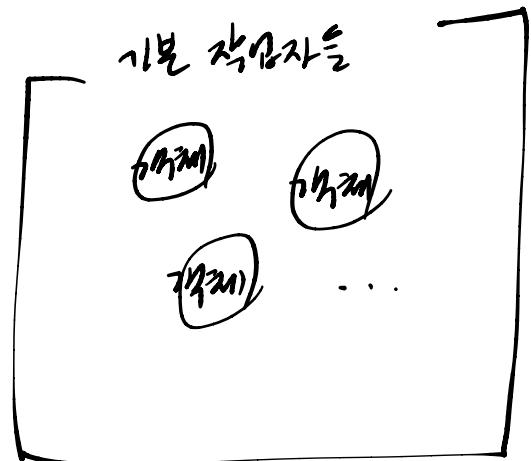
③ " 자바로 뷰를

④ " Annotation 방식 AppConfig 사용

⑤ " XML 기반 설정 파일

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 적용 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>  
</form>
```

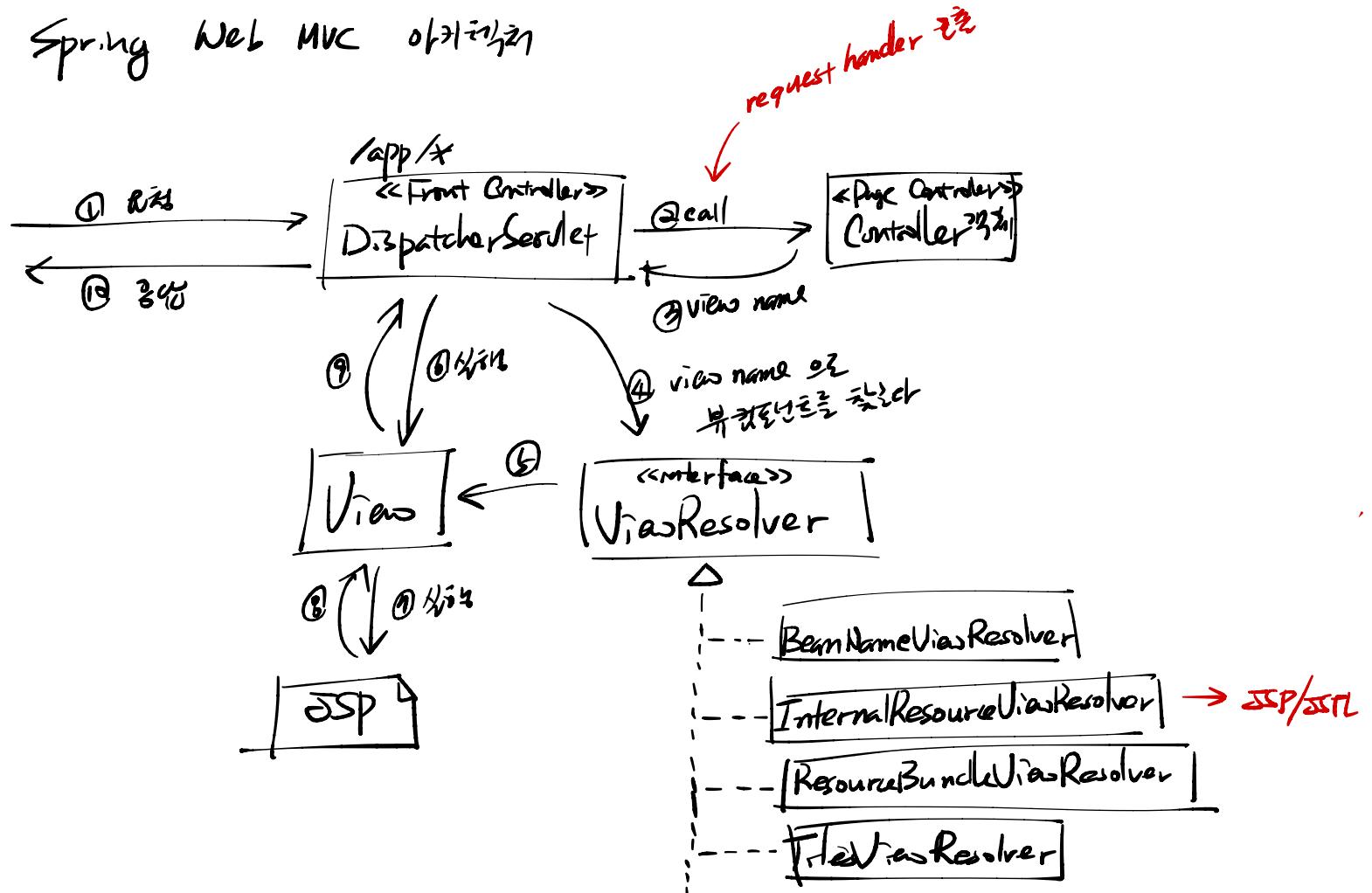
POST로

요청 자세히
email=aaa@test.com & password=1111

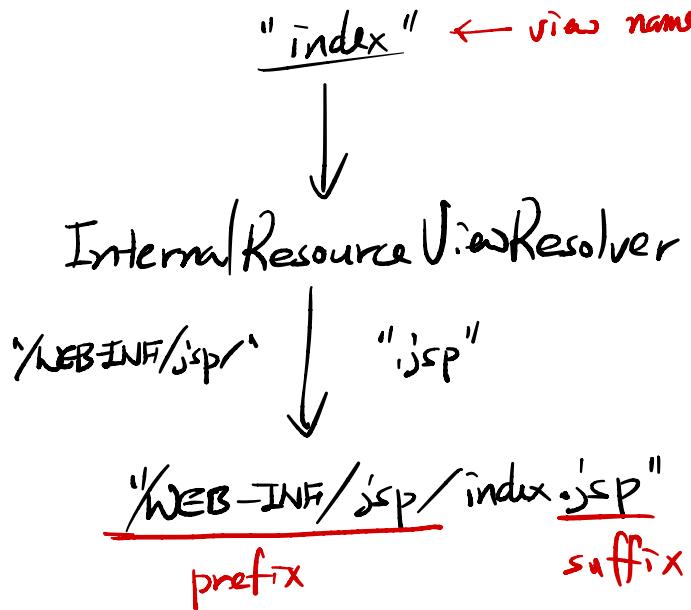
String login(String email, String password) {
 // ...
}

요청 헤더의 자세히

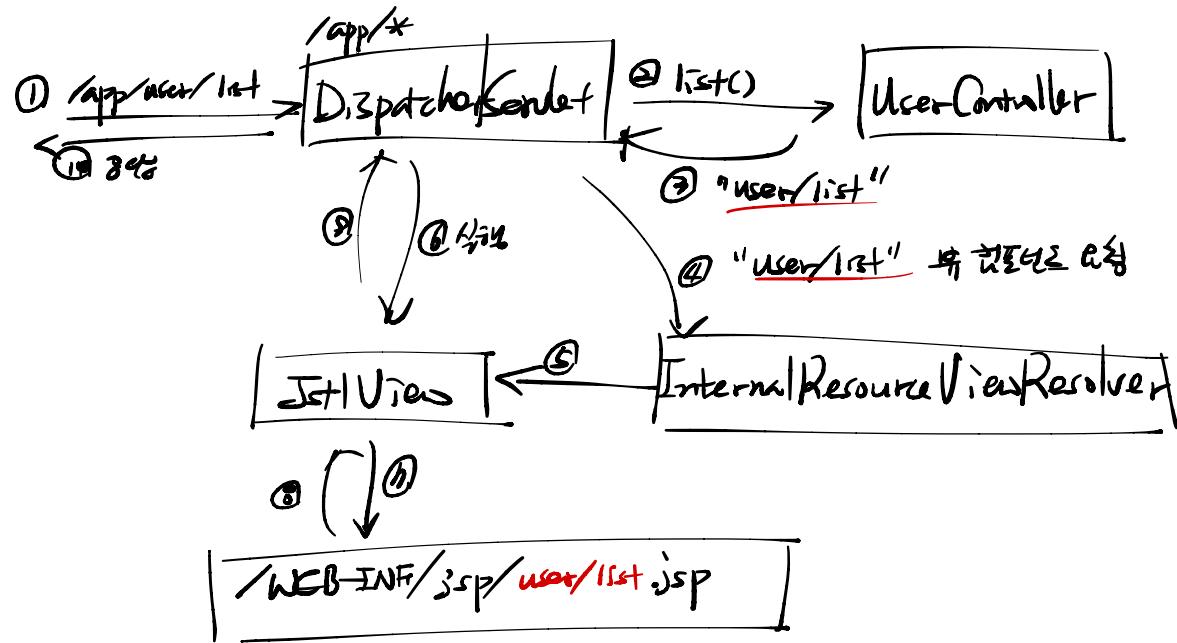
* Spring Web MVC 07/27/21



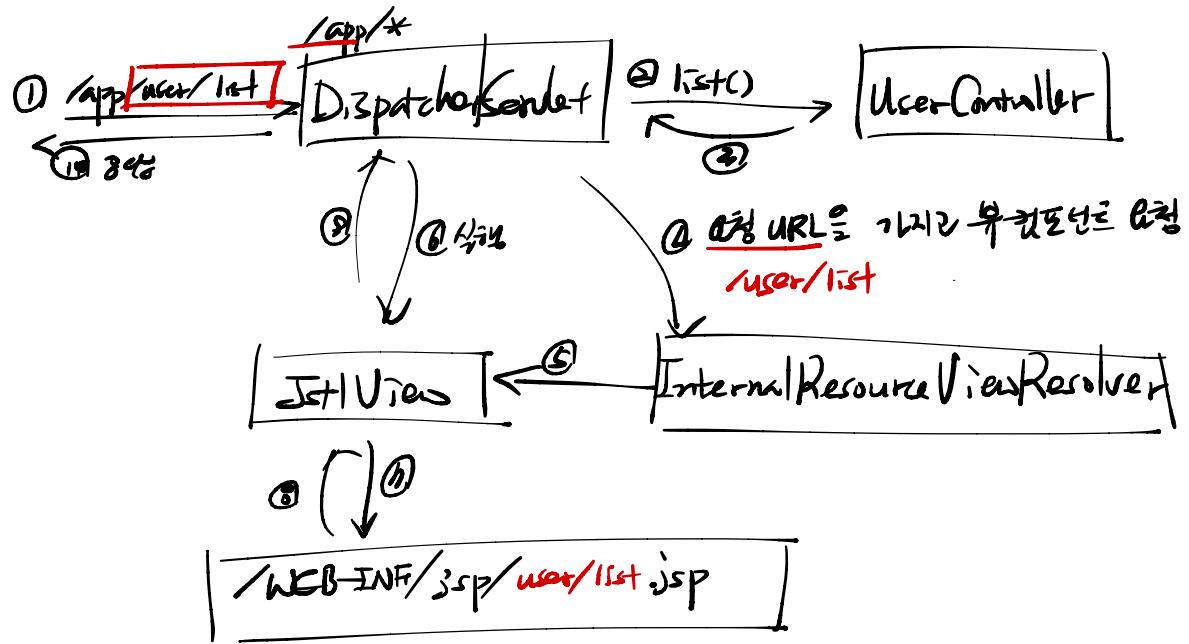
- * Internal/Resource ViewResolver



* view name %*% can

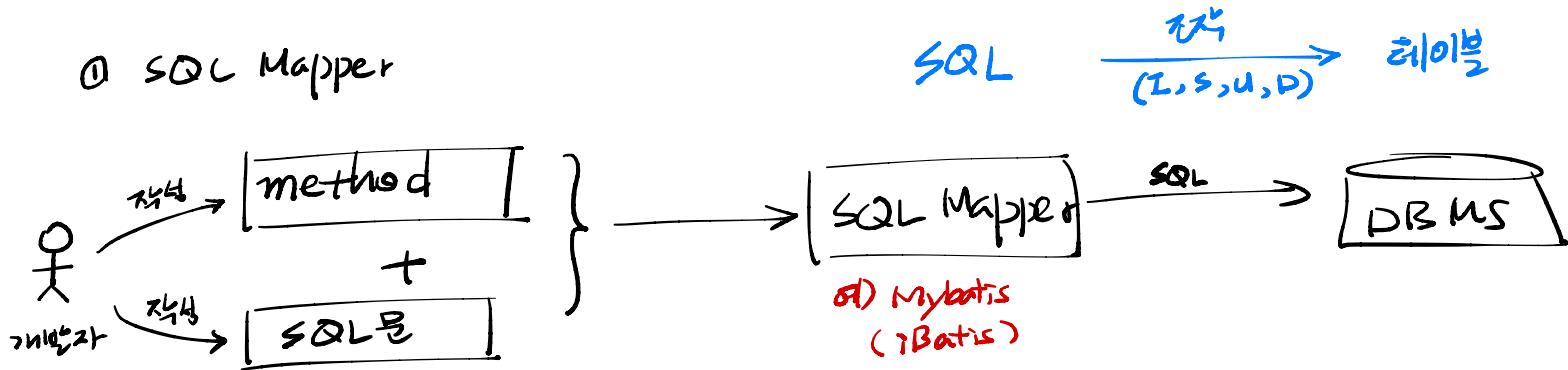


* view name $\stackrel{\text{보통}}{\rightarrow}$ controller

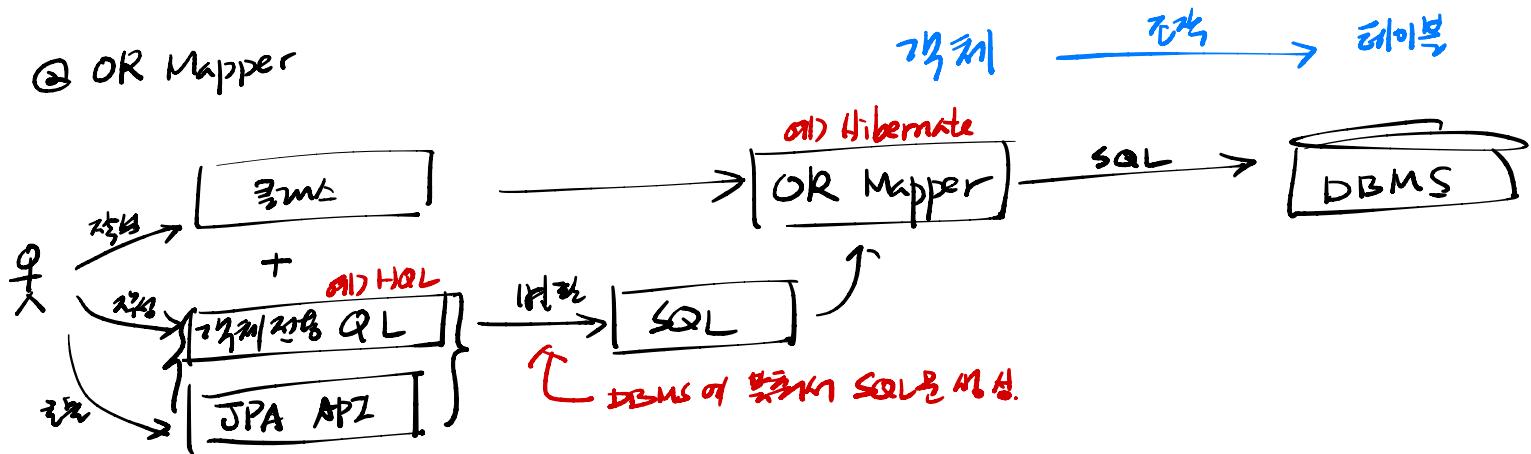


* SQL Mapper vs OR Mapper

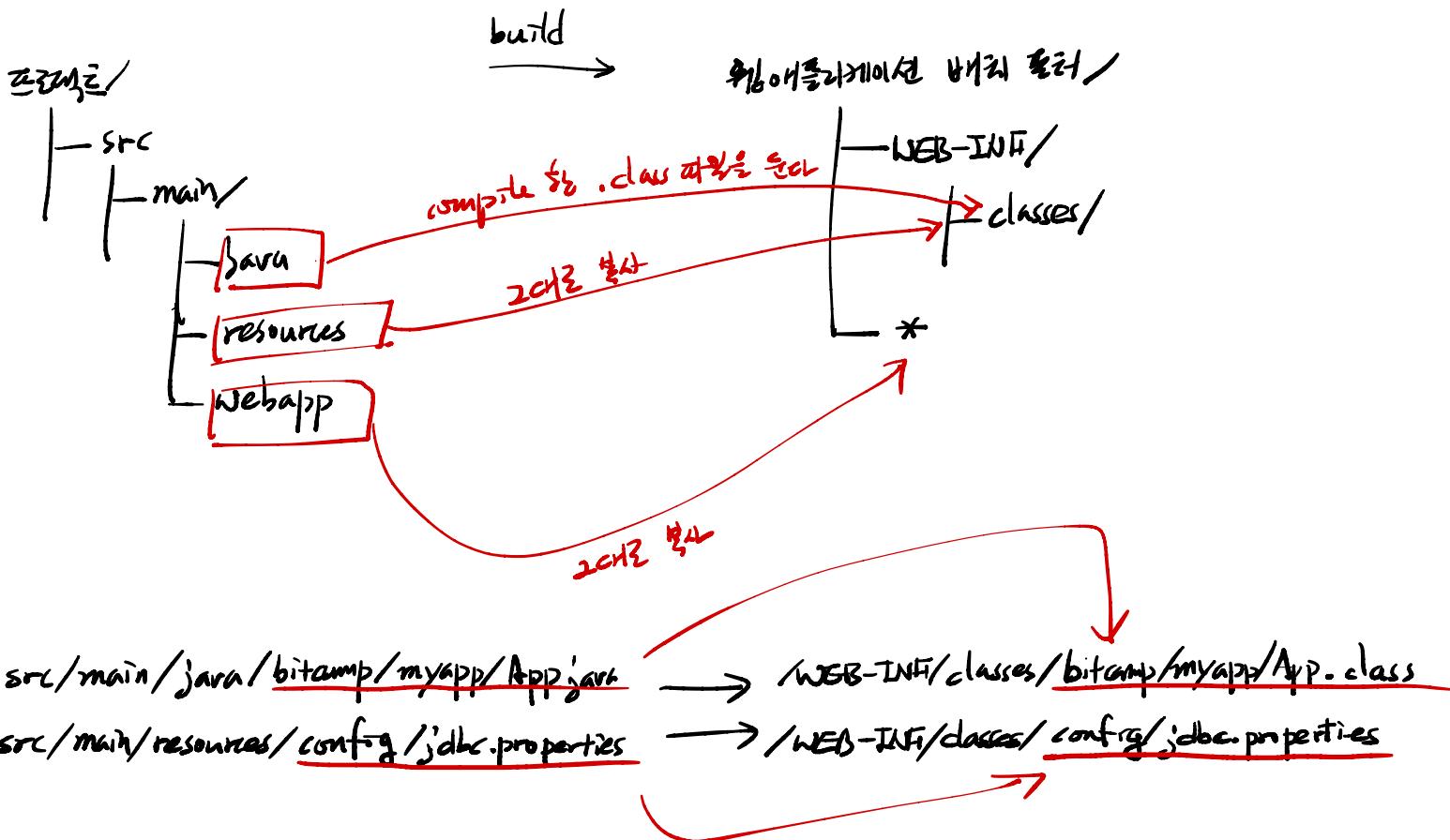
① SQL Mapper



② OR Mapper



* build et classpath



* DaoFactory

↑
생성
DAO

SqISessionTemplate

↑
Mybatis
가져온
DAO

[BoardDao]

insert()

[DaoFactory]

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

[SqISessionTemplate]

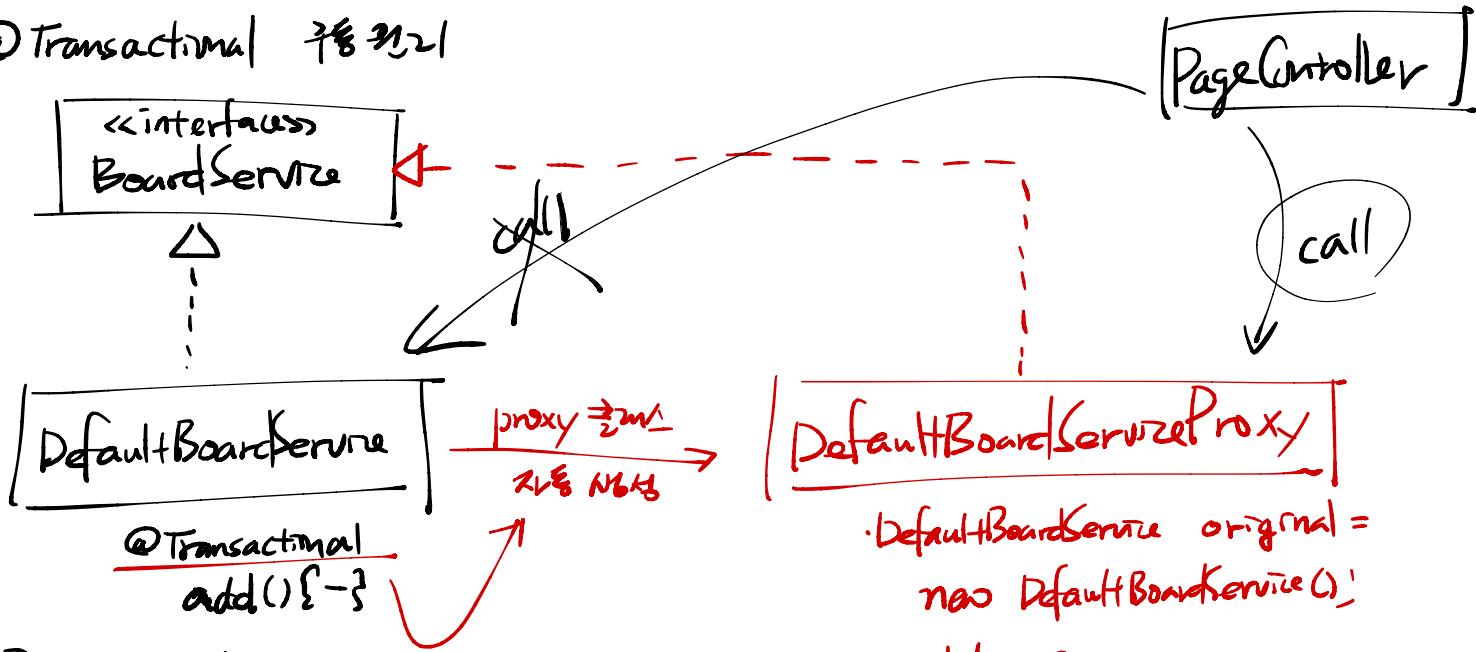
↳ sqISession.insert(

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

SQL
실행
하기위한
SQL문
생성
하기위한
Template

MyBatis
Template

* @Transactional 78 31, 21



. DefaultBoardService original =
new DefaultBoardService();

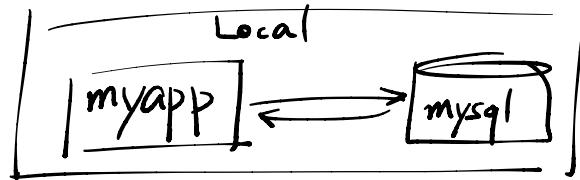
. add () {

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

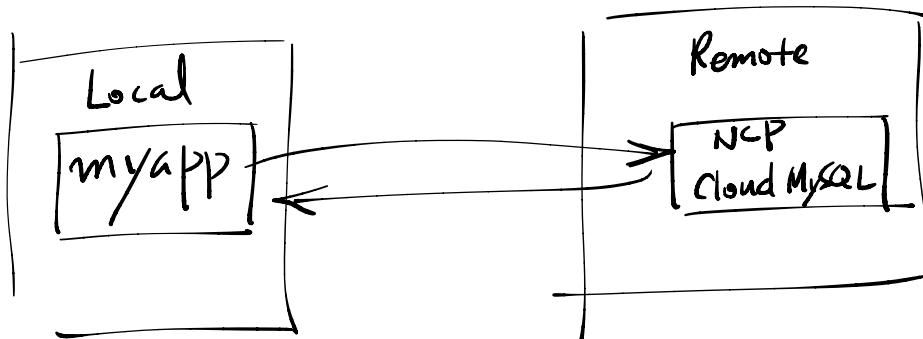
* AOP의 특징, 예제
 ↳ 기존 코드를 대체하지 않고
 추가 기능을 더 넣어가는 방법
 ↳ proxy 구조를 활용하는 방법 ("proxy pattern 사용")

62. Naver Cloud Platform 사용 예

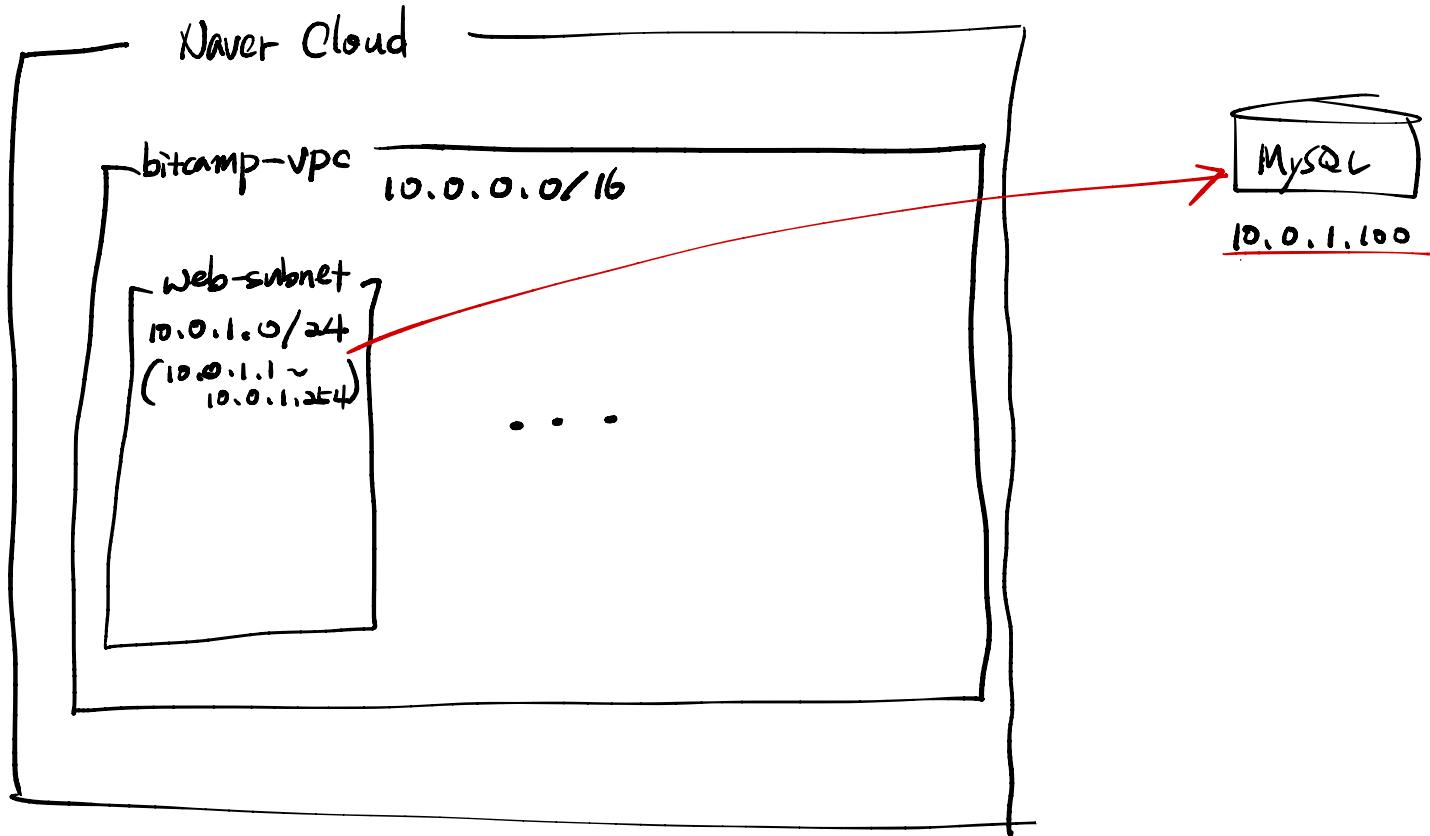
* ORI



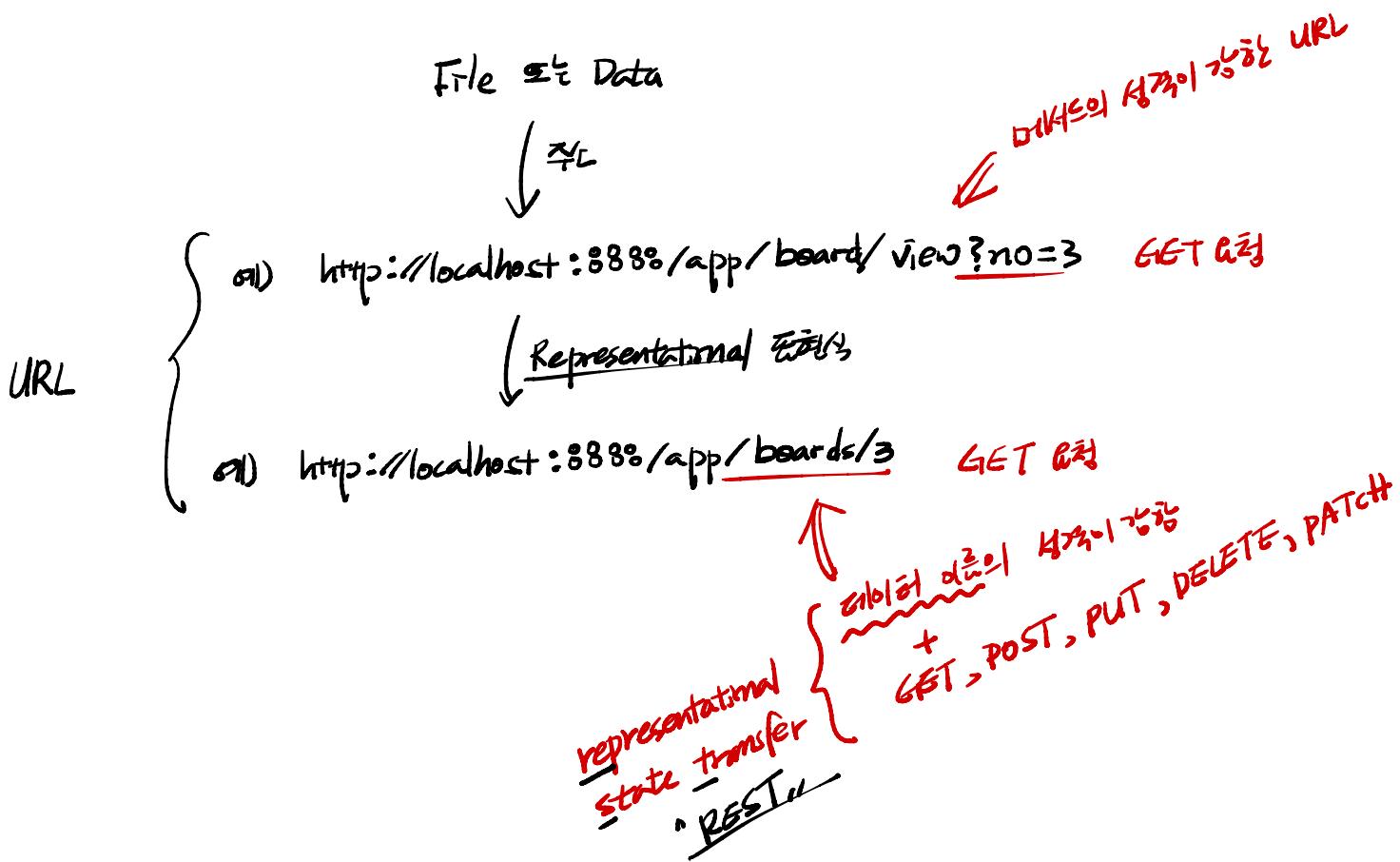
* 대입



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



* RESTful API



* function → REST API

다른 프로토콜

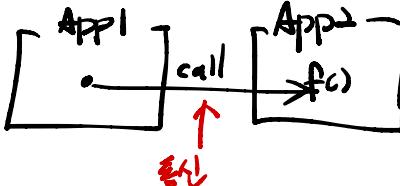
① function

(RPC)

② Remote Procedure Call

(RMI)

③ Remote Method Invocation



C 등 전통적인 프로그래밍

기업의 업무 처리



App의 기능을 분산 (분산 컴퓨팅)



한 App이 하던 일을

여러 App으로 퍼포먼스 향상

App 간의 연동이 필요함



그러나 다른 App의
function을 호출할 수 있는
기능이 등장하게 되었다.



C++ 등 대체적인 프로그래밍



ODP 프로그래밍의 특징이 빠듯

RPC를 개선

* function → REST API

③ RMI → ④ CORBA

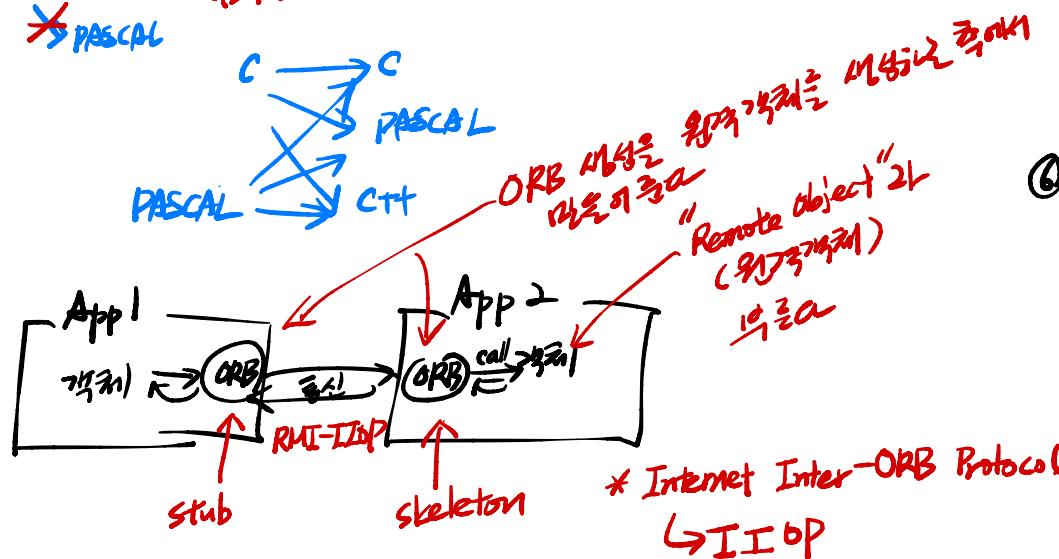
같은 인터프리터
만든 App끼리
함수를 호출하는
방식

C → C
✖ PASCAL

common
Object Request Broker
Architecture

자신의 언어로 만든 App끼리
함수를 호출하는 방식

C → C
✖ PASCAL
✖ C++



⑤ Web-service

✓ 웹으로 HTTP 프로토콜을 활용
✓ ORB를 통과하지 않고 서비스를 제공
→ 단점은?

단점은?
인터넷 규칙에 맞도록 개발하는
작업은 훨씬!

⑥ Enterprise JavaBeans (EJB)

자바언어 기반 EJB 기술

단점!

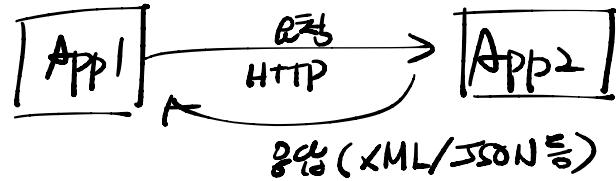
2000년 초반에는 PC에서
제작된 데이터베이스를 써서 EJB로 개발
→ 디자인 단계에서 초기 설계 단계가 오류로

* function → REST API

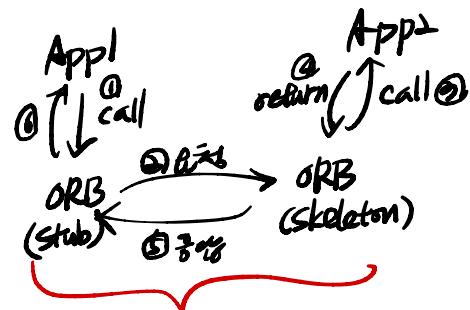
(Web-Service)
EJB

→ REST API

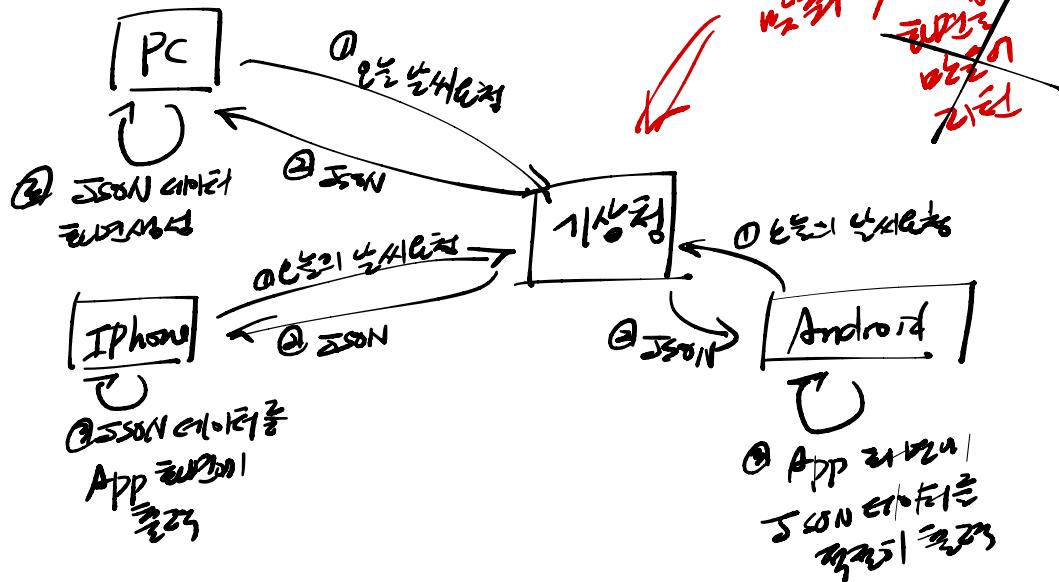
인터넷 원격 제어
ORB는 애플리케이션에



도입 대체로 사용되는
인터넷 원격 제어
API는
ORB는
제거



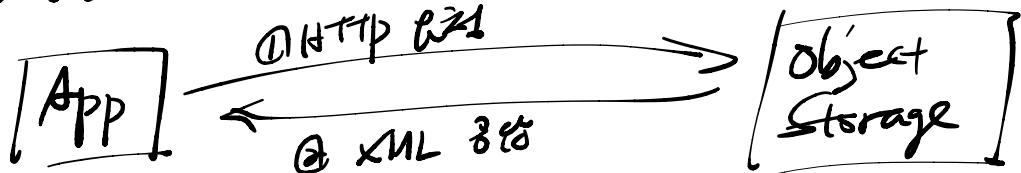
ORB는 통신
인터넷 원격 제어의 대체로



⑦ APP 사용
JSON 형태로
제공되는

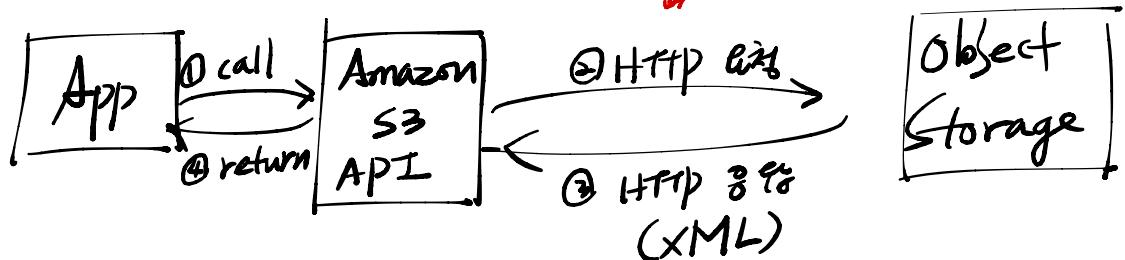
* Object Storage REST API API
↳ HTTP 퀼/값, 키!

① 직접 접근

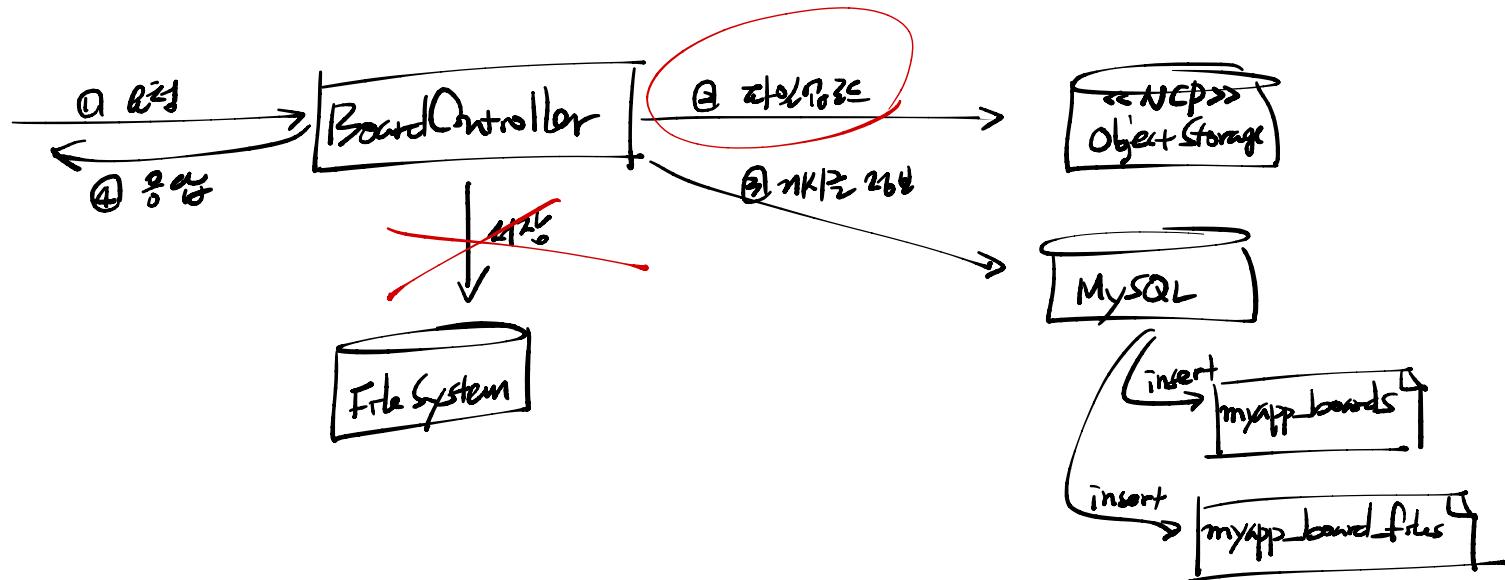


HTTP 퀼/값은 직접 접근하기 편리하다
HTTP 응답을 직접 접근하기 편리하다

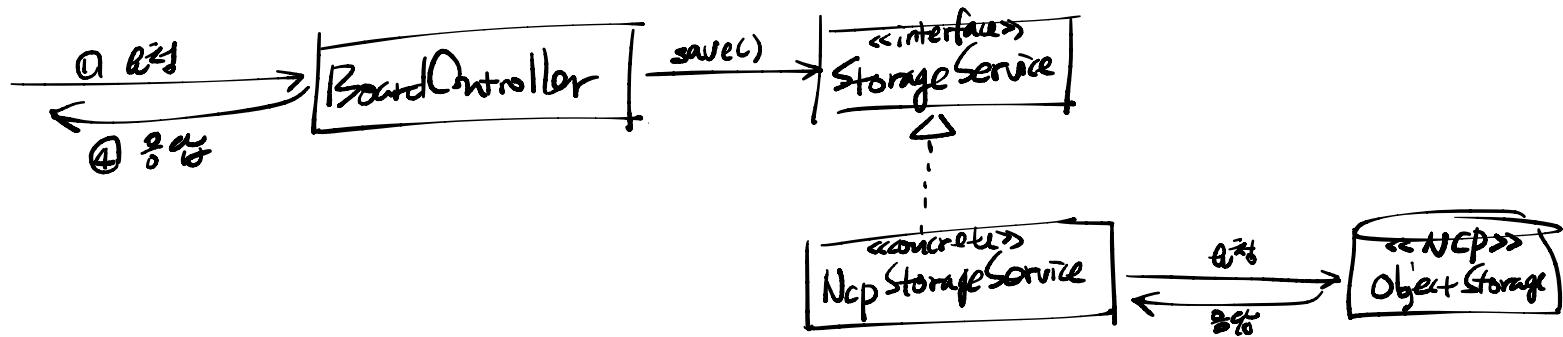
② 경유 애플리케이션



* 내부로 첨부파일은 NCP의 Object Storage에 저장

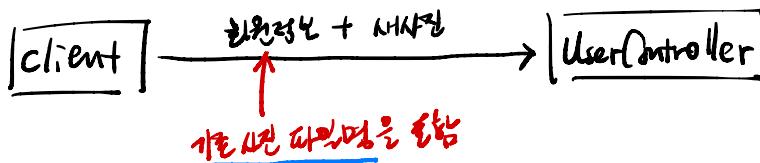


* 게시글 첨부파일은 NCP의 Object Storage에 저장 → NCP 연동로직을 서비스 객체로 분리



* 사용고객과 보안

① 보안에 악용은 예 : 회원정보 변경



A라는 사람이 로그인 한 후에
마술의 정보를 바꿔서 사용자인증을
다른 사람의 것으로 침범할 수 있다.
↳ 다른 사람의 사용자 인증을 흡수할 수 있다.



이제, 변경, 삭제 시

기존 정보를 사용할 때는

항상 새롭게 쓰레기로 사용하자!

기존 개인정보를 클라이언트에서 뺄지 알자! → 꼭 새롭게 쓰레기로
기존 정보를 간수하자!

- i) 기존 사용자인증 → 클라이언트가 보낸 사용자인증으로
사용자 정보를 흡수 처리
- ii) NH 네이버 계정 → 사용자 계정
- iii) 카카오 계정 → DB 변경

* Transaction Propagation

<u>Caller</u>	Transaction	
<u>Propagation</u>	X	O (Tx1)
(default) REQUIRED	Tx1	Tx1
REQUIRES_NEW	Tx1	Tx2
MANDATORY	예외	Tx1
SUPPORTS	선택적 투명성 none	Tx1
NOT_SUPPORTED	불행	현재 트랜잭션 외부의 상황을 적용
NEVER	실행	예외

* propagation \rightarrow ~~to~~ or:

① REQUIRED

UserController.delete() X
↓ call
Tx1 { DefaultUserService.delete()

Tx1 { UserController.delete()
↓ call
DefaultUserService.delete()

② REQUIRES-NEW

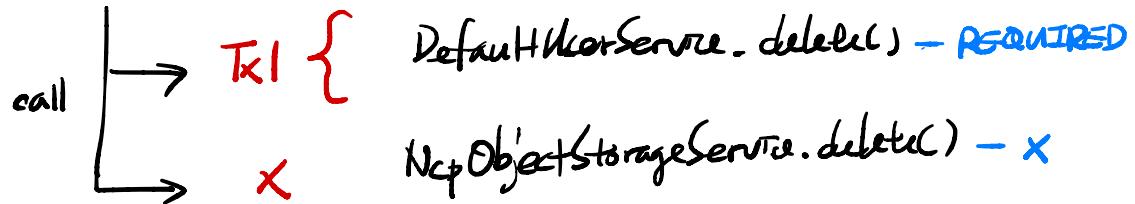
UserController.delete() X
↓ call
Tx1 { DefaultUserService.delete()

Tx1 { UserController.delete()
↓ call
Tx2 { DefaultUserService.delete()

* 키워드 어노테이션 정리

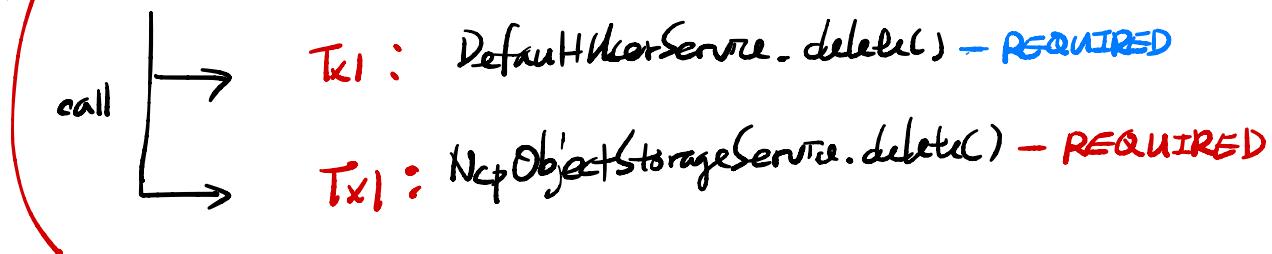
① 허용

X UserController.delete() X

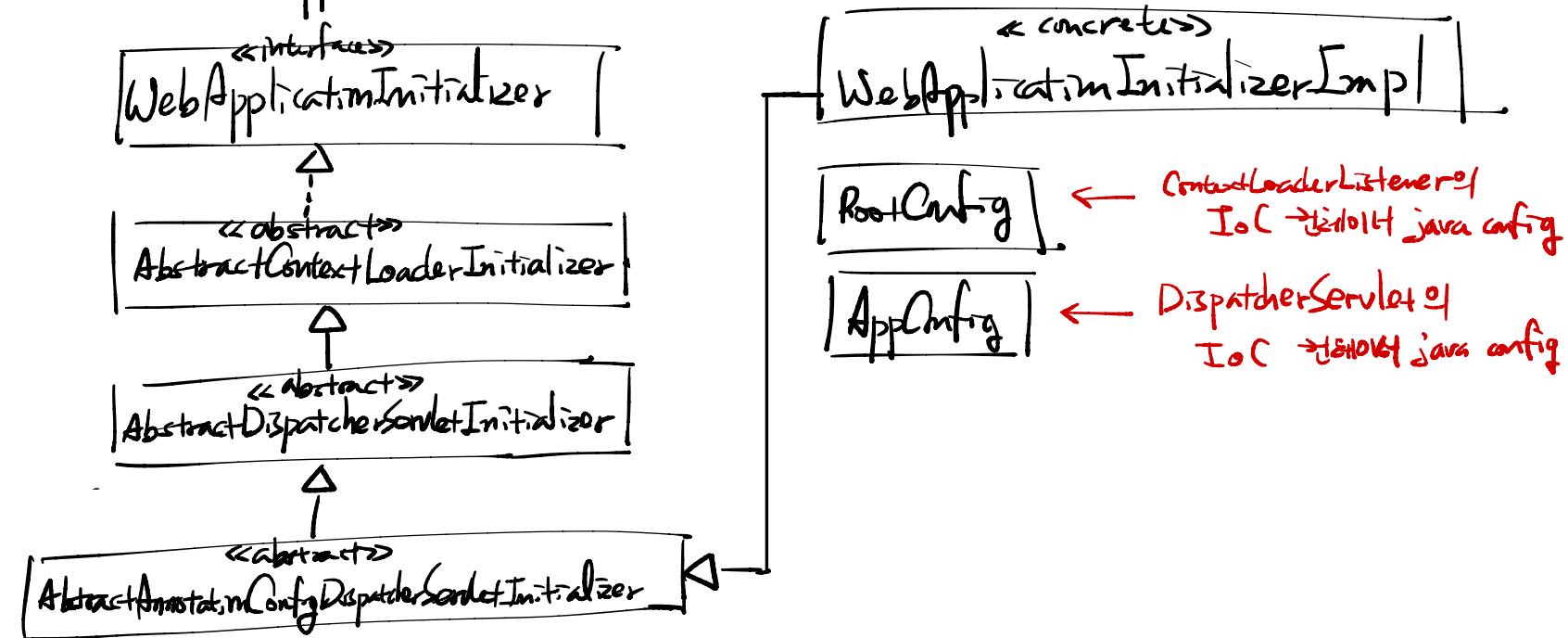


② 허용

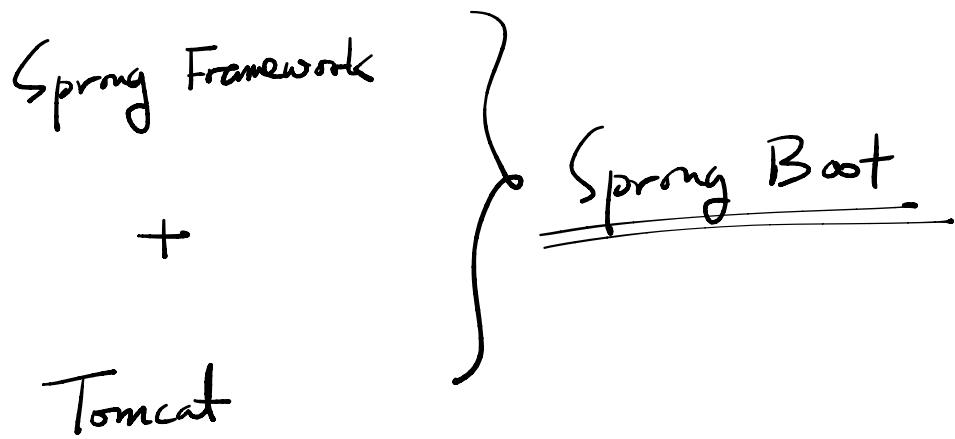
Tx1 UserController.delete() - REQUIRED



63. WebApplicationInitializer



65. Spring Boot mpls



66. $\text{SELECT}_{[012]} \text{ FROM}_{[1]}$

$$\text{SELECT}_{[012]} \text{ FROM}_{[1]} \text{ WHERE} = \frac{(\text{012}(\text{011012})^4 \times 215)}{= \text{0121010101010101}}$$

select
from
where
order by

limit 3개, 215

limit 0, 4

limit 4, 4

limit 8, 4

limit 12, 4

limit 16, 4

- | | | |
|----|-------|---------------------|
| 0 | _____ |) 1 215012 |
| 1 | _____ | |
| 2 | _____ | |
| 3 | _____ | |
| 4 | _____ |) 2 215012 |
| 5 | _____ | |
| 6 | _____ | |
| 7 | _____ | |
| 8 | _____ |) 3 215012 |
| 9 | _____ | |
| 10 | _____ | |
| 11 | _____ | |
| 12 | _____ |) 4 215012 |
| 13 | _____ | |
| 14 | _____ | |
| 15 | _____ |) 5 215012 |
| 16 | _____ | |
| 17 | _____ | |

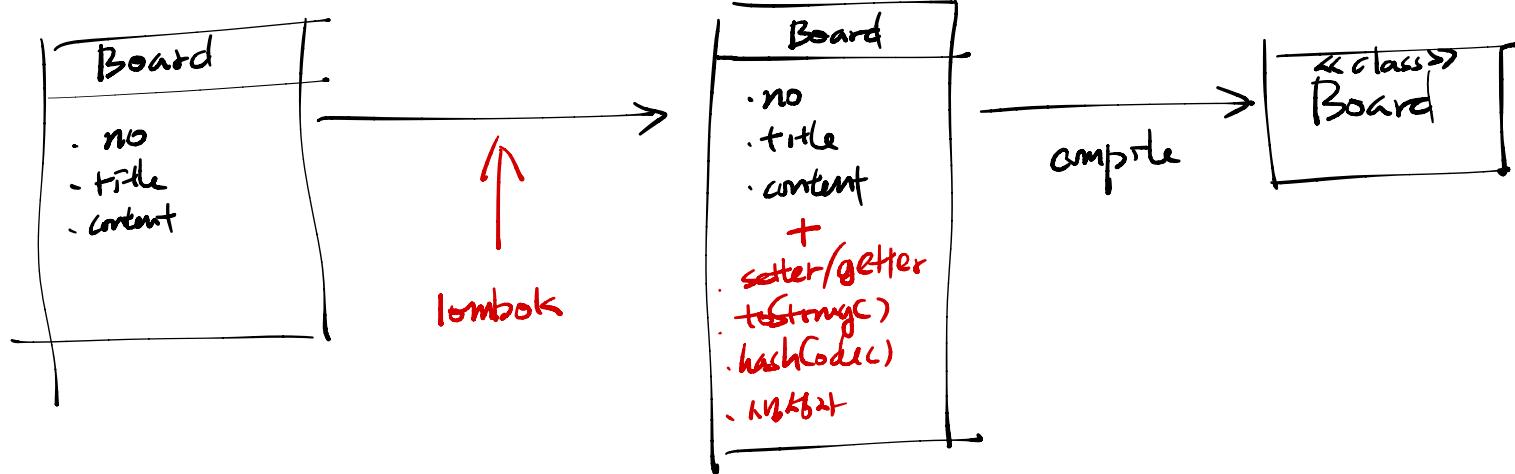
67. Lombok 2010-2021年版

↳ Domain 实体 には 何が付く → setter/getter, toString(), hashCode(), equals()
何が付かない なに?

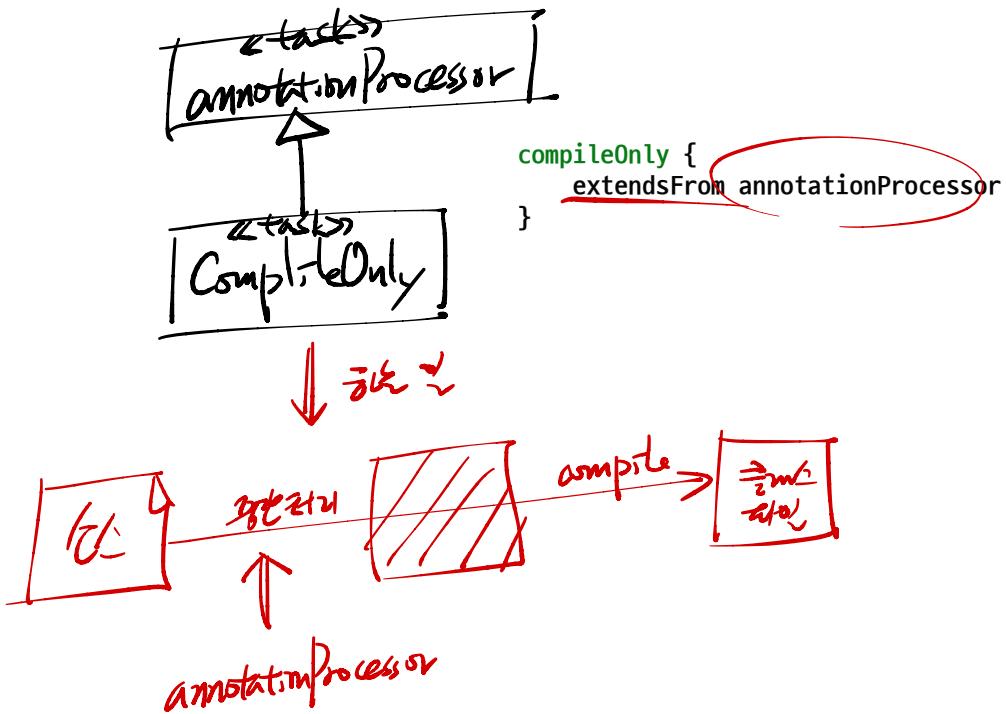
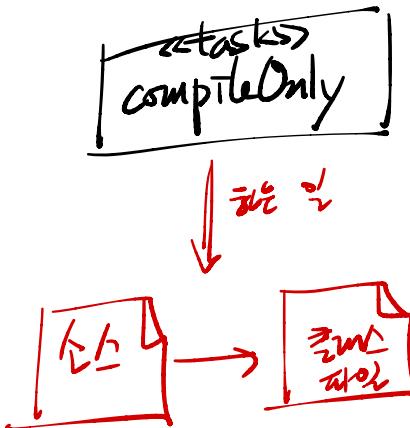
Gradle



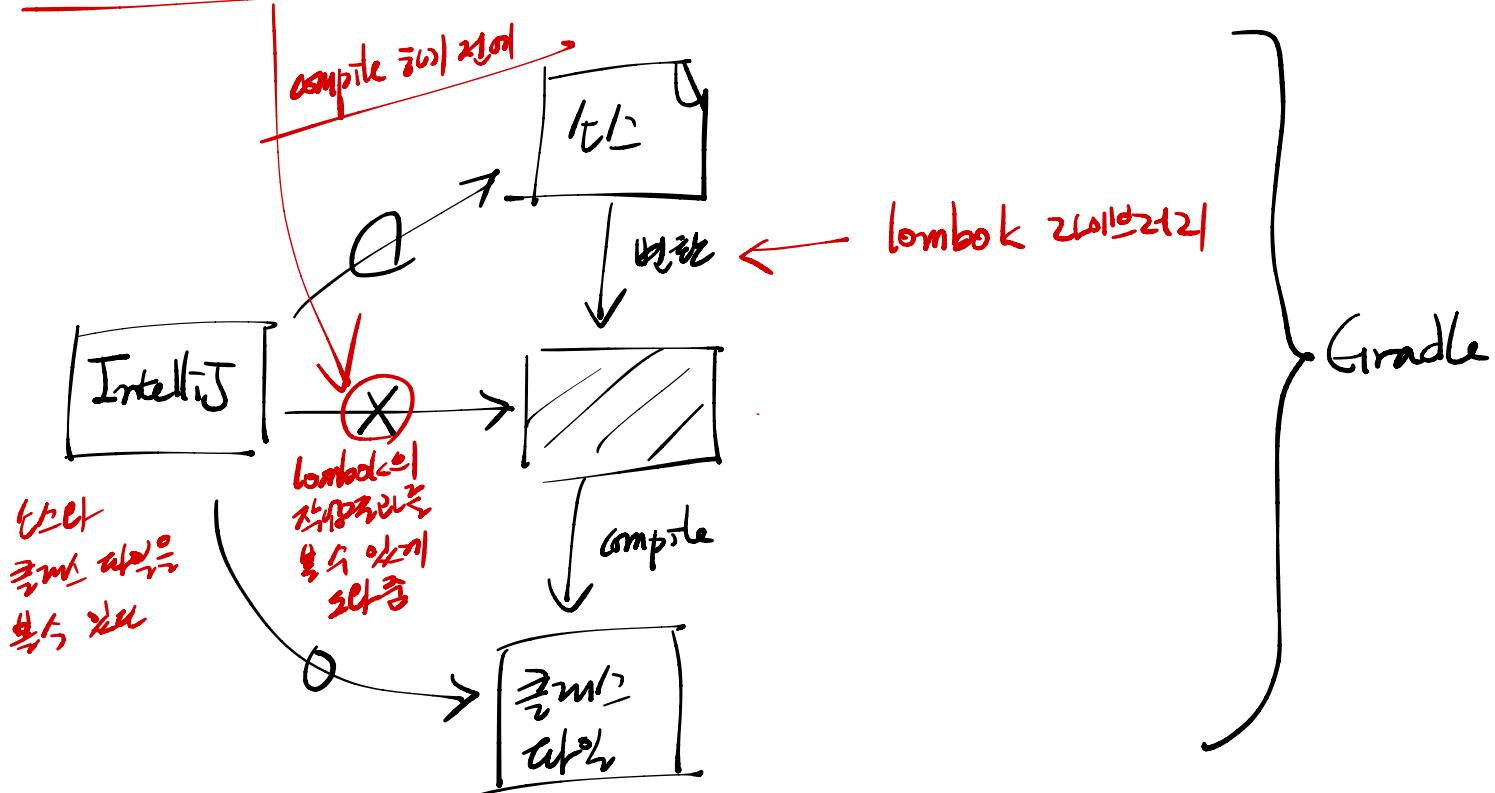
Build



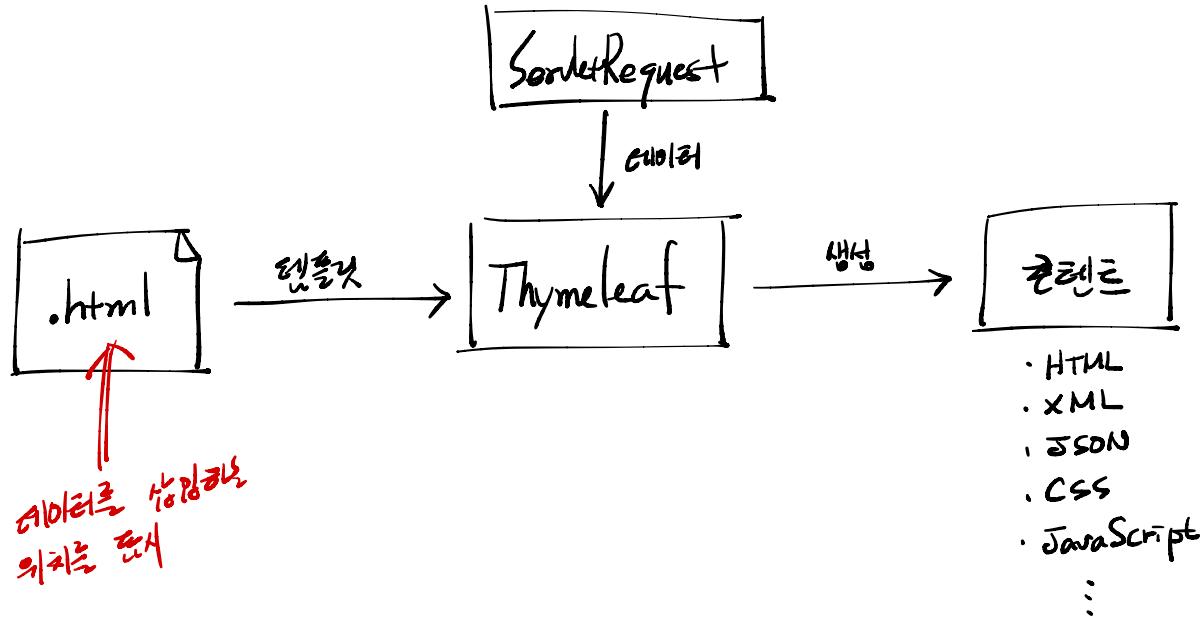
* Gradle et Task



* lombok IntelliJ 풀이방법의 차이



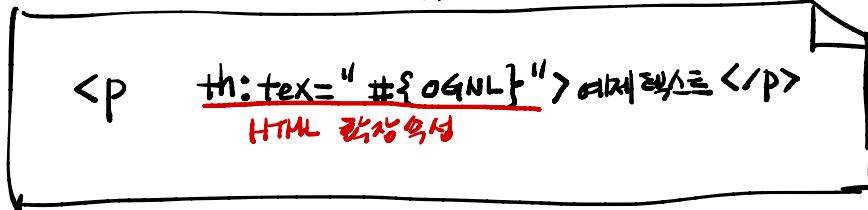
* Thymeleaf 템플릿 인자



* Thymeleaf 템플릿과 HTML

.html

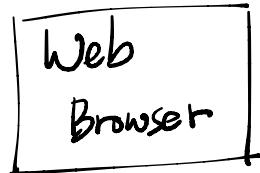
템플릿 파일



JSP → \${변수명}

Mybatis → #{} выражение

Thymeleaf → \${변수명}



HTML 편집과 편집을 허용



디자이너

디자이너는 Thymeleaf가
수동화한 코드에 맞게 편집
하고 편집을 확인할 수 있다

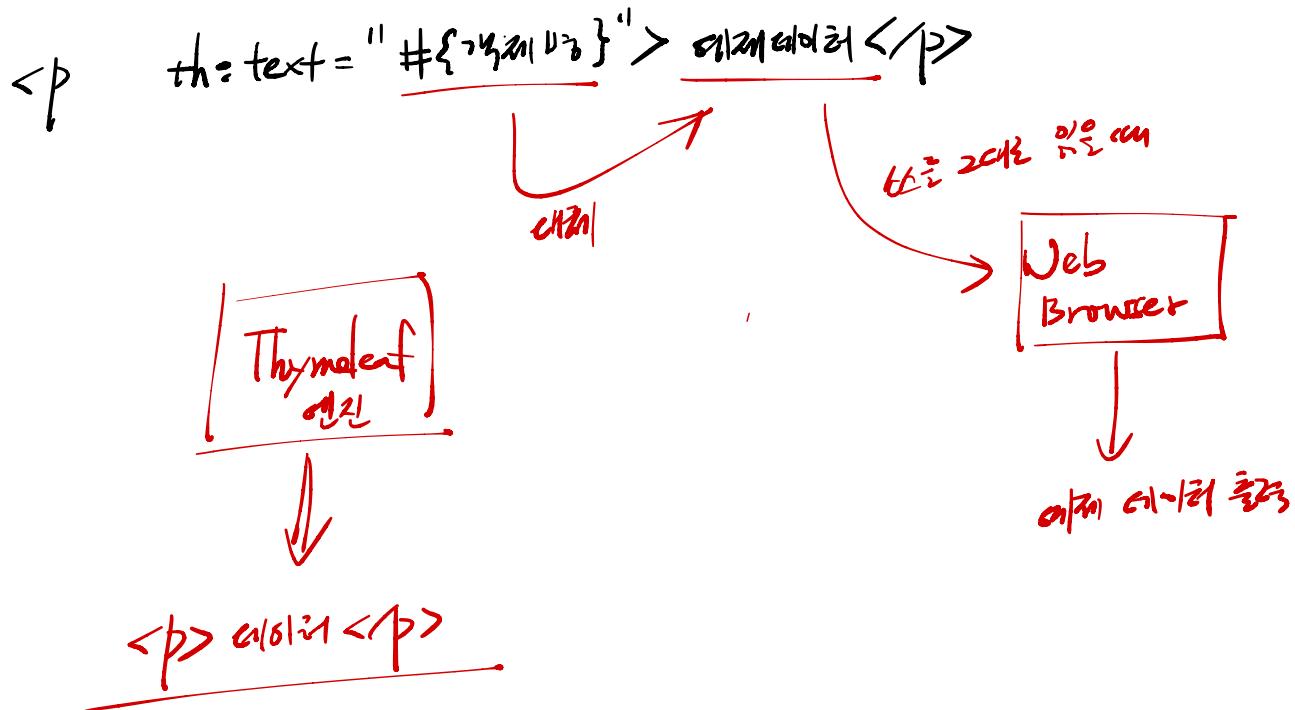
Thymeleaf의 특징

- ① JSP와 다르게 템플릿이 HTML 파일이다.

↳ 템플릿 언어의 사용성이
스스로로 웹브라우저에서
표기되는 대목으로 가능하다.

- ② Thymeleaf 파일이 HTML 편집을 허용하기 때문이다.
스스로로 편집할 때 편집이 되지 않는다.

* Thymeleaf 템플릿 HTML



* HTML의 속성과 属性

<table> th: text = "국민당" <td> </td>

↑
국민당
국민당



<td> </td>

* Thymeleaf Namespace

namespace (прост)

```
<html xmlns:th="http://www.thymeleaf.org">
```

Thymeleaf 엔진이 Thymeleaf 엔진이

th 네임스페이스로 시작하는 것은 사용된다

```
<p th:text="${welcome}">환영합니다!</p>
```

Thymeleaf 엔진은

HTML 템플릿

th:* 대화 속성을 만들기

Thymeleaf 템플릿 속성으로 인식되며
처리되는 것이다

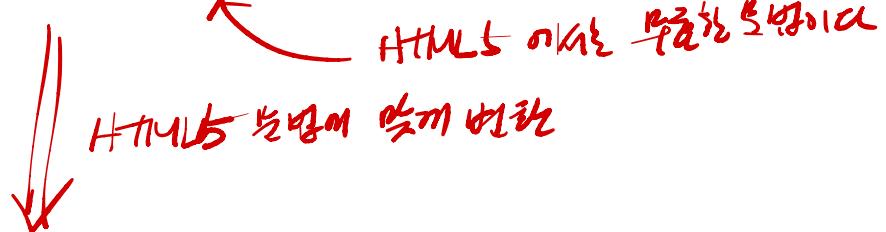
※ XML namespace 처리는
HTML 처리는

th: 를 기본으로 인식한다

그것은 가능할 생각하지 않아!

* HTML5 et Thymeleaf

```
<p th:text="${welcome}"> 안녕 </p>
```



```
<p data-th-text="${welcome}"> 안녕 </p>
```

HTML5 속성이
여러개인 경우

* HTML5의 템플릿 속성

data-th-text = " \${id}"

"data-th-text"

* th-text vs th-utext

welcome = <u>안녕</u> 친구님.

data-th-text

escape 문자를 차단한다

HTML문은 링크처럼 사용하는
HTML 키워드에 해당되는 문자를
보관하여 HTML 키워드가 영향을
끼치지 않게 하는 기능이다.

< b > 안녕<u> / b > 친구님.

↑

HTML Entity

(HTML 특수기)

↳ HTML keyword와 종종나는 문자를
표현하기 위한 특수기로 사용된다.

} HTML 키워드가 아닌
단순 문자로 취급하기
위함.

data-th-utext

unescape

escape 문자를
차단하지
않는다.

HTML 키워드가
포함된다.

< b > 안녕<u> / b > 친구님.

단일 텍스트로
HTML 특수기로
변경된다.

* Link

`data-th-href = "@{url}"`

``

Actual URL: http://localhost:8888/app/board/list

`@{http://localhost:8888/app/board/view}` → `href = "http://localhost:8888/app/board/list"`
자바 경로

`@{~board/view}` → `href = "/board/view"`
위치 root path 가는

`@{/board/view}` → `href = "/app/board/view"`
context path 가는

`@{board/view}` → `href = "board/view"`
같은 위치로 가는

server root path ⇒ /
context path ⇒ /app
(위치와 동일한 경로)

69. Handler & Method ArgumentResolver چی가 뭔가?

request handler

파라미터 ← 사용자에게 주는 값들

String add(
 Board board,
 MultipartFile[] files,
 HttpSession session) {

====

}

어디로?

request handler of
attribution 헤더
method

1/2-3
정보수집
attribution 헤더

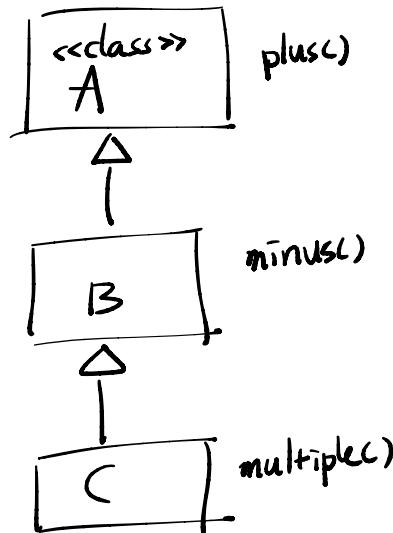
} {
 SendRequest
 SendResponse
 HttpSession
 @RequestParam

* 320! 사용자의 접근을 허용하는 HandlerMethodArgumentResolver 데코더

```
public String add(  
    Board board,  
    MultipartFile[] files,  
    @LoginUser User loginUser  
) throws Exception {  
}  
...  
}
```

- ① LoginUser 어노테이션 탐지
② LoginUserArgumentResolver 을 통해 정의

* Class . isAssignableFrom()



A.class.isAssignableFrom(B.class)

A type의 객체가 B type의 인스턴스를 할당할 수 있나?

A obj = new B();
obj . plus(); A + &

B obj ~~=~~ new A();

obj . minus();
obj . plus();