

57. 파일 업로드

* 텍스트 프로토콜

POST /board/add HTTP/1.1 ↗ request line
Content-Type: multipart/form-data; boundary=----Webxxxxx
↑
<form enctype="multipart/form-data" ...>
:
<input type='file' ...>
:
</form>

자바에서 구동 문자열

header

↑
→ ← 파일-문자열 <boundary>
→ ← 파일
→ ← 파일-문자열
:
} message-body ~~O=OK O=O~~

* 요청 파싱 \Rightarrow parseRequest()

aaaa
name = "title"

bbbbb
name = "content"

dd
name = "files"

ee
name = "category"

--

new

isFormField() = true
FileItem

isFormField() => true
FileItem

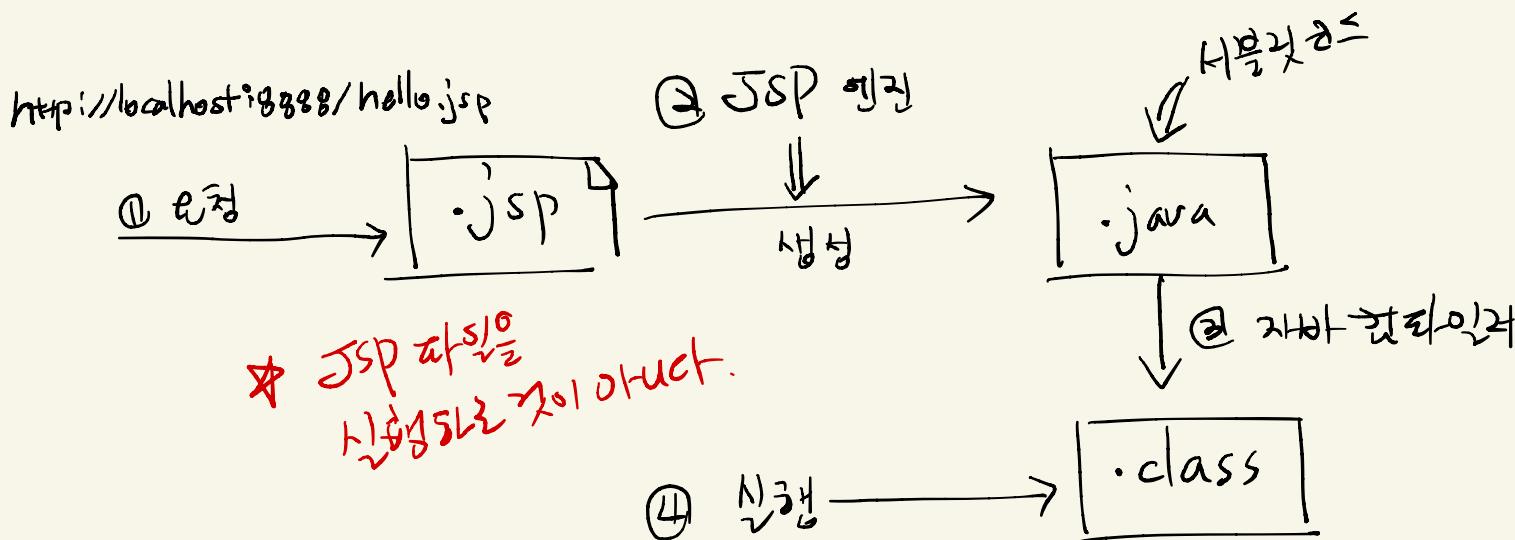
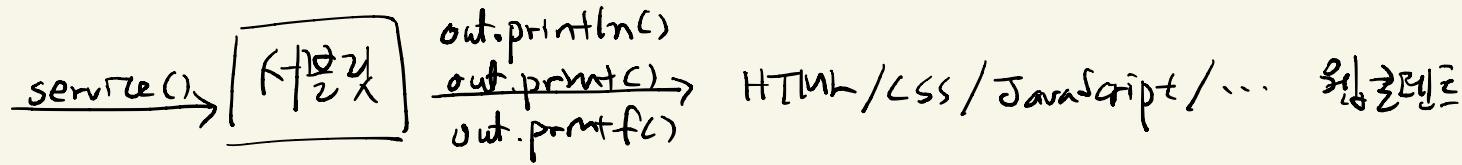
isFormField() => false
FileItem

FileItem

List

getFieldName()
getString()
getFieldName()
getString()
getFieldName()
getString()
~~getString()~~

* JSP → 서블릿 코드 자동생성



* JSP 자원 평로

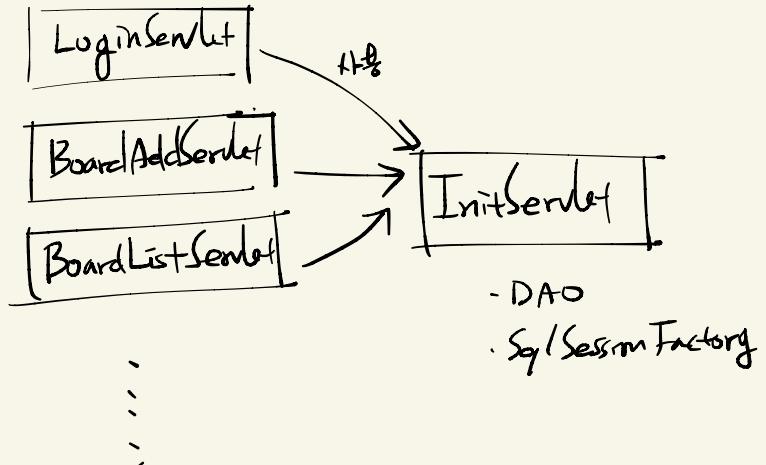
src/main/webapp/*.jsp

↓ 자동생성

.. /temp/*:java

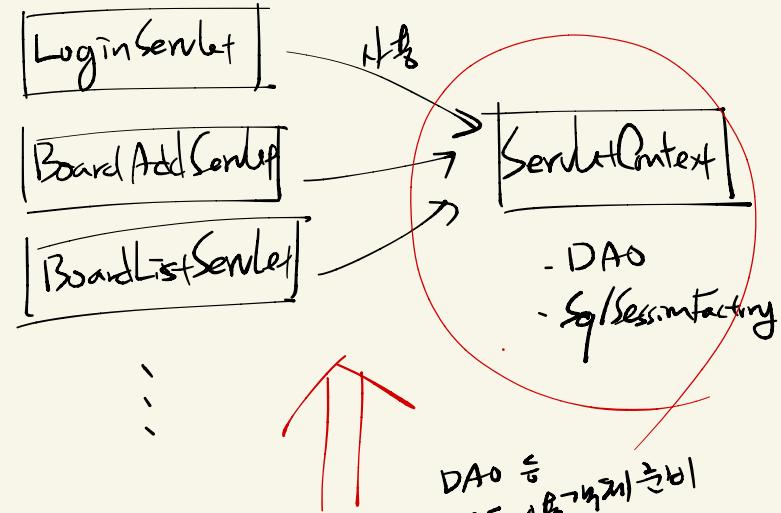
* 63. ⇒ CTX ServletContext 보관/수

① 전통



* 예전에
- ex) static DAO를
InitServlet에 주면
그거 쓰는거.

→ ② InitServlet 종속성 주입

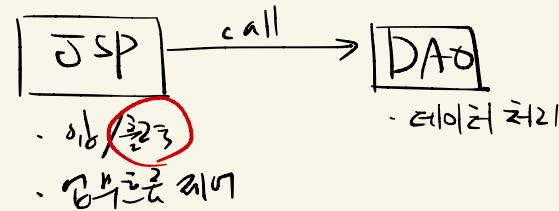
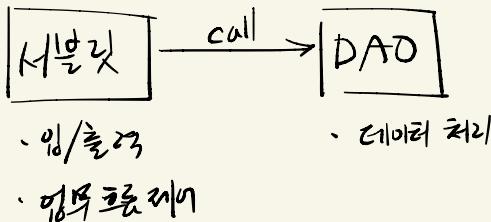


ContextLoaderListener

<interface> ServletContextListener

* 64. MVC 모델

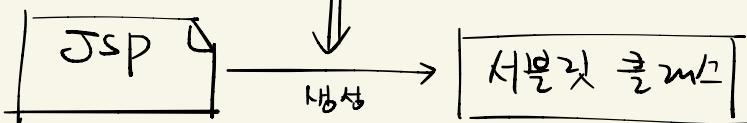
① 이전 방식



* JSP

서블릿 풀러스
Generator
||

JSP Engine



- 템플릿 데이터 → 출력문
- JSP Action tag → 처리코드
- script tag element → 처리코드
- :

} ⇒ 실행되어야 하는
개발 부담을 줄임!

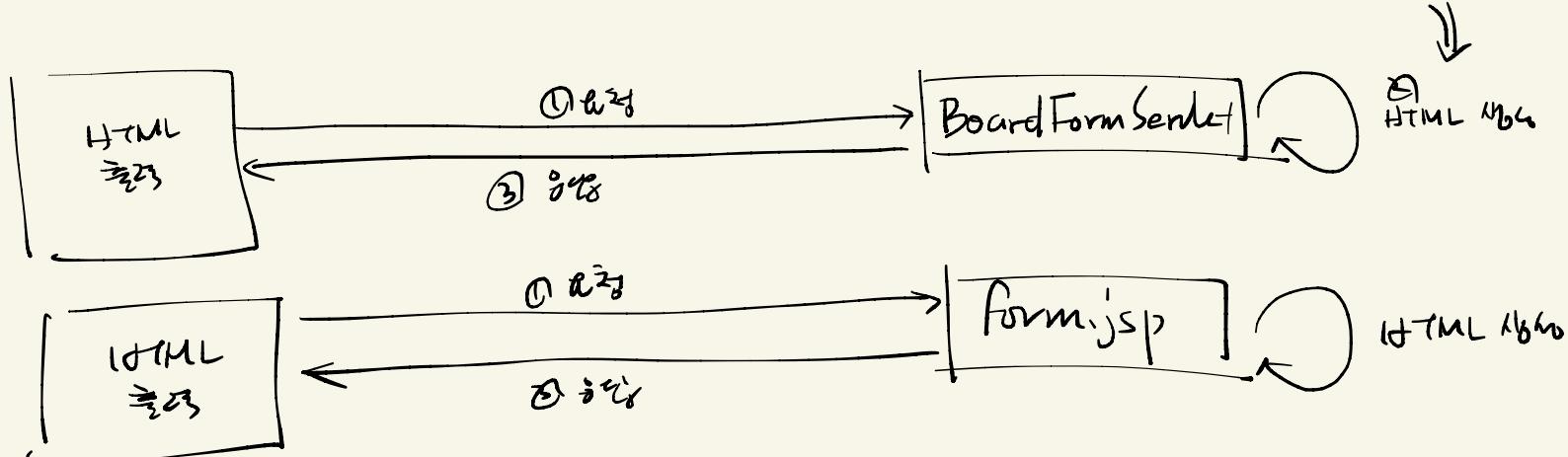
* 이런 방식으로 차이점

- JSP 기술 사용해서 코드를 훨씬 더 간단하게

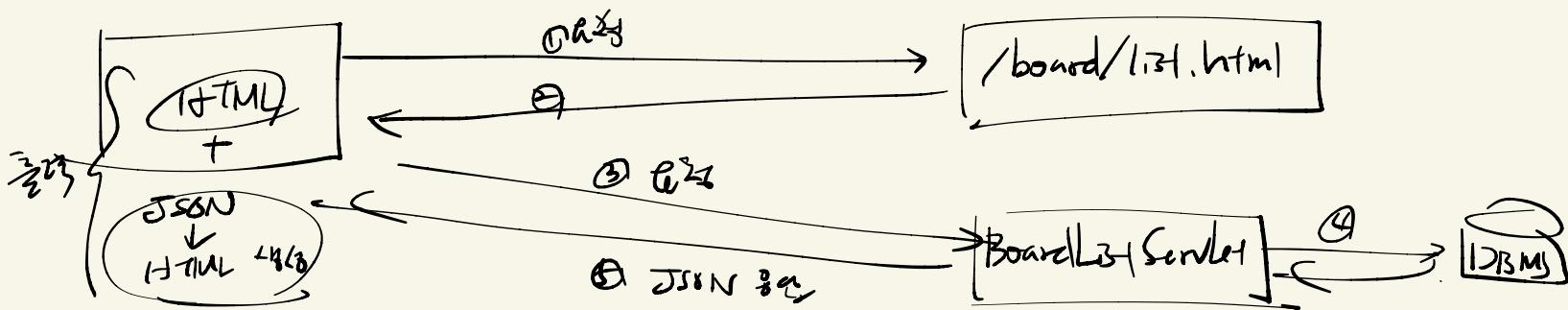
↑
제작 시간
줄여 놓다.

서버 렌더링 (rendering) 과정

서버 렌더링
HTML 흐름



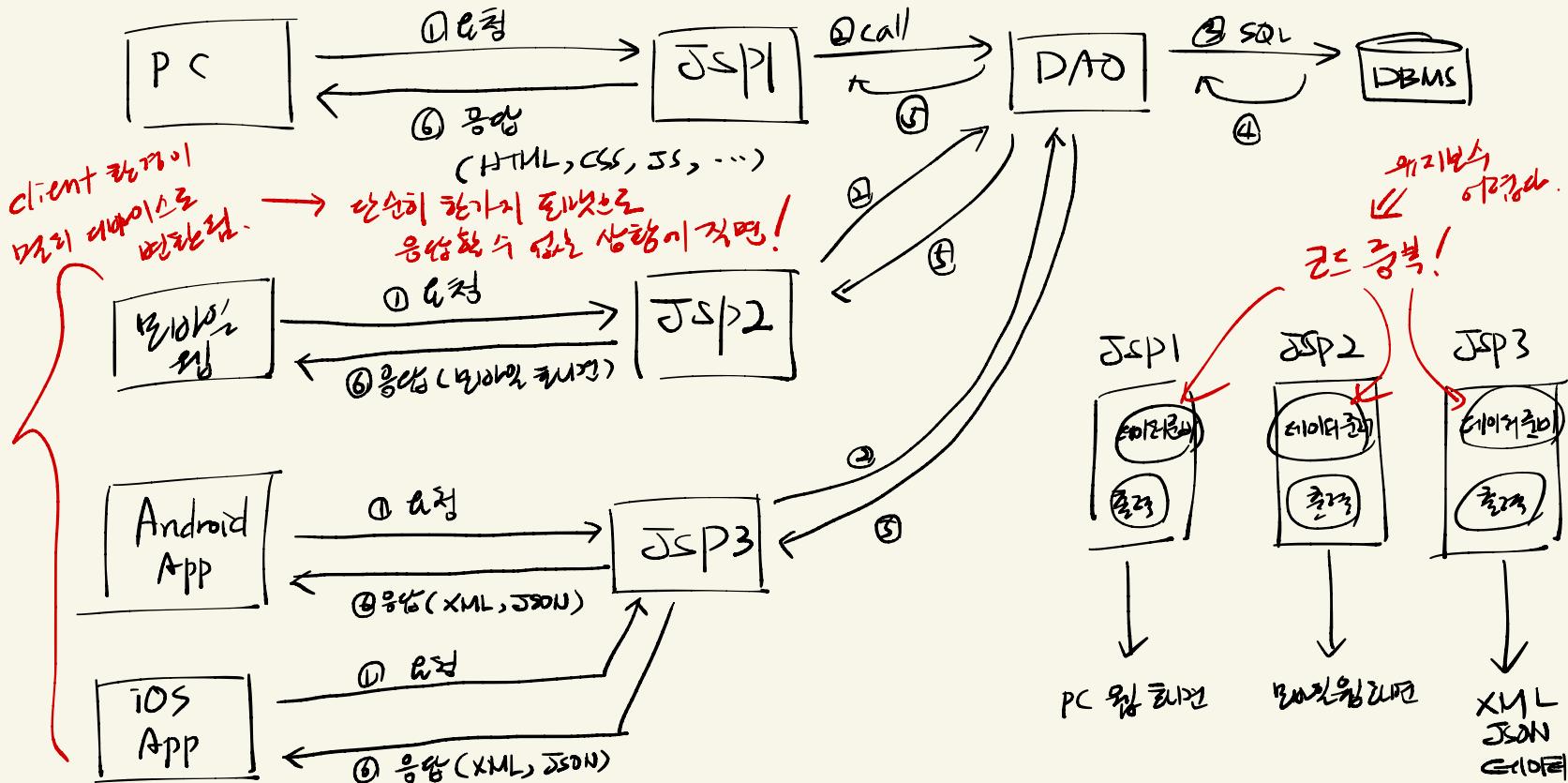
클라이언트 렌더링 과정



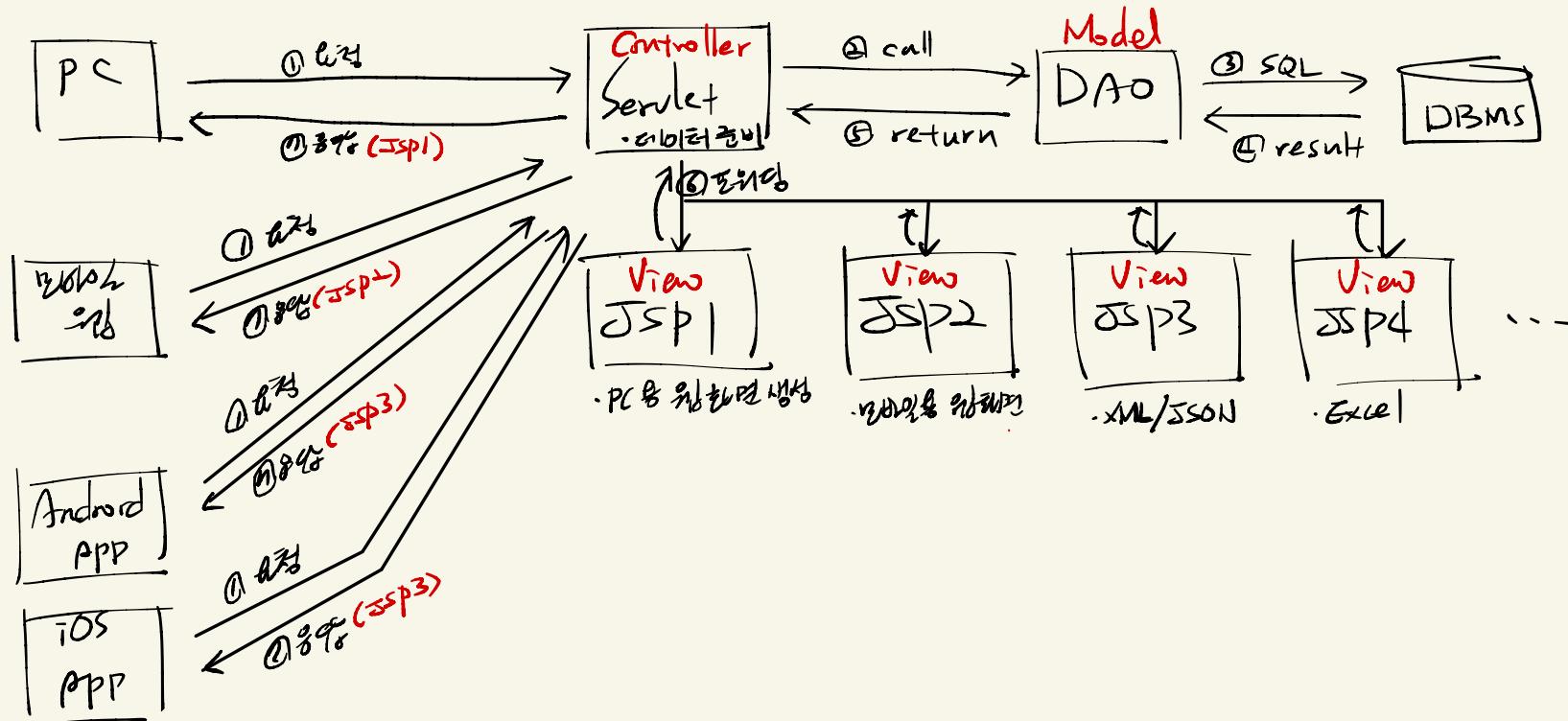
65. MVC 구조 2

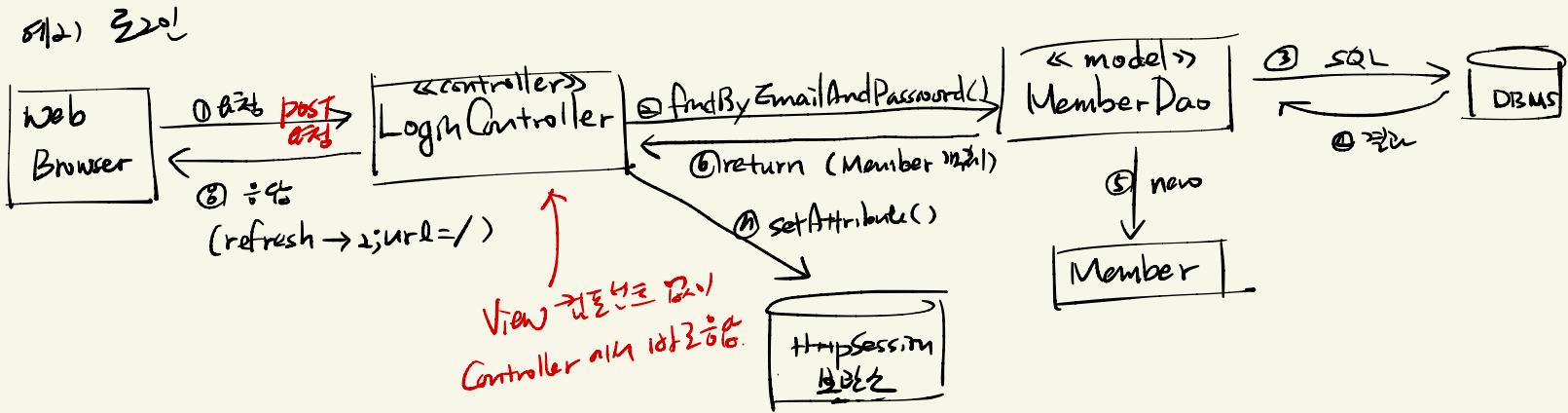
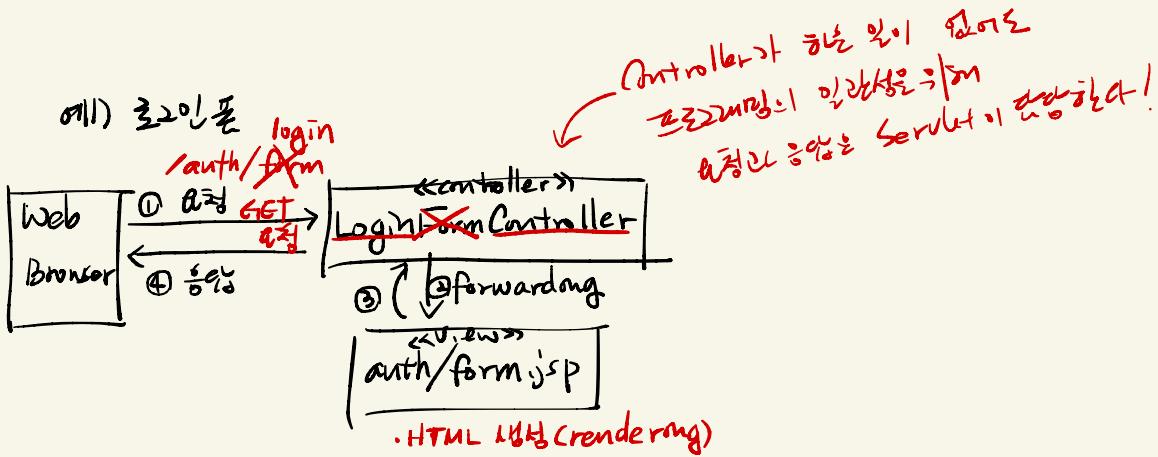
① 이전 방식

②) 개시는 목적
↓



② 인터랙션 흐름





(013) 회원 목록 조회

