

---

---

---

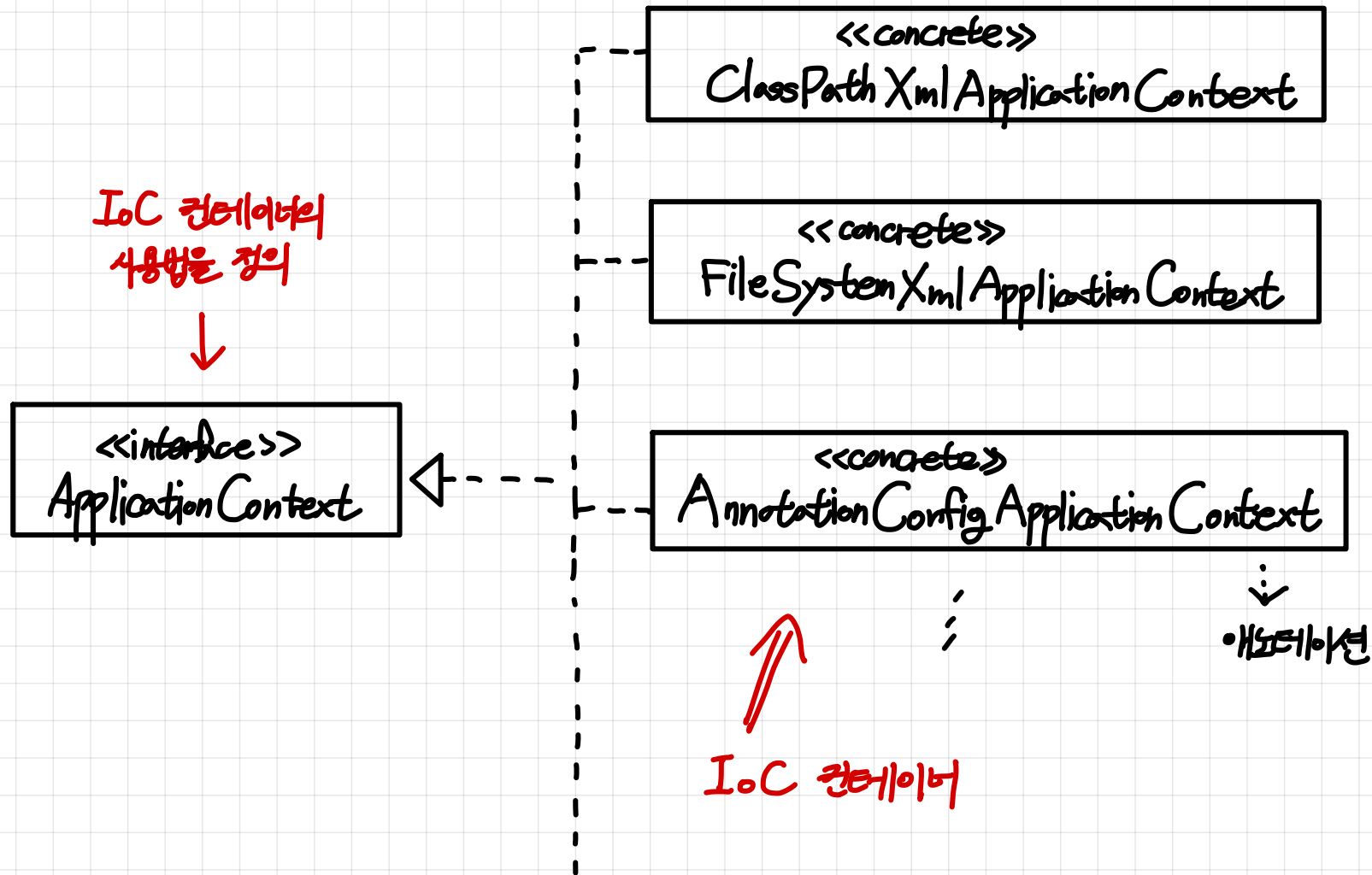
---

---



2021.11.16.라 (95 일화)

## \* Spring IoC 준비



2021. 11. 16. 라 (95 일차)

\* 자바 클래스패스

\* java -classpath bin/main; c:\a\b; xx.jar; c:\a\y.jar 클래스명

classpath

→ JVM은 클래스 파일을 찾을 때 다음 순서로 찾는다.

① Bootstrap ClassLoader → \$JAVA\_HOME / jre / lib / rt.jar

↳ Java 에서 제공하는 라이브러리에서 먼저 찾는다.

②

2021.11.16. 화 (95 일차)

## \* Spring beans 태그

<?xml version = "1.0" encoding = "utf-8"?>

<beans xmlns = "http://\_\_\_\_"

xmlns:xsi = "\_\_\_\_"

xsi: schemaLocation = "이름 URL 이름 URL ... ">

↑  
별명

↑  
(문서 이름)

||  
네임스페이스명

↑  
XML 태그 규칙을 정의한  
문서의 위치

2021.11.16.라 (95 일차)

\* 자바 패키지와 XML 네임스페이스

java.net.Socket

패키지명 ↗ ↖ 클래스명

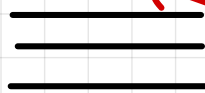
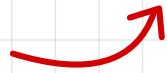
||  
네임스페이스

자바용어 → 태그가 소속된 패키지



C#용어 → 네임스페이스  
별명

< c:forEach items = "\_\_\_\_" var = "\_\_\_\_" >



태그명 ↖

< /c:forEach >

2021.11.16.라 (95 일차)

\* 네임스페이스 vs XML Schema 파일 vs 별명

XML Schema 파일의 URL → http://www.springframework.org/schema/beans/spring-beans.xsd  
↳ 네임스페이스에 소속된  
사용규칙을 정의한다.

네임스페이스의 이름 → http://www.springframework.org/schema/beans

네임스페이스를 사용하기 쉽게 → xmlns:core = "http://www.springframework.org/schema/beans"  
별명을 부여하는 문법

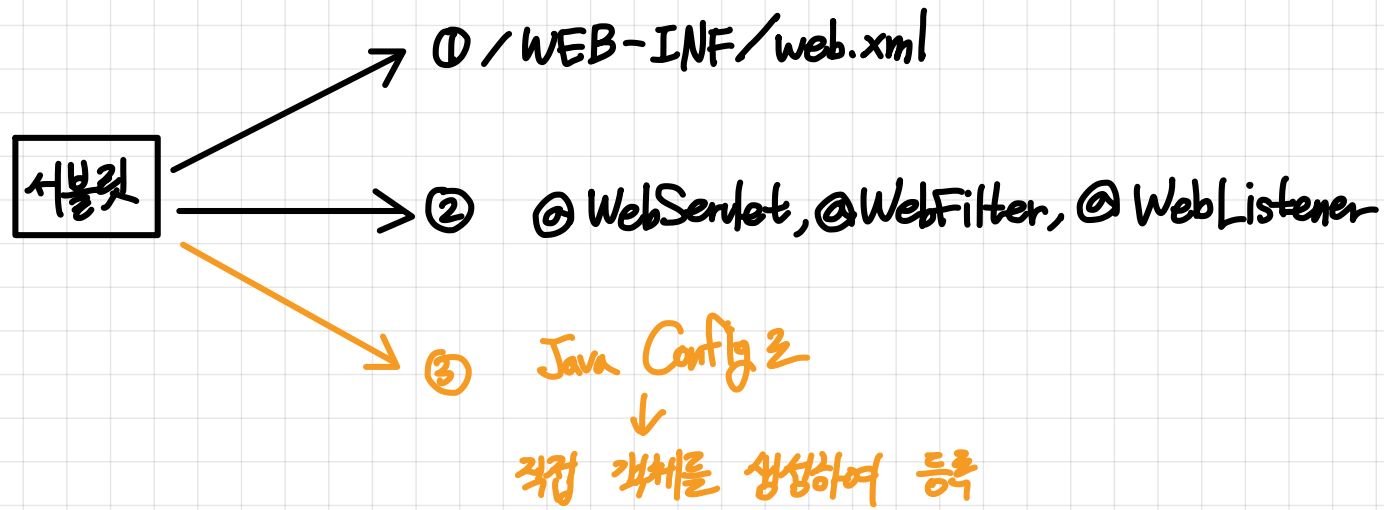
↑  
네임스페이스 별명: 네임스페이스명이 너무 길고,  
태그이름에 사용할 수 많기 때문에  
별명을 애선 사용하여 설명한다.

↓  
네임스페이스명

2021.11.18. 목 ( 일차 )

\* 서블릿을 배치하는 방법

deploy → 서블릿 컨테이너가 찾을 수 있도록 등록하는 것.



2021.11.18. 목 ( 일차 )

## \* Spring WebMVC

①

프론트 컨트롤러 등록



( web.xml  
java.config

②

패러미터 컨트롤러 작성법

③

파라미터 다루는법



파일 업로드 다루는법

④

리턴콘텐츠 다루는법

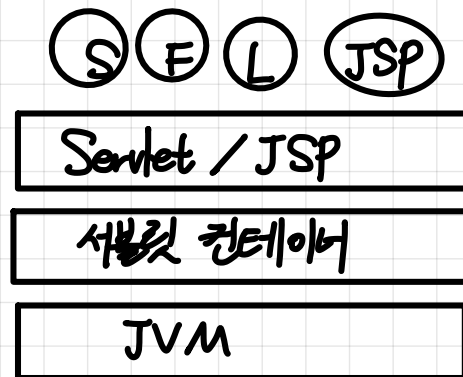
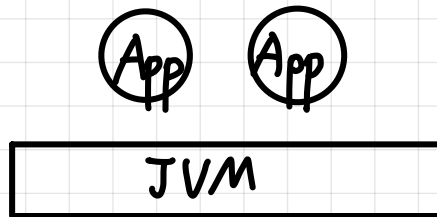


JSON 다루는법

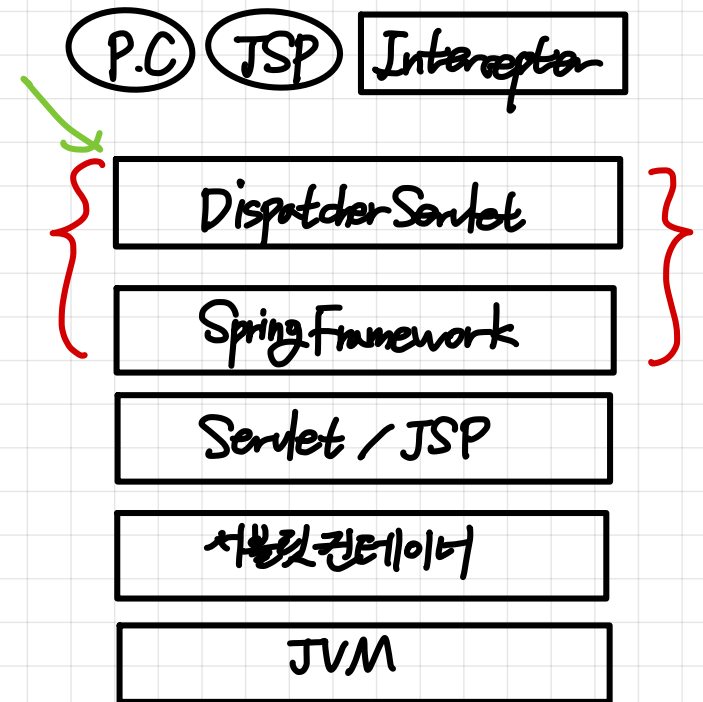


2021.11.18. 목 ( 일차 )

\* Web App. 기술 스택



Front Controller 역할

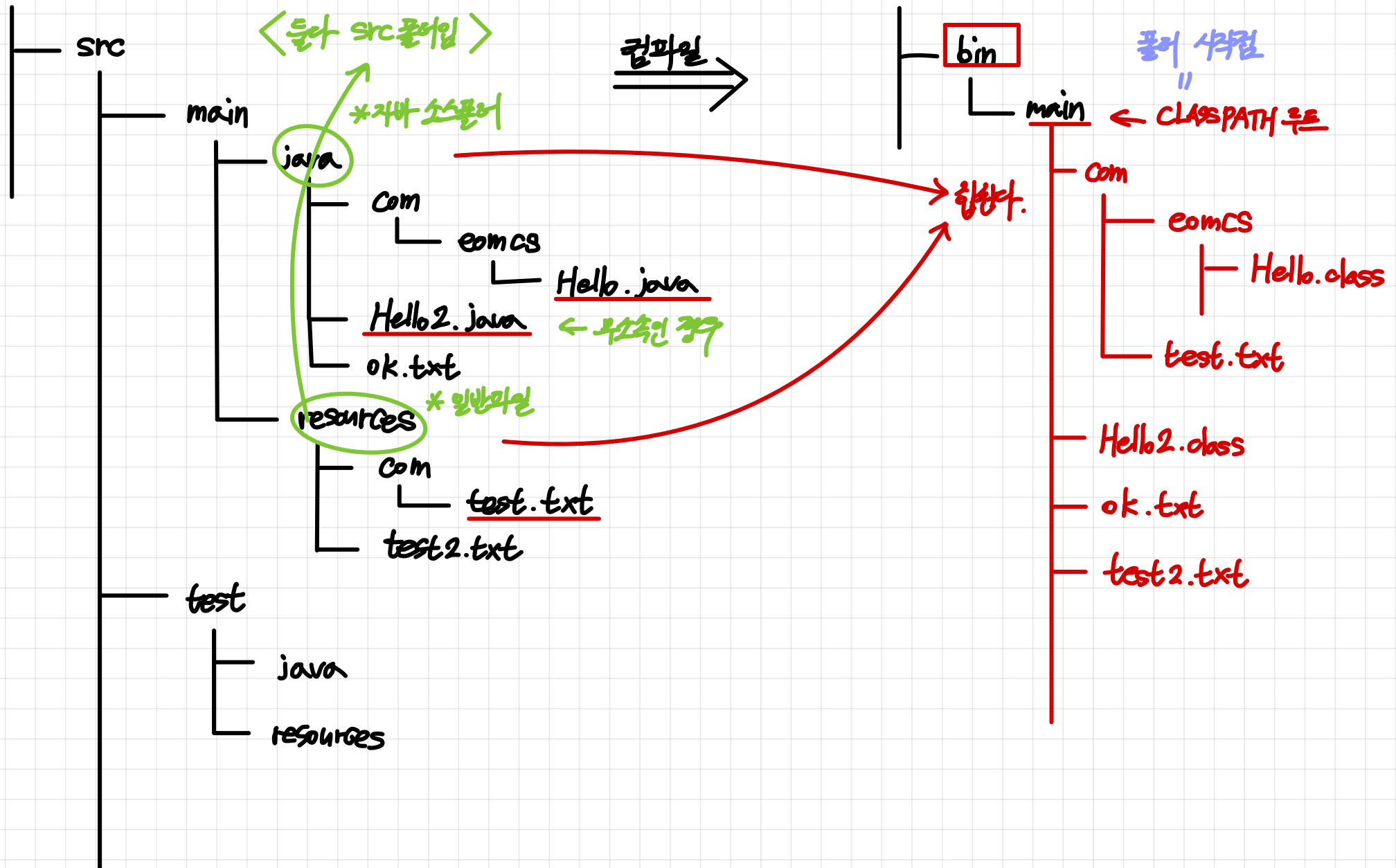


2021.11.18. 목 ( 일차 )

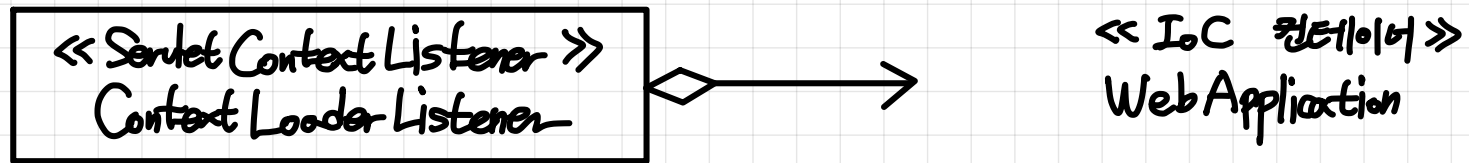
\* 자바 프로젝트 폴더와 빌드 경로

프로젝트 폴더

프로젝트

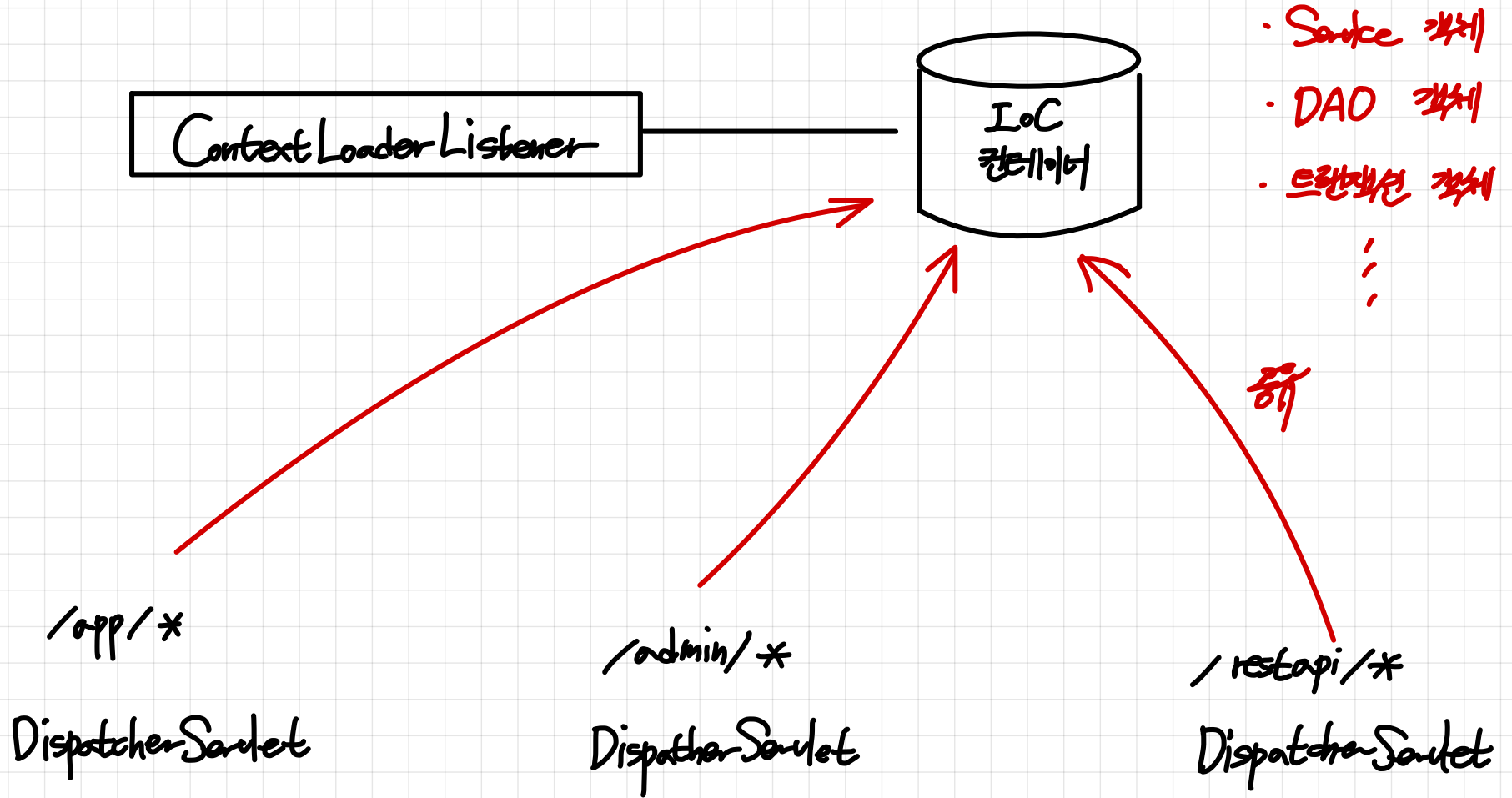


## \* Spring WebMVC 의 IoC 컨테이너



2021. 11. 18 (목, 월차)

## \* DispatcherServlet 과 ContextLoaderListener 의 관계

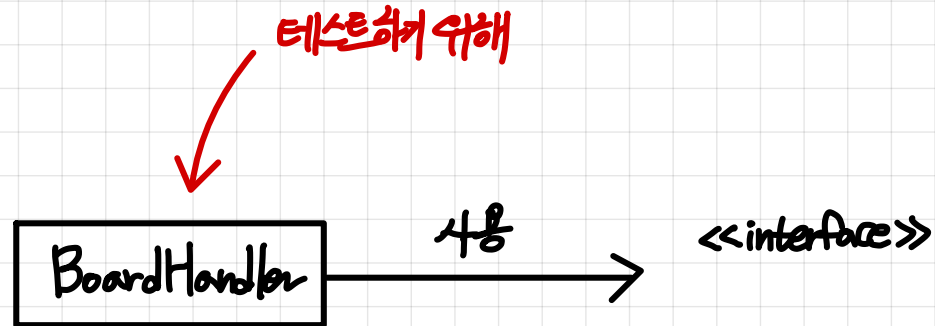


2021. 11. 19 (금, 오후)

\* Mockup 객체 - 테스트용 객체

\* mock up 어디까지 쓰일

만들 필요  
사용하기 전



원래는 안쓰  
test만함!  
↓

2021.11.19 (금, 일사)

## \* Service Provider 기법

→ 각에게 플러그인을 자동으로 장착시키는 기법.  
→ 가능한 자동으로 실행

JVM에는 JDBC API는 有

API 규격은 無 ⊕ DBMS마다 연개체 이름  
∴ DBMS 측에서 가능 제공

JVM

클래스

JDBC Driver

.jar를 불러  
관련 클래스를  
자동으로 실행하는 기술

클래스들 .jar



작동 설정 기술이 필요!



.jar 파일에 들어있는 클래스들  
JVM에서 바로 사용할 수 있도록  
실행해주는 기술

2021.11.19 (금, 일차)

\* 서비스 프로바이더 등록 경로

JVM

.jar

⋮

/META-INF

/services

java.sql.driver

↑

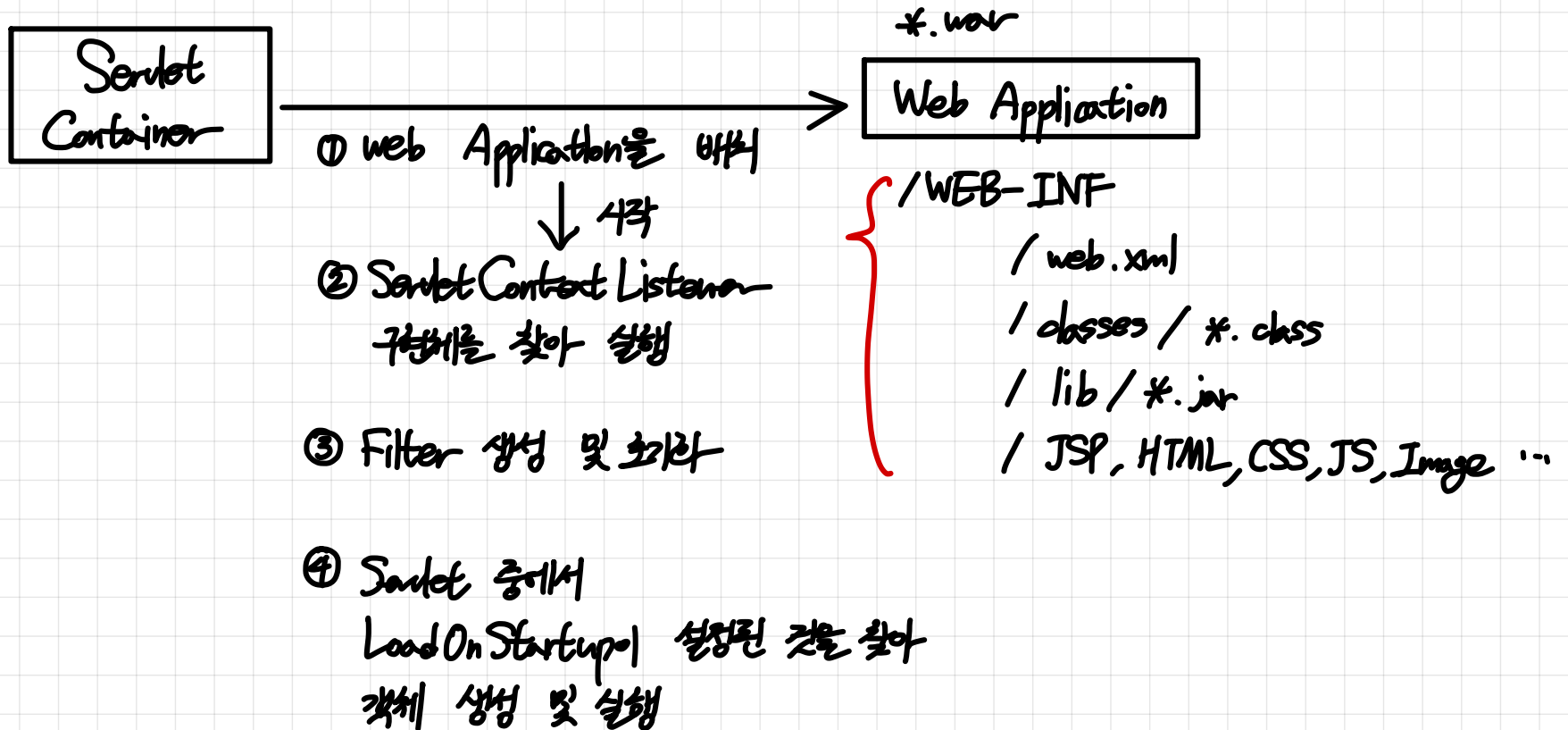
인터페이스명으로 되어있는 파일  
"

이 파일에는 파일명에 해당하는  
인터페이스의 구현클래스명이 들어있다.

## \* Service Provider 응용 예) 2

① 웹 애플리케이션을 초기화시키는 객체를 자동 등록하기

Servlet

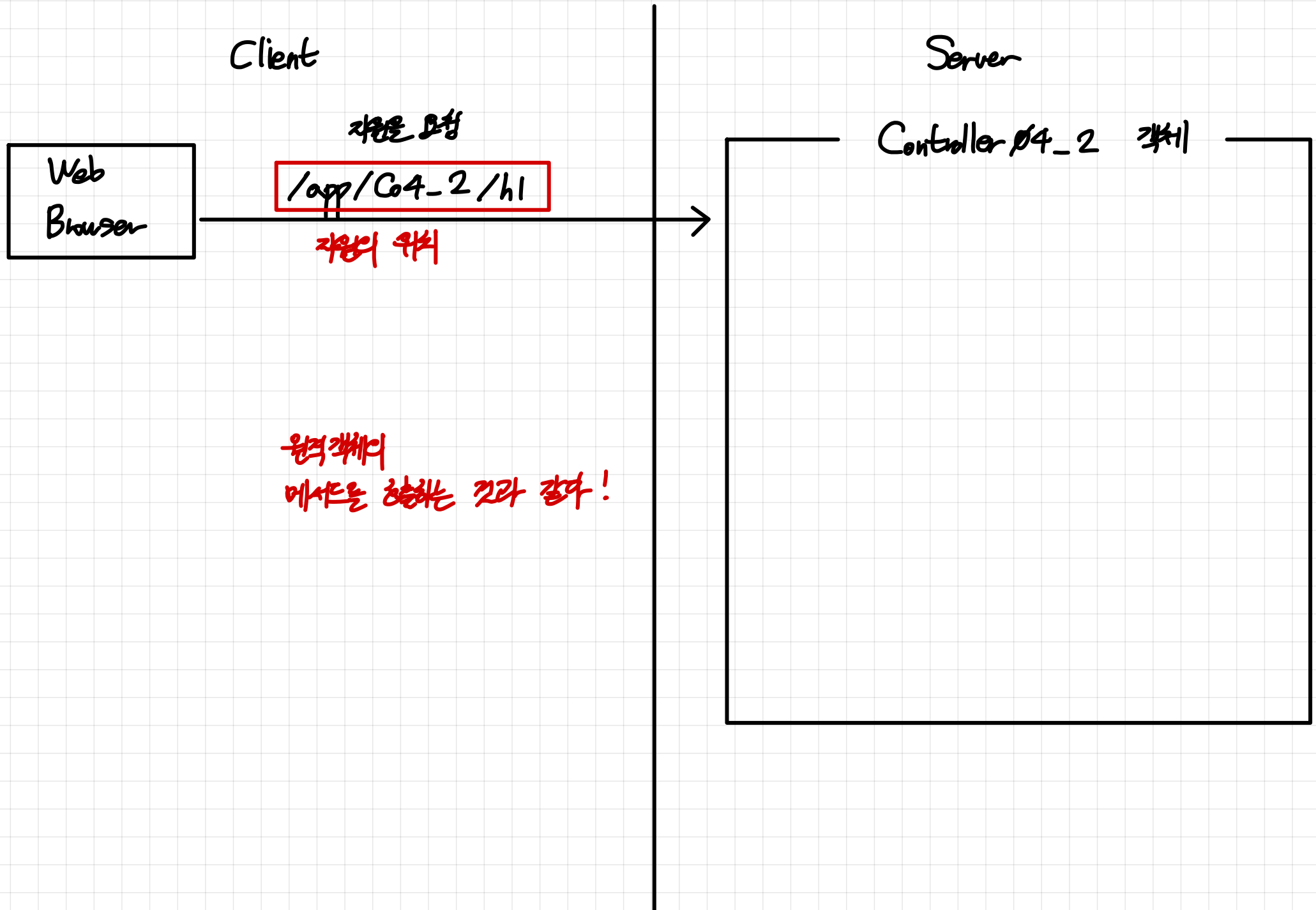




\* ServletContainerInitializer

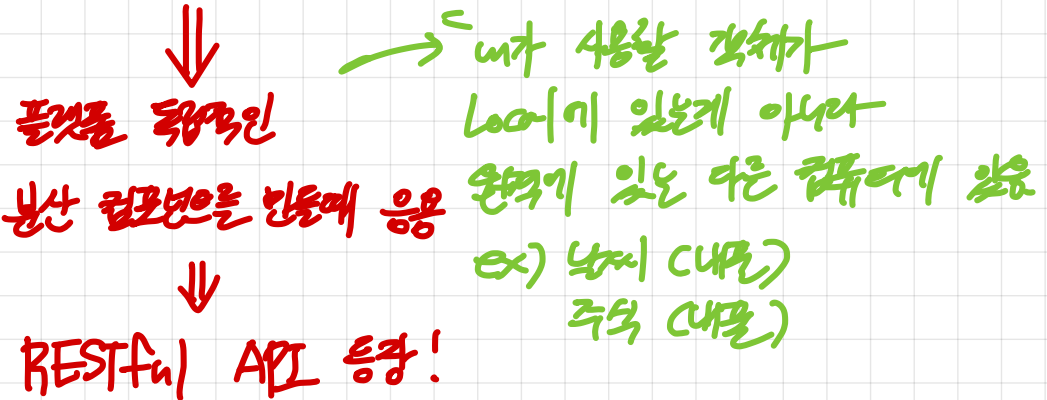
Savlet Carlsina

2021.11.19 (금, 일차)



# \* Web 애플리케이션 기술을 분산 컴퓨팅 환경에 응용

HTTP 프로그래밍 여러 동선하기 때문에 특정 프로그램 언어에 종속되지 않는다!



C App

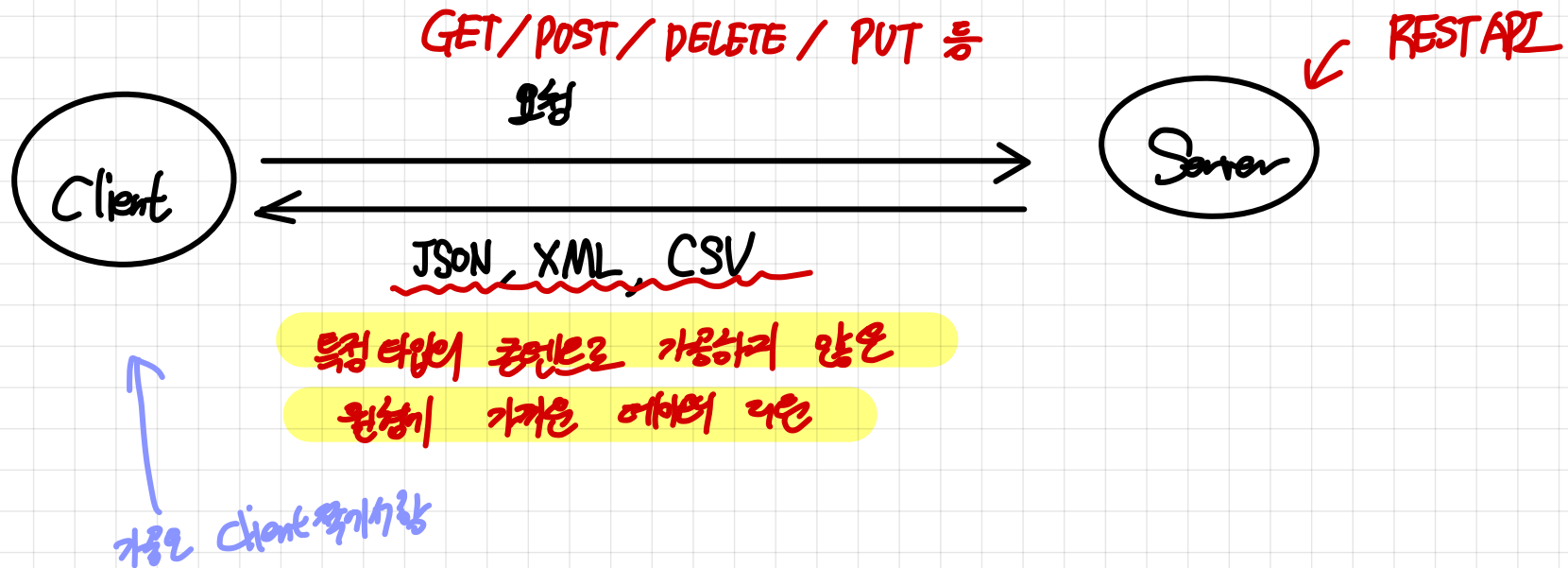
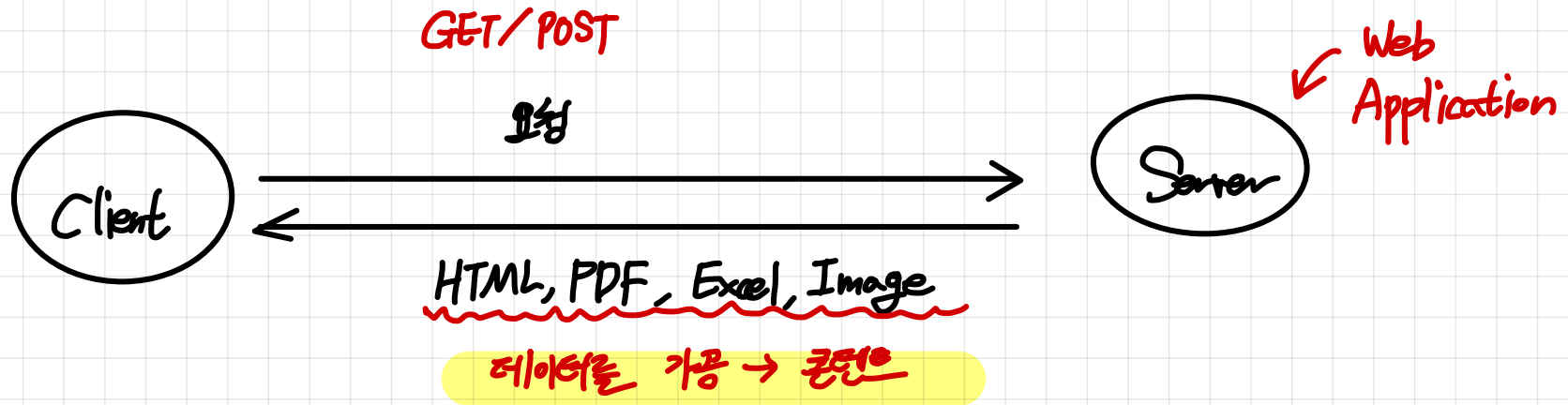
C++ App

Java App

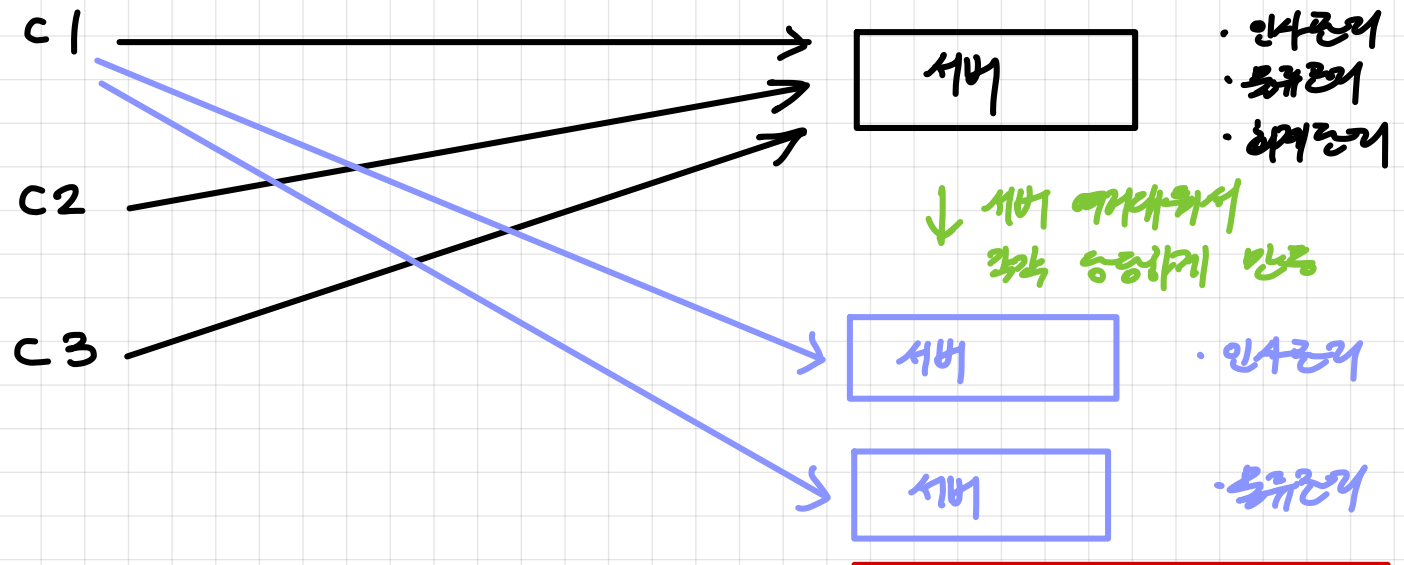
Python App

J.S App

# \* Web App VS REST API 차이점



\*



- PC
- Android
- iOS

불산된 객체로 강주

2021.11.22 (월, 9 일차)

<<front Controller>>







배포할 때 .war 가 아니라

.jar 를 배포함

→ JSP/ JSTL 처리하는 게 배포와 편하다.

( Web App + 서블릿컨테이너  
내장 )