

Web

0.들어가기전에

용어정리

Request : 클라이언트의 요청

Response : 클라이언트 요청에 대한 응답

Synch : 요청 후 응답될 때 까지 Browser 대기 (화면 전체 로딩)

Asynch : 요청 후 응답을 기다리지 않고 Browser 다른 일 (data만 변경)

Browser : IO, Network, Thread를 사용자 입장에서 만들어 놓은 것

GET : 모든 주소정보 노출 (header 를 통해 전달)

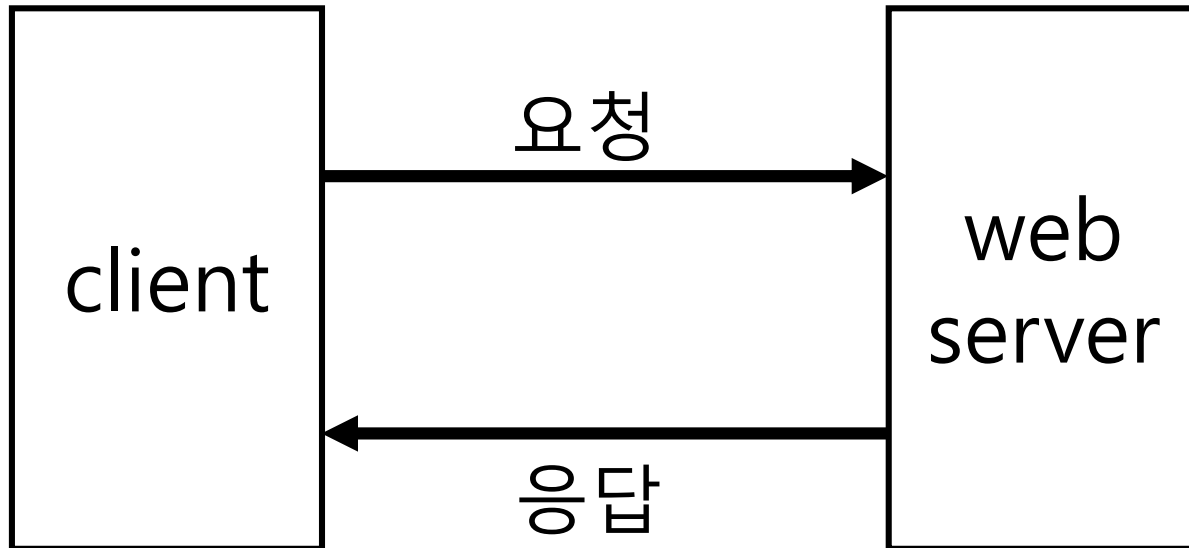
POST: 내부변수로 존재 -> 노출되지 않음 (body 를 통해 전달)

1.WebServer/WAS

web server

Web Server

- http를 통한 요청에 대해, html 문서나 오브젝트(이미지 파일 등)을 응답
(정적인 데이터 서비스)

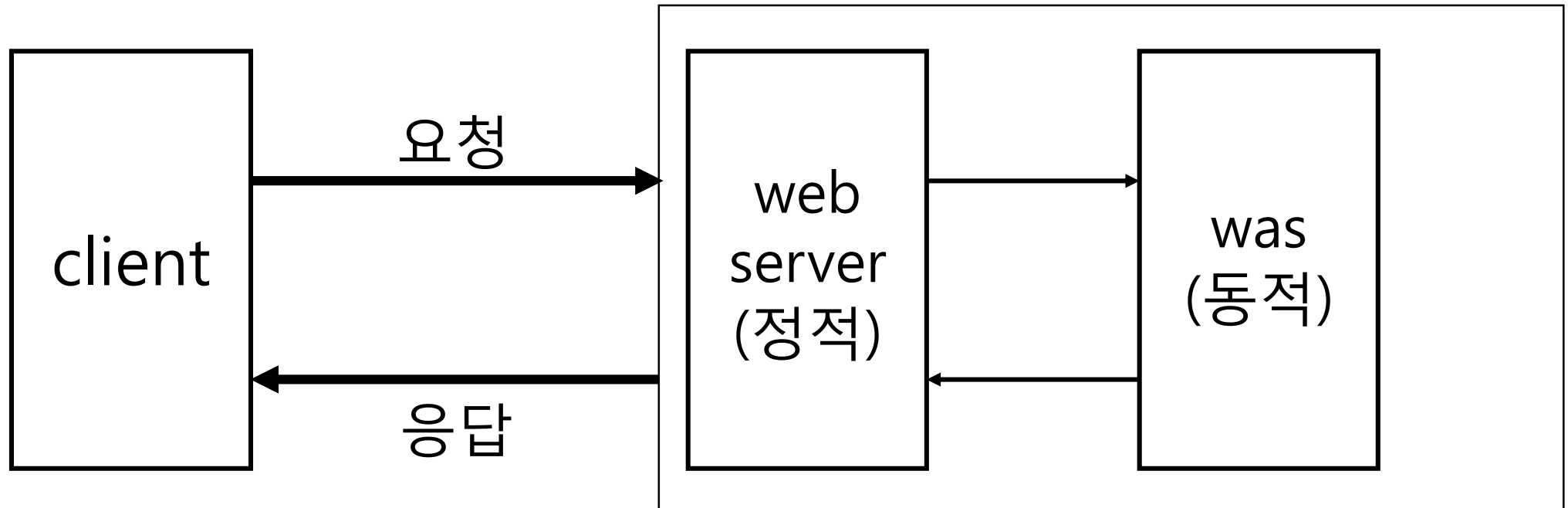


1.WebServer/WAS

was

WAS (Web Application Server)

- 트랜잭션, 보안, 트래픽 관리, DB Connection Pool 등의 기능 제공
(동적인 데이터 서비스 : jsp/Servlet, JNDI, JMAIL, JTA, JTS, JMS, ...)



1.WebServer/WAS

was

JNDI (Java Naming Directory Interface) : 이름과 객체를 mapping 시켜주는 naming service 에 접근하도록 도움

JMAIL : java 메일 서비스

JTA (Java Transaction API) : Transaction 관리를 위한 API

JTS (Java Transaction Service) : JTA를 지원하는 Transaction Manager의 구현 정의

JMS(Java Message Service) : 비동기식 메시징을 위한 표준 API

2.context

구조

물리적인 디렉토리 (배포 단위)

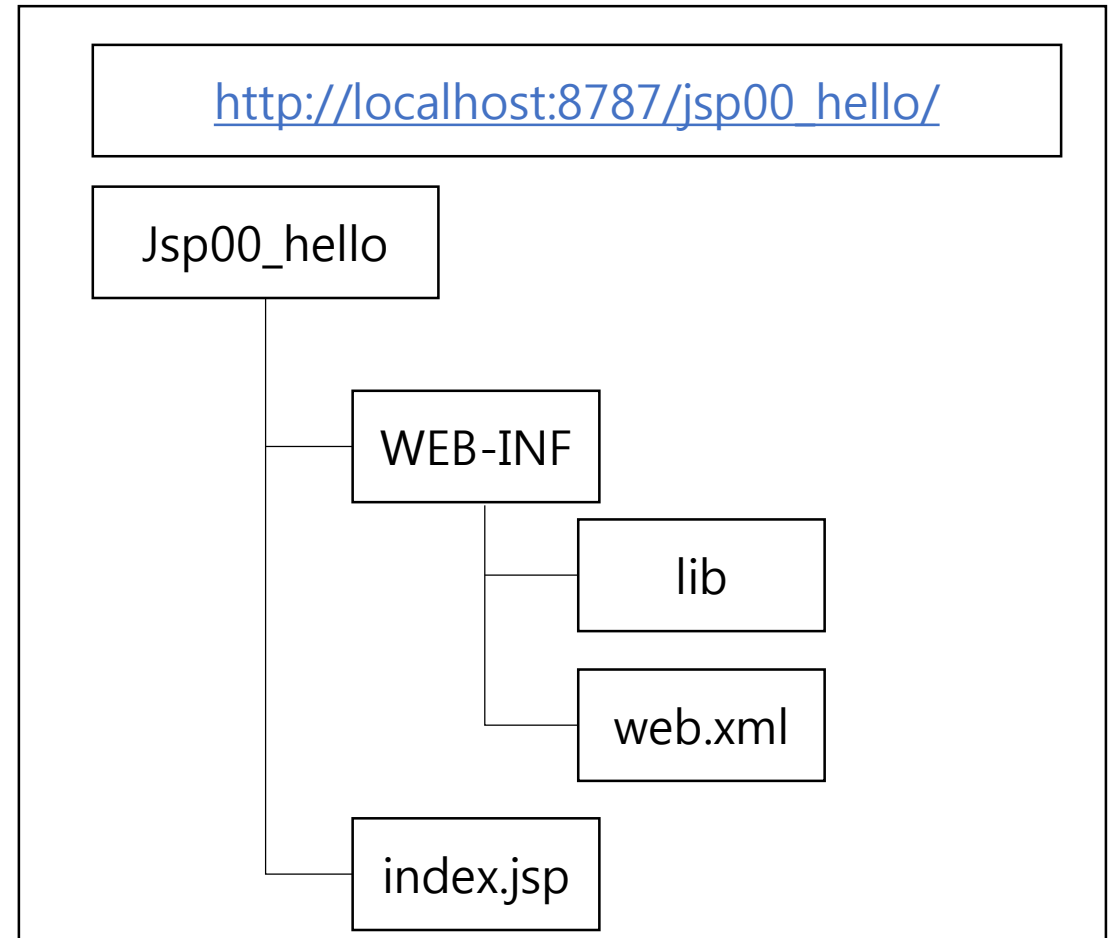
context : Jsp00_hello

WEB-INF : 디렉토리

lib : ~.jar (라이브러리 폴더)

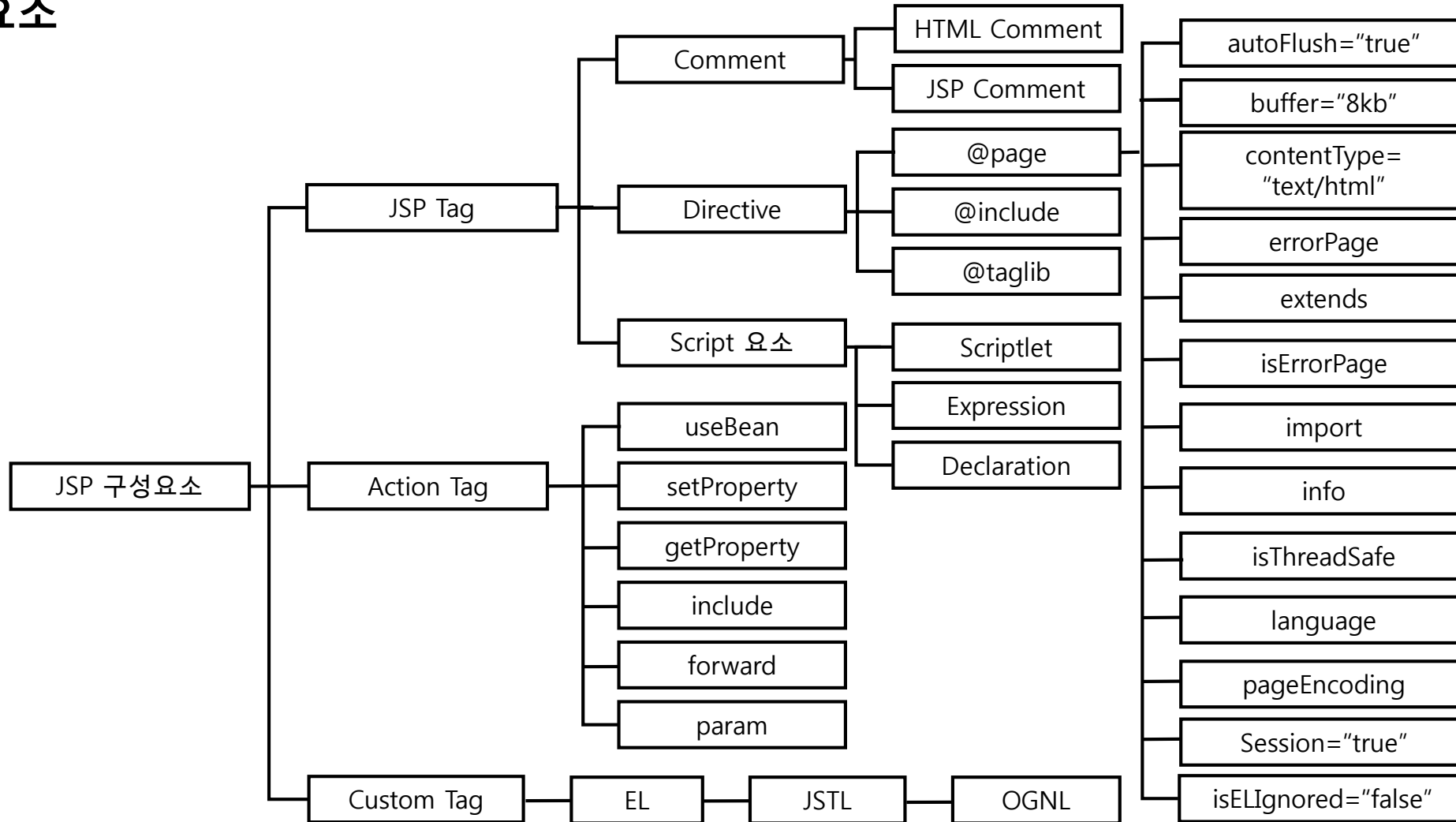
web.xml : 환경설정을 위한 xml

컨텍스트 환경을 설명하는 배포기술자
(Deployment Descriptor)



3.jsp

구성요소



4.객체 전달 범위

scope

scope : 객체 전달 범위

- 1) page : 현재 페이지에서 객체를 전달 (현재 페이지의 객체 유지)
- 2) request : 현재 페이지에서 요청한 다음 페이지까지만 객체 전달
(요청된 페이지까지 객체 유지)
- 3) session : 현재 Browser 에서 context 내에 있는 모든 페이지에 객체 유지
- 4) application : 현재 context 내에서 객체 유지. WebXML 에서도 사용 가능

* 공통 메소드 : `setAttribute`, `getAttribute`, `removeAttribute`, `getAttributeName`

0.챕터

소제목

내용