

발표 정가영

Table of Contents

01

HTTP

02

WAS

03

DB와 INDEX

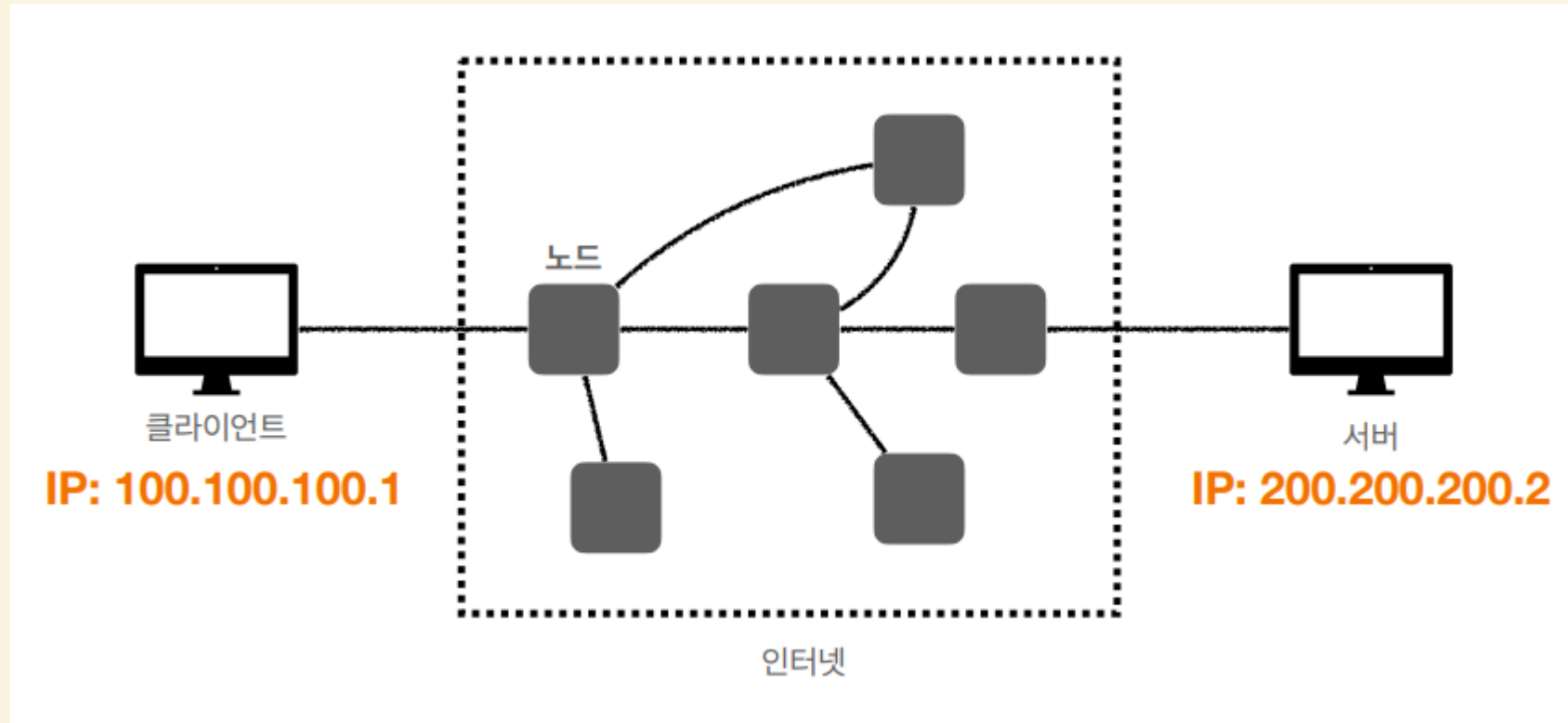
01

HTTP

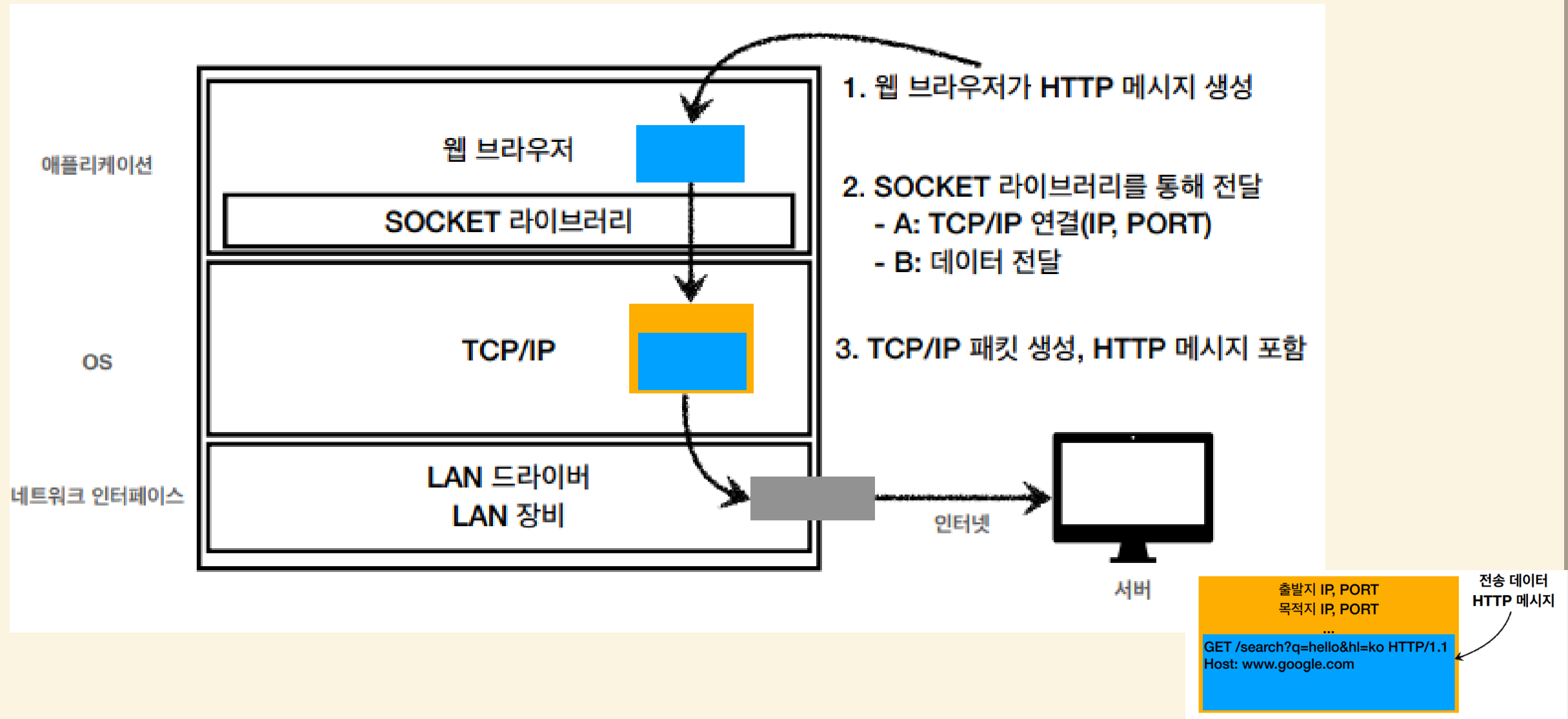
Hyper Text Transfer Protocol

01. HTTP

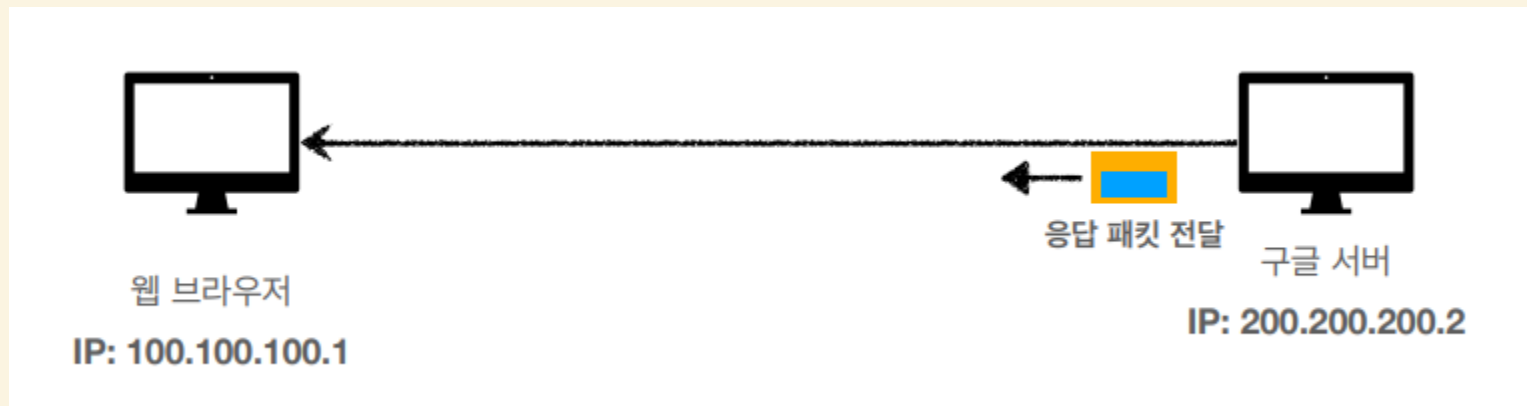
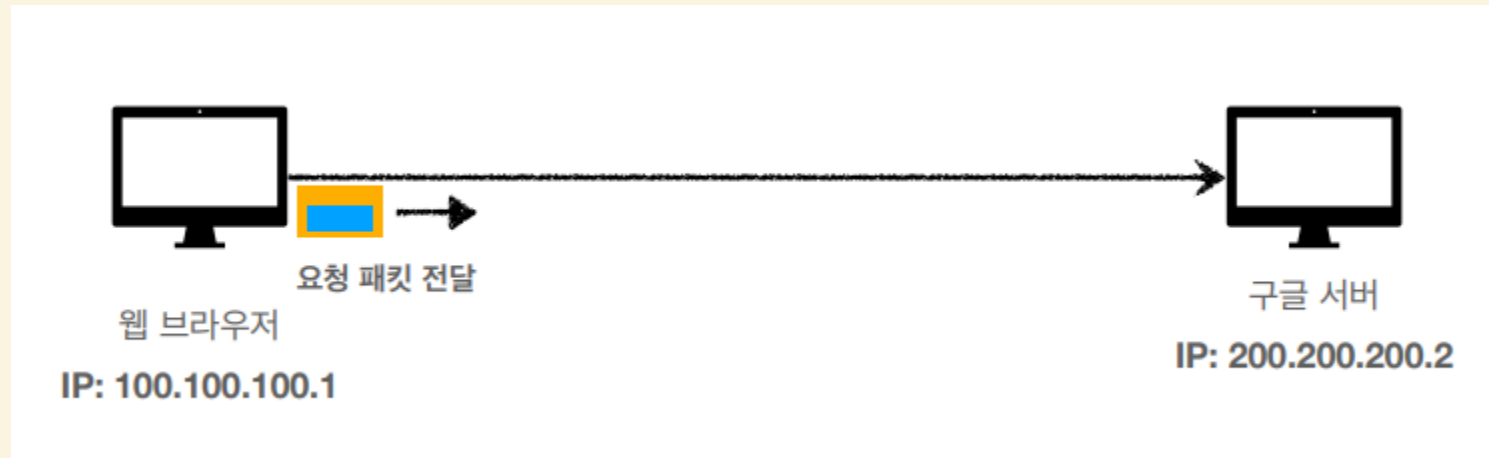
인터넷에서 컴퓨터 둘은 어떻게 통신할까?



01. HTTP



01. HTTP



01. HTTP

Hyper Text Transfer Protocol

HTTP 메시지에 모든 것을 전송

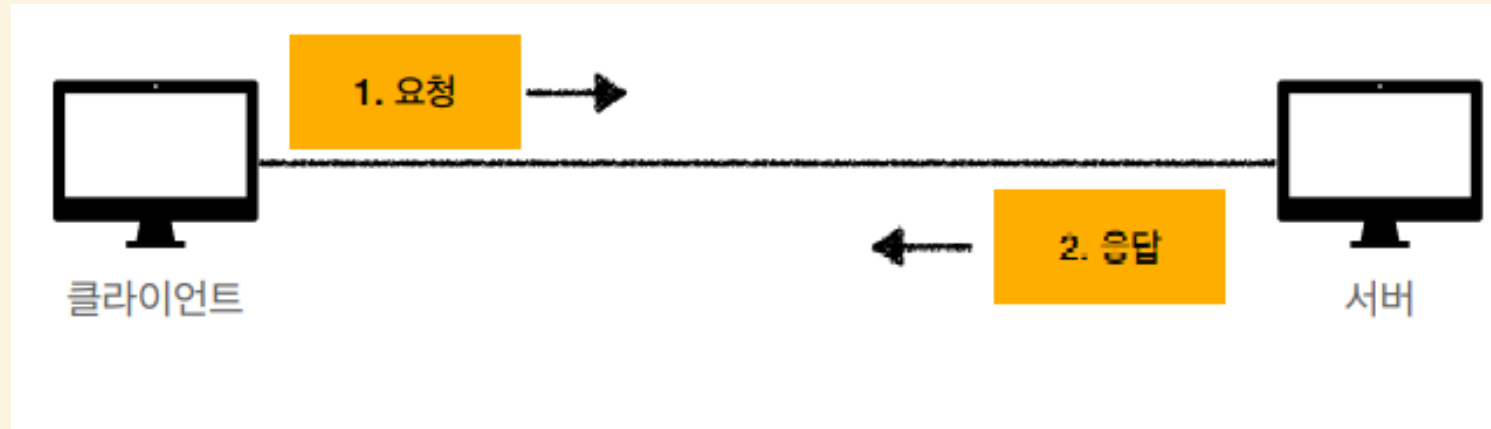
거의 모든 형태의 데이터 전송 가능
HTML, TEXT
IMAGE, 음성, 영상, 파일
JSON, XML(API)

HTTP/1.1 1997년: 가장 많이 사용, 우리에게 가장 중요한 버전

01. HTTP 특징

1. 클라이언트 서버 구조
2. 무상태 프로토콜(스태이스리스), 비연결성
3. HTTP 메시지
4. 단순함, 확장 가능

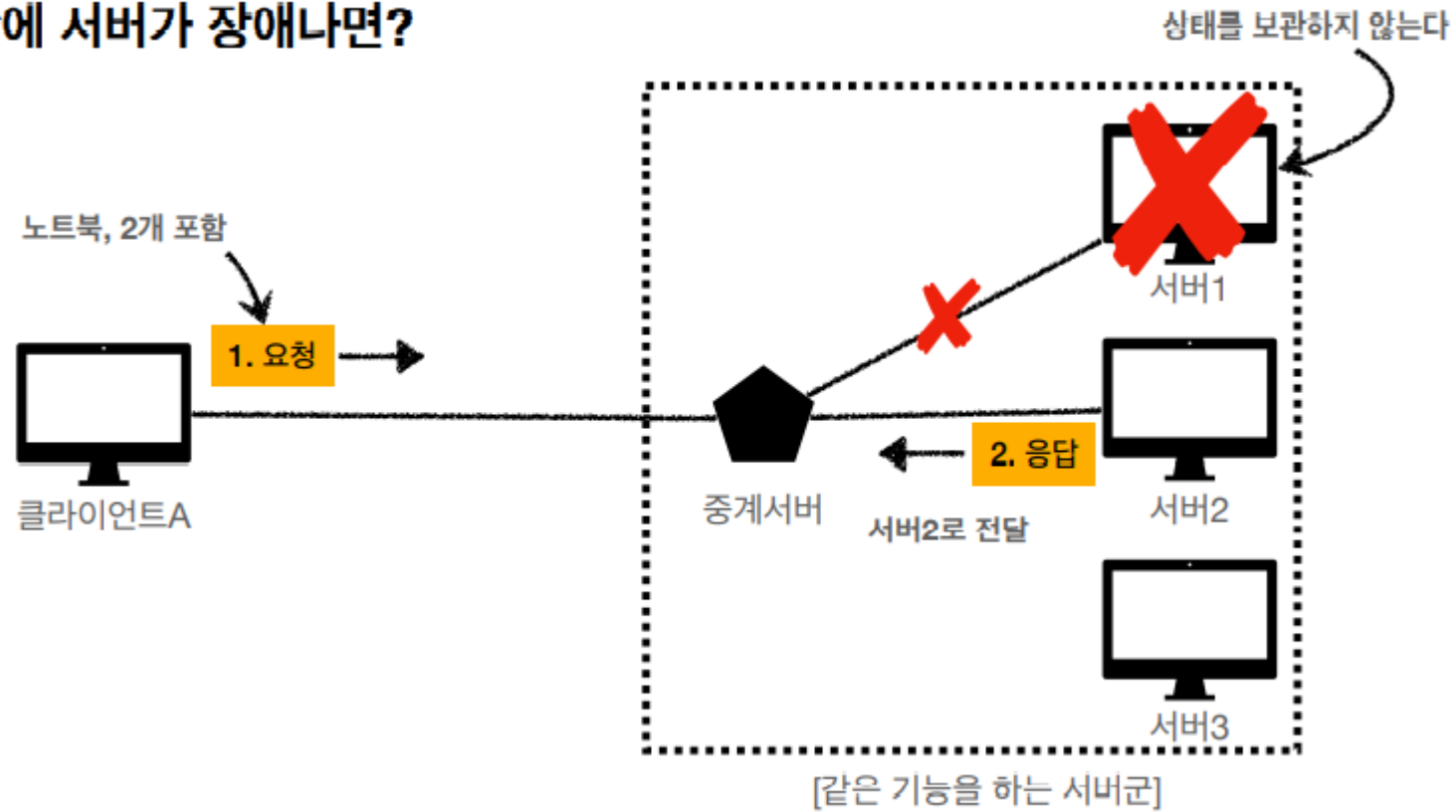
01. HTTP 특징 1. 클라이언트 서버 구조



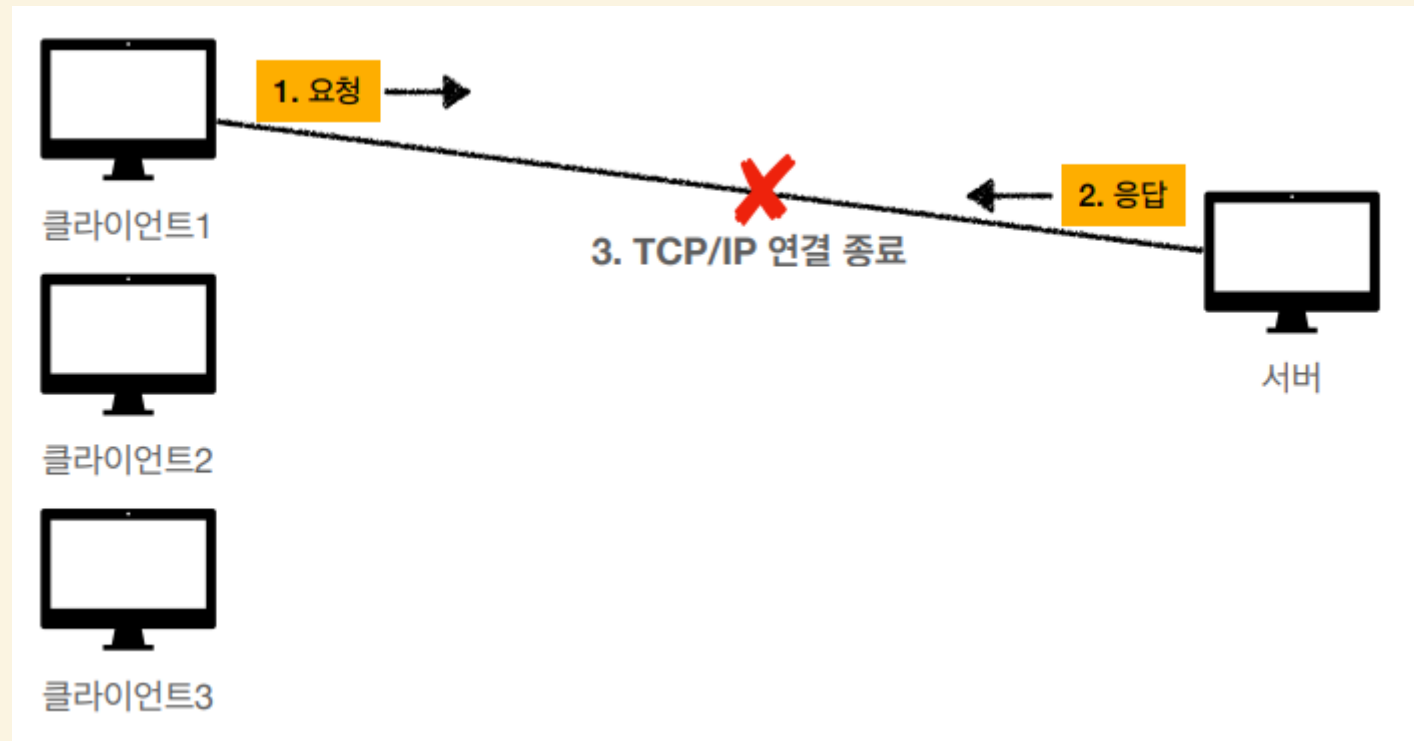
- Request- response 구조
- 클라이언트는 서버에 요청 보내고 응답 대기
- 서버가 요청에 대한 결과 만들어서 응답

01. HTTP 특징 2. 무상태 프로토콜

중간에 서버가 장애나면?



01. HTTP 특징 2. 비연결성



01. HTTP 특징 2. 비연결성



웹 브라우저로 사이트 요청하면 HTML, 자바스크립트, CSS, 이미지 등 많은 자원이 함께 다운로드

과거: 연결, 요청, 응답, 종료 반복 되어 낭비가 됨
지금은 지속연결로 문제 해결(연결 해놓고, 하나씩 응답)

01. HTTP 특징 3. HTTP 메시지

HTTP 요청 메시지



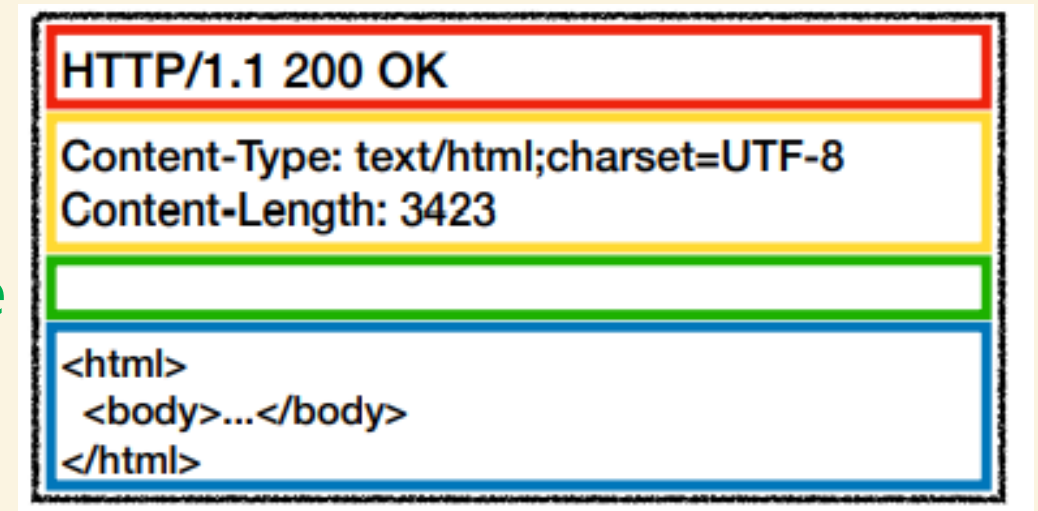
Start-line
Header

Empty line
Message
body

HTTP메서드 요청대상 HTTP버전

HTTP메서드: GET, POST, PUT, DELETE...
서버가 수행해야할 동작 지정
요청대상: 절대경로
HTTP버전

HTTP 응답 메시지



HTTP버전 HTTP상태코드 이유문구

HTTP상태코드: 요청성공.실패 알려줌
200:성공
400: 클라이언트 요청 오류
500: 서버 내부 오류

01. HTTP 메서드

GET: 리소스 조회

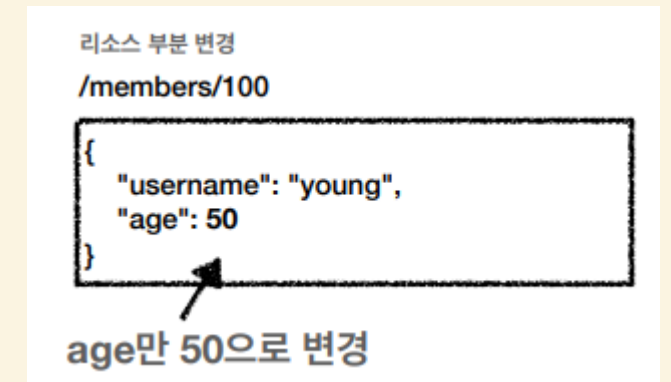
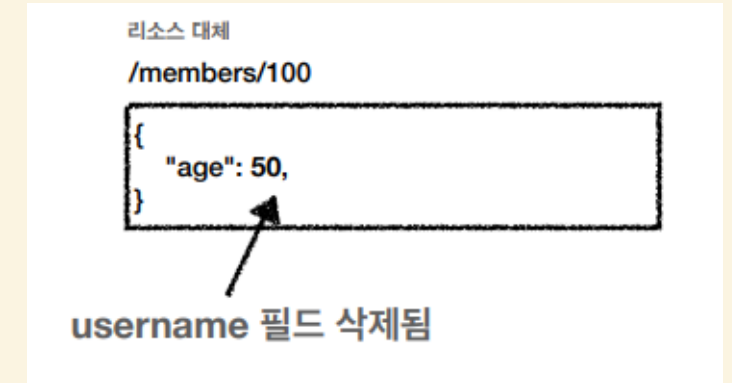
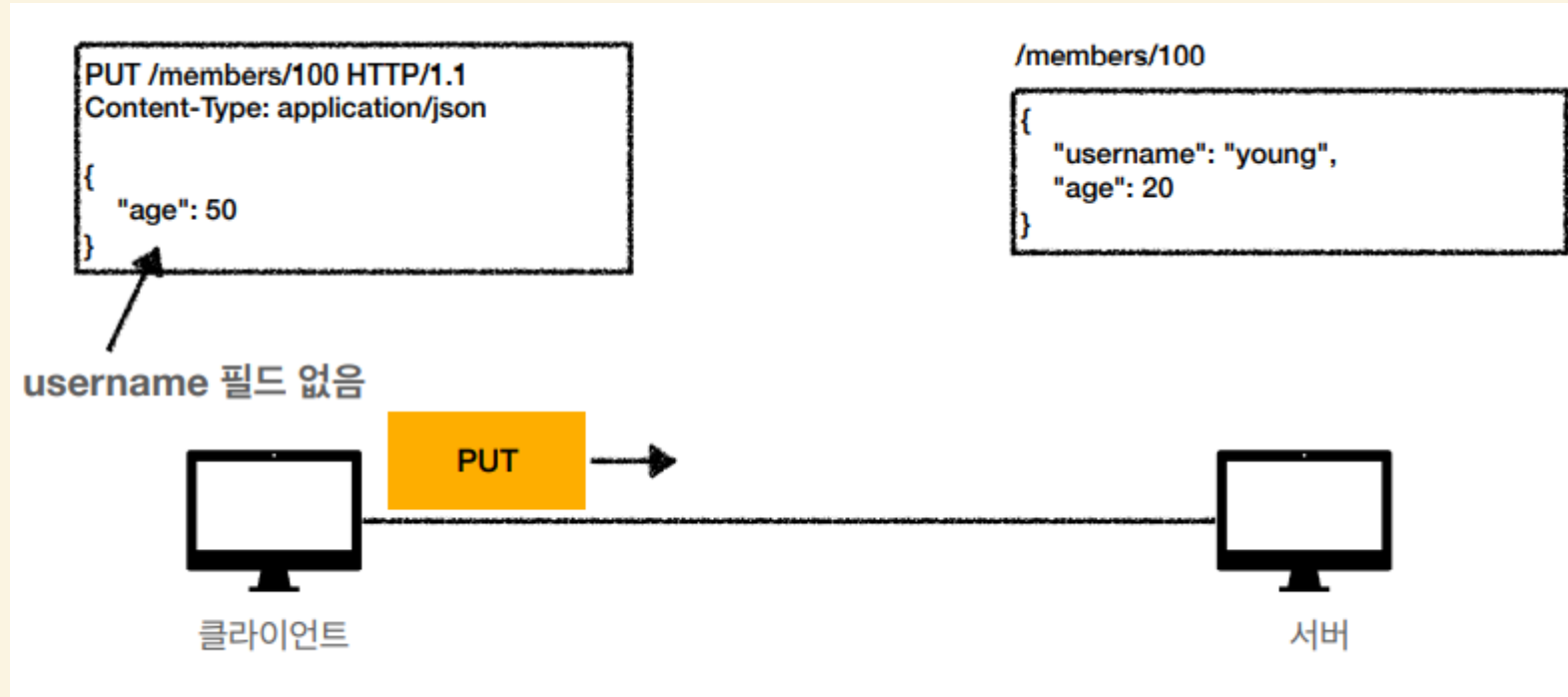
POST: 요청 데이터 처리, 주로 등록, 회원가입, 게시판 글쓰기, 댓글

PUT: 리소스 대체, 해당 리소스 없으면 생성

PATCH: 리소스 부분 변경

DELETE: 리소스 삭제

01. HTTP 메서드 – PUT/PATCH 차이점



01. HTTP 데이터 전달방식

1. 쿼리 파라미터를 통한 데이터 전송

- GET
- 주로 정렬 필터(검색어)

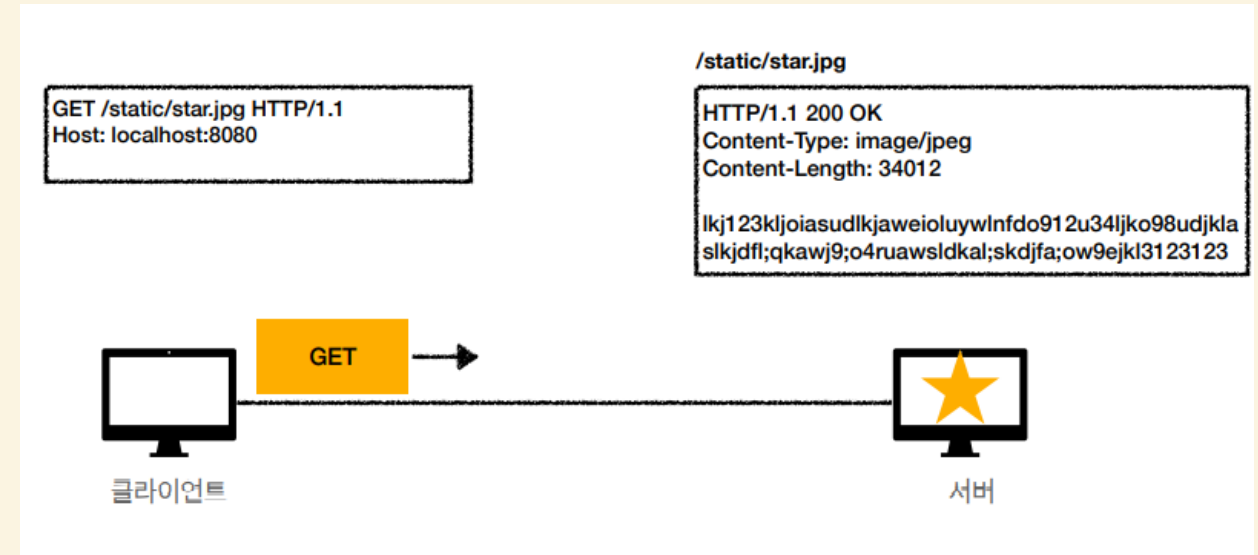
2. 메시지 바디를 통한 데이터 전송

- POST, PUT, PATCH
- 회원가입, 상품 주문, 리소스 등록, 리소스 변경

01. HTTP 데이터 전달방식

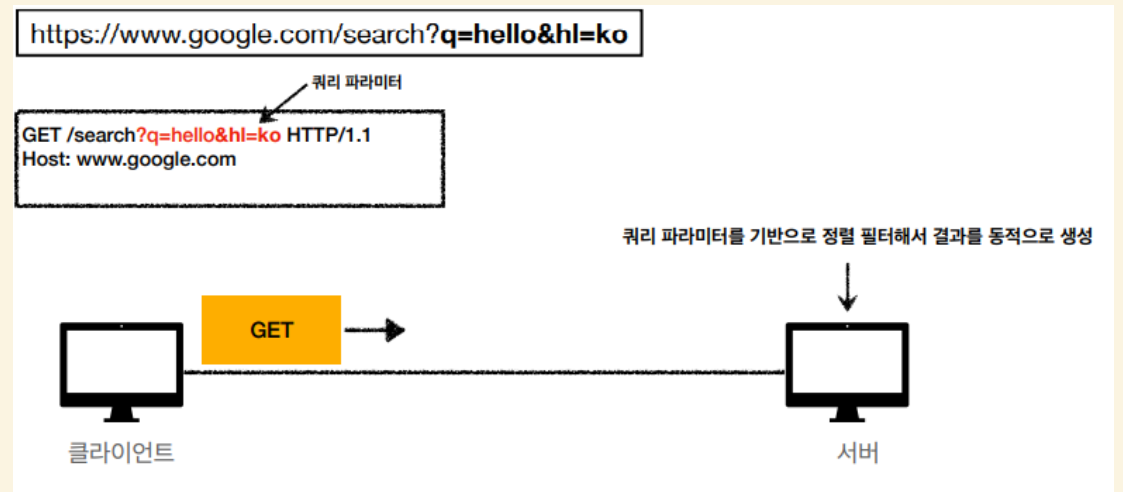
1. 정적 데이터 조회

- 이미지, 정적 텍스트 문서
- GET



2. 동적 데이터 조회

- 쿼리 파라미터 사용
- 검색, 게시판 목록 정렬 필터
- GET



01. HTTP 데이터 전달방식

<p>username: <input type="text" value="kim"/> age: <input type="text" value="20"/> 전송</p> <pre><form action="/save" method="post"> <input type="text" name="username" /> <input type="text" name="age" /> <button type="submit">전송</button> </form></pre>	<p>웹 브라우저가 생성한 요청 HTTP 메시지</p> <pre>POST /save HTTP/1.1 Host: localhost:8080 Content-Type: application/x-www-form-urlencoded username=kim&age=20</pre>
---	---

3. HTML Form 통한 데이터 전송

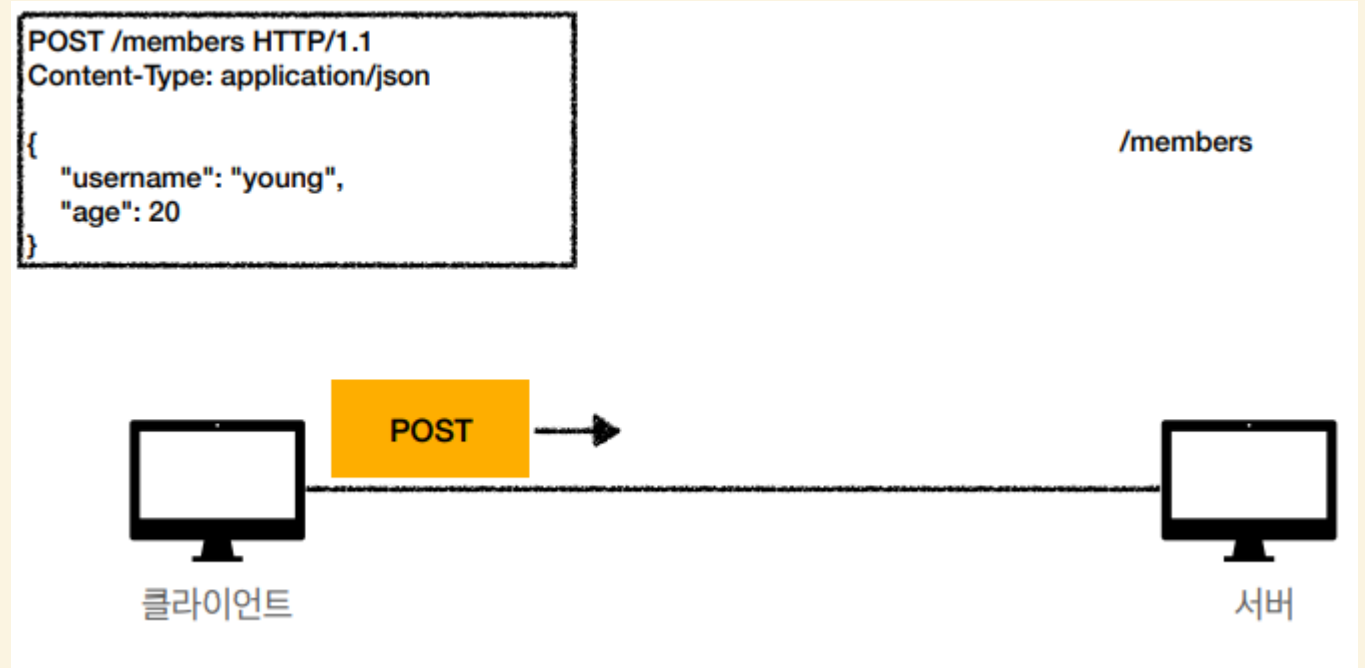
-POST: 저장

- 회원가입, 상품 주문, 데이터 변경

* HTML Form: GET, POST만 지원

4. HTML API 통한 데이터 전송

01. HTTP 데이터 전달방식



4. HTML API 통한 데이터 전송

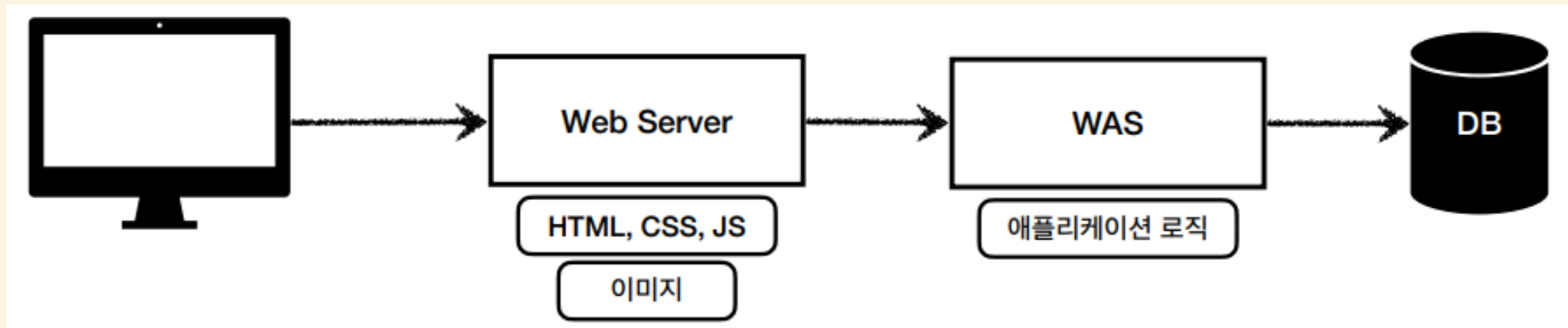
- 서버 TO 서버
- HTML에서 Form 전송 대신 자바 스크립트를 통한 통신에 사용
- React, VueJs 같은 웹 클라이언트와 API 통신
- POST, PUT, PATCH: 메시지 바디를 통해 데이터 전송
- Content-Type: Application/json 주로 사용

02

WAS

Web Application Server

02. 웹 시스템 구성



02. WAS 종류

상용소프트웨어

아파치 Tomcat
아파치 재단에서 관리

Jetty
이클립스 재단에서 관리.

WebLogic
오라클에서 관리

JBoss
레드햇에서 관리.

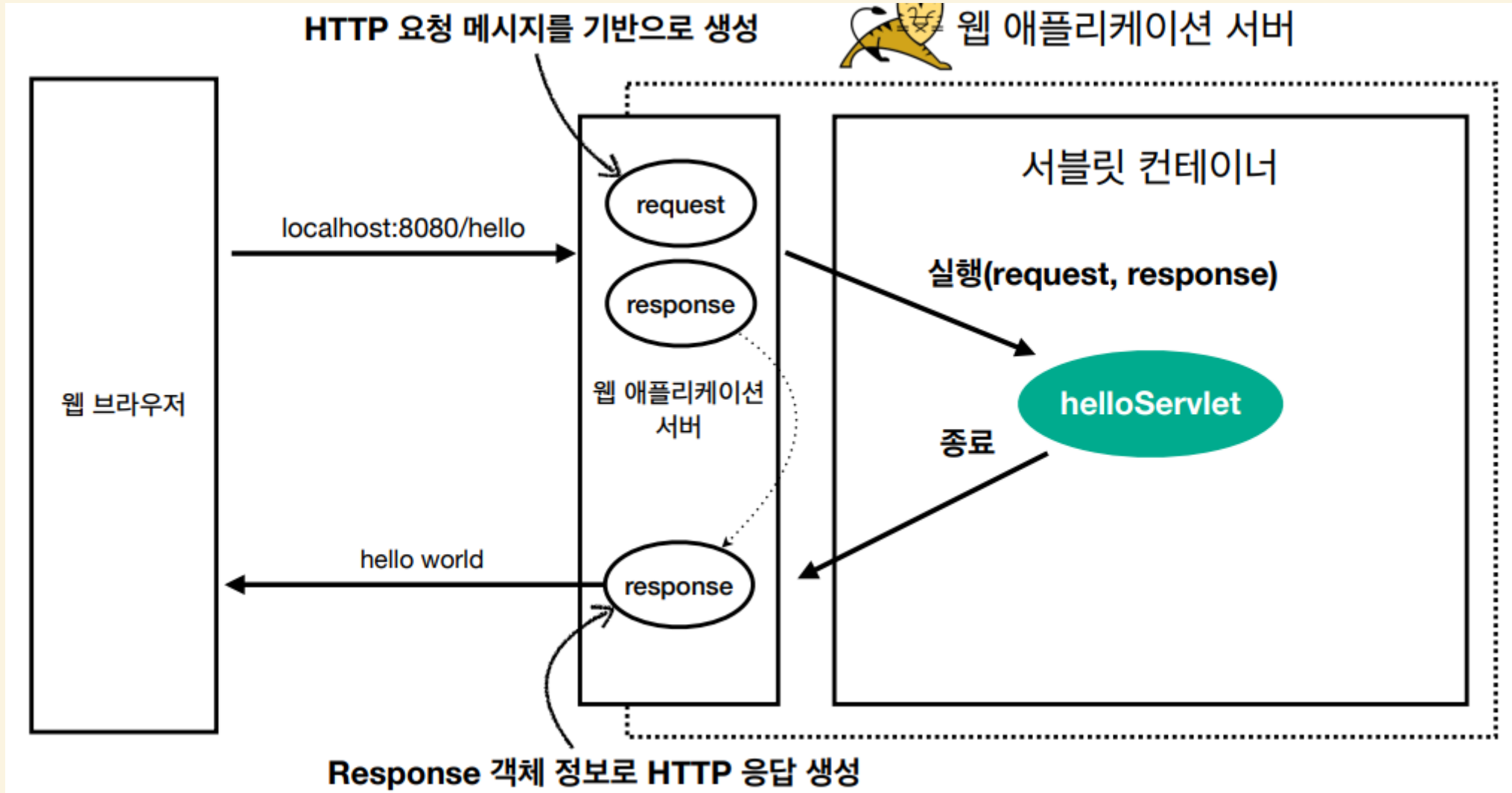
오픈소스

Jeus
국내 티맥스소프트에서 관리.

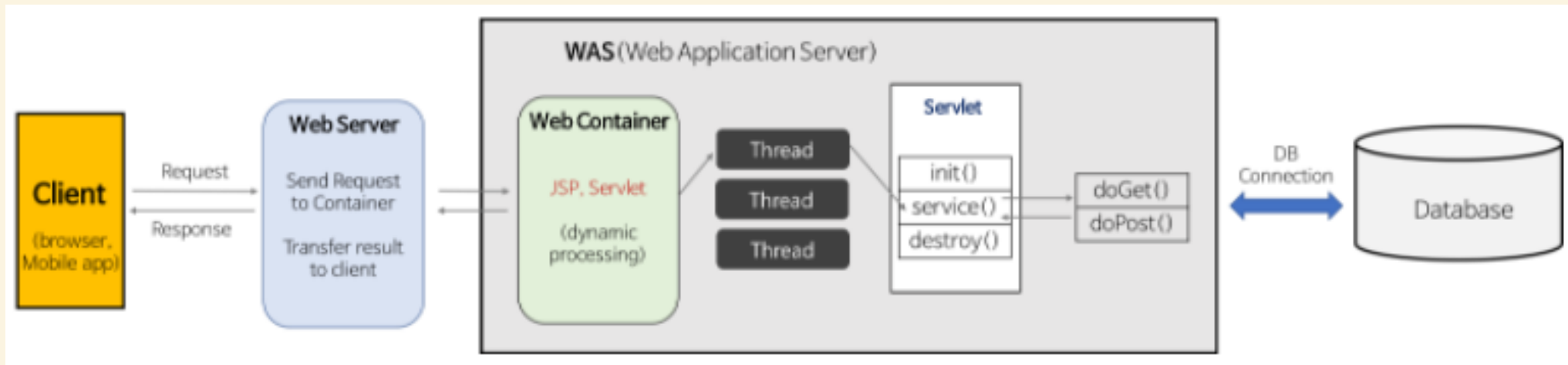
02. WAS 기능

1. 프로그램 실행 환경 및 DB 접속 기능 제공
2. 여러 트랜잭션 관리 기능
3. 업무 처리 비즈니스 로직 수행

02. WAS 동작 과정



02. Web Service Architecture



02. 서블릿

HTTP 요청시

- WAS는 Request, Response 객체를 새로 만들어서 서블릿 객체 호출
- 개발자는 Request 객체에서 HTTP 요청 정보를 편리하게 꺼내서 사용
- 개발자는 Response 객체에 HTTP 응답 정보를 편리하게 입력
- WAS는 Response 객체에 담겨있는 내용으로 HTTP 응답 정보를 생성

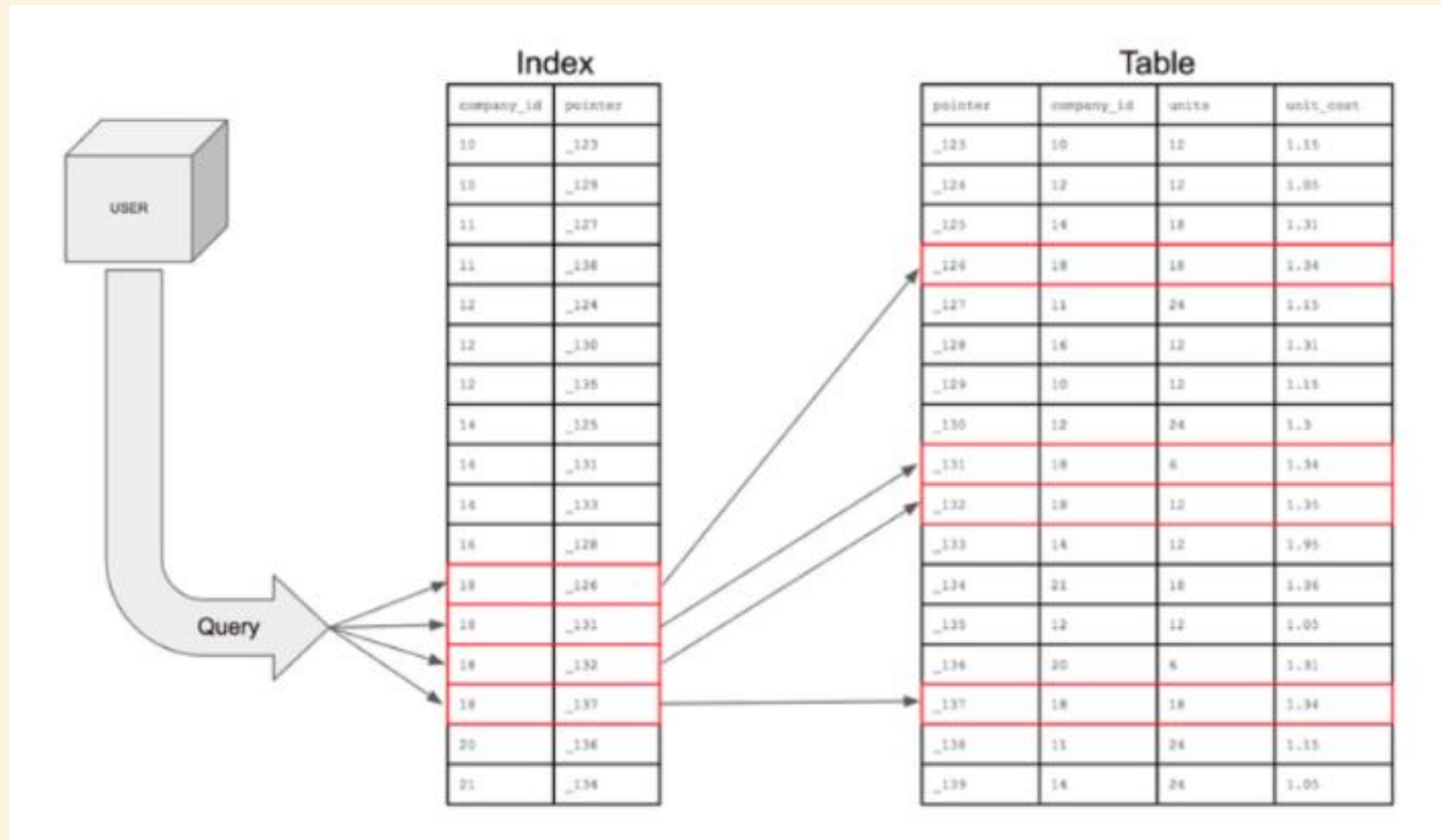
02. 서블릿 컨테이너

- 톰캣처럼 서블릿을 지원하는 WAS를 서블릿 컨테이너라고 함
- 서블릿 컨테이너는 서블릿 객체를 생성, 초기화, 호출, 종료하는 생명주기 관리
- 서블릿 객체는 싱글톤으로 관리
- 고객의 요청이 올 때 마다 계속 객체를 생성하는 것은 비효율
- 최초 로딩 시점에 서블릿 객체를 미리 만들어두고 재활용
- 모든 고객 요청은 동일한 서블릿 객체 인스턴스에 접근
- 서블릿 컨테이너 종료시 함께 종료
- JSP도 서블릿으로 변환 되어서 사용
- 동시 요청을 위한 멀티 쓰레드 처리 지원

03

DB와 INDEX

03. INDEX ?



03. INDEX 장단점?

<장점>

1. 테이블 검색하는 속도와 성능 향상
2. 인덱스를 이용하면 데이터들이 정렬되어 있기 때문에 조건에 맞는 데이터를 빠르게 찾을 수 있다.

<단점>

1. 인덱스를 관리하기 위한 추가 작업이 필요
2. 추가 저장 공간 필요
3. 잘못 사용하는 경우 오히려 검색 성능 저하

03. INDEX 사용하면 좋은 경우?

1. 규모가 큰 테이블
2. 삽입, 수정, 삭제 작업이 자주 발생하지 않는 컬럼
3. WHERE 나 ORDER BY, JOIN 등이 자주 사용되는 컬럼
4. 데이터의 중복도가 낮은 컬럼

03. INDEX 자료구조

1. 해시 테이블(Hash Table)

- key, value 한쌍으로 저장하는 자료구조
- 빠른 시간만에 원하는 데이터를 탐색할 수 있는 구조
- 해시 테이블은 등호(=) 연산에 최적화되어 있음

2. B+Tree

- 테이블을 생성시 생성하고 싶은 인덱스 컬럼을 지정
- 생성 후 인덱스 조회 시, WHERE 절이 포함된 쿼리로 조회
- 인덱스로 저장된 Key-Value값을 참조해서 결과 출력

Q & A

Thank you!

