

# HTTP

🕒 생성일	@2021년 7월 13일 오후 5:01
☰ 태그	

현재 세상의 대부분의 데이터 송수신은 HTTP 를 사용  
바야흐로 HTTP 의 시대

## 웹서버

정적 파일을 요청에 따라 응답. 주고받음.

HTML,CSS,JS,IMG,VIDEO

- Nginx,Apache

## 웹 애플리케이션 서버 WAS (와스)

웹서버의 기능을 포함한다.

큰 차이는 프로그램 코드를 통해 애플리케이션 로직을 수행.

app code 를 수행하는데 특화됨

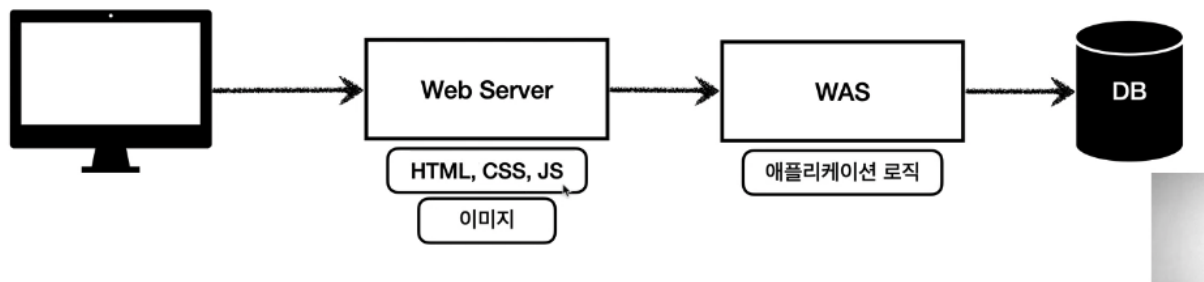
- 동적 HTML, HTTP API
- servlet, JSP, Spring MVC
- tomcat, Jetty

## web system 구성

WAS + DB 가 기본구성임

WAS 의 장애는 너무나 중대함. 아예 에러화면까지 접근 불가능

그래서

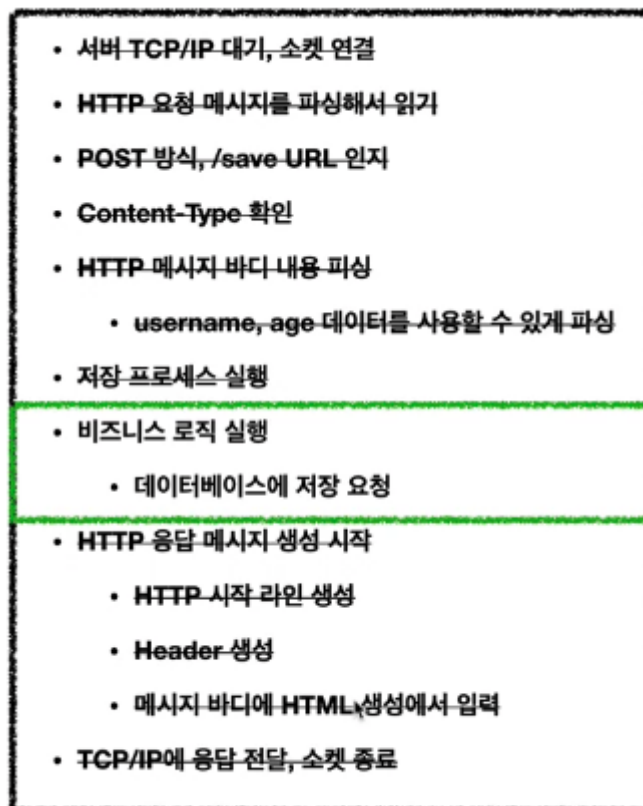


정적소스 : Web server, 동적처리 : WAS

로 역할 분리한다. 아래의 장점들을 가진다.

- 각자의 역할을 구분해 WAS의 무게를 가볍게 함
- 시스템 리소스 효율적으로 사용. 스케일링을 각자 나눠 하면 된다

## 서블릿



개발자가 애플리케이션을 만들고자 할 때, 초록네모 앞 뒤는 언제나 공통되고 반복되는 규약이다.

개발자는 비즈니스 로직에만 집중해야 효율적일 것이다. 공통되는 프로토콜에 대응해 모든 작업을 대신해주는 것이 서블릿이다.

## 서블릿컨테이너

대표적으로 톰캣이 서블릿컨테이너다. 서블릿컨테이너는 **서블릿이 지원되는 WAS** 를 의미한다.

서블릿 컨테이너는 서블릿 객체들의 생명주기를 관리해준다. 이 서블릿 객체는 **싱글톤**이다.

그러므로 서블릿에서 공유변수가 생기지 않게 주의해야한다.

- JSP 도 서블릿으로 변환된다
- 멀티스레드 처리를 지원한다

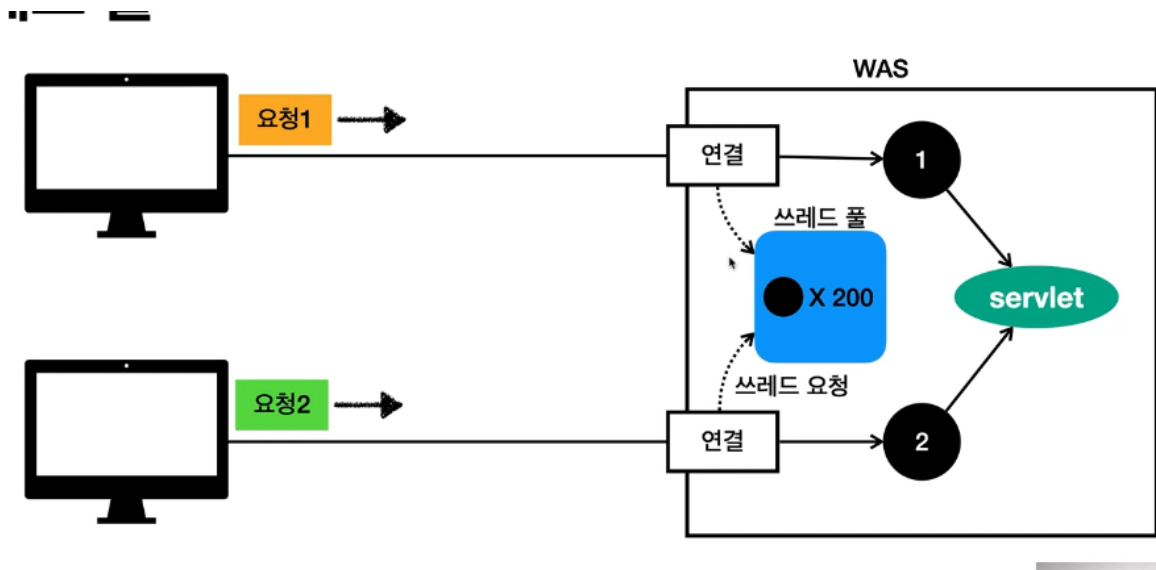
## 멀티스레드

백엔드는 무조건 잘 알아야한다.

**서블릿을 호출하는 주체가 스레드**다. 앱의 코드를 하나하나 실행하는 것은 쓰레드다. 자바는 메인쓰레드가 main이란 이름의 스레드가 실행된다.

- 요청마다 스레드 생성의 장단점
  - 장점
    - 동시 요청 가능
    - 어떤 요청이 지연되어도 독립적이라 다른 요청에 영향을 안미침
  - 단점
    - 스레드 생성 비용 비쌈
    - 컨텍스트 스위칭 자주 발생
    - 스레드 너무 많으면 터짐

## 스레드 풀



## 스레드 주머니

스레드 풀 내의 유휴 스레드가 요청시마다 출동해 서블릿코드 동작시킴.

미리 스레드를 생성하고 **꺼내쓰고 반납하고**를 반복한다. 정해진 수만큼만 스레드 동작이 가능해서

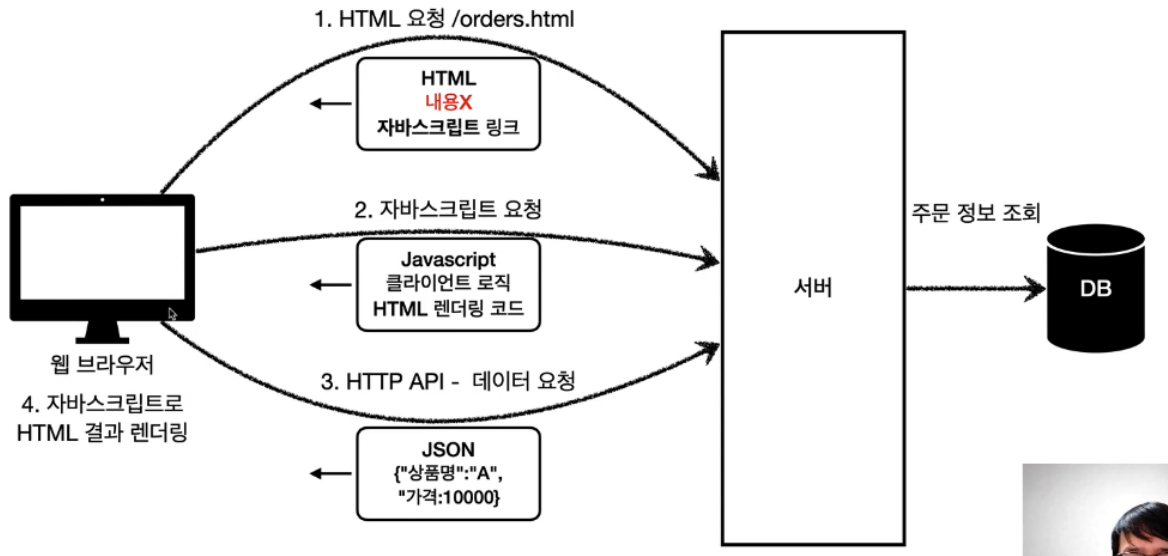
터질 위험이 줄어든다.

## SSR

- JSP
- HTML 최종 결과를 서버에서 만들어 클라이언트에게 전달
- 정적 화면에 사용

## CSR - react,vue

# CSR - 클라이언트 사이드 렌더링



- React,Vue
- 복잡하고 동적인 UI 사용
- 동적인 화면을 잘 처리

## 자바 최신 웹기술

- 웹 서블릿 - Spring MVC
- 웹 리액티브 - Spring WebFlux

## WebFlux

### 특징

- 비동기 논블로킹
- 최소 스레드 최대 성능 - 컨텍스트 스위칭 비용 효율화
- 함수형 스타일 개발 - 동시처리 코드 효율화
- 서블릿 기술 사용 안함

### 그런데

- 기술난이도 매우 높음
- 관계형 DB 지원 부족

- 실무 활용 아직 많이 안함

## 자바 뷰템플릿

- JSP
  - 느리고 기능 부족
- Freemarker, Velocity
- Thymeleaf
  - HTML 모양 유지하며 뷰 템플릿 적용 가능
  - 스프링MVC 와의 강력한 기능 통합
  - 최선의 선택. 성능은 위 2개가 더 빠르다