웹 애플리케이션을 개발할 때 대부분 스프링 MVC를 사용한다.

스프링 MVC는 복잡한 웹 애플리케이션을 쉽고 빠르게 개발할 수 있게 도와주는 실무에서 가장 많이 사용하는 백엔드 프레임 워크다.

자바 웹 기술을 학습하기 어려운 이유

1. HTTP - 모든 웹 기술은 HTTP 를 기본으로 하고 있음. HTTP를 모르면 백엔드를 이해하기 어려움.

2. 자바 백 엔드 웹 개발 역사가 너무 오래됨. -> 많은 것이 자동화되고 추상화됨.

왜 이런 방식으로 기능이 제공되는지 모르고, 사용방법만 알고 있음. 문맥을 알고 이해하기 사용하기 어려움.

3. 스프링 MVC

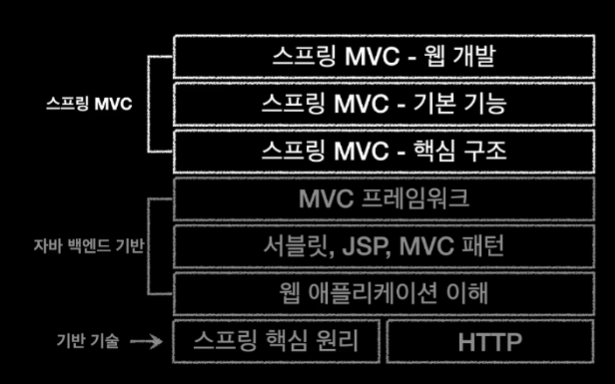
백엔드 개발에 필요한 거의 모든 실무 기술을 제공함. : 방대하고 학습할 것이 많음.

스프링 MVC의 기본 구조가 탄탄하게 되어있어서, 신규 기능이 추가되어도, 기본 구조는 바뀌지 않음.

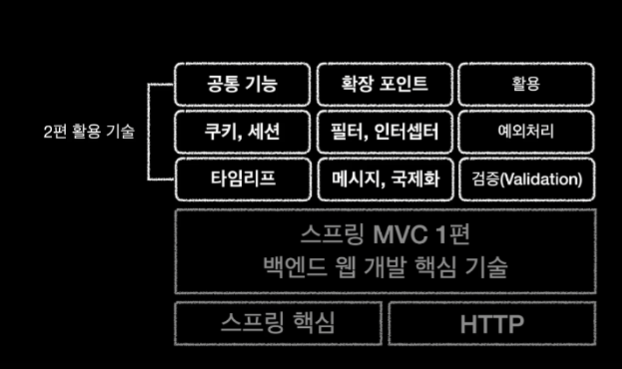
스프링 없이 스프링 MVC를 하나하나 만드는 방식으로 이해해볼 것이다.

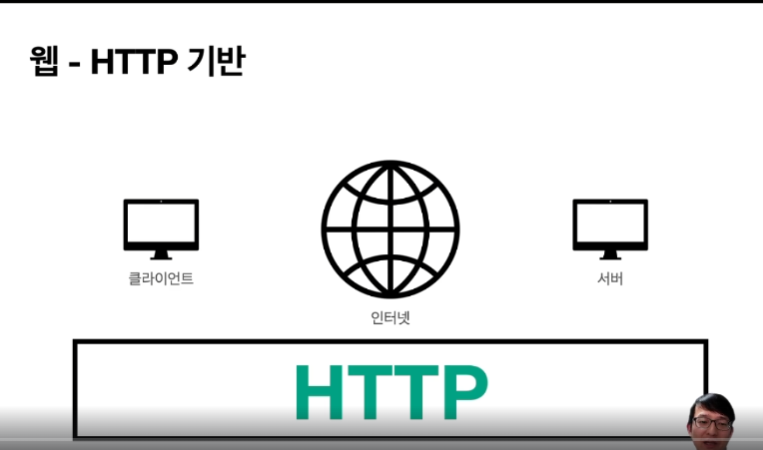
MVC 프레임워크를 새로 만들어볼 것임. -> 이 단계를 통해서 스프링 MVC의 내부구조를 이해하고 왜 이런 기술을 사용하게 되는지 알게 될 것이다.

**스프링 MVC 1편 소개** (백엔드 웹 개발 **핵심** 기술)



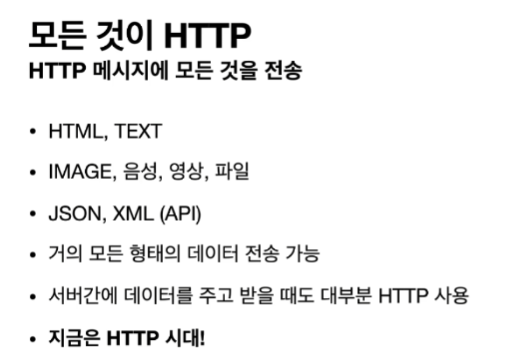
스프링 MVC 2편 (백엔드 웹 개발 **활용** 기술)

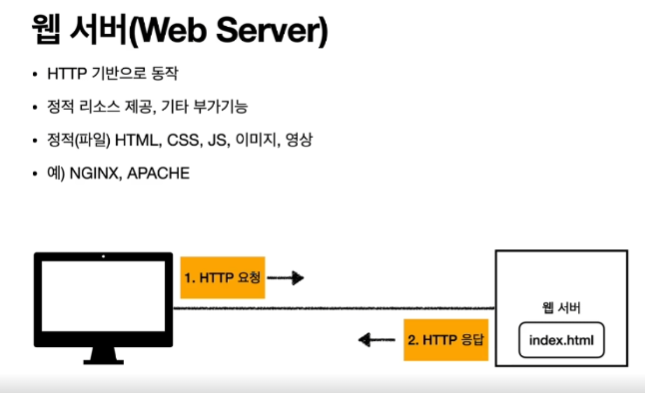


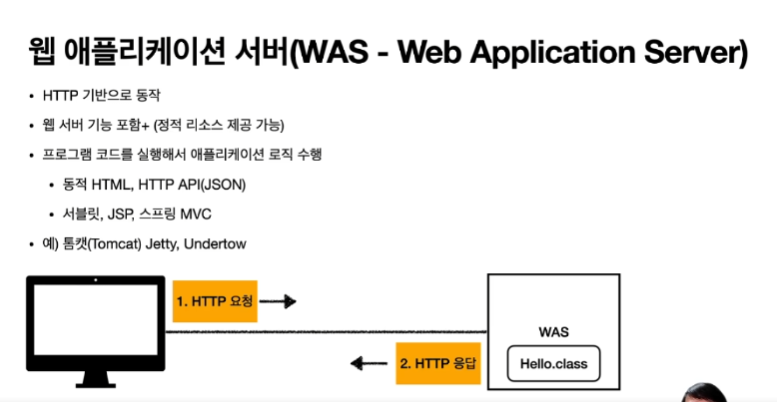


클라이언트가 웹브라우저에서 url을 치면 -> 인터넷을 통해 서버에 접근 -> 서버에서 html을 만들어서 클라이언트에 줌. 웹브라우저가 이를 사용자에게 보여줌.

클라이언트에서 서버로 데이터를 전송하고 서버에서 클라이언트로 응답할 때 HTTP 프로토콜 기반으로 동작하게 된다.



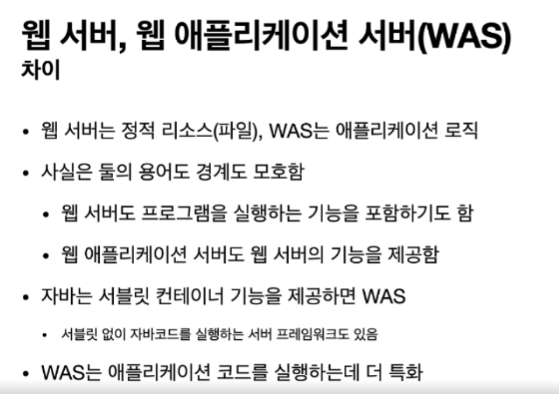


서버에 정적 파일들을 특정 디렉토리에 보관해두고 있다가 HTTP 요청이 들어오면, 이를 전송해줌.

정적 파일 같은 경우에 웹서버가 이를 수정하여 사용자마다 다르게 보여주는 작업을 하지 못함.

WAS는 HTTP요청이 오면 사용자에 따라 프로그램 코드를 실행해서 사용자마다 다른 화면을 보여줄 수 있음.

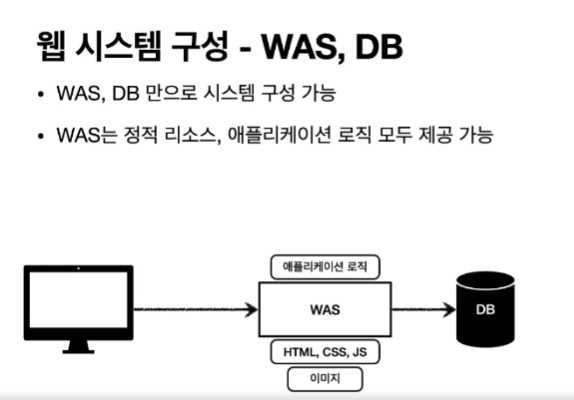
서블릿, jsp, 스프링 MVC같은 것들이 WAS에서 동작하게 된다.



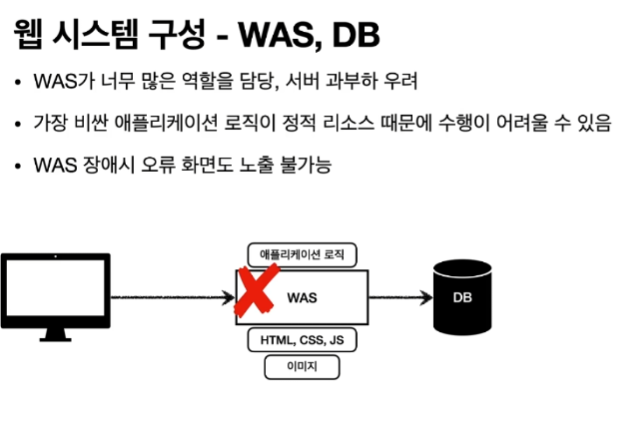
HTTP 기반에 데이터를 주고받는 웹 서버라고 포괄적으로 볼 수도 있고 용어의 경계도 모호하긴 하다. 웹서버에 플러그인을 설치해서 프로그램을 실행하는 기능을 포함하기도 함.

웹 애플리케이션 서버도 웹 서버의 기능을 제공한다.

자바 같은 경우에 보통 서블릿 컨테이너 기능을 제공하면 WAS 라고 본다. 하지만 서블릿 없이 자바코드를 실행하는 서버 프레임워크도 있음.



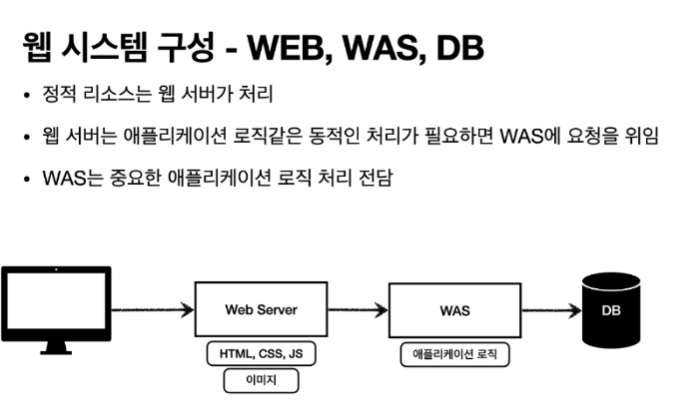
WAS, DB만으로 최소한의 구성 요소로 웹 시스템 구성이 가능하다.



WAS 하나만 가지고 운영하면 WAS 가 너무 많은 것을 제공함.

정적 요소는 파일 하나 두고, 잘 서빙하면 되기 때문에 단순하다.

하지만 애플리케이션 로직은 비싸다. 로직을 실행해서 전달해야함. db에 접근하거나 다른 서버와 통신하거나 하기도 함.



업무 분담이 가능.

