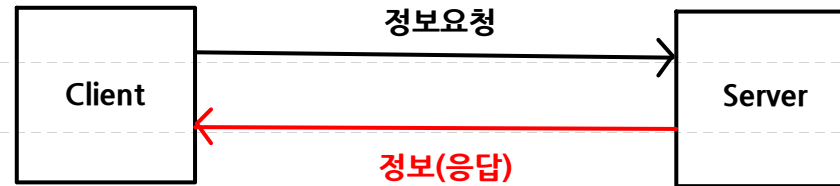


## 백엔드 개발

1. 개발해야 하는 대상 --> 개발 하고자하는 목표 시스템
2. 개발 대상은 어떤 특징을 가지는가?
3. 개발 대상을 어떻게 하면 편하고 안전하게 개발할까? -> 개발 방법

### 1. 백엔드 목표시스템

- 정보의 생산과 저장이 한 곳에 몰리게 된다.



### 2. 백엔드 시스템의 특징

- 클라이언트는 지속해서 증가
  - 1> 단위시간에 정보요청의 개수가 늘어난다. --> 서버의 부화 + 클라이언트의 속도 저하
  - 2> 저장하는 정보가 매우 복잡하고 커진다. --> 디스크 용량 커지고 스키마 복잡해짐
- 서버 보안
  - 1> 법률 준수
  - 2> 비인가된 방법에서의 접근 차단
- 비즈니스 니즈 충족
  - 1> 보편적 기준의 품질을 보장

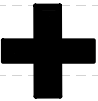
3. 개발 이슈

1> 안정성 있게 만들어야 함 (안전, 보안 니즈 충족)

2> 서버 부하 분산

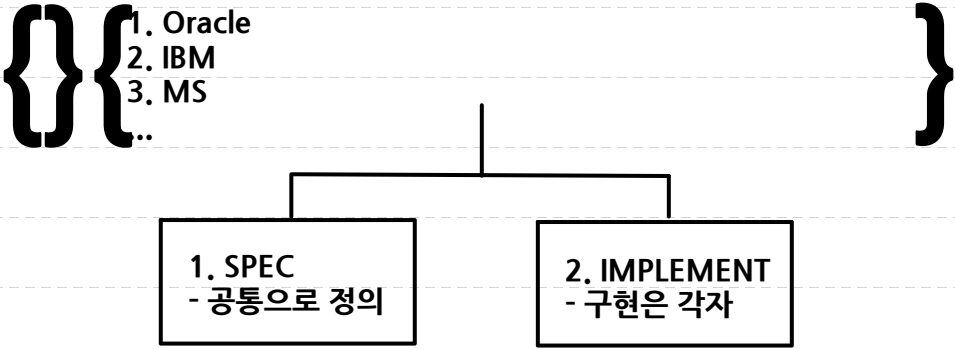
3> 복잡한 데이터베이스를 다루는 기술 필요

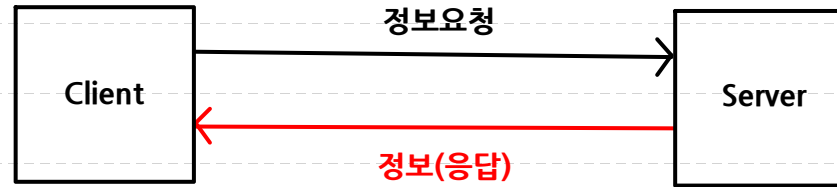
4> 빠른 응답속도



비즈니스 로직

비즈니스 로직 만들기가 까다롭기 때문에  
개발 이슈를 만들어서 판매하기 위해 시장에 뛰어든 회사들





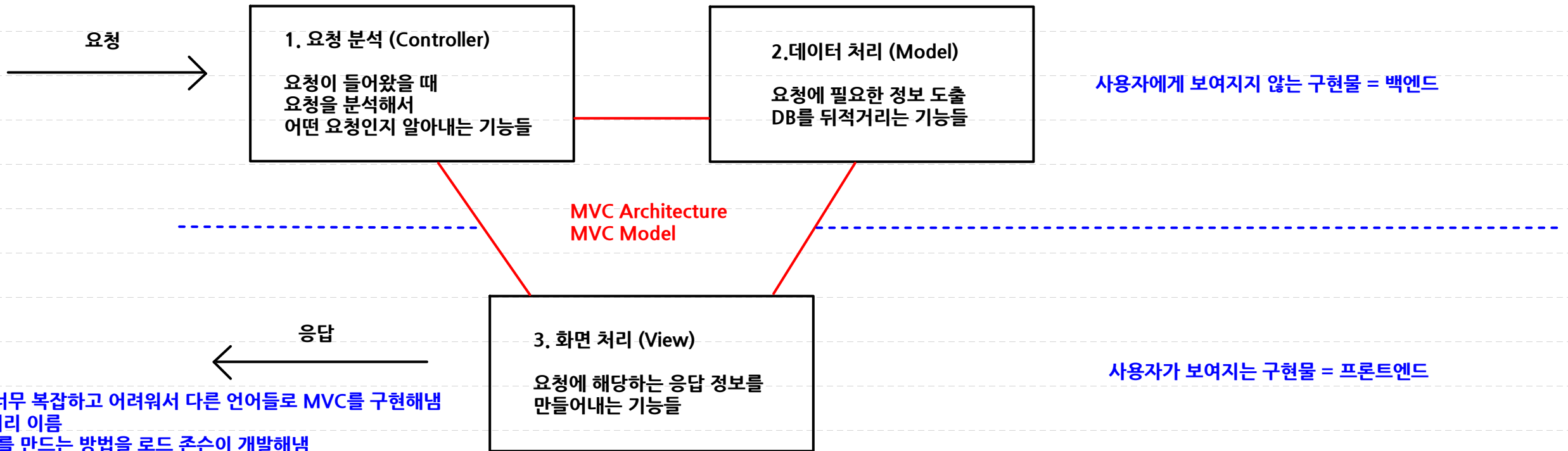
<SPEC>

1. 정보 송/수신을 위한 프로토콜 정의 --> HTTP
  2. 서버가 클라이언트에게 주는 정보 --> HTML
- 즉, 백엔드는 HTML(정보)이 핵심

3. 서버에 필요한 기능들을 어떤 언어로 만들까?
    - JAVA (안정적인 DB 구축)
    - JavaEE (Enterprise Edition)라는 SPEC 발표 --> JCP (Java Community Process)
- (1) Oracle -- Weblogic
  - (2) IBM -- Websphere
  - (3) OpenSource -- Apache Tomcat

>> Servlet

Sun Microsystems : JavaEE를 만족시키는 동시에 편리한 개발을 위해 발표



J2EE 운용이 너무 복잡하고 어려워서 다른 언어들로 MVC를 구현해냄

EJB : 라이브러리 이름

EJB없이 J2EE를 만드는 방법을 로드 존슨이 개발해냄

그래서 사람들은 EJB와 J2EE를 버리게 된다

로드 존슨이 스프링 프레임워크를 개발함

스프링 프레임워크 = MVC 구조를 구현해주는 프레임워크

스프링 프레임 워크는 '틀'을 만들어주는 것이기 때문에 각각의 순서에서 무슨 기능을 사용해도 상관없음

다만, 스프링이 지원/추천해주는 형식의 라이브러리를 사용하면 됨 (타사에서 만든 라이브러리여도 상관없음)

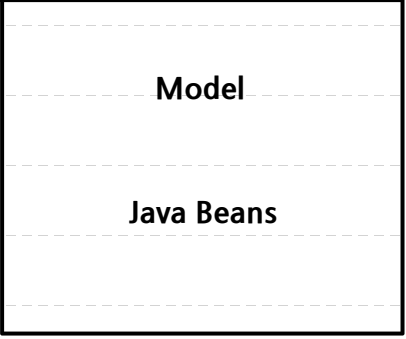
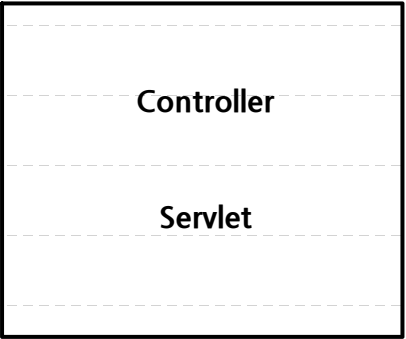
JSP와 Servlet을 해보기

Springframework가 발전한 형태의 Springboot ==> JSP 사용을 권장하지 않음. SB는 프론트/백을 완전히 분리시키는데

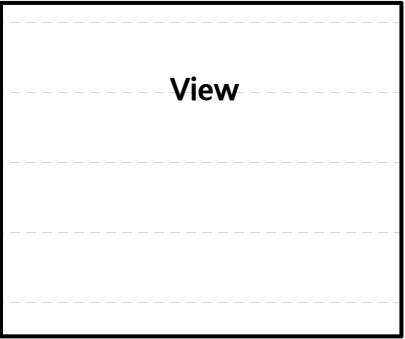
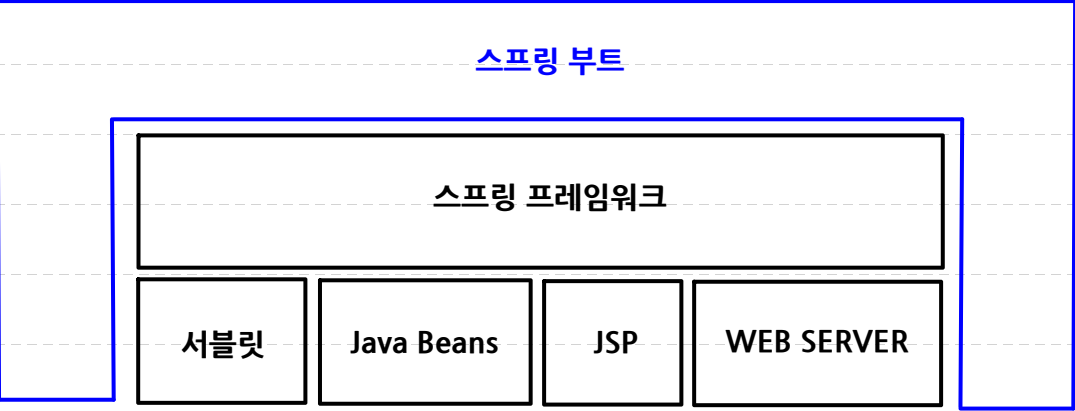
JSP를 사용하게 되면 이것(M-V-C)이 모두 혼동될 수 있기 때문. 하지만 현실에서는 JSP 없이 웹 개발이 어렵기 때문에

JSP와 Springboot를 믹스해서 개발해볼 것 -> 원래 잘 하지 않는 방식 but 필드에 나가보면 이런 코드가 굉장히 많음

요즘 새로 나온 타임리프도 한 번 해볼 것



- 스프링부트란?
- 스프링 프레임워크를 편하게 사용하기 위한 상위 프레임워크



```
<dependencies>
1  <dependency>           공급자
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>         다운받은 라이브러리의 본체(이름)
2  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
  </dependency>
3  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-tomcat</artifactId>
    <scope>provided</scope>
  </dependency>
4  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```