멋쟁이 사자처럼
백엔드 세션 1주차

1. 스프링과 스프링부트

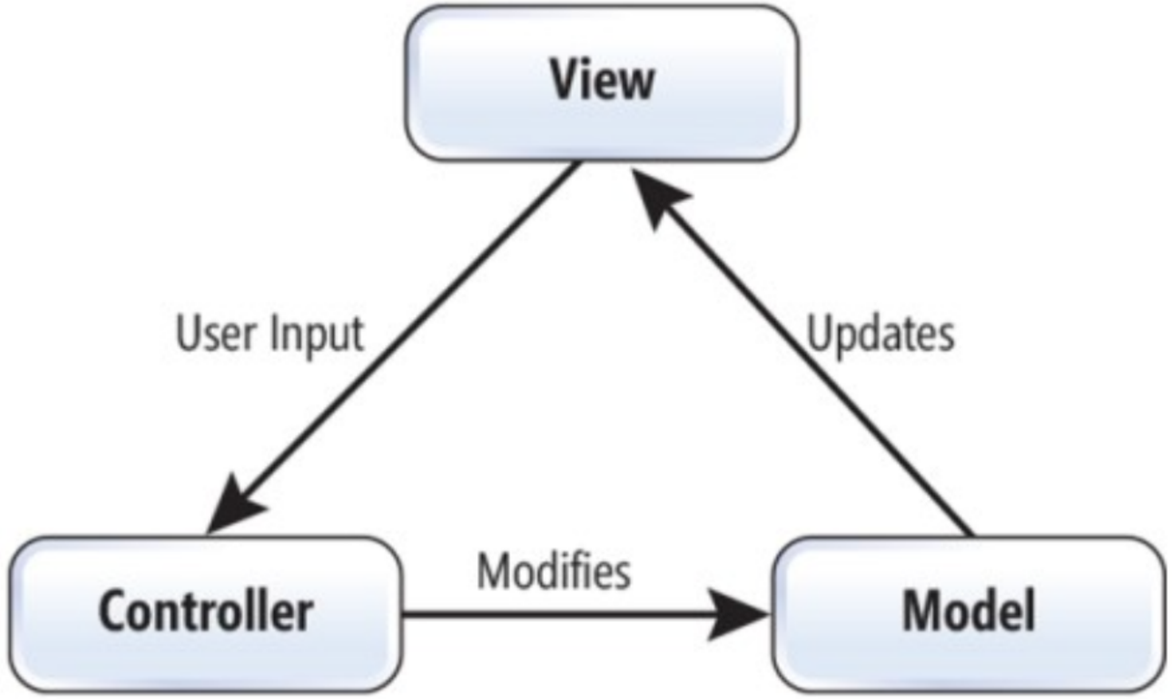
Java 기반 애플리케이션 개발을 지원하는 오픈소스 애플리케이션 프레임워크

많은 문제점 존재
- 설정의 복잡성
- 높은 초기 학습 난이도
- 의존성 관리 문제
- 별도 서버 구축 필요

더 쉽고 빠르게 스프링 애플리케이션을 개발하도록 도와주기 위해 개발

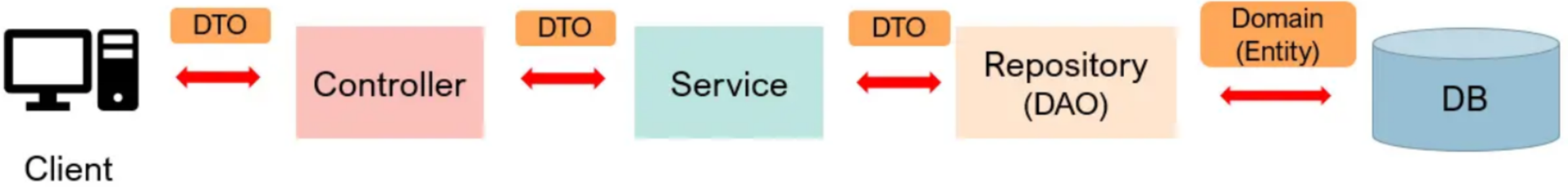
스프링의 모든 문제점 해결
- 간결한 설정
- 내장 서버(Tomcat..)
- 의존성 관리 간소화

2. MVC 모델



Model : 데이터를 조회했을 때 결과를 담을 수 있는 저장공간이 모여있는 클래스
View : 화면
Controller : 데이터베이스에 접근하는 메서드들이 모여있는 곳
스프링부트는 개발 영역을 MVC로 구분하여 각 역할에 맞게 코드를 작성.

3. 패키지 구조


DTO
Controller
DTO
Service
DTO
Repository
(DAO)
Domain
(Entity)
DB
Client

3. 패키지 구조

Controller

Client의 요청을 DTO의 형태로 받아 Service의 기능을 호출하고, 적절한 응답을 DTO의 형태로 반환

Service

DTO를 통해 받은 데이터를 이용해 비즈니스 로직을 처리하고, DAO를 통해 DB에 접근하여 데이터를 관리

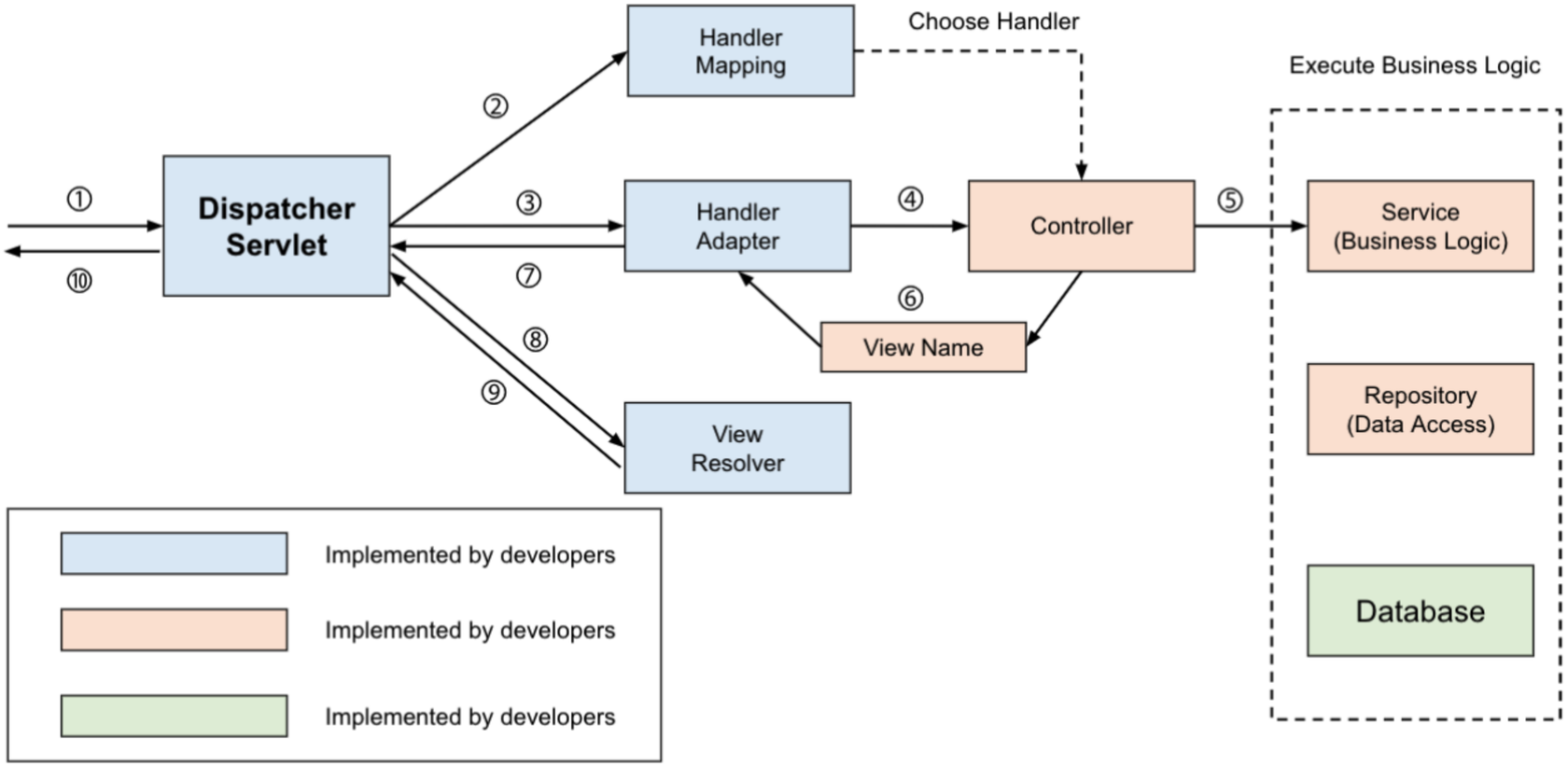
DTO(Data Access Object)

의 약자로 DB의 data에 접근하기 위한 객체

Repository/ DAO(Data Transfer Object)

데이터를 Transfer(이동)하기 위한 객체

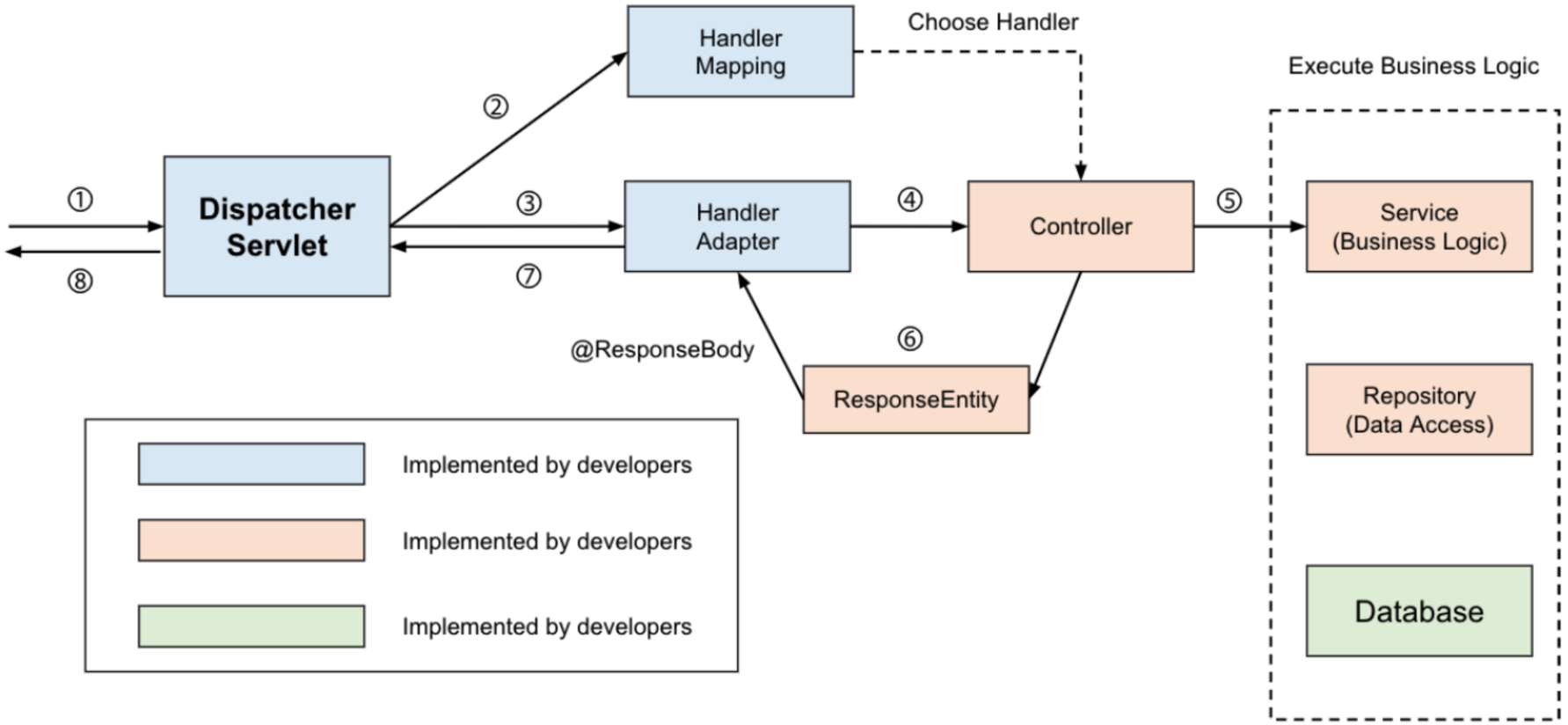
4. @Controller



전통적인 Spring MVC의 컨트롤러 어노테이션인 @Controller는 주로 View(화면)를 반환하기 위해 사용

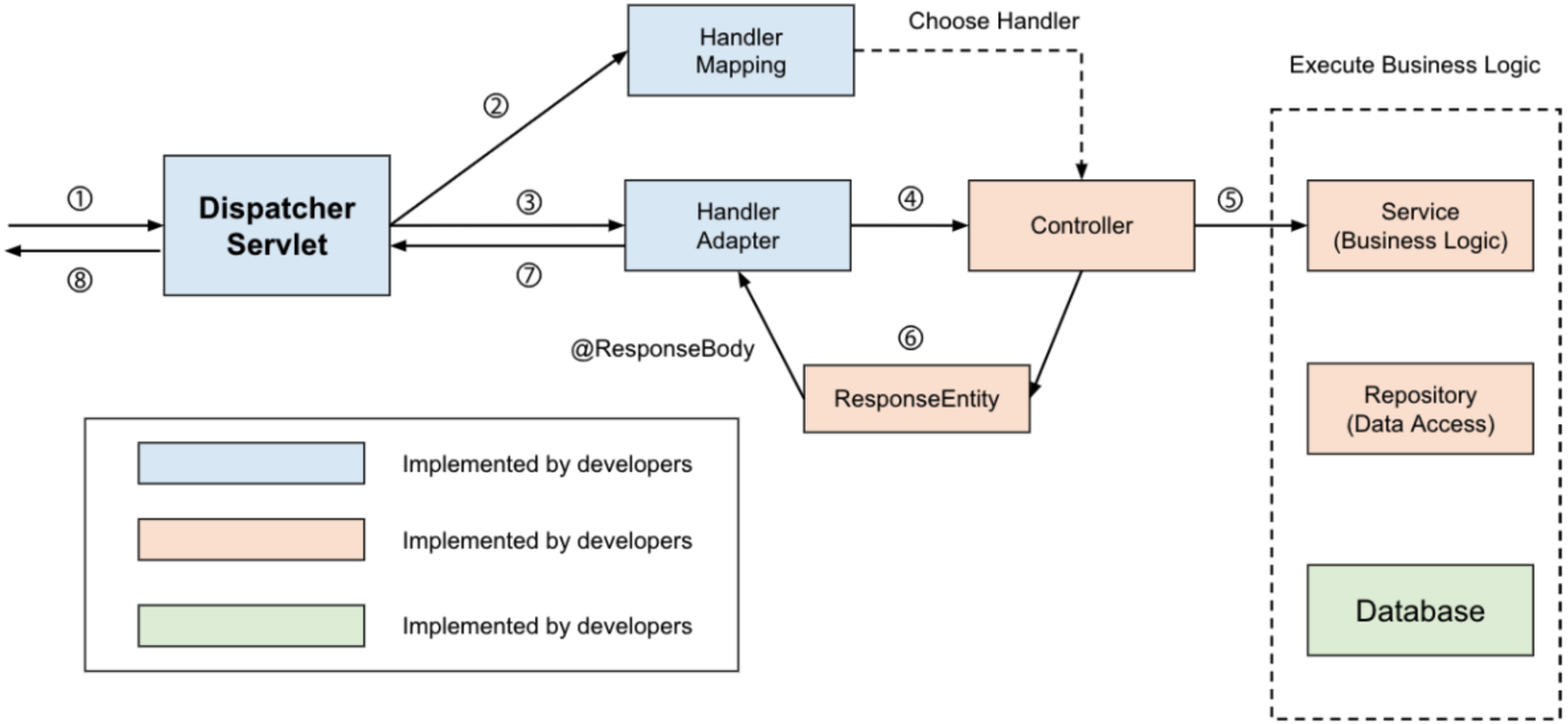
요약해서 요청이 오면 그에 맞는 화면을 반환해준다.

5. @Controller + @ResponseBody



Spring MVC의 컨트롤러를 사용할 때 Data를 반환해야 하는 경우 존재.
@ResponseBody 어노테이션을 활용해서 JSON 형태의 데이터를 반환할 수 있다.

6. @RestController



@Controller + @ResponseBody가 합쳐진 형태로 JSON 형태의 객체 데이터를 반환

과거 JSP, HTML과 같은 View를 전달해 주었기에 주로 @Controller를 사용.

현재 프론트와 백을 나누어 개발하는 경우가 많고 백엔드에서 Rest API를 통해
JSON으로 데이터만 전달하기 때문에 @RestContoller 사용.