

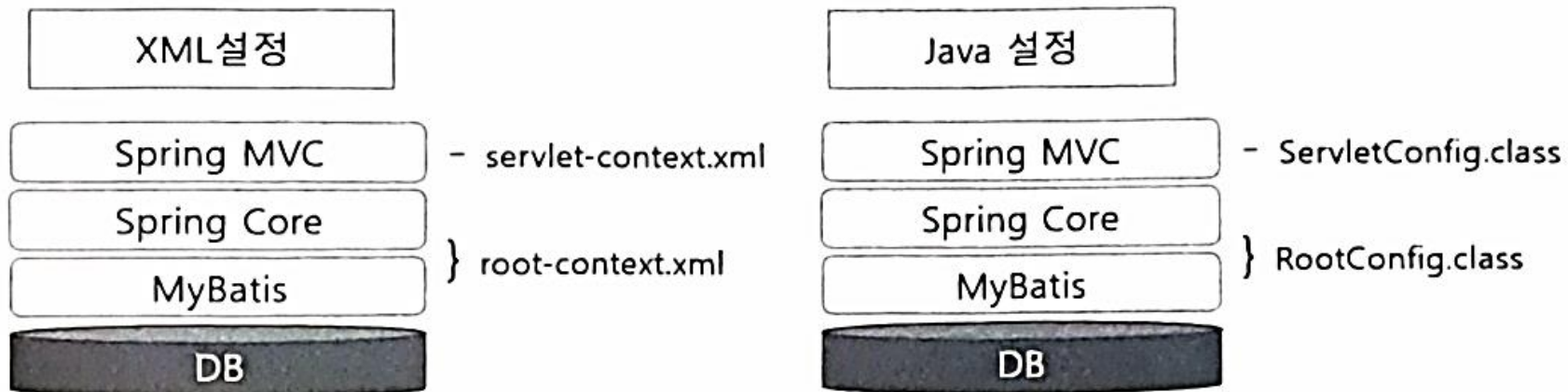
2024년 상반기 K-디지털 트레이닝

스프링 MVC의 기본 구조

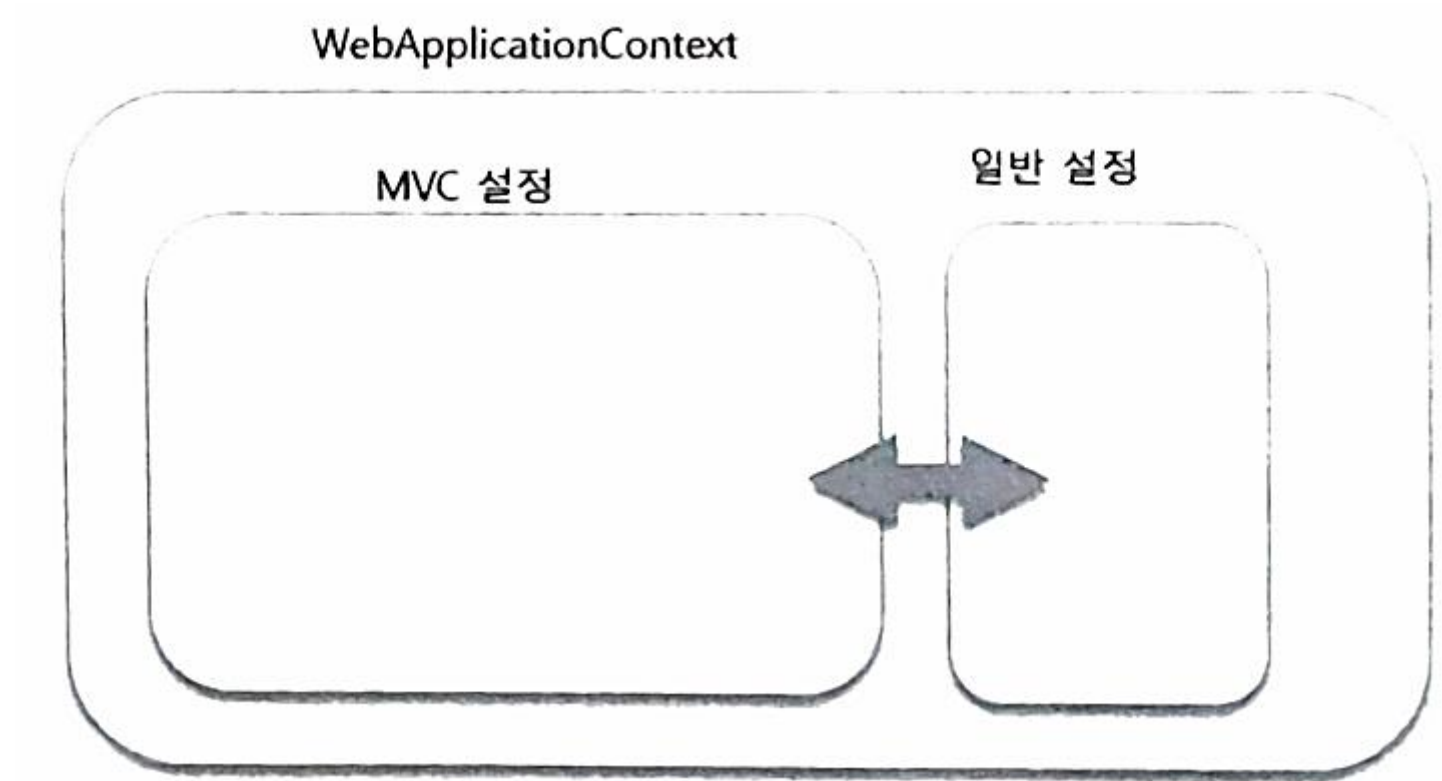
[KB] IT's Your Life

1 스프링 MVC 프로젝트의 내부 구조

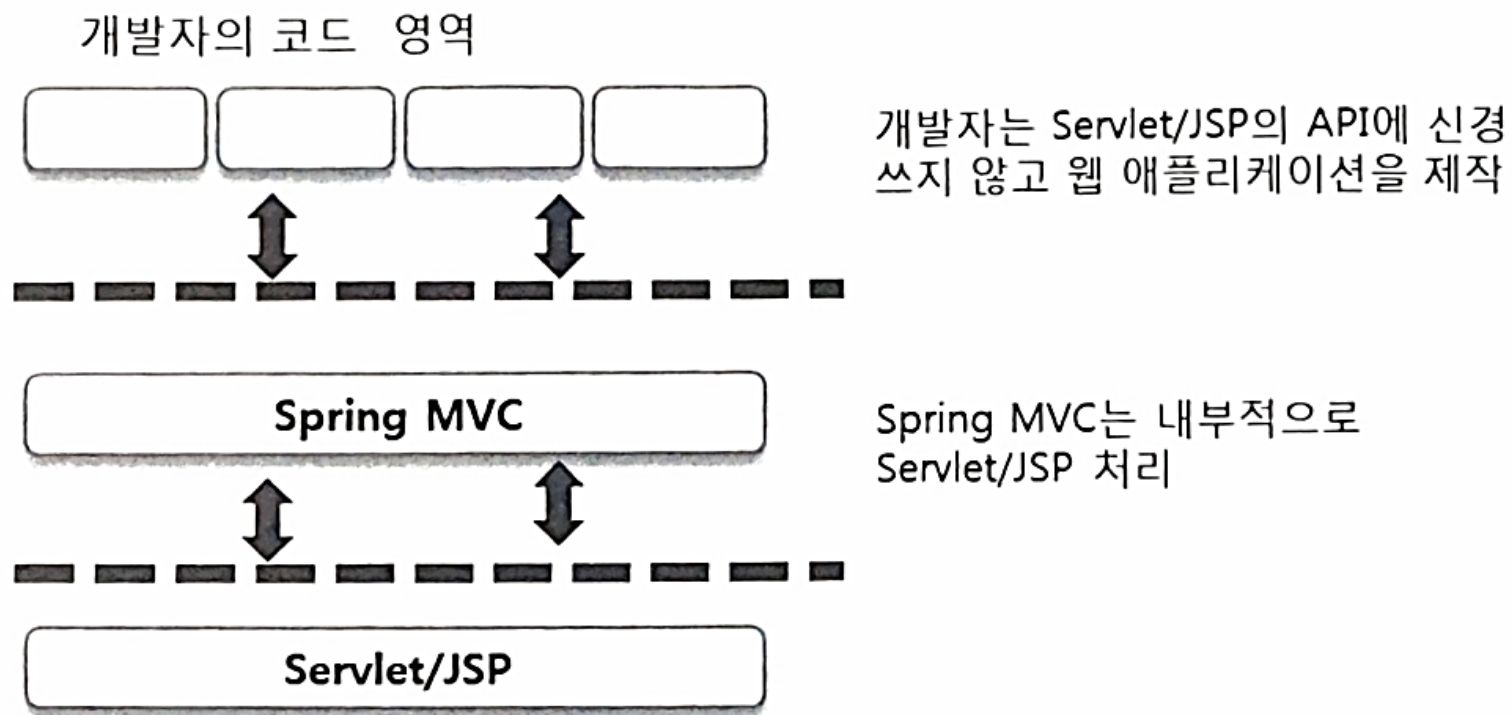
✓ 프로젝트 구조



1 스프링 MVC 프로젝트의 내부 구조

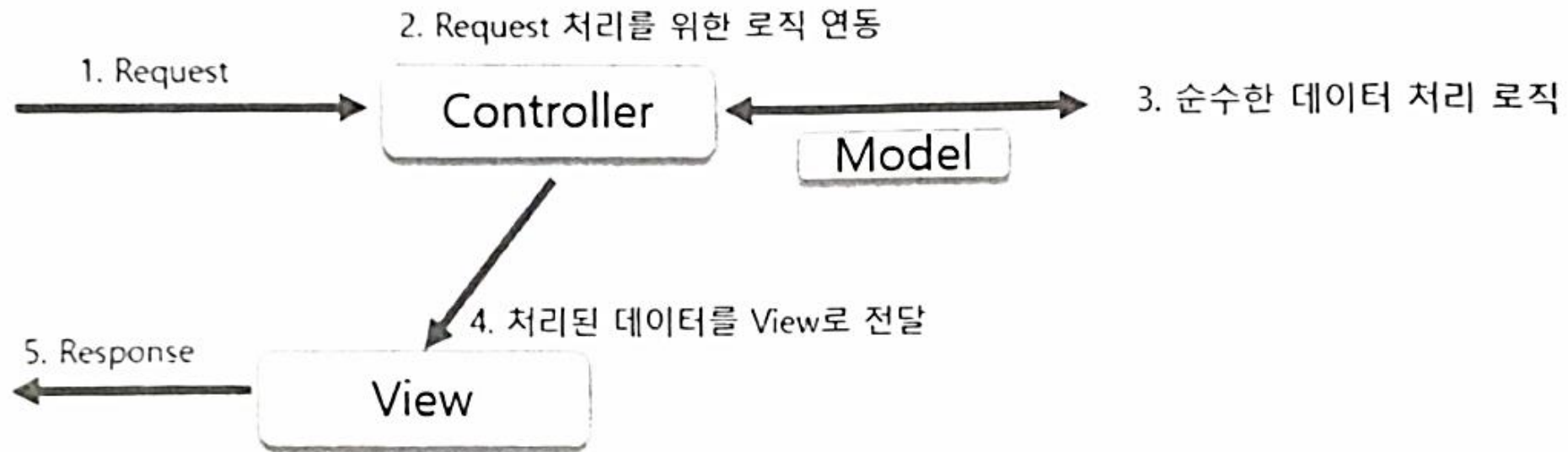


2 스프링 MVC의 기본 사상



2 모델 2와 스프링 MVC

✓ 모델 2



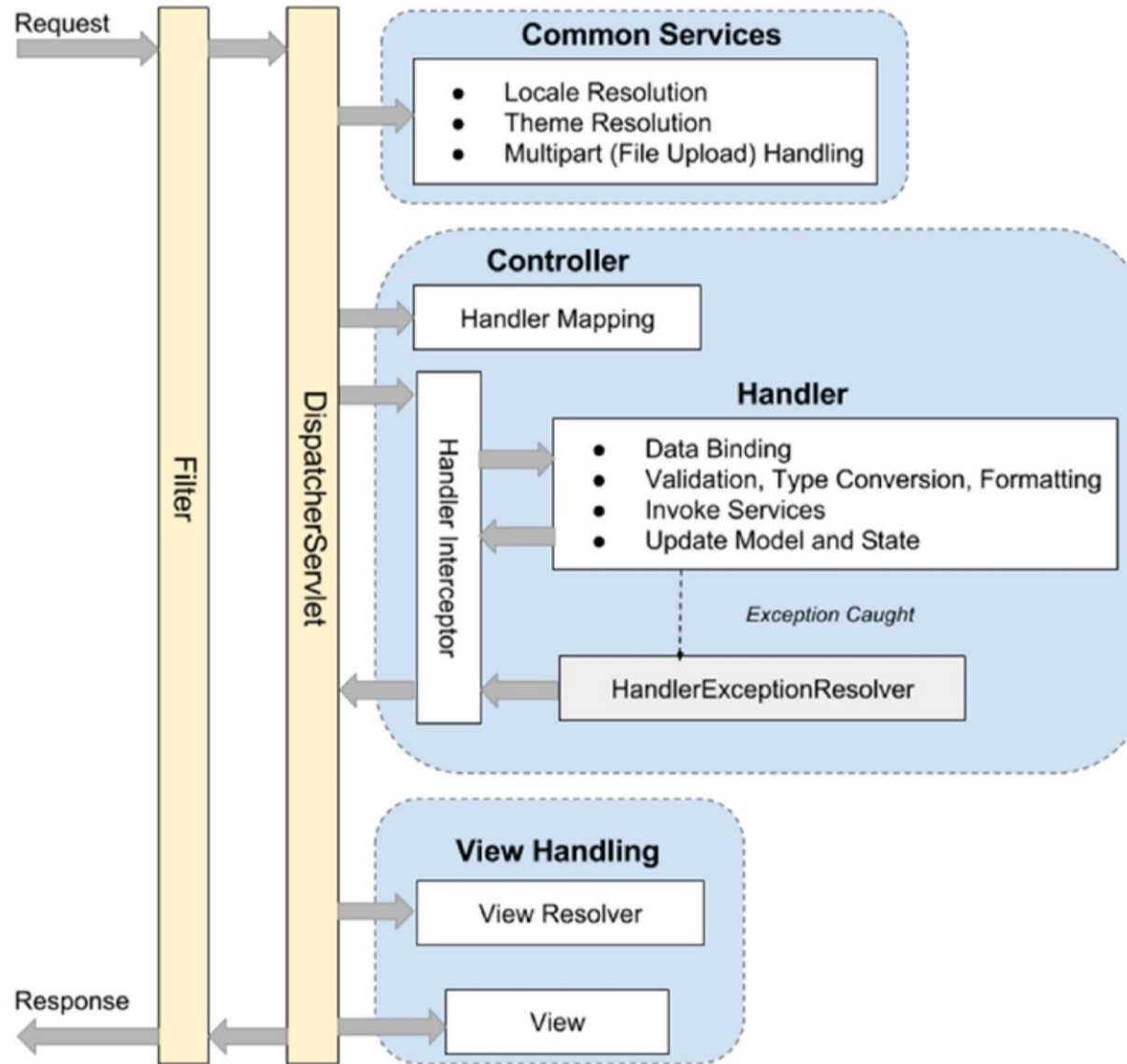
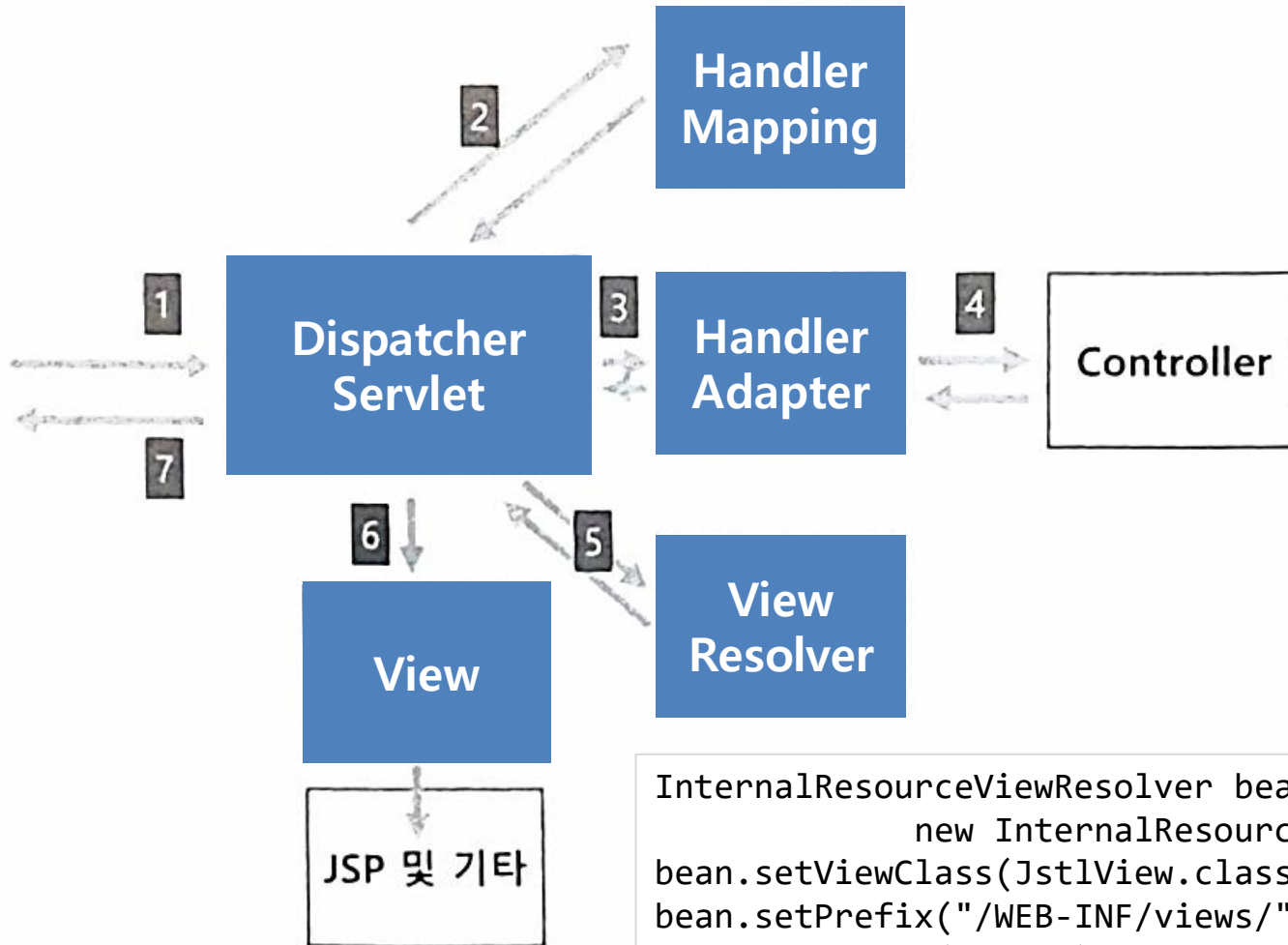


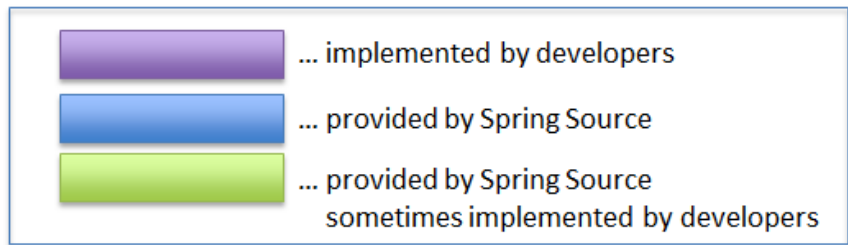
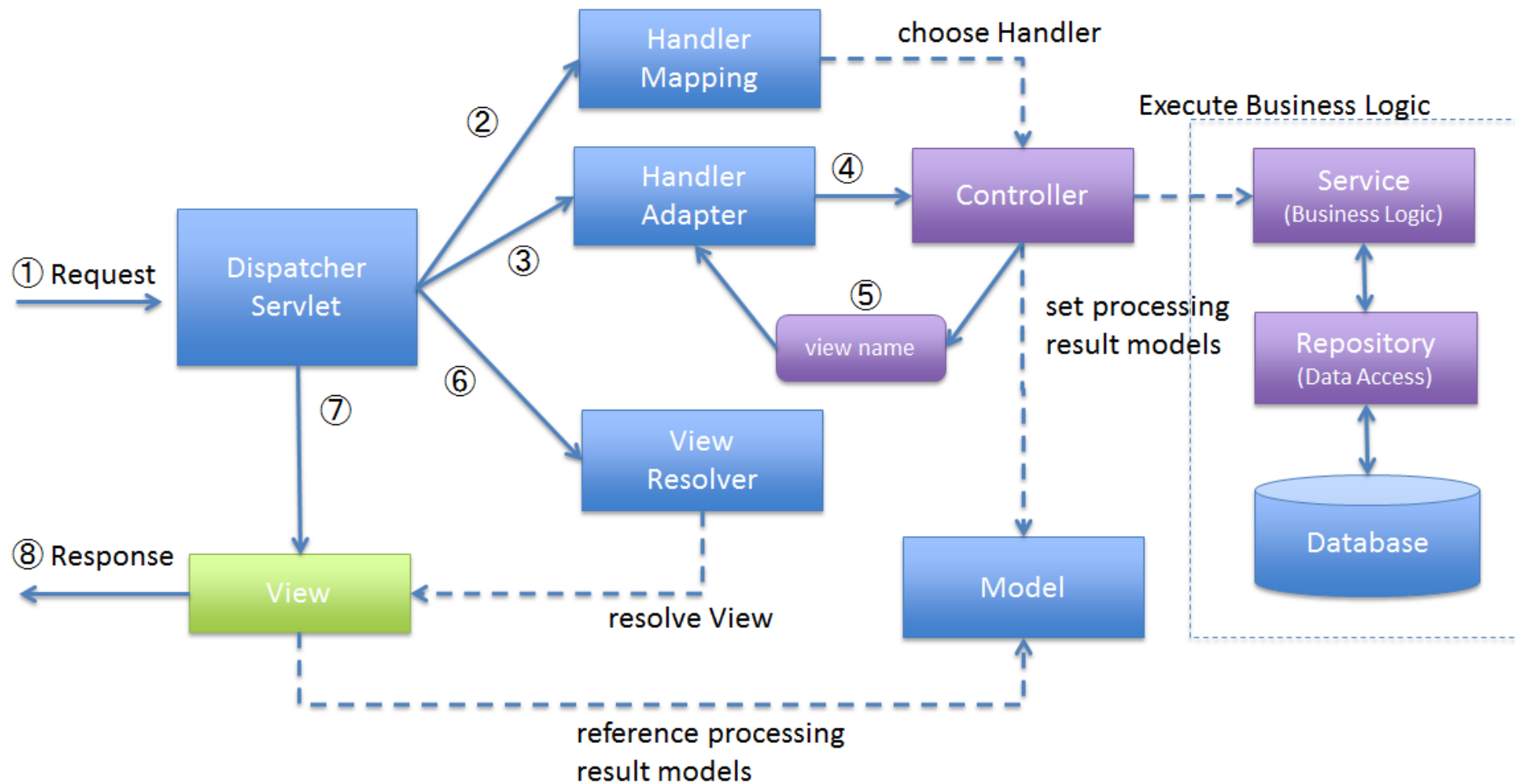
Figure 16-3. Spring MVC request life cycle

2 모델 2와 스프링 MVC

✓ 스프링 MVC



```
InternalResourceViewResolver bean =  
    new InternalResourceViewResolver();  
bean.setViewClass(JstlView.class);  
bean.setPrefix("/WEB-INF/views/");  
bean.setSuffix(".jsp");  
registry.viewResolver(bean);
```



✓ 처리 순서

- Filter → DispatcherServlet → HandlerMapping → HandlerInterceptor → Controller → Service → Repository(Mapper) → ViewResolver 순

✓ Filter

- Web Application의 전역적인 로직을 담당
- Filter라는 단어 뜻에서 알 수 있듯이, 전체적인 필터링(설정)을 하는 곳
- DispatcherServlet에 들어가기 전인 Web Application단에서 실행

✓ DispatcherServlet

- 들어오는 모든 Request를 우선적으로 받아 처리해주는 서블릿
- HandlerMapping에게 Request에 대해 매핑할 Controller 검색을 요청
- HandlerMapping으로부터 Controller 정보를 반환받아 해당 Controller와 매핑
→ Request에 대해 어느 컨트롤러로 매핑시킬 것인지 배치하는 역할

✓ HandlerMapping

- DispatcherServlet으로부터 검색을 요청받은 Controller를 찾아 정보를 리턴

✓ HandlerInterceptor

- Request가 Controller에 매핑되기 전 앞단에서 추가적인 로직을 추가
- 주로 세션, 쿠키, 권한 인증 로직에 많이 사용됩니다.

✓ Controller

- Request와 매핑되는 곳
- Request에 대해 어떤 로직(Service)으로 처리할 것인지를 결정
- 그에 맞는 Service를 호출
- Service Bean을 스프링 컨테이너로부터 주입

✓ Service

- 데이터 처리 및 가공을 위한 비즈니스 로직을 수행
- Request에 대한 실질적인 로직을 수행
- Repository를 통해 DB에 접근하여 데이터의 CRUD(Create, Read, Update, Delete)를 처리

✓ Repository (DAO, Data Access Object)

- DB에 접근하는 객체. 라고 부릅니다.
- Service에서 DB에 접근할 수 있게 하여 데이터의 CRUD 처리

✓ ViewResolver

- Controller에서 리턴한 View의 이름을 DispatcherServlet으로부터 넘겨받고,
- 해당 View로 forward.

→ 이후, 해당 렌더링된 View 화면은 브라우저로 전송(response)