**02. 개발에 앞서 알면 좋은 기초 지식**

**1. 서버 간 통신**

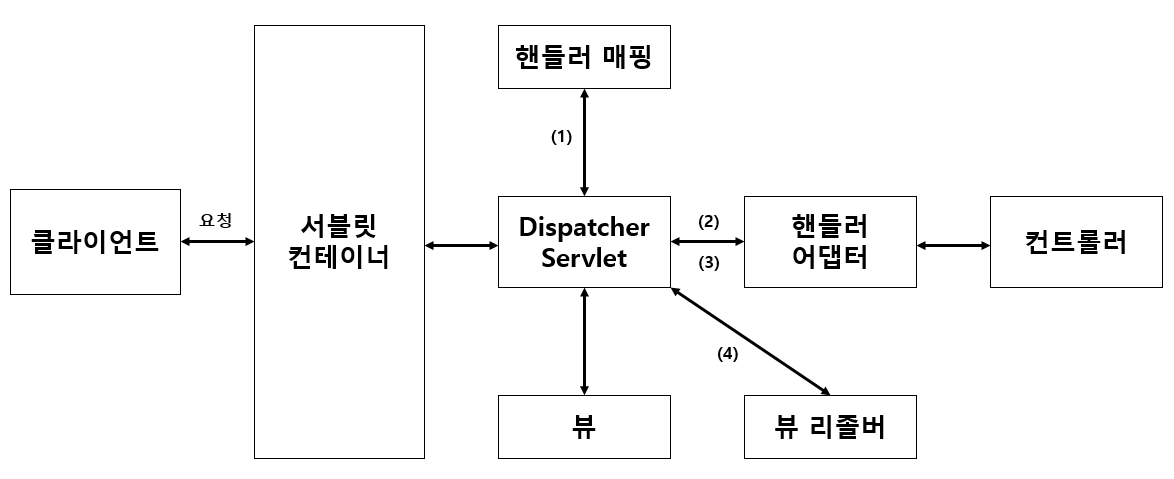
**①** 단일 서비스 아키텍처

: 한 어플리케이션에 모든 기능을 하나로 통합한 구조

**②** 마이크로서비스 아키텍처

: 서비스를 쪼개어 한 어플리케이션을 구현하는 구조

**2. 스프링 부트의 동작 방식**

** ①** 서블릿 컨테이너(Servlet Container)

: 클라이언트의 요청을 처리하고 결과를 반환하는 자바 웹 프로그래밍 기술인 서블릿을 관리 하는 클래스

**특징** – 서블릿 객체를 생성, 초기화, 호출, 종료하는 생명주기 관리

– 서블릿 객체는 싱글톤으로 관리됨

– 멀티 쓰레딩 지원

**②** Dispatcher Servlet

(1) 요청이 들어오면 핸들러 매핑을 이용해 요청 URI에 매핑된 핸들러 검색

(2) 핸들러 어댑터로 컨트롤러 호출

(3) 핸들러 어댑터에 컨트롤러 응답이 들어오면 ModelAndView로 응답을 가공해 반환

(4) 뷰 형식의 컨트롤러 이용 시 뷰 리졸버를 통해 뷰를 받아 리턴

**③** 핸들러 매핑: 요청 정보를 기준으로 컨트롤러를 선택하는 인터페이스

1) BeanNameUrlHandlerMapping: 빈 이름을 URL로 사용

- 슬래시가 들어가면 매핑 대상 ex) @Bean(“/hi”)

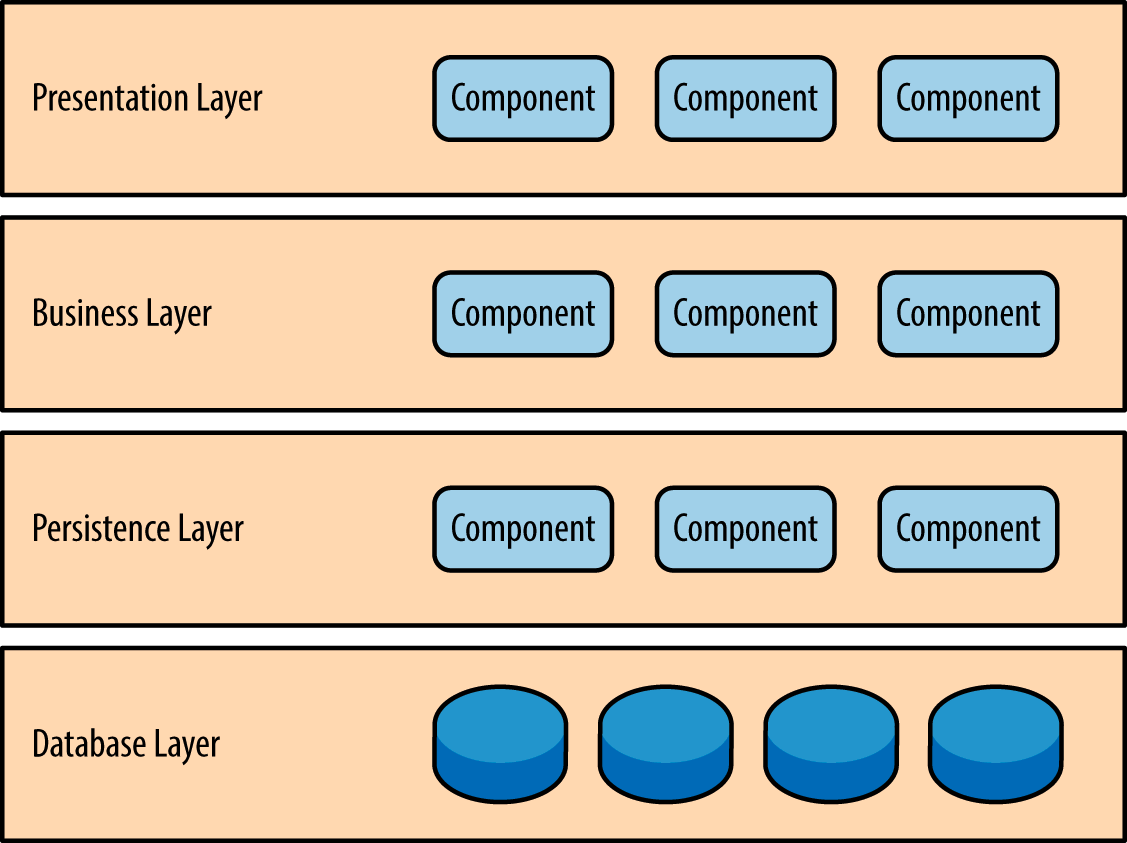
2) ControllerClassNameHandlerMapping

: URL과 일치하는 클래스 이름을 갖는 빈을 컨트롤러로 사용

3) SimpleUrlHandlerMapping: URL 패턴에 매핑된 컨트롤러 사용

4) DefaultAnnotationHandlerMapping: 어노테이션으로 URL과 컨트롤러 매핑

**3. 레이어드 아키텍처**



① 정의: 앱의 컴포넌트를 유사 관심사를 기준으로 레이어드로 묶어 수평적으로 구성한 구조

② 계층 설명

1) 프레젠테이션: 클라이언트의 요청을 해석하고 응답. UI 제공

2) 비즈니스: 기능을 정의하고 수행하는 도메인 객체를 역할 수행

3) 데이터 접근: 유에 접근하는 일련의 작업 수행

③스프링의 레이어드 아키텍처

