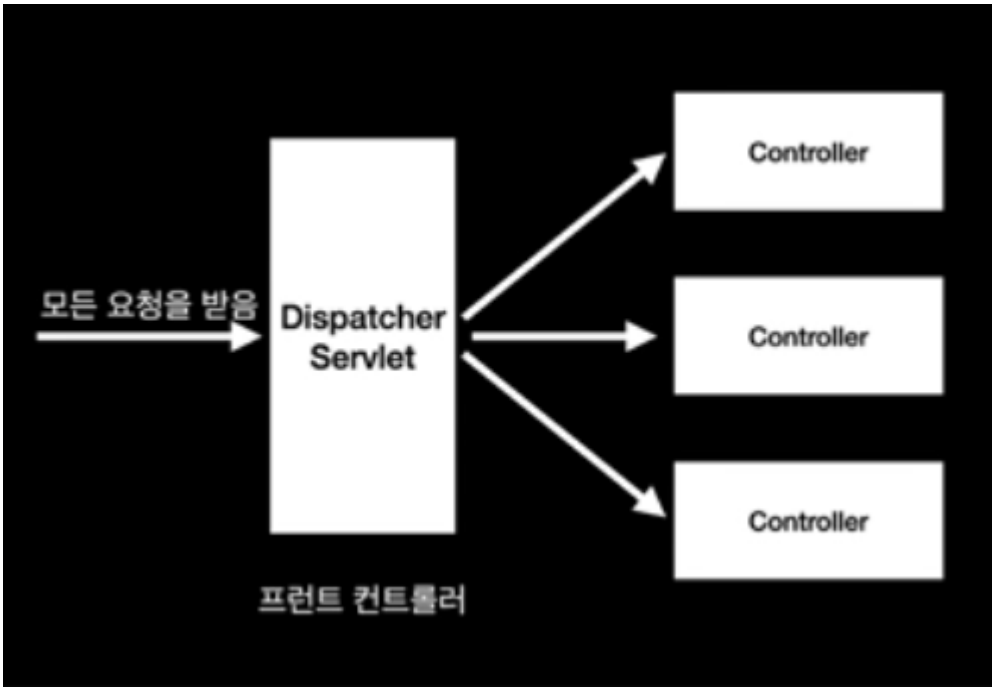
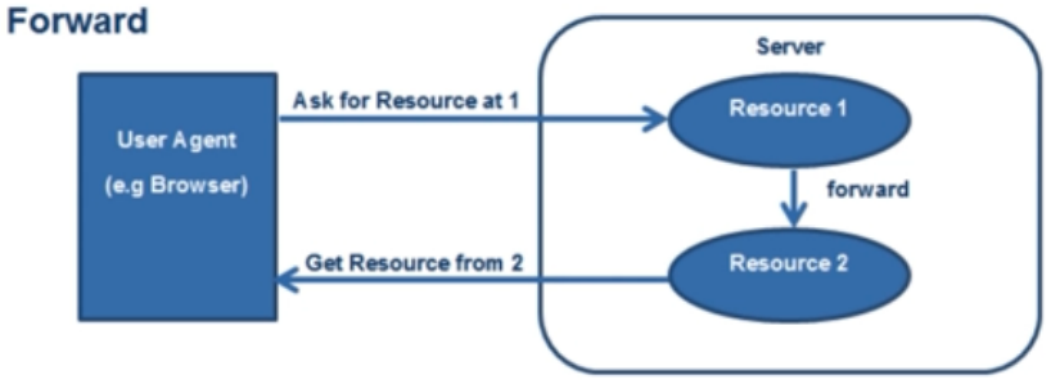
3 / 19 Reflaction, 프런트 컨트롤러 패턴

- Reflection
  - 힙 영역에 로드돼 있는 클래스 타입의 객체를 통해 필드/메소드/생성자를 접근 제어자와 상관없이 사용할 수 있도록 지원하는 API
  - 힙 영역에 로드돼 있는 클래스를 부르는 방법
    - Class.class
    - Instance.class
    - Class.forName
  - 컴파일 시점이 아닌 런타임 시점에 동적으로 특정 클래스의 정보를 추출해낼 수 있는 프로그래밍 기법
  - 주로 프레임워크 또는 라이브러리 개발 시 사용됨
    - Spring 프레임워크 (ex. DI)
    - Test 프레임워크 (ex. JUnit)
    - JSON Serialization / Deserialization 라이브러리 (ex. Jackson) 등
  - https://www.baeldung.com/reflections-library
- 프런트 컨트롤러 패턴
  - 모든 요청을 단일 handler에서 처리하도록 하는 패턴
  - 스프링 웹 MVC 프레임워크의 DispathcerServlet(프런트 컨트롤러 역할)이 프런트 컨트롤러 패턴으로 구현되어있음.



- Forward
  - 서블릿에서 클라이언트로(웹 브라우저)를 거치지 않고, 바로 다른 서블릿(또는 JSP)에게 용청하는 방식
  - Forward방식은 서버 내부에서 일어나는 요청이기 때문에 HttpSercletRequest와 HttpSercletResponse객체가 새롭게 생성되지않고, 공유한다.
  - RequestDispatcher dispatcher = request.getRequestDispatcher(”포워드 할 서블릿 또는 JSP”), dispatcher.forward(request, response)
- Redirect
  - 서블릿이 클라이언트(웹브라우저)를 다시 거쳐 다른 서블릿(또는 JSP)에게 요청하는 방식
  - Redirect 방식은 클라이언트로 부터 새로운 요청이기에 새로운 HttpSercletRequest와 HttpSercletResponse 객체가 생성된다. HttpSercletResponse 객체의 sendRedirect() 이용

Forward



SendRedirect