Servelt & JSP

hannkim

Servlet & JSP

- 자바에서 동적인 웹페이지 만들기 위한 기술
- CGI (프로세스 단위) -> Servlet, php, FastCGI 등 (스레드 단위) -> JSP
- 자바 기반의 데이터를 동적으로 멀티스레드 개념을 통해 처리하기 위함
    - Java Thread 이용하여 동작

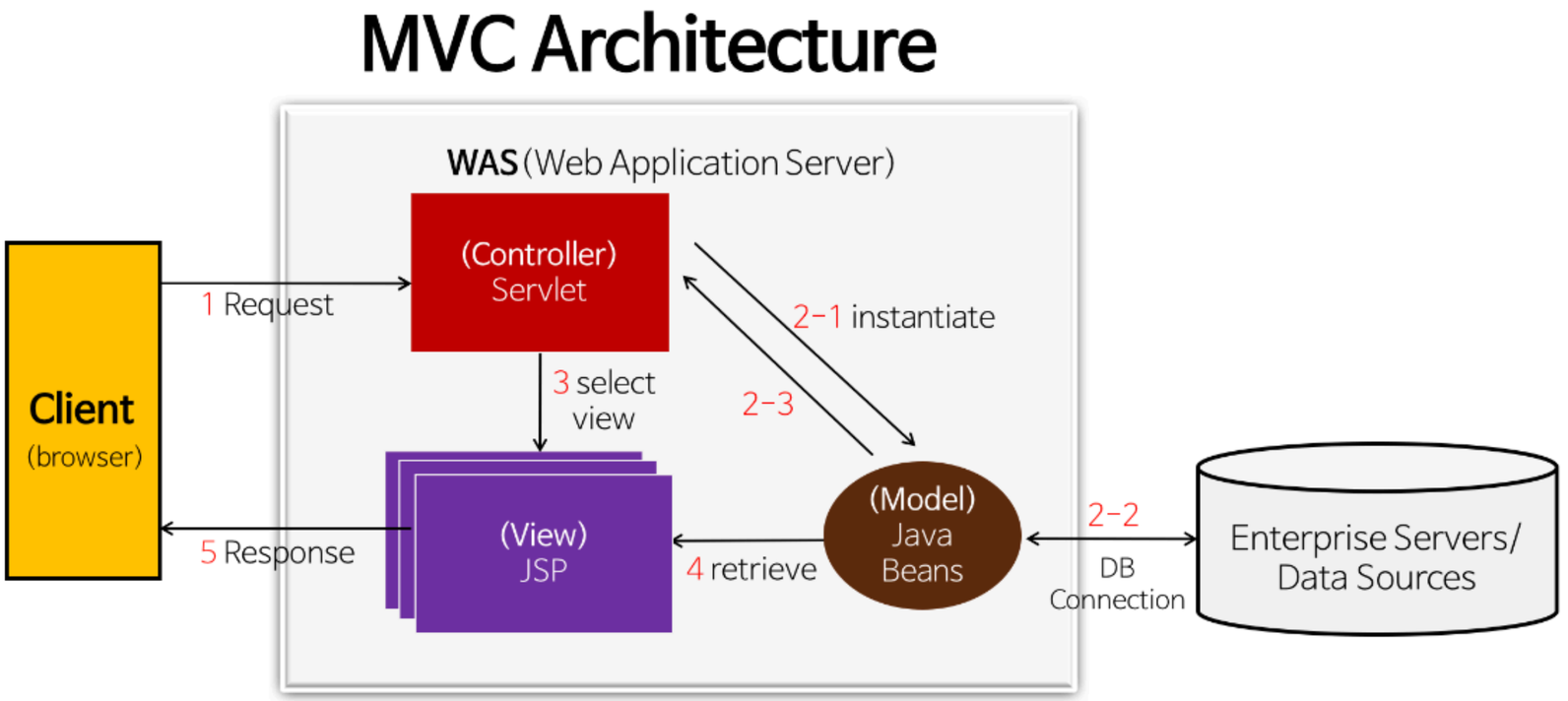
Servlet

- 자바 코드에 HTML코드
- MVC 패턴에서 Controller 역할
- 자바로 구현된 CGI
- HTML 변경시 재컴파일해야 하는 단점

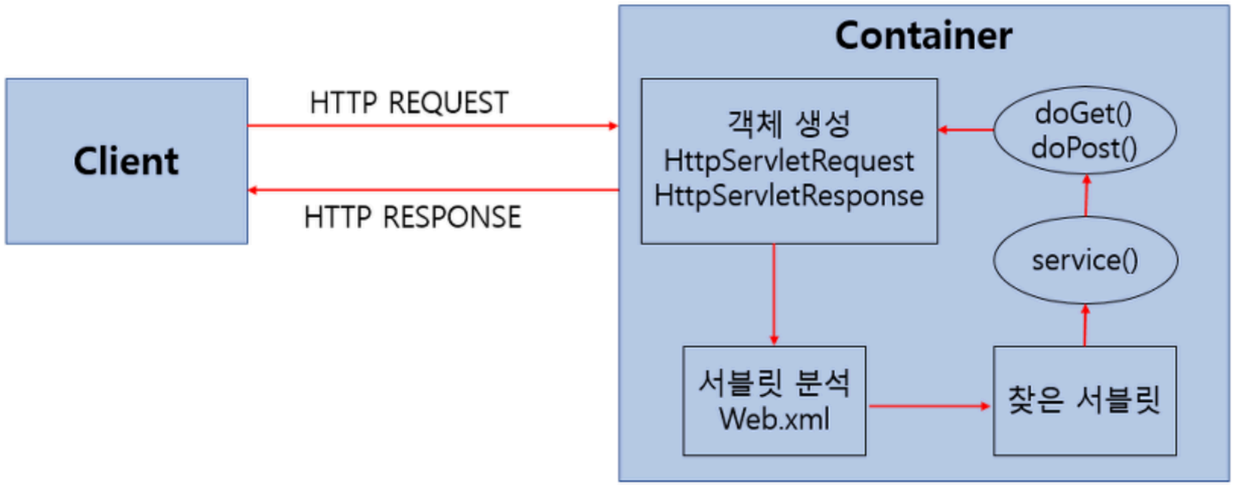
JSP (Java Server Page)

- HTML에 자바 코드
- 수정된 경우 재배포 필요 없음
- 스크립트 언어
- MVC 패턴에서 View 역할

MVC2 구조

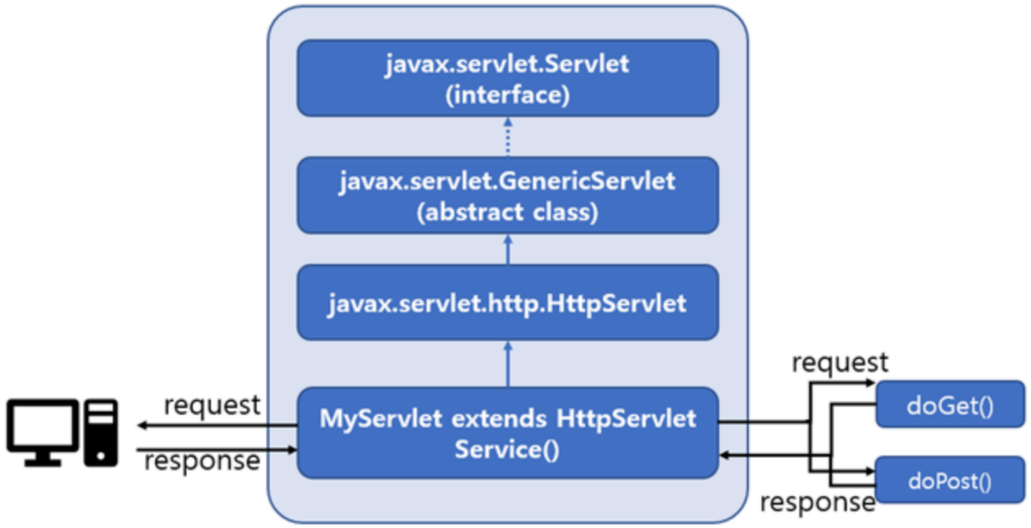

MVC Architecture
WAS (Web Application Server)
(Controller)
Servlet
Client
(browser)
1 Request
2-1 instantiate
2-3
3 select
view
(View)
JSP
5 Response
4 retrieve
(Model)
Java
Beans
2-2
DB
Connection
Enterprise Servers/
Data Sources

Servlet Life Cycle



- 서블릿 매핑 정보를 참조하여 해당 서블릿 호출
- 호출된 서브릿이 로딩 (class loading) -> Servlet 객체 생성
- Servlet 초기화 작업(최초 요청 1번) : `init()`
- 요청에 대한 내용 처리하고 Response :`service()`
- Servlet 소멸 : `destroy()`

Servlet Interface




```java
public class FirstServlet implements Servlet {
        ServletConfig config;

        @Override
        public void init(ServletConfig config) throws ServletException {
                System.out.println("init() called");
                this.config = config;
        }

        @Override
        public void destroy() {
                System.out.println("destroy() called");
        }

        @Override
        public void service(ServletRequest req, ServletResponse res) throws ServletExc
                System.out.println("service() called");
        }

        @Override
        public ServletConfig getServletConfig() {
                System.out.println("getServiceConfig() called");
                return this.config;
        }

        @Override
        public String getServletInfo() {
                System.out.println("getServletInfo() called");
                return "version=1.0;author=butterfield;copyright=butterfield 2020";
        }
}
```

Servlet Container, Tomcat

- Servlet을 관리하는 컨테이너
- 클라이언트의 Request를 받고 Response를 하도록, 웹 서버와 소켓을 만들어 통신하기 위함
- ex) Tomcat *(Servlet Container != Application Server)* -> 의견차 있음
  - tomcat같은 WAS가 java 파일을 컴파일 -> .class 파일 만들고 메모리에 올려 Servlet 객체 생성
- 역할
  - 웹 서버와의 통신 지원
  - Servlet Life Cycle 관리
  - 멀티 스레드 지원 및 관리
  - 선언적인 보안 관리

끝 \^ㅇ^/