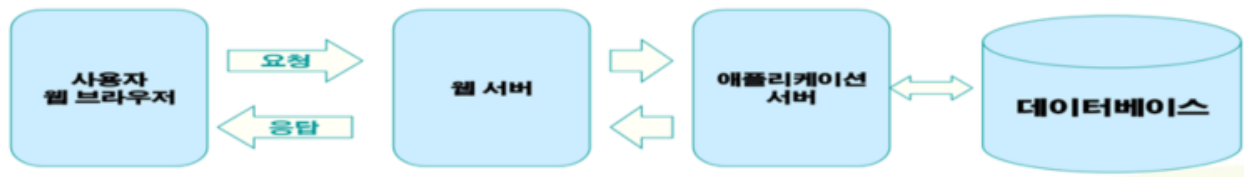


# 1. 웹 프로그래밍

## 1) 웹 프로그래밍이란

먼저 웹(web)은 간단히 말해서 클라이언트(사용자)가 서버에 정보를 요청하면 그에 응답하는 콘텐츠라고 말할 수 있는데 여기서 클라이언트가 필요한 정보를 웹 서버에서 가져온다. 이때 데이터를 가공하거나 별도의 구현이 필요할 경우 웹 어플리케이션 서버 WAS(Web Application Server)가 DB로부터 데이터를 처리하며, 처리한 데이터를 웹 브라우저에 보낸다.



## 2) Web Server

웹 서버는 클라이언트로부터 HTTP요청을 받아들이고 그 요청을 받아 정적인 콘텐츠(단순 HTML문서, CSS, 이미지, 파일 등. 즉시 응답 가능한 콘텐츠)를 제공하는 서버이다. 또한 웹 서버는 동적인 콘텐츠 요청을 받으면 WAS에 클라이언트의 요청(request)을 보내고 was가 처리한 결과를 클라이언트에게 전달(response)하는 역할도 한다. 대표적인 웹서버로는 Apache HTTP, IIS 등이 있다.

## 3) WAS(Web Application Server,)

WAS(웹 어플리케이션 서버)는 웹 서버에서 처리할 수 없는 동적인 정보를 처리하여 웹 서버에 정적인 정보로 변환하여 제공해 주는 역할을 하며 일반적으로 웹 서버의 기능을 가지고 있어 따로 웹 서버가 없어도 작동 가능하다.

### (1) WAS의 역할

WAS는 Web Server(웹 서버)와 Web Container(웹 컨테이너, servlet 컨테이너)를 포함하고 있으며 주로 Web Server 기능들과 구조적으로 분리하여 동적 콘텐츠를 처리하고자 하는 목적으로 제시되었다. 또한 주로 DB 서버와 같이 수행되며 다음과 같은 기능이 있다.

- 프로그램 실행 환경과 DB 접속 기능 제공
- 여러 개의 트랜잭션(논리적인 작업 단위) 관리 기능
- 업무를 처리하는 비즈니스 로직 수행

대표적인 웹 어플리케이션 서버로는 Tomcat, JBoss, JBoss, Web Sphere 등이 있다.

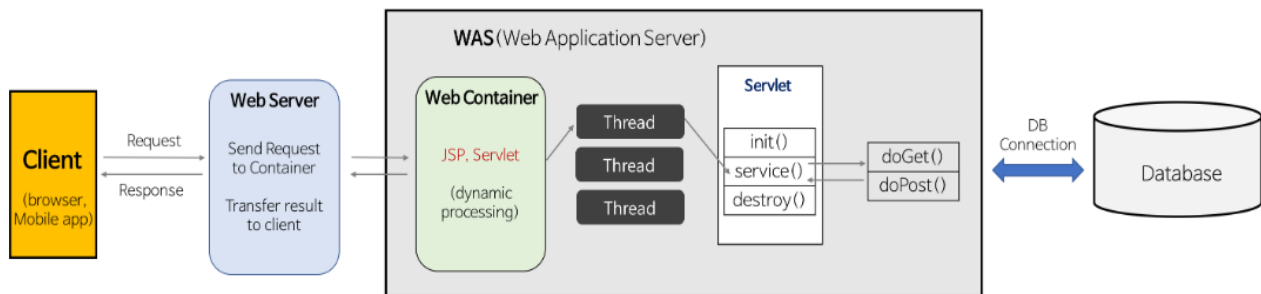
\* WAS와 WEB를 구분하는 이유

WAS는 기본적으로 동적 콘텐츠를 제공하기 위해 존재하는 서버이다. WAS는 DB 조회나 다양한 로직을 처리하느라 바쁘기 때문에 단순한 정적 콘텐츠는 Web Server에서 빠르게 클라이언트에 제공하는 것이 효율적이며 정적 콘텐츠 요청까지 WAS가 처리한다면 정적 데이터 처리로 인해 부하가 커지게 되고, 동적 콘텐츠의 처리가 지연됨에 따라 수행 속도가 느려진다.

즉 단순 정적인 데이터는 웹 서버에서 동적인 데이터는 WAS에서 구분하여 처리하는게 효율적이다.

## (2) WAS의 구조와 동작과정

### Web Service Architecture



1. Web Server는 Client로부터 HTTP요청(Request)을 받는다.
2. Web Server는 Client로부터 받은 요청(Request)을 WAS에 보낸다.
3. WAS는 동적 요청과 관련된 Servlet을 메모리에 올린다.
4. WAS는 web.xml(배포서술자)을 참조하여 해당 Servlet에 대한 Thread를 생성한다. (Thread Pool 이용)  
\* 여러 사용자의 동일한 페이지 요청시 하나의 프로세스만 할당하고 여러 Thread 방식으로 처리한다
5. 요청(HttpServletRequest)과 응답(HttpServletResponse) 객체를 생성하여 Servlet에 전달한다.
6. WAS는 생성된 Thread를 통해 Servlet을 호출(service() 메서드 호출)하며 요청 방식(POST/GET)에 맞게 doGet() 또는 doPost() 메서드를 호출한다
7. 호출된 doGet() 또는 doPost() 메서드는 인자에 맞게 생성된 적절한 동적 페이지를 Response객체에 담아 WAS에 전달한다.
8. WAS는 Response 객체를 HttpServletResponse 형태로 바꾸어 Web Server에 전달한다.
9. 생성된 Thread를 종료하고, HttpServletRequest와 HttpServletResponse 객체를 제거한다.

이때 어플리케이션 서버(WAS)를 이용할 때 필요한 프로그램이 Tomcat(톰캣)이다. 톰캣 등의 WAS를 이용하는 서버개발은 JAVA EE에서 제공하는 추가적인 기능을 필요로 한다. 여기서 JAVA EE는 JAVA SE의 확장 버전으로 서버 개발을 위한 추가 기능을 제공하는 플랫폼이다

#### 4) Servlet

Servlet이란 자바 코드 안에 HTML코드를 포함하여 동적인 웹 페이지로 작성하는 것으로 JAVA EE에서 제공하는 하나의 자바 클래스이다.

- 웹을 다룰 수 있는 'HttpServlet' 클래스를 상속받은 자바 클래스 파일을 의미하며 확장자가 .java이다.
- 웹 화면 구현에 필요한 HTML코드를 작성하는데 단순히 화면에 코드를 출력하는 방식이라 매우 번거롭다.

#### 5) JSP

JSP란 Java Server Pages의 약자로 Servlet과는 반대로 HTML코드안에 자바 코드를 포함하여 동적인 웹 페이지로 작성하는 것이다.

- 웹 화면 구현이 까다로운 servlet의 단점을 보완하기 위해 생긴 servlet 기반 스크립트 기술로 파일 확장자가 .jsp이다.
- JSP파일은 servlet 기반이기 때문에 웹 브라우저로 바로 보내지는 것이 아니라 먼저 웹 서버로 보내지며 이때 웹 서버는 JSP파일을 servlet 클래스로 변환하여 전달한다.

#### 6) 웹 컨테이너(Web container)

웹 컨테이너란 JSP, Servlet을 실행시킬 수 있는 WAS에 포함된 소프트웨어로 servlet 컨테이너라고도 한다. 즉, 웹 컨테이너는 JSP, Servlet의 구동 환경을 제공한다고 볼 수 있다.

#### 7) MVC패턴이란

웹 페이지 제작에 있어서 servlet과 JSP는 만드는 방법에 차이가 있을 뿐 동적 웹 페이지를 생성한다는 역할은 같다. 초기 자바 웹 개발은 servlet 위주였으나 이후 JSP가 등장하면서 JSP형태의 개발이 유행하고 지금에 와서 각각 역할을 나누어 servlet + JSP형태의 기술이 발전되고 MVC: view – controller- model 형식이 되었다. 즉, 화면구현(View)에는 HTML 문서작성에 보다 수월한 JSP로, 데이터 관리(Controller)는 데이터를 정형하는데 용의한 servlet등으로 나누는 것이다.

