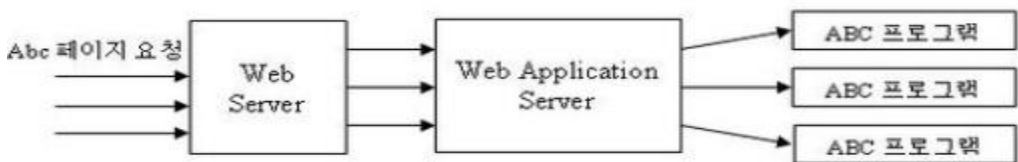


WAS의 동작과 WEB, Servlet, JSP

(주)엑셈 컨설팅본부/APM 박 종현

웹 어플리케이션 서버란?

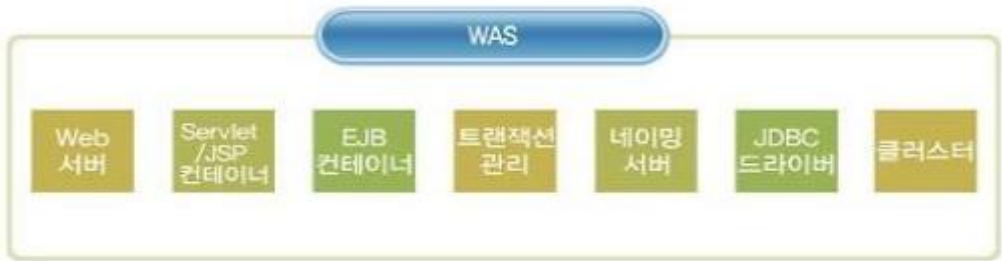
웹 어플리케이션 서버 방식은 웹 서버가 직접 어플리케이션 프로그램을 처리하는 것이 아니라 웹 어플리케이션 서버에게 처리를 넘겨주고 어플리케이션 서버가 어플리케이션 프로그램을 처리한다. 여러 명의 사용자가 동일한 페이지를 요청하여 여러 어플리케이션 프로그램을 처리할 때 오직 **한 개의 프로세스만을 할당하고 사용자의 요청을 thread 방식**(Servlet/JSP)으로 처리한다.



[그림1] WAS 방식

- **Note** 여러 개의 요청에 오직 1개의 프로세스만을 할당하고 사용자의 요청을 thread 방식으로 처리한다. 메모리를 절약할 수 있기 때문에 Servlet/JSP방식은 CGI 방식에 비해 동시에 더 많은 사용자에게 서비스를 할 수 있다. 이것은 전체적인 성능의 향상을 가져와 보다 안정적인 웹 서비스를 제공하는 것이 가능하다.

웹 서버의 기능들을 구조적으로 분리하여 처리하는 목적으로 아래 그림과 같이 웹 서버의 기능과 컨테이너의 기능으로 구성되어 있다.



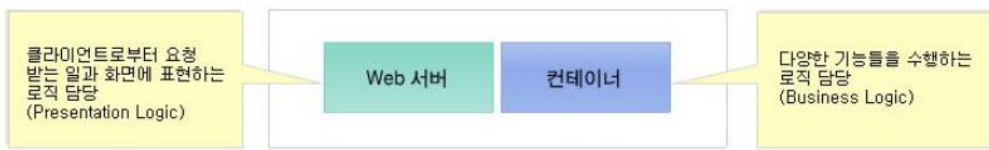
[그림2] WAS 구조

웹 서버와 어플리케이션 서버

웹 서버와 웹 어플리케이션 서버의 차이는 구현된 언어가 다른 것이다. 웹 서버는 C로 구현되며 웹 어플리케이션 서버는 JAVA로 구현이 되어 실행할 때 class 작업을 하며 메모리 할당 후 해제를 자동으로 GC가 관리한다.

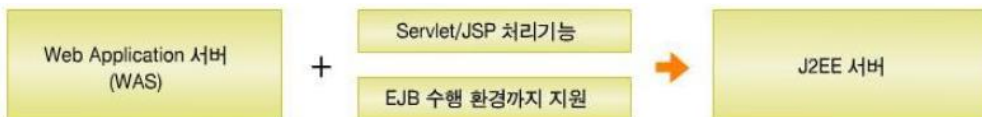
- 웹 서버는 웹에서 서버 기능을 처리하는 프로그램으로써 HTTP 라는 프로토콜을 기반으로 하여 웹 클라이언트로부터 요청을 서비스하는 기능을 담당하여 처리된 결과를 클라이언트로 응답한다.
- 웹 어플리케이션 서버는 웹 서버의 기능들을 구조적으로 분리하여 처리하는 목적이다. business logic 을 수행한다.

WAS는 웹 서버의 기능과 컨테이너의 기능으로 구성된다. 컨테이너는 Servlet 컨테이너이다.



[그림3] WAS의 구성 별 기능

- **Note** 대부분의 상용화된 웹 어플리케이션 서버들은 웹 서버나 컨테이너 기능 외에 엔터프라이즈 환경에 필요한 트랜잭션, 보안, traffic 관리, DB 커넥션 풀, 사용자 관리 등의 다양한 기능을 제공하는 소프트웨어로 개발되었다. J2EE 기술을 지원하는 서버로서 웹 서버 기능에 Servlet/JSP 처리와 EJB 수행을 지원하는 서버로 제공된다. 그래서 WAS 를 J2EE 서버라고 한다. **서버라고 불리지만 실체가 존재하지 않는, 즉 물리적 서버의 개념이 아닌 논리적 서버를 말한다**



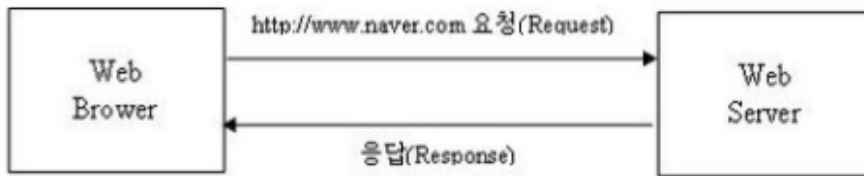
[그림4] WAS와 J2EE

위의 내용으로 WAS 의 구조와 성능에 대해서 알아보았다. InterMax 는 단순 WAS 모니터링이 아닌 business 트랜잭션 성능의 측면에서 모니터링을 하기 위한 툴이다. InterMax 를 통해 수집된 성능지표와 logging 된 데이터를 통해 문제 발생원인을 제대로 파악하기 위해서는 WEB 과 Servlet, JSP 에 대해서 좀 더 조사하고 이해하는 것이 도움이 될 것이다. 먼저 WEB 을 알아보도록 하자.

WEB

웹 프로그램의 이해

기본적으로 클라이언트(Client)/서버(Server) 방식의 형태이다. 클라이언트(웹 브라우저)가 특정 웹 서버에 요청(Request)하게 되면 웹 서버가 이를 처리한 후 결과를 클라이언트(웹 브라우저)에게 응답(Response)을 하게 되는 구조이다.



[그림5] WEB의 요청과 응답

웹 프로그래밍 언어의 종류

개인용 홈페이지와 다르게 많은 양의 데이터 처리나 많은 접속을 동시에 수용할 수 있도록 개발 가능한 언어들이다. CGI, ASP, PHP 등이 있으며 여기에서는 Servlet과 JSP에 대해 알아 볼 것이다.

Servlet(Server + Applet)

JAVA언어를 기반으로 동적인 contents를 생성하는 기술. JAVA 프로그램을 작성하는 형식과 거의 같다. JAVA 코드 안에 HTML 태그가 섞여 있어 작업에 대한 분리적인 측면으로 볼 때 효율성이 떨어진다. 프로그래머가 프로그래밍하기 어려우며 수정하기 어려운 단점이 있다.

JSP(Java Server Pages)

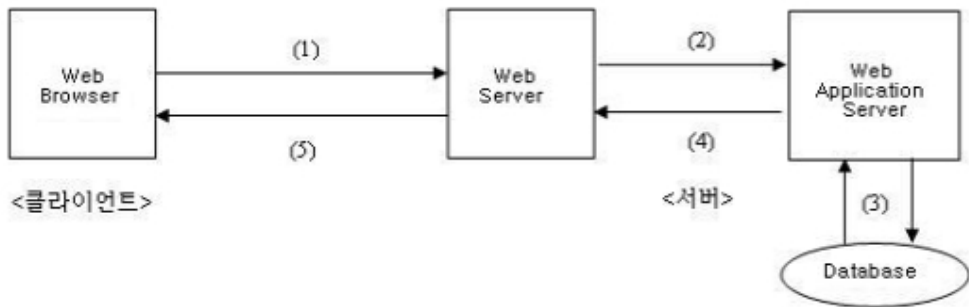
JAVA언어 기반으로 만들어짐. ASP, PHP 처럼 HTML 태그 사이 사이에 동적인 contents 생성을 담당할 JAVA코드가 들어 있는 형태로 Servlet의 형태와 다른 모습을 취하고 있다. 동적 contents를 생성하기 위해 스크립트 언어 형식으로 프로그램을 작성할 수 있어 개발자에게 쉬운 개발을 할 수 있게 한다. 또한 JSP는 사용자가 직접 태그를 정의해서 사용할 수 있는 사용자 정의 태그를 지정할 수 있는 기능을 갖고 있어 효율적인 웹 사이트를 구성할 수 있다.

- JSP 가 Servlet 으로 변환된다.
- 변환된 Servlet 이 compile 되고 실행된다.

웹 어플리케이션 처리 순서

웹 어플리케이션이란 웹을 기반으로 실행되는 프로그램을 의미한다. 따라서 웹 프로그래밍

과 웹 어플리케이션의 관계는 웹 프로그래밍을 통한 웹 어플리케이션의 구현이라 할 수 있다.



[그림6] 웹 어플리케이션의 처리 순서

- 웹 브라우저가 웹 서버에 페이지를 요청한다.
- 해당 웹 서버는 웹 브라우저의 요청을 받아서 요청된 페이지의 logic 및 데이터베이스와의 연동을 위해 어플리케이션 서버에 처리를 요청한다.
- 웹 어플리케이션 서버는 데이터베이스와의 연동이 필요하면 데이터베이스와 데이터의 처리를 수행한다.
- logic 및 데이터베이스 작업의 처리 결과를 웹 서버에 보낸다.
- 웹 서버는 결과를 다시 웹 브라우저에 응답한다.
-

웹 어플리케이션은 웹 브라우저, 웹 서버, 웹 어플리케이션 서버, 데이터베이스로 구성되어 있다.

웹 브라우저	웹에서 클라이언트이며 사용자의 작업 창이다.
웹 서버	웹 브라우저의 요청을 받아들이는 곳으로 작업의 결과를 웹 브라우저에게 응답을 하는 곳이다. 요청된 페이지의 logic 및 데이터베이스와의 연동을 위해 어플리케이션 서버에 이들의 처리를 요청하는 작업을 수행한다.
웹 어플리케이션 서버	요청된 페이지의 logic 및 데이터베이스와의 연동을 처리하는 부분이다.
데이터베이스	데이터의 저장소로 웹에서 발생한 데이터는 모두 이곳에 저장된다.

[표1] 웹 어플리케이션 구성

웹의 동작 방식과 웹의 구성 요소에 대한 조사와 정리를 하였다. 웹의 요청과 응답 과정과 처리 과정을 보면 웹 어플리케이션 서버가 웹 서버 및 데이터베이스와 연동하여 작업을 진행하는 것을 볼 수 있다. 웹 서버의 역할은 클라이언트의 요청에 따라 직접 서비스하거나 웹 어플리케이션 서버의 컨테이너를 이용하여 요청을 처리한 후 클라이언트에게 전달해 주는 것이다. Servlet 에 대한 요청이 발생할 경우 웹 서버는 Servlet 을 처리하기 위해 Servlet 컨테이너에 요청을 할 것이다. 다음 장에서 웹 프로그래밍 언어인 Servlet 과 JSP 에 대해서 알아보자.

JSP, Servlet

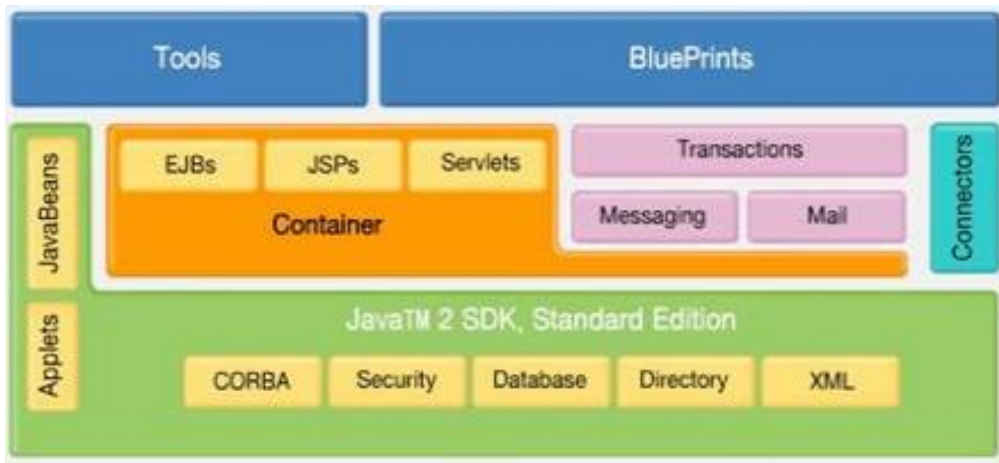
JSP

JSP 는 자바 Servlet 기술을 확장시켜 웹 환경 상에서 자바만으로 서버 사이드 모듈을 개발하기 위한 기술이다.

JSP 도 Servlet 과 마찬가지로 서버 사이드에서 DBMS 와 같은 Back-end Server 와 연동하여 Back-end Server 의 데이터를 가공하여 웹 상의 최종적 사용자에게 디스플레이 할 수 있고 여러 조건에 따라 디스플레이 할 수 있는 내용들을 동적으로 처리할 수 있는 기능을 제공한다.

JSP 는 웹 프로그래밍 언어 중 하나이다. 웹 프로그래밍 언어는 동적인 페이지를 생성하기 위한 서버 측 스크립트 언어이다. 그러한 언어들 중에서 JSP 는 자바를 기반으로 만들어진 것이다.

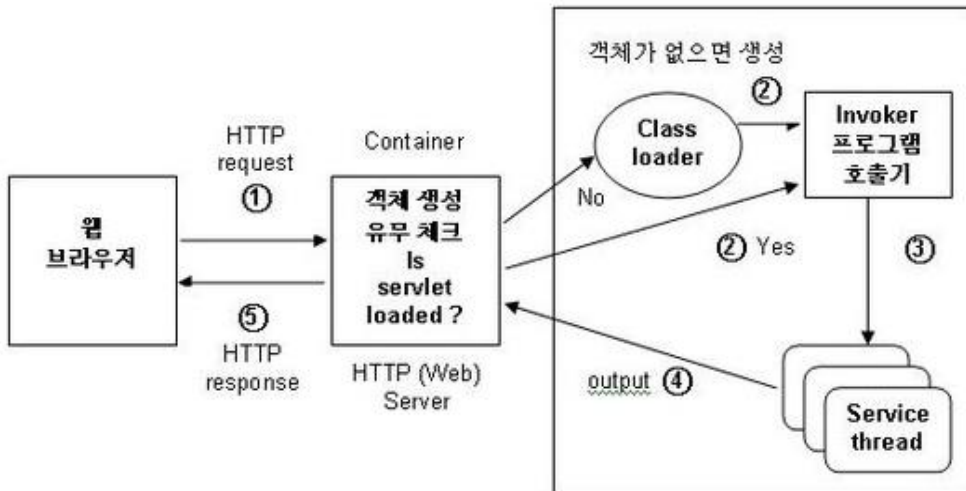
- **Note** 객체 지향적, 플랫폼 독립적, 네트워크 지향적, 뛰어난 보안, 멀티 thread 기능, 친근한 코드가 장점이다. JSP 는 자바가 갖는 특징들을 그대로 갖고 있으며 J2EE 를 구성하는 기술 중에 하나이다.



[그림7] J2EE

Servlet

Servlet은 멀티 threading에 의해 사용자 요구를 처리하고 가공해서 이에 대한 결과를 내보내게 된다. CGI가 클라이언트를 프로세스로 처리하는데 반해 Servlet은 클라이언트를 thread로 처리한다. 많은 클라이언트의 요구를 효과적으로 처리할 수 있다. Servlet 객체는 여러 개의 thread가 돌아가면서 처리하기 때문에 Servlet의 method들은 반드시 멀티 thread에 대한 고려를 해야 한다



[그림8] Servlet 처리 과정

1. 클라이언트의 서비스 요청(객체 생성의 유무 체크)
2. Yes 이면 생성하지 않고 No 면 객체를 생성한다.(메모리에 적재)
3. Invoker 에서 생성된 Thread 에서 service method(response 의 내용이 담김)을 호출. 클라이언트 당 1 개씩 생성된다.
4. 결과를 클라이언트에게 보낸다. 이 때 결과를 mime type 으로 보내는데 웹 브라우저의 mime type 은 text/html 이다.

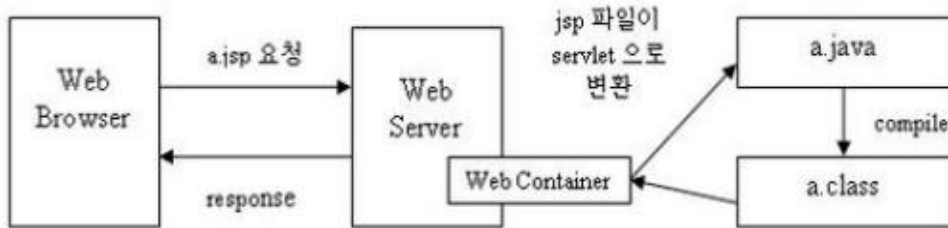
- **Note** Container 는 객체 생성의 유,무를 체크해서 객체가 생성되어 있지 않으면 생성한다.

J2SE, J2EE, J2ME와 Servlet, JSP의 관계

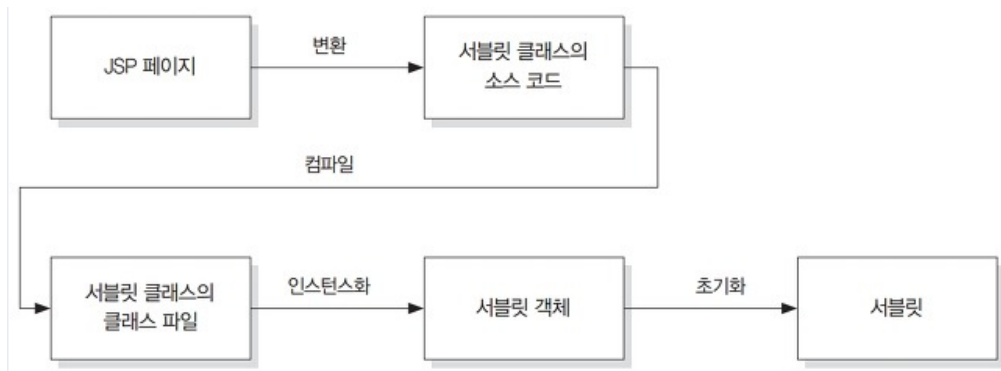
Servlet과 JSP는 J2SE 영역이 아니고 J2EE 영역이기 때문에 JDK(J2SE)에서는 Servlet 관련 API 가 없어 개발할 수 없다. J2EE 환경은 J2SE 환경 위에서 운용하게 되므로 J2EE에 속하는 프로그래밍을 하기 위해서는 먼저 J2SE가 설치되어 있어야 하며 J2EE를 지원하는 서버가 준비되어 있어야 한다.

동작 방식

웹 브라우저에서 JSP 페이지를 웹 서버로 요청하게 되면 웹 서버가 이 요청을 받게 된다. 사용자가 요청하는 것이 웹 서버는 JSP 파일인 것을 인지하고 작업을 웹 컨테이너(JSP 컨테이너)로 넘기게 된다. 이런 요청을 받은 웹 컨테이너는 해당 JSP 페이지를 찾아서 자바 파일로 변환을 시킨 다음 컴파일을 하게 된다. 컴파일 된 Servlet은 최종적으로 웹 브라우저에 응답 되어 사용자는 응답 결과를 보게 된다. 이러한 과정은 JSP 페이지가 최초로 요청되었을 때 단 한 번만 실행된다. 이 후 같은 페이지에 대한 요청이 있으면 변환된 Servlet 파일로 서비스를 처리한다.



[그림9] JSP 파일 Servlet으로 변환 과정1



[그림10] JSP 파일 Servlet으로 변환 과정2

	실행 코드 방식(CGI)	스크립트 방식(웹 어플리케이션 서버)
코드 형식	컴파일 된 실행 프로그램	컴파일 되지 않은 스크립트 코드
실행 방식	컴파일 된 기계어 코드 직접 실행	스크립트 코드를 해석한 뒤 실행
코드 변경	소스 코드를 다시 컴파일 해야 함	스크립트 코드만 고치면 됨
종류	대부분 CGI 프로그램	JSP, ASP, PHP

구현	웹 브라우저 요청 전송 - 웹 서버 요청 받음 - 프로그램 실행	웹 브라우저 요청 전송 - 웹 서버 요청 받음 - 스크립트 코드 번역 - 번역된 코드 실행 (번역이 최초에 한번만 실행 되기 때문 에 속도가 빠름)
----	---	---

[표2] 실행코드 방식과 스크립트 방식

컨테이너 측면과 서비스 API 측면 구분이유

J2EE는 컨테이너(Container)가 관리하는 컴포넌트(container-managed component) 그룹과 서비스 API(Service API) 그룹으로 나누어 진다. 각 분야마다 전문가들이 개별 컴포넌트를 하나만 집중해서 개발하면 된다. 이미 제공되는 서비스들 및 컨테이너를 연계하여 집중적인 튜닝을 통해 성능을 향상할 수 있다.

- **Note** Connection Pooling 과 같은 일반적으로 사용되는 컴포넌트나 서비스 분야와 비즈니스 logic 개발을 나누어 작업할 수 있다.

컨테이너

tomcat은 컨테이너의 좋은 예이다. apache와 같은 웹 서버가 사용자로부터 Servlet에 대한 요청을 받으면 Servlet을 바로 호출하는 것이 아니라 Servlet을 관리하고 있는 컨테이너에게 이 요청을 넘긴다. 이 컨테이너는 Servlet이 배포(deploy)된 컨테이너다. 요청을 넘겨받은 컨테이너는 HTTP Request와 HTTP Response 객체를 만들어 이를 인자로 Servlet doPost()나 doGet() method 중 하나를 호출한다.

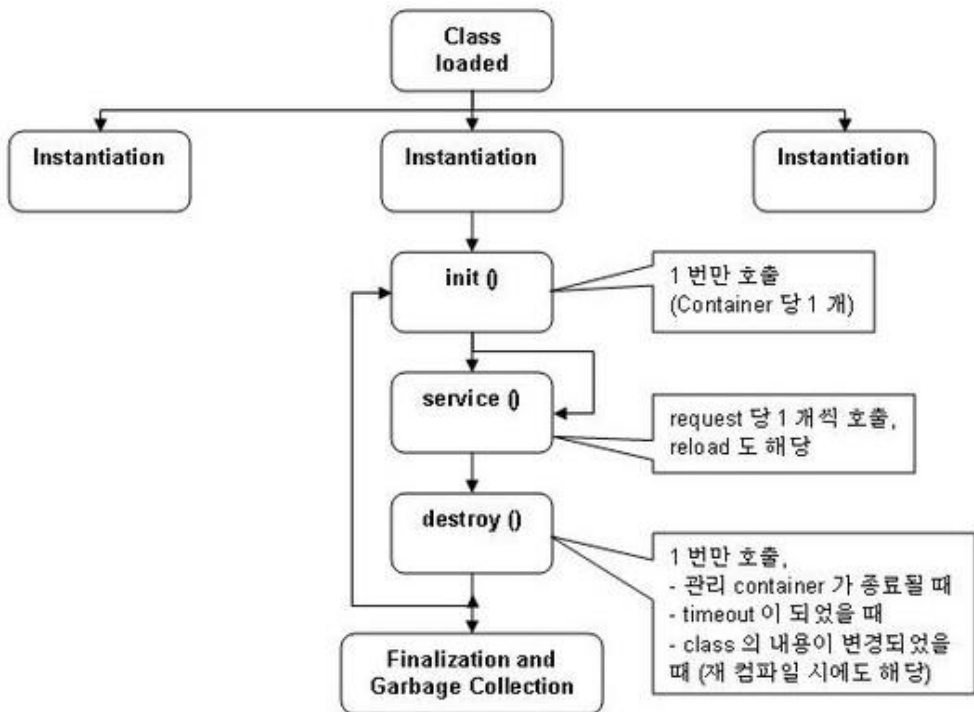
- 웹 컨테이너. JSP 는 서버 측에서 동작하고 웹 서버와 연동이 되어 동적인 페이지를 생성한다. JSP 가 동작하는 고이 서버의 환경이라는 것이다. 이는 JSP 를 사용하기 위해서는 서버 환경이 구축되어야 하며 기본적으로 웹 서버와 웹 컨테이너가 설치되어 있으면 된다. 대부분의 웹 컨테이너는 웹 서버 기능을 가지고 있다.
- Servlet 컨테이너. Servlet 을 동작 시킬 수 있는 환경을 제공하는 서버 프로그램이다. 즉 HTTP 요청을 받아서 해당 Servlet 을 동작시키고 그 결과를 사용자의 브라우저로 전달 해

주는 기능을 제공한다. 보통 컨테이너라고 하는 이유는 Servlet 프레임워크 안에서 동작을 하고 Servlet 이 동작할 수 있는 환경을 제공해주며 기타 필요한 작업 등을 제공해주기 때문이다. HTTP parameter parsing 및 결과 전달을 컨트롤, Forwarding, Redirecting 등의 기능을 컨테이너에서 제공해 준다. 이 때 Servlet 개발자는 자신이 만든 Servlet 을 이 컨테이너에 등록을 하게 되고 실제 동작을 컨테이너가 알아서 하게 되기 때문에 사용되는 언어이다.

- JSP 컨테이너. 실제로 JSP 컨테이너의 의미는 Servlet 컨테이너의 개념과 동일하다. 실제로는 Servlet 컨테이너가 JSP 컨테이너가 된다. JSP 는 PHP/ASP 와 같이 완전히 스크립트 형식으로 동작하지 않고 Servlet 으로 변환이 된 이후에 실행되기 때문이다. JSP 를 Servlet 으로 컴파일을 해주는 것이 JSP 엔진이다. tomcat 의 경우 JSP 엔진이 바로 JSPServlet 이다. 즉 JSP 를 컴파일하고 동작을 시켜주는 것을 특정 Servlet 이 담당하고 있다. JSP 자체가 완전한 Servlet 으로 컴파일 되지 않는다는 것을 알 수 있다. JSPServlet 이 구동할 수 있는 형태의 Servlet 으로 바뀌게 된다. JSP 컨테이너는 Servlet 으로 변환하는 과정에서 parsing 과정이 하나 더 켜져 있는 것이고 나머지는 똑같다.
- EJB 컨테이너. EJB spec 은 확장 가능하고 다중 사용자 처리에 안전한 트랜잭션 기능을 인식하는 분산된 비즈니스 어플리케이션 컴포넌트들을 개발하기 위해 필요한 표준화된 모델을 제공한다. EJB 는 비즈니스 logic 을 구현한 각종 컴포넌트들을 서버에 분산시킴으로 지속성 있는 데이터를 개체 지향적인 방식으로 표현할 수 있도록 해준다.

- **Note** 컨테이너는 Servlet 과 웹 서버가 서로 통신할 수 있는 손쉬운 방법을 제공하고 Servlet 의 라이프 사이클을 관리하고 요청이 들어 올 때마다 생기는 thread 에 대한 멀티 threading 을 지원하며 보안 관리, JSP 를 지원한다. J2EE 에서는 두 타입의 컨테이너를 제공한다. 웹 컨테이너와 EJB 컨테이너이다. 웹 컨테이너는 Servlet 과 JSP 에 대한 실행 환경을 제공하고 EJB 컨테이너는 Enterprise JavaBeans 에 대한 실행 환경을 제공한다.

Servlet의 Lifecycle



[그림11] Servlet의 생명주기

1. `init()` : 초기화 및 자원 할당. 단 1 번만 호출된다. (Container 당 1 개)
2. `service()` : request 당 1 개씩 호출된다. reload 도 해당
3. `destroy()` : 자원 해제 및 소멸 1 번만 호출된다. 관리 container 가 종료될 때 timeout 이 되었을 때, class 의 내용이 변경되었을 때, 재 컴파일 시 호출된다.

Servlet 이 처음에 load 되면 `init()` method 가 실행된다. `init()` method 에서는 Servlet 이 서비스하기 위해 필요한 초기화 작업을 수행하고 `init()` method 가 `service()` method 를 호출한다.

초기화된 Servlet 은 클라이언트의 요청이 있을 때마다 thread 가 생성이 되어서 병행적으로 `service()` method 를 실행한다.

Servlet 은 더 이상 서비스를 하지 않을 경우에는 Servlet 엔진에 의해 메모리에서 unload 된다. 이 때 unload 되기 전에 destroy() method 가 실행된다.

service() method 는 0 회 이상 실행되고 init() method 는 여러 번 실행될 수 있지만 그전에 destroy() method 가 실행되어야 한다.

WAS 내부적으로 JSP 를 JAVA 파일로 변환하고 그것을 다시 compile 하여 class 파일로 바꾸어 실행을 시킨다. 페이지 자체가 class 가 된다. Servlet 은 자체가 JAVA 로 되어 있어 그것을 class 로 compile 한다. Servlet 에다 한 단계 더 나아간 것이 JSP 이다. JAVA 와 HTML 을 분리하여 개발과 디자인을 구분하여 복잡성을 줄이고 upgrade 및 사후 관리를 더욱 편리하게 만들었다.

결론

WAS 를 monitoring 하기 위해서 WAS 의 구성 요소와 그 구성 요소가 무엇인지 조사를 해보았다. 인터넷의 발전으로 사용량이 증가하여 Web Server 의 작업을 세분화 하였다. 1tier 이었던 시스템의 점점 늘어나 n tier 로 쪼개졌다. 어느 부분에서 어떤 작업을 하고 어떤 역할을 하는지 모르면 장애가 발생하였을 때 원인을 찾기도 어려워졌다. 그렇기 때문에 WAS 엔지니어는 각 부분에 대해 어떤 작업이 이루어지는지 바로 파악할 수 있어야 하며 문제 현상을 보고 문제가 발생한 지점에서 적절한 대응을 할 수 있어야 한다.