JAVA EE의 소개

1 JAVA EE의 소개

자바는 플랫폼 독립성 이외에 객체지향 언어의 장점과 프로그램 생산성이 높다는 장점 때문에 많은 분야에서 사용되고 있습니다.
이렇게 많은 분야에서 자바가 사용되기 때문에 다양한 요구 사항과 필요성을 만족 시키기 위해서 자바는 크게 3개의 플랫폼(개발/실행환경)으로 나누어져 발전되고 있습니다.

JAVA SE : 표준 자바기반플랫폼(Java 2 Standard Edition)

가장 일반적으로 사용되는 표준 자바 플랫폼이다. JAVA SE는 우리가 일반적으로 JDK라고 불리는 것으로 가장 일반적인 자바 환경이다. 기본적인 네트워크, 애플릿, 쓰레드, 미디어, JFC, 보안, 주로 클라이언트의 GUI를 위한 JavaBeans(클라이언트 컴포넌트)에 관련된 API들을 제공하고 있다.

JAVA ME : 소형 가전제품 및 MOBILE 관련 자바기반플랫폼(Java 2 Micro Edition)

JAVA ME는 컴퓨팅 파워가 약한 소형 가전 제품들을 위해 개발되었기 때문에 JAVA SE에서 불필요한 기능들 을 많이 축소해서 사용한다. JAVA ME는 퍼스널 자바와 임베디드 자바등의 API를 제공한다. 퍼스널자바는 네크워크에 연결된 가전 제품을 위한 API이고 임베디드 자바는 메모리가 적은 내장 디바이스를 위해 개발된 API이다.

JAVA EE : 서버측 자바기반 인터넷플랫폼(Java 2 Enterprise Edition)

서버측에서 사용되기 위해서 개발된 것이 엔터프라이즈 자바이다.
엔터프라이즈 자바는 서버측에서 실행되는 분산 응용프로그램을 자바를 이용해서 효과적으로 개발하기 위한 목적으로 개발되기 시작하였다.
JAVA EE는 기본적으로 JAVA SE 의 API를 기반으로 하고 있다. 따라서 JAVA EE를 사용하기 위해서는 JAVA EE API는 물론이구 JAVA SE API를 모두 사용할 수 있어야 한다.

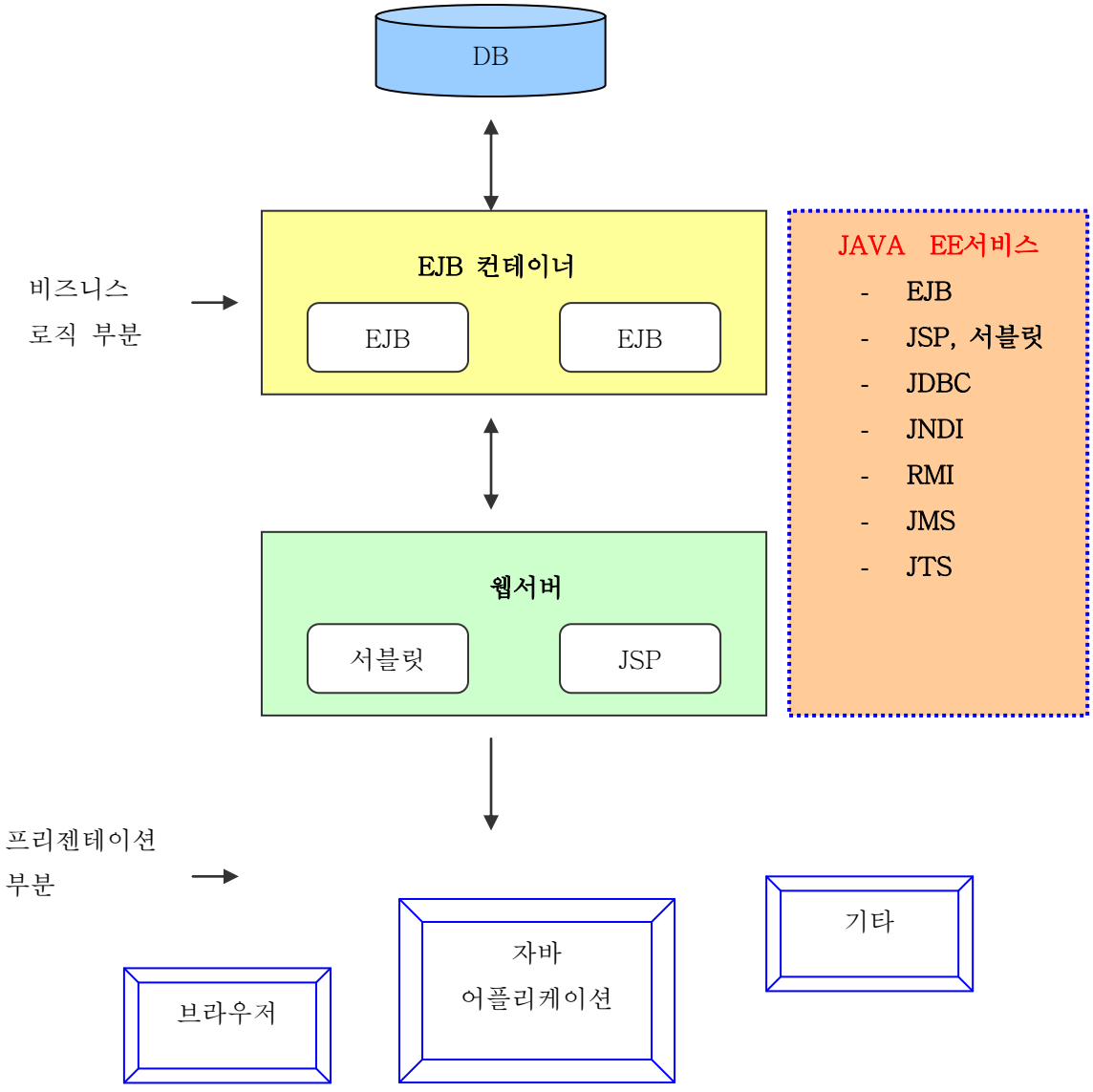
JAVA EE는 서블릿, JSP, JTA(Java Transaction API), JMS(Java Message System), XML, JNDI, EJB 등의
API를 제공한다.
특히 서버 측에서 사용되는 RMI, JDBC, 서블릿, JSP등의 기술은 매우 성공적으로 평가받았다.
이미 대부분의 사이트들이 위의 기술들을 이용하여 사이트를 구축했거나 구축중이거나 구축할 계획이다 이말이죠.

그런데 한가지 문제점은 점점 더 많아지는 데이터 양과 자꾸 대두되는 사용자 인증 및 보안, 속도등 여러가지 문제에 봉착하게 됨으로 써 자바 프로그램의 최고 기술이 집약 EJB란 놈이 대안으로 나타난 것입니다. EJB는 자바 개발자라면 반드시 습득해야 할 핵심 기술이다.

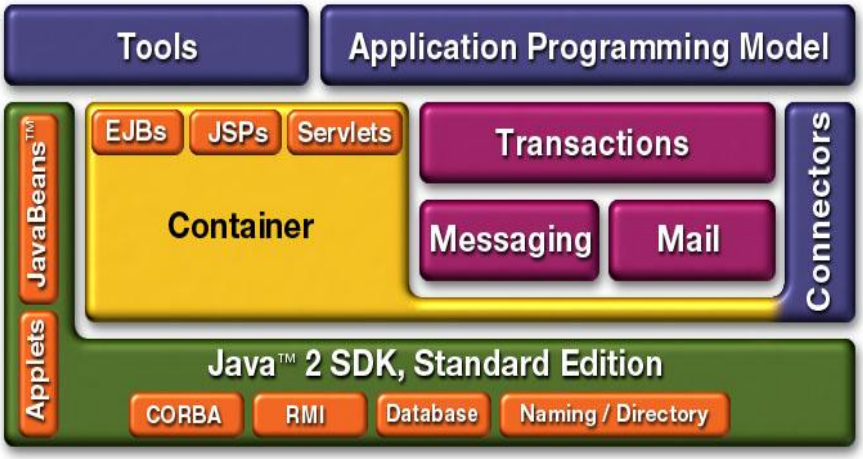
2 JAVA EE의 개요

JAVA EE는 웹 기반의 서버측 어플리케이션을 구축하기 위한 선의 플랫폼으로 클라이언트의 프리젠테이션 부분인 사용자의 브라우저와 서버의 데이터베이스 사이의 중간 계층에서 수행되는 다양한 서비스들로 구성되어 있다. 이러한 JAVA EE의핵심 서비스는 서버측 웹 클라이언트 컴포넌트(프리젠테이션부분)인 서블릿, JSP와 비즈니스 로직을 담당하는 EJB, 그리고 서버측의 정보자원들을 하나로 통합하기 위한 인터페이스들로 구성되어 있다.

JAVA EE에는 데이터베이스를 위한 JDBC, 네이밍과 디렉토리 서비스를 위한 JNDI, 트랜잭션 처리를 위한JTS, 메시지 서비스를 위한 JMS, 전자 우편 시스템을 위한 JavaMail, 그리고 RMI, 코바를 위한 JavaIDL(ORB)등을 포함하고 있다.



2 JAVA EE의 서비스



■ 서블릿/JSP

서블릿과 JSP는 JAVA EE 환경에서 웹의 동적인 문서 생성을 위한 인터페이스 부분을 맡는다.
서블릿은 개발자들이 쉽게 서버측 CGI 프로그램(컴포넌트)을 작성할 수 있도록 API를 제공하고 있으며 JSP는 서블릿의 개발상 단점을 보완한 기술로 기존의 HTML을 그대로 작성할 수 있으며, 동적인 문서 생성을 위해 HTML 문서에 자바코드를 삽입할 수 있도록 되어 있다.
서블릿과 JSP는 자바빈즈(JavaBeans)를 사용해서 비즈니스 로직을 처리하지만,
JAVA EE 환경에서는 자바빈즈 대신에 EJB를 사용해서 비즈니스 로직을 처리한다.
즉, 서블릿과 JSP는 웹 방문자에게 무언가를 보여주는 부분을 주로 담당하는 것이다.

■ JDBC

데이터베이스는 종류마다 특성이 다르기 때문에 호환성을 유지하기 어려웠다. 이를 극복하기 위해서 데이터베이스와 프로그램 사이의 표준 프로토콜을 만들어 사용하는 방식이 등장하였는데 ODBC가 그 대표적인 예이다. JDBC는 자바에서 데이터베이스를 조작하는 표준 방식이다.
데이터베이스 종류에 상관없이 똑같은 코드로 데이터베이스를 조작할 수 있다.
JAVA EE 환경에서 데이터베이스 조작은 JDBC를 통해 수행한다. 다만, 데이터베이스의 데이터를 자바 언어로서 추상화하는 것은 EJB의 엔티티 빈(Entity Bean)이 담당하는데, EJB 컨테이너(혹은 웹 애플리케이션 서버)가 엔티티 빈과 데이터베이스와의 데이터 항상성(persistency)를 유지하는데에도 내부

적으로 JDBC를 사용한다.

■ RMI(Remote Method Invocation)

원격지의 서버에 존재하는 객체를 접근할 수 있는 메커니즘으로 원격지 객체의 메소드를 마치 로컬상에서 호출하는 것과 같은 기능을 제공하는 서비스이다.

■ JNDI(Java Naming and Directory Interface)

JNDI는 객체나 자원들이 네트워크상의 어디에 존재하든 관계없이 이를 사용하고자 하는 어플리케이션이 그것을 찾을 수 있도록 다양한 네이밍과 디렉토리 서비스에 접근하는 표준화된 방식을 제공한다. JNDI를 사용하면 모든 컴퓨터 자원에 이름을 붙여서 사용할 수 있다.

■ JavaMail

JavaMail은 메일 주고 받기를 자바 환경에서 보다 객체 지향적으로 구현할 수 있도록 해 준다. JavaMail이 가장 많이 쓰이는 용도는 프로그램에서 메일을 자동으로 보내도록 하는 경우이다

■ JMS

point-to-point 메시징을 가능하게 해 주는 API이다. EJB 2.0에서는 JMS를 이용하여 비동기적 EJB를 만들 수 있도록 하고 있다. 메시징 관련 프로그램을 개발하려고 한다면 아주 유용하다.

■ JTA

복잡한 작업(transactions)을 효과적으로 관리하는데 사용된다. 예를 들어, A, B, C 세 개의 작업이 연속적으로 행해지다가 C 작업을 수행하는 도중 예외가 발생했다고 하면 A와 B 두 작업을 모두 취소하고 싶은 경우가 있는데 JTA를 사용하면 이런 일들이 가능하다.

■ EJB

EJB는 컴포넌트 기반 분산 객체 기술이다. EJB는 엔터프라이즈 급 개발에서 데이터 추상화와 비즈니스로직에 대한 부분을 담당하는 매우 핵심적인 요소이다. 사실 데이터 추상화와 비즈니스 로직은 개발의 대부분이다. 데이터 추상화는 엔티티 빈(Entity Bean)이 담당하고, 비즈니스 로직은 세션 빈(Session Bean)이 담당하는데 이 빈들은 모두 분산 객체 기술을 그 바탕에 두고 있다. EJB는 1.1 버전부터 RMI-IIOP를 통해 CORBA를 지원하고 있고, 2.0에 이르러서는 JMS를 통해 비동기적인 빈즈를 만들 수 있게 되었다.