아파치 앤트에 대해서, 그리고 빌드xml파일을 만들어서

어떻게 자동으로 빌드하게 되는지 발표하겠습니다.

**아파치 앤트**(영어: Apache Ant)는 자바 프로그래밍 언어에서 사용하는 자동화된 소프트웨어 빌드 도구이다. 유닉스나 리눅스에서 사용되는 make와 비슷하나 자바언어로 구현되어 있어 자바 실행환경이 필요하며 자바 프로젝트들을 빌드하는데 표준으로 사용된다.

앤트는 빌드를 위한 환경구성을 XML 파일을 사용합니다. 기본적인 빌드 파일명은 build.xml 이다.

로고는 이름(ANT)에 따라 개미 모양으로 만들어졌으나 이는 **Another Neat Tool**의 약어이다.

앤트의 최초 버전은 시스템 독립적으로 톰캣을 빌드하기 위한 툴로서 생겨났으며 2000년 7월 19일 독립형 제품으로서 발표되었다(1.1 버전). 처음에는 자카르타 프로젝트의 하위 프로젝트로서 시작했으나 2002년 10월 18일 아파치의 최상위 프로젝트로 승격해서 현재에 이르고 있다.

자카르타 프로젝트란

자바기반의 오픈 소스 소프트웨어를 개발하여 무료 배포하기 위한

목적의 프로젝트이다. 아파치 재단이 지원하며 톰캣, 앤트도 자카르타의

산출물이다.

최초 빌드 도구로서 제작되었으나 점점 많은 기능이 추가되면서 빌드와 배포, 유닛 테스트 등을 포함하는 통합 툴로서 발전되고 있다.

>>>>>아파치앤트 특징

장점

ü Ant 빌드 파일은 이식 가능, 플랫폼이 독립적

ü 파일 간 의존성 정보를 계속 추적. 소스파일이 변경되었을 때만 javac 컴파일러 사용

ü 자바 소스파일을 컴파일 하는 기능 외에도 갖가지 일을 하는 Task가 많이 포함되어 있다.

ü xml을 이용, 클래스 패스에 클래스 추가하기, 소스 파일 컴파일, jar 파일을 묶는 작업,

ü jar 파일을 특정 디렉토리로 이동시키고, 수정된 javadoc로 문서화 시키는 등의 작업을 한번에

ü 수행 가능한 making tool

>>>>>앤트기능

ü 자바 소스 컴파일

ü jar, war, ear, zip 파일의 생성

ü javadoc을 실행해 도움말 생성(api)

ü 유닉스처럼 파일 또는 폴더에 퍼미션 설정

파일또는폴더를 열어볼 수 있거나 수정할 수 있는 권한

ü 외부 프로그램 실행

ü /bin : Ant 실행을 위한 배치 파일이 있다.

ü /lib : jar 파일들

ü /docs : menual과 API 문서가 있다.

>>>>>>>빌드

빌드란, 소스 코드 파일을 컴퓨터에서 실행할 수 있는 독립 소프트웨어 가공물로 변환하는 과정을 말하거나 그에 대한 결과물을 일컫는다. 소프트웨어 빌드에 있어 가장 중요한 단계들 가운데 하나는 소스 코드 파일이 실행 코드로 변환되는 컴파일 과정이다

**빌드 자동화**

**빌드 자동화**의 본질적 목적은 소프트웨어 개발자가 반복해서 하는 코딩을 잘 짜여진 프로세스를 통해 자동으로 실행하여, 믿을 수 있는 결과물도 생산해 낼 수 있는 일련의 작업방식 및 방법을 말한다.

지속적인 통합(continuous Integration)과도 일맥상통되는 의미다. 수없이 반복해서 최종 결과물을 만들어내야 하는 소프트웨어 개발업무에서 빠질 수 없는 개념적 이해가 되었다. 특히 오늘날의 복잡하고 광범위한 협업적 개발 과정에서는 과정과정에서 있을 수 있는 인간적 실수의 가능성을 최소한으로 줄여야 한다.

>>>>>>>앤트 설치

아파치앤트 공식 홈페이지에 들어가서 다운로드 탭에 들어갑니다.

최신버전 다운로드하시구요 다운로드하셔서 원하는 위치에 압축을 풀면 됩니다. 그리고

환경변수를 설정해주어야 하는데

운영체제가 컴퓨터의 어떤 경로에서든 특정 파일(파일의 형식으로 되어있는 어플리케이션)을 인식할 수 있도록 환경변수를 등록하는 것, 운영체제가 프로세스를 처리할 때 쓰는 레퍼런스 정도라고 생각하시면 될 것 같습니다. 자바나 앤트나 실행파일이 들어있는 bin까지 경로를 시스템path에 입력해줍니다.

환경변수를 설정하시고 cmd창에서 ant버젼이나 ant를 쳤을 때 나오는 메시지를 보고 설치가 잘 됐는지 알 수 있습니다.

>>>>>>>>>>>>>>>>war파일 구조

수업시간에 만들었던 웹어플리케이션 프로젝트를 빌드해봤습니다.

우선 웹 어플리케이션을 빌드하고 톰캣서버에 배포 하려면 프로젝트롤 war파일 형식으로 압축할 필요가 있습니다. 보시는 사진은 바로 얼마전까지 지겹게 봐온 디렉터리 구조입니다. 이게바로 웹 어플리케이션의 구조이고 소스코드를 컴파일해서 war파일로 압축할 때 저 구조를 따라야합니다. 이클립스 프로젝트 디렉터리 구조에서 지금 사진의 디렉토리는 webcontent폴더 안에 있었는데 배포할때는 web컨텐츠폴더가 프로젝트 폴더가 된다고 생각하시면 됩니다.

그럼 war파일은 무엇이냐.

Was컨테이너 웹 어플리케이션 아카이브의 약자이고

servlet /jsp 컨테이너에 배치 할 수 있는 웹 어플리케이션 압출 파일 포맷이다.

그럼 서블릿은 무엇이며 서블릿 컨테이너는 무엇인가.

서블릿컨테이너는 웹서버 WAS내부에 있는 서블릿을 관리하고 작동시키는 서버프로그램입니다. 서블릿은 쉽게 말하면 자바로 작성된 서버프로그램으로 doget dopost메서드를 이용해서 클라이언트의 요청을 받고 응답합니다.

war파일이 실행되려면 Tomcat등의 웹서버나

웹컨테이너(WAS)가필요하다

war파일도 java의 java -jar를 이용해 생성하는 jar파일의 일종으로

웹 어플리케이션 전체를 패키징하기 위한 jar파일의 개념이다.

>>>>>>xml문서

앤트로 빌드하기 위해서는 빌드xml문서가 꼭 필요하기 때문에 xml문서에대해 간단하게 알아보는 시간 갖겠습니다.

Xml문서는 xml확장자를 가진 것 처럼 파일은 xml언어로 작성이 됩니다.

Xml언어는 마크업 언어로 만들어졌으며

데이터들을 쉽게 구조화하여 전달하기 위한 문서 형식입니다.

구조화된 문서.

메모장으로 열리고 쉽게 수정, 열람이 가능하다.

비슷한 기능으로 json 파일 형식이 있다.

Xml 선언 <?xml version=＂1.0＂ encoding=＂UTF-8＂ ?>

사람과 기계가 동시에 읽기 편한 구조이다.

Html처럼 데이터를 보여주기 위한 것이 아닌 저장과 전달이 목적이다.

Html처럼 태그가 미리 정의 되어 있지 않고 xml은 사용자가

직접 태그를 정의한다.

또한 사용자가 직접 정의할 수 있으며 다른 언어를 만느는 메타 언어라고도 합니다.

다른 시스템끼리 다양한 종류의 데이터를 교환하게 해준다.

기존 문서에

새로운 태그를 만들어 추가해도 계속 동작하므로 확장성이 좋다

텍스트 데이터 기반의 언어이며 유니코드 문자로만 이루어져 있다.

>>>>>>>>>>>>>xml문서규칙

꺽쇄괄호를 이용해 태그 형식으로 요소를 정의합니다. 계층형 요소 이고

하나의 xml문서는 루트 엘리먼트를 가지고 있습니다. 최상위 요소

앞서 말햇듯이 xml문서는 정보 저장과 전달이 주목적입니다. 책이라는 루트엘리먼트를 정의하고 그 내부에 자식 앨리먼트를 정의해서 책이라는 엘리먼트의 관련 정보를 제목 지은이 출판사 저장할 수 있습니다.

다들 앤트를 사용하기 위해 빌드파일을 작성해보셔서 아시겠지만

우리는 프로젝트라는 최상위 앨리먼트를 가지고

그 속에 일련의 작업단위인 태스크를 정의하는 타겟 엘리먼트들을 작성했습니다.

책 제목 지은이 출판사처럼 그냥 정보자체만 가진건 아니지만 프로젝트와 관련된 클래스패스, 파일구조, 저장위치, 필요한 라이브러리 등등 정보를 가지고 있죠. 이처럼 xml문서는 크게 보면 선언과 엘리먼트로 이루어져있습니다.

>>>>>>>빌드파일 작성

이제 앤트 빌드를 하기 위해서 빌트파일을 작성해보겠습니다.

빌드파일명은 빌드.xml이 디폴트입니다. 이름은 아무렇게나 지어도 상관없으나 빌드라고 지으면 커맨드할때 ant 세 문자만 쳐도 실행이 된다는 이점이 있습니다.

빌드파일은 빌드할 파일의 컴파일 및 배포에 관련된 작업들을 정의합니다.

루트앨리먼트로 프로젝트가 있고 그 아래로 컴파일 및 배포관련된 작업들이 타겟과 태스크로 이루어져 있습니다.

타겟들은 서로 의존성을 부여할 수 있고 하나의 작업이 완료되면 그로인해 다른 작업이 실행할 수 있도록 빌드과정을 짤 수 있습니다.

>>>>>XML문서 선언

저희조가 만든 필드파일을 보면서 설명해드리겠습니다.

Xml문서 답게 xml문서다라는 선언이 있구요 버전은 1.0 그리고 인코딩은 utf-8이 국룰입니다.

그리고 바로 루트엘리먼트인 프로젝트를 정의해줍니다. 이름은 그냥 프로젝트명이었던 웹08로 했습니다. 디폴트는 앤트가 빌드파일을 보고 빌드를 시작할 때 수해행하는 작업, 타겟이름을 정해줍니다. 베이스디렉토리는 빌드폴더가 있는 상대경로를 나타냅니다.

>>>>>>>>>>>>프로퍼티 정의

프로퍼티는 디렉터리 정보를 갖는 일종의 필드라고 생각하면 됩니다.

길고긴 디렉터리를 몇자이내로 줄일 수 있고 아래 타겟들이 짧은 프로퍼티 네임으로 긴 디렉터리를 표현할 수 있습니다. 빌드파일을 다른 컴퓨터같이 환경이 달라졌을 때 상단의 프로퍼티에서 달라진 디렉터리만 적용해주면 일일이 타겟의 내용을 손댈 필요가 없겠죠.

>>>>>>클래스패스

클래스패스의 경로도 지정해줍니다. 프로퍼티 네임 디렉터리 정의와 같은 개념으로 이해할 수 있습니다. 방식은 다르지만 일단 패스id를 정해줍니다. 원하는 이름으로 하셔도 무방하지만 보통 클래스패스를 나타내니까 클래스패스와 관련된 이름으로 하시는게 좋겠죠. 클래스패스는 앤트가 소스코드를 컴파일할 때 필요한 클래스정보를 담고 있는 라이브러리들의 위치를 알려주는 역할을 합니다. 맨날 복붙했던 ojdbc.jar의 위치와 웹에서 요청과 응답을 처리하는 서블릿클래스 정보가 들어있는 톰캣서버의 라이브러리 위치를 파일셋 디렉터리로 추가해줍니다. 인클루드 네임은 보시는 것처럼 저렇게 생긴 파일은 꼭 포함시켜라는 뜻입니다.

>>>>>>타겟정의

딜리트

이전에 빌드한게 남아있다면 제거해줍니다.

크리에이트

웹 어플리케이션이므로 웹인포 폴더내 라이브러리 폴더와 클래스폴더를 만들어줍니다.

컴파일

인클루드앤트런타임: 앤트라이브러리들을 클래스패스에 포함시킬지말지 여부를 결정. True of false. 이 빌드파일의 동작이 환경에 민감하여 잘 안될 수도 있어서 false로 하는게 일반적이다.

War 파일명을 정해주고 서블릿정보가 들어있는 웹xml위치를 알려줍니다.

라이브러리와 컴파일된 코드, 나머지 파일들을 모두 war형식으로 압축합니다.

Web08폴더에 web08.war파일이 생긴걸 볼 수 있습니다. 그리고 다음타겟

디플로이 war파일 복사후 톰캣 웹앱스폴더에 이동

서버스타트 war파일 압축해제

셧다운 서버종료.

이렇게 일련의 빌드와 배포과정이 완료되었습니다.