|  |
| --- |
| Apache Maven  2015년 5월 18일(목) 과제 |
| 통합ICTO팀 이영주 사원 |

**Apache Maven(아파치 메이븐)**

메이븐(Maven)은 자바용 프로젝트 관리 도구로 아파치 라이선스로 배포되는 오픈 소스 소프트웨어이다.

**Maven 사용의 장점**

1. 의존성 관리
   * 오픈 소스 라이브러리 뿐만 아니라 프로젝트 혹은 모듈 간의 의존성 또한 관리가 가능
2. 잘 정의된 convention에 의해서 일관된 프로젝트 구성 가짐
3. IDE에 종속적인 부분을 제거할 수 있음 (Eclipse, IntelliJ, NetBeans, JBuilder, JCreator, JDveloper)
4. 이클립스를 사용하는 경우 maven 이클립스 플러그인을 사용하면 의존성을 갖는 오픈 소스의 소스 코드를 F3키를 사용하여 아주 쉽게 확인할 수 있음
5. Maven을 알면 Java를 이용한 많은 오픈 소스 프로젝트들이 Maven 프로젝트로 구성되어 있어 오픈 소스를 분석할 때 편리
6. Maven Profile 기능을 사용하면 배포환경에 따른 설정 파일을 관리하고 배포 파일을 생성할 수 있음
7. 의존 라이브러리를 pom.xml 파일을 통해서 관리하므로 버전(형상)관리 시스템으로 공유할 파일의 크기가 감소

**Maven 설치 방법**

1. http://maven.apache.org/download.html에서 메이븐 최신 버전을 다운로드
2. 다운받은 압축파일을 원하는 경로에 풀어 놓고, 해당 경로를 시스템 환경 변수에 "MAVEN\_HOME" 을 추가
3. 시스템 환경 변수 "PATH"에 "MAVEN\_HOME/bin"을 추가

**Maven 설정 파일**

* MAVEN\_HOME/settings.xml : 모든 사용자에 적용되는 전역적인 메이븐 설정 정보
* USER\_HOME/.m2/setting.xml : 특정 사용자에 적용되는 메이븐 설정 정보
* pom.xml : 메이븐 프로젝트 설정 파일

**Maven 라이프 사이클**

1. 기본 라이프 사이클
   * compile : 소스 코드를 컴파일한다
   * test : 단위 테스트 실행 (기본설정은 단위 테스트가 실패하면 빌드 실패로 간주함)
   * package : 컴파일된 클래스 파일과 리소스 파일들을 war 혹은 jar와 같은 파일로 패키징
   * install : 패키징한 파일을 로컬 저장소에 배포 (USER\_HOEM/.m2/)
   * deploy : 패키징한 파일을 원격 저장소에 배포 (nexus 혹은 maven central 저장소)
2. clean 라이프 사이클
   * clean : 메이븐 빌드를 통하여 생성된 모든 산출물을 삭제한다.
3. site 라이프 사이클
   * site : 메이븐 설정파일 정보를 활용하여 프로젝트에 대한 문서 사이트를 생성한다.
   * site-deploy : 생성한 문서 사이트를 설정되어 있는 서버에 배포

**Maven을 이용한 의존성 관리**

* 메이븐의 핵심 기능 중 하나

1. 중앙 저장소 : 오픈 소스 라이브러리, 메이븐 플러그인, 메이븐 아키타입을 관리하는 저장소이다. 중앙 저장소는 개발자가 임의로 라이브러리를 배포할 수 없다.
2. 원격 저장소 : 메이븐 중앙 저장소 이외에 각각의 회사 혹은 오픈 소스 재단에서 운영 관리하는 저장소
3. 로컬 저장소 : 메이븐을 빌드할 때 다운로드하는 라이브러리, 플러그인을 관리하는 개발자 PC의 저장소 (USER\_HOME/.m2)

* 메이븐 의존성 관리
  + <dependencies>/<dependency> 엘리먼트를 사용하여 의존성을 관리하며, 의존 라이브러리의 groupId, artifactId, version, scope 정보들을 갖는다.

참조 : http://dimdim.tistory.com/entry/Maven-%EC%A0%95%EB%A6%AC