

<Pom.xml>

1. Maven이 pom.xml라는 빌드파일을 사용하여 빌드 정보를 기술한다. 따라서 다양한 라이브러리와 프레임워크 등을 Maven이 관리해주는데, 이때 프로젝트를 관리하기 위해 빌드파일인 pom.xml 을 사용한다.
2. Pom은 프로젝트의 구조와 내용을 설명하고 있으며, pom.xml 파일에 프로젝트 관리 및 빌드에 필요한 환경 설정, 의존성 관리 등의 정보를 기술한다.
3. 구성(엘리먼트)
 1. Project root : 모든 정보를 이 태그 내에 기술한다.
 2. ModelVersion : POM의 모델 버전이다.
 3. groupId : 그룹 ID로, 프로젝트를 만들 때 입력한다. 제작자와 회사, 단체 등을 식별하기 위한 것이다.
 4. artifactId : 프로젝트에 할당한 고유 ID이다.
 5. Version : 프로그램 버전으로, 기본적으로 1.0-SNAPSHOT 이 설정된다.
 6. Package : 패키지 종류로 jar 또는 zip을 지정한다. Web 어플리케이션의 개발은 war를 지정할 수도 있다.
 7. Name : 프로그램의 이름으로, 아티팩트 ID로 지정해서 많이 사용한다.
 8. url : 프로젝트와 관련된 Web 사이트의 주소이다. 기본적으로 Apache Maven의 Web 사이트 주소가 지정된다.
 9. Properties : pom.xml 에서 사용하는 속성 값 등을 모아 둔다.
 10. Dependencies : 의존 라이브러리 정보를 한곳에 모아 기술하기 위한 것으로, 각각의 의존 라이브러리 정보는 dependency 태그를 사용하여 작성한다.

<web.xml>

1. Web application의 설정을 위한 deployment descriptor로, Deploy할 때 Servlet의 정보를 설정해준다.
2. Web Application 시작 시 메모리에 로딩이 되며 브라우저가 Java Servlet에 접근하기 위해 WAS에 필요한 정보를 알려줘야 해당하는 Servlet을 호출할 수 있으며, 이것을 정하는 곳이 Web.xml이다.
3. 구성(태그 종류)
 1. icon : 웹 애플리케이션을 나타내기 위해 IDE나 GUI툴에서 사용되는 하나 또는 두개의 이미지 파일들의 위치를 지정하는데 사용한다.
 2. Display-name : GUI툴이 웹애플리케이션을 표시하기 위해 사용하는 이름을 지정하는데 사용
 3. description : 웹 어플리케이션에 대한 설명을 나타낸다
 4. distributable : Distributable 요소가 있다는 것은 웹어플리케이션이 다중서버간에 분산배치될 수 있다는 것을 의미한다.
 5. Context-param : 어플리케이션의 초기화 파라미터를 선언하는데 사용
 6. filter : 서블릿이나 jsp 페이지로 들어오는 요청 정보를 사전에 걸러내는 기능
 7. Filter-mapping : 필터를 지정했다면 filter-mapping을 지정하여 하나 이상의 서블릿과 연결함
 8. listener : 서블릿 2.3버전으로부터 세션이나 서블릿 컨텍스트가 생성 또는 수정 되거나 소멸되는 것을 알려주는 이벤트 리스너
 9. servlet : 서블릿이나 jsp페이지에 초기화 파라미터나 사용 url들을 할당할때 사용되는 서블릿 jsp이름을 지정
 10. Servlet-mapping : 상대 URL경로를 좀더 쉽게 다루기 위해 기본 URL을 변경할때 사용
 11. Session-config : 일정시간동안 세션으로 접근이 없을경우 서버는 메모리를 절약하기 위해 사용하지 않는 메모리를 삭제한다(세션의 시간유지기능)

<XXX-servlet.xml>

1. 처음 Web.xml이 구동되면 dispatcher-servlet으로 인해 servlet.xml에 있는 내용이 적용되게 된다. Controller의 매핑은 DispatcherServlet이 담당한다.
2. Web.xml에서 dispatcherServlet 등록 시 설정한 파일로, 설정 파일을 이용해서 스프링 컨테이너를 초기화 시킨다.
3. dispatcherServlet : 스프링 MVC에서 DispatcherServlet은 다른 웹 프레임워크의 프론트 컨트롤러처럼 클라이언트의 request를 컨트롤러에 전달할 뿐만 아니라, 스프링 ioc 컨테이너와 통합하여 스프링의 모든 기능을 제공한다.
4. 구성(태그 종류)
 1. Beans : spring이 메타데이터로 갖고있는 애노테이션이 붙은 클래스들을 빈으로 등록한다.
 2. Context : @component 로 되어있는 애노테이션을 읽어온다.
 3. Mvc : spring의 mvc를 활성화하기 위해 사용한다.
 4. Property : 해당 bean의 속성에 값을 넣어줄 때 사용한다.
 5. HandlerMapping : 사전/사후 처리를 위해 interceptor 리스트와 함께 request를 handler에 매핑한다.
 6. handlerAdapter : dispatcherServlet이 request에 매핑된 처리기를 호출하도록 도와준다.
 7. handlerExceptionResolver : 예외를 해결하기 위한 전략으로, 예외를 handler, Error page등에 매핑할 수 있다.
 8. ViewResolver : string 핸들러에서 반환된 view이름을 실제로 Response에 렌더링 할 view를 확인
 9. LocalResolver : Locale 국제화를 제공할 수 있도록 클라이언트가 사용하고 있는 시간대를 확인
 10. ThemResolver: 웹 응용 프로그램에서 사용할 수 있는 테마를 확인한다.