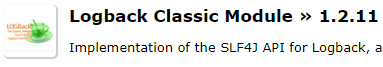
Logging Frameworks

Logback Classic Module 1.2.11 버전 다운로드

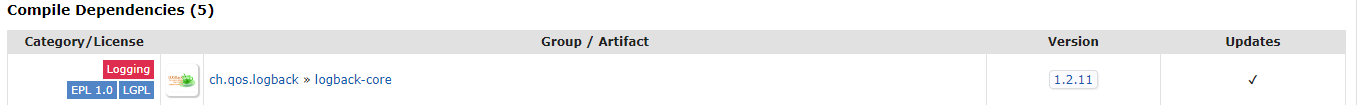
→ log관리

<https://mvnrepository.com/artifact/ch.qos.logback/logback-classic/1.2.11>



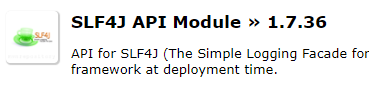
이 페이지에서 밑으로 내리다보면

→ Logback core 1.2.11 다운로드



너무 많은 파일들을 통합으로 만들어 놓은 것

<https://mvnrepository.com/artifact/org.slf4j/slf4j-api/1.7.36>



Logger를 import할 때는



→ Slf4j 선택

Maven Project만들기



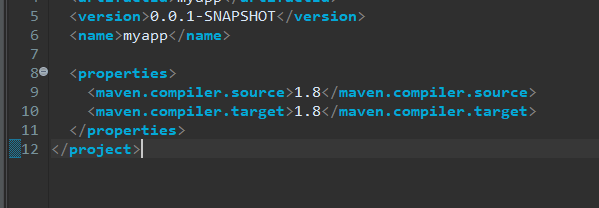
Artifact Id → 어플리케이션이 패키징 되었을 때 jar파일 이름(영소문자 띄어쓰기 없이)

Name → 똑같이 해줘서 헷갈리지 않게.



→ Maven이 만드는 자바 버전은 기본으로 1.5버전이다.

Pom.xml

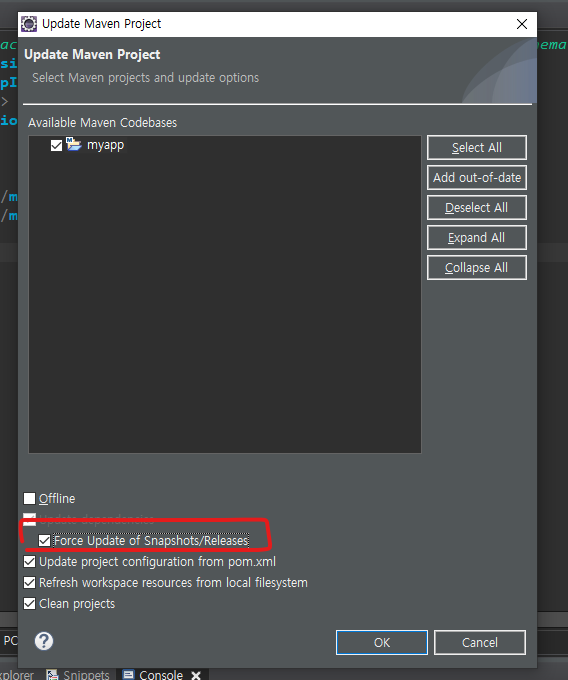


<**properties**>

<**maven.compiler.source**>1.8</**maven.compiler.source**>

<**maven.compiler.target**>1.8</**maven.compiler.target**>

</**properties**>



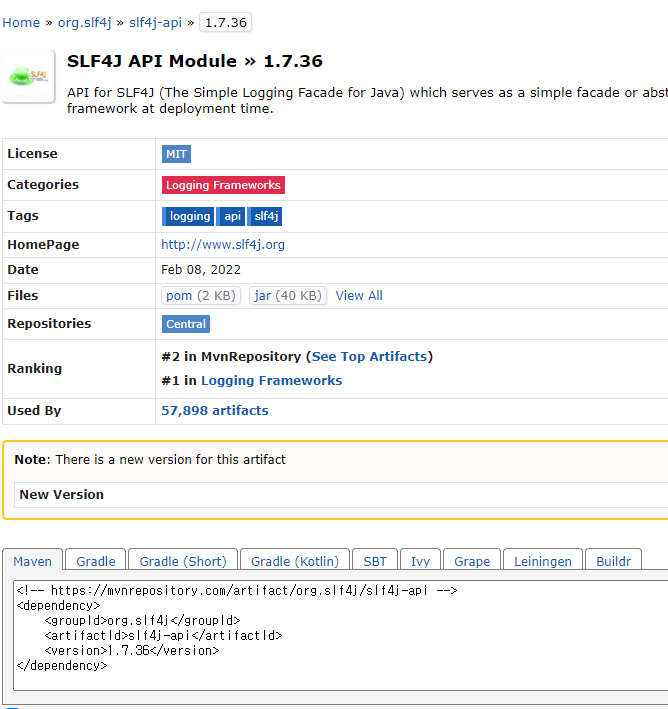
프로젝트 우클릭 → Maven → Update Maven Project

또는 alt + f5

→ 체크박스 클릭 해주면, 자바 1.8버전으로 업뎃해줌.

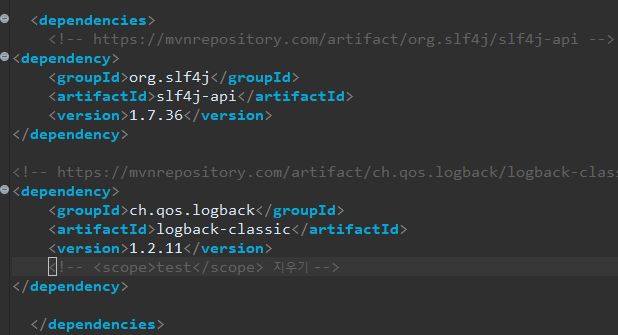
jar파일 자동으로 추가시키기!

SLF4J 검색 후

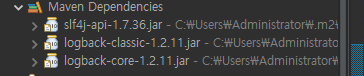


Maven 안의 내용들 복사.

Logback 검색 1.2.11 에서도 똑같이 메이븐 복사 후, slf4j내용 뒤에 붙여넣기

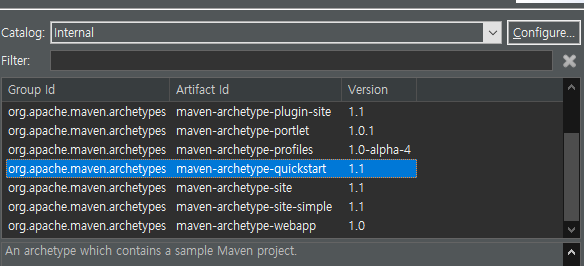


내용을 적고Pom.xml 저장하면,



자동으로 jar파일 생성됨.

Archetype !



Quickstart → 일반 java 프로젝트

보통은 일반 자바나 다이나믹 웹 프로젝트로 만든 후 메이븐 프로젝트로 변경하면 가능하다.

→



Maven은 라이브러리를 pom.xml에서 관리한다.

jar파일 주소를 넣어주면 알아서 생성해줌.

MVC 패턴!

→ Model View Controller

어플리케이션을 구동하기위한 개발자들의 경험을 토대로 만들어진 패턴.

서블릿으로 제어흐름을 빼고, 보여주기만 하는 jsp와 같은 패턴.

모델, Model

애플리케이션의 정보, 데이터를 나타냅니다. 데이터베이스, 처음의 정의하는 상수, 초기화값, 변수 등을 뜻합니다. 또한 이러한 DATA, 정보들의 가공을 책임지는 컴포넌트를 말합니다.

뷰, View

input 텍스트, 체크박스 항목 등과 같은 사용자 인터페이스 요소를 나타냅니다. 다시 말해 데이터 및 객체의 입력, 그리고 보여주는 출력을 담당합니다. 데이터를 기반으로 사용자들이 볼 수 있는 화면입니다.

컨트롤러,Controller

데이터와 사용자인터페이스 요소들을 잇는 다리역할을 합니다.

즉, 사용자가 데이터를 클릭하고, 수정하는 것에 대한 "이벤트"들을 처리하는 부분을 뜻합니다.

위의 개념을 WEB에 적용 시!

1.사용자가 웹사이트에 접속 (Users)

2.Controller는 사용자가 요청한 웹페이지를 서비스하기 위해서 모델을 호출 (Manipulates)

3.Model은 데이터베이스나 파일과 같은 데이터 소스를 제어한 후 그 결과를 Return

4.Controller는 Model이 리턴한 결과를 View에 반영 (Updates)

5.데이터가 반영된 View는 사용자에게 보여짐 (Sees)

Jsp2.3 웹프로그래밍 518페이지 참고