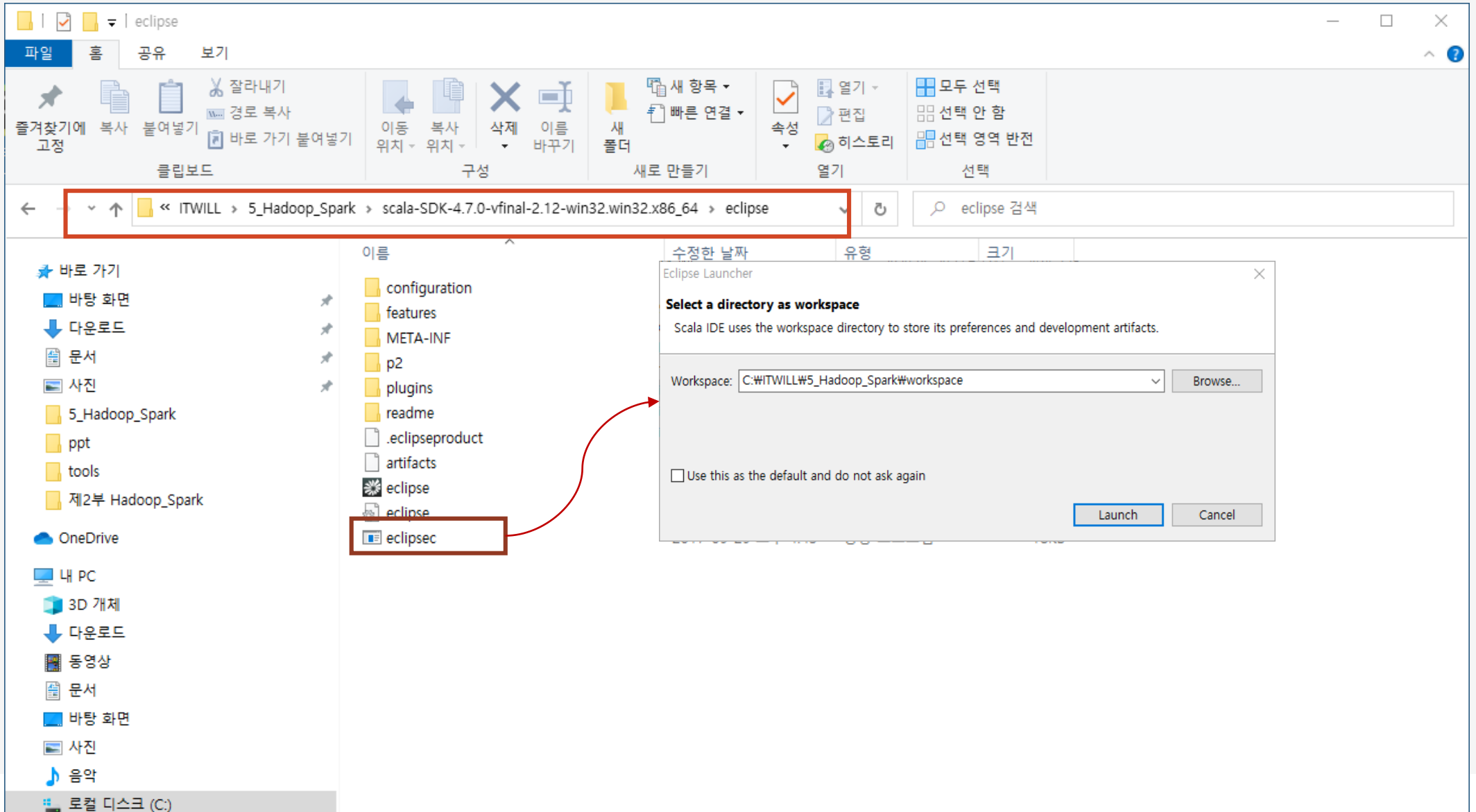
빅데이터 플랫폼 머신러닝 개발을 위한

SparkML 개발환경 구축(Maven Project)

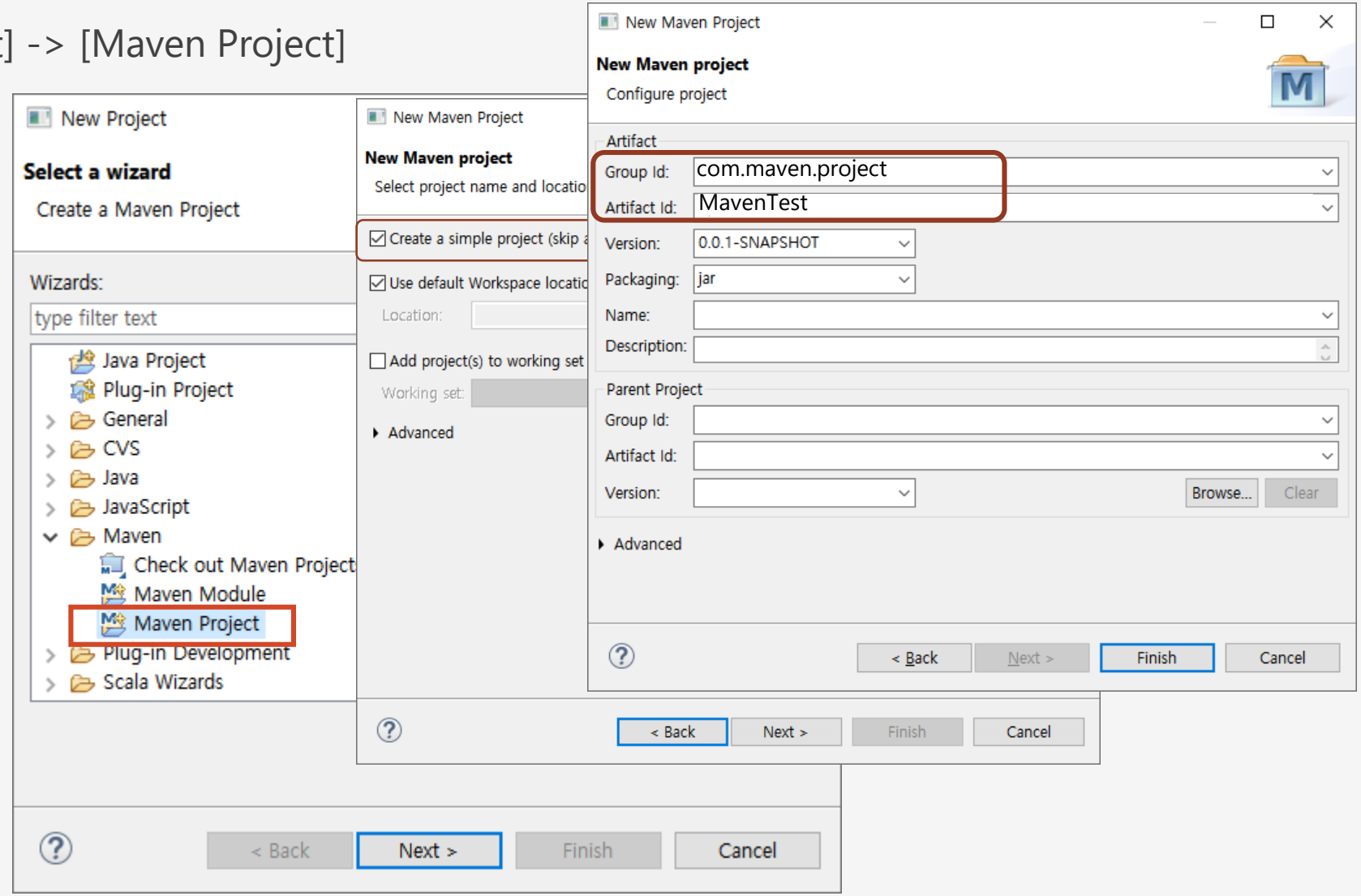
작성자 : 김진성

1. Eclipse 실행


eclipse
파일
홈
공유
보기
즐겨찾기에 고정
복사
붙여넣기
잘라내기
경로 복사
바로 가기 붙여넣기
클립보드
이동 위치
복사 위치
삭제
이름 바꾸기
구성
새 폴더
새 항목
빠른 연결
새로 만들기
속성
열기
편집
히스토리
열기
모두 선택
선택 안 함
선택 영역 반전
선택
« ITWILL › 5_Hadoop_Spark › scala-SDK-4.7.0-vfinal-2.12-win32.win32.x86_64 › eclipse
eclipse 검색
바로 가기
바탕 화면
다운로드
문서
사진
5_Hadoop_Spark
ppt
tools
제2부 Hadoop_Spark
OneDrive
내 PC
3D 개체
다운로드
동영상
문서
바탕 화면
사진
음악
로컬 디스크 (C:)
이름
수정한 날짜
유형
크기
configuration
features
META-INF
p2
plugins
readme
.eclipseproduct
artifacts
eclipse
eclipse
eclipsec
Eclipse Launcher
Select a directory as workspace
Scala IDE uses the workspace directory to store its preferences and development artifacts.
Workspace:
C:₩ITWILL₩5_Hadoop_Spark₩workspace
Browse...
Use this as the default and do not ask again
Launch
Cancel

2. maven project 생성

[File] -> [New] -> [Project] -> [Maven Project]

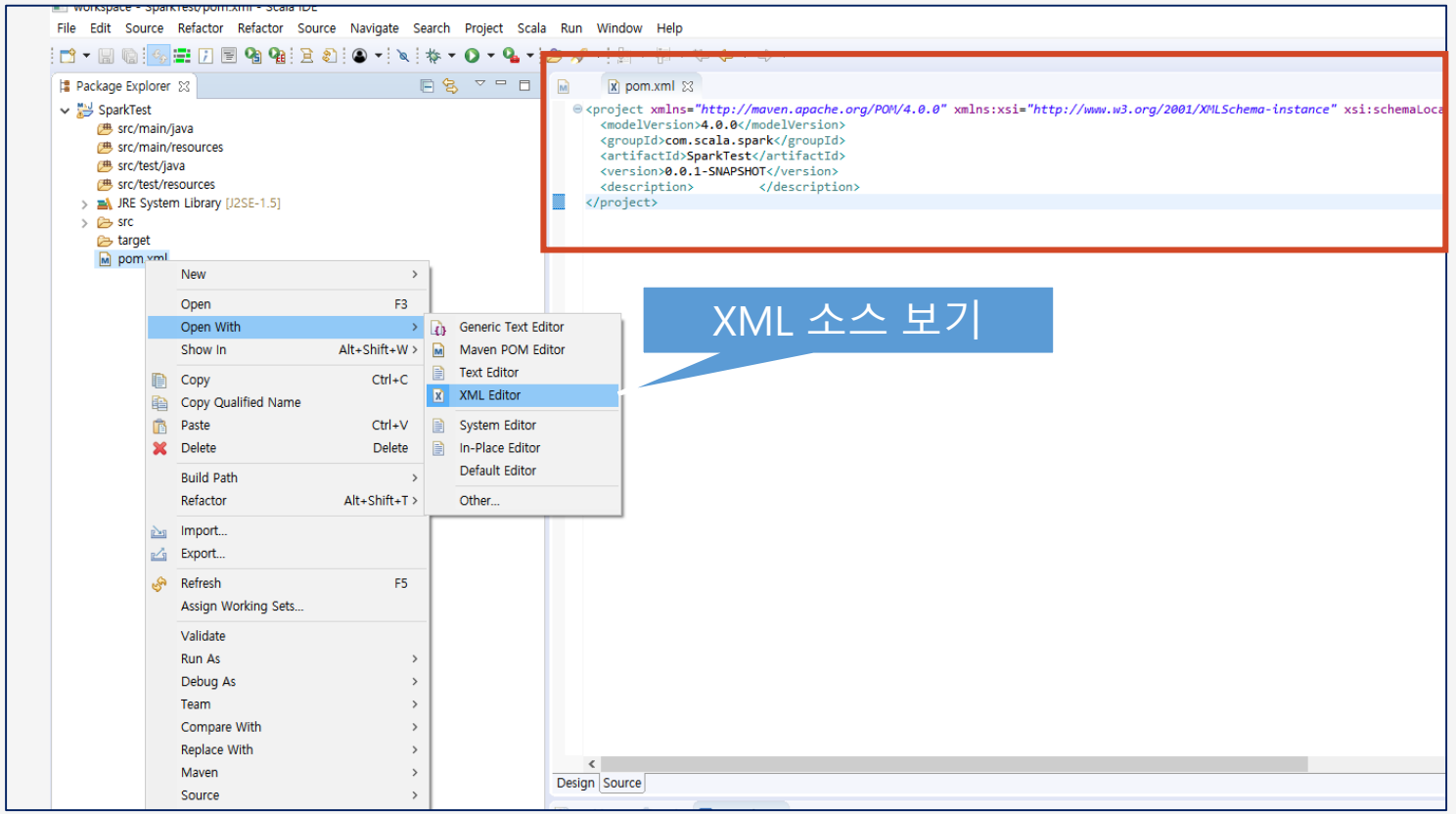


1) pox.xml 설정

Maven Project에서

개발에 필요한 라이브러리를

관리하는 설정파일

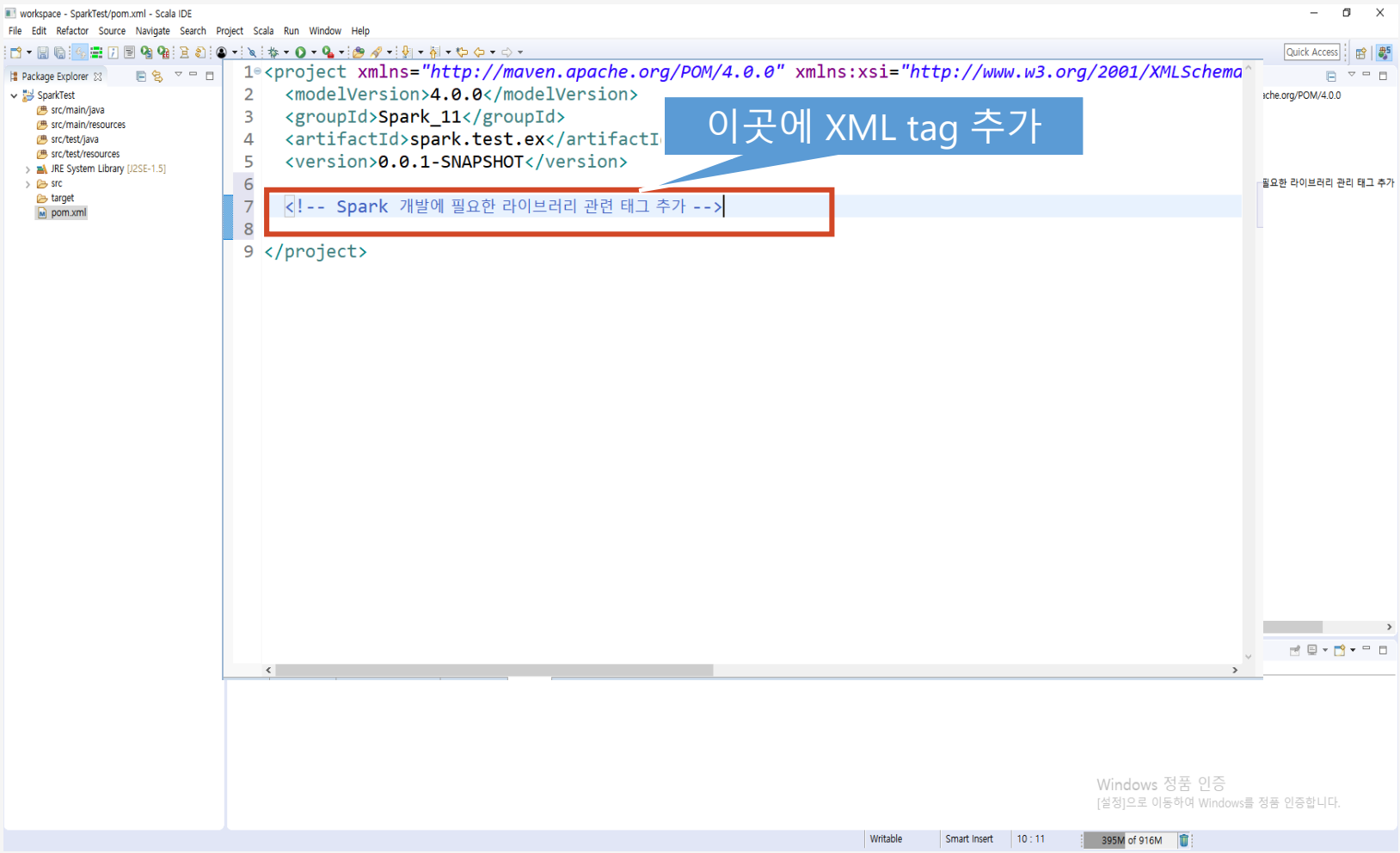


1) pox.xml 설정

Salac, Spark 개발에 필요한

라이브러리 관련 설정

태그 추가



1) pox.xml 설정

라이브러리 관련 설정

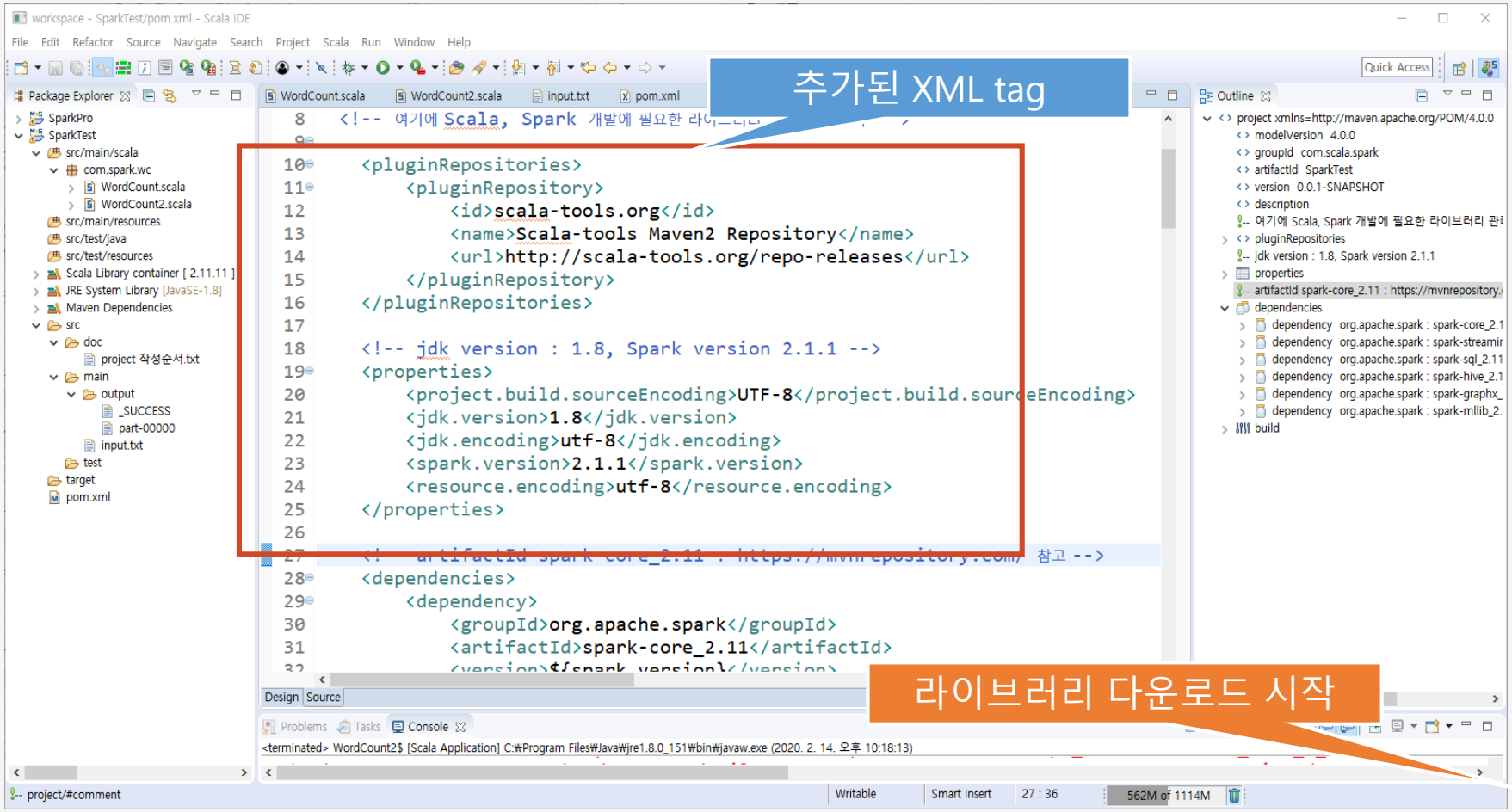
태그를 추가한 후

저장(Save)를 누르면

관련 라이브러리가

다운로드를 시작

Building workspace(100%) 될 때 까지 대기

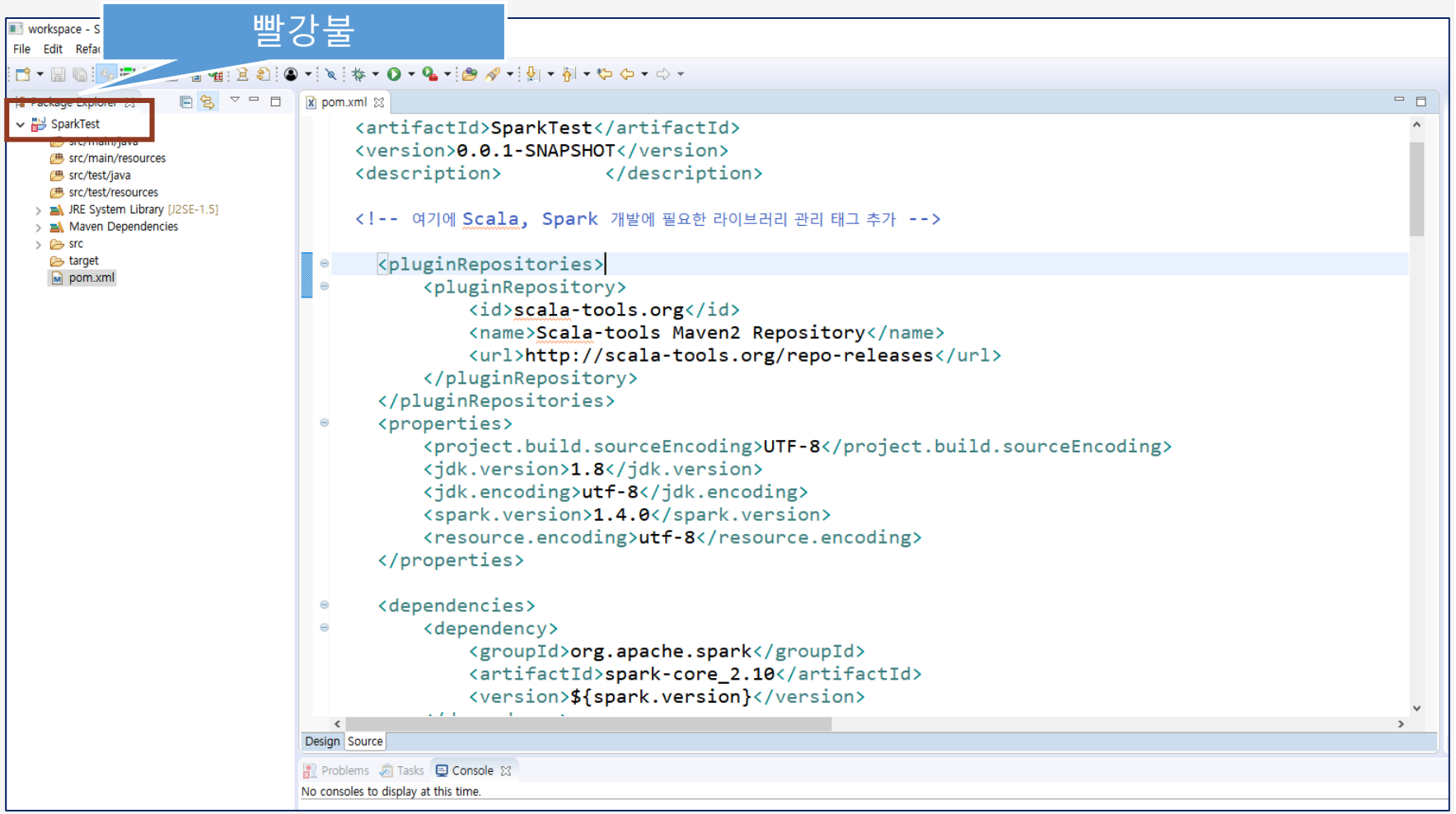


1) pox.xml 설정

[Project] 마우스 오른 버튼

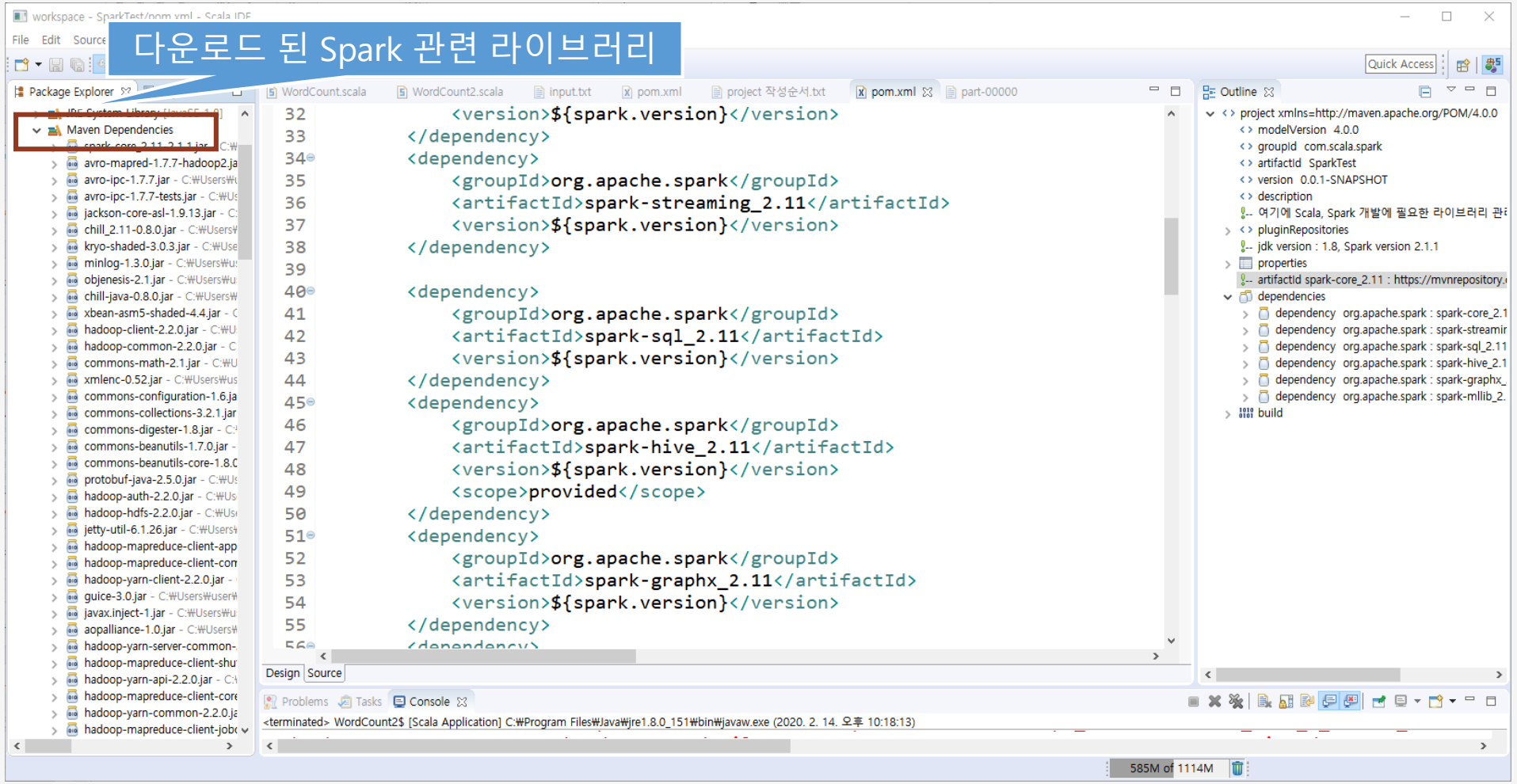
[Maven] -> [Update Project]

메뉴 선택으로 빨강불 제거



1) pox.xml 설정

다운로드 라이브러리 확인

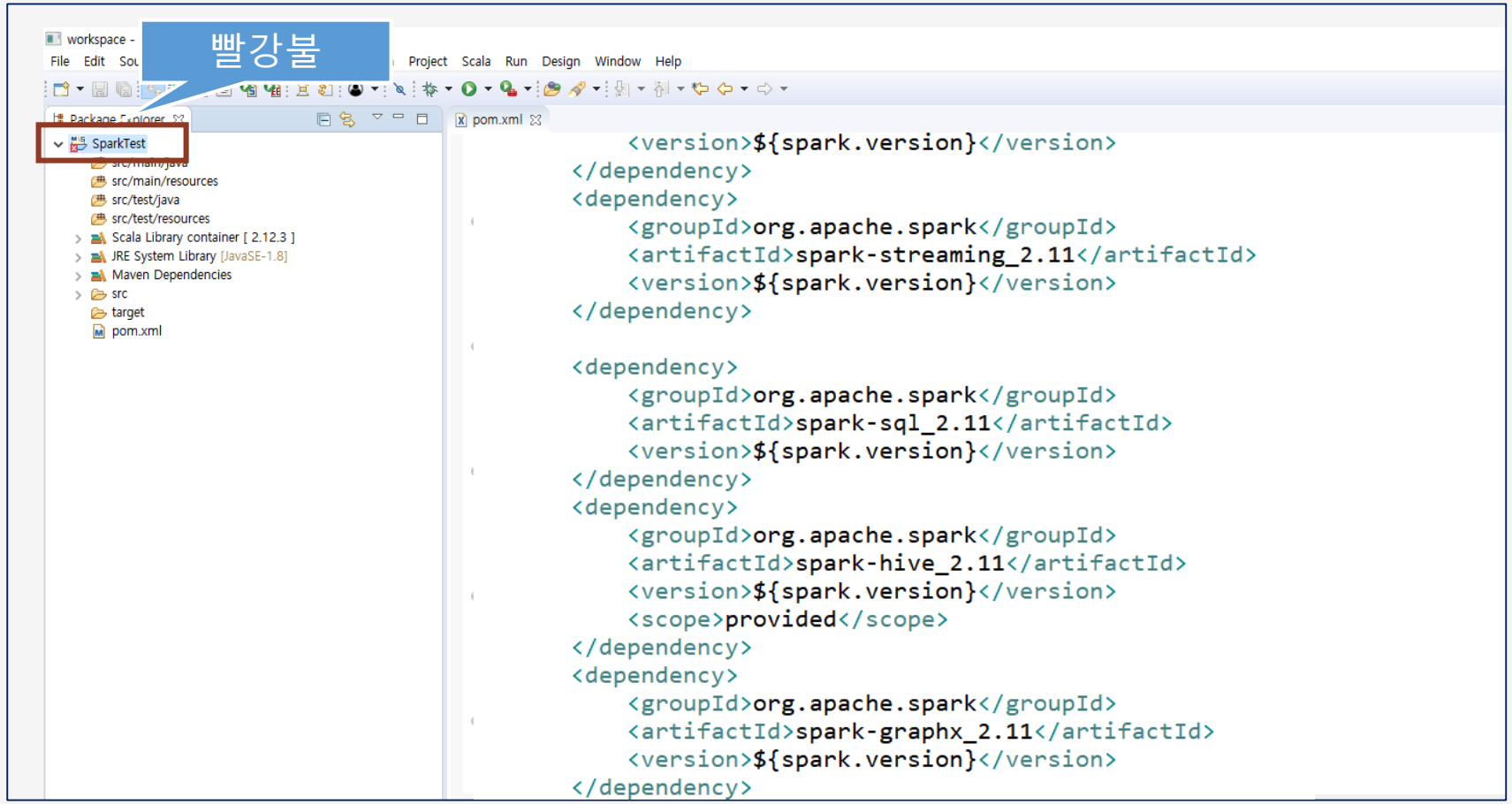


2) Scala Nature 추가 : Scala 환경으로 수정

Project에서 오른쪽 마우스 버튼

Configure -> Add Scala Nature

메뉴 선택 -> Scala Nature를 추가

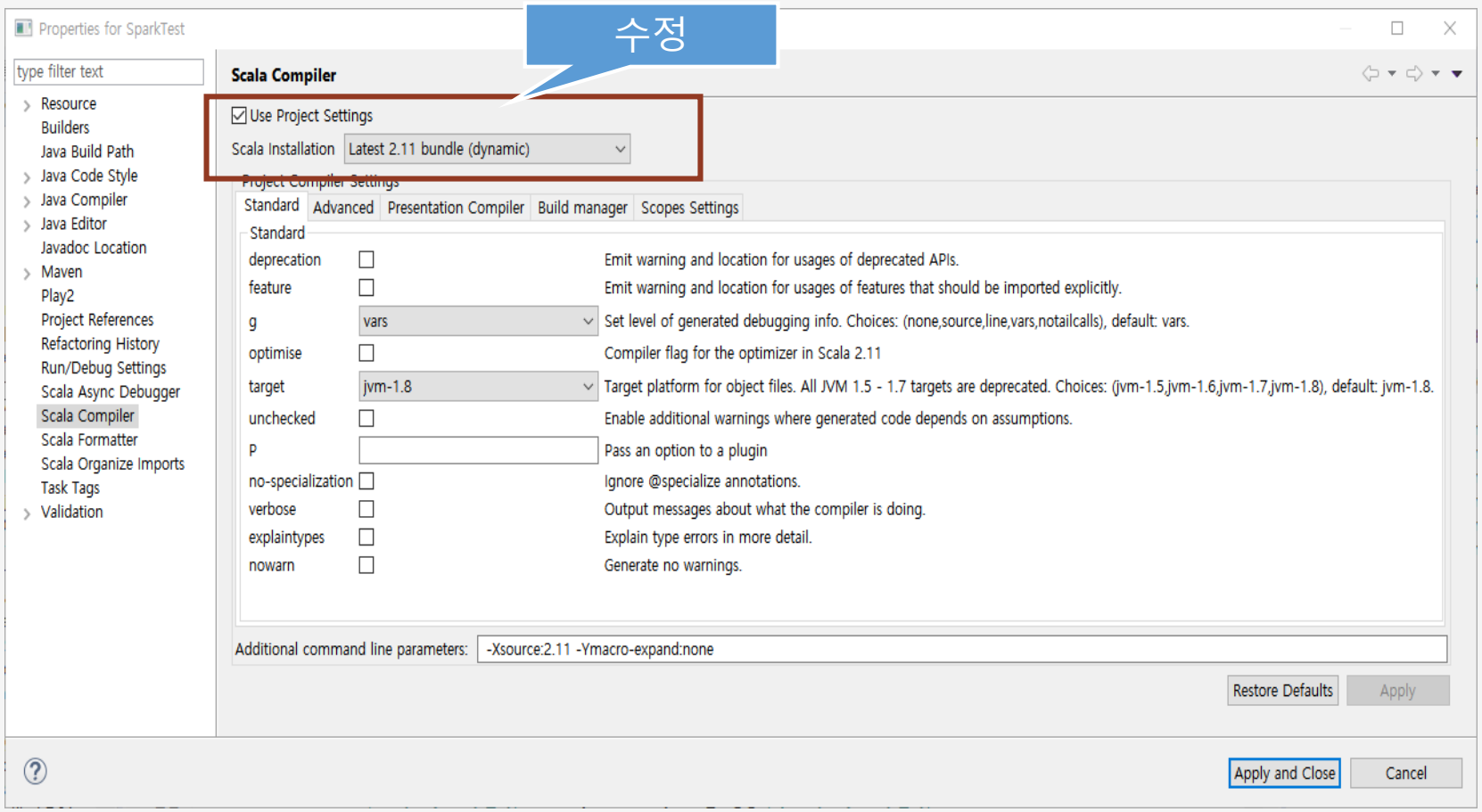


2) Scala Nature 추가

Project에서 오른쪽 마우스 버튼

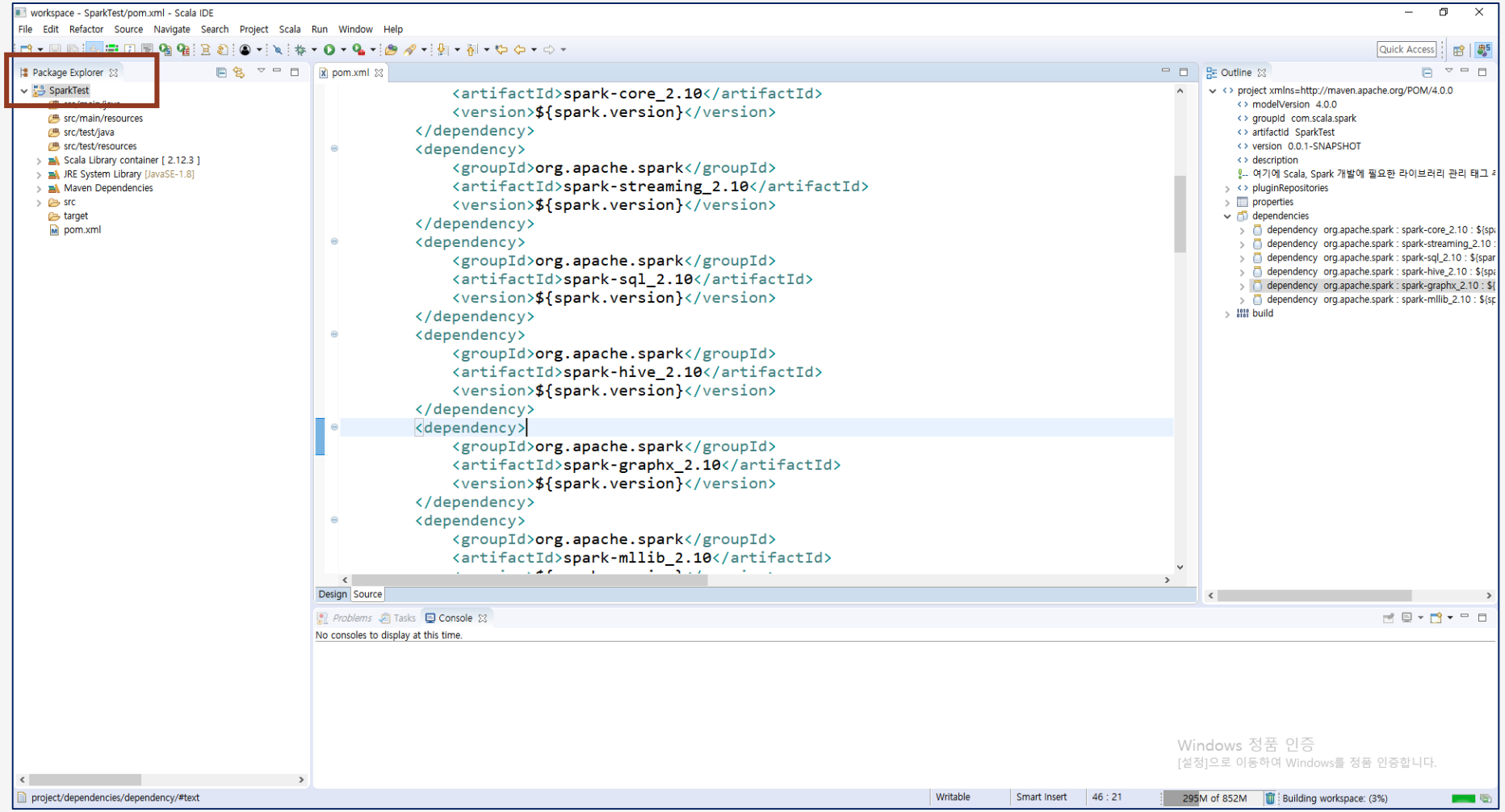
[Properties] -> [Scala compiler] 에서 `Latest 2.11 bundle (dynamic)` 선택

Scala IDE는 기본적으로 Lastest 2.12 bundle (dynamic)을 사용하고 있으나 안정화된 Scala 2.11을 사용하기 위해서 Compiler를 변경한다.



2) Scala Nature 추가

빨강불 제거 확인

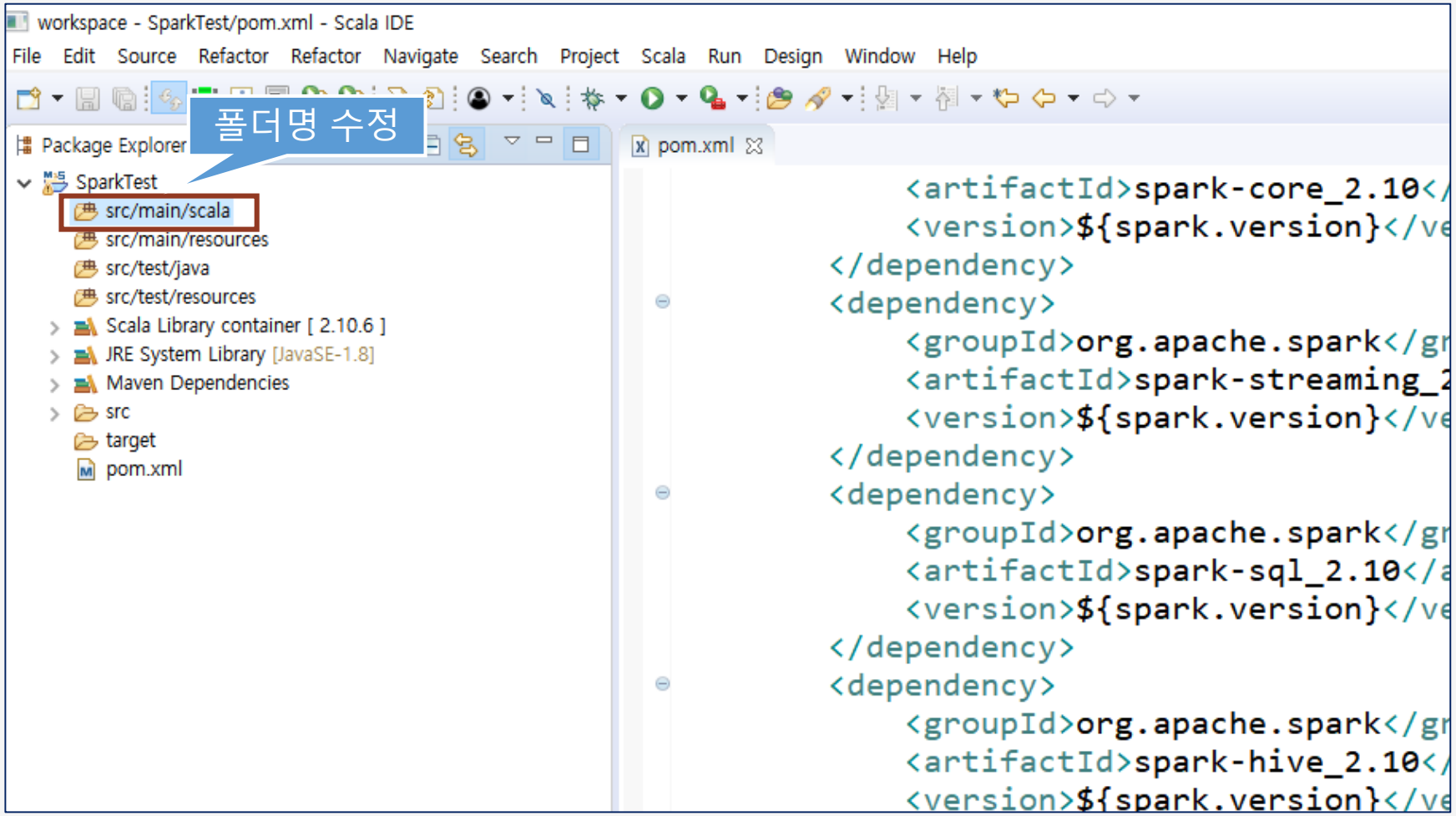


3) java -> scala 수정

폴더에서 오른쪽 마우스 버튼 클릭

-> Refactor -> Rename 선택

scala로 변경



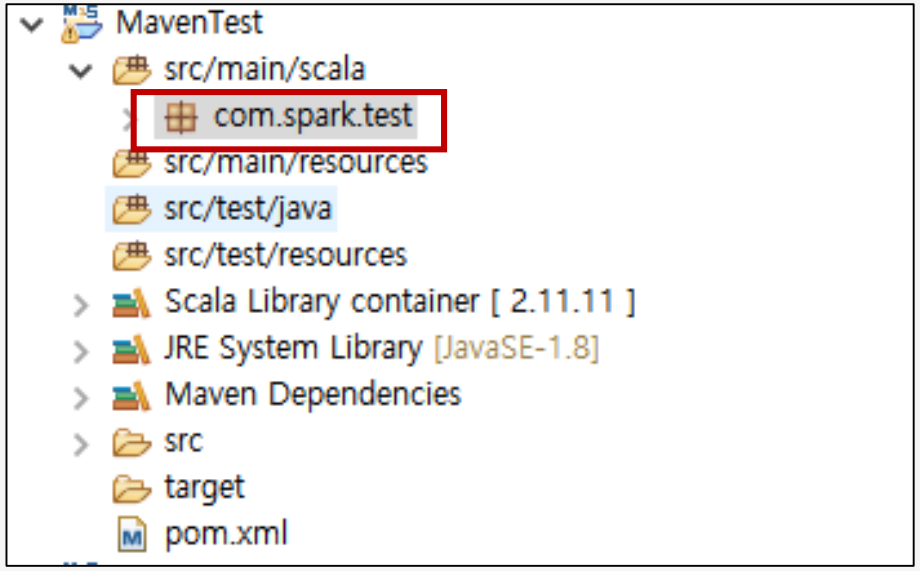
3. Spark 애플리케이션 작성

1) 패키지 만들기

scala 폴더에서 오른쪽 마우스 버튼 클릭

-> New -> Package 선택

-> com.spark.test 패키지 만들기

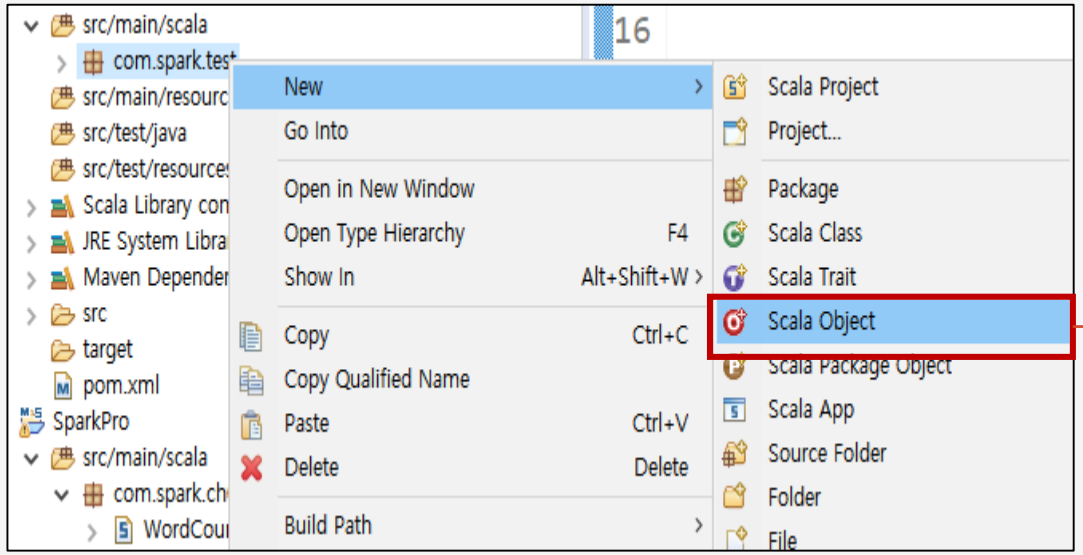


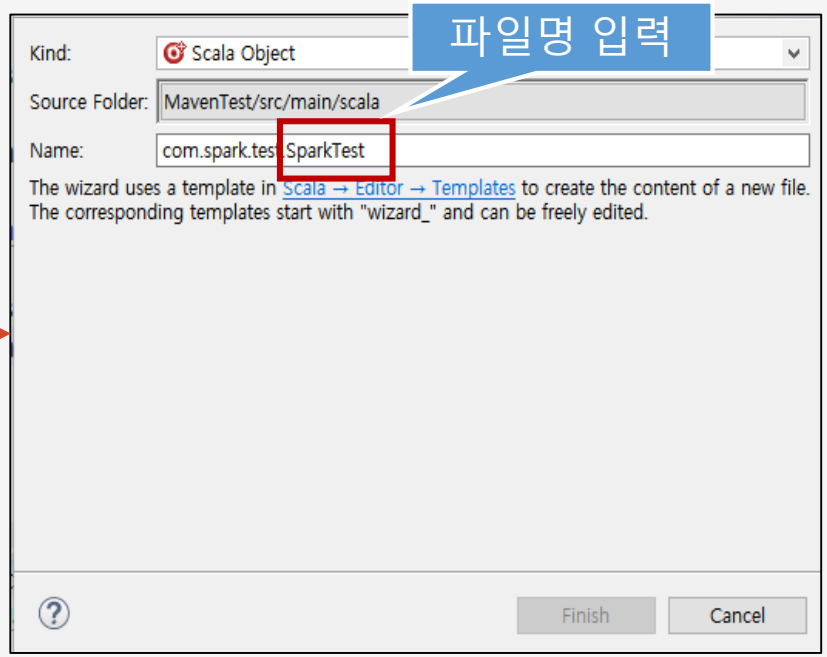
2) Scala Object 만들기

com.spark.wc 패키지에서 오른쪽 마우스 버튼 클릭

-> New -> Scala Object 선택

-> SparkTest 파일명 입력



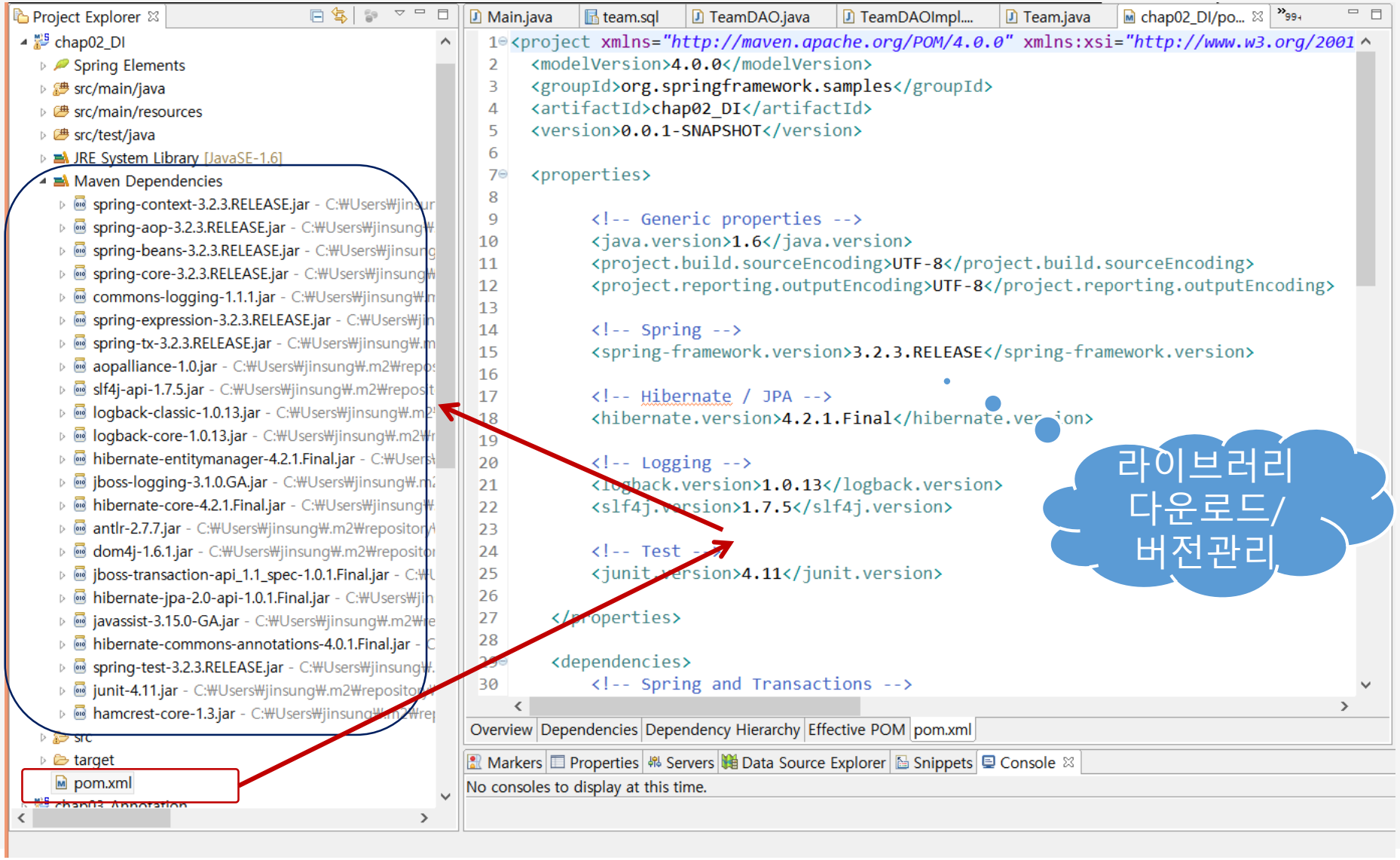


3) SparkTest 어플리케이션 작성

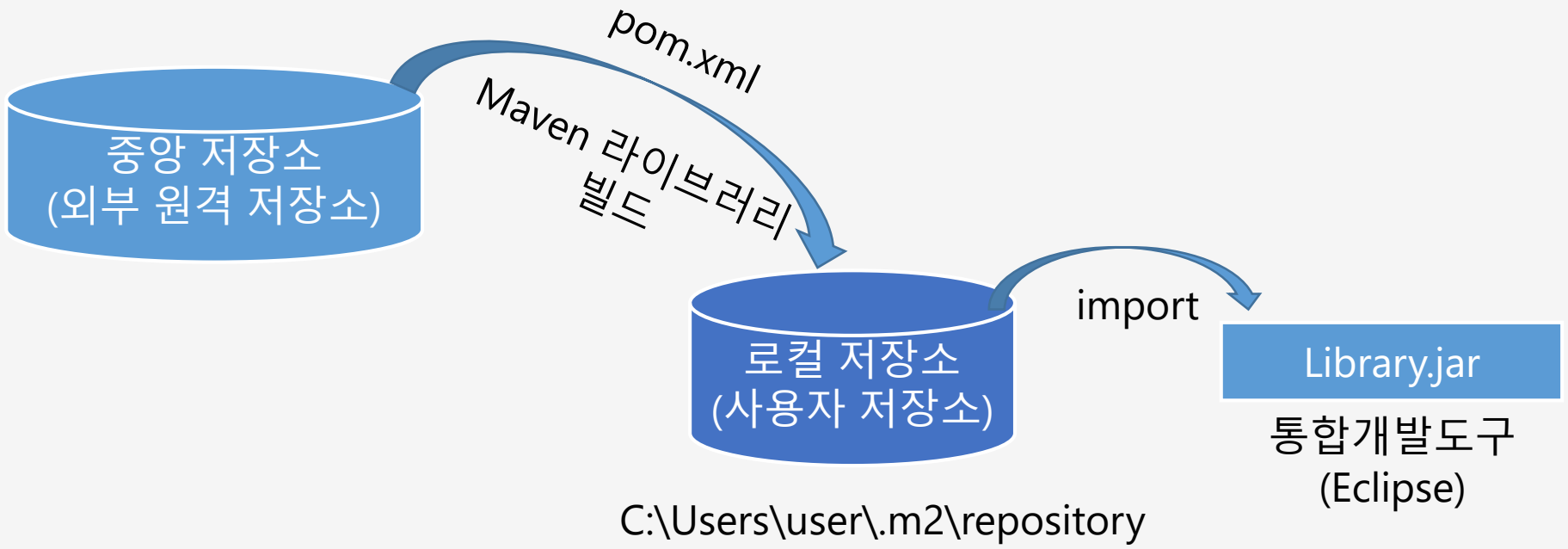
```scala
package com.spark.test

// Maven에서 제공하는 library 추가
import org.apache.spark.SparkConf
import org.apache.spark.SparkContext

object SparkTest {
    def main(args: Array[String]) = {

      // 1. SparkContext object 생성
      val conf = new SparkConf()
        .setAppName("SparkTest")
        .setMaster("local") // Spark 환경 객체

      val sc = new SparkContext(conf) // 분산 파일 읽기

      // 2. Local or Hadoop file 읽기
      val data = sc.textFile("file:/C:/hadoop-2.6.0/README.txt")


      // 3. 줄단위 출력
      data.foreach(println)
      sc.stop // 객체 닫기
   }
}
```

```
For the latest information about Hadoop, please visit our website at:

   http://hadoop.apache.org/core/

and our wiki, at:

   http://wiki.apache.org/hadoop/

This distribution includes cryptographic software.  The country in
which you currently reside may have restrictions on the import,
possession, use, and/or re-export to another country, of
encryption software.  BEFORE using any encryption software, please
check your country's laws, regulations and policies concerning the
import, possession, or use, and re-export of encryption software, to
see if this is permitted.  See <http://www.wassenaar.org/> for more
information.

The U.S. Government Department of Commerce, Bureau of Industry and
Security (BIS), has classified this software as Export Commodity
Control Number (ECCN) 5D002.C.1, which includes information security
software using or performing cryptographic functions with asymmetric
algorithms.  The form and manner of this Apache Software Foundation
distribution makes it eligible for export under the License Exception
ENC Technology Software Unrestricted (TSU) exception (see the BIS
Export Administration Regulations, Section 740.13) for both object
code and source code.

The following provides more details on the included cryptographic
software:
  Hadoop Core uses the SSL libraries from the Jetty project written
by mortbay.org.
```

4. Maven 기반 라이브러리 자동 관리



Maven 저장소


pom.xml
Maven 라이브러리
빌드
중앙 저장소
(외부 원격 저장소)
로컬 저장소
(사용자 저장소)
C:\Users\user\.m2\repository
import
Library.jar
통합개발도구
(Eclipse)

Maven 저장소 설정

- Maven 저장소에 대한 설정

  - pom.xml 의 <repositories /> 에서 설정

  - 최상위 POM에는 중앙 저장소의 정보가 설정되어 있으므로, 중앙 저장소에서 제공하지 않는 라이브러리가 있다면 <repositories /> 내에 설정

  - Maven 라이브러리를 다운로드할 때 <repositories /> 에 설정되어 있는 저장소 순서로 진행

Maven 저장소 설정

Maven 라이브러리 관리

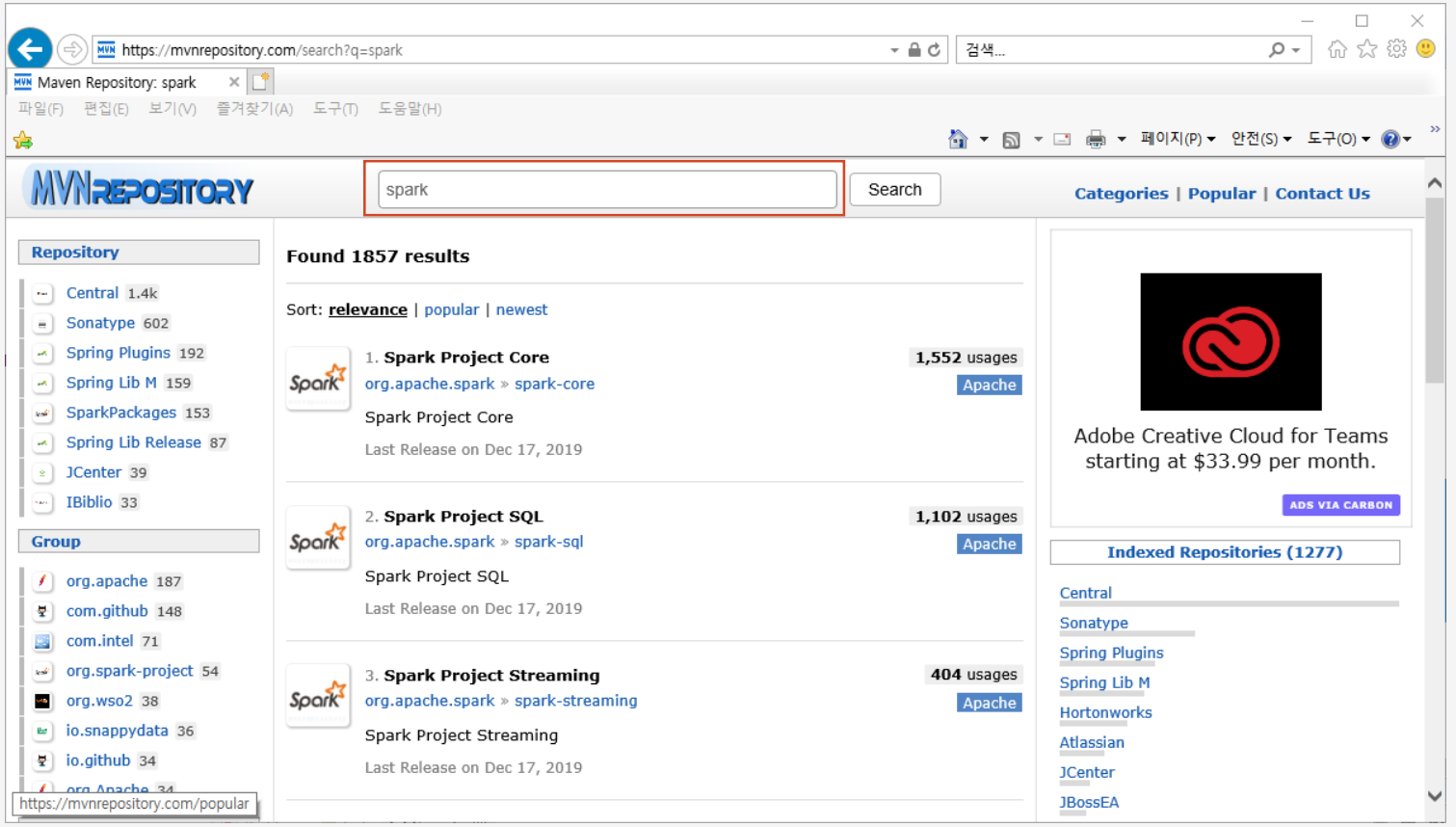
- pom.xml 의 <dependencies /> 설정을 통해서 필요한 라이브러리 로컬 저장소 다운로드

```
<dependencies>
    <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-core_2.11</artifactId>
            <version>${spark.version}</version>
      </dependency>
      <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-mllib_2.11</artifactId>
            <version>${spark.version}</version>
            <scope>runtime</scope>
      </dependency>
<dependencies>
```

pom.xml에 라이브러리 등록

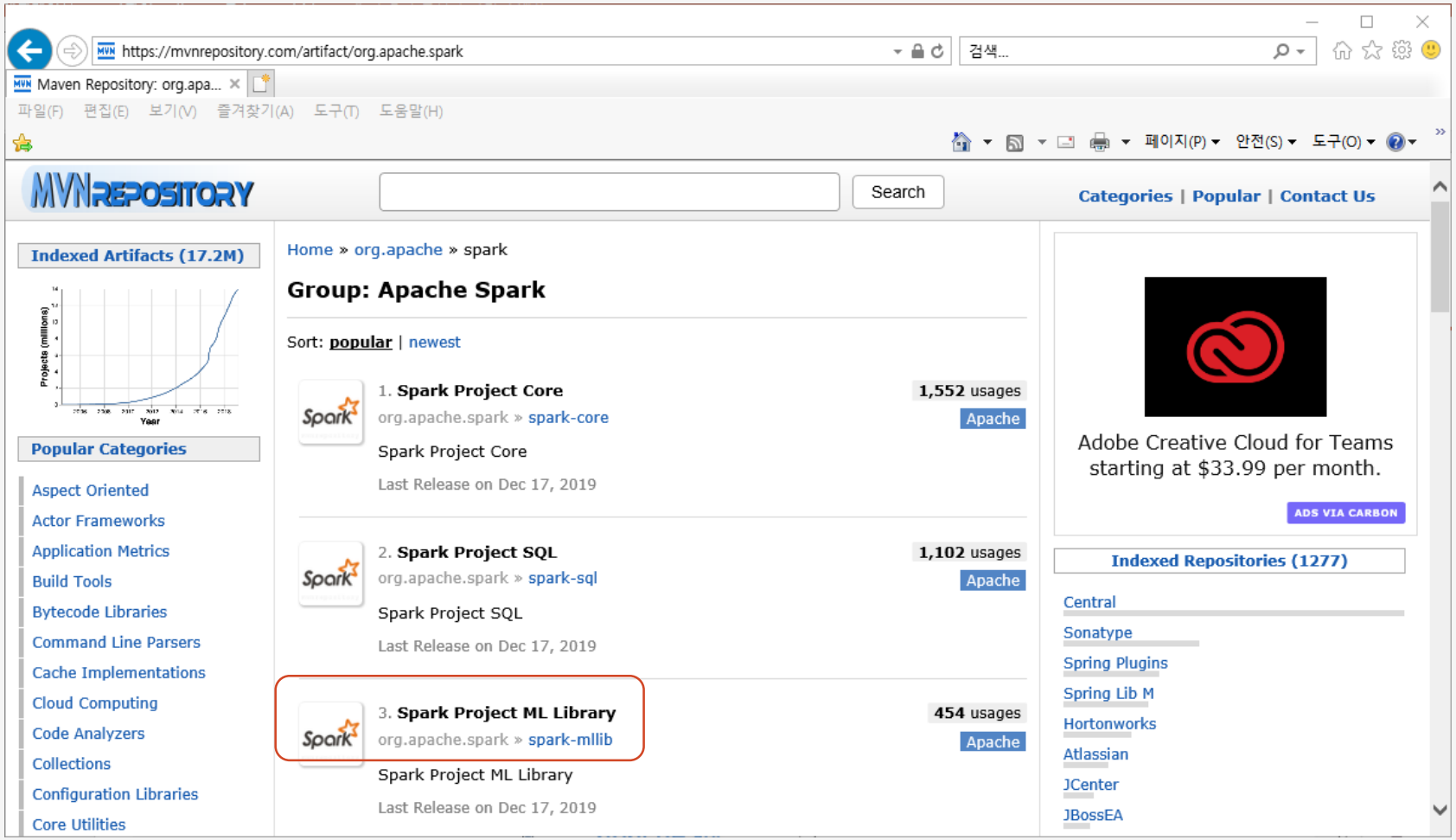
- Spark 라이브러리 등록 예

단계1: http://mvnrepository.com -> spark 검색어 입력



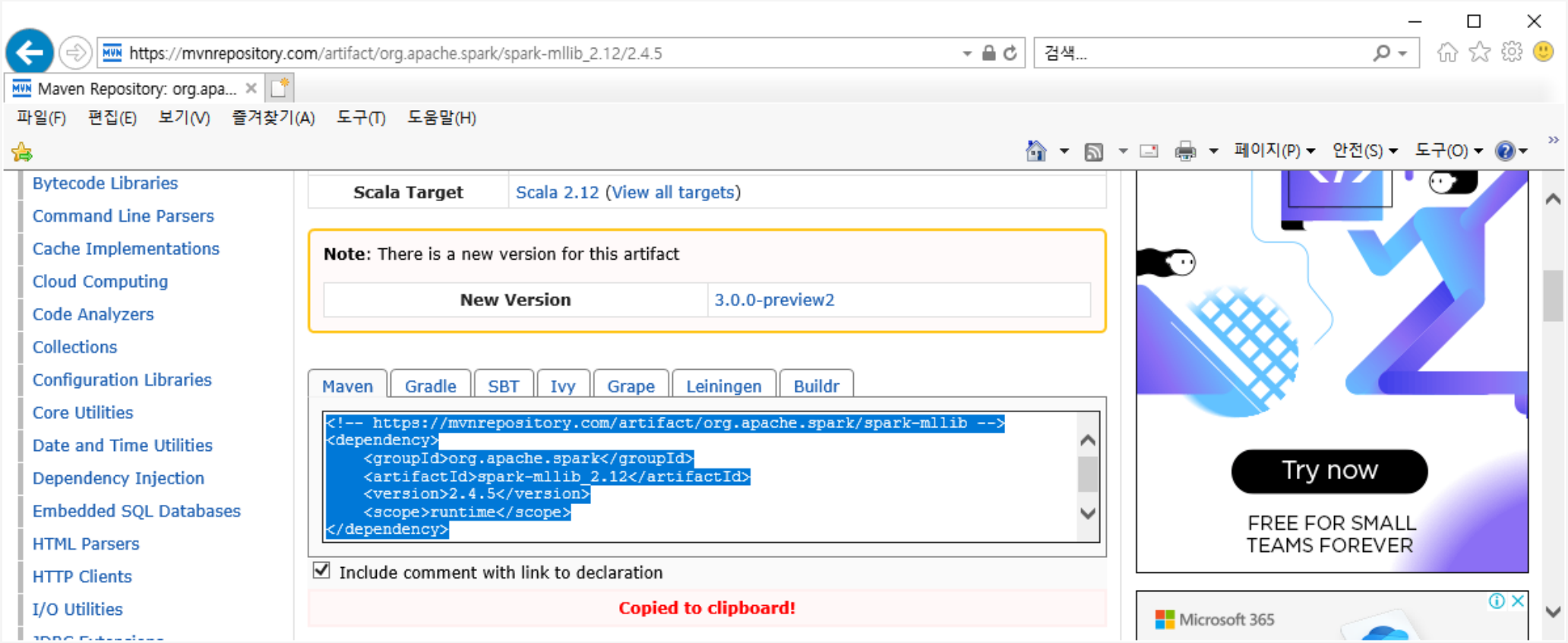
pom.xml에 라이브러리 등록

단계2: spark-mllib 선택



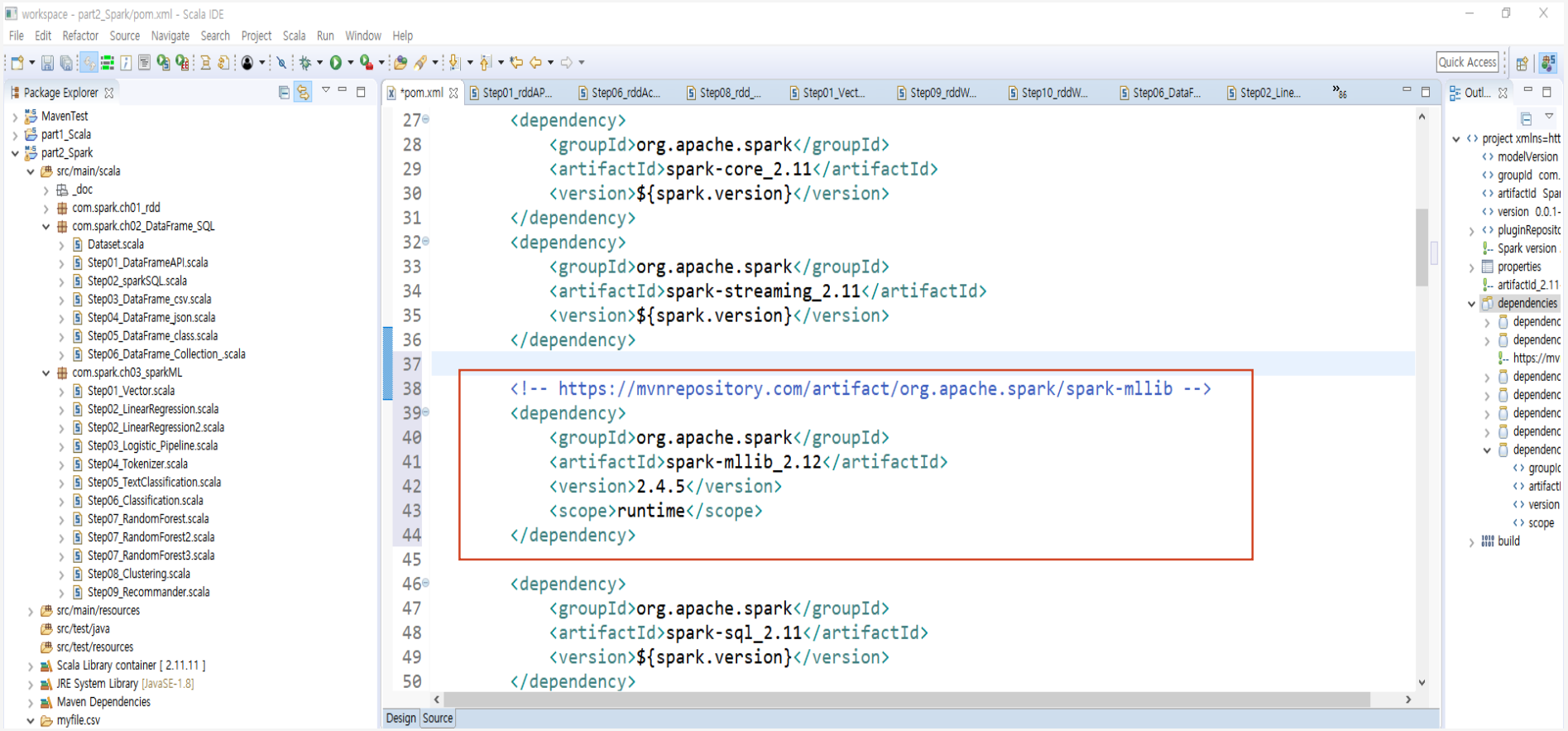
pom.xml에 라이브러리 등록

단계3: groupId, artifactId, version 정보 등 확인



pom.xml에 라이브러리 등록

단계4: pox.xml에 복사 & 붙여넣기




```
27      <dependency>
28          <groupId>org.apache.spark</groupId>
29          <artifactId>spark-core_2.11</artifactId>
30          <version>${spark.version}</version>
31      </dependency>
32      <dependency>
33          <groupId>org.apache.spark</groupId>
34          <artifactId>spark-streaming_2.11</artifactId>
35          <version>${spark.version}</version>
36      </dependency>
37
38      <!-- https://mvnrepository.com/artifact/org.apache.spark/spark-mllib -->
39      <dependency>
40          <groupId>org.apache.spark</groupId>
41          <artifactId>spark-mllib_2.12</artifactId>
42          <version>2.4.5</version>
43          <scope>runtime</scope>
44      </dependency>
45
46      <dependency>
47          <groupId>org.apache.spark</groupId>
48          <artifactId>spark-sql_2.11</artifactId>
49          <version>${spark.version}</version>
50      </dependency>
```