2024-1 Software Engineering
HW2
22100579 Jinju Lee

1. (1p) Select two open-source projects from ASF java projects in Github.

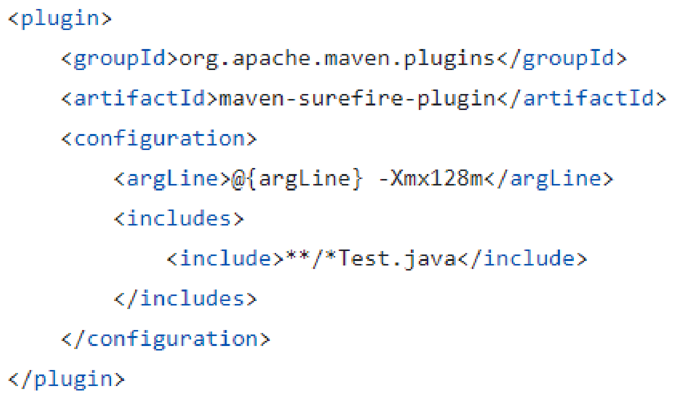
List the selected project name here:

1. Common-jexl https://github.com/apache/commons-jexl.git
2. Cassandra: https://github.com/apache/cassandra

2. (1p) What build tools were used for your selected projects? (You can identify build tools for the selected projects by reading their project web sites. For Java projects, Maven (pom.xml), Gradle (build.graldle), and Ant (*.xml) are popular ones. You may check other build tools.

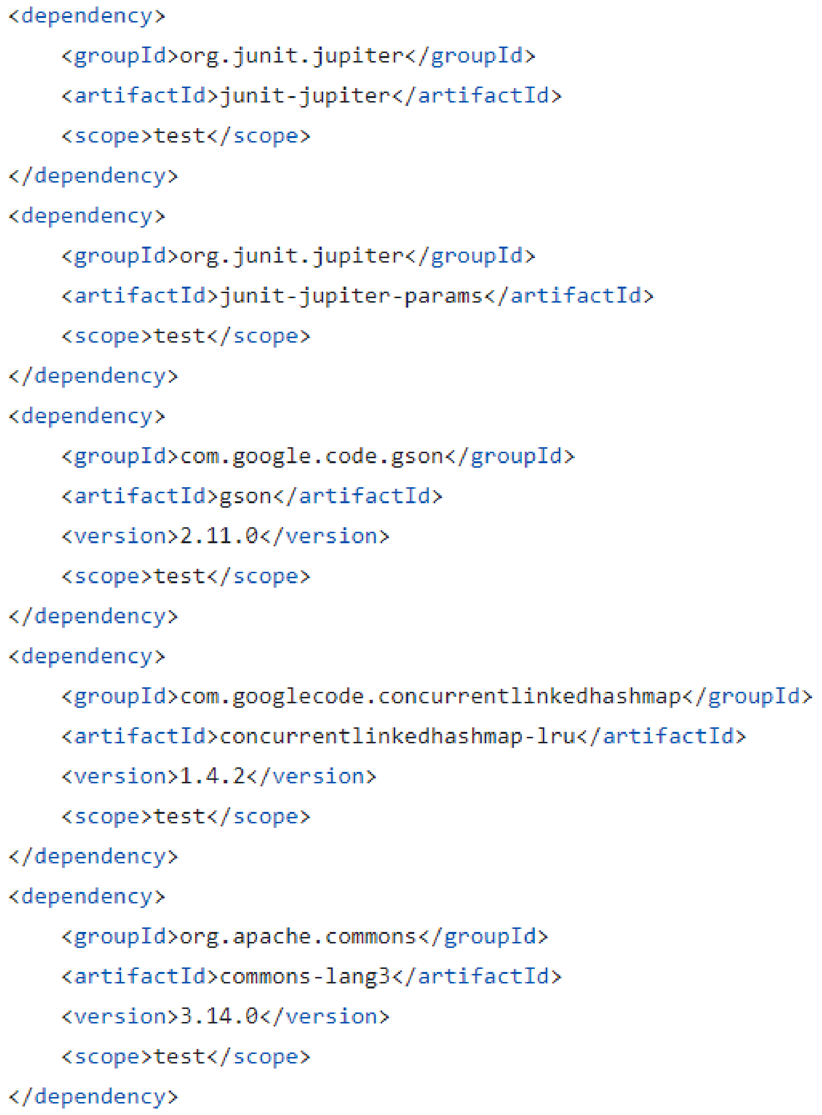
> I'd like to choose maven for common-jexl because the project has pom.xml file.

3. (2p) In a build configuration file, find a section for the automated/regression testing part. Find a build command that conducts the testing only.



```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-surefire-plugin</artifactId>
    <configuration>
        <argLine>@{argLine} -Xmx128m</argLine>
        <includes>
            <include>**/*Test.java</include>
        </includes>
    </configuration>
</plugin>
```

Surefire plugin is using for unit testing. This section is setting about surefire.



```
<dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter</artifactId>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-params</artifactId>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.11.0</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>com.googlecode.concurrentlinkedhashmap</groupId>
    <artifactId>concurrentlinkedhashmap-lru</artifactId>
    <version>1.4.2</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
    <version>3.14.0</version>
    <scope>test</scope>
</dependency>
```

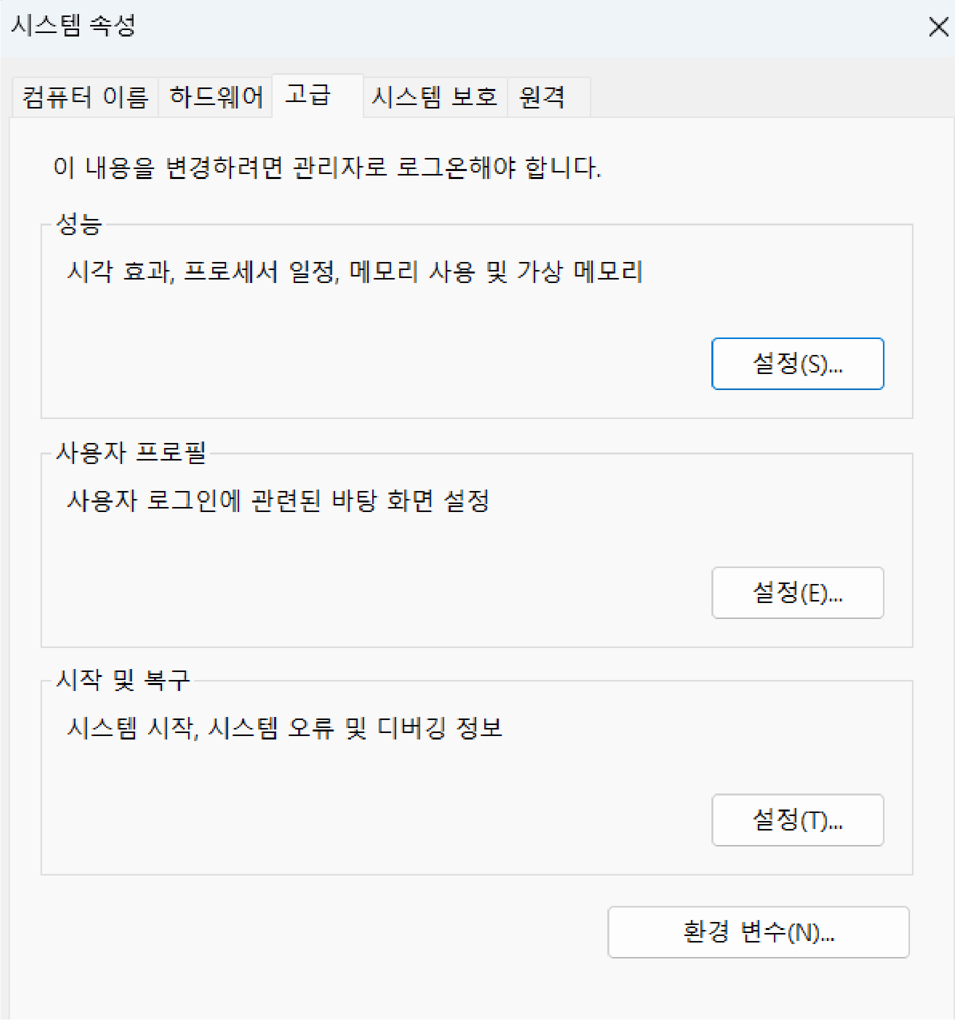
This section is for test building, I guess.

4. (2p) Build the selected projects. Put and explain the steps you have done for successful builds (including the steps installing the build tools). Capture the screens that show your build is successful. (Depending on each project, you may study basic commands of build tools by yourself. Some projects may provide guidelines or steps for how to build.)

1. install maven.

I am Windows user, so I downloaded zip file from https://maven.apache.org/download.cgi and unzip it in the directory I made.

After that, I need to add maven path into environment variable for using maven anywhere in my computer.



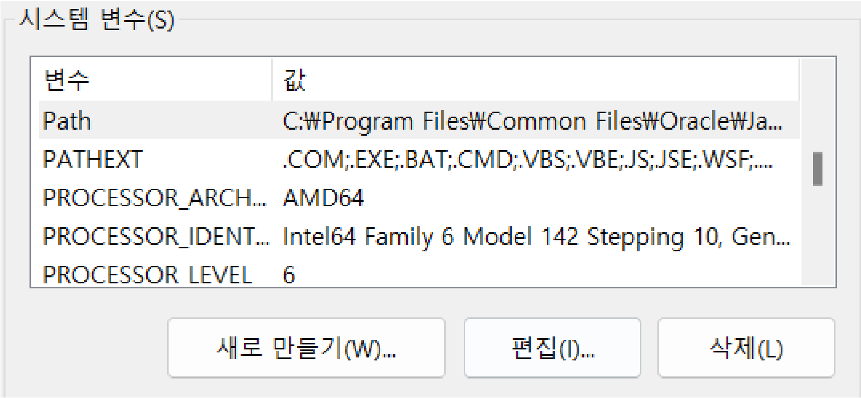
이 내용을 변경하려면 관리자로 로그온해야 합니다.

시각 효과, 프로세서 일정, 메모리 사용 및 가상 메모리

사용자 로그인에 관련된 바탕 화면 설정

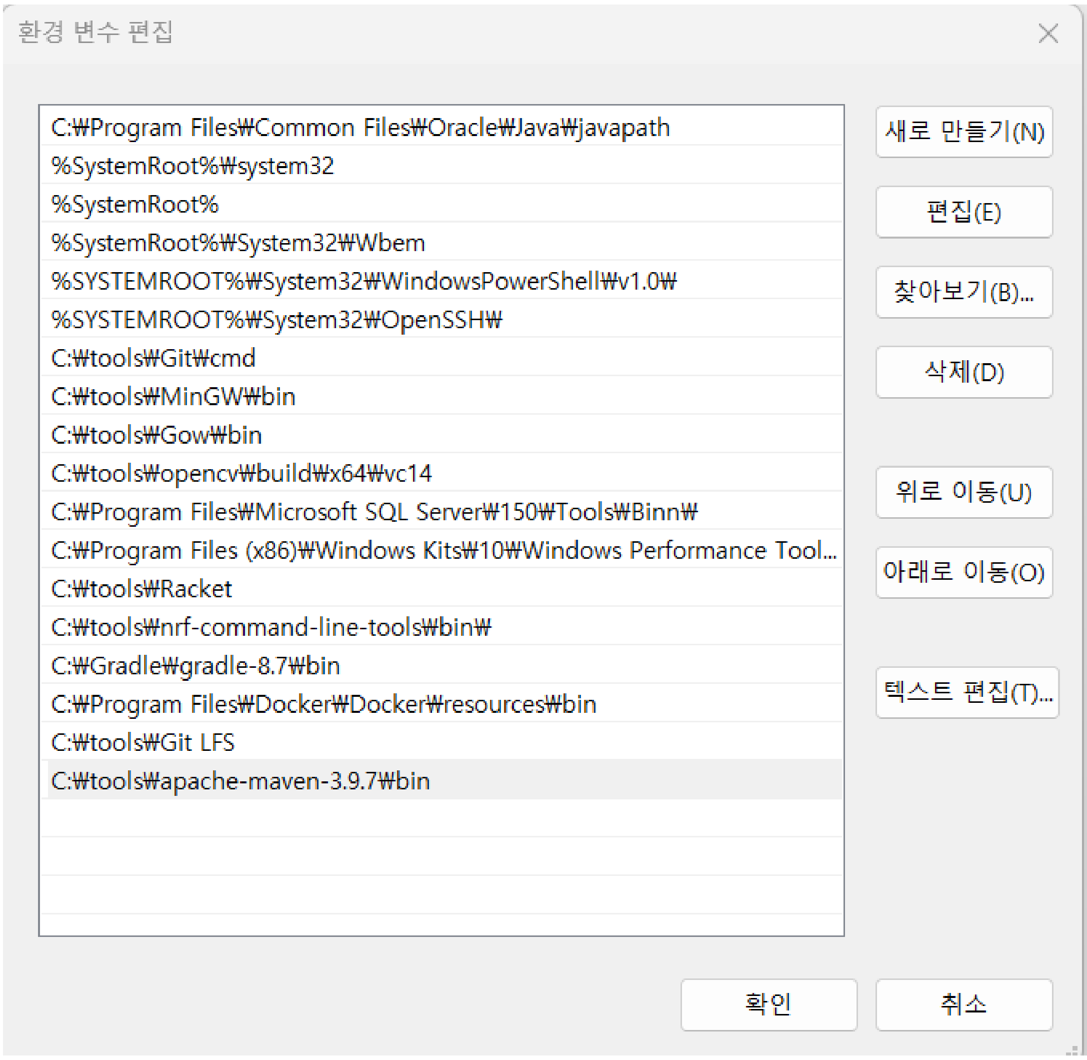
시스템 시작, 시스템 오류 및 디버깅 정보

In here, click 환경변수 button. It means environment variable.



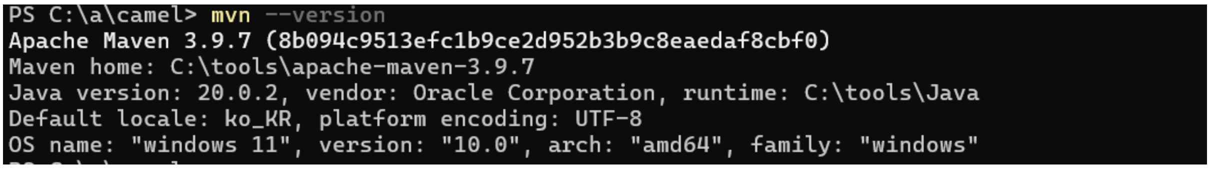
변수	값
Path	C:₩Program Files₩Common Files₩Oracle₩Ja...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;....
PROCESSOR_ARCH...	AMD64
PROCESSOR_IDENT...	Intel64 Family 6 Model 142 Stepping 10, Gen...
PROCESSOR LEVEL	6

Select Path variable and add new maven’s bin path on that.



```
C:₩Program Files₩Common Files₩Oracle₩Java₩javapath
%SystemRoot%₩system32
%SystemRoot%
%SystemRoot%₩System32₩Wbem
%SYSTEMROOT%₩System32₩WindowsPowerShell₩v1.0₩
%SYSTEMROOT%₩System32₩OpenSSH₩
C:₩tools₩Git₩cmd
C:₩tools₩MinGW₩bin
C:₩tools₩Gow₩bin
C:₩tools₩opencv₩build₩x64₩vc14
C:₩Program Files₩Microsoft SQL Server₩150₩Tools₩Binn₩
C:₩Program Files (x86)₩Windows Kits₩10₩Windows Performance Tool...
C:₩tools₩Racket
C:₩tools₩nrf-command-line-tools₩bin₩
C:₩Gradle₩gradle-8.7₩bin
C:₩Program Files₩Docker₩Docker₩resources₩bin
C:₩tools₩Git LFS
C:₩tools₩apache-maven-3.9.7₩bin
```

Now I can check if I can use maven anywhere by using this command

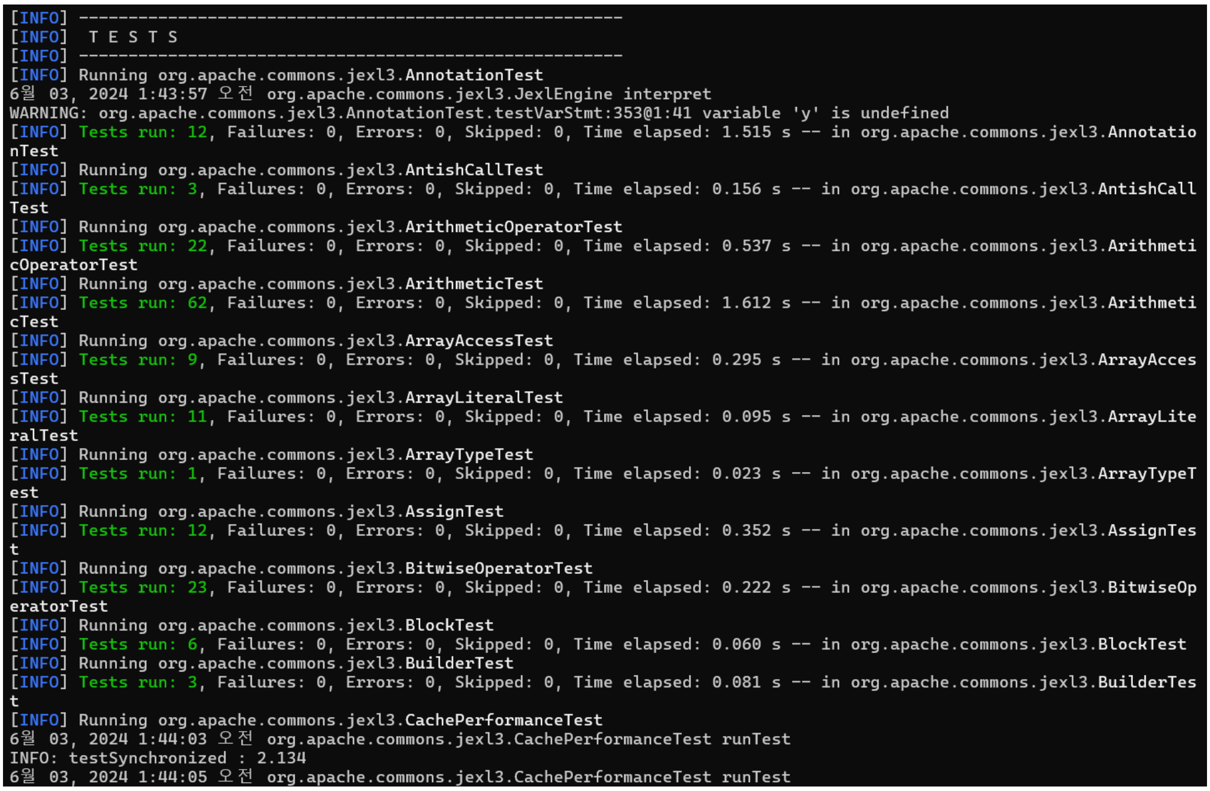


```
PS C:\a\camel> mvn --version
Apache Maven 3.9.7 (8b094c9513efc1b9ce2d952b3b9c8eaedaf8cbf0)
Maven home: C:\tools\apache-maven-3.9.7
Java version: 20.0.2, vendor: Oracle Corporation, runtime: C:\tools\Java
Default locale: ko_KR, platform encoding: UTF-8
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"
```

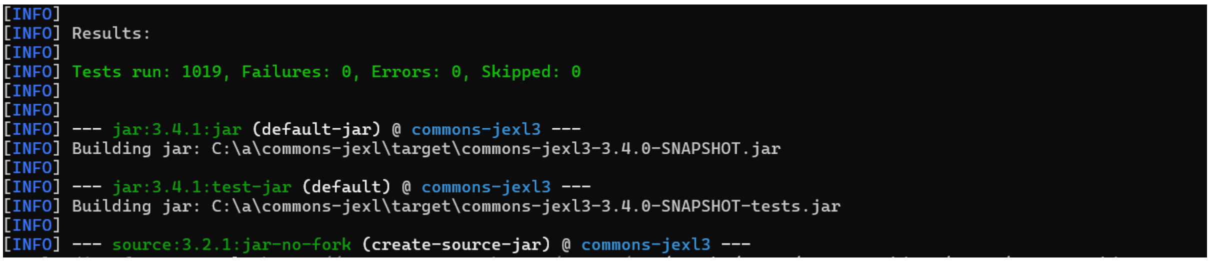
Ok. I am ready now!

2. Build using maven.

In this project, there is pom.xml file for build. So, I used ‘mvn clean install’ command to build it. I used sudo shell for let maven download what I need.

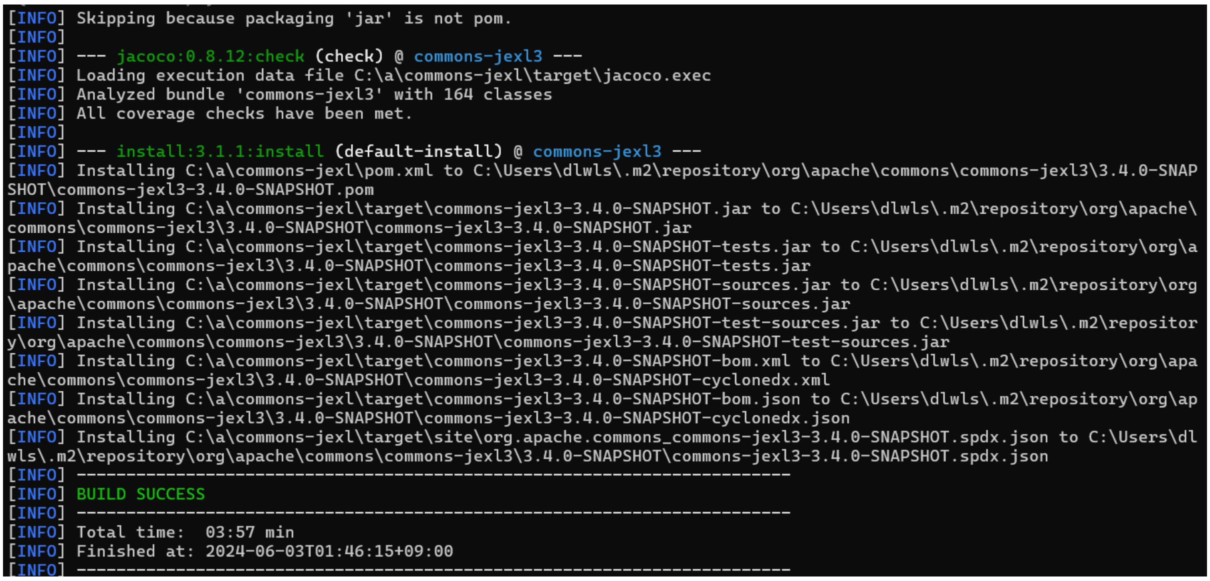


```
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running org.apache.commons.jexl3.AnnotationTest
6월 03, 2024 1:43:57 오전 org.apache.commons.jexl3.JexlEngine interpret
WARNING: org.apache.commons.jexl3.AnnotationTest.testVarStmt:353@1:41 variable 'y' is undefined
[INFO] Tests run: 12, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.515 s -- in org.apache.commons.jexl3.Annotatio
nTest
[INFO] Running org.apache.commons.jexl3.AntishCallTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.156 s -- in org.apache.commons.jexl3.AntishCall
Test
[INFO] Running org.apache.commons.jexl3.ArithmeticOperatorTest
[INFO] Tests run: 22, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.537 s -- in org.apache.commons.jexl3.Arithmeti
cOperatorTest
[INFO] Running org.apache.commons.jexl3.ArithmeticTest
[INFO] Tests run: 62, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.612 s -- in org.apache.commons.jexl3.Arithmeti
cTest
[INFO] Running org.apache.commons.jexl3.ArrayAccessTest
[INFO] Tests run: 9, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.295 s -- in org.apache.commons.jexl3.ArrayAcces
sTest
[INFO] Running org.apache.commons.jexl3.ArrayLiteralTest
[INFO] Tests run: 11, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.095 s -- in org.apache.commons.jexl3.ArrayLite
ralTest
[INFO] Running org.apache.commons.jexl3.ArrayTypeTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.023 s -- in org.apache.commons.jexl3.ArrayTypeT
est
[INFO] Running org.apache.commons.jexl3.AssignTest
[INFO] Tests run: 12, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.352 s -- in org.apache.commons.jexl3.AssignTes
t
[INFO] Running org.apache.commons.jexl3.BitwiseOperatorTest
[INFO] Tests run: 23, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.222 s -- in org.apache.commons.jexl3.BitwiseOp
eratorTest
[INFO] Running org.apache.commons.jexl3.BlockTest
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.060 s -- in org.apache.commons.jexl3.BlockTest
[INFO] Running org.apache.commons.jexl3.BuilderTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.081 s -- in org.apache.commons.jexl3.BuilderTes
t
[INFO] Running org.apache.commons.jexl3.CachePerformanceTest
6월 03, 2024 1:44:03 오전 org.apache.commons.jexl3.CachePerformanceTest runTest
INFO: testSynchronized : 2.134
6월 03, 2024 1:44:05 오전 org.apache.commons.jexl3.CachePerformanceTest runTest
```



```
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1019, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ commons-jexl3 ---
[INFO] Building jar: C:\a\commons-jexl\target\commons-jexl3-3.4.0-SNAPSHOT.jar
[INFO]
[INFO] --- jar:3.4.1:test-jar (default) @ commons-jexl3 ---
[INFO] Building jar: C:\a\commons-jexl\target\commons-jexl3-3.4.0-SNAPSHOT-tests.jar
[INFO]
[INFO] --- source:3.2.1:jar-no-fork (create-source-jar) @ commons-jexl3 ---
```

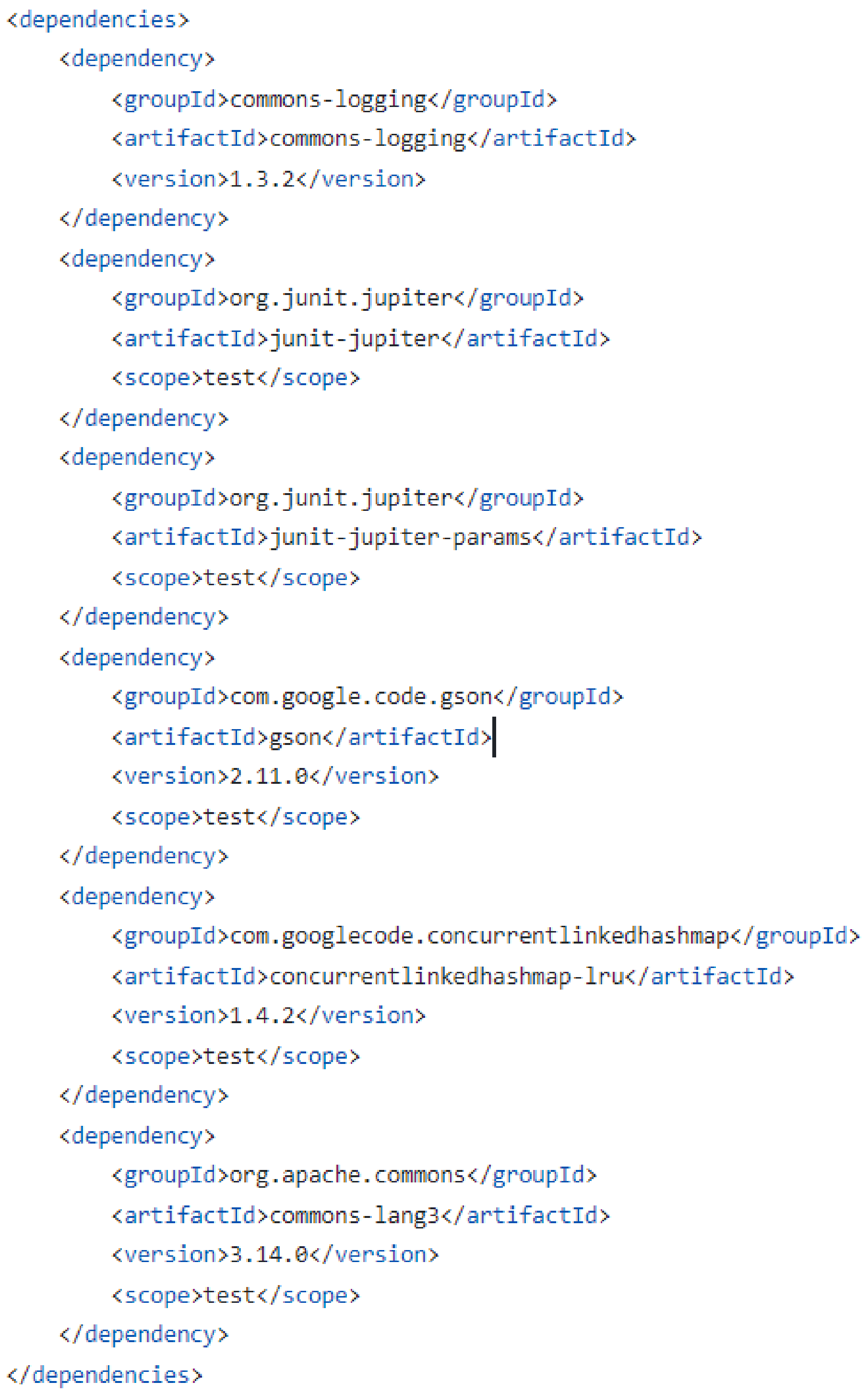
...



```
[INFO] Skipping because packaging 'jar' is not pom.
[INFO]
[INFO] --- jacoco:0.8.12:check (check) @ commons-jexl3 ---
[INFO] Loading execution data file C:\a\commons-jexl\target\jacoco.exec
[INFO] Analyzed bundle 'commons-jexl3' with 164 classes
[INFO] All coverage checks have been met.
[INFO]
[INFO] --- install:3.1.1:install (default-install) @ commons-jexl3 ---
[INFO] Installing C:\a\commons-jexl\pom.xml to C:\Users\dlwls\.m2\repository\org\apache\commons\commons-jexl3\3.4.0-SNAP
SHOT\commons-jexl3-3.4.0-SNAPSHOT.pom
[INFO] Installing C:\a\commons-jexl\target\commons-jexl3-3.4.0-SNAPSHOT.jar to C:\Users\dlwls\.m2\repository\org\apache\
commons\commons-jexl3\3.4.0-SNAPSHOT\commons-jexl3-3.4.0-SNAPSHOT.jar
[INFO] Installing C:\a\commons-jexl\target\commons-jexl3-3.4.0-SNAPSHOT-tests.jar to C:\Users\dlwls\.m2\repository\org\a
pache\commons\commons-jexl3\3.4.0-SNAPSHOT\commons-jexl3-3.4.0-SNAPSHOT-tests.jar
[INFO] Installing C:\a\commons-jexl\target\commons-jexl3-3.4.0-SNAPSHOT-sources.jar to C:\Users\dlwls\.m2\repository\org
\apache\commons\commons-jexl3\3.4.0-SNAPSHOT\commons-jexl3-3.4.0-SNAPSHOT-sources.jar
[INFO] Installing C:\a\commons-jexl\target\commons-jexl3-3.4.0-SNAPSHOT-test-sources.jar to C:\Users\dlwls\.m2\repositor
y\org\apache\commons\commons-jexl3\3.4.0-SNAPSHOT\commons-jexl3-3.4.0-SNAPSHOT-test-sources.jar
[INFO] Installing C:\a\commons-jexl\target\commons-jexl3-3.4.0-SNAPSHOT-bom.xml to C:\Users\dlwls\.m2\repository\org\apa
che\commons\commons-jexl3\3.4.0-SNAPSHOT\commons-jexl3-3.4.0-SNAPSHOT-cyclonedx.xml
[INFO] Installing C:\a\commons-jexl\target\commons-jexl3-3.4.0-SNAPSHOT-bom.json to C:\Users\dlwls\.m2\repository\org\ap
ache\commons\commons-jexl3\3.4.0-SNAPSHOT\commons-jexl3-3.4.0-SNAPSHOT-cyclonedx.json
[INFO] Installing C:\a\commons-jexl\target\site\org.apache.commons_commons-jexl3-3.4.0-SNAPSHOT.spdx.json to C:\Users\dl
wls\.m2\repository\org\apache\commons\commons-jexl3\3.4.0-SNAPSHOT\commons-jexl3-3.4.0-SNAPSHOT.spdx.json
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  03:57 min
[INFO] Finished at: 2024-06-03T01:46:15+09:00
[INFO] ------------------------------------------------------------------------
```

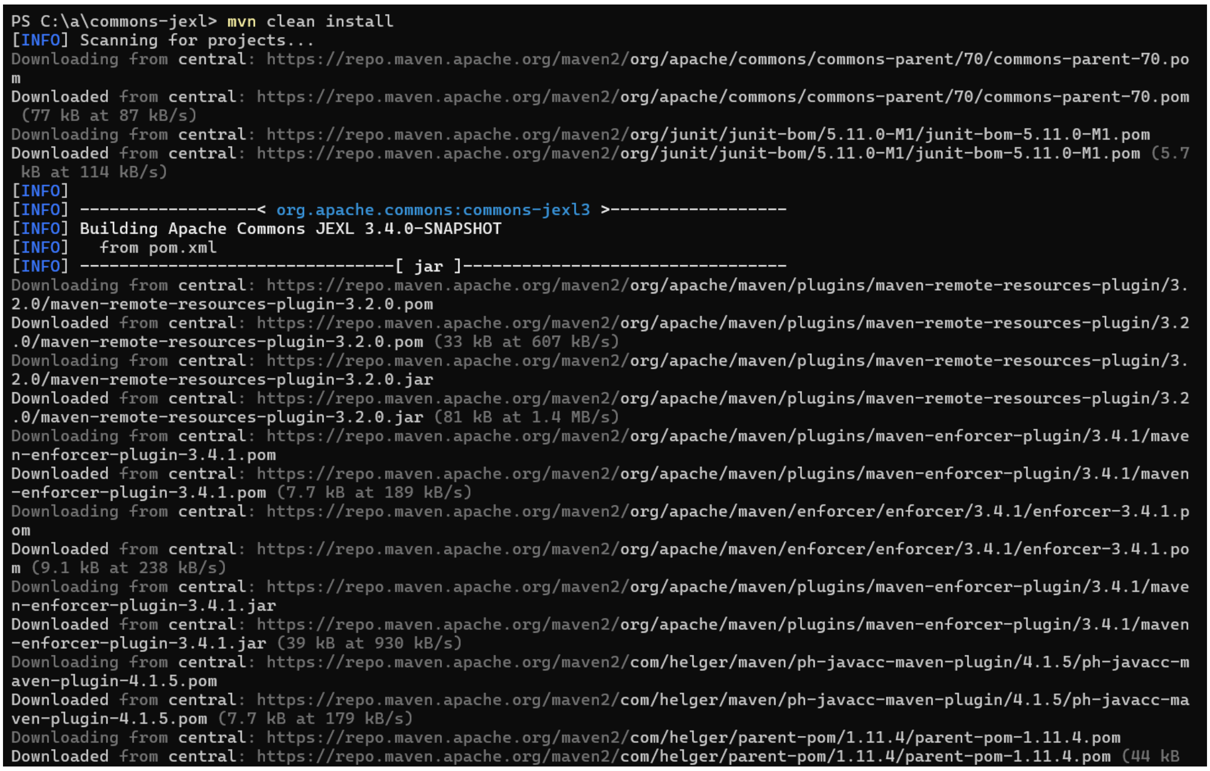
5. (1p) One of the useful functions of build tools is automatically importing or downloading the external libraries/packages/components. Find and put (copy and paste or a screenshot) (1) a section for external libraries from the configuration file and (2) the part of build execution logs that import or download the external libraries.

1. Section for external libraries are <dependencies> part in pom.xml

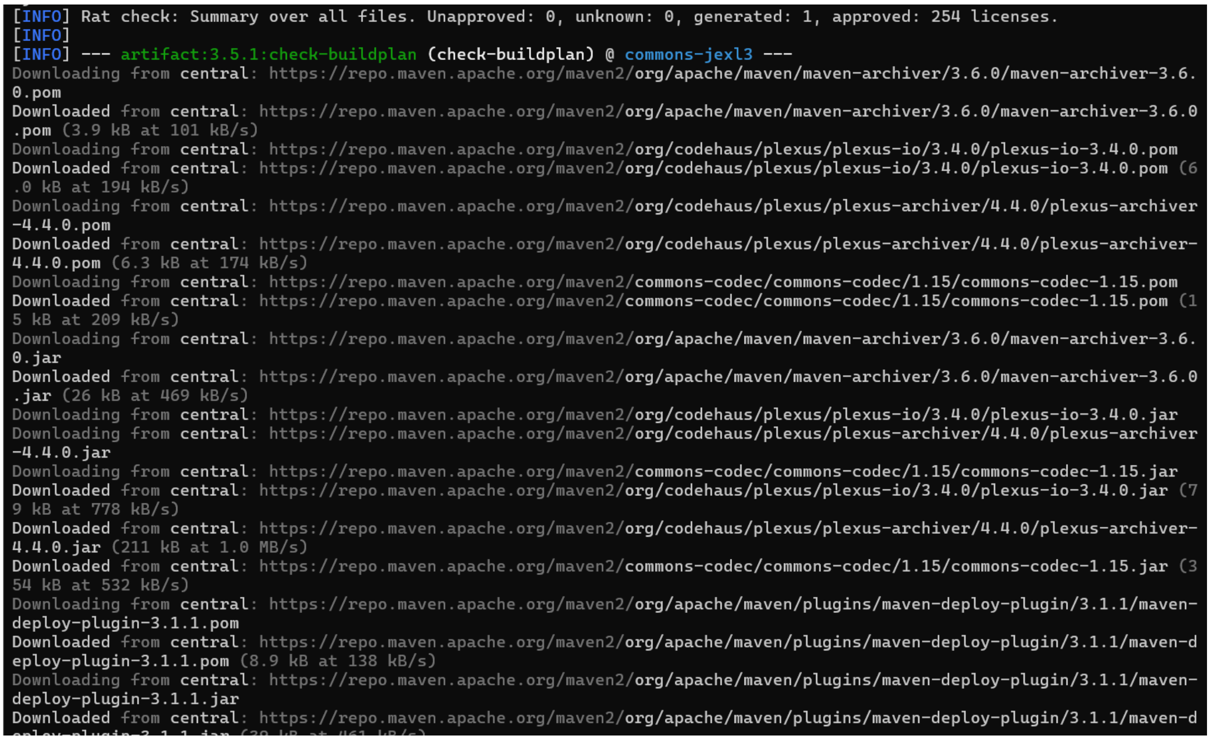


```
<dependencies>
    <dependency>
        <groupId>commons-logging</groupId>
        <artifactId>commons-logging</artifactId>
        <version>1.3.2</version>
    </dependency>
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter</artifactId>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter-params</artifactId>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>com.google.code.gson</groupId>
        <artifactId>gson</artifactId>
        <version>2.11.0</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>com.googlecode.concurrentlinkedhashmap</groupId>
        <artifactId>concurrentlinkedhashmap-lru</artifactId>
        <version>1.4.2</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>org.apache.commons</groupId>
        <artifactId>commons-lang3</artifactId>
        <version>3.14.0</version>
        <scope>test</scope>
    </dependency>
</dependencies>
```

2. and the part of build execution logs that import or download the external libraries are… almost whole part in logs, I guess. Its too long to capture all the logs, but it’s like this.



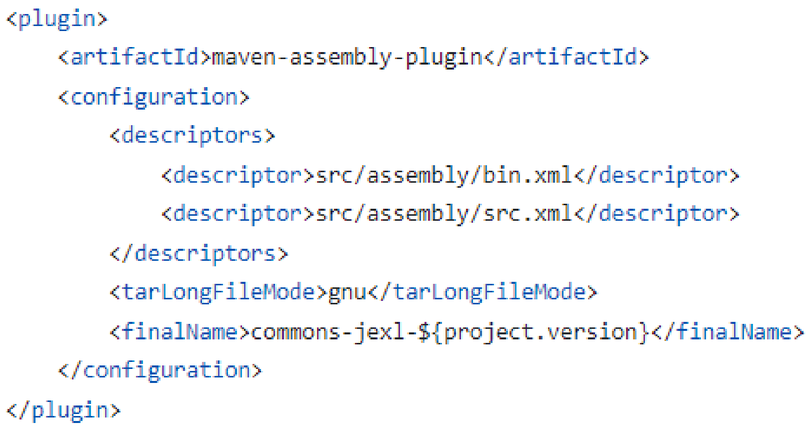
```
PS C:\a\commons-jexl> mvn clean install
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-parent/70/commons-parent-70.po
m
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-parent/70/commons-parent-70.pom
 (77 kB at 87 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/junit/junit-bom/5.11.0-M1/junit-bom-5.11.0-M1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/junit/junit-bom/5.11.0-M1/junit-bom-5.11.0-M1.pom (5.7
 kB at 114 kB/s)
[INFO]
[INFO] ------------------< org.apache.commons:commons-jexl3 >------------------
[INFO] Building Apache Commons JEXL 3.4.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-remote-resources-plugin/3.
2.0/maven-remote-resources-plugin-3.2.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-remote-resources-plugin/3.2
.0/maven-remote-resources-plugin-3.2.0.pom (33 kB at 607 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-remote-resources-plugin/3.
2.0/maven-remote-resources-plugin-3.2.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-remote-resources-plugin/3.2
.0/maven-remote-resources-plugin-3.2.0.jar (81 kB at 1.4 MB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-enforcer-plugin/3.4.1/mave
n-enforcer-plugin-3.4.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-enforcer-plugin/3.4.1/maven
-enforcer-plugin-3.4.1.pom (7.7 kB at 189 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/enforcer/enforcer/3.4.1/enforcer-3.4.1.p
om
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/enforcer/enforcer/3.4.1/enforcer-3.4.1.po
m (9.1 kB at 238 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-enforcer-plugin/3.4.1/mave
n-enforcer-plugin-3.4.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-enforcer-plugin/3.4.1/maven
-enforcer-plugin-3.4.1.jar (39 kB at 930 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/com/helger/maven/ph-javacc-maven-plugin/4.1.5/ph-javacc-m
aven-plugin-4.1.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/com/helger/maven/ph-javacc-maven-plugin/4.1.5/ph-javacc-ma
ven-plugin-4.1.5.pom (7.7 kB at 179 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/com/helger/parent-pom/1.11.4/parent-pom-1.11.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/com/helger/parent-pom/1.11.4/parent-pom-1.11.4.pom (44 kB
```



```
[INFO] Rat check: Summary over all files. Unapproved: 0, unknown: 0, generated: 1, approved: 254 licenses.
[INFO]
[INFO] --- artifact:3.5.1:check-buildplan (check-buildplan) @ commons-jexl3 ---
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-archiver/3.6.0/maven-archiver-3.6.
0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-archiver/3.6.0/maven-archiver-3.6.0
.pom (3.9 kB at 101 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/3.4.0/plexus-io-3.4.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/3.4.0/plexus-io-3.4.0.pom (6
.0 kB at 194 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/4.4.0/plexus-archiver
-4.4.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/4.4.0/plexus-archiver-
4.4.0.pom (6.3 kB at 174 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/commons-codec/commons-codec/1.15/commons-codec-1.15.pom
Downloaded from central: https://repo.maven.apache.org/maven2/commons-codec/commons-codec/1.15/commons-codec-1.15.pom (1
5 kB at 209 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-archiver/3.6.0/maven-archiver-3.6.
0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-archiver/3.6.0/maven-archiver-3.6.0
.jar (26 kB at 469 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/3.4.0/plexus-io-3.4.0.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/4.4.0/plexus-archiver
-4.4.0.jar
Downloading from central: https://repo.maven.apache.org/maven2/commons-codec/commons-codec/1.15/commons-codec-1.15.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/3.4.0/plexus-io-3.4.0.jar (7
9 kB at 778 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/4.4.0/plexus-archiver-
4.4.0.jar (211 kB at 1.0 MB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/commons-codec/commons-codec/1.15/commons-codec-1.15.jar (3
54 kB at 532 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/3.1.1/maven-
deploy-plugin-3.1.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/3.1.1/maven-d
eploy-plugin-3.1.1.pom (8.9 kB at 138 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/3.1.1/maven-
deploy-plugin-3.1.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/3.1.1/maven-d
eploy-plugin-3.1.1.jar (39 kB at 461 kB/s)
```

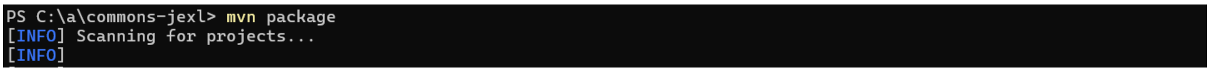
6. (1p) Find the manual of build tools and execute more functionalities such as creating an executable or distributable package, documents, generating various reports, etc. Then, show the commands you used and explain their results.

for create executable JAR, maven-assembly-plugin function can be used like this.

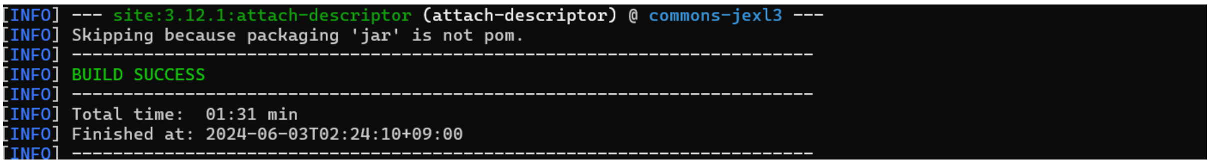


```
<plugin>
    <artifactId>maven-assembly-plugin</artifactId>
    <configuration>
        <descriptors>
            <descriptor>src/assembly/bin.xml</descriptor>
            <descriptor>src/assembly/src.xml</descriptor>
        </descriptors>
        <tarLongFileMode>gnu</tarLongFileMode>
        <finalName>commons-jexl-${project.version}</finalName>
    </configuration>
</plugin>
```

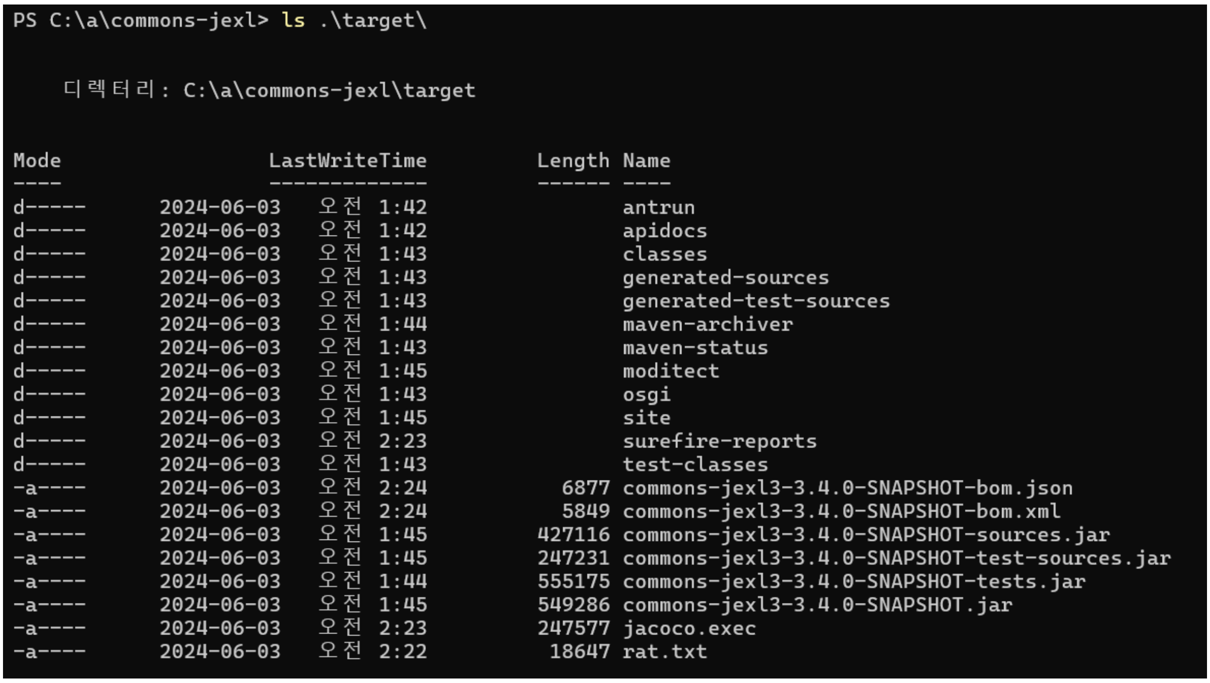
and when use the ‘mvn package’ command, JAR file on target directory is created!



```
PS C:\a\commons-jexl> mvn package
[INFO] Scanning for projects...
[INFO]
```

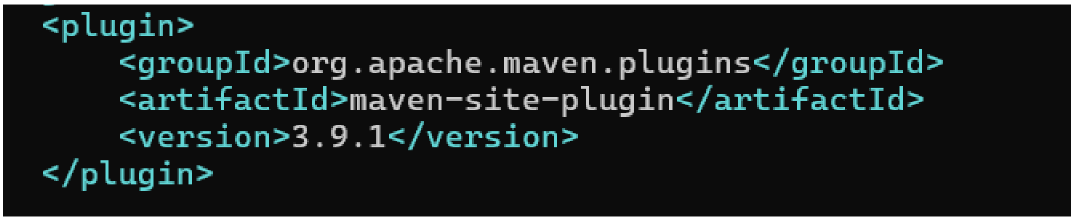


```
[INFO] --- site:3.12.1:attach-descriptor (attach-descriptor) @ commons-jexl3 ---
[INFO] Skipping because packaging 'jar' is not pom.
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  01:31 min
[INFO] Finished at: 2024-06-03T02:24:10+09:00
[INFO] ------------------------------------------------------------------------
```



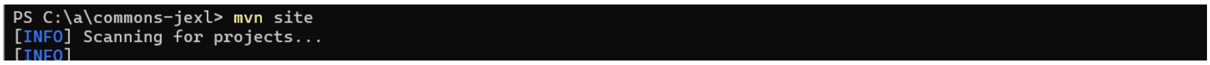
```
PS C:\a\commons-jexl> ls .\target\


    디렉터리: C:\a\commons-jexl\target


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----      2024-06-03   오전 1:42                antrun
d-----      2024-06-03   오전 1:42                apidocs
d-----      2024-06-03   오전 1:43                classes
d-----      2024-06-03   오전 1:43                generated-sources
d-----      2024-06-03   오전 1:43                generated-test-sources
d-----      2024-06-03   오전 1:44                maven-archiver
d-----      2024-06-03   오전 1:43                maven-status
d-----      2024-06-03   오전 1:45                moditect
d-----      2024-06-03   오전 1:43                osgi
d-----      2024-06-03   오전 1:45                site
d-----      2024-06-03   오전 2:23                surefire-reports
d-----      2024-06-03   오전 1:43                test-classes
-a----      2024-06-03   오전 2:24           6877 commons-jexl3-3.4.0-SNAPSHOT-bom.json
-a----      2024-06-03   오전 2:24           5849 commons-jexl3-3.4.0-SNAPSHOT-bom.xml
-a----      2024-06-03   오전 1:45         427116 commons-jexl3-3.4.0-SNAPSHOT-sources.jar
-a----      2024-06-03   오전 1:45         247231 commons-jexl3-3.4.0-SNAPSHOT-test-sources.jar
-a----      2024-06-03   오전 1:44         555175 commons-jexl3-3.4.0-SNAPSHOT-tests.jar
-a----      2024-06-03   오전 1:45         549286 commons-jexl3-3.4.0-SNAPSHOT.jar
-a----      2024-06-03   오전 2:23         247577 jacoco.exec
-a----      2024-06-03   오전 2:22          18647 rat.txt
```

Secondly, maven create documents also. I added this part to pom.xml for let maven create documents.

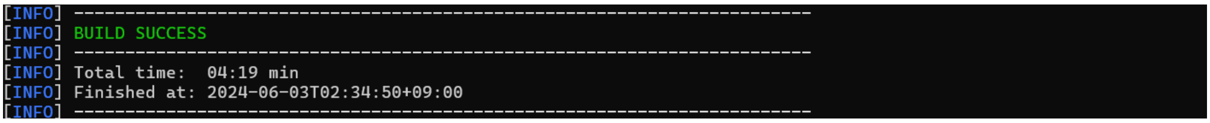


```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-site-plugin</artifactId>
    <version>3.9.1</version>
</plugin>
```

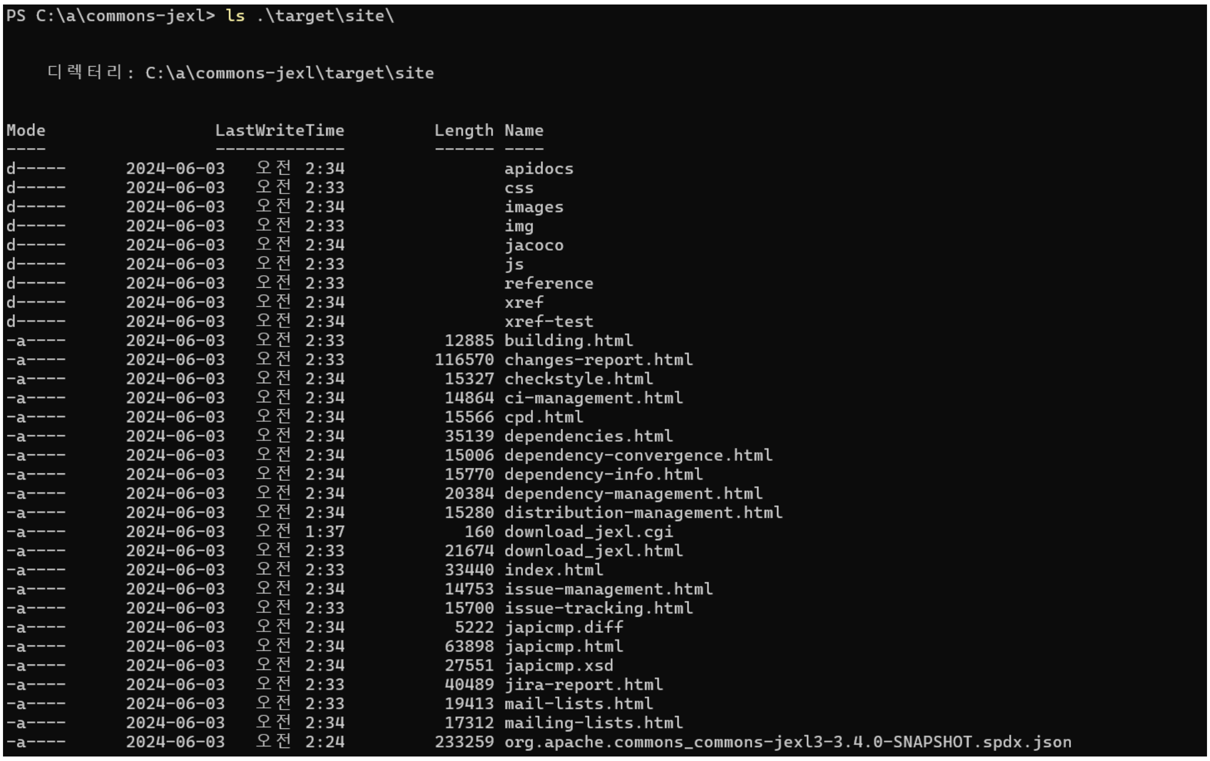
And using command ‘mvn site’, now I got document of common-jexl project in target/site directory!



```
PS C:\a\commons-jexl> mvn site
[INFO] Scanning for projects...
[INFO]
```



```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  04:19 min
[INFO] Finished at: 2024-06-03T02:34:50+09:00
[INFO] ------------------------------------------------------------------------
```



```
PS C:\a\commons-jexl> ls .\target\site\


    디렉터리: C:\a\commons-jexl\target\site


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----      2024-06-03   오전  2:34                apidocs
d-----      2024-06-03   오전  2:33                css
d-----      2024-06-03   오전  2:34                images
d-----      2024-06-03   오전  2:33                img
d-----      2024-06-03   오전  2:34                jacoco
d-----      2024-06-03   오전  2:33                js
d-----      2024-06-03   오전  2:33                reference
d-----      2024-06-03   오전  2:34                xref
d-----      2024-06-03   오전  2:34                xref-test
-a----      2024-06-03   오전  2:33          12885 building.html
-a----      2024-06-03   오전  2:33         116570 changes-report.html
-a----      2024-06-03   오전  2:34          15327 checkstyle.html
-a----      2024-06-03   오전  2:34          14864 ci-management.html
-a----      2024-06-03   오전  2:34          15566 cpd.html
-a----      2024-06-03   오전  2:34          35139 dependencies.html
-a----      2024-06-03   오전  2:34          15006 dependency-convergence.html
-a----      2024-06-03   오전  2:34          15770 dependency-info.html
-a----      2024-06-03   오전  2:34          20384 dependency-management.html
-a----      2024-06-03   오전  2:34          15280 distribution-management.html
-a----      2024-06-03   오전  1:37            160 download_jexl.cgi
-a----      2024-06-03   오전  2:33          21674 download_jexl.html
-a----      2024-06-03   오전  2:33          33440 index.html
-a----      2024-06-03   오전  2:34          14753 issue-management.html
-a----      2024-06-03   오전  2:33          15700 issue-tracking.html
-a----      2024-06-03   오전  2:34           5222 japicmp.diff
-a----      2024-06-03   오전  2:34          63898 japicmp.html
-a----      2024-06-03   오전  2:34          27551 japicmp.xsd
-a----      2024-06-03   오전  2:33          40489 jira-report.html
-a----      2024-06-03   오전  2:33          19413 mail-lists.html
-a----      2024-06-03   오전  2:34          17312 mailing-lists.html
-a----      2024-06-03   오전  2:24         233259 org.apache.commons_commons-jexl3-3.4.0-SNAPSHOT.spdx.json
```

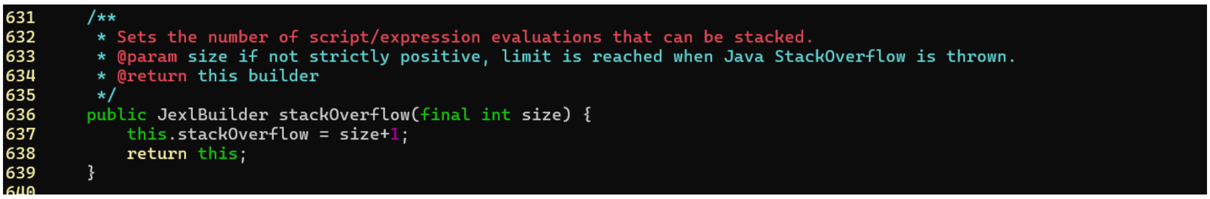
7. (2p) Write what you learned from the HW tasks.

When I check the homework, I worry about doing it and afraid it so much because I am not fluent with Java and java projects. But while do the homework, I surprised a lot because there are so many well-made build tools and they allow me to build and analyze the java projects I never used before, very easily and comfortably. I think tools like that are very meaningful that they let people can use anything without domain information and skills about them. And I think Its kind of similar with the concept encapsulation. They make user don’t know about the things user don’t need to know. How sweet!

* The following tasks are advanced and challenging ones. But these tasks are typical tasks that professional developers do in their projects.

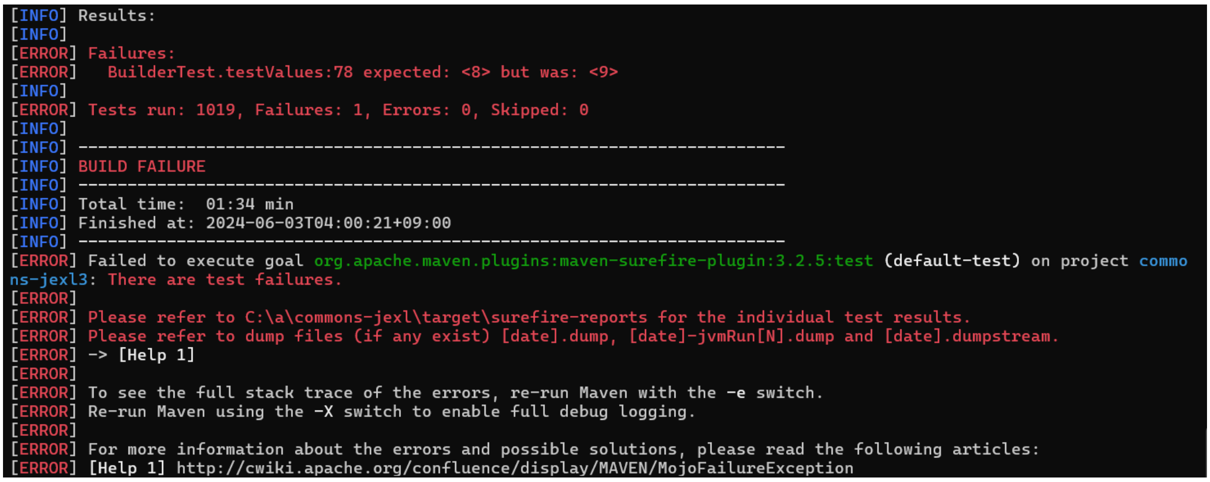
-

8. (2p) Modify the original source code by intentionally introducing a bug. Then, check the existing regression test to catch the bug you introduced. (Must not be errors but purely test failures.) Capture and put a screenshot showing the failure (which means the regression test successfully detected the bug you introduced).



```java
    /**
     * Sets the number of script/expression evaluations that can be stacked.
     * @param size if not strictly positive, limit is reached when Java StackOverflow is thrown.
     * @return this builder
     */
    public JexlBuilder stackOverflow(final int size) {
        this.stackOverflow = size+1;
        return this;
    }
```

I changed this part. *this.stackOverflow* value should be *size*, but I added 1.



```
[INFO] Results:
[INFO]
[ERROR] Failures:
[ERROR]   BuilderTest.testValues:78 expected: <8> but was: <9>
[INFO]
[ERROR] Tests run: 1019, Failures: 1, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  01:34 min
[INFO] Finished at: 2024-06-03T04:00:21+09:00
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-surefire-plugin:3.2.5:test (default-test) on project commo
ns-jexl3: There are test failures.
[ERROR]
[ERROR] Please refer to C:\a\commons-jexl\target\surefire-reports for the individual test results.
[ERROR] Please refer to dump files (if any exist) [date].dump, [date]-jvmRun[N].dump and [date].dumpstream.
[ERROR] -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException
```

And the test case catches the bug I made!

~~9.(3p) In case your modification adding a bug could not be detected by the regression test, developers must update the test case. If you face such a case, modify the test case that can detect the bug and explain your change. Capture and put a screenshot showing the test failure.~~