

Maven Interview / Practice Questions with Answers



1. *Maven compiler plugin error*

If you see: Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:3.8.1:compile

Steps:

- Run with debug: mvn clean install -X
- Check pom.xml for block with correct Java version.

Example:

```
org.apache.maven.plugins  
maven-compiler-plugin  
3.8.1
```

11

11

2. *Dependency not recognized*

Steps:

- Check if dependency exists in .m2: ls ~/.m2/repository/groupId/artifactId/version/
- Force re-download: mvn dependency:purge-local-repository && mvn clean install
- View dependency tree: mvn dependency:tree

3. *Apply a .patch file*

- Save patch file locally, then run: git apply fix.patch OR patch -p1 < fix.patch
- Rebuild with: mvn clean install

4. *Rerun only failed tests*

- Use Surefire rerun: mvn test -Dsfire.rerunFailingTestsCount=1
- Or rerun failed only: mvn surefire:test -Dsfire.failIfNoSpecifiedTests=false
- Failed test results stored in: target/surefire-reports

5. *Unsupported class version error*

Cause: JDK mismatch.

Check maven-compiler-plugin version & Java release.

17

6. *WAR → standalone JAR*

- In pom.xml: jar
- Add spring-boot-maven-plugin or maven-shade-plugin for executable jar.

7. Change build output directory

build_output

8. Skip tests in Maven build

- mvn clean install -DskipTests
- OR mvn clean install -Dmaven.test.skip=true

9. Generate site report

- Command: mvn site
- To include coverage: use jacoco-maven-plugin
- Output: target/site/index.html

10. Build project and generated files

Command: mvn clean install

Generated files:

- target/classes/ → compiled classes
- target/test-classes/ → test classes
- target/{artifactId}-{version}.jar/war
- target/surefire-reports/ → test reports

11. Dependency conflicts

- Maven resolves with nearest-wins strategy.
- View conflicts: mvn dependency:tree
- Force version using

12. Writing & running JUnit tests

Example:

```
@Test
public void testAddition() {
    assertEquals(4, 2+2);
}
```

- Run: mvn test
- Outputs: target/test-classes, target/surefire-reports

13. Executable JAR with main method

Use maven-shade-plugin:

```
org.apache.maven.plugins
maven-shade-plugin
```

package

shade

com.example.Main

14. Install third-party JAR

mvn install:install-file -Dfile=custom.jar -DgroupId=com.custom -DartifactId=custom -Dversion=1.0
-Dpackaging=jar
Confirm with: mvn dependency:tree

15. Create Maven web project (WAR)

mvn archetype:generate -DgroupId=com.example -DartifactId=myweb
-DarchetypeArtifactId=maven-archetype-webapp

Standard structure:

- src/main/java
- src/main/resources
- src/main/webapp/WEB-INF

16. Build WAR and deploy

Build: mvn clean package
WAR generated in: target/myweb.war
Deploy to Tomcat: copy WAR to \$TOMCAT_HOME/webapps/

17. Add JSTL & servlet-api

javax.servlet
javax.servlet-api
4.0.1
provided

javax.servlet
jstl
1.2

Note: 'provided' scope because servlet container already provides it.

18. Multi-module Maven project

Parent pom.xml → pom

Modules:

parent/

pom.xml

core/ (pom.xml, jar)

web/ (pom.xml, war depending on core)

19. Configure Maven web project

- pom.xml → war

- Execution:

WAR project → deploy to external server

Standalone JAR → run with java -jar