Xideral

Java Academy

Week 2 -Day 4

Maven

Presented by:

Edgar Itzak Sánchez Rogers

Monterrey Nuevo León México at 23 august 2024

Introduction:

Maven is a dependency management and build tool that automates the source code compilation and dependency management. Maven is used mainly for Java projects but can also be used to build and manage projects written in C#, Ruby, Scala.

Dependency Management:

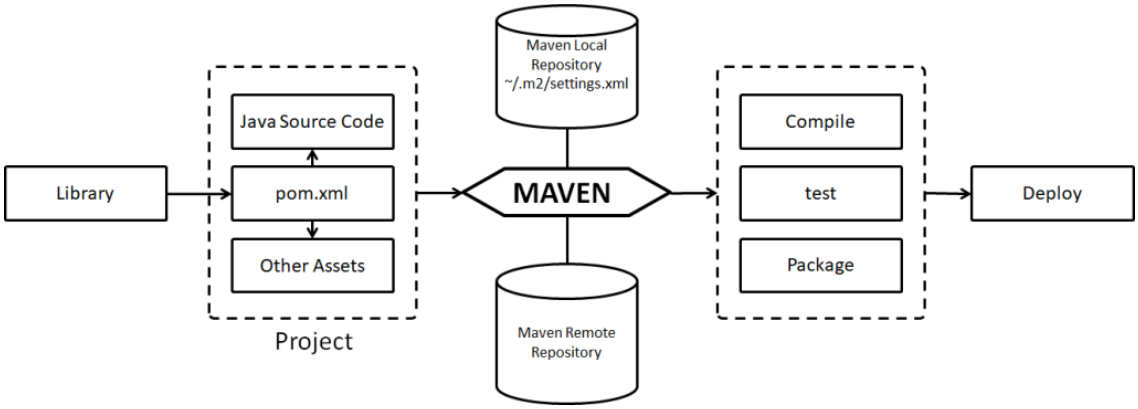
A software dependency is a library, that a software system or application relies on to function. These libraries contain code that has already been written and tested, which developers can reuse and incorporate into their software rather than building everything from scratch. Nearly all projects will require some library code. Maven makes all the work of looking for all dependencies in remote repositories and store them in the current machine.

- Step 1: Maven reads pom.xml file in the project's root directory, it contains information about the project's dependencies, plugins, and other configuration details. Maven combines the settings from the project's POM with those from parent POMs and any profiles that might be active

- Step 2: Maven first checks the local repository (.m2/repository) to see if the dependencies are already available.

- Step 3: If the dependencies are not found in the local repository, Maven queries remote repositories specified in the pom.xml, Maven also downloads dependencies of the project's dependencies.

- Step4: Maven verifies that all dependencies are available and then incorporates them into the build.



Pom.xml

A POM is the basement of the Maven framework. It's a type of XML file that accommodates data from your project and the configuration details. It includes the project, group ID, POM model version, artifact ID, and version. When executing a task or goal, Maven looks for the POM in the current directory. It reads the POM, gets the needed configuration information, then executes the goal.

```
    </plugins>
    <finalName>MavenEnterpriseApp-ear</finalName>
  </build>
    <dependencies>
        <dependency>
            <groupId>com.mycompany</groupId>
            <artifactId>MavenEnterpriseApp-ejb</artifactId>
            <version>1.0-SNAPSHOT</version>
            <type>ejb</type>
        </dependency>
        <dependency>
            <groupId>com.mycompany</groupId>
            <artifactId>MavenEnterpriseApp-web</artifactId>
            <version>1.0-SNAPSHOT</version>
            <type>war</type>
        </dependency>
    </dependencies>
</project>
```

Maven commands:

Maven commands are written and executed in a command-line interface, to to run Maven commands on Linux:

- Step 1: Open the Terminal application
- Step 2: Change to a Maven project directory using the cd command
- Step 3: Type the command

Some Maven commands are listed below:

mvn compile

- compiles the source code and places the resulting .class files in the target/classes directory.

mvn test

- Run the unit tests specified in a project using a suitable testing framework like JUnit. Maven automatically runs tests found in the src/test directory.

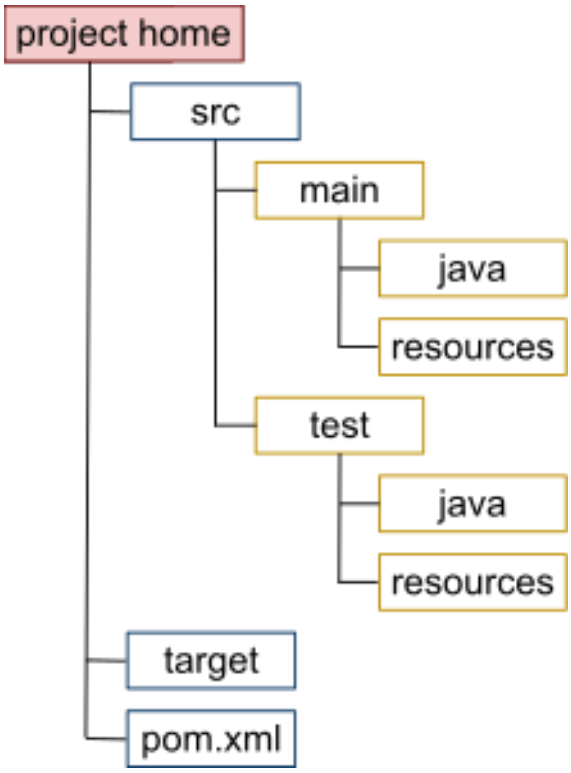
mvn package

- Compiles a project and packages the compiled code along with any resources into a JAR file within the target directory.

mvn clean

- Clean aproject with mvn clean to remove the target directory, which contains all the build files, ensuring a fresh start for the next build.

Standard directory layout maven



Conclusion:

Maven is a great tool that helps developers start working on a new project or module without wasting any time, provides a simple project setup. Maven creates outstanding dependency management for projects, dependency management also includes automatic updates. Maven allows you to keep your JAR files for future usage. As it's extensible, it can easily write plugins using Java and other scripting languages. Maven easily manages project release and publishes the project in a distribution location.

References:

[1] Maven Commands Cheat Sheet | Contabo Blog

[2] Apache Maven - Wikipedia

[3] What is Maven in Java? (Framework and Uses) | BrowserStack

[4] Managing Software Dependencies: Types & Risks | LeanIX

[5] Maven – Introduction to the POM (apache.org)