Maven: A Complete Guide

What is Maven?

Maven is a **build automation tool** primarily used for **Java projects**. It helps manage **project dependencies, builds, documentation, and reporting in a standardized way.**

Maven was created by **Jason van Zyl in 2002** and began as a sub-project of **Apache Turbine. In 2003 Maven** was accepted as a top level **Apache Software Foundation project.**

Key Features of Maven

1. **Project Object Model (POM)**

   Maven uses an XML file **(pom.xml)** to configure project dependencies and build settings.

   Example pom.xml:

```
<project xmlns=http://maven.apache.org/POM/4.0.0
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<groupId>com.coforge</groupId>
<artifactId>SpringDI</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>jar</packaging>
</project>
```

2. **Dependency Management**

   Automatically downloads and manages libraries (JAR files) from repositories.

   Example of adding a dependency (e.g., **Spring Context**):

```
<dependencies>

<!--
https://mvnrepository.com/artifact/org.springframework/spring-context -->
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-context</artifactId>
<version>6.1.14</version>
```

```
    </dependency>

  </dependencies>
```

3. **Build Automation**
   - Simplifies compiling, testing, packaging, and deploying applications using commands like:
   - **mvn clean install**
   - Common Maven **Phases/Lifecycle:**

Phase	Description
validate	Checks if the project is correct.
compile	Compiles the source code.
Test	Runs unit tests.
package	Creates a JAR/WAR file.
Install	Installs the package in a local repository.
deploy	Deploys the project to a remote repository.

4. **Plugins & Goals**
   - Plugins enhance functionality (e.g., maven-compiler-plugin for Java compilation).

     Example:

```
<build>
  <plugins>
          <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-war-plugin</artifactId>
    <version>3.3.1</version>
</plugin>
        </plugins>
    <finalName>SpringMVC</finalName>
  </build>
```

5. **Standard Directory Structure**
   - Maven follows a **convention-over-configuration** approach:
   - my-app/
   - ├── src/
   - │   ├── main/java/ (**Source Code**)
   - │   ├── main/resources/ (**Config files**)
   - │   ├── test/java/ (**Test Code**)
   - ├── pom.xml (**Project Configuration**)
6. **Repository Management**
   - **Local Repository:** Stored on the developer's machine (**.m2/repository**).
   - **Central Repository:** Maven's default online repository (**https://repo.maven.apache.org/maven2**).
   - **Remote Repository:** Company-hosted private repositories (e.g., **mvnrepository,** Nexus, Artifactory).

How to Install and Use Maven

1. Install Maven

- **Windows:** Download and set up environment variables.
- **Linux/Mac:** Install using:
- sudo apt install maven # Ubuntu
- brew install maven # macOS

2. Check Maven Version

mvn -version

3. Create a Maven Project

mvn archetype:generate -DgroupId=com.example -DartifactId=my-app -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

---

Advantages of Maven

Standardized Build Process
Automatic Dependency Management
Integration with CI/CD Tools (Jenkins, GitHub Actions, etc.)
Supports Multi-Module Projects
Works with Various Technologies (Spring, Hibernate, etc.)

Conclusion

Maven is a powerful tool for managing Java projects efficiently, automating builds, and handling dependencies. Its **simplicity and scalability** make it a preferred choice for developers working on enterprise applications.