Apache Maven

What is Maven?

Apache Maven is a **build automation and project management tool** primarily used for **Java projects**. It simplifies the build process, dependency management, and project documentation using a standardized configuration (pom.xml).

Why is Maven Important?

- **Dependency Management**: Automatically downloads required libraries.
- **Standardized Build Process**: Ensures consistency across projects.
- **Efficient Project Management**: Provides lifecycle management and reporting.
- **Integration with Git and CI/CD Pipelines**: Works seamlessly with repositories like GitHub and CI tools like Jenkins.

Installing Java, Git, and Maven

Step 1: Install Java

Maven requires Java to run. We will install **Amazon Corretto 8 (OpenJDK 1.8)**.

sudo yum install java-1.8.0-amazon-corretto.x86_64

After installation, verify Java:

which java java --version

Step 2: Install Git

Git is required to clone repositories from GitHub.

yum install -y git

Verify Git installation:

git --version

Step 3: Install Maven

1. Download the Maven repository file:

sudo wget http://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-maven.repo -O /etc/yum.repos.d/epel-apache-maven.repo

2. Modify the repository configuration:

sudo sed -i s/\$releasever/6/g /etc/yum.repos.d/epel-apache-maven.repo

3. Install Apache Maven:

sudo yum install -y apache-maven

4. Verify Maven installation:

mvn -version

Cloning a GitHub Repository

To download a Maven project from GitHub:

git clone https://github.com/Anand1827/myweb.git

Navigate into the project directory:

cd myweb

Check if the pom.xml file exists:

ls

Building the Maven Project

Step 1: Validate the Maven Project

mvn validate

This ensures the pom.xml file is correct.

Step 2: Compile the Java Code

mvn compile

This converts Java source code into bytecode (.class files).

Step 3: Run Tests

mvn test

This executes all unit tests in the project.

Step 4: Package the Project

mvn package

This generates a .war file (since your project uses <packaging>war</packaging> in pom.xml).

Step 5: Verify the Build

mvn verify

This checks if the generated package meets project requirements.

Step 6: Install the Package Locally

mvn install

This installs the package in Maven's local repository (~/.m2/repository/).

Directory Structure & Viewing Files

Use tree to visualize project files:

yum install tree -y # Install tree if not available tree

This command shows the hierarchical structure of the project.

Conclusion

By following these steps, you have:

- Installed Java, Git, and Maven.
- Cloned a GitHub repository.
- Built, tested, and packaged the project.
- Installed the package locally for further use.

Maven is a powerful tool for automating Java builds and managing dependencies efficiently.