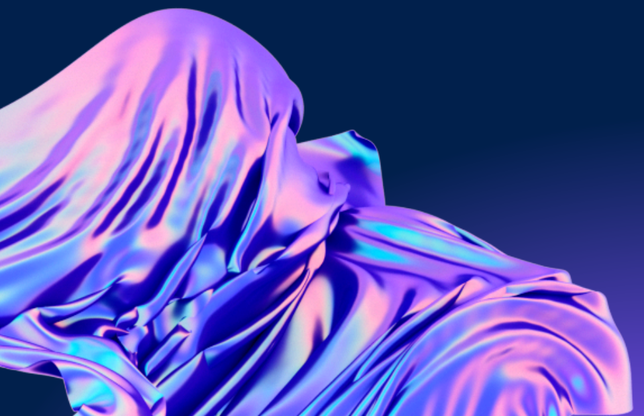
MAVEN

Karen Giselle Valdez Muñoz

What is Maven?

Maven is a build automation and project management tool used primarily for Java projects, though it can be used for other programming languages as well. It simplifies the process of building, managing, and understanding your project's dependencies, configurations, and lifecycle.



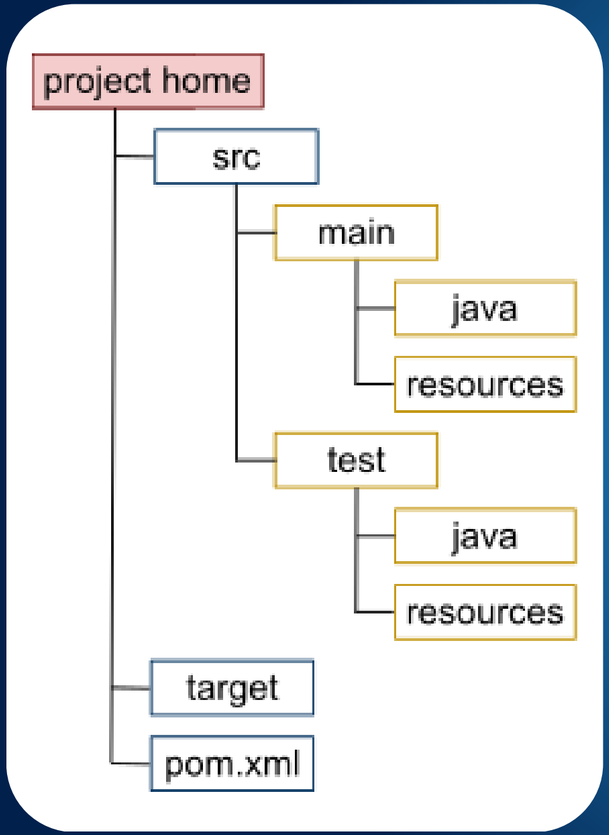
Maven Advantages

- Automatic Handling of Dependency Management
- Version Control
- Standardized Project Structure
- Build Automation
- Centralized Project Information
- Reproducible Builds
- Multi-Module Project Support

What Maven do?

Enforce a Directory Structure

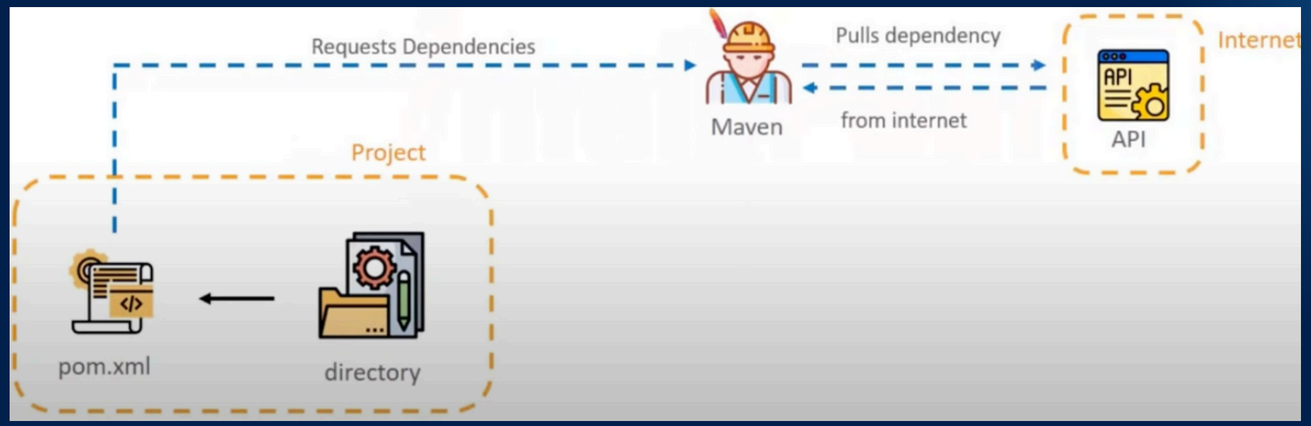
Maven promotes a standardized directory structure for projects, which helps ensure consistency and makes it easier for developers to understand and navigate projects.



What Maven do?

Manage and download Project Dependencies

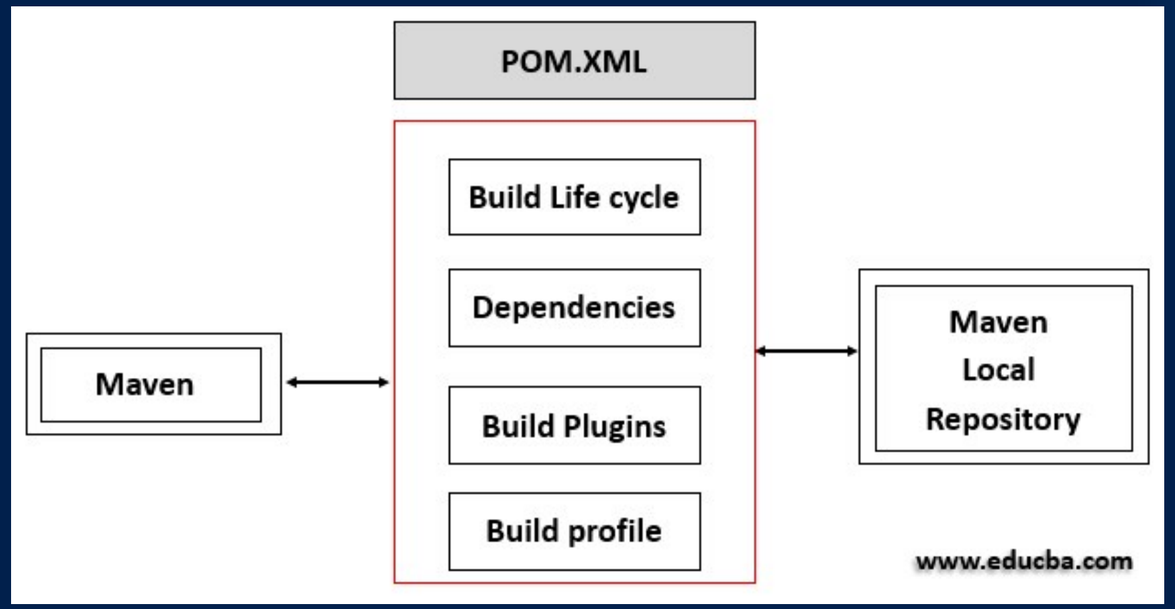
Maven automates the process of retrieving dependencies, external libraries or components that your project needs, that are defined in the pom.xml file and ensuring that the correct versions are used.



POM File

Project Object Model

- Every Maven project directory needs a pom.xml
- It contains the details that are necessary for maven effectively execute a project.
- The POM file is stored in ‘src’ or the root directory



Maven Repositories

After Maven understands which all depenedencies are needed from the pom.xml file, it will download them from where it need to online and then store them in the local repository for curren or future use.

- Local Repository: a cache on your local machine where Maven stores artifacts that you have downloaded or built.
- Remote Repository: a centralized repository hosted on a remote server where artifacts are stored and can be accessed by Maven projects.
- Central Repositor: the default public repository provided by the Maven community.
- Snapshot Repository: used to store and retrieve snapshot versions of artifacts. Snapshots are typically used for development and testing before a final release is made.

Maven Life Cycle

- When maven builds a software it follows a build cycle, they are three types:

Default Lifecycle

Clean Lifecycle

Site Lifecycle

Default Lifecycle

1. **Validate: Checks if the project is correct and all necessary information is available.**
2. **Compile: Compiles the source code of the project.**
3. **Test: Runs the unit tests of the project.**
4. **Package: Packages the compiled code into a distributable format, such as a JAR (Java ARchive), WAR (Web Application Archive), or EAR (Enterprise Archive).**
5. **Verify: Runs any additional verification tasks to ensure the quality of the package.**
6. **Install: Installs the packaged artifact into the local Maven repository**
7. **deploy: Deploys the artifact to a remote repository.**

Clean Lifecycle

In this lifecycle all the files that were produced by maven will be automatically removed

1. **pre-clean: Executes before the cleaning process begins. It can be used to perform any tasks required before the actual cleaning starts.**
2. **clean: Removes the target directory, which is where Maven typically stores build outputs and intermediate files. This is the core phase of the clean lifecycle.**
3. **post-clean: Executes after the cleaning process is completed. It can be used to perform any tasks that need to happen after the project is cleaned.**

Site Lifecycle

This lifecycle generates a project file documentation in the HTML format, but this cycle is usually ignored as its output is not that useful.

1. **pre-site: Executes before the site generation process starts. This phase can be used for tasks that need to be performed before generating the site, such as preparing resources or configurations.**
2. **site: Generates the project's site documentation. This phase involves running site plugins and generating various reports based on the project's configuration and dependencies.**
3. **post-site: Executes after the site generation is complete. This phase is used for any tasks that need to occur after the site has been generated, such as additional processing or deployment tasks.**
4. **site-deploy: Deploys the generated site to a remote server. This phase typically uploads the site documentation to a web server or repository for public access.**

Maven Plugins

Plugin is a file that contains the knowledge for what is supposed to happen when a phase is executed

Phase	Maven Plugin	Goal
Clean	Clean Plugin	Clean
Site	Site Plugin	Site
Process-resources	Resources Plugin	Resource
Compile	Compile Plugin	Compile
Test	Surefire Plugin	Test
Package	Varies based on the packaging[JAR/WAR/EAR]	Jar (in case of JAR packaging)
Install	Install Plugin	Install
Deploy	Deploy Plugin	Deploy