

# Maven & POM – Quick Notes (School2Skill)

## 1 What is Maven?

- Maven is a **build automation and dependency management tool**
  - It automates:
    - Compile
    - Test
    - Package
    - Clean
    - Install
    - Deploy
  - Maven follows a **standard project structure**
- 

## 2 What is POM?

- POM stands for **Project Object Model**
- Written inside **pom.xml**
- It is the **heart of a Maven project**
- Contains:
  - Project information
  - Dependencies
  - Build lifecycle configuration
  - Plugins

👉 POM is **configuration**, Maven is **execution**

---

## 3 Why POM is called Project Object Model?

- ♦ **Project**
  - Identified using **GAV**
    - **GroupId** → Organization name
    - **ArtifactId** → Project name

- **Version** → Project version
  - GAV uniquely identifies a project
- 

#### ◆ **Object**

- Project depends on **dependencies**
  - Dependencies can depend on **other dependencies**
  - This is called **transitive dependency**
  - Maven automatically resolves all dependencies
- 

#### ◆ **Model**

- Defines **how the project is built and run**
  - Includes tools for:
    - Compile
    - Test
    - Package
    - Clean
    - Install
    - Deploy
    - Run
  - Managed through **Maven build lifecycle**
- 

## 4 Maven Build Lifecycles

Maven has **three lifecycles**:

1. **Clean**
    - Removes old build files
  2. **Default (Build)**
    - Compile
    - Test
    - Package
    - Install
    - Deploy
  3. **Site**
    - Generates project documentation
-

## 5 Why Maven is called the Heart of POM?

- POM only defines **what to do**
- Maven actually **does the work**

- ✓ Reads POM
- ✓ Resolves dependencies
- ✓ Builds project
- ✓ Runs tests
- ✓ Creates packages

👉 Without Maven, POM cannot function

---

## 6 Dependency Resolution in Maven

When Maven needs a dependency:

1. Checks **local repository** (`.m2` folder)
2. If not found → goes to **Maven Central Repository**
3. Downloads dependency
4. Stores it in `.m2` for future use

- ✓ Faster builds
  - ✓ No repeated downloads
- 

## 7 Maven Repositories

- **Local Repository** → `.m2`
  - **Central Repository** → Public remote repository
  - **Remote Repository** → Company/private repositories
- 

## 8 Packaging in Maven

Maven can create:

- **JAR** → Java applications
- **WAR** → Web applications

These packages can be:

- Shared with teams
  - Deployed to remote repositories
  - Used as dependencies by other projects
- 

## 9 Effective POM

- **Effective POM** is the final merged POM
  - It includes:
    - Super POM
    - Parent POM
    - Default Maven configurations
  - Shows what Maven actually uses during build
- 

## 🔑 One-Page Summary (Very Important)

Term	Meaning
Maven	Build & dependency management tool
POM	Project configuration file
GAV	Project identity
Dependency	External library
Transitive Dependency	Dependency of dependency
.m2	Local repository
Effective POM	Final merged configuration