18/8/2020

Maven fundamentals

- transitive dependencies & its management
- Repeatable builds
- IDE & standalone
- Consistency
- Steeper learning curve

For Unit tests: Pom.xml
src/main/java

Compilation Directory

Four different parts:

Project Information
GroupId same as the Package names
artifact Id same as name of application
Version should be 1.0, 2.0, 3.0..

Dependencies are imported by naming Conventions, groupId, artifactId, version & Dependencies Section

Goals → clean, Compile, Package, install, deploy

→ Avoids duplication

Version → release number of Artifacts we use

latest one → SNAPSHOT

Packaging Types

↳ Pom, jar, war, ear, maven-Plugin

Default jar.

Dependency pom

Scopes → Compile, provided & runtime, test, System, import

Repository → http locations, Super pom.xml, Default location, Default Maven Installation, Multiple repositories allowed, Corporate Repository, Nexus

Dependency Repository:
↳ identical to plugin Repository

Releases/ Snapshots → Same repo, Snapshots, Milestones, Release Candidate, Release policies

plugins → What Maven uses to build &

Super pom has goals defined inherited from pom.xml

phases →
- Validate → validate project & structure
- compile → Compile any source in project
- Test → Test Compiled code
- Package → Packages the code in specified Package type

Jar Plugin:
↳ Package, Package phase, Configuration, includes/excludes manifest.

Source Plugin → Package source code.

Javadoc plugin → Package javadocs, Package phase.

Pom Viewer → Default View, Highlevel Elements

Maven lifecycle:
~~three~~ inbuilt lifecycles:
- default → handles Project deployment
- clean → handles project cleaning
- site → handles creation of your project's site documentation