Maven Fundamentals - Bryan Hansen (kind of useless)

Ant

-another build tool

Introduction to Maven

Maven

-Tool to build source code, produce an artifact (think of it as a jar or zip file)

-Owned by Apache

-open source

Local repo

-historically downloaded JARs and kept them inside your project (could be 20 projects)

-maven allows you to download it once and then reference it

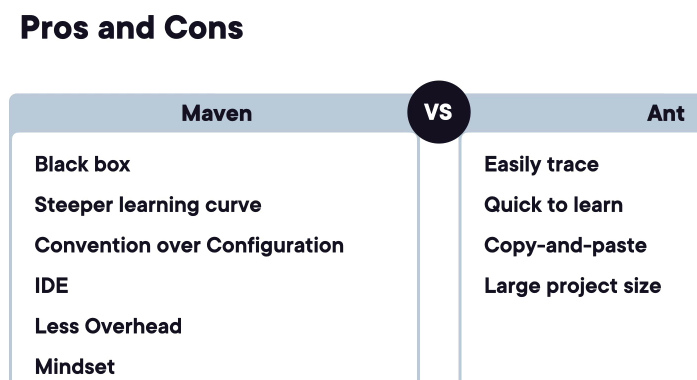
Ant

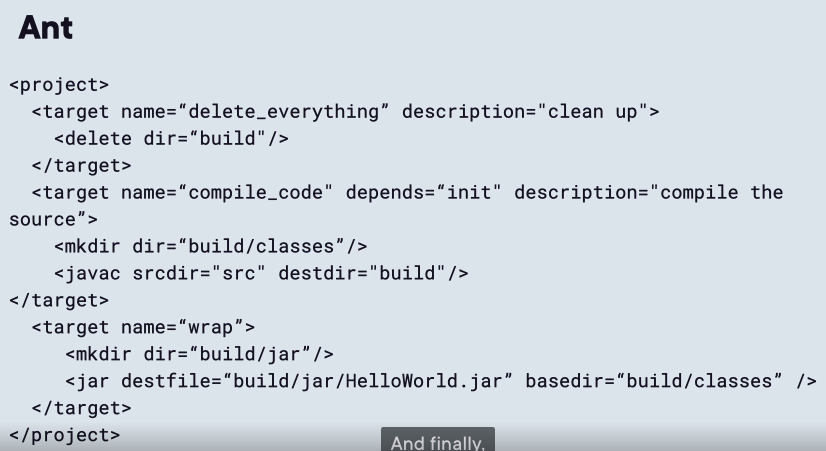
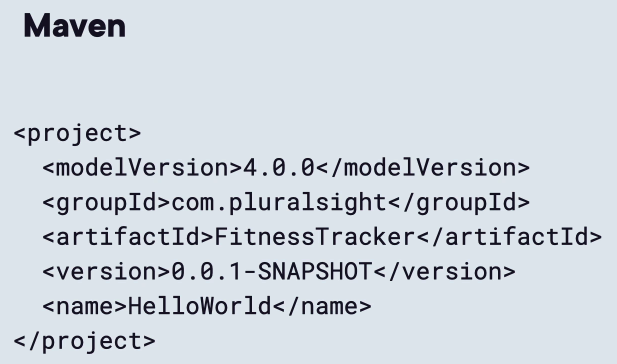
-replacement for Make (wasn’t cross platform)

-very explicit

Maven

-for than a scripting tool





Gradle

-Maven without XML

-yaml based

mvn clean

-first time its going to download all dependencies

Project Structure using Maven

Maven Default Structure

-Tool to build source code, produce an artifact (think of it as a jar or zip file)

-Owned by Apache

target

-its where tests are ran



groudId

-package structure

Packaging

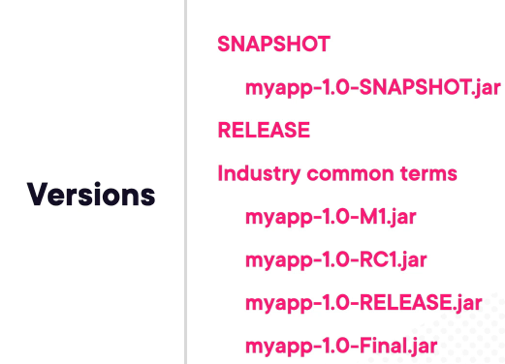
-is how we distribute our application (jar, war, etc..)

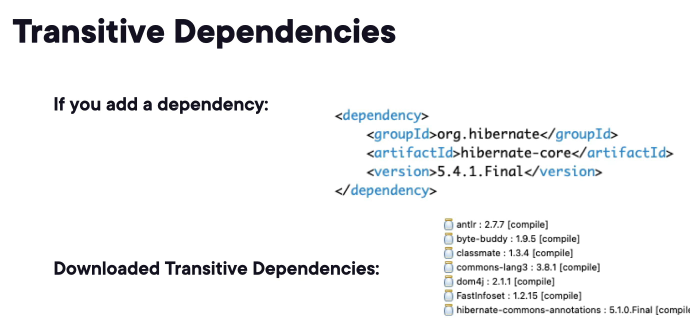
Working with Dependencies

Maven Default Structure

-Tool to build source code, produce an artifact (think of it as a jar or zip file)

-Owned by Apache





Scope

-determines when the dependency is included in the project’s build process

Compilation, Packaging (JAR), Runtime

compile

-default (resource is available for all)

provided

-artifact is provided through all the build cycle expect packaging

-good idea to not include it because it will provided by the container

runtime

-included in runtime and packaging (opposite of provided )

test

-test compilation and execution phase

system

-never use

-going to hard path to a JAR on your file system (defeats the purpose of Maven!)

-not in package but in compilation and runtime

import

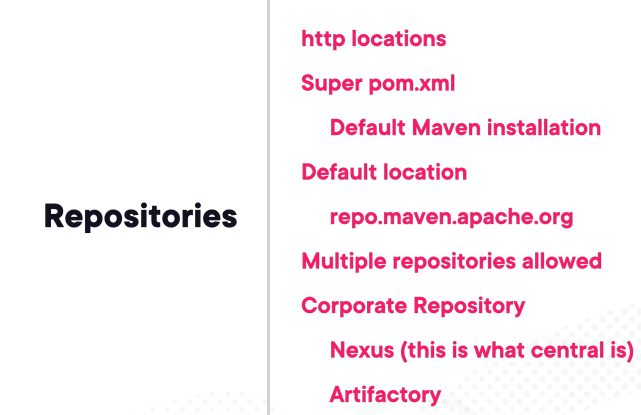
-used to import dependencies from other projects

Maven Repositories

~/.m2/repository

-Maven looks at local repo first, if it doesn’t it will look for remote





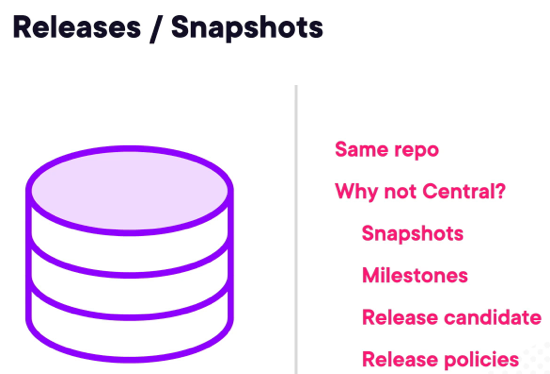
-Nexus is backing repo.maven.apac.org

Dependency Repository



-a repository where we download all our repo from

Demo



Summary

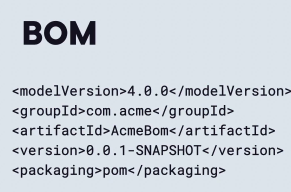
-Dependency/Plugin can be in same or different repo

-Projects will not upload their SNAPSHOT to central because of headaches and frequent changes code will go through

-Plugins are usually in the same repo as dependencies but can be broken out

-Companies should be looking at their own corporate repo to help lighten the load on central repo

Using Plug Ins



-packaging type is pom