<https://www.youtube.com/watch?v=Xatr8AZLOsE>

Maven

-build tool

-used for java, Kotlin

-easy to make to download 3rd party dependencies

-gives you a structure

-plugins for quality check or to deploy your code

-pom.xml is for maven

-need to add the mvn executable inside the bin folder to your path variable

Go to “Env Variable for your System”, Path -> Edit -> New -> Paste in the Path (to your bin)

To check click mvn -v to double check

-if not using a package manager, set the env variable

maven wrapper (mvnw)

mvn:wrapper wrapper //install wrapper

- mvnw.cmd // for windows

- mvnw //for linux/macos

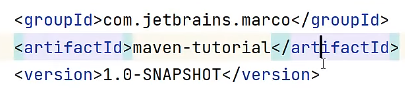
if you didn’t have mvn, you can use mwnw

-embedded maven version that comes with your project

-don’t need mvn installed if you have mvnw (just need mvn for initial creation)

mvn validate

-looks at project and checks to see if it a valid maven project or broken xml file



-group and artifact Id uniquely identify your project among all the projects in the word

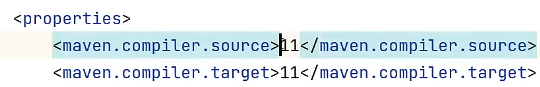
groupId: take the reverse of the domain names

artifactId: projectId

SNAPSHOT-means WIP (not a release version)

properties

-which configures something inside maven



-means my project is made for java 11

build <plugins>

dependencies



scope

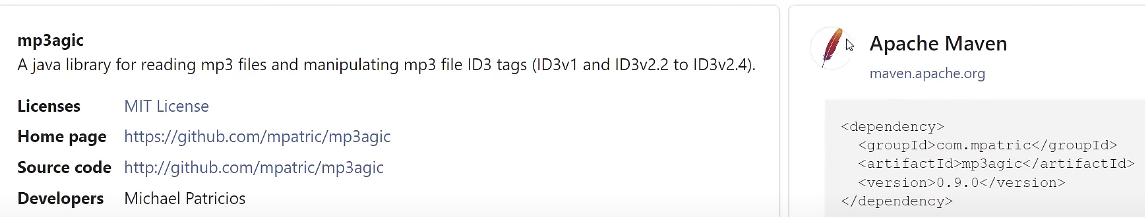
-compile is default?

-compile means project needs dependency when you compile, test, deploy

-test: only need it for testing

searh.maven.org

-if you know what you are looking for, and grab the dependencies



mvn clean

-deletes target folder

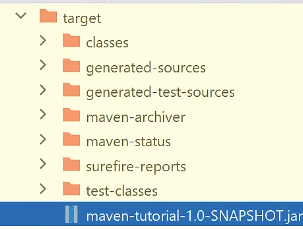
mvn compile

-looks src/main and compiles and puts it under target/classes

mvn test

-internally runs mvn compile, then looks at src/test

mwnw

-runs it with ./mwnw?

mvn package

-includes compile & test

-takes your source code in jar file

-inside target

-can send jar file to your friend

mvn install

-does a package AND jar file is being put into local maven repository (m2)

mvn repository

-Your team can have their own maven repository

-when maven tries to download dependency it tries to ask the remotes (global repo), and the downloads it into m2 so it can be reused later on

src/main/resources

-find your application properties

parent pom

-doesn’t produce a jar file

modules

-parent will build all these modules

children

-will inherit all dependencies

-doesn’t show the version because its get from the parent



-model is combined into a jar file

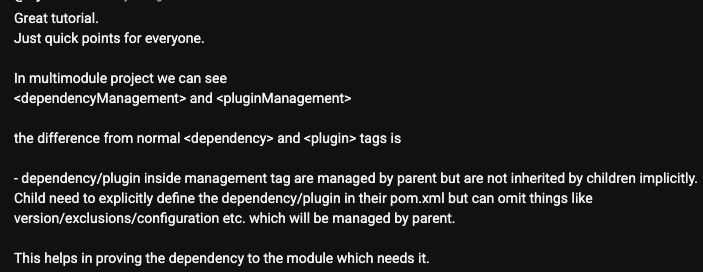
-modules can be dependent on each other



-as long they live in the same project

multi module maven projects

-children can specify their own stuff and that is essentially multi module maven project



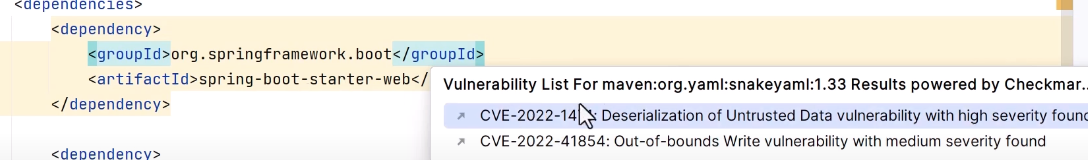
<https://www.youtube.com/watch?v=6GJAjDcFHTo>

How to approach dependency management in java

bom (bill of material)



-shows newer versions

vulnerability

-alt + enter

how to trust dependencies