Maven Tool

What is Maven???

=>Project management tool

=>Helper for build management and dependencies

Java based application: continous req of additional jar files

Traditional : download from each project website

manually need to add to buildpath

Spring

Hibernate

JSON Convertors...   Developer

Spring
Hibernate
JSON Convertors...   Maven

Developer

Tell the maven about your req. (as of a list : config file)

Maven will go out and download the req. jar file

Maven will also download the sub-dependencies of those jar file

Maven will integrate those res in build path...

Maven working!!

Maven Local Repo

list(config file)

(your computer)

Spring

Hibernate

1

JSON Convertors...   Maven

5

2

4

Maven Central Repo(Remote)

3

Developer

1. Read Config file

2. Check the local repo

3. Get res from remote repo

4. Save in local repo

5. Make it available to app

Maven will provide a standard Directory structure

Problem : dev teams might dream up any directory structure(no standardized)

any new person will have problem integrating at later stages

Standard Maven Directory structure for a webapp

Directory:

    src/main/java : Java source code

    src/main/resources : home for properties/config

    src/main/webapp : home of web assets(JSP,web config files,images,js,css...)

src/test: Unit testing codes

target : Destination dir for compiled code....

Major IDEs have built in support for maven(easily imports maven projects)

Maven projects are portable (easilt shared among diff ide)

maven key concepts

1. POM file-pom.xml

2. Project Coordinates

pom.xml:

Located at the root of project:

Project Object Model file

Configuration file for project (shopping list)

Pom file structure:

1. Current Project details : Project name,version, type of output : jar,war

2. list of dependencies :

3. Additional custom tasks

Project Coordinates:

 unique maven way to refer any project (just like coordinates of any address)

Maven project coordinates as GAV coordinates:

G : GroupId: Name of the org/company/group to which project belongs

eg: com.cts

org.springframework

org.hibernate

A: ArtifactId: Name of the project

V : Version : specific version release

eg:

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>5.1.6.RELEASE</version>

Maven plugin :

archetype : templates for diff types of java application

Maven project by default uses J2SE 1.5 spec:

need to update it to use our current version

Maven Spring web mvc project:

    Maven archetype : webapp

    After project buildup: Configure build path : add library: server Runtime : apache tomcat

    update maven to use java1.8

    add dependency

        1. spring-webmvc (spring web mvc framework)

        2.servlet-api (for jsp support/DS)

        3.servlet-jsp-api (for jsp support)

delete web.xml file

Add support for build management :

 Add a special build plugin of maven (so that maven is completly responsible for build)

Eclipse by defualt does not uses Web Module 4.0

Change Web Module in project properties

Manually:

Project Folder:settings: facets.core.xml : :jst.web: 4.0

Update the project

Add java config files