CURSO DE MAVEN

//------------------------------------------------------//---------------------------------------------------//

Requisitos

* Instalar el JDK de JAVA
* Instalar Apache Maven
* Intalar eclipse

//------------------------------------------------------//---------------------------------------------------//

Instalación en Linux

Comando para instalar el JDK:

* sudo apt-get update
* sudo apt-upgrade / s /
* sudo apt-get install default-jdk
* sudo apt-get install default-jre

Comando para instalar el maven:

sudo apt install maven / s /

Para mirar la versión del JDK en el terminal es: java -vercion

Para mirar la versión del MAVEN en el terminal es: mvn -vercion

//------------------------------------------------------//---------------------------------------------------//

PRIMEROS PASOS

COMANDOS BASICOS:

* mvn
* mvn archetype:generate descarga todas las lIbrerIas que necesita para funcionar
* mvn archetype:generate -Dfilter=org.apache:struts filtrar plantillas

//------------------------------------------------------//---------------------------------------------------//

Archetypes: son plantillas de proyecto predefinidas para acelerar nuestro poryecto partiendo de ellas.

mvn archetype:generate -DarchetypeGroupId=org.apache.maven.archetypes -DarchetypeArtifactId=maven-archetype-quickstart

* groupId: co.empresa.maven // nombre agrupamiento de proyectos
* artIfactId: ejemplo1 // nombre del proyecto
* versIon: por defecto / uno u otro
* package: co.empresa.java // nombre del paquete

RESUMEN / yes defecto

//------------------------------------------------------//---------------------------------------------------//

Instalar el proyecto ya generado

COMANDOS

* mvn install

//------------------------------------------------------//---------------------------------------------------//

CONFiGURACiON DE ECLiPSE

windows > preferences > java > installedJREs > add > agregamos nuestro JDK

windows > preferences > maven > activar checkbox download repository index updates on startup

windows > preferences > maven > hay uno enbebido EMBEDDED > add > agregamos nuestro maven descargado

//------------------------------------------------------//---------------------------------------------------//

IMPORTAR UN PROYECTO A ECLIPSE

File > import > Existing Maven Projects : sabiendo que ya es un proyecto maven > buscamos el proyecto y detecta el pom.xml > finish.

//------------------------------------------------------//---------------------------------------------------//

CREAR UN PROYECTO EN ECLIPSE

File > new > Maven Project > next > all catalog > next > datos id > finish

//------------------------------------------------------//---------------------------------------------------//

COMANDOS EN ECLIPSE

mvn -B clean : borra el directorio para volver a comenzar desde 0

PARA EJECUTAR COMANDOS QUE NO ESTAN EN EL RUN AS

Click derecho en el Proyecto > run as > maven build…> goals > fases plugin > run

* package

CONCATENAR COMANDOS

Click derecho en el Proyecto > run as > maven build…> goals > clean package > apply > run

HERRAMiENTA POM

* pom.xml interface :
* pom.xml :
* Efective POM : todas las instalaciones de maven disponen de un pom xml donde vienen precunfigurado todas las rutas e información de como debe ejecutarce – el pom es especifico para el proyecto por que es exclusivo. Pom meiven + pom del proyecto = efective pom

//------------------------------------------------------//---------------------------------------------------//

El CiCLO DE ViDA MAVEN

Loa ciclos de vida se pueden alterar y maven declara 3 cilos de vida

* Ciclo de vida por defecto
* Ciclo de vida de limpieza 3 faces
* Ciclo de vida de generación de documentación (Site) 4 faces

Se puede llamar de manera predecterimada y ejecuta todas sus subfaces.

Saber que hace una face en particular = help:describe -Dcmd=compile

//------------------------------------------------------//---------------------------------------------------//

MAVEN COMPiLER PLUGiN

compiler:compile plugin:face queremos ejecutar

clean:clean compiler:compile compiler:testCompile

sobrescribir la información del compilador predeterminado a su mejor versión

como establecer una versión de configuración a medida

como afectar al comportamiento del compilador por defecto

<build>

<pluginManagement>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.1</version>

<configuration>

<!--<verbose>true</verbose>-->

</configuration>

</plugin>

</plugins>

</pluginManagement>

</build>

//------------------------------------------------------//---------------------------------------------------//

CAMBiANDO EL NiVEL DE COMPiLACiON

<build>

<pluginManagement>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</pluginManagement>

</build>

MarKers como consol para mirar los posibles errores

Click derecho > maven > actualizar proyecto

//------------------------------------------------------//---------------------------------------------------//

SUREFiRE PLUGiN

clean:clean compiler:compile surefire:test

clean test : para ejcutar sin especificar

clean install : ejecutar sin especificar eh instalar

//------------------------------------------------------//---------------------------------------------------//

ASSEMBLY PLUGiN

Una única face single se utiliza llamando a esta.

Se agrega el plugin al pom.xml

<plugin>

<artifactId>maven-assembly-plugin</artifactId>

<version>3.3.0</version>

<configuration>

<descriptorRefs>

<descriptorRef>jar-with-dependencies</descriptorRef>

</descriptorRefs>

</configuration>

</plugin>

Agregar una dependencia.

En el pom.xml > dependecies > add > commons-math: 2.2 > save

O por pom.xml > explorador

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons-math3</artifactId>

<version>3.2</version>

</dependency>

Empaquetar

clean package assembly:single

//------------------------------------------------------//---------------------------------------------------//

CREAR EJECUTABLE

<configuration>

<archive>

<manifest>

<mainClass>co.empresa.java.App</mainClass>

</manifest>

</archive>

</configuration>

clean package assembly:single

//------------------------------------------------------//---------------------------------------------------//

CAMBiAR EL COMPORTAMiENTO DE LAS FASES

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-assembly-plugin</artifactId>

<version>3.3.0</version>

<configuration>

<descriptorRefs>

<descriptorRef>jar-with-dependencies</descriptorRef>

</descriptorRefs>

<archive>

<manifest>

<mainClass>co.empresa.java.App</mainClass>

</manifest>

</archive>

</configuration>

<executions>

<execution>

<id>crear-ejecutable</id>

<phase>package</phase>

<goals>

<goal>single</goal>

</goals>

</execution>

</executions>

</plugin>

</plugins>

Debe estar fuera del pluginManagmet

Clean > package

//------------------------------------------------------//---------------------------------------------------//

MODiFiCACiON DEL CiCLO DE ViDA

Effective pom > buscamos el plugin > copy > pom.xml > paste an plugin > sobrescribe > quitarle de la face > <phase>none</phase> > delete goals.

Eliminamos el plugin eh inculimos el nuestro

clean package

//------------------------------------------------------//---------------------------------------------------//

PROFiLES

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>co.empresa.maven</groupId>

<artifactId>ejemplo2</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>ejemplo2</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<profiles>

<profile>

<id>produccion</id>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-assembly-plugin</artifactId>

<version>3.3.0</version>

<configuration>

<descriptorRefs>

<descriptorRef>jar-with-dependencies</descriptorRef>

</descriptorRefs>

<archive>

<manifest>

<mainClass>co.empresa.java.App</mainClass>

</manifest>

</archive>

</configuration>

<executions>

<execution>

<id>crear-ejecutable</id>

<phase>package</phase>

<goals>

<goal>single</goal>

</goals>

</execution>

</executions>

</plugin>

<plugin>

<artifactId>maven-jar-plugin</artifactId>

<version>2.4</version>

<executions>

<execution>

<id>default-jar</id>

<phase>none</phase>

</execution>

</executions>

</plugin>

</plugins>

<pluginManagement>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.1</version>

<configuration>

<!--<verbose>true</verbose>-->

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</pluginManagement>

</build>

</profile>

</profiles>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons-math3</artifactId>

<version>3.2</version>

</dependency>

</dependencies>

</project>

actualizar el proyecto click derecho > maven > update proyect

clean package

cambio de perfil

clean package -Pproduccion

o agregamos en profile de eclipce el perfil

//------------------------------------------------------//---------------------------------------------------//

CREANDO ARCHETYPES PROPiOS

clonamos repositorio > perpestiva de git > clonar repositoryo > next > next >finish > importar al area de trabajo > click derecho a la carpeta > import proyect > proyecto general > next > finish

usuario@DESA-5192-T:~/git/cursoMaven/proyectoWeb$ mvn archetype:create-from-proje

se navega hasta la carpeta generada

mvn clean install

//------------------------------------------------------//---------------------------------------------------//

BUSQUEDAS LOCALES POR CONSOLA

Entramos a proyectos > ejecutamos una termianl de comandos > mvn archetype:generate -DarchetypeCatalog=local -Dfilter=es.adama: > elegimos el numero > colocamos los datos,

//------------------------------------------------------//---------------------------------------------------//

GESTiON DE DEPENDENCiAS

SCOPE:

* Compile
* Provided
* Runtine
* Test
* System

DEPENDENCiAS TRANSiTiVAS:

Etiqueta <optional></optional> o <exclusión></exclucion>

PROBLEMAS DE LAS DEPENDENCiAS TRANSiTiVAS

Contruir el árbol de dependencia > ejecutamos > dependency:tree(plugin) > miramos el result en consola

Verlo gráficamente accedemos al pom.xml > hierarchy dependency

Evitando dependencias cíclicas que se relacione n veces con la siguiente eclipce omite la descarga al detectar que ya esta solo es necesario una sola vez

Download Tomcat > descomprimimos > ahora incluir en la config de eclipse > servers > link créate > apache versión 8 > next > buscamoes el derictorio de instalación > next > finish. > lo iniciamos > arrancamos navegador de eclipce > <http://localhost:8080/> > webEmpresaDontCom >Run as > run on server > tomcat > yes. > acceso url/numero.

SUSTiTUCiON DE JCL