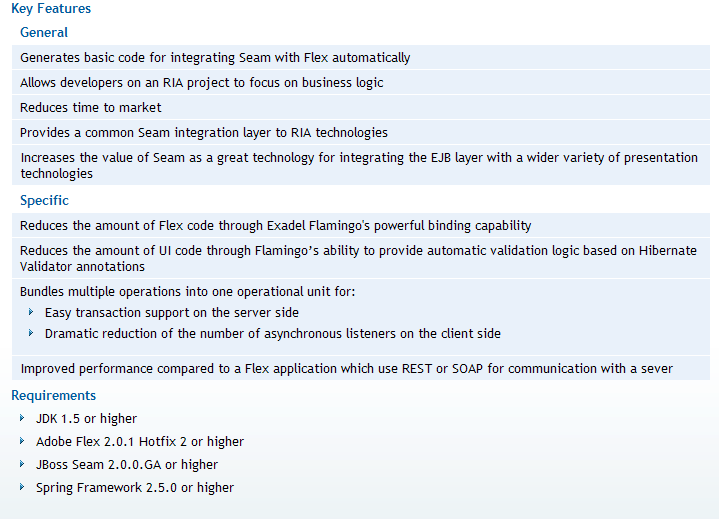
With Exadel Flamingo, Flex, JavaFX,  Swing, J2ME, iPhone SDK, and Android SDK can easily be glued to Seam, Spring, and JavaEE. And, in the future, Flamingo will be extended to provide glue for other RIA technologies.



Otros RIA,

**Opción 1** Adobe Flex

JavaFx **versus** …………

MicroSoft Silverlight

**Opción 2**

AJAX (no necesita plugin-browser)

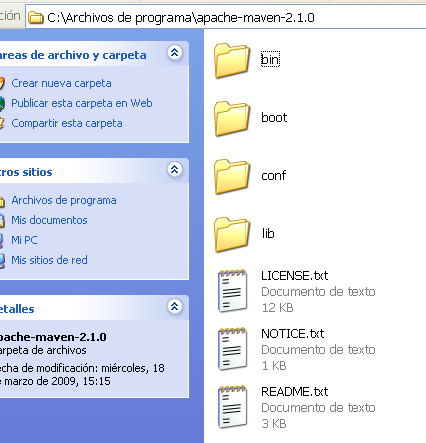
SO moviles

Android (Google)

Symbian (Nokia)

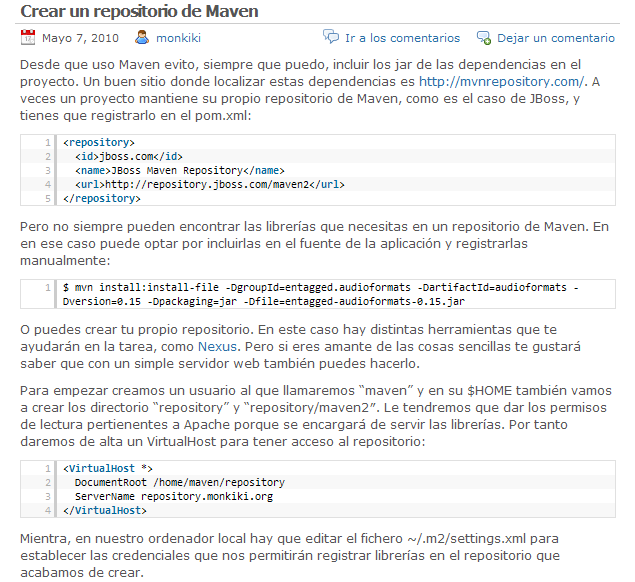
Balckberry(RIM)

# Instal.lar Maven al nostre PC (utilitzem Maven com a gestor de dependències, compilador, generador de wars, jars,…)



# Configuració Repositori Maven + Web

[*http://monkiki.wordpress.com/2010/05/07/crear-un-repositorio-de-maven/*](http://monkiki.wordpress.com/2010/05/07/crear-un-repositorio-de-maven/)





# Instal.lar Flamingo

Java desktop UI

<http://flamingo.exadel.com>

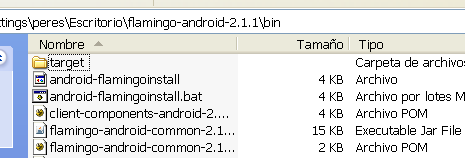
Si volem una versió de Flamingo per suportar únicament Android SDK com a front-end ens baixem la versió 2.1.1

Si volem suportar altres front-ends (Flex, JavaFX ,iphone SDK, Swing,..)

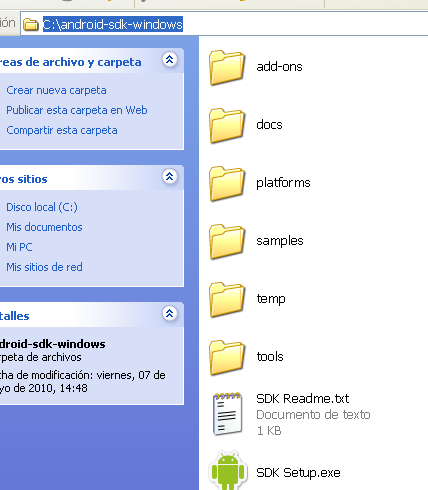
instal.lem la versió 2.1.0

# Incloure llibreries Flamingo al repositori Maven (cas android)

Executar **android-flamingoinstall** shell script from the bin directory. This will install Flamingo libraries into the Maven repository.



# Instal.lar Android SDK + IDE Eclipse + plugin (<https://dl-ssl.google.com/android/eclipse/>) 🡪 Entorn de Desenvolupament

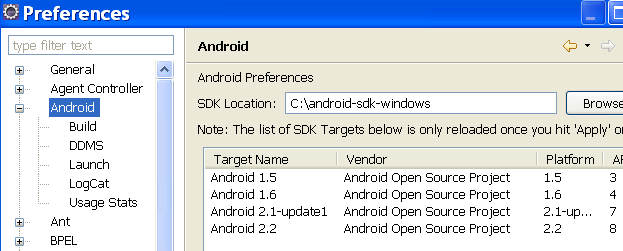


*Crear un nou projecte Android*

## *Plugin Eclipse*



*Open Android SDK i AVD Manager*



*Basat en kernel Linux 2.6.29*

*Basat en kernel Linux 2.6.27*

# EXEMPLE : Crear els projectes Android (UI), Seam (Servidor) , Spring (Servidor) pel projecte que vulguem generar (provem amb un exemple : HelloWorld)

**<path\_projecte>\pom.xml (Ex. : helloworld\pom.xml)**

Projecte Seam

Projecte component Java

*<profile>*

*<id>****android****</id>*

*<modules>*

*<module>common</module>*

*<module>web-seam</module>*

*<module>web-spring</module>*

*<module>android</module>*

Projecte Spring

*</modules>*

*</profile>*

Projecte UI Android

……………

*<profile>*

*<id>****javame****</id>*

*<modules>*

*<module>common</module>*

*<module>web-seam</module>*

Projecte UI JavaME

*<module>web-spring</module>*

*<module>javame</module>*

*</modules>*

*</profile>*

C:\Documents and Settings\peres\Escritorio\flamingo-android-2.1.1\examples\helloworld>**mvn clean install –Pandroid**

[INFO] Scanning for projects...

[INFO] Reactor build order:

[INFO] Exadel Flamingo Helloworld Sample

[INFO] Exadel Flamingo Helloworld Sample :: Java Components

[INFO] Exadel Flamingo Helloworld Sample :: Seam Version

[INFO] Exadel Flamingo Helloworld Sample :: Spring Version

[INFO] Exadel Flamingo Helloworld Sample :: Android UI

[INFO] ------------------------------------------------------------------------

* *helloworld\common\pom.xml*

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <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/maven-v4\_0\_0.xsd">  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>com.exadel.flamingo.samples.helloworld</groupId>  <artifactId>parent</artifactId>  <version>1.0-SNAPSHOT</version>  </parent>  <artifactId>helloworld-sample-common</artifactId>  <name>${projectName} :: Java Components</name>  <build>  <plugins>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-compiler-plugin</artifactId>  <configuration>  <source>1.3</source>  <target>1.1</target>  <encoding>${encoding}</encoding>  </configuration>  </plugin>  </plugins>  </build>  </project> |

[INFO] Building Exadel Flamingo Helloworld Sample [INFO] task-segment: [clean, install]

remove files generated at build-time in a project's directory.

[INFO] ------------------------------------------------------------------------

[INFO] [clean:clean]

[INFO] Deleting directory C:\Documents and Settings\peres\Escritorio\flamingo-android-2.1.1\examples\helloworld\target

Installs a file in the local repository.

[INFO] [site:attach-descriptor]

[INFO] [install:install]

[INFO] Installing C:\Documents and Settings\peres\Escritorio\flamingo-android-2.

1.1\examples\helloworld\target\pom-transformed.xml to C:\Documents and Settings\

peres\.m2\repository\com\exadel\flamingo\samples\helloworld\parent\1.0-SNAPSHOT\parent-1.0-SNAPSHOT.pom

[INFO] ------------------------------------------------------------------------

[INFO] **Building Exadel Flamingo Helloworld Sample :: Java Components**

**[**INFO] task-segment: [clean, install]

[INFO] ------------------------------------------------------------------------

[INFO] [clean:clean]

Copy resources for the main source code to the main output directory.

[INFO] Deleting directory C:\Documents and Settings\peres\Escritorio\flamingo-an droid-2.1.1\examples\helloworld\common\target

[INFO] [resources:resources]

[INFO] Using ‘UTF-8’ encoding to copy filtered resources.

[INFO] Copying 0 resource

Compiles application sources

[INFO] [compiler:compile]

[INFO] Compiling 2 source files to C:\Documents and Settings\peres\Escritorio\fl amingo-android-2.1.1\examples\helloworld\common\target\classes

[INFO] [resources:testResources]

[INFO] Using ‘UTF-8’ encoding to copy filtered resources.

[INFO] skip non existing resourceDirectory C:\Documents and Settings\peres\Escri torio\flamingo-android-2.1.1\examples\helloworld\common\src\test\resources

[INFO] [compiler:testCompile]

[INFO] Nothing to compile - all classes are up to date

[INFO] [surefire:test]

[INFO] No tests to run.

Compiles application sources

[INFO] [jar:jar]

[INFO] Building jar: C:\Documents and Settings\peres\Escritorio\flamingo-android

-2.1.1\examples\helloworld\common\target\helloworld-sample-common-1.0-SNAPSHOT.jar

Instal.lem el component Java en elrepositori Maven

[INFO] [install:install]

[INFO] Installing C:\Documents and Settings\peres\Escritorio\flamingo-android-2.

1.1\examples\helloworld\common\target\helloworld-sample-common-1.0-SNAPSHOT.jar

to C:\Documents and Settings\peres\.m2\repository\com\exadel\flamingo\samples\he

lloworld\helloworld-sample-common\1.0-SNAPSHOT\helloworld-sample-common-1.0-SNAPSHOT.jar

[INFO] ------------------------------------------------------------------------

[INFO] **Building Exadel Flamingo Helloworld Sample :: Seam Version**

[INFO] task-segment: [clean, install]

[INFO] ------------------------------------------------------------------------

[INFO] [clean:clean]

[INFO] Deleting directory C:\Documents and Settings\peres\Escritorio\flamingo-android-2.1.1\examples\helloworld\web-seam\target

[INFO] [resources:resources]

[INFO] Using ‘UTF-8’ encoding to copy filtered resources.

## *[INFO] Copying 1 resource*

[INFO] Copying 1 resource to ../seam-helloworld-sample

[INFO] [compiler:compile]

[INFO] Compiling 1 source file to C:\Documents and Settings\peres\Escritorio\fla mingo-android-2.1.1\examples\helloworld\web-seam\target\classes

war

[INFO] [resources:testResources]

[INFO] Using ‘UTF-8’ encoding to copy filtered resources.

[INFO] skip non existing resourceDirectory C:\Documents and Settings\peres\Escritorio\flamingo-android-2.1.1\examples\helloworld\web-seam\src\test\resources

[INFO] [compiler:testCompile]

[INFO] No sources to compile

[INFO] [surefire:test]

[INFO] No tests to run.

[INFO] [war:war]

[INFO] Packaging webapp

[INFO] Assembling webapp[seam-helloworld-sample-web] in [C:\Documents and Settin

gs\peres\Escritorio\flamingo-android-2.1.1\examples\helloworld\web-seam\target\seam-helloworld-sample]

## *[INFO] Processing war project*

[INFO] Copying webapp resources[C:\Documents and Settings\peres\Escritorio\flamingo-android-2.1.1\examples\helloworld\web-seam\src\main\webapp]

[INFO] Webapp assembled in[297 msecs]

war que deploy al Jboss

[INFO] Building war: C:\Documents and Settings\peres\Escritorio\flamingo-android -2.1.1\examples\helloworld\web-seam\target\seam-helloworld-sample.war

[INFO] [install:install]

[INFO] Installing C:\Documents and Settings\peres\Escritorio\flamingo-android-2.

1.1\examples\helloworld\web-seam\target\seam-helloworld-sample.war to C:\Documents and Settings\peres\.m2\repository\com\exadel\flamingo\samples\helloworld\seam-

helloworld-sample-web\1.0-SNAPSHOT\seam-helloworld-sample-web-1.0-SNAPSHOT.war

war segons Maven

*disk2 helloworld\web-seam\pom.xml*

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <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/maven-v4\_0\_0.xsd">  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>com.exadel.flamingo.samples.helloworld</groupId>  <artifactId>parent</artifactId>  <version>1.0-SNAPSHOT</version>  </parent>  <artifactId>seam-helloworld-sample-web</artifactId>  <packaging>war</packaging>  <name>${projectName} :: Seam Version</name>  <dependencies>  <dependency>  <groupId>com.exadel.flamingo.samples.helloworld</groupId>  Hessian  <artifactId>helloworld-sample-common</artifactId>  </dependency>  <dependency>  <groupId>com.caucho</groupId>  Flamingo  <artifactId>hessian</artifactId>  </dependency>  <dependency>  <groupId>com.exadel.flamingo</groupId>  Seam  <artifactId>flamingo-service</artifactId>  </dependency>  <dependency>  <groupId>org.jboss.seam</groupId>  <artifactId>jboss-seam</artifactId>  </dependency>  </dependencies>  <build>  <finalName>seam-helloworld-sample</finalName>  <resources>  <resource>  <directory>src/main/resources</directory>  </resource>  <resource>  <directory>src/main/filtered-resources</directory>  <filtering>true</filtering>  <targetPath>../${project.build.finalName}</targetPath>  </resource>  </resources>  </build>  </project> |

[INFO] ------------------------------------------------------------------------

[INFO] **Building Exadel Flamingo Helloworld Sample :: Spring Version** [INFO] task-segment: [clean, install]

[INFO] ------------------------------------------------------------------------

[INFO] [clean:clean]

[INFO] Deleting directory C:\Documents and Settings\peres\Escritorio\flamingo-android-2.1.1\examples\helloworld\web-spring\target

[INFO] [resources:resources]

[INFO] Using ‘UTF-8’ encoding to copy filtered resources.

## *[INFO] Copying 1 resource*

[INFO] Copying 1 resource to ../spring-helloworld-sample

[INFO] [compiler:compile]

[INFO] Compiling 1 source file to C:\Documents and Settings\peres\Escritorio\flamingo-android-2.1.1\examples\helloworld\web-spring\target\classes

[INFO] [resources:testResources]

[INFO] Using ‘UTF-8’ encoding to copy filtered resources.

[INFO] skip non existing resourceDirectory C:\Documents and Settings\peres\Escri torio\flamingo-android-2.1.1\examples\helloworld\web-spring\src\test\resources

[INFO] [compiler:testCompile]

[INFO] No sources to compile

[INFO] [surefire:test]

[INFO] No tests to run.

[INFO] [war:war]

[INFO] Packaging webapp

[INFO] Assembling webapp[spring-helloworld-sample-web] in [C:\Documents and Settings\peres\Escritorio\flamingo-android-2.1.1\examples\helloworld\web-spring\targ et\spring-helloworld-sample]

## *[INFO] Processing war project*

[INFO] Copying webapp resources[C:\Documents and Settings\peres\Escritorio\flamingo-android-2.1.1\examples\helloworld\web-spring\src\main\webapp]

[INFO] Webapp assembled in[344 msecs]

war segons Maven

war que deploy al Jboss

[INFO] Building war: C:\Documents and Settings\peres\Escritorio\flamingo-android -2.1.1\examples\helloworld\web-spring\target\spring-helloworld-sample.war

[INFO] [install:install]

[INFO] Installing C:\Documents and Settings\peres\Escritorio\flamingo-android-2.

1.1\examples\helloworld\web-spring\target\spring-helloworld-sample.war to C:\Documents and Settings\peres\.m2\repository\com\exadel\flamingo\samples\helloworld\

spring-helloworld-sample-web\1.0-SNAPSHOT\spring-helloworld-sample-web-1.0-SNAPSHOT.war

[INFO] ----------------------------------------------------------------------

[INFO] **Building Exadel Flamingo Helloworld Sample :: Android UI**

[INFO] task-segment: [clean, install]

[INFO] ----------------------------------------------------------------------

* *helloworld\android\pom.xml*

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <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/maven-v4\_0\_0.xsd">  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>com.exadel.flamingo.samples.helloworld</groupId>  <artifactId>parent</artifactId>  <version>1.0-SNAPSHOT</version>  </parent>  <artifactId>helloworld-sample-android</artifactId>  <name>${projectName} :: Android UI</name>  common sample  <packaging>apk</packaging>  <dependencies>  <dependency>  <groupId>com.exadel.flamingo.samples.helloworld</groupId>  <artifactId>helloworld-sample-common</artifactId>  </dependency>  <dependency>  <groupId>android</groupId>  <artifactId>android</artifactId>  <version>${android.platform}</version>  android-common  <scope>provided</scope>  </dependency>  <dependency>  <groupId>com.exadel.flamingo.android</groupId>  <artifactId>flamingo-android-common</artifactId>  </dependency>  </dependencies>  <build>  <resources>  <resource>  <directory>res</directory>  </resource>  </resources>  <plugins>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-compiler-plugin</artifactId>  <configuration>  <source>1.5</source>  <target>1.5</target>  <encoding>${encoding}</encoding>  </configuration>  </plugin>  <plugin>  <groupId>com.jayway.maven.plugins.android.generation2</groupId>  <artifactId>maven-android-plugin</artifactId>  <version>2.3.3</version>  <extensions>true</extensions>  <configuration>  <sdk>  <path>${env.ANDROID\_HOME}</path>  <platform>${android.apilevel}</platform>  </sdk>  <undeployBeforeDeploy>true</undeployBeforeDeploy>  <deleteConflictingFiles>true</deleteConflictingFiles>  </configuration>  </plugin>  </plugins>  </build>  </project> |

[INFO] [clean:clean]

Downloading: <http://repository.jboss.org/maven2/android/android/1.6_r2/android-1.6_r2.pom>

[INFO] Unable to find resource ‘android:android:pom:1.6\_r2’ in repository repository.jboss.org (<http://repository.jboss.org/maven2>)

Downloading: <http://repository.springsource.com/maven/bundles/release/android/an>

droid/1.6\_r2/android-1.6\_r2.pom

[INFO] Unable to find resource ‘android:android:pom:1.6\_r2’ in repository com.springsource.repository.bundles.release (<http://repository.springsource.com/maven/> bundles/release)

Downloading: <http://repository.springsource.com/maven/bundles/external/android/a>

ndroid/1.6\_r2/android-1.6\_r2.pom

[INFO] Unable to find resource ‘android:android:pom:1.6\_r2’ in repository com.springsource.repository.bundles.external (<http://repository.springsource.com/maven> /bundles/external)

Downloading: <http://maven.springframework.org/milestone/android/android/1.6_r2/a>

ndroid-1.6\_r2.pom

[INFO] Unable to find resource ‘android:android:pom:1.6\_r2’ in repository com.springsource.repository.bundles.milestone (<http://maven.springframework.org/milest> one)

Downloading: <http://repository.springsource.com/maven/bundles/milestone/android/>

android/1.6\_r2/android-1.6\_r2.pom

[INFO] Unable to find resource ‘android:android:pom:1.6\_r2’ in repository com.springsource.repository.bundles.milestone.alt (<http://repository.springsource.com/> maven/bundles/milestone)

Downloading: <http://repo1.maven.org/maven2/android/android/1.6_r2/android-1.6_r2>

## .pom

[INFO] Unable to find resource ‘android:android:pom:1.6\_r2’ in repository central (<http://repo1.maven.org/maven2>)

Downloading: <http://repository.jboss.org/maven2/android/android/1.6_r2/android-1>

## *.6\_r2.jar*

[INFO] Unable to find resource ‘android:android:jar:1.6\_r2’ in repository repository.jboss.org (<http://repository.jboss.org/maven2>)

Downloading: <http://repository.springsource.com/maven/bundles/release/android/an>

droid/1.6\_r2/android-1.6\_r2.jar

[INFO] Unable to find resource ‘android:android:jar:1.6\_r2’ in repository com.springsource.repository.bundles.release (<http://repository.springsource.com/maven/> bundles/release)

Downloading: <http://repository.springsource.com/maven/bundles/external/android/a>

ndroid/1.6\_r2/android-1.6\_r2.jar

[INFO] Unable to find resource ‘android:android:jar:1.6\_r2’ in repository com.springsource.repository.bundles.external (<http://repository.springsource.com/maven> /bundles/external)

Downloading: <http://maven.springframework.org/milestone/android/android/1.6_r2/a>

ndroid-1.6\_r2.jar

[INFO] Unable to find resource ‘android:android:jar:1.6\_r2’ in repository com.springsource.repository.bundles.milestone (<http://maven.springframework.org/milest>one)

Downloading: <http://repository.springsource.com/maven/bundles/milestone/android/>

android/1.6\_r2/android-1.6\_r2.jar

[INFO] Unable to find resource ‘android:android:jar:1.6\_r2’ in repository com.springsource.repository.bundles.milestone.alt (<http://repository.springsource.com/> maven/bundles/milestone)

Downloading: <http://repo1.maven.org/maven2/android/android/1.6_r2/android-1.6_r2>.jar

[INFO] Unable to find resource ‘android:android:jar:1.6\_r2’ in repository central (<http://repo1.maven.org/maven2>)

[INFO] ----------------------------------------------------------------------

### [ERROR] BUILD ERROR

[INFO] ----------------------------------------------------------------------

[INFO] Failed to resolve artifact.

Missing:

1) android:android:jar:1.6\_r2

* Try downloading the file manually from the project website.
* Then, install it using the command:

*mvn install:install-file -****DgroupId****=android -****DartifactId****=android -****Dversion****=*

*1.6\_r2 -****Dpackaging****=jar -****Dfile****=/path/to/file*

Exemple :

*mvn install:install-file -DgroupId=android -DartifactId=android -Dversion=1.6\_r2 -Dpackaging=jar -Dfile=c:\android-1.6\_r2.jar*

Alternatively, if you host your own repository you can deploy the file there: mirar 2.

*mvn deploy:deploy-file -****DgroupId****=android -****DartifactId****=android -****Dversion****=1.6\_r2 -****Dpackaging****=jar -****Dfile****=/path/to/file -****Durl****=[url] -****DrepositoryId****=[id]*

*http://repository.summa.org/maven2*

*<?xml version=”1.0” encoding=”UTF-8”?>*

*<settings>*

*<localRepository>c:\Documents and Settings\peres\.m2/repository</localRepository>*

*<servers>*

*<server>*

*<id>summa.org</id>*

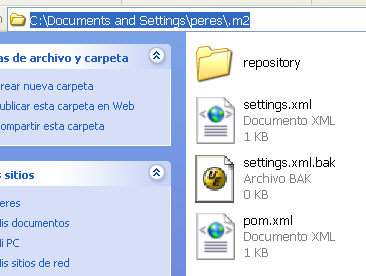
*<username>maven</username>*

*<password>s3cr3t0</password>*

*</server>*

*</servers>*

*</settings>*



*versió*

*artifactId*

*groupId*

*packaging*

Path to dependency:

1) com.exadel.flamingo.samples.helloworld:helloworld-sample-android:apk:

1.0-SNAPSHOT

2) android:android:jar:1.6\_r2

**1 required artifact is missing.**

for artifact:

com.exadel.flamingo.samples.helloworld:helloworld-sample-android:apk:1.0-SNAPSHOT

from the specified remote repositories:

com.springsource.repository.bundles.release (<http://repository.springsource.co>m/maven/bundles/release),

com.springsource.repository.bundles.milestone.alt (<http://repository.springsou>rce.com/maven/bundles/milestone),

com.springsource.repository.bundles.external (<http://repository.springsource.c>om/maven/bundles/external),

central (<http://repo1.maven.org/maven2>),

com.springsource.repository.bundles.milestone (<http://maven.springframework.or>g/milestone),

repository.jboss.org (<http://repository.jboss.org/maven2>)

[INFO] ----------------------------------------------------------------------

[INFO] For more information, run Maven with the -e switch

[INFO] ----------------------------------------------------------------------

## *[INFO] Total time: 17 seconds*

[INFO] Finished at: Thu Jul 08 09:29:30 CEST 2010 [INFO] Final Memory: 27M/49M

[INFO] ----------------------------------------------------------------------

# Instal.lar el jar/war/.. que falta al Repositori Maven

C:\Documents and Settings\peres\Escritorio\flamingo-android-2.1.1\examples\hello world\android>***mvn install:install-file -DgroupId=android -DartifactId=android -D version=1.6\_r2 -Dpackaging=jar -Dfile=c:\android-1.6\_r2.jar***

[INFO] Scanning for projects...

[INFO] Searching repository for plugin with prefix: ‘install’.

[INFO] ------------------------------------------------------------------------

[INFO] Building Exadel Flamingo Helloworld Sample :: Android UI

[INFO] task-segment: [install:install-file] (aggregator-style)

[INFO] ------------------------------------------------------------------------

[INFO] [install:install-file]

[INFO] Installing c:\android-1.6\_r2.jar to C:\Documents and Settings\peres\.m2\repository\android\android\1.6\_r2\android-1.6\_r2.jar

[INFO] ----------------------------------------------------------------------

### [INFO] BUILD SUCCESSFUL

[INFO] ----------------------------------------------------------------------

## *[INFO] Total time: 1 second*

[INFO] Finished at: Thu Jul 08 11:33:59 CEST 2010 [INFO] Final Memory: 5M/10M

[INFO] ----------------------------------------------------------------------

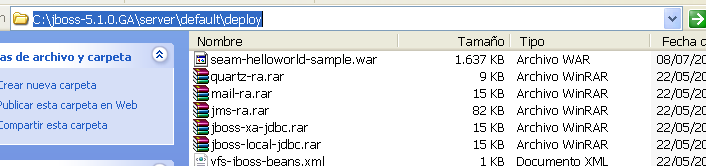
1. Deploy aplicació war (seam/spring) a JBoss AS 5.1

*<path\_flamingo-android-2.1.1>\examples\helloworld\web-seam* (Seam)

**

*<path\_flamingo-android-2.1.1>\examples\helloworld\web-spring* (Spring)

**

**

1. Executar Android emulator

Des de l’Eclipse



o, des de *<path\_android>\android-sdk-windows\tools\emulator.exe*

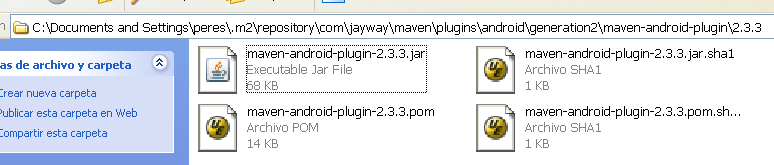
*Plugin : groupID*

10. Deploy aplicació Android

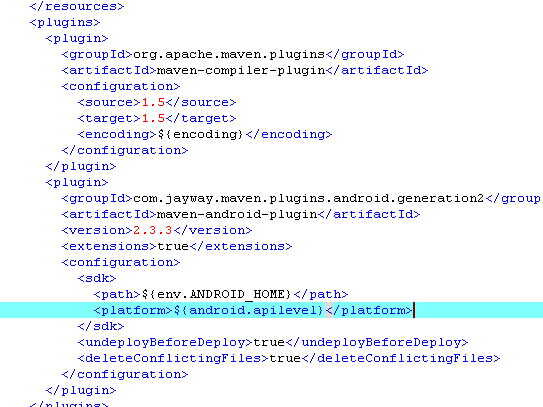
*Plugin : artifactID*

***mvn com.jayway.maven.plugins.android.generation2:maven-android-plugin:deploy***  des de *<path\_flamingo-android-2.1.1>\examples\helloworld\android*

*deploy*

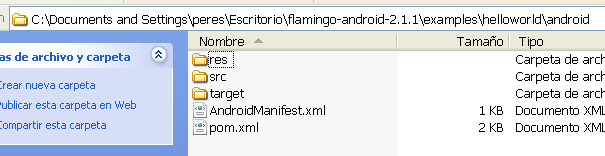


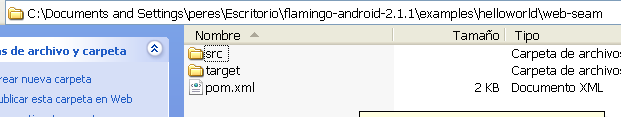
*Pom.xml*



1. Crear projecte Maven

*Estructura Estàndard de projectes Maven*





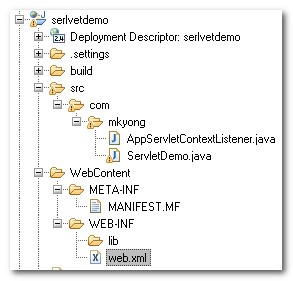
* *mvn archetype:generate* ***-DgroupId****=com.mycompany.app -****DartifactId****=my-app* ***-DarchetypeArtifactId****=maven-archetype*
* *cd my-app*
* *mvn package (genera el jar, war, o el que sigui)*
* *java -cp target/my-app-1.0-SNAPSHOT.jar com.mycompany.app.App*

*jar*

*Classe*

Convertir Java Web Project a Maven Project

A simple Java servlet web application , with one “javaee.jar” dependency library



#### 1. Create Maven web project folder structure

- Create the following new Maven’s folder structure.  
- Move all exiting java source files to the “\src\main\java”.  
- Move “web.xml” file to the “\src\main\webapp\WEB-INF”.  
- Create a new xml file named “pom.xml” , put under the project root folder.

Please refer to [Maven Standard Directory Layout](http://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html)

\--projectname

\--src

\--main

\--java (all java source here)

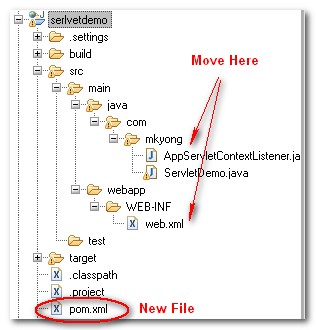
\--test

\--- (all unit test here)

\--webapp

\--WEB-INF

\--web.xml



#### 2. Project details

- Fill in the existing project details in “pom.xml” file, add common remote repository, war plugin and compiler plugin.

**<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/maven-v4\_0\_0.xsd"**>**

**<modelVersion>**4.0.0**</modelVersion>**

**<groupId>**com.mkyong**</groupId>**

**<artifactId>**serlvetdemo**</artifactId>**

**<packaging>**jar**</packaging>**

**<version>**1.0-SNAPSHOT**</version>**

**<name>**serlvetdemo**</name>**

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

**<repositories>**

**<repository>**

**<id>**java.net**</id>**

**<url>**http://download.java.net/maven/2**</url>**

**</repository>**

**</repositories>**

**<build>**

**<plugins>**

**<plugin>**

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

**<artifactId>**maven-war-plugin**</artifactId>**

**<configuration>**

**<webResources>**

**<resource>**

**<directory>**${basedir}/src/main/java**</directory>**

**<targetPath>**WEB-INF/classes**</targetPath>**

**<includes>**

**<include>**\*\*/\*.properties**</include>**

**<include>**\*\*/\*.xml**</include>**

**<include>**\*\*/\*.css**</include>**

**<include>**\*\*/\*.html**</include>**

**</includes>**

**</resource>**

**</webResources>**

**</configuration>**

**</plugin>**

**<plugin>**

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

**<configuration>**

**<source>**1.6**</source>**

**<target>**1.6**</target>**

**</configuration>**

**</plugin>**

**</plugins>**

**</build>**

**</project>**

#### 3. Configure the dependency libraries

This is the most annoying and time consuming session, you need to add your project’s dependency libraries in “pom.xml” manually. Try find your project’s dependency libraries in [Maven Central Repository](http://repo1.maven.org/maven2) , [Java.net Maven Repository](http://download.java.net/maven/2) , or other remote repository. Patient… find the Maven library code and include it in “pom.xml” file :)

P.S For custom library, you need to [*install it into your Maven Local Repository manually*](http://www.mkyong.com/maven/how-to-include-library-manully-into-maven-local-repository/).

##### Full Maven “pom.xml” example – with dependency library

**<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/maven-v4\_0\_0.xsd"**>**

**<modelVersion>**4.0.0**</modelVersion>**

**<groupId>**com.mkyong**</groupId>**

**<artifactId>**serlvetdemo**</artifactId>**

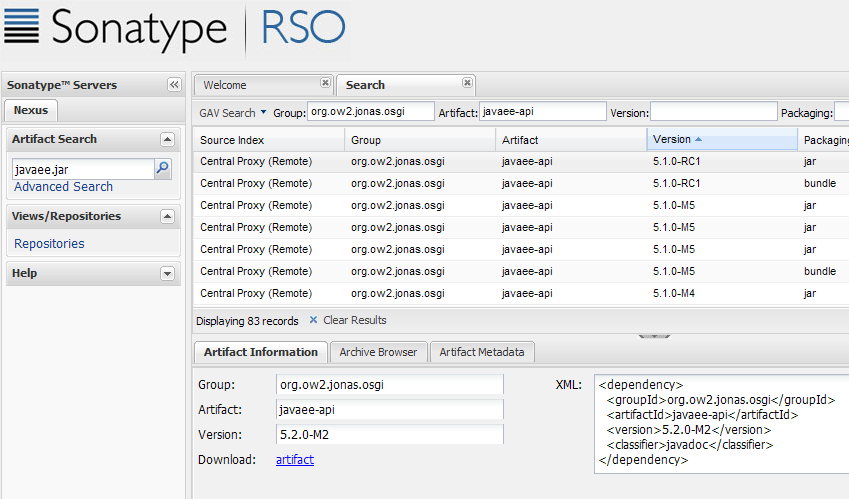
**<packaging>**jar**</packaging>**

**<version>**1.0-SNAPSHOT**</version>**

**<name>**serlvetdemo**</name>**

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

[*https://repository.sonatype.org/index.html*](https://repository.sonatype.org/index.html)



**<repositories>**

**<repository>**

**<id>**java.net**</id>**

**<url>**http://download.java.net/maven/2**</url>**

**</repository>**

**</repositories>**

**<dependencies>**

**<dependency>**

**<groupId>**javax**</groupId>**

**<artifactId>**javaee-api**</artifactId>**

**<version>**6.0**</version>**

**</dependency>**

**</dependencies>**

**<build>**

**<plugins>**

*Fonts a compilar*

*<!-- Maven War file generator plugin -->*

**<plugin>**

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

**<artifactId>maven-war-plugin</artifactId>**

**<configuration>**

**<webResources>**

**<resource>**

**<directory>**${basedir}/src/main/java**</directory>**

**<targetPath>**WEB-INF/classes**</targetPath>**

**<includes>**

**<include>**\*\*/\*.properties**</include>**

**<include>**\*\*/\*.xml**</include>**

**<include>**\*\*/\*.css**</include>**

**<include>**\*\*/\*.html**</include>**

**</includes>**

**</resource>**

**</webResources>**

**</configuration>**

**</plugin>**

*<!-- Maven compiler plugin -->*

**<plugin>**

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

**<configuration>**

**<source>**1.6**</source>**

**<target>**1.6**</target>**

**</configuration>**

**</plugin>**

**</plugins>**

**</build>**

**</project>**

#### 4. Compile It – “mvn compile”

- Compile it and make sure build is successful. Maven will download all the dependency libraries into your local repository now.

E:\workspace\serlvetdemo**>**mvn compile

**[**INFO**]** Scanning **for** projects...

.......

**[**INFO**]** ------------------------------------------------------------------------

**[**INFO**]** BUILD SUCCESSFUL

**[**INFO**]** ------------------------------------------------------------------------

#### 5. Update project settings

Update the existing Eclipse setting with “**mvn eclipse:eclipse**“, all the previous Eclipse errors will gone :)

E:\workspace\serlvetdemo**>**mvn eclipse:eclipse

**[**INFO**]** Scanning **for** projects...

**[**INFO**]** Searching repository **for** plugin with prefix: 'eclipse'.

...............

**[**INFO**]** ------------------------------------------------------------------------

**[**INFO**]** BUILD SUCCESSFUL

**[**INFO**]** ------------------------------------------------------------------------

**[**INFO**]** Total **time**: 2 seconds

**[**INFO**]** Finished at: Wed Dec 02 17:48:24 SGT 2009

**[**INFO**]** Final Memory: 7M**/**14M

**[**INFO**]** ------------------------------------------------------------------------

#### 6. Generate WAR file for deployment

- Generate project’s WAR file with “mvn war:war”, the new generated WAR file will store in “/rootproject/target/” folder named “serlvetdemo-1.0-SNAPSHOT.war”, Maven will pack it with all dependency libraries, classes and create the deployment structure automatically.

E:\workspace\serlvetdemo**>**mvn war:war

**[**INFO**]** Scanning **for** projects...

.......

**[**INFO**]** Processing war project

**[**INFO**]** Copying webapp resources**[**E:\workspace\serlvetde

**[**INFO**]** Webapp assembled **in[**47 msecs**]**

**[**INFO**]** Building war: E:\workspace\serlvetdemo\target\s

**[**INFO**]** -----------------------------------------------

**[**INFO**]** BUILD SUCCESSFUL

**[**INFO**]** -----------------------------------------------

...

#### 7. Done

Your war file is ready for the deployment.

The whole conversion processes are mainly involve the folder structure and “pom.xml”, this is what Maven is required. The dependency libraries setting is very time consuming especially if your project has 30-40 dependency libraries, convert it with patient :). Hurry up , convert your existing old Java project to Maven now!

#### For Eclipse IDE users only

After the migration, you may find out Eclipse will deploy your “web.xml” file to a wrong folder during debugging session. This is cause by the Eclipse deployment setting. you have to update the Eclipse deployment setting file as well , [check this](http://www.mkyong.com/maven/tomcat-deploy-maven-project-web-xml-to-a-wrong-folder-in-eclipse/).

#### For Eclipse Dynamic web project

If you want to convert the Maven project to Dynamic Web Project in Eclipse, make sure the packaging element in “pom.xml” is “war” and issue the following command.

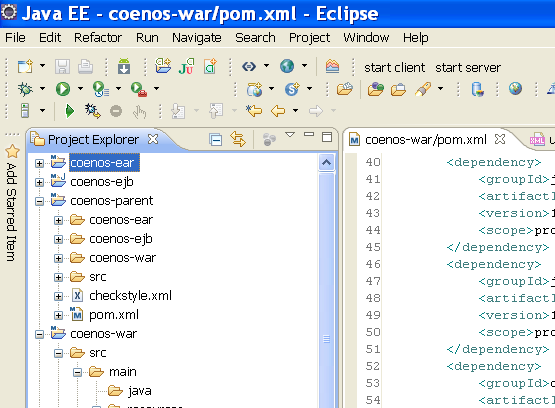
mvn eclipse:eclipse -Dwtpversion=1.5

Check here for detail about [how to use Maven to create a Dynamic Web Project in Eclipse.](http://www.mkyong.com/maven/how-do-use-maven-to-create-a-dynamic-web-project-in-eclipse/)

**Estructura Projecte Seam + Maven + Eclipse**

[Seam Maven2 skeleton Eclipse project for your](http://seamframework.org/Community/SeamMaven2SkeletonEclipseProjectForYourConvenience)  (Coenos)

Failed to execute goal com.jayway.maven.plugins.android.generation2:maven-android-plugin:2.3.3:generate-sources (default-generate-sources) on project helloworld-sample-android: Execution default-generate-sources of goal com.jayway.maven.plugins.android.generation2:maven-android-plugin:2.3.3:generate-sources failed: An API incompatibility was encountered while executing com.jayway.maven.plugins.android.generation2:maven-android-plugin:2.3.3:generate-sources: java.lang.NoSuchMethodError: org.codehaus.plexus.util.cli.Commandline.getPid()J



Resum Maven

*mvn archetype:create \*

*-DarchetypeGroupId=org.apache.maven.archetypes \*

*-DgroupId=com.mycompany.app \*

*-DartifactId=my-app*

crea un projecte Maven2 simple

*Maven Goal*

*plugin*

*mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false*

crea un projecte Maven2 basat en un archetype

|  |  |
| --- | --- |
| **Archetype ArtifactIds** | **Description** |
| maven-archetype-archetype | An archetype which contains a sample archetype. |
| maven-archetype-j2ee-simple | An archetype which contains a simplifed sample J2EE application. |
| maven-archetype-mojo | An archetype which contains a sample a sample Maven plugin. |
| maven-archetype-plugin | An archetype which contains a sample Maven plugin. |
| maven-archetype-plugin-site | An archetype which contains a sample Maven plugin site. |
| maven-archetype-portlet | An archetype which contains a sample JSR-268 Portlet. |
| maven-archetype-quickstart | An archetype which contains a sample Maven project. |
| maven-archetype-simple | An archetype which contains a simple Maven project. |
| maven-archetype-site | An archetype which contains a sample Maven site which demonstrates some of the supported document types like APT, XDoc, and FML and demonstrates how to i18n your site. |
| maven-archetype-site-simple | An archetype which contains a sample Maven site. |
| maven-archetype-webapp | An archetype which contains a sample Maven Webapp project. |

*mvn compile*

compila un projecte Maven a partir del pom.xml

*mvn package*

crea el package (jar , war , rar, o el que sigui)

*mvn install*

