

Hemos visto en los post anteriores como instalar el plugin de eclipse para gestionar proyectos de maven . En este post veremos como definir un conjunto de dependencias de librerias para nuestro proyecto. En nuestro caso vamos a utilizar JSF 2.0 y Log4J para crear una aplicación sencilla. Para ello necesitaremos acceder al repositorio principal de maven que es el que tiene instalados todos los artefactos y buscar los que nosotros necesitamos (JSF y Log4j) con sus versiones especificas.



La url de acceso al repositorio central de Maven es :

<http://www.mvnrepository.com>

Una vez accedemos a esta URL vemos la siguiente página que nos permite buscar artefactos.

Repository

- Plugins
- Tag Cloud

Artifacts/Jars

467k

233k

0

2004 2013

Artifacts/Year

What's New in Maven

RSS ATOM

Apr 27 » [net.java.dev.jogl](#) » [jogl](#) » [1.1.1-rc6](#)

[Go to Java OGL Binding - Basic Package](#)

Apr 27 » [ru.concerteza.util](#) » [ctz-utils](#) » [2.13](#)

Various utilities library

[Go to Con Certeza Utilities Library](#)

Apr 27 » [tv.cntt](#) » [xitrum_2.10](#) » [2.3](#)

xitrum

Web site

Vodafone In

Añade gratis

Si buscamos log4j nos aparecerá lo siguiente :

Repository

- Plugins
- Tag Cloud

Artifacts/Jars

467k

233k

0

2004 2013

Artifacts/Year

Search Results: log4j

[Apache Log4j](#)

Apache Log4j 1.2

[log4j](#) » [log4j](#)

[SLF4J LOG4J-12 Binding](#)

SLF4J LOG4J-12 Binding

[org.slf4j](#) » [slf4j-log4j12](#)

Web site

“MM”

Localizado Log4j pulsamos sobre el link de Apache Log4j y nos aparecerá la siguiente página :

mvnrepository.com/artifact/log4j/log4j

Más visitados ▾ Linux Mint Community Forums Blog News ▾

MVNREPOSITORY

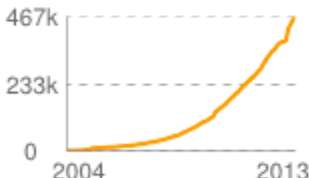
 +1 1.3k *Search by group, artifact or description.
E.g: log4j, spring, hibernate*

home » log4j » log4j

Repository

- [Plugins](#)
- [Tag Cloud](#)

Artifacts/Jars



467k
233k
0

2004 2013

Artifacts/Year

Popular Tags

[ajax](#) [analysis](#)
[annotations](#) [ant](#) [apache](#)

Apache Log4j

Apache Log4j 1.2

tags: [logging](#)

Available versions

✓ **Ad muted.** [Undo](#) We'll do our best to show you more relevant ads in the future. Help us show you better ads by updating your [ads preferences](#).

Version	Type	Download
1.2.17	release	Binary (0 bytes)
1.2.16	release	Binary (471 KB)
1.2.15	release	Binary (383 KB)

En la cual podemos enlazar a las distintas versiones de artefactos que el repositorio tiene almacenadas. Pulsamos sobre la 1.2.17 y accederemos a la información que necesitamos para nuestro proyecto en concreto.




1.3k

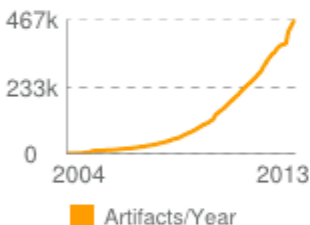
Search by group, artifact or description.
 E.g: log4j, spring, hibernate

home » log4j » log4j » 1.2.17

Repository

- Plugins
- Tag Cloud

Artifacts/Jars



Artifacts/Year

Popular Tags

ajax analysis
 annotations ant apache
 api archetype aspect
 asynchronously beans
 binding bpm build
 buildsystem
 bytecode cache cms

Apache Log4j

Apache Log4j 1.2

Artifact	Download (BUNDLE) (0 bytes)
POM File	View
HomePage	http://logging.apache.org/log4j/1.2/
Organization	Apache Software Foundation
Issue Tracker	https://issues.apache.org/bugzilla/describecomponents.cgi?product=Log4j

Maven

Ivy

Grape

Gradle

Buildr

SBT

```

<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
</dependency>

```

Deberemos copiarnos la sección XML de dependencias que nos aparece en la página:

```

&lt;dependency&gt;
  &lt;groupId&gt;log4j&lt;/groupId&gt;
  &lt;artifactId&gt;log4j&lt;/artifactId&gt;
  &lt;version&gt;1.2.17&lt;/version&gt;
&lt;/dependency&gt;

```

Una vez hecho esto deberemos repetir la misma operación para las dependencias de JSF que son las siguientes :

```
&lt;dependency&gt;
&lt;groupId&gt;com.sun.faces&lt;/groupId&
&lt;artifactId&gt;jsf-api&lt;/artifactId&
&lt;version&gt;2.1.20&lt;/version&
&lt;/dependency&gt;
&lt;pre&gt;&nbsp;
```

```
&lt;dependency&gt;
&lt;groupId&gt;com.sun.faces&lt;/groupId&
&lt;artifactId&gt;jsf-impl&lt;/artifactId&
&lt;version&gt;2.1.20&lt;/version&
&lt;/dependency&gt;
&lt;pre&gt;
```

Realizada esta operación añadiremos todas las dependencias en nuestro fichero pom.xml en la sección de dependencias(<dependencies>)

```
&lt;project
xmlns=&quot;http://maven.apache.org/POM/4.0.0&quot;
xmlns:xsi=&quot;http://www.w3.org/2001/XMLSchema-instance&quot;
```

```

    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>com.arquitecturajava.proyecto1</groupId>
    <artifactId>proyecto1</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>war</packaging>
    <dependencies>
    <dependency>
    <groupId>com.sun.faces</groupId>
    <artifactId>jsf-api</artifactId>
    <version>2.1.20</version>
    </dependency>
    <dependency>
    <groupId>com.sun.faces</groupId>
    <artifactId>jsf-impl</artifactId>
    <version>2.1.20</version>
    </dependency>
    <dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.17</version>

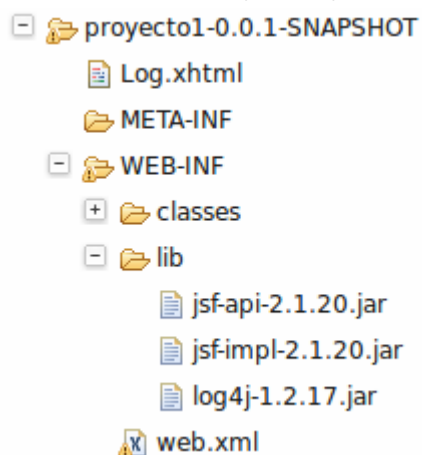
```

</dependency>

</dependencies>

</project>

Una vez definidas las dependencias si empaquetamos la aplicación con “mvn package” podremos ver como Maven se baja del repositorio central las dependencias que necesita para nuestro proyecto y las añade a la carpeta lib.



Hecho esto podemos proceder a construir la aplicación que realizará log sin ningún tipo de problema ya que tenemos bien definido el conjunto de dependencias. En casos mas complejos podemos encontrarnos con artefactos que dependen de otros artefactos en cuyo caso la herramienta se encargará de bajar todo lo necesario eliminandonos problemas.