

Instal·lació de Tomcat i Maven per al desplegament d'aplicacions Java

Guillermo Vidal Frasquet

Desplegament
d'Aplicacions Web
Pràctica



Continguts

1 Instal·lació de Tomcat i Maven per al desplegament d'aplicacions Java	2
1.1 Introducció	2
1.1.1 Per què?	2
1.2 Instal·lació de Tomcat	2
1.2.1 Desplegament manual mitjançant la GUI d'administració	3
1.2.2 Tasca 1	3
1.3 Instal·lació de Maven	3
1.3.1 Configuració de Maven	4
1.3.2 Desplegament amb Maven	5
1.3.3 Tasca 2	7

1 Instal·lació de Tomcat i Maven per al desplegament d'aplicacions Java

1.1 Introducció

Si consultem l'apartat de [versions de Tomcat](#) en la seua pàgina oficial, ens adonarem que no usarem l'última versió, la 11, per a esta pràctica, sinó la 9. La pregunta és quasi inevitable:

1.1.1 Per què?

En l'enllaç anterior veiem com des de la seua versió 9, **Tomcat** dona suporta a Java 8 i superiors. No obstant això, a partir de la versió 10.1.x, dona suport a Java 11 i superiors.

Què significa això?

En Java 9 es van introduir novetats com un nou sistema de mòduls (**Jigsaw**), entre altres.

A Java 11 es va fer un pas més en haver canviat de nom completament les rutes de paquets `javax` a `jakarta`. Oracle, malgrat haver fet públic el desenvolupament de Java, no va fer el mateix amb el seu nom.

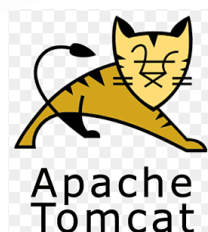
Així les coses, resulta que Java 8 pot ser que hui dia encara siga la més usada en projectes reals. Podria realitzar-se un procés de migració d'un projecte de Java 8 a Java 11 i utilitzar-ho en **Tomcat 11**.

No obstant això, per a Java 8 el seu suport per a ús comercial (pagant) va acabar al març de 2022, però per a ús no comercial segueix fins a 2030.

En conclusió, no és rar trobar-se en el món real un projecte a desplegar realitzat a Java 8. Podria realitzar-se una migració i els conceptes de desplegament que veurem continuarien aplicant. Així les coses, per facilitar en la realització de les pràctiques utilitzarem **Tomcat 9** i el plugin oficial de **Maven** per a **Tomcat 7** per al desplegament (després veurem el motiu).

1.2 Instal·lació de Tomcat

Esta pràctica és molt senzilla i consistirà a realitzar la instal·lació del servidor d'aplicacions **Tomcat 9**, en una màquina virtual corrent Debian 11 Bullseye.



Es pot fer tant amb l'administrador de paquets **apt** com de manera manual. La forma més recomanable per la seua senzillesa és la primera.

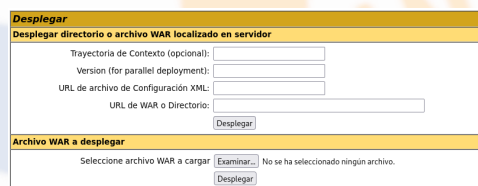
Per a això, i com a suggeriment, podeu recolzar-vos en este [tutorial](#) en línia, encara que sou lliures de consultar tantes fonts com desitgeu.

Òbviament, heu d'utilitzar el vostre propis usuaris i contrasenya.

1.2.1 Desplegament manual mitjançant la GUI d'administració

Realitzarem el desplegament manual d'una aplicació ja prèviament empaquetada en format **WAR**. Per a això:

1. Fem login amb l'usuari prèviament creat.
2. Busquem la secció que ens permet desplegar un WAR manualment, seleccionem el nostre arxiu i el despleguem.



Després d'estos passos, se'ns llistarà l'aplicació ja desplegada com un directori més i podrem accedir a ella.

1.2.2 Tasca 1

Documenta el desplegament manual de l'aplicació que us podeu descarregar per a tal efecte en Aules (arxiu .war).

1.3 Instal·lació de Maven

Per a instal·lar **Maven** en el nostre Debian tenim, de nou, dues opcions:

1. Instal·lació mitjançant gestor de paquets APT.
2. Instal·lació manual.

La primera, recomanada, és molt més senzilla i automatitzada (estableix tots els paths i variables d'entorn), encara que amb la segona es podria aconseguir un paquet més actualitzat.

Tots dos mètodes venen explicats [ací](#).

Si decidim seguir el primer mètode, el més senzill, veiem que és tan simple com actualitzar els repositoris i instal·lar Maven:

```
1 sudo apt update
2 sudo apt install maven
```

Per a comprovar que tot ha anat correctament, podem veure la versió instal·lada de Maven:

```
mvn --v
```

1.3.1 Configuració de Maven

Per a poder realitzar desplegaments en el nostre Tomcat prèviament instal·lat, necessitem realitzar la configuració adequada per a Maven. Ja sabem que això en Linux significa editar els arxius de configuració adequats. Anem a això.

1. En primer lloc necessitem assegurar-nos que en l'apartat anterior de la pràctica hem afegit tots els usuaris necessaris, així com els seus respectius rols. Hem d'afegir el rol de `manager-script` per a permetre que Maven s'autentique contra Tomcat i pugui realitzar el desplegament.

Els rols utilitzats per Tomcat venen detallats en la seua [documentació](#), que mereix ser consultada:

You can find the role names in the `web.xml` file of the Manager web application. The available roles are:

- `manager-gui` — Access to the HTML Interface.
- `manager-status` — Access to the "Server Status" page only.
- `manager-script` — Access to the tools-friendly plain text interface that is described in this document, and to the "Server Status" page.
- `manager-jmx` — Access to JMX proxy interface and to the "Server Status" page.

En aquesta documentació se'ns indica que, per temes de seguretat, és recomanable no atorgar els rols de `manager-script` o `manager-jmx` al mateix usuari que tinga el rol de `manager-gui`.



Tindrem dos usuaris, un per a la GUI i un altre exclusivament per a fer els **deploys** de Maven.

Així les coses, modifiquem l'arxiu `/etc/tomcat9/tomcat-users.xml` concorde a les nostres necessitats (els noms d'usuari i contrasenya hauran de ser els que trieu per a vosaltres):

```
<role rolename="admin"/>
<role rolename="admin-gui"/>
<role rolename="manager"/>
<role rolename="manager-gui"/>
<role rolename="manager-script"/>

<user username="guillermo" password="*****" roles="admin,admin-gui,manager,manager-gui"/>
<user username="guillermo-deploy" password="*****" roles="manager-script"/>
</tomcat-users>
```

2. Editar l'arxiu `/etc/maven/settings.xml` per a indicar-li a Maven, un identificador per al servidor sobre el qual desplegarem (no és més que un nom, poseu-li el nom que considereu), així com les credencials. Tot això es farà dins del bloc **servers** del XML:

```
<server>
  <id>Tomcat.P.3.1</id>
  <username>guillermo-deploy</username>
  <password>*****</password>
</server>
</servers>
```

1.3.2 Desplegament amb Maven

Tenint ja tot llest per a realitzar desplegaments, ara crearem una aplicació Java de prova per a veure si podem desplegar-la sobre l'arquitectura que hem muntat. Per a això utilitzem el comando:

```
mvn archetype:generate -DgroupId=guillermo -DartifactId=war-deploy -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false
```

Podeu substituir els valors de **groupId** i **artifactId** (este serà el nom de l'aplicació) pel que vulgueu.

Ara hem de modificar el **POM** del projecte perquè faci referència al fet que el desplegament es realitzi amb el plugin de Maven per a Tomcat.



No existeixen plugins **oficials** per a Tomcat més enllà de la versió 7 del servidor. No obstant això, el plugin per a Tomcat 7 continua funcionant correctament amb Tomcat 9. Una altra opció seria utilitzar el plugin [Cargo](#).

```
<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>guillermo</groupId>
  <artifactId>war-deploy</artifactId>
  <packaging>war</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>war-deploy Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
    <finalName>war-deploy</finalName>
    <plugins>
      <plugin>
        <groupId>org.apache.tomcat.maven</groupId>
        <artifactId>tomcat7-maven-plugin</artifactId>
        <version>2.2</version>
        <configuration>
          <url>http://localhost:8080/manager/text</url>
          <server>Tomcat.P.3.1</server>
          <path>/myapp</path>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```


On el que afegim és el bloc

```
1 <build>
2   <finalName>war-deploy</finalName>
3   <plugins>
4     <plugin>
5       <groupId>org.apache.tomcat.maven</groupId>
6       <artifactId>tomcat7-maven-plugin</artifactId>
7       <version>2.2</version>
8       <configuration>
9         <url>http://localhost:8080/manager/text</url>
10        <server>Tomcat.P.3.1</server>
11        <path>/myapp</path>
12      </configuration>
13    </plugin>
14  </plugins>
15 </build>
```

- **finalName:** Nom final de l'executable `.jar` que es generarà.
- **url:** URL del servidor Tomcat on es farà el desplegament. Com en el nostre cas Maven i Tomcat estan en el mateix servidor, la URL correspon a localhost. Esta URL ha d'anar seguida per `/manager/text` tal com llegim en la [documentació del plugin](#).
- **server:** Nom del server on es desplegarà l'aplicació. El nom ha de ser consistent amb el que hàgem posat en el `settings.xml` del pas anterior.
- **path:** Nom que l'aplicació utilitzarà en el path de la URL.

Després de generar esta aplicació, els comandos finals que s'utilitzen en Maven per a desplegar, tornar a desplegar o desplegar una aplicació, són:

- `mvn tomcat7:deploy`
- `mvn tomcat7:redploy`
- `mvn tomcat7:undeploy`

Així doncs, després del desplegament amb Maven ens indicarà que tot ha anat correctament amb un missatge de **BUILD SUCCESS**, tal que així:

```
[INFO] tomcatManager status code:200, ReasonPhrase:
[INFO] OK - Desplegada aplicaci3n en trayectoria de contexto [/myapp]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 38.978 s
[INFO] Finished at: 2023-10-19T19:20:33+02:00
[INFO] -----
```

I, accedint a través de la GUI, hem de veure que l'aplicació està desplegada i que podem accedir a ella perfectament:

Gestor de Aplicaciones Web de Tomcat

Mensaje:

OK

Gestor

Listar Aplicaciones

Ayuda HTML de Gestor

Ayuda de Gestor

Estado de Servidor

Aplicaciones

Ruta	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado		true	0	<div>ArrancarPararRecargarReplegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div>
/host-manager	Ninguno especificado	Tomcat Host Manager Application	true	0	<div>ArrancarPararRecargarReplegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div>
/manager	Ninguno especificado	Tomcat Manager Application	true	1	<div>ArrancarPararRecargarReplegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div>
myapp	Ninguno especificado	Archetype Created Web Application	true	0	<div>ArrancarPararRecargarReplegar</div> <div>Expirar sesiones sin trabajar ≥ 30 minutos</div>

1.3.3 Tasca 2

Realitzar el desplegament amb l'aplicació de prova.

Repetir el desplegament però esta vegada amb una altra aplicació que no és la de prova. Usarem els comandos que veurem a continuació.

Ens clonem el repositori:

```
git clone https://github.com/cameronmcnz/rock-paper-scissors.git
```

Ens situem dins d'ell:

```
cd rock-paper-scissors
```

I canviem de branca:

```
git checkout patch-1
```

Després d'això hem de procedir exactament igual que en el cas anterior, amb l'avantatge que ja tenim configurats els usuaris de Tomcat i els paràmetres de Maven.

Així doncs, només caldria afegir el bloc `<plugin>...</plugin>` adequat per a poder fer el nostre desplegament.



Documenta, incloent-hi captures de pantalles, el procés que has seguit per a realitzar el desplegament d'esta nova aplicació, així com el resultat final.