

Tomcat e le servlet

(I lucidi con sfondo violetto non sono essenziali)

Avvio (dalla sottodirectory *bin*): `./startup.sh`

Accesso a Tomcat, dopo l'avvio
(configurazione di default):

<http://localhost:8080>

Per un tutorial di base (se si è già
installato Tomcat):

<http://localhost:8080/docs/appdev/>

Tomcat e le servlet

- Tomcat è un progetto Apache ed è un'applicazione Java
- Tomcat è un *Java application container*, cioè esegue applicazioni Java accessibili via Web: **Java Web Applications** o **Web App**
- Primariamente, il bytecode eseguibile come parte di una Web App è una classe detta **Servlet**
- Il codice eseguibile può essere scritto in altre forme (p.es. JSP), ma, alla fine, sarà tradotto in, ed eseguito come, **Servlet**
- Tomcat si può usare in modalità **stand-alone** (è a sua volta un' applicazione Java) oppure come **modulo** del web server **Apache**
- In generale, un *application server* è tale se supporta, oltre alle servlet, l'architettura (di applicazioni per il Web) **Java Enterprise**, di Oracle (un tempo Sun)
- Altri application container: Jetty, JBoss (Red Hat)
 - IIS (Microsoft) non dà (da molti anni) supporto nativo per Java

Tomcat: CATALINA_HOME

La directory di installazione di Tomcat, detta **CATALINA_HOME**, ha le subdirectory:

bin – script *startup.sh*/*shutdown.sh*, di configurazione e alcuni *jar*

lib – librerie (jar) e classi

conf – file di configurazione (tra i quali *server.xml* designa il *Connector port* per *http* come 8080 (default), o altro, p.es. 28080)

logs – log e output files, tra cui il file *catalina.out*

- all'avvio Tomcat apre (se c'è) o crea il file *catalina.out*, che è il supporto fisico del *PrintWriter System.out*, cioè la *standard output*, della JVM che esegue Tomcat

webapps – web application eseguite da Tomcat

work – directory di lavoro temporanee per le web application

- p.es., ogni pagina JSP viene tradotta (la prima volta) in una servlet memorizzata in *work/* e verrà eseguita attraverso questa servlet

NB: a volte, eliminare *work* risolve anomalie di comportamento di Tomcat

temp – la proprietà *java.io.tmpdir* (file temp.) della JVM che esegue Tomcat

NB: per la JVM standard, *java.io.tmpdir* è diversa, come si vede qui:

```
$ java -XshowSettings 2>&1 | grep tmpdir      # 2>&1 ridirige standard error su standard output per passarla alla pipe |
java.io.tmpdir = /tmp                         # questo vale per la JVM standard
```

Download e installazione “a mano” personale

Da <https://tomcat.apache.org/whichversion.html> si sceglie una versione **major** di Tomcat, p.es. la **8** (anche secondo la vers. di JDK in uso)
Si visiti quindi (si noti l’**“8”**): <https://tomcat.apache.org/download-80.cgi> o <http://downloads.apache.org/tomcat/tomcat8>, per scaricarne Tomcat in un archivio **.tar.gz** o **.zip**, p. es. **apache-tomcat-8.5.73.tar.gz** oppure **apache-tomcat-8.5.73.zip** (8.5.73 è in atto l’ultima versione)

Semplicemente decomprimendo questo archivio, un utente comune (non super-user) può installare Tomcat “a mano”, in una directory **CATALINA_HOME** sotto il proprio controllo, nella propria **home** **~**:

```
~ $ unzip ~/Downloads/apache-tomcat-8.5.73.zip      # decomprime l'archivio Tomcat scaricato
~ $ ls -l apache-tomcat-8.5.73                      # la radice dell'archivio zip è la dir apache-tomcat-8.5.73
LICENSE NOTICE RELEASE-NOTES RUNNING.txt bin conf logs lib temp work webapps
~ $ mv apache-tomcat-8.5.73 tomcat8                 # cambia nome alla CATALINA_HOME
```

Installazione personale “a mano”: caveat

- Installando “a mano” Tomcat, il Java/JDK che esso richiede potrebbe non essere quello di default del sistema e potrebbe dover essere installato in aggiunta a quello

- In effetti, che ciò accada è improbabile, almeno per Tomcat e JDK recenti (2021)

Si consideri la tabella a fianco, tratta da

<https://tomcat.apache.org/whichversion.html>

Servlet Spec	Tomcat Version	Latest Version	Java Versions
	5.0	10.0.x	10.0.14 8 and later
	4.0	9.0.x	9.0.56 8 and later
3.1	8.5.x	8.5.73	7 and later

Che il JDK sia 8, 11 o 17 andrà bene comunque!

- In ogni caso (e anche per Windows) se è definita la variabile d'ambiente JAVA_HOME e punta a un JDK installato, sarà questo a eseguire Tomcat (che, si ricordi, è un'applicazione Java)

- (anche usando il JDK di default) **conviene sempre definire JAVA_HOME**

```
$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64      # esempio valido per Debian/Ubuntu
```

- Il comando qui sopra, che definisce JAVA_HOME, si può inserire una volta per tutte nel file CATALINA_HOME/bin/setenv.sh, anziché ogni volta sulla shell

☞ Sotto Unix, i file .sh in CATALINA_HOME/bin devono essere eseguibili (o essere resi tali con chmod +x CATALINA_HOME *.sh); si potrà così invocare:

```
$ ~ /tomcat8/bin/startup.sh      # lancia Tomcat, con l'installazione “a mano” nella CATALINA_HOME vista prima
```

Avvio di Tomcat: test della configurazione

- Proseguendo l'esempio precedente, in cui si è installato Tomcat da zip nella **CATALINA_HOME** `~/tomcat8`, si rendono gli script eseguibili:

```
~ $ chmod +x ~/tomcat8/bin/*.sh
```

rende eseguibili (decomprimendo da zip non lo sono) gli script in `bin/`

- **e invocabili con il nome semplice (senza premettere `~/tomcat/bin`):**

```
~ $ export PATH=~/tomcat8/bin:$PATH # rende invocabili col nome semplice gli script (startup.sh ... ) in bin/
```

- **Prima di lanciare Tomcat, si verifichi la configurazione:**

```
~ $ configtest.sh |& egrep -e '^Using|^INFO|^\$SEVERE' | \
  egrep -i '^Using|^SEVERE| j.*home|nio|-D.*=/|Server v' # si filtra dal superfluo l'output di configtest.sh
Using CATALINA_BASE:          /home/gp/tomcat8
Using CATALINA_HOME:          /home/gp/tomcat8
Using CATALINA_TMPDIR:        /home/gp/tomcat8/temp
Using JRE_HOME:               /usr/lib/jvm/java-1.8.0-openjdk-amd64
Using CLASSPATH:              /home/gp/tomcat8/bin/bootstrap.jar:/home/gp/tomcat8/bin/tomcat-juli.jar
INFO: Server version:         Apache Tomcat/8.5.39
INFO: Java Home:              /usr/lib/jvm/java-8-openjdk-amd64/jre
INFO: Command line argument:  -Dcatalina.base=/home/gp/tomcat8
INFO: Command line argument:  -Dcatalina.home=/home/gp/tomcat8
INFO: Command line argument:  -Djava.io.tmpdir=/home/gp/tomcat8/temp
INFO: Initializing ProtocolHandler ["http-nio-8080"]
INFO: Initializing ProtocolHandler ["ajp-nio-8009"]
```

In caso di problemi, si vedrebbe anche un rigo di output marcato SEVERE:

```
SEVERE: Failed to ...
```

Avvio di Tomcat e verifiche

Effettuata (o no, se si ama il rischio ☺) mediante `configtest.sh` la verifica della configurazione, si può avviare (e successivamente arrestare) Tomcat:

```
~ $ startup.sh      # lanciare Tomcat, che ascolta richieste HTTP su http://localhost:8080
~ $ shutdown.sh     # arrestare Tomcat
```

L'avvio crea un processo **java** (JVM che esegue il bytecode di Tomcat):

```
~ $ pgrep -f tomcat  # cerca un processo che contiene la stringa "tomcat" nel comando che lo ha creato
20901                # e ne fa vedere il pid (process id)
~ $ ps 24478          # vediamo quale comando ha creato il processo 20901
 PID TTY STAT TIME COMMAND
20901 ? S1 0:13 /usr/lib/jvm/default-java/bin/java -D...
```

Si può verificare che Tomcat ascolta su *localhost:8080* con un browser o:

```
~ $ curl http://localhost:8080  # curl è un cliente HTTP da riga di comando
...
```

Per vedere i porti aperti da Tomcat (noto il *PID*): **lsof -i4 -a -p PID -Pn**
(-**i**: *IP socket*, -**i4**: *IPv4*, -**a**: *and*, -**P**: *port #, not names*, -**n**: *IP addr, not names*)

```
~ $ sudo lsof -i -a -p $(pgrep -f tomcat) -Pn  # sudo consente a lsof di vedere ogni socket/porta aperte
java    20901  gp  36u  IPv6 0x366d3073e280a2cf      0t0    TCP *:8080 (LISTEN)
java    20901  gp  40u  IPv6 0x366d3073e280a88f      0t0    TCP *:8009 (LISTEN)
java    20901  gp  81u  IPv6 0x366d3073e28074cf      0t0    TCP 127.0.0.1:8005 (LISTEN)
```