

Primi passi con Apache Tomcat



L'application server dell'Apache group

J2EE

- Direttive Sun Microsystem
- Che cos'è un application server
- EJB
- Maggiori application server open source

Tomcat

- Open Source e Free Software
- Disponibile per diverse piattaforme
- Si integra con un Web Server (Apache e IIS)
- Può essere utilizzato in maniera stand-alone

Installazione

- Decomprimere il pacchetto nella cartella in */usr/lib*
- Configurare le variabili di sistema (JAVA_HOME e TOMCAT_HOME)
- Avviare il servizio (`$TOMCAT_HOME/bin/startup.sh`)

Struttura

- **Bin** (contiene gli script per avviare, riavviare e arrestare Tomcat)
- **Conf** (sono contenuti i file di configurazione di tomcat, i più importanti sono *server.xml* e *web.xml*)
- **Server** (contiene i file di archivio Java che permettono il funzionamento di Tomcat)
- **Lib** (contiene i file jar di cui Tomcat fa uso)
- **Log** (sono presenti tutti i file di log per monitorare l'andamento del server)
- **Webapps** (Il cuore di Tomcat. In questa cartella ci sono tutti i file **war** dei quali abbiamo fatto il *deployed*)
- **Work** (sono contenute tutte le Servlet che sono state generate da una pagina JSP)

WEB-INF

- Contiene i file di configurazione e delle risorse necessarie all'applicazione
- Contiene proprie classi e librerie di tag
- È protetta e nascosta dal server

Struttura di WEB-INF

- **Classes** (contiene le classi Java che si intendono utilizzare)
- **Lib** (contiene gli archivi JAR che si intendono utilizzare)
- **Tlds** (contiene i descrittori TLD per l'estensione dei tag JSP)

web.xml

- Descrive le risorse e le informazioni aggiuntive necessarie al funzionamento dell'applicazione
- È contenuto nella directory WEB-INF
- Tutte le informazioni sono contenute dentro tag con questa forma:

```
<context-param>  
<param-name></param-name>  
<param-value></param-value>  
</context-param>
```


web.xml in pratica

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
  <!-- Qui vengono inserite tutte le info e le risorse per
        il funzionamento dell'applicazione -->

  <display-name>Example MyApp</display-name>
  <description>
    Example MyApp
  </description>

  <context-param>
    <param-name>DB.DRIVER</param-name>
    <param-value>com.mysql.jdbc.Driver</param-value>
  </context-param>

  <context-param>
    <param-name>DB.URL</param-name>
    <param-value>jdbc:mysql://localhost:3306/db_examples</param-value>
  </context-param>
</web-app>

```

Gestire gli Utenti

- Attraverso il file *tomcat-user.xml* contenuto nella directory *conf*
- Aggiungendo e/o modificando stringhe in questo modo:
 - ✓ `<role rolename= "tomcat" />`
 - ✓ `<user username= "tomcat" password= "tomcat" roles= "tomcat" />`
- Al primo utilizzo aggiungere un utente che abbia i permessi per amministrare e gestire l'application server
 - ✓ `<user username= "user" password= "password" roles= "admin,manager" />`

Il pacchetto Admin

- Serve per amministrare Tomcat con un interfaccia grafica, in particolare:
 - ✓ Gestisce gli utenti e i gruppi
 - ✓ Crea connessioni con i database
 - ✓ gestisce le variabili d'ambiente Tomcat
 - ✓ Crea nuovi host

Installare il pacchetto Admin

- Decomprimere il file nella cartella:
\$TOMCAT_HOME/server/webapps
- Spostare *admin/index.html* in *webapps/ROOT/*
- Per accedere alle funzionalità lanciare:
http://localhost:8080/admin/
- Utilizzare nome e utente “*tomcat*”

Admin in pratica

TOMCAT WEB SERVER ADMINISTRATION TOOL

Commit Changes

Log Out

- Tomcat Server
 - Service (Catalina)**
 - Resources
 - Data Sources
 - Mail Sessions
 - Environment Entries
 - User Databases
 - User Definition
 - Users
 - Groups
 - Roles

Service (Catalina)

Service Actions -----Available Actions-----

Save Reset

Service Properties

Property	Value
Name:	Catalina

Engine Properties

Property	Value
Name:	Catalina
Default Hostname:	localhost

Save Reset

Caricare le librerie

- Nella variabile d'ambiente CLASSPATH
- In `common/lib`
- `WEB-INF/lib` della nostra applicazione

Deploy di un applicazione

- Inserendo il **war** file nella cartella *webapps*
- Attraverso l'interfaccia di Tomcat "*tomcat manager*" in due modi:
 - ✓ Scegliendo la cartella dov'è contenuto il file **war**
 - ✓ Inserendo il file dentro la struttura di Tomcat e poi indicando il nome e il percorso dell'applicazione

Deploy in pratica

Deploy

Deploy directory or WAR file located on server

Context Path (optional):

XML Configuration file URL:

WAR or Directory URL:

Deploy

WAR file to deploy

Select WAR file to upload Sfoglia...

Deploy

JK Connector

- non e' altro che un pezzo di software che si installa in Apache (compilandolo staticamente o caricandolo come modulo a runtime) e che '*chiama*' Tomcat quando necessario passando la palla quando istruito
- L'alternativa e' usare `mod_proxy` e "*proxare*" le richieste verso un'applicazione verso Tomcat stesso

JK su Tomcat

- Su Tomcat 5.5 è già configurato di default
- Per le altre versioni bisogna creare e/o modificare il file *workers.properties* contenuto nella directory *conf*
- Il Workers e' il file che contiene la configurazione per i vari processi di comunicazione con tomcat

Workers in pratica

Una configurazione minimale e' la seguente:

```
# workers.properties -
workers.tomcat_home=/usr/local/tomcat
workers.java_home=/usr/lib/java
ps=/

worker.list=ajp13
worker.ajp13.port=8009
worker.ajp13.host=localhost
worker.ajp13.type=ajp13
```

JK su Apache 2

- Nel file di configurazione Apache (httpd.conf) bisogna:
 - ✓ Caricare il modulo del connettore (mod_jk.so)
 - ✓ JkWorkersFile
\$TOMCAT_HOME/conf/workers.properties
 - ✓ JkLogFile \$TOMCAT_HOME/logs/jk_log
 - ✓ JkLogLevel debug
 - ✓ JkMount /myapp/* ajp13

Materiale di supporto

- <http://www.olografix.org/acme/nism/tomcat.pdf>
- <http://www.olografix.org/acme/nism/myapp.war>

Linkgrafia

- <http://jakarta.apache.org/tomcat/>
- <http://java.sun.com/j2ee/index.jsp>
- <http://www.dazeroadieci.org>
- <http://jakarta.apache.org/tomcat/connectors-doc/>
- <http://www.javaportal.it>

Altri eventi

- JIP Day 30 settembre 2005 – Roma
- Java Software Distribution 2005 – fine ottobre (<http://212.103.211.4/jsd/>)

Ringraziamenti

- Comune di Francavilla
- PescaraLug, MetroOlografix, LaLiPe
- Javaportal (*per il supporto*)
- Il pubblico (*per averci sopportato*)

Primi passi con Apache Tomcat



L'application server dell'Apache group

Primi passi con Apache Tomcat



L'application server dell'Apache group

J2EE

- Direttive Sun Microsystem
- Che cos'è un application server
- EJB
- Maggiori application server open source

Le specifiche della SUN MicroSystem sull'ambiente Java Enterprise Edition (J2EE) sono specifiche aperte e riguardano le Java Server Page (JSP), le Servlet e gli Enterprise Java Bean (EJB). Tutti i sviluppatori che intendono realizzare un application server ne devono tenere conto.

Un application server è un tool che permette di eseguire su una macchina le specifiche del J2EE, naturalmente bisogna aver già installato il jdk (java development kit).

I maggiori Application server gratuiti sono jboss che permette anche l'uso degli EJB e Apache Tomcat.

Tomcat

- Open Source e Free Software
- Disponibile per diverse piattaforme
- Si integra con un Web Server (Apache e IIS)
- Può essere utilizzato in maniera stand-alone

Tomcat che è sviluppato dall'Apache Group permette sia di eseguire le JSP, i Servlet ed inoltre è possibile sfruttarlo in maniera stand-alone cioè è possibile usarlo come web server a tutti gli effetti.

Installazione

- Decomprimere il pacchetto nella cartella in */usr/lib*
- Configurare le variabili di sistema (JAVA_HOME e TOMCAT_HOME)
- Avviare il servizio (`$TOMCAT_HOME/bin/startup.sh`)

In ambiente Linux è meglio non installare l'applicativo come utente root ma bensì creare un utente con i diritti per avviare l'application server.

Di solito viene creato un utente dal nome tomcat.

Struttura

- **Bin** (contiene gli script per avviare, riavviare e arrestare Tomcat)
- **Conf** (sono contenuti i file di configurazione di tomcat, i più importanti sono *server.xml* e *web.xml*)
- **Server** (contiene i file di archivio Java che permettono il funzionamento di Tomcat)
- **Lib** (contiene i file jar di cui Tomcat fa uso)
- **Log** (sono presenti tutti i file di log per monitorare l'andamento del server)
- **Webapps** (Il cuore di Tomcat. In questa cartella ci sono tutti i file **war** dei quali abbiamo fatto il *deployed*)
- **Work** (sono contenute tutte le Servlet che sono state generate da una pagina JSP)

La struttura proposta può differire dipende dalla versione di Tomcat che state utilizzando.

Noi ci riferiamo alla 5.5.x

WEB-INF

- Contiene i file di configurazione e delle risorse necessarie all'applicazione
- Contiene proprie classi e librerie di tag
- È protetta e nascosta dal server

La cartella **WEB-INF** è una speciale cartella che viene protetta dal server e non permette in remoto di accedervi cioè non è possibile digitare un URL di questo tipo: *<http://www.miosito.ext/WEB-INF/miofile.ext>*

Inoltre ad ogni applicazione deve essere creata, anche se alcuni web-server permettono di lasciarla vuota.

Tomcat richiede che ci sia almeno il file **web.xml**

Struttura di WEB-INF

- ▶ **Classes** (contiene le classi Java che si intendono utilizzare)
- ▶ **Lib** (contiene gli archivi JAR che si intendono utilizzare)
- ▶ **Tlds** (contiene i descrittori TLD per l'estensione dei tag JSP)

Quelle proposte sono le maggiori directory contenute in un applicazione, **tlds** molto spesso viene ignorato perché non tutte le applicazioni vengono create propri tag.

web.xml

- Descrive le risorse e le informazioni aggiuntive necessarie al funzionamento dell'applicazione
- È contenuto nella directory WEB-INF
- Tutte le informazioni sono contenute dentro tag con questa forma:

```
<context-param>  
<param-name></param-name>  
<param-value></param-value>  
</context-param>
```

Nel file web XML è possibile creare dei “parametri” che contengono i “valori” per connettersi ad un database.

web.xml in pratica

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
  <!-- Qui vengono inserite tutte le info e le risorse per
        il funzionamento dell'applicazione -->

  <display-name>Example MyApp</display-name>
  <description>
    Example MyApp
  </description>

  <context-param>
    <param-name>DB.DRIVER</param-name>
    <param-value>com.mysql.jdbc.Driver</param-value>
  </context-param>

  <context-param>
    <param-name>DB.URL</param-name>
    <param-value>jdbc:mysql://localhost:3306/db_examples</param-value>
  </context-param>
</web-app>
```

In questo esempio è riportato i parametri che si dicevano prima. Qui viene utilizzato il database MySQL. Per utilizzare questi parametri bisogna inserire in una pagina JSP le seguenti istruzioni:

Gestire gli Utenti

- Attraverso il file `tomcat-user.xml` contenuto nella directory `conf`
- Aggiungendo e/o modificando stringhe in questo modo:
 - ✓ `<role rolename="tomcat" />`
 - ✓ `<user username="tomcat" password="tomcat" roles="tomcat" />`
- Al primo utilizzo aggiungere un utente che abbia i permessi per amministrare e gestire l'application server
 - ✓ `<user username="user" password="password" roles="admin,manager" />`

Questo file è presente in tutte le versioni di Apache anche se l'uso viene consigliato solo su Linux perché sul S.O. Ms Windows viene creato un utente in fase d'installazione dal tool dell'installazione stesso

Il pacchetto Admin

- Serve per amministrare Tomcat con un interfaccia grafica, in particolare:
 - ✓ Gestisce gli utenti e i gruppi
 - ✓ Crea connessioni con i database
 - ✓ gestisce le variabili d'ambiente Tomcat
 - ✓ Crea nuovi host

Questo pacchetto che bisogna prelevarlo a parte è molto utile perché permette di gestire tomcat attraverso una GUI senza andare a modificare a mano i file XML.

Anche se è sconsigliato ai principianti di usarlo per permettere loro di imparare i fondamenti di Tomcat.

Installare il pacchetto Admin

- Decomprimere il file nella cartella:
`$TOMCAT_HOME/server/webapps`
- Spostare `admin/index.html` in `webapps/ROOT/`
- Per accedere alle funzionalità lanciare:
`http://localhost:8080/admin/`
- Utilizzare nome e utente “*tomcat*”

L'installazione in ambiente Linux è molto più semplice perché bisogna decomprimere il pacchetto dov'è stato installato Tomcat.

Ad esempio se abbiamo installato Tomcat in `usr/lib/jakarta-tomcat/` installeremo **admin** in `usr/lib`, lui si installerà nel percorso giusto.

Admin in pratica

TOMCAT WEB SERVER ADMINISTRATION TOOL Commit Changes Log Out

- Tomcat Server
 - Service (Catalina)**
 - Resources
 - Data Sources
 - Mail Sessions
 - Environment Entries
 - User Databases
 - User Definition
 - Users
 - Groups
 - Roles

Service (Catalina) Service Actions -----Available Actions-----

Save Reset

Service Properties

Property	Value
Name:	Catalina

Engine Properties

Property	Value
Name:	Catalina
Default Hostname:	localhost

Save Reset

Questa è una schermata del pacchetto admin, in questo caso ci troviamo a visualizzare i servizi di Tomcat...

Caricare le librerie

- Nella variabile d'ambiente CLASSPATH
- In `common/lib`
- `WEB-INF/lib` della nostra applicazione

Normalmente il CLASSPATH è settato su `%JAVA_HOME%/jre/lib`.

Se si vogliono sviluppare le Servlet bisogna copiare la libreria **servlet-api.jar** contenuta nella cartella `%TOMCAT_HOME%/common/lib` nel `%JAVA_HOME%/jre/lib/ext` oppure inserire il percorso della libreria nel CLASSPATH.

Deploy di un applicazione

- Inserendo il **war** file nella cartella *webapps*
- Attraverso l'interfaccia di Tomcat "*tomcat manager*" in due modi:
 - ✓ Scegliendo la cartella dov'è contenuto il file **war**
 - ✓ Inserendo il file dentro la struttura di Tomcat e poi indicando il nome e il percorso dell'applicazione

Penso sia chiaro, non necessità di ulteriori commenti

  

Deploy in pratica

Deploy

Deploy directory or WAR file located on server

Context Path (optional):

XML Configuration file URL:

WAR or Directory URL:

WAR file to deploy

Select WAR file to upload

16 luccio@daxeroadieci.org
acme@olografix.org

Gli ultimi 2 punti della slide precedente sono mostrati in questa figura...

JK Connector

- non e' altro che un pezzo di software che si installa in Apache (compilandolo staticamente o caricandolo come modulo a runtime) e che '*chiama*' Tomcat quando necessario passando la palla quando istruito
- L'alternativa e' usare `mod_proxy` e "*proxare*" le richieste verso un'applicazione verso Tomcat stesso

La versione con il proxy non è stata analizzata, ma solo suggerita perché è una soluzione molto "sporca". Quindi ne sconsigliamo l'uso.

JK su Tomcat

- Su Tomcat 5.5 è già configurato di default
- Per le altre versioni bisogna creare e/o modificare il file *workers.properties* contenuto nella directory *conf*
- Il Workers e' il file che contiene la configurazione per i vari processi di comunicazione con tomcat

Sul sito ufficiale del JK connector è presente una guida molto esaustiva su come configurare il file *workers.properties*. Il link è presente nella slide 22

Workers in pratica

Una configurazione minimale e' la seguente:

```
# workers.properties -
workers.tomcat_home=/usr/local/tomcat
workers.java_home=/usr/lib/java
ps=/

worker.list=ajp13
worker.ajp13.port=8009
worker.ajp13.host=localhost
worker.ajp13.type=ajp13
```

JK su Apache 2

➤ Nel file di configurazione Apache (httpd.conf) bisogna:

- ✓ Caricare il modulo del connettore (mod_jk. so)
- ✓ JkWorkersFile
\$TOMCAT_HOME/conf/workers.properties
- ✓ JkLogFile \$TOMCAT_HOME/logs/jk_log
- ✓ JkLogLevel debug
- ✓ JkMount /myapp/* ajp13

Materiale di supporto

- <http://www.olografix.org/acme/nism/tomcat.pdf>
- <http://www.olografix.org/acme/nism/myapp.war>

Linkgrafia

- <http://jakarta.apache.org/tomcat/>
- <http://java.sun.com/j2ee/index.jsp>
- <http://www.dazeroadieci.org>
- <http://jakarta.apache.org/tomcat/connectors-doc/>
- <http://www.javaportal.it>

Altri eventi

- JIP Day 30 settembre 2005 – Roma
- Java Software Distribution 2005 – fine ottobre (<http://212.103.211.4/jsd/>)

Ringraziamenti

- Comune di Francavilla
- PescaraLug, MetroOlografix, LaLiPe
- Javaportal (*per il supporto*)
- Il pubblico (*per averci sopportato*)

Primi passi con Apache Tomcat



L'application server dell'Apache group