

Configuración y administración del servidor de aplicaciones TOMCAT


Server.xml y context.xml

Principales ficheros de configuración

server.xml - Para establecer **servicios, comportamientos** del servidor y **recursos globales** (ej. base de datos común). Ubicado en /conf

context.xml - Descriptor de contexto para una o múltiples aplicaciones. Sirve para **definir recursos compartidos**. Ubicado en /META-INF de la aplicación o en la configuración de un host específico.

web.xml - **Descriptor de despliegue** de una aplicación o servlet. Para identificar y configurar servlets en Tomcat. Ubicado en /WEB-INF dentro de la parte privada del servlet.




Server.xml

En el archivo de configuración por defecto podemos ver que se establece un **único servicio y una única instancia del servidor**. El tag <Server> tiene la siguiente forma:

```
<Server port="8005" shutdown="SHUTDOWN">
```

Pese a que **el puerto de Tomcat es 8080**, esta etiqueta está configurada con el puerto **8005**.

Este puerto es en el que se inicia una instancia del servidor (JVM, Java machine) y en el que **escucha por si llegan señales de apagado** (shutdown). Esta señal no se puede mandar desde otra máquina por motivos de seguridad, pero evidentemente se puede ejecutar el comando de apagado desde otra máquina y la señal llegará desde la misma en la que está el servidor.



Server.xml | <Server>

Dentro de la etiqueta Server se puede definir:

- **<Service>**: Un grupo de conectores asociados con un motor. Es necesario al menos uno.
- **<Listener>**: Clases que tienen que escuchar y manejar eventos que tienen que ver con el ciclo de vida del servidor, por ejemplo después de arrancar, etc.
- **<GlobalNamingResources>**: Recursos globales que pueden ser usados en esta instancia del servidor por los componentes que los necesiten, por ejemplo una *base de datos*.



Server.xml | <Server> | <Service>

El propósito de la etiqueta de “servicio” es agrupar un **motor que procese las peticiones** con uno o más **conectores que gestionen los protocolos de comunicación**. El servicio por defecto es el motor **Catalina**.

```
<Service name="Catalina">
```

Aquí se le ha dado al servicio el nombre del motor, pero no es necesario.

Un **<Service>** **contiene** al menos un **<Connector>** y solo un **<Engine>**.




Server.xml | <Server> | <Service> | <Connector>

Este elemento “conector” tiene mucho que ver con los dos modos de funcionamiento de Tomcat:

- **Como servidor único:** En el que Tomcat realiza las funciones de servidor de aplicaciones y de servidor web.
- **Como servidor de aplicaciones** detrás de un servidor web: En el que Tomcat colabora con un servidor web que hace de frontend. El servidor web redirige todas peticiones de JSPs y Servlets a Tomcat.

En un entorno con acceso público se suele usar la segunda configuración ya que el servidor web está mucho más preparado en términos de seguridad y privacidad.



Server.xml | <Server> | <Service> | <Connector>

Los dos conectores más comunes son **HTTP** y **AJP**. El segundo es usado para conectar con servidores web en el modo (con Apache u otros).

Ambos pueden funcionar con **SSL** para mejorar la seguridad.

El puerto por defecto para HTTP es 8080.

Se puede cambiar; si Tomcat va a estar en producción como un servidor en solitario, se podría modificar al puerto estándar de HTTP, 80.



Server.xml | <Server> | <Service> | <Connector>

El conector HTTP está habilitado por defecto:

```
<Connector port="8080" protocol="HTTP/1.1"  
    connectionTimeout="20000"  
    URIEncoding="UTF-8"  
    redirectPort="8443" />
```

mientras que el AJP no:

```
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

Ninguno de los dos usa SSL por defecto.



Server.xml | <Server> | <Service> | <Engine>

El motor es quien procesa las peticiones. El nombre es el que le demos a la instancia y *defaultHost* indica a qué host virtual se le pasará una petición en caso de que no se especifique ninguno, ya que el mismo motor puede procesar peticiones dirigidas a múltiples hosts virtuales que se especifiquen en este archivo de configuración.

```
<Engine name="Catalina" defaultHost="localhost">
```

Un motor contiene uno o más **<Host>**, uno o ningún **<Context>**, uno o ningún **<Realm>** y múltiples, uno o ningún **<Valve>** y **<Listener>**.



Server.xml | <Server> | <Service> | <Engine> | <Realm>

El tag de “reino” es un **mecanismo de seguridad** que sirve para **autenticar usuarios y establecer seguridad** a nivel de servicio, contenedor o aplicación.

La configuración por defecto hace que Tomcat lea los usuarios del archivo *tomcatusers.xml* pero, en un entorno laboral, sería mejor configurarlo para que se usará una base de datos o servidor LDAP.



Server.xml | <Server> | <Service> | <Engine> | <Realm>

Por defecto:

<!-- Use the LockOutRealm to prevent attempts to guess user passwords via a brute-force attack -->

*<**Realm** className="org.apache.catalina.realm.LockOutRealm">*

<!-- This Realm uses the UserDatabase configured in the global JNDI resources under the key "UserDatabase". Any edits that are performed against this UserDatabase are immediately available for use by the Realm. -->

<Realm className="org.apache.catalina.realm.UserDatabaseRealm"

resourceName="UserDatabase"/>

*</**Realm**>*




Server.xml | <Server> | <Service> | <Engine> | <Host>

Este elemento **representa un host virtual** en Tomcat.

La configuración por defecto sólo define localhost. Si tenemos configurado el servidor DNS con un nombre para nuestro servidor, usaremos éste.

El atributo ***appBase*** establece el **directorio raíz de las aplicaciones**. Se establece a partir de <CATALINA_BASE> si no se indica lo contrario. Por defecto la URL de cada aplicación es la resultante de añadir su directorio raíz a la del servidor. Por defecto se han instalado cuatro aplicaciones: docs, examples, host-manager y manager. También se establece ROOT que indica la aplicación por defecto si no añadimos otras, cada una en su directorio.



Server.xml | <Server> | <Service> | <Engine> | <Host>

El atributo ***unpackWARs*** indica si este tipo de archivos debe ser descomprimido o no. En caso de no descomprimirlos, la ejecución será un poco más lenta.

autoDeploy indica si el despliegue de una aplicación que situemos en el directorio debe ser automático o no.

```
<Host name="localhost" appBase="webapps" unpackWARs="true"  
autoDeploy="true">
```



Server.xml | <Server> | <Service> | <Engine> | <Host>

Además, el host por defecto contiene un **<Valve>** para registrar los accesos.

```
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
prefix="localhost_access_log." suffix=".txt" pattern="%h %l %u %t &quot;%r&quot; %s
%b" />
```



Context.xml

El contexto en Tomcat se puede establecer a muchos niveles y se aplicará el más específico en cada aplicación (aquel que se defina a nivel más bajo sobrescribe los superiores).

Los **descriptores de contexto** para la agrupación y administración de cada aplicación se encuentran en \$CATALINA_BASE/conf/nombre_motor/nombre_host

En el caso por defecto están en **/etc/tomcat/Catalina/localhost**



Context.xml

Por ejemplo el de host-manager sería:

```
<Context path="/host-manager"  
docBase="/usr/share/tomcat10-admin/host-manager" antiResourceLocking="false"  
privileged="true" />
```

El **path** indica cómo se accederá a la aplicación en nuestro servidor. El problema para cambiarlo es que habría que modificar todos los archivos .xml relacionados para que Tomcat siga encontrando la aplicación.

docBase es el directorio de despliegue de la aplicación.



Context.xml

Los **descriptores de contexto específicos de cada aplicación web** están en el directorio de cada aplicación en /META-INF

El de la misma aplicación está en: /usr/share/tomcat10-admin/**host-manager/META-INF**

```
<Context antiResourceLocking="false" privileged="true" >
```

```
<!--Remove the comment markers from around the Valve below to limit access to the host-manager application to clients connecting from localhost -->
```

```
<!--<Valve className="org.apache.catalina.valves.RemoteAddrValve"  
allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->
```

```
</Context>
```



Context.xml

Entonces ¿para qué sirve context.xml? **contiene los parámetros que vayan a ser comunes a todas las aplicaciones.** Si no se sobrescriben en algún contexto más concreto se aplicarán éstos.

```
<Context>
```

```
<WatchedResource>WEB-INF/web.xml</WatchedResource>
```

```
</Context>
```

En este caso se especifica qué elemento de cada aplicación debe ser vigilado por si cambia. Si cambian, entonces se vuelve a desplegar la aplicación automáticamente.

