

Version : **2023.01.**

Dernière mise-à-jour : 2023/12/05 09:11

SER303 - Configuration du serveur Tomcat

Contenu du Module

- **SER303 - Configuration du serveur Tomcat**

- Contenu du Module
- Architecture du Serveur
- Fichiers de Configuration
 - Le Fichier `$CATALINA_HOME/conf/server.xml`
 - L'élément `<Server>`
 - L'élément `<Service>`
 - L'élément `<Connector>`
 - L'élément `<Executor>`
 - L'élément `<Engine>`
 - L'élément `<Host>`
 - L'élément `<Context>`
 - L'élément `<Realm>`
 - L'élément `<Loader>`
 - L'élément `<Manager>`
 - L'élément `<Store>`
 - L'élément `<Valve>`
 - Filtrage de l'adresse IP
 - Filtrage de nom de la machine du client
 - LAB #1 -Journalisation des Requêtes Client dans un Fichier Texte
 - LAB #2 -Journalisation des Requêtes Client dans une Base de Données
 - L'élément `<Listener>`
 - Le Fichier `$CATALINA_HOME/conf/web.xml`

- Le Fichier \$CATALINA_HOME/conf/tomcat-users.xml
- Le Fichier \$CATALINA_HOME/conf/catalina.policy
- Configuration des Ressources
 - Portée des Ressources
 - Pools de Connexion
 - Sessions JavaMail
 - JavaBeans
 - Entrées D'Environnement

Architecture du Serveur

Le répertoire d'installation de Tomcat est défini par la valeur de la variable \$CATALINA_HOME :

```
[root@centos7 ~]# echo $CATALINA_HOME
/usr/tomcat8
```

Avant de continuer, installez l'utilitaire **tree** :

```
[root@centos7 tomcat8]# yum install tree
```

L'arborescence du répertoire \$CATALINA_HOME est :

```
[root@centos7 ~]# cd $CATALINA_HOME
[root@centos7 tomcat8]# ls
bin  conf  lib  LICENSE  logs  NOTICE  RELEASE-NOTES  RUNNING.txt  temp  webapps  work
[root@centos7 tomcat8]# tree | more
.
├── bin
│   ├── bootstrap.jar
│   ├── catalina.bat
│   ├── catalina.sh
│   └── catalina-tasks.xml
```

- ├── commons-daemon.jar
- ├── commons-daemon-native.tar.gz
- ├── configtest.bat
- ├── configtest.sh
- ├── daemon.sh
- ├── digest.bat
- ├── digest.sh
- ├── setclasspath.bat
- ├── setclasspath.sh
- ├── shutdown.bat
- ├── shutdown.sh
- ├── startup.bat
- ├── startup.sh
- ├── tomcat-juli.jar
- ├── tomcat-native.tar.gz
- ├── tool-wrapper.bat
- ├── tool-wrapper.sh
- ├── version.bat
- └── version.sh

- ├── conf

- ├── Catalina
 - └── localhost
- ├── catalina.policy
- ├── catalina.properties
- ├── context.xml
- ├── logging.properties
- ├── server.xml
- ├── tomcat-users.xml
- ├── tomcat-users.xsd
- └── web.xml

- ├── lib

- ├── annotations-api.jar
- ├── catalina-ant.jar
- └── catalina-ha.jar

--More--

Le répertoire **bin** contient des scripts et fichiers nécessaires au démarrage du serveur.

```
[root@centos7 tomcat8]# cd bin
[root@centos7 bin]# tree
.
├── bootstrap.jar
├── catalina.bat
├── catalina.sh
├── catalina-tasks.xml
├── commons-daemon.jar
├── commons-daemon-native.tar.gz
├── configtest.bat
├── configtest.sh
├── daemon.sh
├── digest.bat
├── digest.sh
├── setclasspath.bat
├── setclasspath.sh
├── shutdown.bat
├── shutdown.sh
├── startup.bat
├── startup.sh
├── tomcat-juli.jar
├── tomcat-native.tar.gz
├── tool-wrapper.bat
├── tool-wrapper.sh
├── version.bat
└── version.sh

0 directories, 23 files
```

Le répertoire **conf** contient les quatre fichiers de configuration importants :

- **server.xml**,
- **tomcat-users.xml**,
- **web.xml**,
- **catalina.policy**.

```
[root@centos7 bin]# cd ../conf
[root@centos7 conf]# tree
```

```
.
├── Catalina
│   └── localhost
├── catalina.policy
├── catalina.properties
├── context.xml
├── logging.properties
├── server.xml
├── tomcat-users.xml
├── tomcat-users.xsd
└── web.xml
```

2 directories, 8 files

Le répertoire **libs** contient les bibliothèques Java (fichiers .jar) disponibles à toutes les applications ainsi qu'à Tomcat lui-même :

```
[root@centos7 conf]# cd ../lib/
[root@centos7 lib]# tree
```

```
.
├── annotations-api.jar
├── catalina-ant.jar
├── catalina-ha.jar
├── catalina.jar
├── catalina-storeconfig.jar
├── catalina-tribes.jar
├── ecj-4.5.jar
└── el-api.jar
```

```
├── jasper-el.jar
├── jasper.jar
├── jsp-api.jar
├── servlet-api.jar
├── tomcat-api.jar
├── tomcat-coyote.jar
├── tomcat-dbcp.jar
├── tomcat-il8n-es.jar
├── tomcat-il8n-fr.jar
├── tomcat-il8n-ja.jar
├── tomcat-jdbc.jar
├── tomcat-jni.jar
├── tomcat-util.jar
├── tomcat-util-scan.jar
├── tomcat-websocket.jar
└── websocket-api.jar
```

0 directories, 24 files

Le répertoire **logs** contient les fichiers de journalisation du serveur :

```
[root@centos7 lib]# cd ../logs
[root@centos7 logs]# tree
.
├── catalina.2017-04-27.log
├── catalina.out
├── host-manager.2017-04-27.log
├── localhost.2017-04-27.log
├── localhost_access_log.2017-04-27.txt
└── manager.2017-04-27.log
```

0 directories, 6 files

Le répertoire **webapps** contient les deux applications pour la gestion du serveur :

```
[root@centos7 logs]# cd ../webapps/  
[root@centos7 webapps]# tree | more
```

```
.  
├── docs  
│   ├── aio.html  
│   ├── api  
│   │   └── index.html  
│   ├── appdev  
│   │   ├── build.xml.txt  
│   │   ├── deployment.html  
│   │   ├── index.html  
│   │   ├── installation.html  
│   │   ├── introduction.html  
│   │   ├── processes.html  
│   │   ├── sample  
│   │   │   ├── build.xml  
│   │   │   ├── docs  
│   │   │   │   └── README.txt  
│   │   │   ├── index.html  
│   │   │   ├── sample.war  
│   │   │   ├── src  
│   │   │   │   └── mypackage  
│   │   │   │       └── Hello.java  
│   │   │   └── web  
│   │   │       ├── hello.jsp  
│   │   │       ├── images  
│   │   │       │   └── tomcat.gif  
│   │   │       ├── index.html  
│   │   │       ├── WEB-INF  
│   │   │       │   └── web.xml  
│   │   └── source.html  
│   └── web.xml.txt  
├── apr.html  
└── architecture
```

```
| | | | index.html  
| | | | overview.html  
| | | | requestProcess  
| | | |   ├── authentication-process.png  
| | | |   └── request-process.png  
| | | | requestProcess.html  
| | | | startup  
| | | |   ├── serverStartup.pdf
```

--More--

Le répertoire **temp** est utilisé comme répertoire temporaire par les applications :

```
[root@centos7 webapps]# cd ../temp/
[root@centos7 temp]# tree
.
└─ safeToDelete.tmp

0 directories, 1 file
```

Le répertoire **work** contient de sous-répertoires pour chaque application utilisés pour la transformation des pages JSP en classes Java :

```
[root@centos7 temp]# cd ../work
[root@centos7 work]# tree
```

```

├── Catalina
│   ├── localhost
│   │   ├── docs
│   │   ├── examples
│   │   ├── host-manager
│   │   ├── manager
│   │   └── ROOT
│   │       ├── org
│   │       │   └── apache
│   │       │       └── jsp

```

```
├─ index_jsp.class
├─ index_jsp.java
```

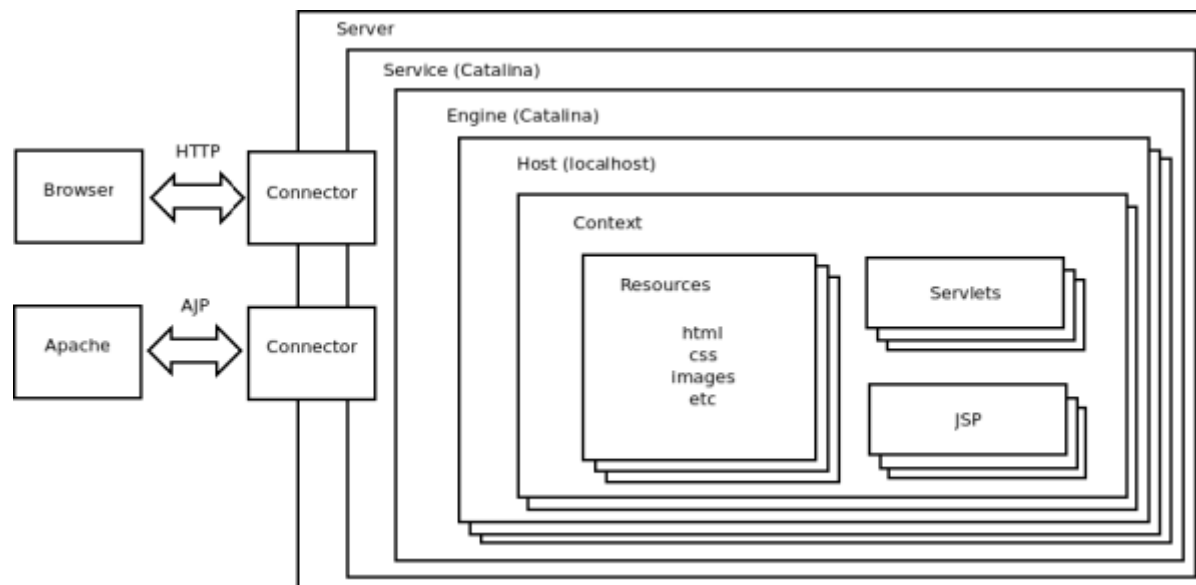
10 directories, 2 files

Fichiers de Configuration

Les composants de Tomcat sont appelés des **conteneurs**. Il en existe 5 :

- Server,
- Service,
- Engine,
- Host,
- Context.

L'organisation est imbriquée comme montre l'exemple suivant :



Cette organisation est détaillée dans le principal fichier de configuration de Tomcat : **\$CATALINA_HOME/conf/server.xml**.

Le Fichier \$CATALINA_HOME/conf/server.xml

A l'installation de Tomcat, ce fichier comporte les directives actives suivantes :

```
<?xml version='1.0' encoding='utf-8'?>
<Server port="8005" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
  <GlobalNamingResources>
    <Resource name="UserDatabase" auth="Container"
      type="org.apache.catalina.UserDatabase"
      description="User database that can be updated and saved"
      factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
      pathname="conf/tomcat-users.xml" />
  </GlobalNamingResources>
  <Service name="Catalina">
    <Connector port="8080" protocol="HTTP/1.1"
      connectionTimeout="20000"
      redirectPort="8443" />
    <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
    <Engine name="Catalina" defaultHost="localhost">
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
          resourceName="UserDatabase"/>
      </Realm>
      <Host name="localhost" appBase="webapps"
        unpackWARs="true" autoDeploy="true">
        <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" />

    </Host>
</Engine>
</Service>
</Server>
```

Dans ce fichier, les éléments **obligatoires** sont démontrés par l'exemple suivant :

```
<Server ...>
  <Service ...>
    <Connector ...>
    </Connector ...>
    <Connector ...>
    </Connector ...>
    etc
  <Engine ...>
    <Host ...>
      <Context ...>
      </Context ...>
      <Context ...>
      </Context ...>
      <Context ...>
      etc
    </Host ...>
    etc
  </Engine ...>
</Service ...>
<Service ...>
  <Connector ...>
  </Connector ...>
  <Connector ...>
  </Connector ...>
```

```
        etc
    <Engine ...>
        <Host ...>
            <Context ...>
            </Context ...>
            <Context ...>
            </Context ...>
            <Context ...>
            etc
        </Host ...>
    </Engine ...>
</Service ...>
etc
</Server ...>
```

D'autres éléments peuvent aussi être employés. L'exemple suivant démontre leur positionnement dans l'organisation du fichier :

```
<Server ...>
    <Listener>
    </Listener>
    <GlobalNamingResources ...>
        <Environment ...>
        </Environment ...>
        <Resource ...>
        </Resource ...>
    </GlobalNamingResources...>
    <Service ...>
        <Executor...>
        </Executor>
    <Engine ...>
        <Logger ...>
        </Logger ...>
        <Realm ...>
```

```
</Realm ...>
<Valve ...>
</Valve ...>
<Listener ...>
</Listener ...>
<Host ...>
  <Alias>
  </Alias>
  <Logger ...>
  </Logger ...>
  <Realm ...>
  </Realm ...>
  <Valve ...>
  </Valve ...>
  <Listener ...>
  </Listener ...>
  <Context ...>
    <Logger ...>
    </Logger ...>
    <Realm ...>
    </Realm ...>
    <Valve ...>
    </Valve ...>
    <Loader ...>
    </Loader ...>
    <Manager ...>
      <Store ...>
      </Store ...>
    </Manager ...>
    <Resource ...>
    </Resource ...>
    <ResourceLink ...>
    </ResourceLink ...>
  </Context ...>
```

```
    </Host ...>
  </Engine ...>
</Service ...>
</Server ...>
```

L'élément <Server>

Cet élément est la racine du fichier server.xml et comporte deux **attributs** :

- **port** - le port que Tomcat écoute pour une commande d'arrêt du serveur. Sa valeur par défaut est de **8005**,
- **shutdown** - la chaîne de caractères à envoyer au **port** afin que le serveur s'arrête. Sa valeur par défaut est **SHUTDOWN**.

<note warning> **Attention** - Afin d'éviter à ce qu'une personne malintentionnée puisse se connecter au port par défaut en utilisant telnet et envoyer la chaîne de caractères par défaut, il est impératif de modifier ses deux valeurs. </note>

L'élément <Service>

Cet élément permet d'associer des <Connector> et des <Executor> à un <Engine>. Le nom de chaque <Service> déclaré doit être unique.

L'attribut obligatoire est **name** :

Attribut	Description	Valeur par défaut	Valeur recommandée
name	Spécifie le nom du service	Catalina	A Modifier

L'élément <Connector>

Cet élément permet de mettre en place le transport de requêtes clients vers le moteur de servlet défini par l'élément <Engine> dans le même élément <Service>.

Il existe deux types selon le protocole utilisé :

- **HTTP** - le protocole HTTP/1.1 sur le port **8080**,
- **AJP** - le protocole AJP/1.3 sur le port **8009**.

Les attributs les plus importants de cet élément sont :

Attribut	Description	Valeur par défaut	Valeur recommandée
port	Spécifie le port d'écoute	8080/8009	8080/8009
address	Spécifie une adresse IP d'écoute	-	-
secure	Spécifie l'utilisation de HTTPS	false	-
redirectport	Spécifie le port vers lequel les requêtes HTTPS doivent être redirigées. L'attribut se trouve dans un Connector HTTPS.	-	-
disableUploadTimeout	Spécifie le temps maximum pour l'envoi de fichiers	false	-
enableLookups	Active la résolution des adresses IP en nom d'hôtes	true	false
proxyName	Spécifie le nom du proxy éventuel derrière lequel se trouve le serveur Tomcat	-	-
proxyPort	Spécifie le numéro du port du proxy éventuel derrière lequel se trouve le serveur Tomcat	-	-
maxThreads	Spécifie le nombre maximum de threads qui peut être créés dans un pool	200	200
acceptCount	Spécifie le nombre de requêtes qui sont mises en attente une fois la valeur de maxThreads ait été dépassée.	100	100
connectionTimeout	Spécifie le nombre de millisecondes qu'une requête puisse rester dans la file d'attente	60000 ms	60000 ms
executor	Spécifie le nom de l'élément <Executor> dont le <Connector> doit utiliser le pool	-	-

L'élément <Executor>

Pour traiter les requêtes entrantes, les <Connectors> utilisent des **threads** qui sont des sous-processus dans la machine virtuelle Java. Chaque thread est dédié à une connexion. Quand la connexion est terminée, le thread peut être ré-utilisé. Tomcat utilise un **pool** de threads pour éviter la création et la destruction inutiles de threads. Si les threads sont inactifs pour une période trop importante, le pool détruit les threads.

Les attributs les plus importants de cet élément sont :

Attribut	Description	Valeur par défaut	Valeur recommandée
----------	-------------	-------------------	--------------------

Attribut	Description	Valeur par défaut	Valeur recommandée
maxThreads	Spécifie le nombre maximum de threads qui peut être créés dans un pool	200	200
maxSpareThreads	Spécifie le nombre maximum de threads inactifs qui peut être dans un pool	50	50
minSpareThreads	Spécifie le nombre minimum de threads inactifs qui peut être dans un pool	25	25
maxIdleTime	Spécifie le nombre de millisecondes du délai d'inactivité d'un thread avant sa destruction	60000 ms	60000 ms
namePrefix	Spécifie un préfixe de nommage pour chaque thread du pool	-	-

L'élément <Engine>

Cet élément permet de mettre en place le transport de requêtes clients reçues des <Connector> vers les applications concernées.

Les attributs les plus importants de cet élément sont :

Attribut	Description	Valeur par défaut	Valeur recommandée
name	Spécifie le nom du moteur	Catalina	Catalina
defaultHost	Spécifie le nom d'hôte par défaut spécifiés par les éléments <HOST>	-	-
jvmRoute	Spécifie les informations de l' affinité de session utilisées lors du clustering	-	-

L'élément <Host>

Cet élément permet de mettre en place un hôte unique, virtuel ou réel, dans le serveur.

Les attributs les plus importants de cet élément sont :

Attribut	Description	Valeur par défaut	Valeur recommandée
name	Spécifie le nom d'hôte	localhost	-
appBase	Spécifie le répertoire de stockage par défaut des applications	\$CATALINA_HOME/webapps	\$CATALINA_HOME/webapps
autoDeploy	Spécifie si les applications déposées dans \$CATALINA_HOME/webapps sont à déployer automatiquement	true	false
deployOnStartup	Spécifie si les applications sont déployées automatiquement lors du démarrage du serveur	true	true

Attribut	Description	Valeur par défaut	Valeur recommandée
unpackWars	Spécifie si les archives compressées au format .war doivent être décompressées automatiquement	true	true
deployXML	Spécifie que le déploiement des applications peut être autorisé à partir des fichiers de contexte	true	true
workDir	Spécifie le répertoire de travail de l'application	-	-

L'élément <Context>

L'élément <Context> décrit une application déployée dans le serveur. Il existe deux types de <Context> :

- Les Contexts déclarés par un élément <Context> ,
- Les Contexts déclarés automatiquement par le serveur.

Dans le premier des deux cas, la déclaration de l'élément peut se trouver soit dans :

- le fichier **server.xml**,
- le fichier de contexte XML de l'application concernée.

C'est ce deuxième cas qui est le plus souvent utilisé.

Les attributs les plus importants de cet élément sont :

Attribut	Description	Valeur par défaut	Valeur recommandée
docBase	Spécifie le chemin absolu ou relatif du répertoire des données ou du fichier .war de l'application	-	-
path	Spécifie le chemin de contexte de l'application	-	-
reloadable	Active une surveillance des répertoires /WEB-INF/lib et /WEB-INF/classes. Chaque modification apportée au contenu des deux répertoires est automatiquement prise en charge par le serveur	false	false
useNaming	Active un contexte de nommage JNDI permettant la recherche des ressources	true	true
swallowOutput	Spécifie que la sortie standard et la sortie d'erreur doivent être consignées dans le fichier journal de l'application	false	true
workDir	Spécifie le répertoire de travail de l'application en remplaçant la valeur de l'attribut dans l'élément <Host>	-	-

Attribut	Description	Valeur par défaut	Valeur recommandée
privileged	Autorise l'application d'utiliser les servlets du moteur Catalina (p.e. Manager)	false	false
cookies	Spécifie que les cookies sont utilisés pour implémenter les sessions HTTP	true	true
override	Spécifie que les valeurs des attributs du <Context> surchargent celles du <DefaultContext>	false	false

Le contexte par défaut <DefaultContext> se trouve dans le fichier **\$CATALINA_HOME/conf/context.xml** :

```
<?xml version='1.0' encoding='utf-8'?>
<Context>

    <!-- Default set of monitored resources. If one of these changes, the    -->
    <!-- web application will be reloaded.                                   -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
    <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>

    <!-- Uncomment this to disable session persistence across Tomcat restarts -->
    <!--
    <Manager pathname="" />
    -->

    <!-- Uncomment this to enable Comet connection tacking (provides events
         on session expiration as well as webapp lifecycle) -->
    <!--
    <Valve className="org.apache.catalina.valves.CometConnectionManagerValve" />
    -->
</Context>
```

L'élément <Realm>

Cet élément permet de gérer le système d'authentification. L'élément <Realm> peut être défini dans un des éléments suivants :

- <Engine> - le système d'authentification s'applique au moteur,

- <Host> - le système d'authentification s'applique au hôte,
- <Context> - le système d'authentification s'applique à l'application.

A noter qu'il ne peut être défini qu'une seule fois par élément et qu'il existe un héritage de l'élément enfant à partir de l'élément parent.

L'attribut fondamental de l'élément <Realm> est **className** qui prend une des valeurs suivantes :

Attribut	Valeur	Description
className	org.apache.catalina.realm.UserDatabaseRealm	Authentification utilisant le fichier \$CATALINA_HOME/conf/tomcat-users.xml
className	org.apache.catalina.realm.JDBCRealm	Authentification utilisant une base de données
className	org.apache.catalina.realm.DataSourceRealm	Authentification utilisant une base de données
className	org.apache.catalina.realm.JNDIRealm	Authentification basée sur un annuaire LDAP
className	org.apache.catalina.realm.JAASRealm	Authentification basée sur l'API de sécurité Java (JAAS)
className	org.apache.catalina.realm.CombinedRealm	Authentification basée sur plusieurs Realms définis en tant qu'enfants

L'élément <Loader>

Les classes Java sont chargées dans un ordre spécifique par défaut :

- les classes dans le répertoire **/WEB-INF/classes**,
- les classes dans les fichiers .jar dans le répertoire **/WEB-INF/lib**,
- les classes rendues accessibles aux applications par le moteur de servlets.

L'élément facultatif <Loader> permet de modifier l'ordre par défaut pour une application donnée en chargeant d'abord les classes recherchées par les chargeurs parents de Tomcat.

Les attributs les plus importants de cet élément sont :

Attribut	Description	Valeur par défaut	Valeur recommandée
delegate	Permet de modifier l'ordre par défaut	false	false
reloadable	Active une surveillance des répertoires /WEB-INF/lib et /WEB-INF/classes. Chaque modification apportée au contenu des deux répertoires est automatiquement prise en charge par le serveur	Héritée de <Context>	-
checkInterval	Intervalle de temps de la vérification de l'attribut reloadable	15s	15s

L'élément <Manager>

Cet élément permet de configurer la gestion des sessions.

L'attribut fondamental de l'élément <Manager> est **className** qui prend une des valeurs suivantes :

Attribut	Valeur	Description
className	org.apache.catalina.session.StandardManager	Les sessions sont stockées dans un fichier
className	org.apache.catalina.session.PersistentManager	Les session sont stockées dans un entrepôt persistant tel une base de données

Les attributs communs aux deux valeurs de **className** sont :

Attribut	Description	Valeur par défaut	Valeur recommandée
maxActiveSessions	Le nombre maximum de sessions actives (-1 équivaut pas de limite)	-1	-1
distributable	Spécifie si les attributs de session peuvent être stockés dans un fichier ou une base de données	false	false

L'attribut spécifique à **org.apache.catalina.session.StandardManager** est :

Attribut	Description	Valeur par défaut	Valeur recommandée
pathname	Spécifie un chemin relatif ou absolu vers le fichier de stockage	SESSIONS.ser	SESSIONS.ser

Les attributs spécifiques à **org.apache.catalina.session.PersistentManager** sont :

Attribut	Description	Valeur par défaut	Valeur recommandée
saveOnRestart	Spécifie que les sessions soient sauvegardées quand Tomcat est arrêtée	true	true
maxIdleBackup	Spécifie l'intervalle de temps en secondes depuis le dernier accès à la session avant qu'elle soit éligible pour être persistée. La valeur de -1 désactive cette fonctionnalité	-1	-1
maxIdleSwap	Spécifie l'intervalle de temps en secondes depuis le dernier accès à la session avant qu'elle soit persistée et supprimée de la mémoire. Cette valeur doit être supérieure ou égale à la valeur de maxIdleBackup . La valeur de -1 désactive cette fonctionnalité	-1	-1

L'élément <Store>

Cet élément est un élément enfant de <Manager>. Il doit être spécifié si la valeur de **org.apache.catalina.session.PersistentManager** est **true**.

L'attribut fondamental de l'élément <Store> est **className** qui prend une des valeurs suivantes :

Attribut	Valeur	Description
className	org.apache.catalina.session.FileStore	L'entrepôt persistant est un fichier
className	org.apache.catalina.session.JDBCStore	L'entrepôt persistant est une base de données

L'élément <Valve>

L'élément <Valve> est à considérer comme un **filtre**.

L'élément <Valve> peut être défini dans un des éléments suivants :

- <Engine> - le filtre s'applique au moteur,
- <Host> - le filtre s'applique au hôte,
- <Context> - le filtre s'applique à l'application.

L'attribut fondamental de l'élément <Valve> est **className** qui prend une des valeurs suivantes :

Attribut	Valeur	Description
className	org.apache.catalina.valves.AccessLogValve	Génère un journal des accès au serveur dans un fichier
className	org.apache.catalina.valves.ExtendedAccessLogValve	Génère un journal des accès au serveur dans un fichier au format Extended Log File Format du W3C
className	org.apache.catalina.valves.JDBCAccessLogValve	Génère un journal des accès au serveur dans une base de données
className	org.apache.catalina.valves.RemoteAddrValve	Applique une restriction des accès en fonction de l'IP du client
className	org.apache.catalina.valves.RemoteHostValve	Applique une restriction des accès en fonction du nom de la machine du client
className	org.apache.catalina.valves.SingleSignOn	Permet une authentification unique entre plusieurs applications
className	org.apache.catalina.valves.RemoteIPValve	Permet de récupérer l'adresse IP du client même si la requête est passée par un reverse proxy

Pour clarifier l'utilisation des filtres voici deux exemples :

Filtrage de l'adresse IP

```
<Valve className="org.apache.catalina.valves.RemoteAddrvalve" allow="127.0.0.1, 10.*" />
```

Filtrage de nom de la machine du client

```
<Valve className="org.apache.catalina.valves.RemoteHostValve" Deny="*.fenestros.com" />
```

LAB #1 -Journalisation des Requêtes Client dans un Fichier Texte

Tomcat propose deux formats ou *motifs* prédéfinis pour générer des journaux :

- **common**
 - %h %l %u %t "%r" %s %b
- **combined**
 - %h %l %u %t "%r" %s %b "%{Referer}i" "%{User-Agent}i"

Dans les patterns ci-dessus, les significations des chaînes de formatage sont les suivantes :

Chaîne	Description
%h	Le nom d'hôte distant ou l'adresse IP si resolveHosts vaut false
%u	Le nom du compte de l'utilisateur distant s'il y a eu authentification
%t	L'heure de la réception de la requête par le serveur
%r	La première ligne de la requête
%s	Le statut de la dernière requête
%b	La taille de la réponse sans les entêtes HTTP. Le caractère - indique 0
%{Referer}i	Le nom du site d'où vient la requête, contenu dans l'entête de la requête HTTP

Chaîne	Description
<code>%{User-Agent}i</code>	Le nom du navigateur utilisé par le client, contenu dans l'entête de la requête HTTP

Pour mettre en place la journalisation personnalisée modifiez le fichier **\$CATALINA_HOME/conf/server.xml** en commentant la section suivante :

```
...
    <!-- <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" /> -->

    </Host>
</Engine>
</Service>
```

et ajoutez la section suivante juste après :

```
...
    <Valve className="org.apache.catalina.valves.AccessLogValve"
        directory="logs" prefix="tomcat_access." suffix=".txt"
        resolveHosts="false" fileDateFormat="yyyy_MM_dd"
        pattern="%t - %a %H %s - %r"/>
...
```

Vous obtiendrez un résultat similaire au suivant :

```
...
    <!-- Access log processes all example.
        Documentation at: /docs/config/valve.html
        Note: The pattern used is equivalent to using pattern="common" -->
    <!--<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" /> -->

    <Valve className="org.apache.catalina.valves.AccessLogValve"
```

```
    directory="logs" prefix="tomcat_access." suffix=".txt"
    resolveHosts="false" fileDateFormat="yyyy_MM_dd"
    pattern="%t - %a %H %s - %r"/>

</Host>
</Engine>
</Service>
```

Dans cet exemple, on peut noter l'utilisation de plusieurs attributs supplémentaires :

Attribut	Description	Valeur par défaut	Valeur de l'exemple
directory	Le chemin absolu ou relatif vers le répertoire de sauvegarde des journaux	\$CATALINA_HOME/logs	logs
prefix	Le nom du fichier suivi par un point	access_log.	tomcat_access.
suffix	L'extension du fichier, précédée par un point	vide	.txt
resolveHosts	Transforme les adresse IP en nom d'hôte en utilisant un serveur DNS	false	false
fileDateFormat	Spécifie le format de date utilisé pour nommer le fichier	-	yyyy_MM_dd

Voici le fichier **server.xml** au complet :

```
<?xml version='1.0' encoding='utf-8'?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements.  See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License.  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
```

```
limitations under the License.
-->
<!-- Note: A "Server" is not itself a "Container", so you may not
       define subcomponents such as "Valves" at this level.
       Documentation at /docs/config/server.html
-->
<Server port="8005" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <!-- Security listener. Documentation at /docs/config/listeners.html
  <Listener className="org.apache.catalina.security.SecurityListener" />
  -->
  <!--APR library loader. Documentation at /docs/apr.html -->
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <!-- Prevent memory leaks due to use of particular java/javax APIs-->
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />

  <!-- Global JNDI resources
       Documentation at /docs/jndi-resources-howto.html
  -->
  <GlobalNamingResources>
    <!-- Editable user database that can also be used by
         UserDataBaseRealm to authenticate users
    -->
    <Resource name="UserDatabase" auth="Container"
              type="org.apache.catalina.UserDatabase"
              description="User database that can be updated and saved"
              factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
              pathname="conf/tomcat-users.xml" />
  </GlobalNamingResources>

  <!-- A "Service" is a collection of one or more "Connectors" that share
       a single "Container" Note: A "Service" is not itself a "Container",
```

```
    so you may not define subcomponents such as "Valves" at this level.
    Documentation at /docs/config/service.html
-->
<Service name="Catalina">

    <!--The connectors can use a shared executor, you can define one or more named thread pools-->
    <!--
    <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
        maxThreads="150" minSpareThreads="4"/>
    -->

    <!-- A "Connector" represents an endpoint by which requests are received
        and responses are returned. Documentation at :
        Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)
        Java AJP Connector: /docs/config/ajp.html
        APR (HTTP/AJP) Connector: /docs/apr.html
        Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
    -->
    <Connector port="8080" protocol="HTTP/1.1"
        connectionTimeout="20000"
        redirectPort="8443" />
    <!-- A "Connector" using the shared thread pool-->
    <!--
    <Connector executor="tomcatThreadPool"
        port="8080" protocol="HTTP/1.1"
        connectionTimeout="20000"
        redirectPort="8443" />
    -->
    <!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
        This connector uses the NIO implementation that requires the JSSE
        style configuration. When using the APR/native implementation, the
        OpenSSL style configuration is required as described in the APR/native
        documentation -->
```

```
<!--  
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"  
    maxThreads="150" SSLEnabled="true" scheme="https" secure="true"  
    clientAuth="false" sslProtocol="TLS" />  
-->  
  
<!-- Define an AJP 1.3 Connector on port 8009 -->  
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />  
  
<!-- An Engine represents the entry point (within Catalina) that processes  
    every request. The Engine implementation for Tomcat stand alone  
    analyzes the HTTP headers included with the request, and passes them  
    on to the appropriate Host (virtual host).  
    Documentation at /docs/config/engine.html -->  
  
<!-- You should set jvmRoute to support load-balancing via AJP ie :  
<Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">  
-->  
<Engine name="Catalina" defaultHost="localhost">  
  
    <!--For clustering, please take a look at documentation at:  
        /docs/cluster-howto.html (simple how to)  
        /docs/config/cluster.html (reference documentation) -->  
    <!--  
    <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>  
    -->  
  
    <!-- 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>

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

    <!-- SingleSignOn valve, share authentication between web applications
        Documentation at: /docs/config/valve.html -->
    <!--
    <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
    -->
    <Valve className="org.apache.catalina.valves.AccessLogValve"
        directory="logs" prefix="tomcat_access." suffix=".txt"
        resolveHosts="false" fileDateFormat="yyyy_MM_dd"
        pattern="%t - %a %H %s - %r"/>
    <!-- Access log processes all example
        Documentation at: /docs/config/valve.html
        Note: The pattern used is equivalent to using pattern="common" -->

</Host>
</Engine>
</Service>
</Server>
```

Dans ce cas, le fichier généré sera nommé **tomcat_access.yyyy_MM_dd.txt**.

Un dernier attribut peut également être présent. Il s'agit de l'attribut **rotatable**. Dans le cas où cet attribut porte la valeur **vrai**, les fichiers de journalisation subiront une rotation en fonction du **fileDateFormat** :

Format de fileDateFormat	Périodicité de la rotation
yyyy_MM_dd	Journalier
yyyy_MM	Hebdomadaire

Format de fileDateFormat	Périodicité de la rotation
yyyy	Annuelle

Dernièrement les significations des chaînes de formatage supplémentaires dans l'exemple sont les suivantes :

Chaîne	Description
%a	L'adresse IP distante
%H	Le protocole utilisé dans la requête

Redémarrez maintenant le serveur Tomcat :

```
[root@centos7 work]# cd ../bin
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Consultez la page web <http://www.i2tch.loc/docs/> afin de générer des traces dans le log puis consultez le répertoire `/$CATALINA_HOME/logs/` :

```
[root@centos7 bin]# lynx --dump http://www.i2tch.loc/docs/
[1] Tomcat Home
[2] The Apache Software Foundation

Apache Tomcat 8
```

Version 8.0.36, Jun 9 2016

Links

- * [3]Docs Home
- * [4]FAQ
- * [5]User Comments

...

```
[root@centos7 bin]# ls /$CATALINA_HOME/logs/
catalina.2019-10-28.log      localhost.2019-10-29.log
catalina.2019-10-29.log      localhost_access_log.2019-10-28.txt
catalina.out                localhost_access_log.2019-10-29.txt
host-manager.2019-10-28.log  manager.2019-10-28.log
host-manager.2019-10-29.log  manager.2019-10-29.log
localhost.2019-10-28.log     tomcat_access.2019_10_29.txt
```



Important : Notez la présence d'un fichier dont le nom est au format **tomcat_access.yyyy-MM-dd.txt**.

Consultez le contenu du fichier **tomcat_access.yyyy-MM-dd.txt**. Vous obtiendrez un résultat similaire à celui-ci :

```
[root@centos7 bin]# cat /$CATALINA_HOME/logs/tomcat_access.2019_10_29.txt
[29/Oct/2019:13:11:56 +0100] - 10.0.2.51 HTTP/1.0 200 - GET /docs/ HTTP/1.0
```



A Faire : Vérifiez que le résultat est conforme au *motif* spécifié dans le fichier **\$CATALINA_HOME/conf/server.xml**.

Pour pouvoir utiliser MariaDB pour consigner le journal de connexions, il convient d'abord de créer le fichier **tomcat.sql** contenant les lignes suivantes :

```
USE `tomcat`;
CREATE TABLE `tomcat`.`AccessLog` (
  `id` INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  `remoteHost` CHAR(15) NOT NULL DEFAULT '',
  `userName` CHAR(15),
  `timestamp` TIMESTAMP NOT NULL DEFAULT 0,
  `virtualHost` VARCHAR(64) NOT NULL DEFAULT '',
  `method` VARCHAR(8) NOT NULL DEFAULT '',
  `query` VARCHAR(255) NOT NULL DEFAULT '',
  `status` INTEGER UNSIGNED NOT NULL DEFAULT 0,
  `bytes` INTEGER UNSIGNED NOT NULL DEFAULT 0,
  `referer` VARCHAR(128),
  `userAgent` VARCHAR(128),
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARACTER SET latin1;
```

Par exemple :

```
[root@centos7 bin]# vi tomcat.sql
[root@centos7 bin]# cat tomcat.sql
USE `tomcat`;
CREATE TABLE `tomcat`.`AccessLog` (
  `id` INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  `remoteHost` CHAR(15) NOT NULL DEFAULT '',
  `userName` CHAR(15),
  `timestamp` TIMESTAMP NOT NULL DEFAULT 0,
  `virtualHost` VARCHAR(64) NOT NULL DEFAULT '',
  `method` VARCHAR(8) NOT NULL DEFAULT '',
  `query` VARCHAR(255) NOT NULL DEFAULT '',
  `status` INTEGER UNSIGNED NOT NULL DEFAULT 0,
  `bytes` INTEGER UNSIGNED NOT NULL DEFAULT 0,
```

```
`referer` VARCHAR(128),  
`userAgent` VARCHAR(128),  
PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARACTER SET latin1;
```

Ce fichier contient les commandes SQL nécessaires pour créer la table **AccessLog** qui recevra les traces.

Créez donc la base de données **tomcat** :

```
[root@centos7 bin]# mysql -u root -p  
Enter password:fenestros  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 6  
Server version: 5.5.64-MariaDB MariaDB Server  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> CREATE DATABASE tomcat;  
...  
MariaDB [(none)]> exit
```

Injectez donc les commandes SQL dans MariaDB :

```
[root@centos7 bin]# mysql -u root -p tomcat < tomcat.sql  
Enter password:fenestros  
[root@centos7 bin]# mysql -u root -p  
Enter password:fenestros  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 6  
Server version: 5.5.64-MariaDB MariaDB Server  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
```

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SHOW DATABASES;

```
+-----+
| Database          |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| test               |
| tomcat              |
+-----+
5 rows in set (0.00 sec)
```

MariaDB [(none)]> USE tomcat;

Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed

MariaDB [tomcat]> SHOW TABLES;

```
+-----+
| Tables_in_tomcat |
+-----+
| AccessLog         |
+-----+
1 row in set (0.00 sec)
```

MariaDB [tomcat]> DESCRIBE AccessLog;

```
+-----+-----+-----+-----+-----+-----+
| Field          | Type              | Null | Key | Default          | Extra          |
+-----+-----+-----+-----+-----+-----+
| id             | int(10) unsigned  | NO   | PRI | NULL              | auto_increment |
| remoteHost     | char(15)           | NO   |     |                   |                 |
| userName       | char(15)           | YES  |     | NULL              |                 |
```

timestamp	timestamp	NO		0000-00-00 00:00:00		
virtualHost	varchar(64)	NO				
method	varchar(8)	NO				
query	varchar(255)	NO				
status	int(10) unsigned	NO		0		
bytes	int(10) unsigned	NO		0		
referer	varchar(128)	YES		NULL		
userAgent	varchar(128)	YES		NULL		

+-----+-----+-----+-----+-----+-----+

11 rows in set (0.00 sec)

MariaDB [tomcat]> exit

Bye

Téléchargez maintenant le pilote JDBC pour MySQL :

```
[root@centos7 bin]# wget http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.39.tar.gz
```

Décompressez-le

```
[root@centos7 bin]# tar xvf mysql-connector-java-5.1.39.tar.gz
```

...

Copiez le fichier **mysql-connector-java-5.1.39-bin.jar** dans le répertoire **\$CATALINA_HOME/lib** :

```
[root@centos7 bin]# cp mysql-connector-java-5.1.39/mysql-connector-java-5.1.39-bin.jar $CATALINA_HOME/lib
```

Ajoutez maintenant la section suivante au fichier **\$CATALINA_HOME/conf/server.xml** à **la place des lignes précédemment ajoutées** :

```
[root@centos7 bin]# vi $CATALINA_HOME/conf/server.xml
[root@centos7 bin]# tail -n25 $CATALINA_HOME/conf/server.xml
    <Host name="localhost" appBase="webapps"
        unpackWARs="true" autoDeploy="true">
```

```
<!-- SingleSignOn valve, share authentication between web applications
      Documentation at: /docs/config/valve.html -->
<!--
<Valve className="org.apache.catalina.authenticator.SingleSignOn" />
-->

<!-- Access log processes all example.
      Documentation at: /docs/config/valve.html
      Note: The pattern used is equivalent to using pattern="common" -->
<!-- <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" /> -->

<Valve className="org.apache.catalina.valves.JDBCAccessLogValve"
      connectionURL="jdbc:mysql://localhost:3306/tomcat?user=root&password=fenestros"
      driverName="com.mysql.jdbc.Driver" tableName="AccessLog"
      resolveHosts="false" pattern="common" />

</Host>
</Engine>
</Service>
</Server>
```

Redémarrez le serveur Tomcat :

```
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:        /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
```



```
-----+
1 row in set (0.00 sec)

MariaDB [tomcat]> exit
Bye
```

L'élément <Listener>

Cet élément permet de définir des écouteurs d'événements sur les éléments suivants :

- <Server>,
- <Engine>,
- <Host>,
- <Context>.

Le seul attribut de <Listener> est **className** qui prend une valeur différente selon les informations à collecter.

Cinq écouteurs par défaut sont détaillés dans le fichier **\$CATALINA_HOME/conf/server.xml** :

```
<Server port="8005" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
  <GlobalNamingResources>
    ...
```

\$CATALINA_HOME/conf/web.xml

Ce fichier définit les paramètres de configuration communs à toutes les applications de Tomcat. Les directives actives de ce fichier sont :

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  version="3.1">

  <servlet>
    <servlet-name>default</servlet-name>
    <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
    <init-param>
      <param-name>debug</param-name>
      <param-value>0</param-value>
    </init-param>
    <init-param>
      <param-name>listings</param-name>
      <param-value>false</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet>
    <servlet-name>jsp</servlet-name>
    <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
    <init-param>
      <param-name>fork</param-name>
      <param-value>false</param-value>
    </init-param>
    <init-param>
      <param-name>xpoweredBy</param-name>
      <param-value>false</param-value>
    </init-param>
    <load-on-startup>3</load-on-startup>
```

```
</servlet>

<servlet-mapping>
  <servlet-name>default</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>jsp</servlet-name>
  <url-pattern>*.jsp</url-pattern>
  <url-pattern>*.jspx</url-pattern>
</servlet-mapping>

<session-config>
  <session-timeout>30</session-timeout>
</session-config>

<mime-mapping>
  <extension>l23</extension>
  <mime-type>application/vnd.lotus-1-2-3</mime-type>
</mime-mapping>
<mime-mapping>
  <extension>3dml</extension>
  <mime-type>text/vnd.in3d.3dml</mime-type>
...
<mime-mapping>
  <extension>xht</extension>
  <mime-type>application/xhtml+xml</mime-type>
</mime-mapping>
<mime-mapping>
  <extension>xhtml</extension>
  <mime-type>application/xhtml+xml</mime-type>
</mime-mapping>
<mime-mapping>
  <extension>zir</extension>
```

```
<mime-type>application/vnd.zul</mime-type>
</mime-mapping>
<mime-mapping>
  <extension>zirz</extension>
  <mime-type>application/vnd.zul</mime-type>
</mime-mapping>
<mime-mapping>
  <extension>zmm</extension>
  <mime-type>application/vnd.handheld-entertainment+xml</mime-type>
</mime-mapping>

<welcome-file-list>
  <welcome-file>index.html</welcome-file>
  <welcome-file>index.htm</welcome-file>
  <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
</web-app>
```

Dans ce fichier, on peut constater la présence de plusieurs sections :

- `<servlet-name>default</servlet-name>`,
 - Cette section contient un paramètre **listings** qui permet l'affichage du contenu d'un répertoire dans le cas où aucune page d'accueil soit présente,
- `<servlet-name>jsp</servlet-name>`,
 - Cette section est responsable de la transformation des pages JSP en servlets,
- `<servlet-mapping>`,
 - Cette section définit les servlets responsable en fonction des extensions de fichiers,
- `<session-config>`,
 - Cette section définit le timeout par défaut. La valeur est en minutes.
- `<mime-mapping>`,
 - Ces sections définissent les types de fichiers et leurs extensions,
- `<welcome-file-list>`,
 - Cette section définit les pages d'accueil par défaut.

\$CATALINA_HOME/conf/tomcat-users.xml

Ce fichier est utilisé par le mécanisme d'authentification par défaut de Tomcat et contient des entrées de type *nom d'utilisateur*, *mot de passe* et *rôle* :

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users xmlns="http://tomcat.apache.org/xml"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
               version="1.0">
<!--
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
  <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
  <user username="role1" password="<must-be-changed>" roles="role1"/>
-->
</tomcat-users>
```

Depuis la version 7 de Tomcat, l'application **manager** un des rôles suivants :

- **manager-gui**,
 - permet l'accès au manager en mode HTML graphique,
- **manager-script**,
 - permet l'accès au manager en mode texte,
- **manager-jmx**,
 - permet uniquement l'accès aux pages de supervision JMX et d'état du serveur,
- **manager-status**,
 - permet uniquement l'accès à la page d'état du serveur.

Afin donc de pouvoir utiliser l'application manager en mode texte, **remplacez** le contenu du fichier **\$CATALINA_HOME/conf/tomcat-users.xml** avec le contenu suivant :

```
[root@centos7 bin]# vi $CATALINA_HOME/conf/tomcat-users.xml
```

```
[root@centos7 bin]# cat $CATALINA_HOME/conf/tomcat-users.xml
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users xmlns="http://tomcat.apache.org/xml"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
               version="1.0">
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <role rolename="manager-script"/>
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user username="both" password="tomcat" roles="tomcat,role1"/>
  <user username="role1" password="tomcat" roles="role1"/>
  <user username="admin" password="fenestros" roles="manager-script"/>
</tomcat-users>
```

Arrêtez puis démarrez le serveur Tomcat :

```
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

\$CATALINA_HOME/conf/catalina.policy

Par défaut, Tomcat est démarré sans **Gestionnaire de Sécurité** (*SecurityManager*). Dans ce cas, les applications peuvent faire tout ce qu'elles veulent. Dans le cas où on souhaite démarrer Tomcat en utilisant le Gestionnaire de Sécurité, il convient de lancer Tomcat avec le script **startup.sh** en stipulant l'argument **security**. Dans ce cas, tout ce qui n'est pas autorisé dans le fichier **catalina.policy** est interdit.

```
// Licensed to the Apache Software Foundation (ASF) under one or more
// contributor license agreements. See the NOTICE file distributed with
// this work for additional information regarding copyright ownership.
// The ASF licenses this file to You under the Apache License, Version 2.0
// (the "License"); you may not use this file except in compliance with
// the License. You may obtain a copy of the License at
//
//      http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// =====
// catalina.policy - Security Policy Permissions for Tomcat
//
// This file contains a default set of security policies to be enforced (by the
// JVM) when Catalina is executed with the "-security" option. In addition
// to the permissions granted here, the following additional permissions are
// granted to each web application:
//
// * Read access to the web application's document root directory
// * Read, write and delete access to the web application's working directory
// =====
```

```
// ===== SYSTEM CODE PERMISSIONS =====

// These permissions apply to javac
grant codeBase "file:${java.home}/lib/-" {
    permission java.security.AllPermission;
};

// These permissions apply to all shared system extensions
grant codeBase "file:${java.home}/jre/lib/ext/-" {
    permission java.security.AllPermission;
};

// These permissions apply to javac when ${java.home} points at $JAVA_HOME/jre
grant codeBase "file:${java.home}/../lib/-" {
    permission java.security.AllPermission;
};

// These permissions apply to all shared system extensions when
// ${java.home} points at $JAVA_HOME/jre
grant codeBase "file:${java.home}/lib/ext/-" {
    permission java.security.AllPermission;
};

// ===== CATALINA CODE PERMISSIONS =====

// These permissions apply to the daemon code
grant codeBase "file:${catalina.home}/bin/commons-daemon.jar" {
    permission java.security.AllPermission;
};

// These permissions apply to the logging API
```

```
// Note: If tomcat-juli.jar is in ${catalina.base} and not in ${catalina.home},
// update this section accordingly.
// grant codeBase "file:${catalina.base}/bin/tomcat-juli.jar" {...}
grant codeBase "file:${catalina.home}/bin/tomcat-juli.jar" {
    permission java.io.FilePermission
        "${java.home}${file.separator}lib${file.separator}logging.properties", "read";

    permission java.io.FilePermission
        "${catalina.base}${file.separator}conf${file.separator}logging.properties", "read";
    permission java.io.FilePermission
        "${catalina.base}${file.separator}logs", "read, write";
    permission java.io.FilePermission
        "${catalina.base}${file.separator}logs${file.separator}*", "read, write";

    permission java.lang.RuntimePermission "shutdownHooks";
    permission java.lang.RuntimePermission "getClassLoader";
    permission java.lang.RuntimePermission "setContextClassLoader";

    permission java.lang.management.ManagementPermission "monitor";

    permission java.util.logging.LoggingPermission "control";

    permission java.util.PropertyPermission "java.util.logging.config.class", "read";
    permission java.util.PropertyPermission "java.util.logging.config.file", "read";
    permission java.util.PropertyPermission "org.apache.juli.AsyncLoggerPollInterval", "read";
    permission java.util.PropertyPermission "org.apache.juli.AsyncMaxRecordCount", "read";
    permission java.util.PropertyPermission "org.apache.juli.AsyncOverflowDropType", "read";
    permission java.util.PropertyPermission "org.apache.juli.ClassLoaderLogManager.debug", "read";
    permission java.util.PropertyPermission "catalina.base", "read";

    // Note: To enable per context logging configuration, permit read access to
    // the appropriate file. Be sure that the logging configuration is
    // secure before enabling such access.
    // E.g. for the examples web application (uncomment and unwrap
```

```
// the following to be on a single line):
// permission java.io.FilePermission "${catalina.base}${file.separator}
// webapps${file.separator}examples${file.separator}WEB-INF
// ${file.separator}classes${file.separator}logging.properties", "read";
};

// These permissions apply to the server startup code
grant codeBase "file:${catalina.home}/bin/bootstrap.jar" {
    permission java.security.AllPermission;
};

// These permissions apply to the servlet API classes
// and those that are shared across all class loaders
// located in the "lib" directory
grant codeBase "file:${catalina.home}/lib/-" {
    permission java.security.AllPermission;
};

// If using a per instance lib directory, i.e. ${catalina.base}/lib,
// then the following permission will need to be uncommented
// grant codeBase "file:${catalina.base}/lib/-" {
//     permission java.security.AllPermission;
// };

// ===== WEB APPLICATION PERMISSIONS =====

// These permissions are granted by default to all web applications
// In addition, a web application will be given a read FilePermission
// for all files and directories in its document root.
grant {
    // Required for JNDI lookup of named JDBC DataSource's and
```

```
// javamail named MimePart DataSource used to send mail
permission java.util.PropertyPermission "java.home", "read";
permission java.util.PropertyPermission "java.naming.*", "read";
permission java.util.PropertyPermission "javax.sql.*", "read";

// OS Specific properties to allow read access
permission java.util.PropertyPermission "os.name", "read";
permission java.util.PropertyPermission "os.version", "read";
permission java.util.PropertyPermission "os.arch", "read";
permission java.util.PropertyPermission "file.separator", "read";
permission java.util.PropertyPermission "path.separator", "read";
permission java.util.PropertyPermission "line.separator", "read";

// JVM properties to allow read access
permission java.util.PropertyPermission "java.version", "read";
permission java.util.PropertyPermission "java.vendor", "read";
permission java.util.PropertyPermission "java.vendor.url", "read";
permission java.util.PropertyPermission "java.class.version", "read";
permission java.util.PropertyPermission "java.specification.version", "read";
permission java.util.PropertyPermission "java.specification.vendor", "read";
permission java.util.PropertyPermission "java.specification.name", "read";

permission java.util.PropertyPermission "java.vm.specification.version", "read";
permission java.util.PropertyPermission "java.vm.specification.vendor", "read";
permission java.util.PropertyPermission "java.vm.specification.name", "read";
permission java.util.PropertyPermission "java.vm.version", "read";
permission java.util.PropertyPermission "java.vm.vendor", "read";
permission java.util.PropertyPermission "java.vm.name", "read";

// Required for OpenJMX
permission java.lang.RuntimePermission "getAttribute";

// Allow read of JAXP compliant XML parser debug
permission java.util.PropertyPermission "jaxp.debug", "read";
```

```
// All JSPs need to be able to read this package
permission java.lang.RuntimePermission "accessClassInPackage.org.apache.tomcat";

// Precompiled JSPs need access to these packages.
permission java.lang.RuntimePermission "accessClassInPackage.org.apache.jasper.el";
permission java.lang.RuntimePermission "accessClassInPackage.org.apache.jasper.runtime";
permission java.lang.RuntimePermission
    "accessClassInPackage.org.apache.jasper.runtime.*";

// Precompiled JSPs need access to these system properties.
permission java.util.PropertyPermission
    "org.apache.jasper.runtime.BodyContentImpl.LIMIT_BUFFER", "read";
permission java.util.PropertyPermission
    "org.apache.el.parser.COERCE_TO_ZERO", "read";

// The cookie code needs these.
permission java.util.PropertyPermission
    "org.apache.catalina.STRICT_SERVLET_COMPLIANCE", "read";
permission java.util.PropertyPermission
    "org.apache.tomcat.util.http.ServerCookie.STRICT_NAMING", "read";
permission java.util.PropertyPermission
    "org.apache.tomcat.util.http.ServerCookie.FWD_SLASH_IS_SEPARATOR", "read";

// Applications using Comet need to be able to access this package
permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.comet";

// Applications using WebSocket need to be able to access these packages
permission java.lang.RuntimePermission "accessClassInPackage.org.apache.tomcat.websocket";
permission java.lang.RuntimePermission "accessClassInPackage.org.apache.tomcat.websocket.server";
};

// The Manager application needs access to the following packages to support the
// session display functionality. These settings support the following
```

```
// configurations:
// - default CATALINA_HOME == CATALINA_BASE
// - CATALINA_HOME != CATALINA_BASE, per instance Manager in CATALINA_BASE
// - CATALINA_HOME != CATALINA_BASE, shared Manager in CATALINA_HOME
grant codeBase "file:${catalina.base}/webapps/manager/-" {
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.ha.session";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.manager";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.manager.util";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.util";
};
grant codeBase "file:${catalina.home}/webapps/manager/-" {
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.ha.session";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.manager";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.manager.util";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.util";
};

// You can assign additional permissions to particular web applications by
// adding additional "grant" entries here, based on the code base for that
// application, /WEB-INF/classes/, or /WEB-INF/lib/ jar files.
//
// Different permissions can be granted to JSP pages, classes loaded from
// the /WEB-INF/classes/ directory, all jar files in the /WEB-INF/lib/
// directory, or even to individual jar files in the /WEB-INF/lib/ directory.
//
// For instance, assume that the standard "examples" application
// included a JDBC driver that needed to establish a network connection to the
// corresponding database and used the scrape taglib to get the weather from
// the NOAA web server. You might create a "grant" entries like this:
//
// The permissions granted to the context root directory apply to JSP pages.
// grant codeBase "file:${catalina.base}/webapps/examples/-" {
```

```
//      permission java.net.SocketPermission "dbhost.mycompany.com:5432", "connect";
//      permission java.net.SocketPermission "*.noaa.gov:80", "connect";
// };
//
// The permissions granted to the context WEB-INF/classes directory
// grant codeBase "file:${catalina.base}/webapps/examples/WEB-INF/classes/-" {
// };
//
// The permission granted to your JDBC driver
// grant codeBase "jar:file:${catalina.base}/webapps/examples/WEB-INF/lib/driver.jar!/-" {
//      permission java.net.SocketPermission "dbhost.mycompany.com:5432", "connect";
// };
// The permission granted to the scrape taglib
// grant codeBase "jar:file:${catalina.base}/webapps/examples/WEB-INF/lib/scrape.jar!/-" {
//      permission java.net.SocketPermission "*.noaa.gov:80", "connect";
// };
```

Configuration des Ressources

Les ressources disponibles à Tomcat sont configurées en leur donnant un **nom JNDI**. L'API JNDI est un service de nommage sous forme d'une arborescence de ressources.

Portée des Ressources

Une ressource peut être déclarée à trois endroits différents. Selon cet endroit, sa portée est différente :

- dans l'élément `<Context>`, elle est alors disponible à l'application uniquement,
- dans le contexte par défaut, elle est alors disponible à toutes les applications d'un hôte,
- dans `<GlobalNamingResources>`, elle est alors disponible au serveur.

Une ressource est déclarée avec l'élément **`<Resource>`**. Elle :

- est immédiatement disponible à l'application si elle est déclarée dans un <Context> ,
- est immédiatement disponible aux applications si elle est déclarée dans le contexte par défaut,
- nécessite d'être appelée par un élément **<ResourceLink>** si elle est déclarée dans <GlobalNamingResources> .

Les attributs de <Resource> sont :

Attribut	Description	Valeur par défaut	Valeur recommandée
name	Spécifie le nom JNDI	-	-
type	Spécifie le type de données java de la ressource	-	-
description	Une description de la ressource	-	-
auth	Le type d'authentification à la ressource	-	-

L'attribut **auth** peut avoir plusieurs valeurs différentes en fonction de l'emplacement de la configuration de l'authentification :

- Container - la configuration est dans le serveur,
- Application - la configuration est dans l'application.

L'**exemple** suivant, issu du fichier server.xml démontre la syntaxe :

```
<GlobalNamingResources>
  <Resource name="UserDatabase" auth="Container"
    type="org.apache.catalina.UserDatabase"
    description="User database that can be updated and saved"
    factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
    pathname="conf/tomcat-users.xml" />
</GlobalNamingResources>
```

Pools de Connexion

Pour augmenter le temps de réponse aux requêtes multiples de connexions JDBC, JEE utilise un système de pool de connexions.

Un pool de connexion est :

- créé par le serveur et non par l'application,
- est utilisable par plusieurs applications en même temps,
- ré-utilise des connexions non-utilisées,
- basé sur la bibliothèque **commons-dbcp**,
- est défini dans un élément **<Resource>**.

Sessions JavaMail

JavaMail :

- est l'API de messagerie pour l'envoi et la réception d'emails,
- supporte les protocoles SMTP, POP3 et IMAP4,
- nécessite l'API **JAF** (*Java Activation Framework*),
- doit être téléchargé car il n'est pas inclus en standard dans Tomcat,
- est défini dans un élément **<Resource>**,

Pour activer JavaMail, il convient de copier les fichiers **mail.jar** et **activation.jar** dans le répertoire **\$CATALINA_HOME/lib**.

JavaBeans

Un **JavaBean** :

- est un objet contenant :
 - des données,
 - des fonctions, appelées **méthodes**,
- est un conteneur de différentes données regroupées, par exemple, nom, prénom, adresse etc,
- est un conteneur de méthodes pour avoir accès aux données,
- est déclaré dans un élément **<Resource>**,
- nécessite un **<Factory>** pour se fabriquer,

Le **<Factory>** par défaut est **org.apache.naming.factory.BeanFactory**.

Entrées D'Environnement

Une **Entrée d'environnement** est à considérer comme une variable. Une Entrée d'environnement :

- se déclare par un élément <Environment> au même niveau que <Resource> et contient un élément <type> qui peut être :
 - java.lang.Boolean,
 - java.lang.Byte,
 - java.lang.Character,
 - java.lang.Double,
 - java.lang.Float,
 - java.lang.Integer,
 - java.lang.Long,
 - java.lang.Short,
 - java.lang.String.