



EXTRANETS

Manuel d'installation

Date	Auteur	Version	Commentaires
10-05-2023	WORLDLINE	1.0	Création de la documentation

TABLE DES MATIERES

1.	Introduction	3
1.1.	Objet du document.....	3
2.	Installation de Java	4
2.1.	Installation du JDK 1.8	4
2.2.	Installation sous Windows.....	4
2.2.1.	Installation sans invité de commande.....	4
2.2.2.	Installation avec invité de commande.....	6
2.3.	Installation sous Linux.....	6
2.4.	Paramétrage des variables d'environnement	6
3.	Installation du serveur Tomcat.....	8
3.1.	Récupérer la version de Tomcat à installer	8
3.2.	Installation sous Windows.....	8
3.3.	Installation sous Linux.....	12
3.4.	Ajout des bibliothèques communes aux applications	13
3.5.	Installation de plusieurs serveurs Tomcat	13
4.	Paramétrage SSL.....	15
4.1.	Création d'un certificat	15
4.2.	Paramétrage	15

1. Introduction

1.1. Objet du document

Ce document a pour but de décrire la procédure d'installation de l'environnement de déploiement des applications Extranets ainsi que le déploiement de ces applications.

Cette documentation doit être complétée avec la documentation spécifique de chaque Extranet, le présent document ne servant qu'à installer l'environnement.

2. Installation de Java

2.1. Installation du JDK 1.8

Plusieurs versions de Java peuvent être utilisées pour fonctionner avec un serveur Tomcat 9. Pour information, les développements des extranets ont été testés avec les versions suivantes :

- java-1.8.0-openjdk-1.8.0.282-1
- jdk1.8.0_144
- jre1.8.0_181

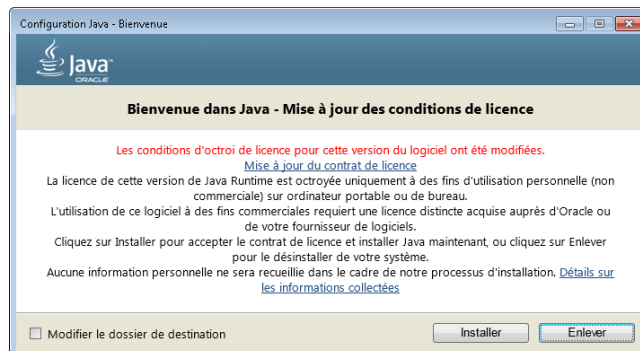
Si vous êtes accoutumés avec une installation manuelle, nous pouvons vous fournir les JRE ou JDK. Si vous souhaitez installer la dernière version de Java 8, procédez comme suit :

- Allez sur la page <https://www.java.com/fr/download/manual.jsp>
- Cliquez sur la version correspondant à votre système d'exploitation (Windows, Linux) (Même si Java est disponible pour MacOS, nous déconseillons de l'utiliser car Tomcat 9 n'est pas disponible pour Solaris et MacOS sur le lien donné plus bas).

2.2. Installation sous Windows

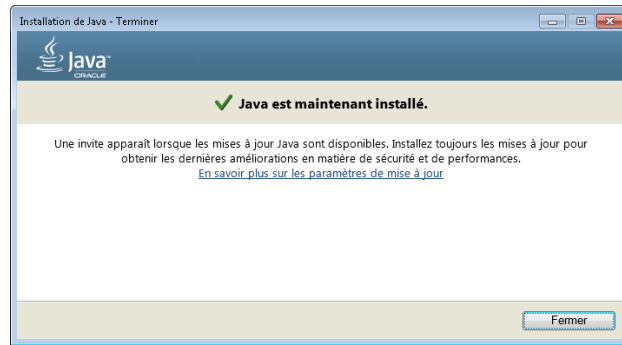
2.2.1. Installation sans invité de commande

- Une fois l'exécutable téléchargé, lancez-le pour commencer l'installation (les règles de sécurité peuvent varier en fonction de votre environnement, il peut être nécessaire d'installer Java avec des droits administrateur. Si tel est le cas et que vous n'arrivez pas à obtenir ces droits, contactez votre helpdesk / DSI / responsable technique).

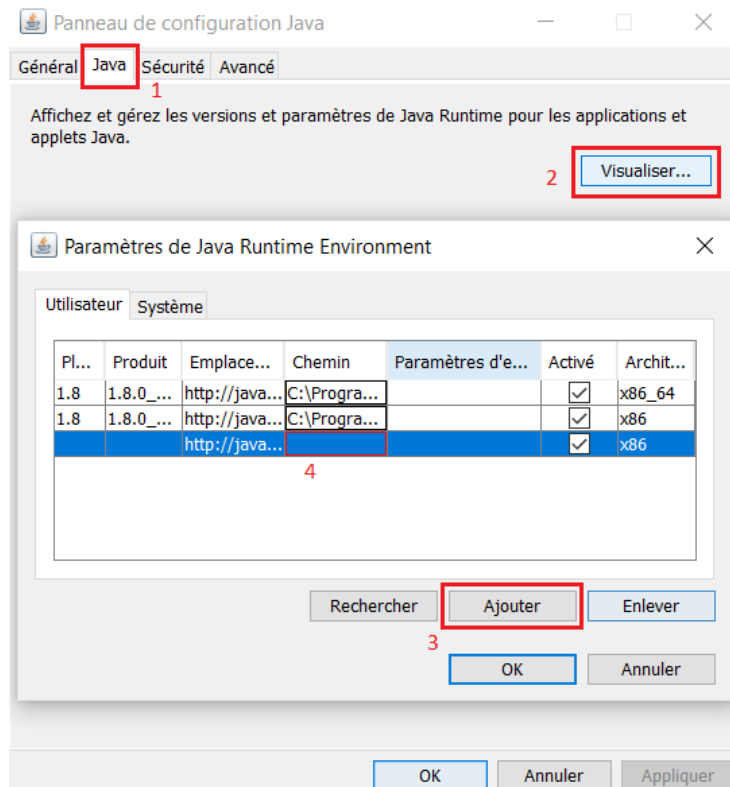


- Suivez les étapes d'installation





- Une fois l'installation terminée, ouvrez un invité de commande (menu Windows puis **cmd**)
- Lancez la commande **java -version** et vérifiez que la version affichée correspond à celle installée
- Si ce n'est pas le cas, lancez le programme **Configure Java** (menu Windows puis **Java**)
- Allez dans l'onglet **Java** (1)
- Cliquez sur le bouton **Visualiser...** (2)
- Vérifiez dans la liste que ladite version de Java installée apparaît bien dans le tableau
- Cochez la case activé si cette dernière n'est pas encore cochée
- Si votre version de Java n'apparaît pas dans la liste, cliquez sur le bouton **Ajouter** (3)
- Renseignez le chemin du répertoire d'installation de Java (4) (i.e. le répertoire où se sont installés les fichiers de Java, ne renseignez pas le chemin du répertoire de l'exécutable Java. En cas de doute vous pouvez prendre exemple sur les chemins d'une autre version de Java ou faire appel à votre helpdesk / DSI / responsable technique.)
- Validez vos modifications, dans certains cas, il est préférable de redémarrer la machine sur laquelle vous avez installé Java.



2.2.2. Installation avec invité de commande

Cette section décrit les options de ligne de commande pour le programme d'installation Windows hors connexion de l'environnement JRE. Exécutez le programme d'installation de la façon suivante :

```
jre [INSTALLCFG=configuration_file_path] [options]
```

`jre` fait référence au nom de fichier de base du programme d'installation Windows hors ligne de l'environnement JRE (par exemple, `jre-8u05-windows-i586.exe`).

`INSTALLCFG=configuration_file_path` indique le chemin du fichier de configuration du programme d'installation. Pour plus d'informations, reportez-vous à Installation à l'aide d'un fichier de configuration (docs.oracle.com).

`options` sont des options avec des valeurs indiquées, séparées par des espaces. Utilisez les mêmes options que celles indiquées dans Tableau 20-1, Options de fichier de configuration (docs.oracle.com). En outre, vous pouvez utiliser l'option `/s` pour que le programme d'installation Windows hors ligne de l'environnement JRE effectue une installation sans invite.

Sources : https://www.java.com/fr/download/help/silent_install_fr.html

2.3. Installation sous Linux

Les instructions ci-dessous se rapportent à l'installation de la version Java 8 Update 73 (8u73). Si vous installez une autre version, veuillez à modifier le numéro de version lorsque vous saisissez les commandes sur le terminal. Exemple : pour Java 8u79, remplacez 8u73 par 8u79. Comme dans l'exemple précédent, le numéro de version est parfois précédé de la lettre `u` et il est parfois précédé d'un trait de soulignement, par exemple `jre1.8.0_73`.

Remarque sur l'accès root : Pour installer Java dans un emplacement de niveau système tel que `/usr/local`, vous devez vous connecter en tant qu'utilisateur root pour obtenir les droits d'accès nécessaires. Si vous ne pouvez pas prendre le rôle d'utilisateur root, installez Java dans votre répertoire d'origine Oracle Home ou dans un sous-répertoire pour lequel vous disposez de droits en écriture.

Accédez au répertoire dans lequel vous souhaitez effectuer l'installation. Entrez :

```
cd chemin_du_répertoire
```

Par exemple, pour installer le logiciel dans le répertoire `/usr/java/`, entrez :

```
cd /usr/java/
```

Déplacez le fichier binaire d'archive `.tar.gz` dans le répertoire en cours.

Décompressez l'archive TAR et installez Java.

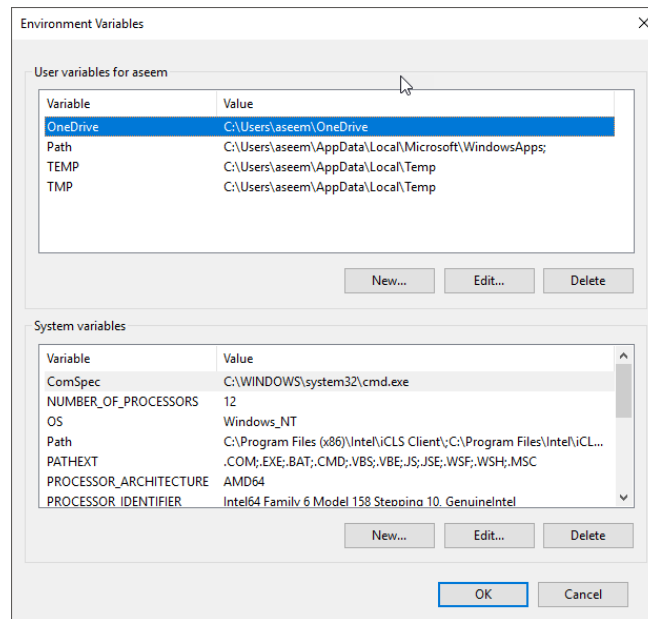
```
tar zxvf jre-8u73-linux-x64.tar.gz
```

Les fichiers Java sont installés dans un répertoire nommé **jre1.8.0_73** sous le répertoire en cours. Dans cet exemple, il est installé dans le répertoire `/usr/java/jre1.8.0_73`. Une fois l'installation terminée, le mot **Done** apparaît. Pour libérer de l'espace disque, vous pouvez supprimer le fichier `.tar.gz`.

Sources : https://www.java.com/fr/download/help/linux_x64_install.html

2.4. Paramétrage des variables d'environnement

- Dans Windows allez dans le menu
- *Paramètres > Panneau de configuration > Système > avancée > variables d'environnement*
- Cliquez sur Variables d'environnement et ajoutez le chemin du JRE à la variable PATH



3. Installation du serveur Tomcat

3.1. Récupérer la version de Tomcat à installer

Pour télécharger un serveur Tomcat 9 suivez les instructions suivantes :

- Allez sur la page <https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.56/bin> (ce lien sert à télécharger la version 9.0.56 de Tomcat, si vous êtes contraint par votre organisation d'utiliser une autre version, merci de nous le signaler afin que nous puissions tester la compatibilité du serveur avec les applicatifs existants.)
- Téléchargez la version souhaitée en fonction de votre système d'exploitation (Tomcat 9 n'est pas disponible pour les distributions Solaris et MacOS sur le lien donné).

3.2. Installation sous Windows

Assurez-vous que la variable d'environnement Java est correctement configurée :

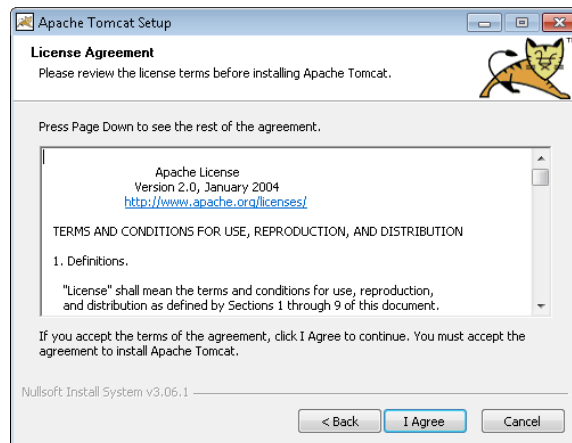
- Localisez votre répertoire d'installation de Java et copiez le chemin d'accès. Le chemin par défaut est C:\Program Files\Java\jdk_<numéro de version>.
- Depuis le menu Démarrer de Windows, accédez à **Paramètres système avancés**. La manière d'accéder à ces propriétés varie en fonction de votre version de Windows. Par exemple, sous Windows 10, recherchez **Variables d'environnement**, puis sélectionnez Modifier les variables d'environnement système.
- Cliquez sur **Variables d'environnement**.
- Dans la section **Variables système**, cliquez sur **Nouvelle**.
- Dans le champ Nom de la variable, spécifiez `JAVA_HOME`.
- Dans le champ Valeur de la variable, spécifiez le chemin d'accès à votre installation Java, comme défini à la première étape. Cliquez sur **OK**.

Si vous êtes familier avec l'installation de Tomcat vous pouvez directement copier / coller les fichiers du serveur sur la machine hôte et passer à la configuration, sinon, suivez les instructions suivantes :

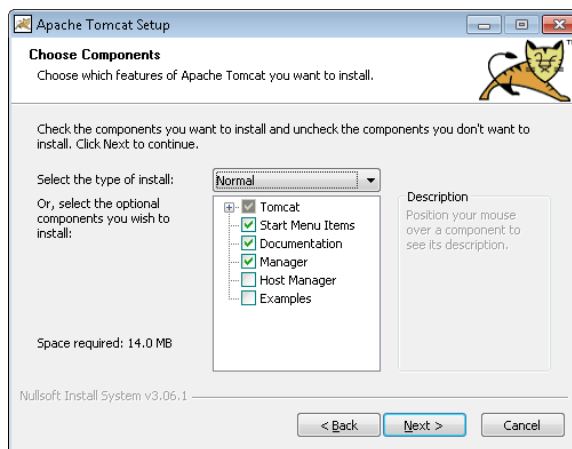
- Lancez l'assistant d'installation de Tomcat si vous l'avez téléchargé. Cliquez sur **Suivant**.



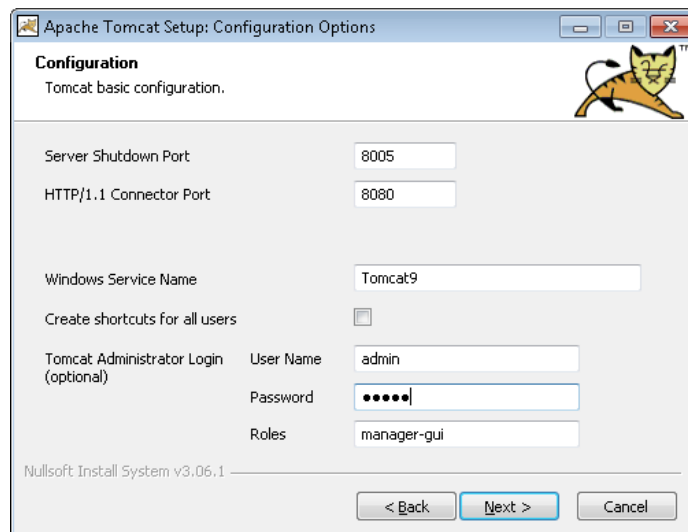
- Cliquez sur **I Agree**



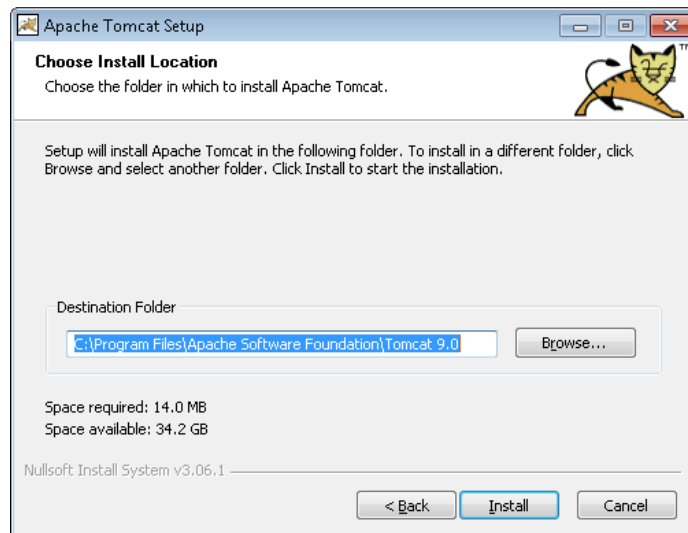
- Dans la section **Choose Components**, conservez les paramètres par défaut.



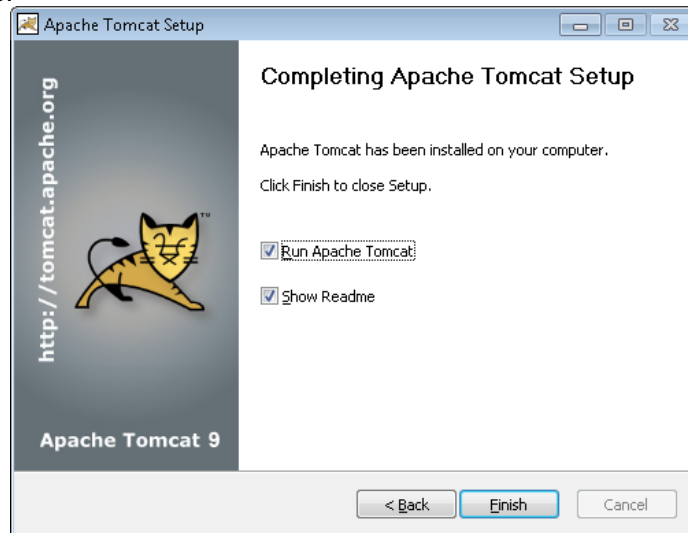
- Cliquez sur **Suivant**.
- Dans le champ **HTTP/1.1 Connector Port**, saisissez les ports de votre configuration.
- Dans les champs **Tomcat Administrator Login**, saisissez un nom d'utilisateur Tomcat et un mot de passe unique et sécurisé pour l'administration de Tomcat.
- Cliquez sur **Suivant**.



- Cliquez sur **Suivant** et choisissez le répertoire d'installation du serveur.

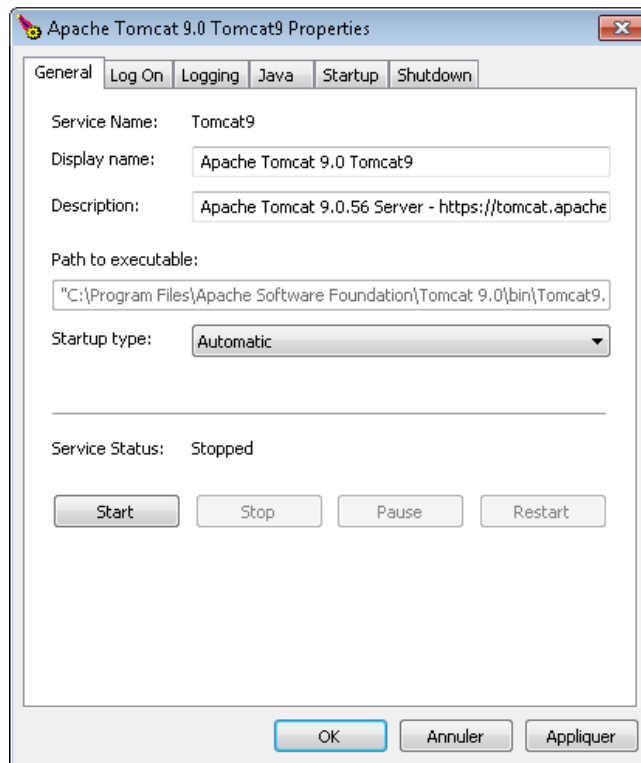


- Cliquez sur **Install**.
- Cliquez sur **Terminé**.

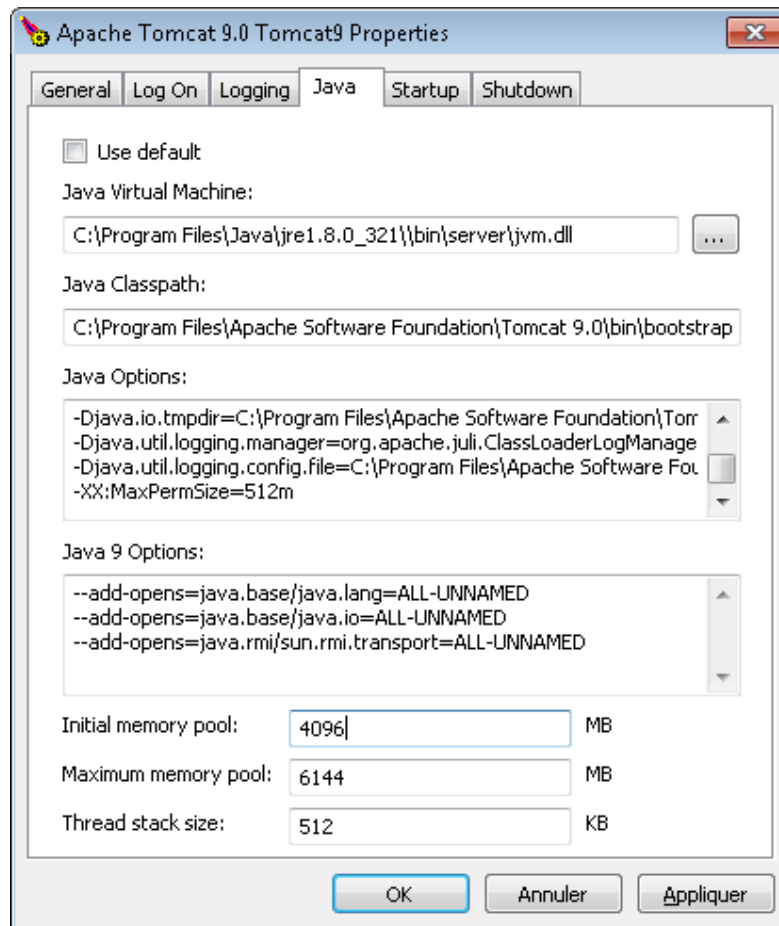


Configuration :

- Arrêtez Tomcat. Cliquez sur **Configure Tomcat** puis sur l'onglet **Java**. Dans un premier temps nous allons rendre le démarrage de Tomcat automatique en modifiant le startup type dans l'écran suivant :



- Paramétrez la mémoire utilisée par le serveur comme suit :
- **Initial memory pool** : est la taille mémoire de départ de allouée par la JVM.
- **Maximum memory pool** : est la taille mémoire maximum par défaut elle est à 64 KB, dans le cadre des applications Extranet Worldline on peut atteindre 2Go donc il faut saisir la taille maximum.
- La plupart des machines 32bits n'accepte pas plus que 1024 Mo, ce qui empêche le démarrage de Tomcat. **Sur les environnements en 32 Bits cette valeur est donc positionnée à 1024 Mo.**
- Une machine 64bits jusqu'à 4Go.
- **Thread stack size** : taille de stack trace pour chaque processus java/tâche elle est par défaut à 512Ko, la laisser ainsi.
- Augmentez le **MaxPermSize** pour les applications en ajoutant la ligne suivante à la suite des Java Options.



- Pour des raisons de sécurité, il est essentiel de désactiver le connecteur AJP. A l'emplacement de l'installation de Tomcat, ouvrez **conf/server.xml** et recherchez la ligne suivante. Si vous la trouvez, transformez-la en commentaire et enregistrez le fichier :
 - o `<Connector port="8009" protocol="AJP/1.3" redirectPort="8443"/>`
- L'application manager de Tomcat 9 est paramétrée par défaut pour télécharger des applications de 50 Mo maximum. Ce paramètre peut être modifié dans le fichier web.xml se trouvant dans **webapps/manager/WEB-INF** :

```
<multipart-config>
<!-- 50MB max -->
<max-file-size>52428800 </max-file-size>
<max-request-size>52428800 </max-request-size>
<file-size-threshold> 0</file-size-threshold>
</multipart-config>
```

Pour augmenter la valeur de la taille des fichiers WAR qui peuvent être téléchargés dans Tomcat il suffit d'augmenter les deux valeurs surlignées. Ces valeurs peuvent être doublées.

3.3. Installation sous Linux

- Copier les fichiers du serveur téléchargés précédemment dans le répertoire destiné à accueillir Tomcat.
- Lancez l'extraction des fichiers (vous pouvez lancer la commande suivante :
 - o `sudo tar xzvf /tmp/apache-tomcat-9.0.*tar.gz -C /opt/tomcat --strip-components=1`
- Si votre installation actuelle est régie par un groupe, donnez à ce dernier la propriété du repertoire
 - o `sudo chgrp -R [nom du groupe] /[chemin du répertoire Tomcat]`

- Ensuite, vous devez donner à l'utilisateur Tomcat l'accès au répertoire de **conf** pour voir son contenu et exécuter l'accès au répertoire lui-même :
 - o `sudo chmod -R g+r conf`
 - o `sudo chmod g+x conf`
- Rendez l'utilisateur Tomcat propriétaire des répertoires **webapps**, **work**, **temp** et **logs** :
 - o `sudo chown -R tomcat webapps/ work/ temp/ logs/`
- Si votre installation actuelle est installée en tant que service, vous devez créer un nouveau fichier d'unité pour faire fonctionner Tomcat en tant que service. Ouvrez votre le fichier avec le nom du service concerné dans le répertoire `/etc/systemd/system/` :
- Ajoutez votre configuration enregistrer le serveur en tant que service. Enregistrez et fermez le fichier.
- Ensuite, informez le système que vous avez créé un nouveau fichier en lançant la commande suivante dans la ligne de commande :
 - o `sudo systemctl daemon-reload`
- Les commandes suivantes vous permettront d'exécuter le service Tomcat :
 - o `cd /opt/tomcat/bin`
 - o `sudo ./startup.sh run`
- Suivez la commande ci-dessous pour ajouter un login à votre utilisateur Tomcat et modifier le fichier `tomcat-users.xml` :
 - o `sudo nano /opt/tomcat/conf/tomcat-users.xml`
- Maintenant, définissez l'utilisateur qui peut accéder aux fichiers et ajoutez un nom d'utilisateur et des mots de passe :
 - o `tomcat-users.xml - Admin User`
 - o `<tomcat-users . . .>`
 - o `<tomcat-users . . .>`
 - o `<user username="admin" password="password" roles="manager-gui,admin-gui"/>`
 - o `</tomcat-users>`
- Installez les fichiers **.war** dans le répertoire **webapps** du serveur.
- Redémarrer le service Tomcat
 - o `sudo systemctl restart tomcat`

Sources : <https://www.hostinger.fr/tutoriels/comment-installer-tomcat-sur-ubuntu>

Note : Si vous êtes sur une autre distribution de Linux comme Debian, certaines commandes peuvent varier mais le processus d'installation reste le même.

3.4. Ajout des bibliothèques communes aux applications

La bibliothèque **ojdbc6.jar** doit être présente dans le répertoire : **Tomcat 9.0\lib**

Ces bibliothèques doivent être installées avant le déploiement des applications.

Vous pouvez les récupérer dans l'application **AccueilInterventionAnnuaire.war** ou **Dui.war** ou **PrestatairesExt.war**.

Ces bibliothèques doivent être installées une seule fois et ne doivent pas être supprimées des applications

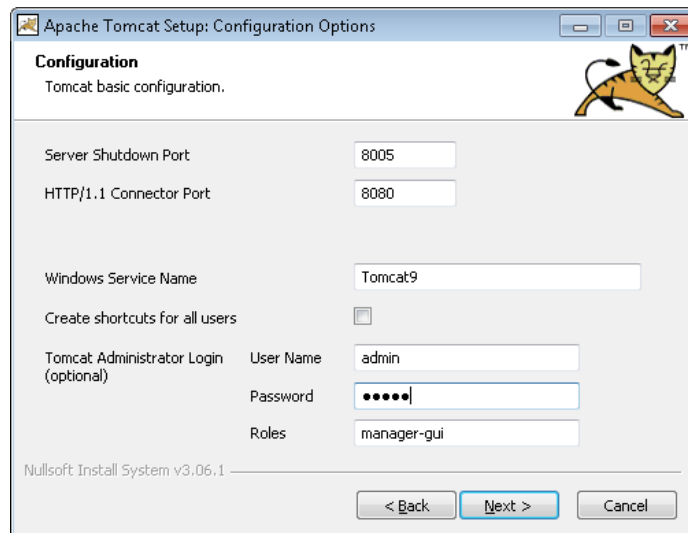
AccueilInterventionAnnuaire.war, **Dui.war**, ou **PrestatairesExt.war**.

Les bibliothèques **lockrealm-tomcat8.jar**, **customrealm-tomcat8.jar**, et **version-reader.jar** (optionnel) doivent également être placées dans le répertoire **Tomcat 9.0\lib**

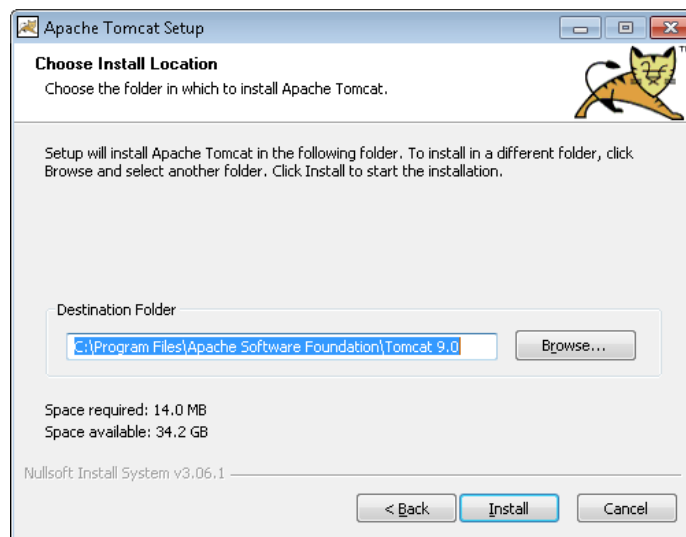
3.5. Installation de plusieurs serveurs Tomcat

Il est possible d'installer plusieurs serveurs Tomcat sur une même machine afin de bien séparer les applications. Pour cela vous pouvez lancer l'installation de Tomcat à nouveau il faut juste modifier le paramétrage des ports ainsi que le répertoire d'installation du Tomcat :

- Le paramétrage par défaut est le suivant



- Les valeurs à remplacer pour un deuxième Tomcat sont :
 - Shutdown Port : 8006
 - Http/11 Connector Port : 8081
 - Windows Service Name : Tomcat92
- Cliquez sur Suivant
- Choisissez un dossier d'installation différent du serveur précédemment installé



- Paramétrez le serveur comme le précédent
- Une fois le serveur Tomcat lancé il est accessible à l'adresse : <http://localhost:8081>

4. Paramétrage SSL

4.1. Création d'un certificat

La création d'un certificat passe par la création d'un **keystore**. Pour cela, il faut exécuter la commande suivante depuis le répertoire d'installation du JDK dans le dossier **bin**:

```
keytool -genkey -keyalg RSA -alias nomAlias -keystore nomKeystore.jks -storepass password -keysize 2048
```

Dans l'exemple ci-dessus, nous générons un fichier `nomKeystore.jks` dans le répertoire où la commande a été exécutée. Dans le **keyStore** se trouve le certificat avec sa clé privée et publique associées. Ce **keystore** est protégé par le mot de passe donné en paramètre. Il suffit à présent d'indiquer le répertoire du keyStore à Tomcat.

4.2. Paramétrage

Dans le cadre d'applications fonctionnant en SSL, un paramétrage de Tomcat est nécessaire. Ouvrez le fichier « **server.xml** » présent dans le dossier « **conf** » du répertoire d'installation de Tomcat.

Cherchez la ligne :

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

Supprimez la redirection vers le port 8443, tel que :

```
<Connector port="8080" protocol="HTTP/1.1"
connectionTimeout="20000" />
```

Cherchez à présent la section dédiée au port 8443 qui est commentée par défaut et décommentez là. Ajoutez les paramètres manquants « **keystoreFile** » et « **keystorePass** » pour obtenir un résultat du type :

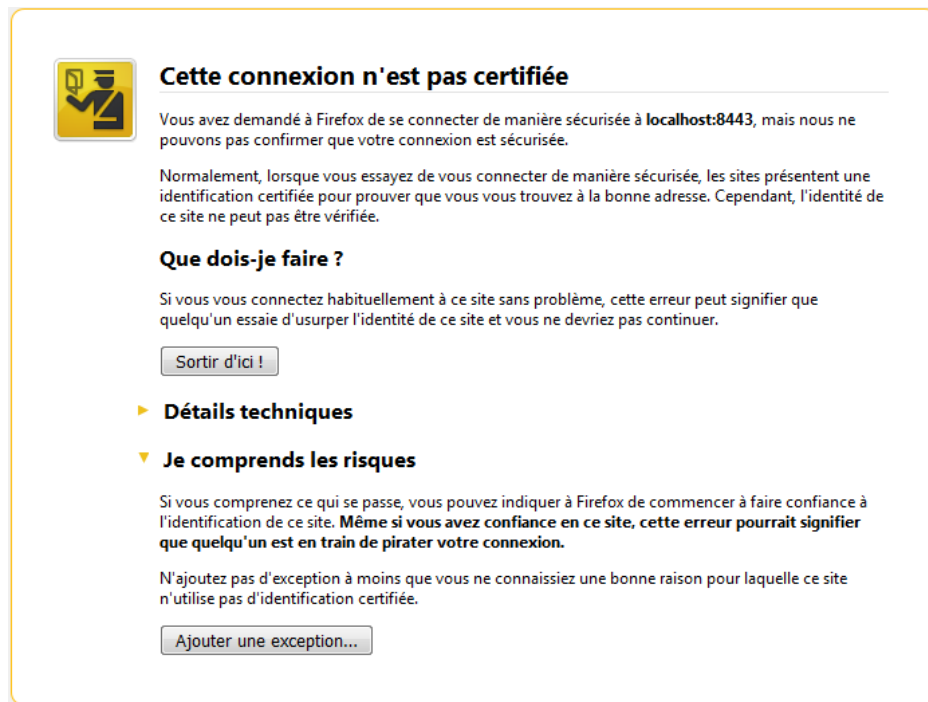
```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
maxThreads="150" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="C:\test\keystore.jks"
keystorePass="motdepas"/>
```

Démarrez le serveur Tomcat, si tout se passe bien vous trouverez les lignes suivantes dans les logs de Tomcat :

```
15 déc. 2014 10:00:22 org.apache.coyote.AbstractProtocol init
INFO: Initializing ProtocolHandler ["http-bio-8080"]
15 déc. 2014 10:00:22 org.apache.coyote.AbstractProtocol init
INFO: Initializing ProtocolHandler ["http-bio-8443"]
```

Vous pourrez alors accéder à Tomcat via le lien suivant : <https://localhost:8443/>

Attention : le navigateur affichera inévitablement le message suivant :



Ce message est tout à fait normal. Le certificat généré précédemment n'est signé par aucune autorité. Pour que cela soit le cas, il est nécessaire de générer une demande de certificat (*Certificate Signing Request : CSR*) afin de la soumettre à une Autorité de certification (Ex : VeriSign) en faisant la commande suivante :

```
keytool -certreq -alias nomAlias -file demande.csr -keystore keystore.jks (le nom d'alias doit être le même que celui utilisé lors de la génération du keystore)
```

Cette commande générera un fichier au format **csr**. Il est à envoyer à l'autorité de certification, qui après quelques jours enverra le certificat signé.

Une fois que le certificat sera signé, il faudra :

- Télécharger le certificat racine de l'autorité
- Importer le certificat racine de l'autorité dans notre keyStore en utilisant la commande suivante :
 - o `keytool -import -alias root -keystore keystore.jks -trustcacerts -file nomDuFichier`
- Importer le nouveau certificat signé par l'autorité de certification en utilisant la commande suivante :
 - o `keytool -import -monAlias signobjis -keystore keystore.jks -trustcacerts -file nomdufichier`