Instructions: Generation of CSR for Tomcat

N.B To generate a Certificate Signing Request (CSR), you need to create a new keystore, only after that you can successfully generate a new CSR.

1. Create a keystore with keytool

To create a new keystore with keytool, you may need to add the java /bin/ directory to your PATH before the keytool command is recognized. When it's done, then enter following comment in keytool:

keytool -genkey -keyalg RSA -keysize 2048 -keystore domain.keystore *(Keystore Directory)*

Now enter the password. (Default – change it)

In the next step, you will be asked for organization details like:

- Enter keystore password: (NOTE remember this for later use)
- Your first and last name- This is the Common Name (Domain Name)
- What is the name of your organizational unit
- Organization Name
- City or Locality
- State or Province
- Two-letter country code for this unit

Confirm the information you filled is correct by entering 'y' or 'yes'.

In this step you will be asked for password confirmation. (Remember the password to create a new CSR)

After this step, your new keystore is created.

Check Certificate keystore:
Check the CSR with the following command:

Windows: keytool –list –keystore C:\Java\apache-tomcat-8.5.9\keystore\tomcat

Linux: keytool –list -keystore /opt/tomcat/keystore/tomcat

Copy and paste the CSR into the TrustFactory complete order to register your CSR for your TrustFactory certificate.

2. Certificate installation procedure in Tomcat

To secure the Tomcat web-server, it is very important for a user to install an SSL certificate on it. Here is a step-by-step, detailed guide on how to install an SSL certificate on it successfully:

Step 1: Download the certificate from the CA

The installation procedure starts with downloading your certificate file from the certificate authority. Now, save it to the directory, where you saved your keystore during CSR-generation.

Login to your TrustFactory account and download the certificate from the certificates tab

Step 2: Install the Root certificate

While installing the certificate to the keystore, you have to enter the exact password that you chose when you generated it. To install the Root certificate file, enter the following code:

keytool -import -trustcacerts -alias root –file RootCertFileName.crt -keystore domain.keystore

Proceed by selecting ‘Yes’, when you receive a message that asks “Certificate already exist in system-wide CA keystore under <>. Do you still want to add it to your own keystore?” After that, you should soon get a confirmation stating a successful certificate installation in keystore

Step 3: Intermediate certificate file Installation

The intermediate certificate file provided by a certificate authority can be installed by typing the following command:

keytool -import -trustcacerts -alias intermediate -file IntermediateCertFileName.crt -keystore domain.keystore

Note:
Depending on the type of certificate purchased, there may be more than one Intermediate certificate in the chain of trust. Please install all intermediates in numerical order until you get to the domain/end entity certificate. Upon successful installation, the following message will appear: ‘Certificate was added to keystore’.

Please click here to determine which chain of trust you have. (Which is the Root? Which is the Intermediate?)

Step 4: Primary certificate file Installation

For installing the primary certificate file for your domain name, you need to type following command:

keytool -import -trustcacerts -alias tomcat -file PrimaryCertFileName.crt -keystore domain.keystore

This would be followed by a message that informs about the successful installation in keystore.

With this step, your keystore shall have all the certificates successfully installed in it. All you need to do to use the keystore file is server configuration.

Step 5: Configure SSL Connector

For Tomcat to accept secure connections, it requires successful configuration of an SSL Connector. Here is how to do it:

By default Tomcat looks into the home directory for your keystore with the file name *.keystore* and keystore password *‘changeit’* and you can change the password and file location. Usually, the home directory in Unix and Linux system is */home/user_name/* and *C:\Documents and Settings\user_name\* on Microsoft© Windows systems

Option – 1

Open Tomcat server.xml file in a text editor. This ‘.xml’ file is usually located in your Tomcat’s home directory folder.

Proceed by locating the connector which you want to secure using the new keystore. Generally, the connector having port 443 or 8443 is used.

Uncomment the SSL connection configuration by removing the comment tags (<!– and –>), if necessary.

In the connector configuration, specify the correct keystore filename and password. Your connector should now look something like this.

To use a JKS (Java Key Store) file: <Connector port=”443″

maxHttpHeaderSize=”8192″ maxThreads=”150″ minSpareThreads=”25″

maxSpareThreads=”75″ enableLookups=”false” disableUploadTimeout=”true”

acceptCount=”100″ scheme=”https” secure=”true” SSLEnabled=”true”

clientAuth=”false” sslProtocol=”TLS” keyAlias=”server”

keystoreFile=”/home/user_name/ your_keystore_file”

keystorePass=”your_keystore_password” />

To use a PFX/P12 (PKCS#12) file: <Connector port=”443″

maxHttpHeaderSize=”8192″ maxThreads=”150″ minSpareThreads=”25″

maxSpareThreads=”75″ enableLookups=”false” disableUploadTimeout=”true”

acceptCount=”100″ scheme=”https” secure=”true” clientAuth=”false”

sslProtocol=”TLS” keystoreFile=”/home/user_name/your_keystore_file”

keystorePass=”your_keystore_password” keystoreType=”PKCS12″/>

Please note, if you are using a version prior to Tomcat 7, you need to change ‘keystorePass’ to ‘keypass’.

Save all the changes in the server.xml file.

Note 1: You may need to comment out the following line:

<Listener className=”org.apache.catalina.core.AprLifecycleListener”

SSLEngine=”on” />like so: <!–<Listener

className=”org.apache.catalina.core.AprLifecycleListener” SSLEngine=”on” />–>

Note 2: You may also need to set SSLEnabled=”true” on the Connector for the SSL connection to function or else only an HTTP connection would be triggered. However, this is often not required.

Now restart Tomcat.