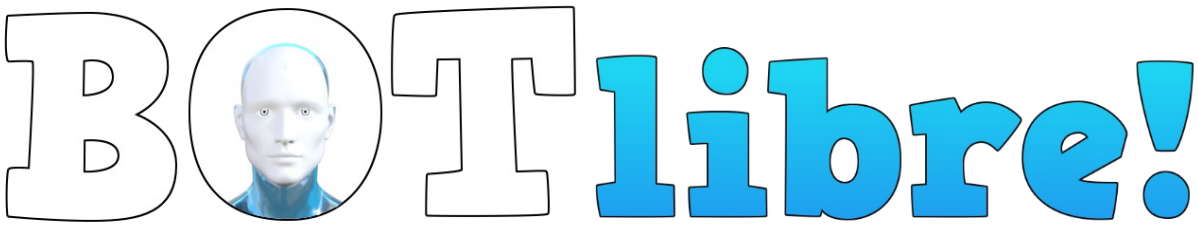


Bot Libre Platform
SSL Guide

Overview

This installation guides gives an overview for configuring Bot Libre Platform with HTTPS support.

HTTPS, Secure Sockets Layer (SSL), and Transport Layer Security (TLS) provide an encrypted web connection to your website. HTTPS is becoming increasingly standard for the web and is now required for most websites.

https://en.wikipedia.org/wiki/Transport_Layer_Security

If you Bot Libre server will support calls from HTTPS websites, apps, or integrate with services such as Facebook or Twilio then HTTPS support is required.

This guide gives an overview of configuring the Tomcat webserver for HTTPS.

Creating your Certificate

To support HTTPS you must purchase an SSL certificate from a trusted service provider such as Go Daddy,

https://godaddy.com/web-security/ssl-certificate

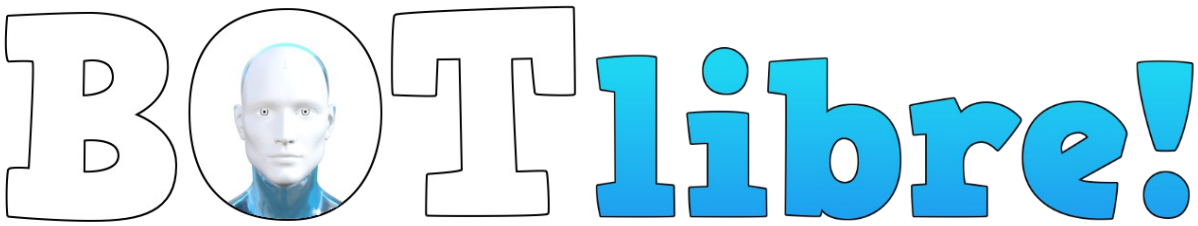
To create a SSL certificate you first need to provide a CSR from the server you have Tomcat installed on. You only need to do this when you are creating your certificate for the first time (or changing servers), not when you update your certificate.

To generate a CSR for Tomcat use the Java keytool executable.

```
cd /opt/jdk1.8.0_131/bin
./keytool -keysize 2048 -genkey -alias tomcat -keyalg RSA -keystore tomcat.keystore
```

This will prompt you for first/last name, do not enter your name, but your domain name i.e. botlibre.com. Then enter your company name and address information.

```
./keytool -certreq -keyalg RSA -keysize 2048 -alias tomcat -file csr.csr -keystore tomcat.keystore
cat csr.csr
```



This will print your CSR text, copy the complete text and paste into your service providers certificate request. It will take some time for it to generate the certificate, and you may need to verify your website or business. Once complete they will give your certificate file, or a zip of your certificate files.

Installing your Certificate

Download your certificate files onto your server and copy them to your Java directory. Normally you will have 2 or 3 .crt files.

```
cp *.crt /opt/jdk1.8.0_77/bin
cd /opt/jdk1.8.0_131/bin
./keytool -import -alias root -keystore tomcat.keystore -trustcacerts -file gd_bundle-g2-g1.crt
./keytool -import -alias intermed -keystore tomcat.keystore -trustcacerts -file gdig2.crt
./keytool -import -alias tomcat -keystore tomcat.keystore -trustcacerts -file
1234xxxx38583365.crt
cp tomcat.keystore /usr/local/tomcat
```

This is an example of the files, your .crt files may have different file names.

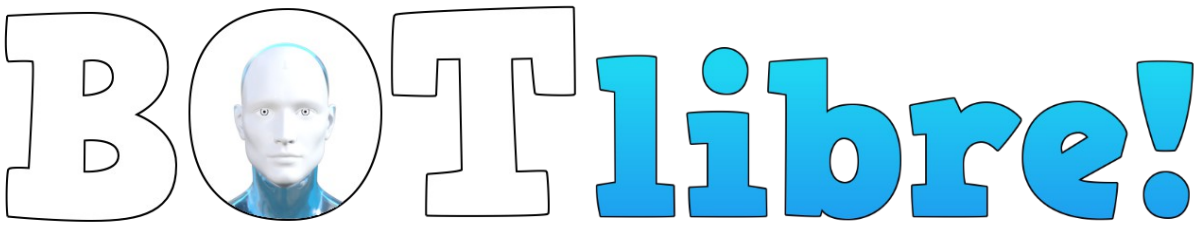
Now that you have your keystore you need to copy it to your Tomcat directory and configure Tomcat to use the keystore and support HTTPS.

Edit your Tomcat /config/server.xml and add the following,

```
<Connector port="443" protocol="org.apache.coyote.http11.Http11NioProtocol"
        maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
        clientAuth="false" sslProtocol="TLS"
        keystoreFile="/usr/local/tomcat8/tomcat.keystore" keystorePass="yourpassword"
        URIEncoding="UTF-8"/>
```

Also ensure your default connector has the redirect port set,

```
<Connector port="80" protocol="HTTP/1.1"
        connectionTimeout="20000"
```



redirectPort="443"

URIEncoding="UTF-8"/>

Troubleshooting

Some services may not provide 3 .crt files, they may provide only 1, or may provide another format, or may not request a CSR. It is always easiest to follow the above guide, but sometimes you may need to to a more advanced installation. You can normally find instructions or information on what you are trying to do on the web by searching for “Tomcat” and the error or issue you are having.

Here are some other useful commands,

>> remove intermediate and root certificate (in case you need to reset your keystore)

./keytool -delete -alias intermed -keystore tomcat.keystore

./keytool -delete -alias root -keystore tomcat.keystore

>> Install with no CRS, key

>> See https://stackoverflow.com/questions/53439545/how-to-install-godaddy-ssl-certificates-in-tomcat-without-csr

openssl pkcs12 -export -in STAR_mycertificate.crt -inkey mykey.key -out cert_and_key.p12 -name tomcat -CAfile gd_bundle-g2-g1.crt -caname root

keytool -importkeystore -srckeystore cert_and_key.p12 -srcstoretype PKCS12 -alias tomcat -keystore tomcat.keystore

keytool -import -trustcacerts -alias root -file $certdir/gd_bundle-g2-g1.crt -noprompt -keystore tomcat.keystore