Apache Tomcat

10.0.27

Integration Guide

CryptoServer JCE

4.50.0.1

utimaco®

Imprint

Copyright 2026	Utimaco IS GmbH
	Krefelder Straße 220
	52070 Aachen
	Germany
Phone	AMERICAS +1-844-UTIMACO (+1 844-884-6226)
	EMEA +49 800-627-3081
	APAC +81 800-919-1301
Internet	https://support.hsm.utimaco.com/
e-mail	support@utimaco.com
Document Version	1.0.0
Date	2026-02-24
Status	PUBLISHED
Document No.	IG-2026-0002

Table of Contents

1 Introduction

This guide is part of the information and support provided by Utimaco. Additional documentation produced to support your Utimaco SecurityServer product can be found in the document directory of the Utimaco SecurityServer product bundle.

All Utimaco SecurityServer product documentation is available from Utimaco's website at https://utimaco.com/.

1.1 About This Guide

This guide describes how to integrate an Utimaco CryptoServer Hardware Security Module (HSM) with Apache Tomcat. Utimaco HSM securely stores the private key used by Apache Tomcat for SSL and offload the cryptographic operations to the HSM.

1.1.1 Target Audience for This Guide

This guide is intended for Apache Tomcat and Utimaco HSM administrators.

1.1.2 Abbreviations

The following abbreviations are used in this guide:

Abbreviation	Meaning
CA	Certificate Authority
CD	Compact Disc
CSADM	CryptoServer Command-line Administration Tool
CSR	Certificate Signing Request
GUI	Graphical User Interface

Abbreviation	Meaning
HSM	Hardware Security Module
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
JCE	Java Cryptographic Extension
JDK	Java Development Kit
LAN	Local Area Network
MBK	Master Backup Key
P11CAT	PKCS#11 CryptoServer Administration Tool
PCIe	PCI Express Interface
PIN	Personal Identification Number
RSA	Rivest-Shamir-Adleman
SSL	Secure Socket Layer
SO	Security Officer
URL	Uniform Resource Locator

Table 1: Abbreviations

1.1.3 Document Conventions

The following conventions are used in this guide:

Convention	Use	Example
Bold	Items of the Graphical User Interface (GUI), e.g., menu options	Press **OK**
`Monospaced`	Code that is given for explanation or as an example, file paths	`chsm-create`
Italic	References and important terms	See *Sample Chapter* in the *CryptoServer - Sample Manual*

Table 2: Document Conventions

We use special icons to highlight the most important notes and information.

Here you will find important safety information that should be followed.

Here you will find additional notes or supplementary information.

This message indicates the expected result after the successful execution of an instruction.

2 Overview

2.1 Apache Tomcat

Apache Tomcat software powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. The Apache Tomcat software is an open-source implementation of the Jakarta Servlet, Jakarta Server Pages, Jakarta

Expression Language, Jakarta WebSocket, Jakarta Annotations and Jakarta

Authentication specifications. These specifications are part of the Jakarta EE platform.

2.2 Utimaco SecurityServer HSM

SecurityServer is a hardware security module developed by Utimaco IS GmbH. SecurityServer is a physically protected specialized computer unit designed to perform sensitive cryptographic tasks and to securely manage as well as store cryptographic keys and data. It can be used as a universal, independent security component for heterogeneous computer systems.

3 Integration Requirements and Prerequisites

Ensure that the system environment you will be using, meets the following hardware and software requirements.

This guide assumes that the user has already installed and configured required Software.

3.1 Tested Versions

The integrations that have been successfully tested with the Utimaco HSM and Apache Tomcat.

Operating System	Apache Tomcat Version	Java Version	Utimaco Security Server Version	Utimaco HSM
Rhel 8	10.0.27	Java 8 Java 11	SecurityServer V4.50.0.1	CryptoServer CSeSeries/Se-Series

Table 3: Tested Versions

3.2 Software Requirements

Software	Requirements
HSM Interfaces	CryptoServer JCE
OpenJDK 8	1.8.0_232-b09
OpenJDK 11	11.0.2
Host OS	Redhat 8 and above
HSM software	Utimaco Crypto Server Software 4.50.0.1

Software	Requirements
cxitool	cxitool from product package Utimaco SecurityServer 4.50.0.1
Apache Tomcat	Version 10.0.27

Table 4: Software Requirements

3.3 Hardware Requirements

Hardware	Requirements
Utimaco LAN HSM	CryptoServer CSe-Series/Se-Series LAN with firmware SecurityServer 4.50.0.1 or higher
Utimaco PCI-e HSM	CryptoServer CSe-Series/Se-Series PCI-e with firmware SecurityServer 4.50.0.1or higher

Table 5: Hardware Requirements

Set up an account on the Utimaco support portal and request download access at https://support.hsm.utimaco.com/.

3.4 Prerequisites

Before you begin, please ensure that you have installed/set up:

- CryptoServer is set up and configured. Refer to the CryptoServer documentation to se tup the HSM
- CryptoServer Default Admin should be replaced with a new admin user
- MBK must be created and stored on each HSM. Refer to the CryptoServer documentation to set up the MBK

- Operating system listed in Tested Versions
- SecurityServer listed in Tested Versions
- Familiarize yourself with the Apache Tomcat documents and setup process
- Admin user for installing software on Apache Tomcat server
- Allow port 443 through Firewall

4 Installing and Configuring Utimaco SecurityServer

4.1 Download and Install Utimaco Software

If you have not already done so, please create and request an Utimaco Support Portal Account. This allows you to download the software components for this installation.

1. Copy the downloaded software at the appropriate location on the Apache Tomcat Server
2. Create an `utimaco` folder under the `/opt` directory and further create 2 directories `/opt/utimaco/bin and /opt/utimaco/lib`.

›_ Console

```
# mkdir -p /opt/utimaco/bin

# mkdir /opt/utimaco/lib
```

3. Copy the file `CryptoServerJCE.jar` to the `/opt/utimaco/lib` directory.

›_ Console

```
# cp ~/path_to_application_folder/Linux/x86-

64/Crypto_APIs/JCE/lib/CryptoServerJCE.jar/opt/utimaco/lib
```

4. Copy the csadm, `ADMIN.key` and cxitool files from Utimaco CryptoServer software to the `/opt/utimaco/bin` directory and make both the files executable.

›_ Console

```
# cd ~/path_to_application_folder

# cp csadm ADMIN.key cxitool/opt/utimaco/bin

# chmod +x /opt/utimaco/bin/csadm/opt/utimaco/bin/cxitool
```

4.2 CryptoServer JCE Configuration

1. Locate the Utimaco JCE configuration file in your SecurityServer directory, `Linux/x86-64/Crypto_APIs/JCE/sample/CryptoServer.cfg`.
2. Create a non-root user and set its password.

›_ Console

```
# useradd tomcat

# passwd tomcat
```

3. Copy the Utimaco JCE configuration file `CryptoServer.cfg` to the user's home directory.

›_ Console

```
# cd <installation_directory>/Software/Linux/x86-64/Crypto_APIs/

JCE/sample/CryptoServer.cfg

# cp CryptoServer.cfg $home
```

4. Create one Cryptographic User with CXI group.

›_ Console

```
#  /opt/utimaco/bin/csadm Dev=3001@127.0.0.1

LogonSign=ADMIN,/opt/utimaco/bin/ADMIN.key

AddUser=<user_name>,00000002{CXI_GROUP=<cxi_group_name>},hmacpwd,<PIN>
```

Figure 1 : User creation with csadm

5. Edit the `$home/CryptoServer.cfg` file and make the appropriate changes to the file.

CryptoServer.cfg

```
# Configuration File for JCE CryptoServer Provider

LogFile = /tmp/CryptoServerJCE.log

LogLevel = 1

LogSize = 10000

Device = <HSM_IP>


ConnectionTimeout = 3000

Timeout = 30000

#EndSessionOnShutdown = 1

KeepSessionAlive = 1


#Provide cryptographic username with cxi_group name

DefaultUser = <Cryptographic_User_Name>

KeyGroup = <CXI_Group_Name>


StoreKeysExternal = false

#KeyStorePath = C:/<user directory>/JCE.sdb
```

For more information regarding the commands and command parameters please check

the CryptoServer documentation. The device may be a CryptoServer (PCIe or LAN) device.

The device line will follow one of these patterns, based on the HSM form-factor:

`Device = 288@<HSM IP address>` Hardware (LAN) HSM

OR

`Device = /dev/cs2.0` Hardware (PC Ie) HSM

To make your testing easier, it would be good to enable the Cryptoserver JCE log file. That can be enabled by editing the `Logging` Loglevel. Set the `LogFile` and Logging `Loglevel` to 1. For testing you may want to increase it to 4. The added `LogFile` points to a file. If you encounter problems, check the log file named `CryptoServerJCE.log` in the LogFile defined file. When you are done testing, you should change Logging to `1` or `2`. This limits the logging to only critical and important messages.

6. Obtain the below jurisdiction (unlimited strength) policy files from Oracle for your country and Java version:
   `US_export_policy.jar`
   `local_policy.jar`

The unlimited policy files are required only for JDK 8 updates earlier than 8u161. On those versions and later, the stronger cryptographic algorithms are available by default.

8. Copy these jurisdiction policy files into the directory <java-home>/lib/security.

›_ Console

```
# cp US_export_policy.jar <java_home>/lib/security

# cp local_policy.jar <java_home>/lib/security
```

5 Apache Tomcat Download and Installation

To install Apache Tomcat:

1. (Optional) It is recommended to update the system with the latest security patch.
2. Install OpenJDK.

For Java 8:

```
›_ Console

# dnf install java-1.8.0-openjdk java-1.8.0-openjdk-devel
```

For Java 11:

```
›_ Console

# dnf -y install java-11-openjdk java-11-openjdk-devel
```

3. Download Tomcat 10.

```
›_ Console

# wget https://dlcdn.apache.org/tomcat/tomcat-10/v10.0.27/bin/apache-tomcat-

10.0.27.tar.gz
```

4. Create a directory.

```
›_ Console
```

```
# mkdir -p /opt/tomcat
```

5. Extract the archived file to `/opt/tomcat`.

›_ Console

```
# tar -xvf apache-tomcat-10.0.27.tar.gz -C /opt/tomcat --strip-components=1
```

6. Change the ownership of the `/opt/directory` to tomcat user.

›_ Console

```
# chown -R tomcat:tomcat /opt/tomcat
```

7. Set executable permissions to scripts.

›_ Console

```
# chmod +x /opt/tomcat/bin/*.sh
```

8. Create Apache Tomcat Systemd file `/etc/systemd/system/tomcat.service` to manage the Tomcat service through `systemctl` and add the lines below.

›_ Console

```
[Unit]

Description=Apache Tomcat Web Application Container

Wants=network.target

After=network.target


[Service]

Type=forking


Environment=JAVA_HOME=/usr/lib/jvm/jre


Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid

Environment=CATALINA_HOME=/opt/tomcat

Environment='CATALINA_OPTS=-Xms512M -Xmx1G -Djava.net.preferIPv4Stack=true'
Environment='JAVA_OPTS=-Djava.awt.headless=true'


ExecStart=/opt/tomcat/bin/startup.sh

ExecStop=/opt/tomcat/bin/shutdown.sh

SuccessExitStatus=143


User=tomcat

Group=tomcat

UMask=0007

RestartSec=10

Restart=always


[Install]

WantedBy=multi-user.target
```

Change the values according to your system configuration.

```
[root@tomcat ~]# vim /etc/systemd/system/tomcat.service
[Unit]
Description=Apache Tomcat Web Application Container
Wants=network.target
After=network.target

[Service]
Type=forking

Environment=JAVA_HOME=/usr/lib/jvm/jre

Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment='CATALINA_OPTS=-Xms512M -Xmx1G -Djava.net.preferIPv4Stack=true'
Environment='JAVA_OPTS=-Djava.awt.headless=true'

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh
SuccessExitStatus=143

User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always

[Install]
WantedBy=multi-user.target
```

Figure 2 : /etc/systemd/system/tomcat.service file output

9. Reload the daemon.

›_ Console

```
# systemctl daemon-reload
```

10. Start and enable the Tomcat service.

›_ Console

```
# systemctl start tomcat

# systemctl enable tomcat
```

11. Confirm that the Tomcat service is running.

›_ Console

```
# systemctl status tomcat.service
```

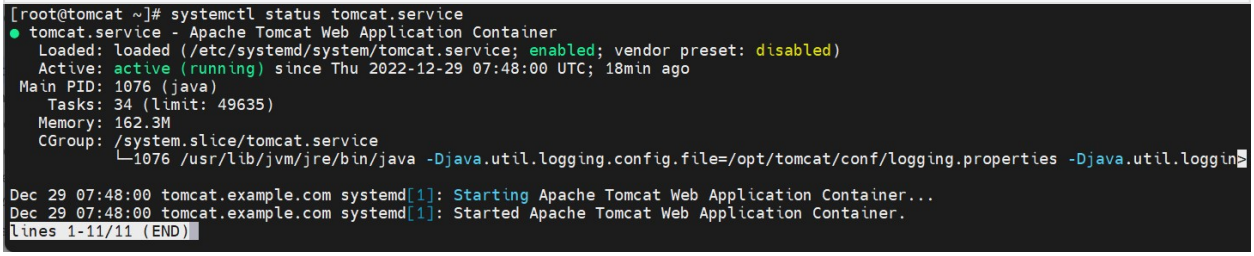

Figure 3 : Tomcat Service Status

12. Open `http://<apache_tomcat_-server-ip>:8080` in any web browser and verify if Apache Tomcat page is visible.

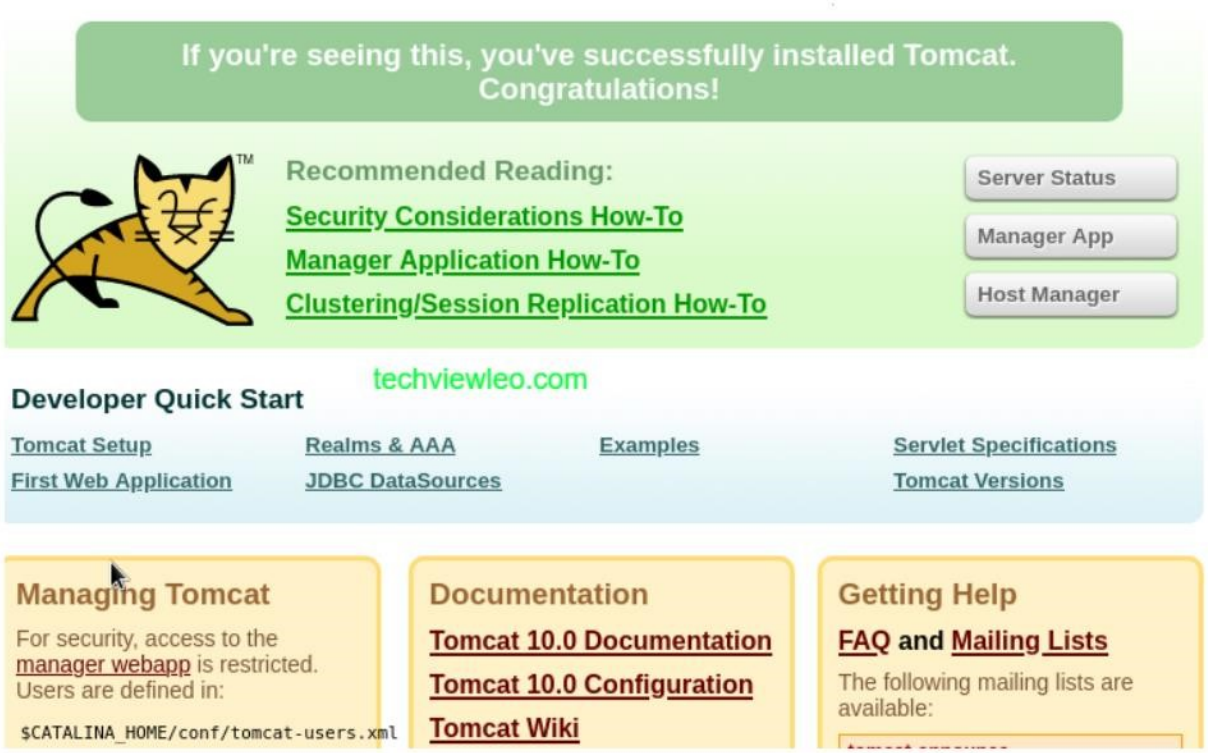

Figure 4 : Browser Output over page 8080

6 Java Configuration to use Utimaco HSM

6.1 Update OpenJDK 8 to use Utimaco HSM

1. Copy the CryptoServerJCE.jar file to `JAVA_HOME/jre/lib/ext/`.

›_ Console

```
# cp /opt/utimaco/lib/CryptoServerJCE.jar /usr/lib/jvm/java-1.8.0-
openjdk1.8.0.232.b09-2.el8_1.x86_64/jre/lib/ext
```

2. Go to the `<JDK_Installation_directory>/jre/lib/security` directory.

›_ Console

```
# cd /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.232.b092.el8_1.x86_64/jre/lib/
security/
```

3. Edit the `java.security` configuration file to add CryptoServerJCE provider as highlighted below.

›_ Console

```
security.provider.1=sun.security.provider.Sun
security.provider.2=sun.security.rsa.SunRsaSign
security.provider.3=sun.security.ec.SunEC

security.provider.4=com.sun.net.ssl.internal.ssl.Provider
security.provider.5=com.sun.crypto.provider.SunJCE
security.provider.6=sun.security.jgss.SunProvider
security.provider.7=com.sun.security.sasl.Provider
security.provider.8=org.jcp.xml.dsig.internal.dom.XMLDSigRI
security.provider.9=sun.security.smartcardio.SunPCSC
security.provider.10=CryptoServerJCE.CryptoServerProvider

<home_directory>/CryptoServer.cfg
```

Specify the correct provider number and path for the CryptoServerJCE Provider.

6.2 Update OpenJDK 11 to use Utimaco HSM

1. Copy the `CryptoServerJCE.jar` file to the `/opt/tomcat/bin` directory.

›_ Console

```
# cp /opt/utimcao/lib/CryptoServerJCE.jar /opt/tomcat/bin/
```

2. Go to the `<JDK_Installation_directory> conf/security` directory.

›_ Console

```
# cd /usr/lib/jvm/java-11-openjdk-11.0.2.7-2.el8.x86_64/conf/security/
```

1. Edit the `java.security` configuration file to add the **CryptoServerJCE** provider.

›_ Console

```
security.provider.1=SUN security.provider.2=SunRsaSign
security.provider.3=SunEC security.provider.4=SunJSSE
security.provider.5=SunJCE security.provider.6=SunJGSS
security.provider.7=SunSASL security.provider.8=XMLDSig
security.provider.9=SunPCSC security.provider.10=JdkLDAP
security.provider.11=JdkSASL security.provider.12=SunPKCS11

security.provider.13=CryptoServerJCE.CryptoServerProvider
```

Specify the correct provider number for the CryptoServerJCE Provider. Apache Tomcat requires the `CryptoServer.cfg` configuration file to be present in the home directory of the user.

7 Integrating Apache Tomcat and Utimaco HSM

7.1 Generating CA signed SSL certificate

7.1.1 For OpenJDK 8 with RSA Key

1. Generate an RSA key pair on the Utimaco HSM.

›_ Console

```
# keytool -genkey -keyalg RSA -keysize 2048 -keystore NONE -storetype
CryptoServer -storepass 12345678 -providername CryptoServer -alias
tomcatsslkey
```

Provide information when prompted.

Here:

- `RSA` is the key algorithm
- `2048` is the key size
- `NONE` is the key store for HSM
- `CryptoServer` is the store type
- `12345678` is the PIN
- `CryptoServer` is the provider name
- `tomcatsslkey` is the key name that will be generated on the Utimaco HSM

```
[root@tomcat ~]# keytool -genkey -keyalg RSA -keysize 2048 -keystore NONE -storetype CryptoServer -storepass 12345678 -provide
rname CryptoServer -alias tomcatsslkey
What is your first and last name?
  [Unknown]:  test test
What is the name of your organizational unit?
  [Unknown]:  security
What is the name of your organization?
  [Unknown]:  Utimaco
What is the name of your City or Locality?
  [Unknown]:  Pune
What is the name of your State or Province?
  [Unknown]:  MH
What is the two-letter country code for this unit?
  [Unknown]:  IN
Is CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN correct?
  [no]:  yes

Enter key password for <tomcatsslkey>
        (RETURN if same as keystore password):
[root@tomcat ~]#
```

Figure 5 : Key generation using keytool command output

2. Verify that the keys have been generated with the `keytool` command.

›_ Console

```
# keytool -list -keystore NONE -storetype CryptoServer -providername

CryptoServer -storepass 12345678 -v
```

Here:

- `NONE` is the key store for HSM
- `CryptoServer` is the store type
- `12345678` is the PIN
- `CryptoServer` is the provider name

```
[root@tomcat ~]# keytool -list -keystore NONE -storetype CryptoServer -providername CryptoServer -storepass 12345678 -v
Keystore type: CRYPTOSERVER
Keystore provider: CryptoServer

Your keystore contains 1 entry

Alias name: tomcatsslkey
Creation date: Jan 10, 2023
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Serial number: 25a483b8
Valid from: Tue Jan 10 12:41:46 UTC 2023 until: Mon Apr 10 12:41:46 UTC 2023
Certificate fingerprints:
         MD5:  3E:2F:F8:97:F4:5F:88:90:77:4D:5C:6E:F1:48:3C:D2
         SHA1: B1:85:36:9F:69:FD:A2:64:7A:D8:0B:54:48:A2:C1:BE:19:21:A5:93
         SHA256: 2F:4A:D2:D9:C9:CC:2C:B4:71:37:8C:20:39:2A:F4:61:E0:89:32:E4:11:BD:C0:0D:4C:F2:A9:35:93:2D:CC:15
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: A3 29 3C E5 1C 7D 10 40   B2 78 C4 AE 93 3C 5B CF  .)<....@.x...<[.
0010: 09 70 22 F8                                        .p".
]
]



*******************************************
*******************************************


[root@tomcat ~]#
```

Figure 6 : Keytool list output

3. List the keys using the cxitool.

›_ Console

/opt/utimaco/bin/cxitool Dev=3001@127.0.0.1 Logonpass=tomcat,12345678 Group=JCE ListKeys

```
[root@tomcat ~]# /opt/utimaco/bin/cxitool Dev=3001@127.0.0.1 Logonpass=tomcat,12345678 Group=JCE ListKeys

idx algo  size type     group                    name                             spec
-------------------------------------------------------------------------------------
1   RSA   2048 pub+prv  JCE                      tomcatsslkey

[root@tomcat ~]#
```

Figure 7 : List keys output using the cxitool

4. Generate a CSR using the `keytool` command.

›_ Console

```
# keytool -certreq -alias tomcatsslkey -file test.csr -storetype
CryptoServer -keystore NONE -v
```

Provide the user PIN when you're prompted for the key store password.

Here:

- `tomcatsslkey` is the key name
- `CryptoServer` is the store type
- `test.csr` is the CSR file name that will be generated
- `NONE` is the keystore for HSM

5. Get this CSR signed by the CA.
6. Copy the signed certificate along with the root CA certificate chain to the Tomcat server.
7. Import the signed certificate chain reply using the command below.

›_ Console

```
# keytool -importcert -trustcacerts -alias tomcatsslkey -file

/root/test_test.pem -storetype CryptoServer -keystore NONE -providername
CryptoServer -storepass 12345678
```

```
[root@tomcat ~]# keytool -importcert -trustcacerts -alias tomcatsslkey -file /root/test_test.pem -storetype CryptoServer -keystore NONE -p
rovidername CryptoServer -storepass 12345678

Top-level certificate in reply:

Owner: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Issuer: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Serial number: 40f8f17a48d0bcc3
Valid from: Tue Nov 29 05:36:00 UTC 2022 until: Mon Nov 29 05:36:00 UTC 2032
Certificate fingerprints:
         MD5:  80:E7:CE:91:84:D6:E9:D9:03:53:DB:F3:11:84:F3:34
         SHA1: D9:BE:FA:00:CB:52:0E:7F:7B:82:CE:D3:54:02:BB:1F:43:21:B8:A1
         SHA256: 58:E9:C6:A3:12:0D:A9:3A:97:E8:0D:03:06:9B:89:0F:05:E6:EB:1F:46:1C:E8:B1:B6:DF:DE:3E:4D:0B:EC:64
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 4096-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.16.840.1.113730.1.13 Criticality=false
0000: 16 0E 49 6E 66 6F 73 65   63 20 4C 61 62 20 43 41  ..Infosec Lab CA


#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  Key_CertSign
  Crl_Sign
]

#4: ObjectId: 2.16.840.1.113730.1.1 Criticality=false
```

Figure 8 : Import the user certificate into the key store

```
#4: ObjectId: 2.16.840.1.113730.1.1 Criticality=false
NetscapeCertType [
   SSL CA
   S/MIME CA
   Object Signing CA]

#5: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 85 42 28 42 84 18 55 2C   73 AA 74 DC 23 EE 74 7A  .B(B..U,s.t.#.tz
0010: 00 FE 2E DC                                        ....
]
]


... is not trusted. Install reply anyway? [no]:  yes
Certificate reply was installed in keystore
[root@tomcat ~]#
```

Signed certificates must also contain the certificate chain.

8. Verify that the `keytool` command shows the signed certificate.

›_ Console

```
# keytool -list -keystore NONE -storetype CryptoServer -providername

CryptoServer -storepass 12345678 -v
```

Here:

- `NONE` is the keystore for the HSM
- `CryptoServer` is the store type
- `12345678` is the PIN
- `CryptoServer` is the provider name

```
[root@tomcat ~]# keytool -list -keystore NONE -storetype CryptoServer -providername CryptoServer -storepass 12345678 -v
Keystore type: CRYPTOSERVER
Keystore provider: CryptoServer

Your keystore contains 1 entry

Alias name: tomcatsslkey
Creation date: Jan 10, 2023
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=test test, OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Serial number: 76a22ec44822080
Valid from: Tue Jan 10 12:52:00 UTC 2023 until: Wed Jan 10 12:52:00 UTC 2024
Certificate fingerprints:
         MD5:  6A:B5:12:13:F9:E5:9A:F2:43:27:E8:B2:49:30:13:3C
         SHA1: 0F:DC:57:AD:5D:18:16:A9:A7:47:9F:ED:07:D3:EB:12:AC:9D:81:F7
         SHA256: 76:A5:D3:85:6B:0D:70:C7:95:59:74:54:4F:AB:59:26:4E:C6:E3:35:AB:5C:8D:3E:6C:CF:13:DD:B0:67:64:62
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: A3 29 3C E5 1C 7D 10 40   B2 78 C4 AE 93 3C 5B CF  .)<....@.x...<[.
0010: 09 70 22 F8                                        .p".
]
]



*******************************************
*******************************************


[root@tomcat ~]#
```

Figure 9 : Keytool list output

7.1.2 For OpenJDK 8 with EC Key

HR 1. Generate an EC key pair on the Utimaco HSM.

›_ Console

```
# keytool -genkey -keyalg EC -keystore NONE -storetype CryptoServer
-storepass

12345678 -providername CryptoServer -alias tomcatsslECKey
```

Provide information when prompted.

Here:

- `EC` is the key algorithm
- `NONE` is the keystore for HSM
- `CryptoServer` is the store type
- `12345678` is the PIN
- `CryptoServer` is the provider name
- `tomcatsslECKey` is the key name that will be generated on the Utimaco HSM

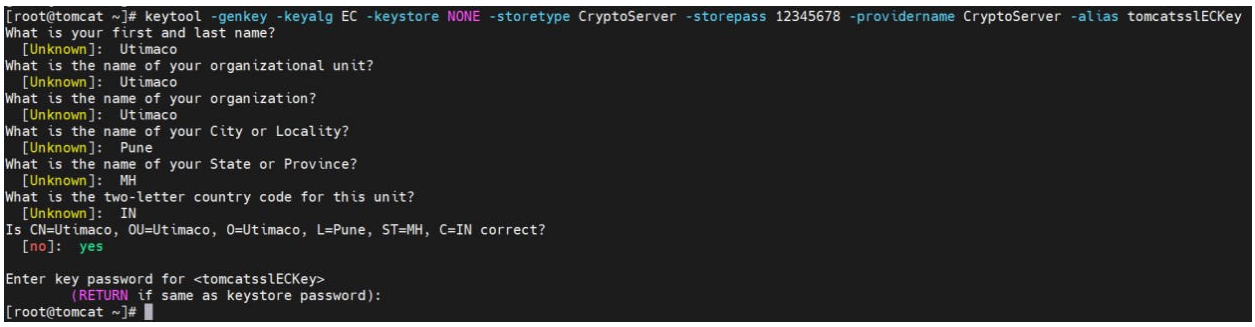

Figure 10 : Key generation using keytool command output

2. Verify that the keys have been generated using the `keytool` command.

›_ Console

```
# keytool -list -keystore NONE -storetype CryptoServer -providername

CryptoServer -storepass 12345678 -v
```

Here:

- `NONE` is the keystore for HSM
- `CryptoServer` is the store type
- `12345678` is the slot PIN
- `CryptoServer` is the provider name

```
[root@tomcat ~]# keytool -list -keystore NONE -storetype CryptoServer -providername CryptoServer -storepass 12345678 -v
Keystore type: CRYPTOSERVER
Keystore provider: CryptoServer

Your keystore contains 1 entry

Alias name: tomcatsslECKey
Creation date: Jan 13, 2023
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Utimaco, OU=Utimaco, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN=Utimaco, OU=Utimaco, O=Utimaco, L=Pune, ST=MH, C=IN
Serial number: 650fa648
Valid from: Fri Jan 13 10:17:09 UTC 2023 until: Thu Apr 13 10:17:09 UTC 2023
Certificate fingerprints:
         MD5:  FB:E1:77:2D:D2:E6:02:6B:25:EA:B6:62:CA:90:C7:EB
         SHA1: 9A:15:18:8A:52:FB:E1:6F:F2:6E:04:92:2E:8F:31:14:CB:18:30:1E
         SHA256: 25:D3:10:1F:BC:1D:78:3C:D6:08:72:A2:4F:97:A1:E3:41:C6:98:B4:20:BE:E6:38:B4:F0:97:9F:E2:4A:74:FC
Signature algorithm name: SHA256withECDSA
Subject Public Key Algorithm: 256-bit EC key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 62 75 38 B0 D9 E5 27 5F   35 7E 9F 9D 85 50 33 FF  bu8...'_5....P3.
0010: 52 9F 76 4F                                        R.vO
]
]



*******************************************
*******************************************


[root@tomcat ~]#
```

Figure 11 : Keytool list output

3. List the keys using cxitool.

›_ Console

```
 # /opt/utimaco/bin/cxitool Dev=3001@127.0.0.1 Logonpass=tomcat,12345678
Group=JCE ListKeys
```

```
[root@tomcat ~]# /opt/utimaco/bin/cxitool Dev=3001@127.0.0.1 Logonpass=tomcat,12345678 Group=JCE ListKeys

idx algo  size type     group                   name                            spec
-------------------------------------------------------------------------------------
1   ECDSA 256  pub+prv  JCE                     tomcatsslECKey

[root@tomcat ~]#
```

Figure 12 : List keys output using cxitool

4. Generate a CSR using the `keytool` command.

›_ Console

```
# keytool -certreq -alias tomcatsslkey -file test.csr -storetype CryptoServer -keystore NONE -v
```

```
[root@tomcat ~]# keytool -certreq -alias tomcatsslECKey -file test.csr -storetype CryptoServer -keystore NONE -v
Enter keystore password:
Certification request stored in file <test.csr>
Submit this to your CA
[root@tomcat ~]#
```

Figure 13 : keytool -certreq output

Provide user PIN when prompted for keystore password.

Here:

- `tomcatsslECKey` is the key name
- `NONE` is the keystore for HSM
- `CryptoServer` is the store type
- `test.csr` is the CSR file name that will be generated

5. Get this CSR signed by CA.
6. Copy the signed certificate along with the root CA certificate chain to the Tomcat server.
7. Import the signed certificate chain reply using the command below.

›_ Console

```
# keytool -importcert -trustcacerts -alias tomcatsslECKey -file /root/
Utimaco_Signedcert.p7b -storetype CryptoServer -keystore

NONE -providername CryptoServer -storepass 12345678
```

```
[root@tomcat ~]# keytool -importcert -trustcacerts -alias tomcatsslECKey -file /root/Utimaco_Signedcert.p7b -storetype CryptoServer -keystore
 NONE -providername CryptoServer -storepass 12345678

Top-level certificate in reply:

Owner: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Issuer: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Serial number: 40f8f17a48d0bcc3
Valid from: Tue Nov 29 05:36:00 UTC 2022 until: Mon Nov 29 05:36:00 UTC 2032
Certificate fingerprints:
         MD5:  80:E7:CE:91:84:D6:E9:D9:03:53:DB:F3:11:84:F3:34
         SHA1: D9:BE:FA:00:CB:52:0E:7F:7B:82:CE:D3:54:02:BB:1F:43:21:B8:A1
         SHA256: 58:E9:C6:A3:12:0D:A9:3A:97:E8:0D:03:06:9B:89:0F:05:E6:EB:1F:46:1C:E8:B1:B6:DF:DE:3E:4D:0B:EC:64
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 4096-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.16.840.1.113730.1.13 Criticality=false
0000: 16 0E 49 6E 66 6F 73 65   63 20 4C 61 62 20 43 41  ..Infosec Lab CA


#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  Key_CertSign
  Crl_Sign
]
```

Figure 14 : Import user certificate into key store

```
#4: ObjectId: 2.16.840.1.113730.1.1 Criticality=false
NetscapeCertType [
   SSL CA
   S/MIME CA
   Object Signing CA]

#5: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 85 42 28 42 84 18 55 2C   73 AA 74 DC 23 EE 74 7A  .B(B..U,s.t.#.tz
0010: 00 FE 2E DC                                        ....
]
]


... is not trusted. Install reply anyway? [no]:  yes
Certificate reply was installed in keystore
[root@tomcat ~]#
```

Signed certificates must also contain the certificate chain.

8. Verify that the keytool command shows the signed certificate

›_ Console

```
# keytool -list -keystore NONE -storetype CryptoServer -providername

CryptoServer -storepass 12345678 -v
```

Here:

- `NONE` is the key store for the HSM
- `CryptoServer` is the store type
- `12345678` is the slot PIN
- `CryptoServer` is the provider name

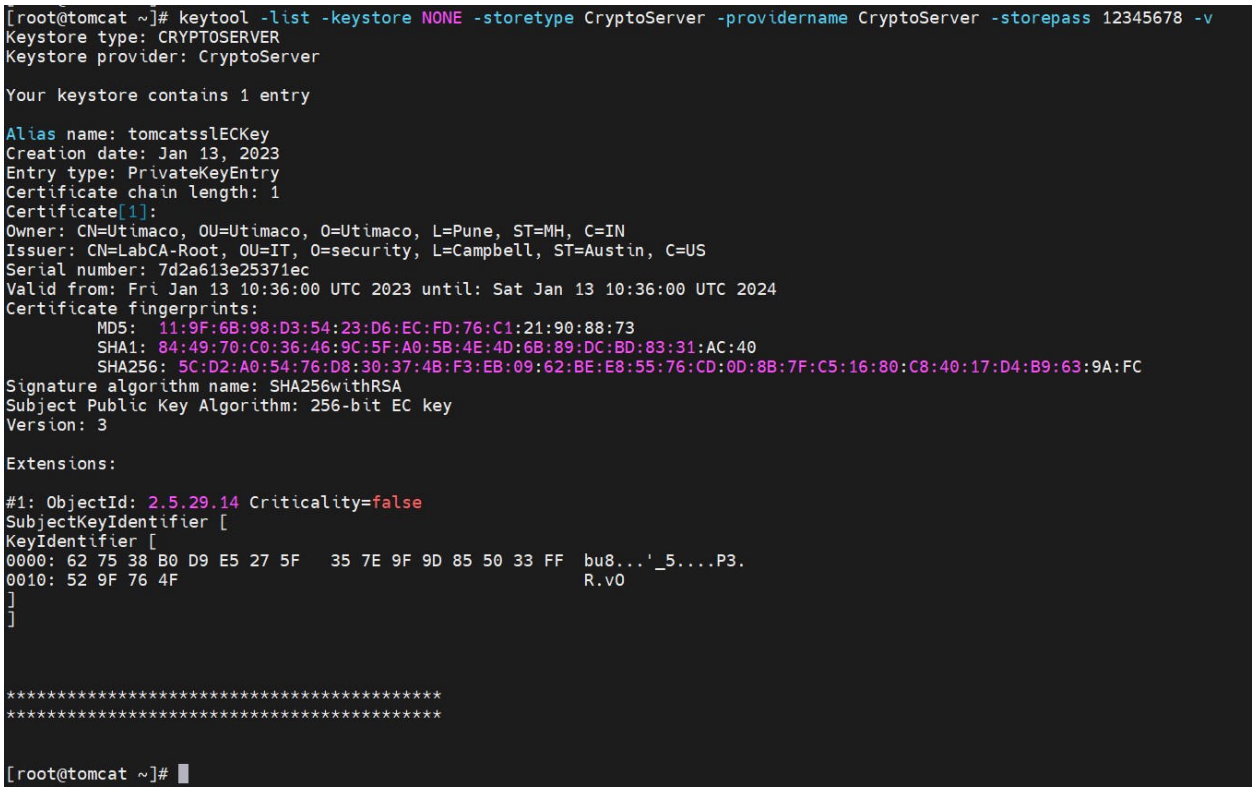

Figure 15 : Keytool list output

7.1.3 For OpenJDK 11 with EC Key

1. Generate a key pair on the Utimaco HSM.

›_ Console

```
# keytool -genkeypair -alias tomsslec1 -keyalg EC -keystore NONE -storetype

CryptoServer -storepass 123456 -providerpath

"/opt/tomcat/bin/CryptoServerJCE.jar" -providerclass

CryptoServerJCE.CryptoServerProvider -J-Djava.library.path=/opt/tomcat/bin/
-

J-cp -J/opt/tomcat/bin/CryptoServerJCE.jar/opt/tomcat/bin/
CryptoServerJCE.jar -providername CryptoServer -v
```

Provide information when prompted.

Here:

- `EC` is the key algorithm
- `NONE` is the key store for HSM
- `CryptoServer` is the store type
- `123456` is the slot PIN
- `CryptoServer` is the provider name
- `tomsslec1` is the key name that will be generated on the Utimaco HSM

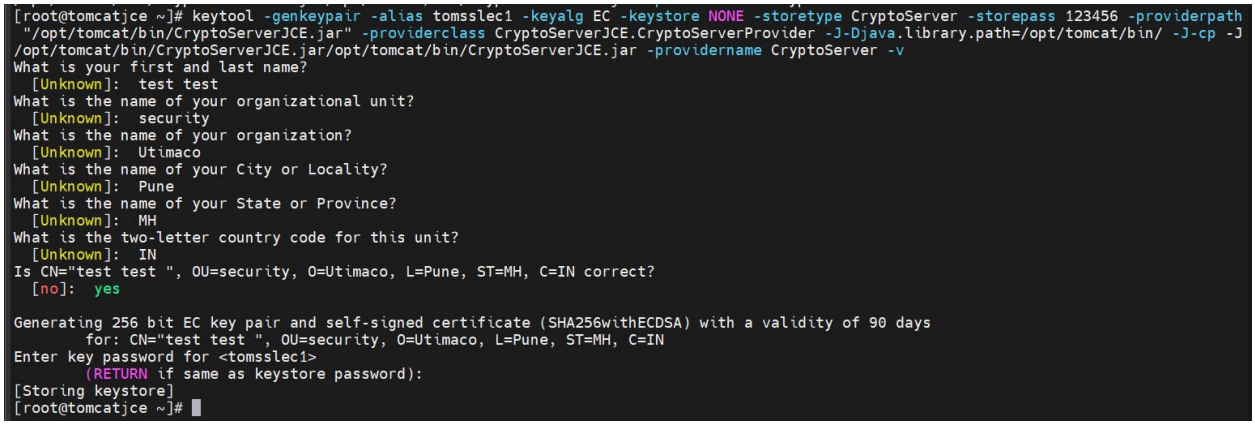

Figure 16 : Key generation using the keytool command output

For OpenJDK 11 RSA key algorithm is not supported with Utimaco HSM.

2. Verify that the keys have been generated using the `keytool` command.

›_ Console

```
# keytool -list -keystore NONE -storetype CryptoServer -storepass 123456
providerpath "/usr/lib/jvm/java-11-openjdk-11.0.2.7-

2.el8.x86_64/lib/CryptoServerJCE.jar" -providerclass

CryptoServerJCE.CryptoServerProvider -J-Djava.library.path=/usr/lib/jvm/
java-

11-openjdk-11.0.2.7-2.el8.x86_64/lib/ -J-cp -J/usr/lib/jvm/java-11-
openjdk-

11.0.2.7-2.el8.x86_64/lib/CryptoServerJCE.jar -providername CryptoServer -v
```

Here:

- `NONE` is the keystore for HSM
- `CryptoServer` is the store type
- `123456` is the PIN
- `CryptoServer` is the provider name

```
[root@tomcatjce ~]# keytool -list -keystore NONE -storetype CryptoServer -storepass 123456 -providerpath "/usr/lib/jvm/java-11-openjdk-11.0.2.7-2.el8.x8
rclass CryptoServerJCE.CryptoServerProvider -J-Djava.library.path=/usr/lib/jvm/java-11-openjdk-11.0.2.7-2.el8.x86_64/lib/ -J-cp -J/usr/lib/jvm/java-11-o
oServerJCE.jar -providername CryptoServer -v
Keystore type: CRYPTOSERVER
Keystore provider: CryptoServer

Your keystore contains 1 entry

Alias name: tomsslec1
Creation date: Dec 21, 2022
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Serial number: 227a8fa6
Valid from: Wed Dec 21 13:17:51 UTC 2022 until: Tue Mar 21 13:17:51 UTC 2023
Certificate fingerprints:
         SHA1: 9B:A7:43:15:A6:24:FB:FF:CF:8B:52:81:7B:C6:BA:35:91:62:DA:78
         SHA256: 12:7A:C0:9F:B8:94:6F:B4:D7:0A:3C:39:6B:57:1D:0A:A8:48:05:F7:1D:AA:27:3B:DF:05:24:6C:D3:3B:3E:FD
Signature algorithm name: SHA256withECDSA
Subject Public Key Algorithm: 256-bit EC (secp256r1) key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 13 17 A2 B8 AB 32 96 77   EB 07 13 37 6E 24 06 97  .....2.w...7n$..
0010: 9A 3F 5D 0D                                        .?].
]
]



*******************************************
*******************************************


[root@tomcatjce ~]#
```

Figure 17 : Keytool list output

3. List the keys using the cxitool.

›_ Console

```
# /opt/utimaco/bin/cxitool Dev=3001@127.0.0.1 LogonPass=tom,123456 Listkeys
```

```
[root@tomcatjce ~]# /opt/utimaco/bin/cxitool Dev=3001@127.0.0.1 LogonPass=tom,123456 Listkeys

idx algo  size type     group                    name                             spec
-------------------------------------------------------------------------------------
1   ECDSA 256  pub+prv  JCE2                     tomsslec1

[root@tomcatjce ~]#
```

Figure 18 : List keys output using cxitool

4. Generate a CSR using the `keytool` command

›_ Console

```
# keytool -certreq -alias tomsslec1 -file 21dec.csr -keystore NONE
-storetype

CryptoServer -storepass 123456 -providerpath

"/opt/tomcat/bin/CryptoServerJCE.jar" -providerclass

CryptoServerJCE.CryptoServerProvider -J-Djava.library.path=/opt/tomcat/bin/
-

J-cp -J/opt/tomcat/bin/CryptoServerJCE.jar/opt/tomcat/bin/
CryptoServerJCE.jar -providername CryptoServer -v
```

Here:

- `tomsslec1` is the key name
- `NONE` is the key store for HSM
- `CryptoServer` is the store type
- `123456` is the slot PIN
- `CryptoServer` is the provider name
- `21dec.csr` is the CSR file name that will be generated

5. Get this CSR signed by the CA.
6. Copy the signed certificate along with the root CA certificate chain to the Tomcat server.
7. Import the signed certificate chain reply using the command below.

›_ Console

```
# keytool -importcert -trustcacerts -alias tomsslec1 -file /root/21dec.pem
storetype CryptoServer -keystore NONE -providerpath "/opt/tomcat/bin/
CryptoServerJCE.jar" -providerclass

CryptoServerJCE.CryptoServerProvider -J-Djava.library.path=/opt/tomcat/bin/
-

J-cp -J/opt/tomcat/bin/CryptoServerJCE.jar -providername CryptoServer
```

Signed certificate must also contain certificate chain.

8. Verify that the `keytool` command shows the signed certificate as well as the root CA certificate.

›_ Console

```
# keytool -list -keystore NONE -storetype CryptoServer -storepass 123456
providerpath "/usr/lib/jvm/java-11-openjdk-11.0.2.7-

2.el8.x86_64/lib/CryptoServerJCE.jar" -providerclass

CryptoServerJCE.CryptoServerProvider -J-Djava.library.path=/usr/lib/jvm/
java-

11-openjdk-11.0.2.7-2.el8.x86_64/lib/ -J-cp -J/usr/lib/jvm/java-11-
openjdk-

11.0.2.7-2.el8.x86_64/lib/CryptoServerJCE.jar -providername CryptoServer -v
```

Here:

- `NONE` is the key store for HSM
- `CryptoServer` is the store type
- `123456` is the slot PIN

- `CryptoServer` is the provider name

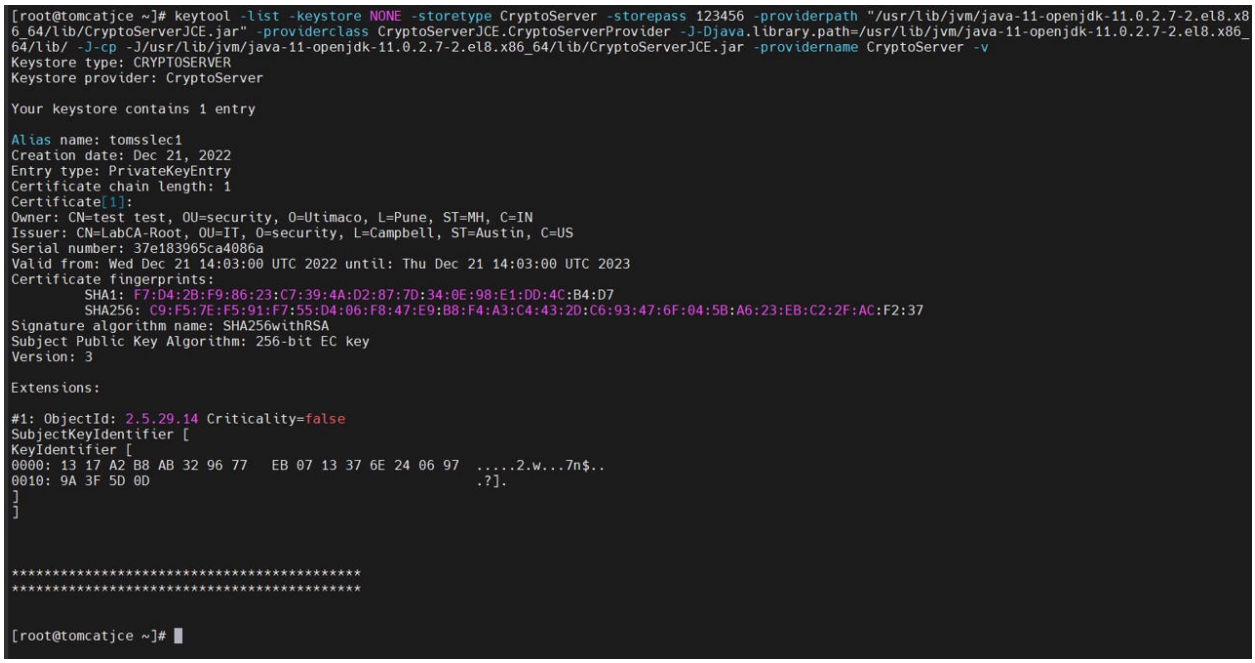

Figure 19 : Keytool list output

7.2 Generating Self-signed SSL certificate

It is recommended to use a CA signed certificate for a production environment.

7.2.1 For OpenJDK 8 with RSA Key

1. Generate a key pair on the Utimaco HSM.

›_ Console

```
# keytool -genkey -keyalg RSA -keysize 2048 -keystore NONE -storetype
CryptoServer -storepass 12345678 -providername CryptoServer -alias tomcatsslkey
```

Provide information when prompted.

Here:

- `RSA` is the key algorithm
- `2048` is the key size
- `NONE` is the keystore for HSM
- `CryptoServer` is the store type
- `12345678` is the slot PIN
- `CryptoServer` is the provider name
- `tomcatsslkey` is the key name that will be generated on the Utimaco HSM

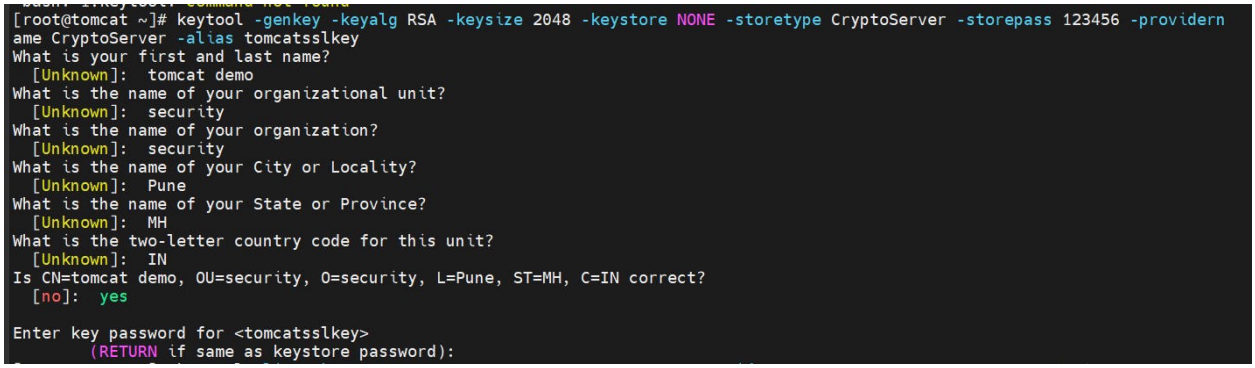

Figure 20 : keytool command to generate keys

2. List the keys using the cxitool.

›_ Console

```
 # /opt/utimaco/bin/cxitool Dev=3001@127.0.0.1 Logonpass=tomcat,12345678
Group=JCE ListKeys
```

Figure 21 : List keys output using cxitool

3. Verify that the `keytool` command shows the created certificate.

›_ Console

```
# keytool -list -keystore NONE -storetype CryptoServer -providername

CryptoServer -storepass 12345678 -v
```

Here:

- `NONE` is the key store for HSM
- `CryptoServer` is the store type
- `12345678` is the slot PIN
- `CryptoServer` is the provider name

```
[root@tomcat ~]# keytool -list -keystore NONE -storetype CryptoServer -providername CryptoServer -storepass 12345678 -v
Keystore type: CRYPTOSERVER
Keystore provider: CryptoServer

Your keystore contains 1 entry

Alias name: tomcatsslkey
Creation date: Jan 10, 2023
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Serial number: 25a483b8
Valid from: Tue Jan 10 12:41:46 UTC 2023 until: Mon Apr 10 12:41:46 UTC 2023
Certificate fingerprints:
         MD5:  3E:2F:F8:97:F4:5F:88:90:77:4D:5C:6E:F1:48:3C:D2
         SHA1: B1:85:36:9F:69:FD:A2:64:7A:D8:0B:54:48:A2:C1:BE:19:21:A5:93
         SHA256: 2F:4A:D2:D9:C9:CC:2C:B4:71:37:8C:20:39:2A:F4:61:E0:89:32:E4:11:BD:C0:0D:4C:F2:A9:35:93:2D:CC:15
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: A3 29 3C E5 1C 7D 10 40   B2 78 C4 AE 93 3C 5B CF  .)<....@.x...<[.
0010: 09 70 22 F8                                        .p".
]
]



*******************************************
*******************************************


[root@tomcat ~]#
```

Figure 22 : keytool list output

7.2.2 For OpenJDK 8 with EC Key

1. Generate an EC key pair on the Utimaco HSM.

›_ Console

```
# keytool -genkey -keyalg EC -keystore NONE -storetype CryptoServer
-storepass

12345678 -providername CryptoServer -alias tomcatselfsignedECKey
```

Provide information when prompted.

Here:

- `EC` is the key algorithm
- `NONE` is the keystore for HSM
- `CryptoServer` is the store type
- `12345678` is the slot PIN
- `CryptoServer` is the provider name
- `tomcatselfsignedECKey` is the key name that will be generated on the Utimaco HSM

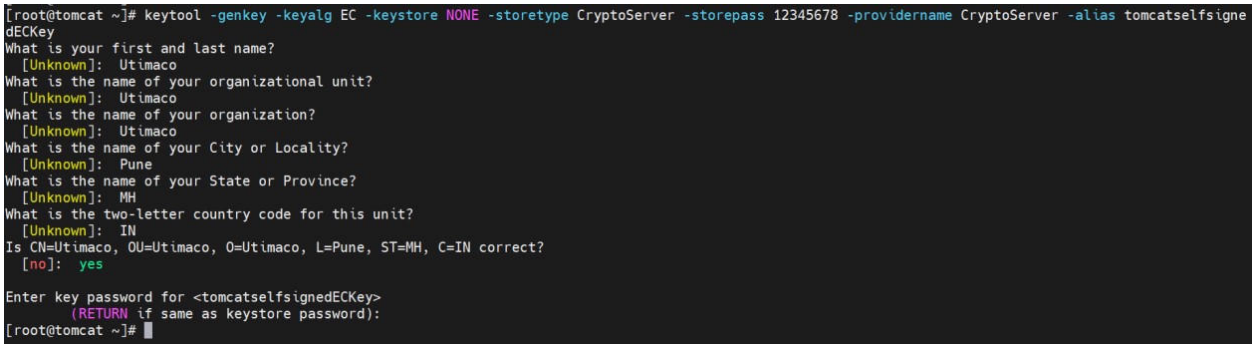

Figure 23 : keytool command to generate keys

2. List the keys using the cxitool.

›_ Console

```
# /opt/utimaco/bin/cxitool Dev=3001@127.0.0.1 Logonpass=tomcat,12345678
Group=JCE ListKeys
```

```
[root@tomcat ~]# /opt/utimaco/bin/cxitool Dev=3001@127.0.0.1 Logonpass=tomcat,12345678 Group=JCE ListKeys

idx algo  size type     group                    name                              spec
-------------------------------------------------------------------------------------
1   ECDSA 256  pub+prv  JCE                      tomcatselfsignedECKey

[root@tomcat ~]#
```

Figure 24 : List keys output using cxitool

3. Verify that the `keytool` command shows the created certificate.

›_ Console

```
# keytool -list -keystore NONE -storetype CryptoServer -providername

CryptoServer -storepass 12345678 -v
```

Here:

- `NONE` is the key store for HSM
- `CryptoServer` is the store type
- `12345678` is the PIN
- `CryptoServer` is the provider name

```
[root@tomcat ~]# keytool -list -keystore NONE -storetype CryptoServer -providername CryptoServer -storepass 12345678 -v
Keystore type: CRYPTOSERVER
Keystore provider: CryptoServer

Your keystore contains 1 entry

Alias name: tomcatselfsignedECKey
Creation date: Jan 13, 2023
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Utimaco, OU=Utimaco, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN=Utimaco, OU=Utimaco, O=Utimaco, L=Pune, ST=MH, C=IN
Serial number: 682a665f
Valid from: Fri Jan 13 12:03:24 UTC 2023 until: Thu Apr 13 12:03:24 UTC 2023
Certificate fingerprints:
         SHA1: 5E:D0:AE:B5:96:3E:E1:39:68:1D:71:C3:02:48:53:13:41:1B:AD:2C
         SHA256: A7:5E:D0:1C:C8:63:C0:42:52:95:0D:A5:EC:36:D1:67:35:44:65:28:A2:05:9D:73:83:A7:68:DF:C8:7E:A5:57
Signature algorithm name: SHA256withECDSA
Subject Public Key Algorithm: 256-bit EC (secp256r1) key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: EC 96 18 8C 08 4B 0A 70   04 F6 09 D1 CB 3A 76 0E  .....K.p.....:v.
0010: 45 D5 03 42                                        E..B
]
]



*******************************************
*******************************************

[root@tomcat ~]#
```

Figure 25 : keytool list output

7.2.3 For OpenJDK 11 with EC Key

1. Generate a key pair on the Utimaco HSM.

›_ Console

```
# keytool -genkeypair -alias tomsslec1 -keyalg EC -keystore NONE -storetype

CryptoServer -storepass 123456 -providerpath

"/opt/tomcat/bin/CryptoServerJCE.jar" -providerclass

CryptoServerJCE.CryptoServerProvider -J-Djava.library.path=/opt/tomcat/bin/
-

J-cp -J/opt/tomcat/bin/CryptoServerJCE.jar/opt/tomcat/bin/
CryptoServerJCE.jar -providername CryptoServer -v
```

Provide information when prompted.

Here:

- `EC` is the key algorithm
- `2048` is the key size
- NONE is the key store for HSM
- `CryptoServer` is the store type
- `123456` is the PIN
- `CryptoServer` is the provider name
- `tomsslec1` is the key name that will be generated on the Utimaco HSM

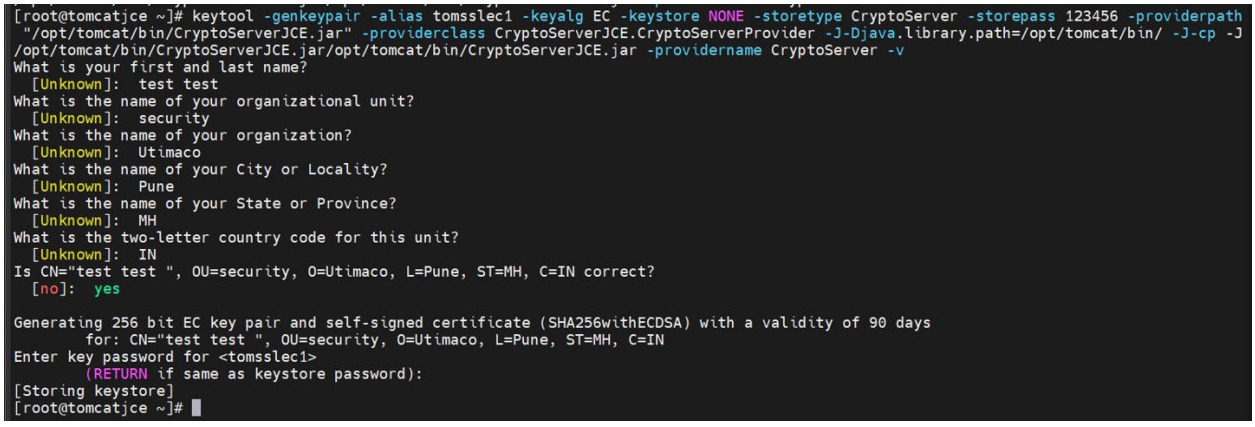

Figure 26 : keytool command to generate keys

For OpenJDK 11 RSA key algorithm is not supported with Utimaco HSM.

2. List the keys using the cxitool.

›_ Console

```
# /opt/utimaco/bin/cxitool Dev=3001@127.0.0.1 LogonPass=tom,123456 Listkeys
```

```
[root@tomcatjce ~]# /opt/utimaco/bin/cxitool Dev=3001@127.0.0.1 LogonPass=tom,123456 Listkeys

idx algo  size type     group                   name                            spec
-------------------------------------------------------------------------------------
1   ECDSA 256  pub+prv  JCE2                    tomsslec1

[root@tomcatjce ~]#
```

Figure 27 : List keys output using cxitool

3. Verify that the `keytool` command shows the created certificate.

›_ Console

```
# keytool -list -keystore NONE -storetype CryptoServer -storepass 123456
providerpath "/usr/lib/jvm/java-11-openjdk-11.0.2.7-

2.el8.x86_64/lib/CryptoServerJCE.jar" -providerclass

CryptoServerJCE.CryptoServerProvider -J-Djava.library.path=/usr/lib/jvm/
java-

11-openjdk-11.0.2.7-2.el8.x86_64/lib/ -J-cp -J/usr/lib/jvm/java-11-
openjdk-

11.0.2.7-2.el8.x86_64/lib/CryptoServerJCE.jar -providername CryptoServer -v
```

Here:

- `NONE` is the key store for HSM
- `CryptoServer` is the store type
- `123456` is the PIN
- `CryptoServer` is the provider name

```
[root@tomcatjce ~]# keytool -list -keystore NONE -storetype CryptoServer -storepass 123456 -providerpath "/usr/lib/jvm/java-11-openjdk-11.
0.2.7-2.el8.x86_64/lib/CryptoServerJCE.jar" -providerclass CryptoServerJCE.CryptoServerProvider -J-Djava.library.path=/usr/lib/jvm/java-11
-openjdk-11.0.2.7-2.el8.x86_64/lib/ -J-cp -J/usr/lib/jvm/java-11-openjdk-11.0.2.7-2.el8.x86_64/lib/CryptoServerJCE.jar -providername Crypt
oServer -v
Keystore type: CRYPTOSERVER
Keystore provider: CryptoServer

Your keystore contains 1 entry

Alias name: tomsslec1
Creation date: Dec 21, 2022
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN="test test ", OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Serial number: 227a8fa6
Valid from: Wed Dec 21 13:17:51 UTC 2022 until: Tue Mar 21 13:17:51 UTC 2023
Certificate fingerprints:
         SHA1: 9B:A7:43:15:A6:24:FB:FF:CF:8B:52:81:7B:C6:BA:35:91:62:DA:78
         SHA256: 12:7A:C0:9F:B8:94:6F:B4:D7:0A:3C:39:6B:57:1D:0A:A8:48:05:F7:1D:AA:27:3B:DF:05:24:6C:D3:3B:3E:FD
Signature algorithm name: SHA256withECDSA
Subject Public Key Algorithm: 256-bit EC key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 13 17 A2 B8 AB 32 96 77   EB 07 13 37 6E 24 06 97  .....2.w...7n$..
0010: 9A 3F 5D 0D                                        .?].
]
]



*******************************************
*******************************************
```

Figure 28 : keytool list output

7.3 Configuring Apache Tomcat to use Utimaco HSM for SSL

1. Create a file `/opt/tomcat/bin/setenv.sh` and add the environment variables listed below.

›_ Console

```
# vim /opt/tomcat/bin/setenv.sh
```

```
export CLASSPATH=/opt/tomcat/bin/CryptoServerJCE.jar
export CATALINA_OPTS=-Djava.library.path=/opt/tomcat/bin
export JRE_HOME=/usr/lib/jvm/jre
~
~
```

Figure 29 : Setting up environment

2. Open the `server.xml` file

›_ Console

```
# vim /opt/tomcat/conf/server.xml
```

3. Add the following entries for SSL to the connector section.

›_ Console

```
<Connector port="443"
protocol="org.apache.coyote.http11.Http11NioProtocol"
maxThreads="150" SSLEnabled="true">
      <UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />

      <SSLHostConfig>
            <Certificate certificateKeystoreFile=""
                         certificateKeystorePassword="123456"
                         certificateKeystoreProvider="CryptoServer"
                         certificateKeystoreType="CryptoServer"
                         certificateKeyAlias="tomsslec"
       </SSLHostConfig>

    </Connector>
```

Here:

- `certificateKeystoreFile` is blank as HSM is being used
- `certificateKeystoreType` is `CryptoServer`
- `certificateKeystoreProvider` is `CryptoServer`
- `certificateKeyAlias` is the name of the key generated using the `keytool` command
- `certificateKeystorePassword` is the PIN of the HSM key store

4. Reload the daemon using:

›_ Console

```
# systemctl daemon-reload
```

5. Restart Tomcat Service using:

›_ Console

```
# systemctl restart tomcat
```

6. Confirm that the Tomcat status is `running` using:

›_ Console

```
# systemctl status tomcat
```

7. The output below shows that it is running.

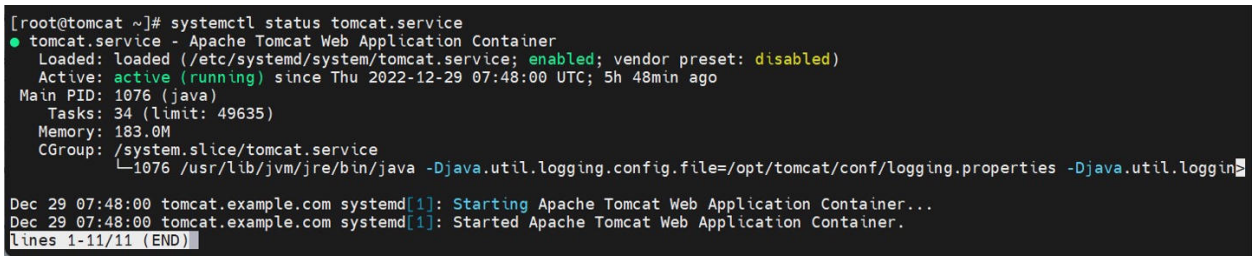

Figure 30 : Tomcat service status output

8. Now access the page over https using `https://<apache_tomcat_server_ip>:443`

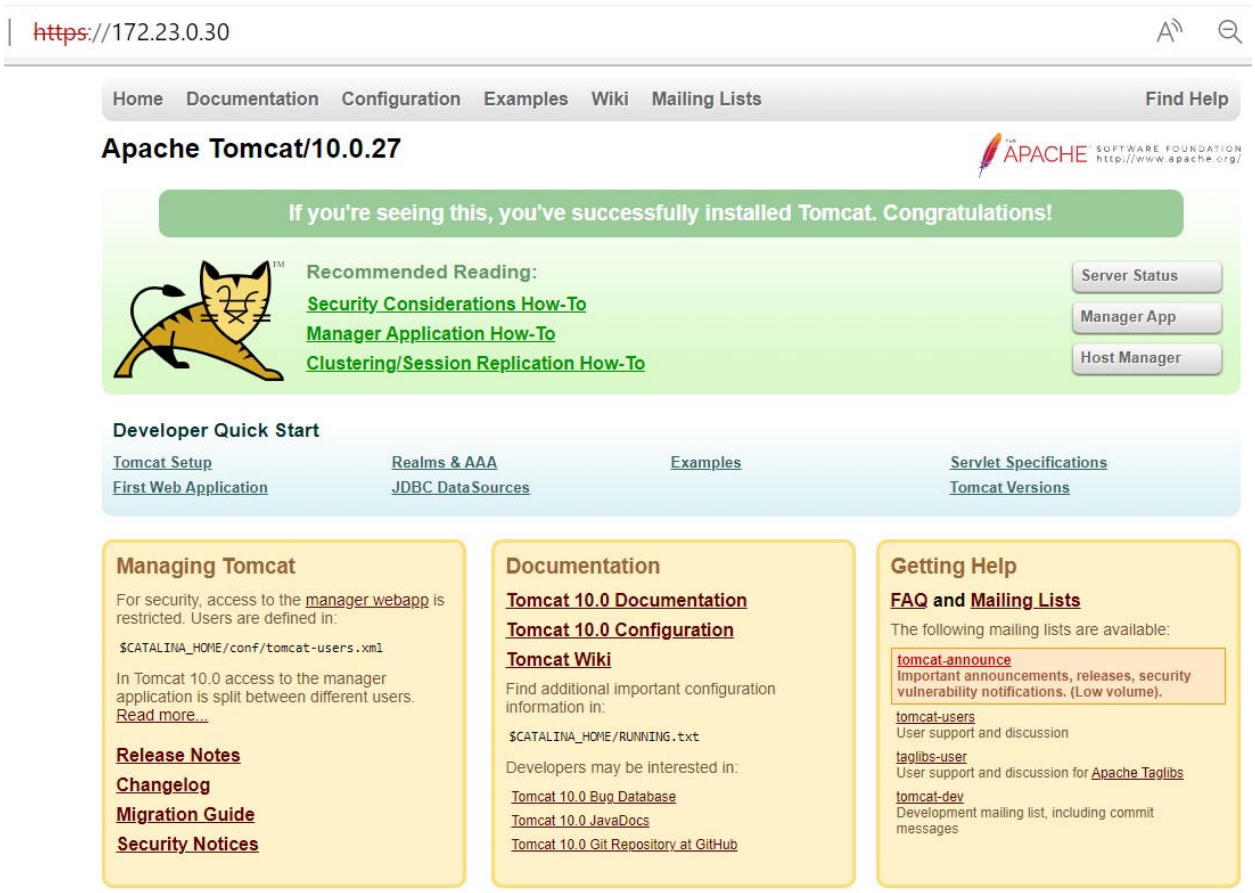

Figure 31 : Browsing the page over https

This completes the integration of Apache Tomcat and Utimaco HSM.

8 Troubleshooting

Error	Diagnosis
No supported CertificateVerify signature algorithm for RSA key	RSA algorithm is not supported with Java 11, use instead EC algo
java.net.ConnectException: Connection refused (Connection refused)	Shut down the tomcat service first and start it again.
Listkeys HSM::ConnectionException thrown in login Error::NO_DEVICE_AVAILABLE	Cryoptoserver should be up and running
keytool error: java.lang.Exception: Incomplete certificate chain in reply	Import certificate reply chain in correct format i.e .pem
keytool error: java.lang.UnsupportedOperationException	Downgrade the Java to version 11.0.2.7- 2.el8.x86_64.rpm using dnf install java-11-openjdk-1:11.0.2.7-2.el8

9 Further Information

This document forms a part of the information and support which is provided by Utimaco IS GmbH. Additional documentation can be found on the product CD in the Documentation directory.

All CryptoServer product documentation is also available at the Utimaco IS GmbH website:

https://utimaco.com/

10 References

Reference	Title/Company	Document No.
[CSADMIN]	CryptoServer – csadm Manual/Utimaco IS GmbH	2009-0003
[CSTrSh]	CryptoServer Troubleshooting/Utimaco IS GmbH	M011-0008-en
[CSADMIN2]	CryptoServer_csadm_Manual_Systemadministrators.pdf	2009-0003
[CSP11Tool2]	CryptoServer_p11tool2_Manual.pdf	2012-0004
[CSPKCSM]	CryptoServer - PKCS#11 P11CAT Manual	M013-0001-en
[CSLAN5]	CryptoServerLAN_Manual_Systemadministrators.pdf	2018-0004