HW #7 - Security

Step One: Creating Root CA
Cloning the Repository:
git clone https://bitbucket.org/stefanholek/pki-example-1

Navigating to the Repository Directory:
cd pki-example-1

Creating Directories for Root CA:
mkdir ca\root-ca\private
mkdir ca\root-ca\db
mkdir crl
mkdir certs

Initializing the Database for Root CA:
New-Item -Path ca\root-ca\db\root-ca.db -ItemType "file"
New-Item -Path ca\root-ca\db\root-ca.db.attr -ItemType "file"
"01" | Set-Content -Path ca\root-ca\db\root-ca.crt.srl
"01" | Set-Content -Path ca\root-ca\db\root-ca.crl.srl

Generating Root CA's Private Key and CSR:
openssl req -new -config etc/root-ca.conf -out ca/root-ca.csr -keyout ca/root-ca/private/root-ca.key

Self-Signing the Root CA Certificate:
openssl ca -selfsign -config etc/root-ca.conf -in ca/root-ca.csr -out ca/root-ca.crt -extensions root_ca_ext

Confirmation of step 1:
Check the Created Directories:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\joash\Desktop\272_Security\pki-example-1> ls


    Directory: C:\Users\joash\Desktop\272_Security\pki-example-1


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        11/29/2023   2:53 AM                ca
d-----        11/30/2023  12:06 AM                certs
d-----        11/29/2023   2:47 AM                crl
d-----        11/28/2023  10:57 AM                etc


PS C:\Users\joash\Desktop\272_Security\pki-example-1>
```

Check Root CA's Private Key and CSR: Verify the presence of the Root CA's private key and CSR

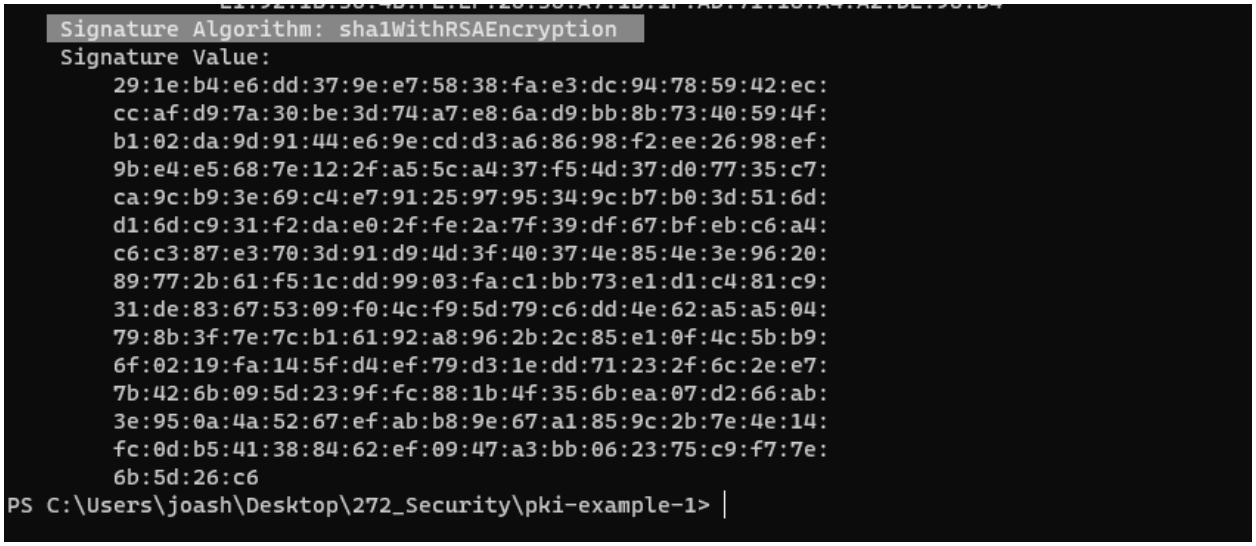
```
PS C:\Users\joash\Desktop\272_Security\pki-example-1> ls ca\root-ca\private


    Directory: C:\Users\joash\Desktop\272_Security\pki-example-1\ca\root-ca\private


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/28/2023  11:34 AM           1884 root-ca.key


PS C:\Users\joash\Desktop\272_Security\pki-example-1> ls ca\root-ca.csr


    Directory: C:\Users\joash\Desktop\272_Security\pki-example-1\ca


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/28/2023  11:34 AM           1150 root-ca.csr


PS C:\Users\joash\Desktop\272_Security\pki-example-1>
```

Verify the Root CA Certificate: Confirm the existence of the self-signed Root CA certificate.

```
PS C:\Users\joash\Desktop\272_Security\pki-example-1> ls ca\root-ca.crt


    Directory: C:\Users\joash\Desktop\272_Security\pki-example-1\ca


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/28/2023  11:46 AM           4598 root-ca.crt


PS C:\Users\joash\Desktop\272_Security\pki-example-1>
```

Inspect the Root CA Certificate: View the details of the Root CA certificate.

```
PS C:\Users\joash\Desktop\272_Security\pki-example-1> openssl x509 -in ca\root-ca.crt -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC = org, DC = simple, O = Simple Inc, OU = Simple Root CA, CN = Simple Root CA
        Validity
            Not Before: Nov 28 19:38:13 2023 GMT
            Not After : Nov 27 19:38:13 2033 GMT
        Subject: DC = org, DC = simple, O = Simple Inc, OU = Simple Root CA, CN = Simple Root CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:c0:28:4b:c6:d0:42:09:50:bc:4b:b9:5a:15:89:
                    24:8f:83:d4:e9:f8:53:99:cc:36:1e:7d:dd:14:34:
                    22:a1:d6:cb:32:56:d3:fc:34:b6:7d:f9:04:c8:f9:
                    e4:c8:64:ce:6f:aa:2a:7c:5f:dd:fb:ec:26:c5:60:
                    e5:53:fe:b4:12:c6:90:d7:de:80:97:a2:75:78:8b:
                    95:09:24:bb:c7:0a:f3:3f:37:09:e5:29:69:dd:6a:
                    79:b4:07:06:d0:8f:83:c2:4d:10:f6:51:5c:64:e6:
                    a5:c7:f9:b1:4e:44:33:33:5b:ab:27:4f:ec:64:1d:
                    f9:03:13:d2:cb:91:a7:12:b0:fa:f9:db:93:f0:bf:
                    3e:64:c4:9d:0b:c7:57:d5:b5:1d:26:20:83:63:19:
                    0b:2c:84:d1:47:e6:f8:53:a5:6f:ab:35:99:f3:25:
                    5a:c4:c2:98:c9:86:82:6e:4c:22:f5:0c:64:df:1b:
                    46:7e:52:3d:d6:19:64:a2:45:a9:64:5e:e8:3a:b8:
                    90:f2:ea:f7:66:fb:82:6f:5a:ff:4f:a4:a0:7b:ba:
                    b7:87:d7:b0:31:61:ca:13:59:72:a4:25:4a:9e:da:
                    58:c9:70:39:34:bb:0e:51:d3:cb:e5:a9:2b:bd:bd:
                    5c:38:6f:d1:98:bf:44:be:10:b2:2e:0a:68:9e:df:
                    44:d1
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Subject Key Identifier:
                E1:92:1B:36:4B:FE:EF:25:35:A7:1B:1F:AD:71:18:A4:A2:DE:95:D4
            X509v3 Authority Key Identifier:
                E1:92:1B:36:4B:FE:EF:25:35:A7:1B:1F:AD:71:18:A4:A2:DE:95:D4
    Signature Algorithm: sha1WithRSAEncryption
    Signature Value:
        29:1e:b4:e6:dd:37:9e:e7:58:38:fa:e3:dc:94:78:59:42:ec:
        cc:af:d9:7a:30:be:3d:74:a7:e8:6a:d9:bb:8b:73:40:59:4f:
        b1:02:da:9d:91:44:e6:9e:cd:d3:a6:86:98:f2:ee:26:98:ef:
        9b:e4:e5:68:7e:12:2f:a5:5c:a4:37:f5:4d:37:d0:77:35:c7:
        ca:9c:b9:3e:69:c4:e7:91:25:97:95:34:9c:b7:b0:3d:51:6d:
        d1:6d:c9:31:f2:da:e0:2f:fe:2a:7f:39:df:67:bf:eb:c6:a4:
```



Step 2: Creating and setting up the Signing CA
Creating Directories for the Signing CA:
mkdir -p ca\signing-ca\private ca\signing-ca\db crl certs

Initializing the Database for the Signing CA:
New-Item -Path ca\signing-ca\db\signing-ca.db -ItemType "file"
New-Item -Path ca\signing-ca\db\signing-ca.db.attr -ItemType "file"
"01" | Set-Content -Path ca\signing-ca\db\signing-ca.crt.srl
"01" | Set-Content -Path ca\signing-ca\db\signing-ca.crl.srl

Creating the CSR for the Signing CA:
openssl req -new -config etc/signing-ca.conf -out ca/signing-ca.csr -keyout ca/signing-ca/private/signing-ca.key

Signing the CSR to Create the Signing CA Certificate:
openssl ca -config etc/root-ca.conf -in ca/signing-ca.csr -out ca/signing-ca.crt -extensions signing_ca_ext

Confirming creation and setting up the Signing CA

Check for Signing CA Directories: Verify that the directories for the Signing CA were created.

```
PS C:\Users\joash\Desktop\272_Security\pki-example-1> ls ca\signing-ca


    Directory: C:\Users\joash\Desktop\272_Security\pki-example-1\ca\signing-ca


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        11/29/2023   4:32 AM                db
d-----        11/29/2023   2:51 AM                private
-a----        11/29/2023   4:32 AM           4836 01.pem
-a----        11/29/2023   3:03 AM           4627 signing-ca.crt


PS C:\Users\joash\Desktop\272_Security\pki-example-1>
```

Verify Signing CA's Private Key and CSR: Confirm the presence of the Signing CA's private key and CSR.

```
PS C:\Users\joash\Desktop\272_Security\pki-example-1> ls ca\signing-ca\private


    Directory: C:\Users\joash\Desktop\272_Security\pki-example-1\ca\signing-ca\private


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/29/2023   2:51 AM           1884 signing-ca.key

PS C:\Users\joash\Desktop\272_Security\pki-example-1> ls ca\signing-ca.csr


    Directory: C:\Users\joash\Desktop\272_Security\pki-example-1\ca


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/29/2023   2:51 AM           1162 signing-ca.csr


PS C:\Users\joash\Desktop\272_Security\pki-example-1>
```

Check the Signing CA Certificate: Confirm the existence of the Signing CA certificate.

```
PS C:\Users\joash\Desktop\272_Security\pki-example-1> ls ca\signing-ca.crt


    Directory: C:\Users\joash\Desktop\272_Security\pki-example-1\ca


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/29/2023   2:53 AM           4627 signing-ca.crt


PS C:\Users\joash\Desktop\272_Security\pki-example-1>
```

Inspect the Signing CA Certificate: view the details of the Signing CA certificate.

```
PS C:\Users\joash\Desktop\272_Security\pki-example-1> openssl x509 -in ca\signing-ca.crt -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC = org, DC = simple, O = Simple Inc, OU = Simple Root CA, CN = Simple Root CA
        Validity
            Not Before: Nov 29 10:52:42 2023 GMT
            Not After : Nov 28 10:52:42 2033 GMT
        Subject: DC = org, DC = simple, O = Simple Inc, OU = Simple Signing CA, CN = Simple Signing CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:bd:8e:fe:9a:23:c5:0f:f1:7d:80:a3:0b:40:b1:
                    e6:23:e0:8f:a6:d7:4c:e3:e6:a7:84:91:f6:f3:1c:
                    fa:c4:b5:c2:cb:dd:68:a5:14:6b:8a:d0:34:6a:2c:
                    f3:62:8b:9c:e1:57:84:fa:d3:66:15:b1:83:61:fb:
                    6b:4b:78:61:25:54:ba:d7:b8:28:e7:74:1f:92:00:
                    42:9e:25:0b:05:05:a1:4c:1c:36:9d:56:35:f3:5c:
                    e8:b0:e2:12:e5:1f:17:56:79:e9:16:74:73:33:f0:
                    9a:01:26:ba:5d:be:ce:00:c5:04:6b:1a:e7:10:b0:
                    02:37:15:d9:ee:21:89:82:06:a8:cd:56:f5:52:03:
                    c7:62:db:4a:44:7d:08:99:70:22:76:f4:d3:c4:f8:
                    eb:b4:49:b0:70:dd:92:dd:70:4c:e1:29:46:b4:9b:
                    a5:77:46:17:cd:53:fb:49:de:ab:8e:d0:d6:cf:b5:
                    9e:75:a8:0d:3a:b7:f4:25:a8:27:31:e9:ee:ed:3e:
                    0e:dc:20:e9:d2:f2:03:2b:2d:e6:ae:8e:fb:28:a7:
                    64:c0:fe:8b:b1:79:2c:d8:9a:3b:7d:81:f6:5d:c8:
                    3b:9d:07:00:4d:3d:fb:c9:82:d9:23:4c:a8:00:76:
                    c0:a3:de:6f:4c:a9:e4:a5:d3:5b:8c:0e:4d:11:f2:
                    6b:fd
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
            X509v3 Subject Key Identifier:
                9A:10:C3:E9:A3:A4:7A:F9:63:17:26:11:43:58:66:1D:E7:D1:F0:EE
            X509v3 Authority Key Identifier:
                E1:92:1B:36:4B:FE:EF:25:35:A7:1B:1F:AD:71:18:A4:A2:DE:95:D4
```

Step 3: Creating the TLS certificate

Generating the Web Server's Private Key and CSR:
openssl req -new -nodes -out certs/webserver.csr -keyout certs/webserver.key -config etc/server.conf

Signing the Web Server's CSR with the Signing CA:
openssl ca -config etc/signing-ca.conf -in certs/webserver.csr -out certs/webserver.crt -extensions server_ext

Converting the TLS Certificate to PKCS#12 Format:
openssl pkcs12 -export -out certs/keystore.p12 -inkey certs/webserver.key -in certs/webserver.crt -certfile ca/signing-ca.crt

Confirm that the respective files for each step were successfully created:

Confirm Generating the Web Server's Private Key and CSR: To check if the private key (webserver.key) and CSR (webserver.csr) exist:

```
PS C:\Users\joash\Desktop\272_Security\pki-example-1> ls certs/webserver.key


    Directory: C:\Users\joash\Desktop\272_Security\pki-example-1\certs


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/29/2023   4:19 AM           1732 webserver.key


PS C:\Users\joash\Desktop\272_Security\pki-example-1> ls certs/webserver.csr


    Directory: C:\Users\joash\Desktop\272_Security\pki-example-1\certs


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/29/2023   4:30 AM           1158 webserver.csr


PS C:\Users\joash\Desktop\272_Security\pki-example-1>
```

Confirm Signing the Web Server's CSR with the Signing CA:

- To check if the signed TLS certificate (webserver.crt) exists:

```
PS C:\Users\joash\Desktop\272_Security\pki-example-1> ls certs/webserver.crt


    Directory: C:\Users\joash\Desktop\272_Security\pki-example-1\certs


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/29/2023   4:32 AM           4836 webserver.crt


PS C:\Users\joash\Desktop\272_Security\pki-example-1>
```

Confirm Converting the TLS Certificate to PKCS#12 Format:

- To check if the PKCS#12 keystore file (keystore.p12) exists:

```
PS C:\Users\joash\Desktop\272_Security\pki-example-1> ls certs/keystore.p12


    Directory: C:\Users\joash\Desktop\272_Security\pki-example-1\certs


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/30/2023  12:06 AM           2800 keystore.p12


PS C:\Users\joash\Desktop\272_Security\pki-example-1>
```

Step 4: onfiguring Apache Tomcat to use the generated TLS certificate for enabling HTTPS connections.

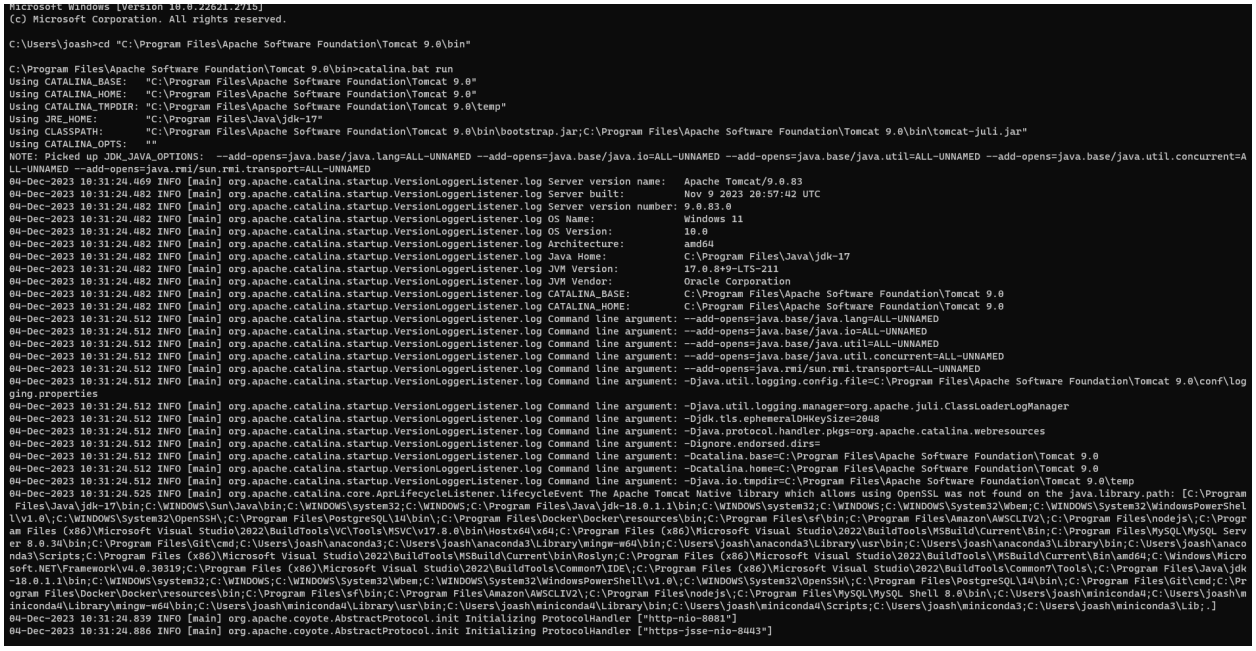
Locate and Open Tomcat's server.xml Configuration File:

```
PS C:\Users\joash> ls "C:\Program Files\Apache Software Foundation\Tomcat 9.0\conf\server.xml"


    Directory: C:\Program Files\Apache Software Foundation\Tomcat 9.0\conf


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/30/2023   1:13 AM           8353 server.xml


PS C:\Users\joash>
```

Confirm SSL/TLS Connector Configuration:

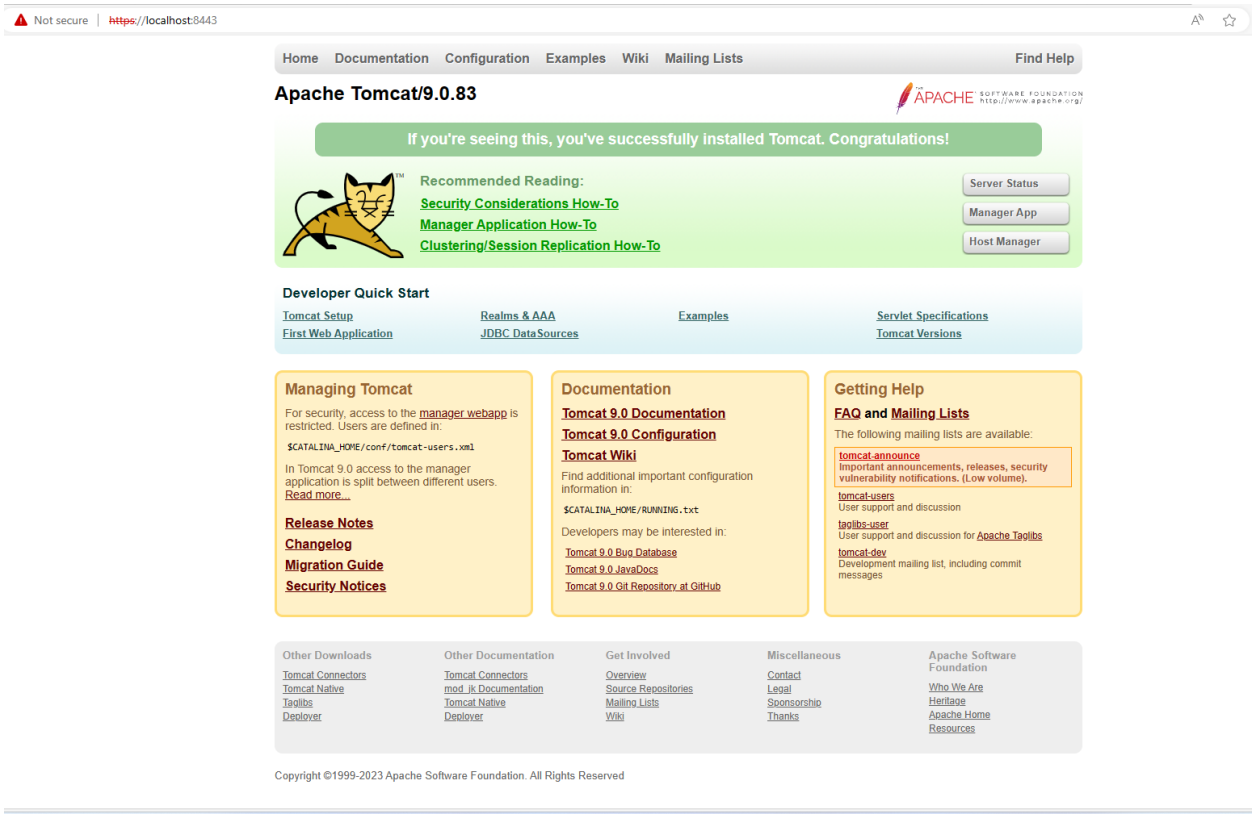
```
    -->
    <!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443
         This connector uses the NIO implementation. The default
         SSLImplementation will depend on the presence of the APR/native
         library and the useOpenSSL attribute of the AprLifecycleListener.
         Either JSSE or OpenSSL style configuration may be used regardless of
         the SSLImplementation selected. JSSE style configuration is used below.
    -->
  <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
             maxThreads="150" SSLEnabled="true"
             scheme="https" secure="true" clientAuth="false"
             sslProtocol="TLS" keystoreFile="conf/keystore.p12"
             keystorePas[redacted] keystoreType="PKCS12"
  />



    <!--
```

Running Tomcat:



Test HTTPS Connection:

- To verify that Tomcat is correctly configured with SSL/TLS, open a web browser and navigate to https://localhost:8443.
- If the Tomcat homepage is displayed without any security warnings (or with an expected security warning due to the use of a self-signed certificate), it confirms that SSL/TLS has been successfully configured.


Not secure | https://localhost:8443
Home Documentation Configuration Examples Wiki Mailing Lists Find Help
Apache Tomcat/9.0.83
APACHE SOFTWARE FOUNDATION http://www.apache.org/
If you're seeing this, you've successfully installed Tomcat. Congratulations!
Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To
Server Status
Manager App
Host Manager
Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions
Managing Tomcat
For security, access to the manager webapp is restricted. Users are defined in:
$CATALINA_HOME/conf/tomcat-users.xml
In Tomcat 9.0 access to the manager application is split between different users.
Read more...
Release Notes
Changelog
Migration Guide
Security Notices
Documentation
Tomcat 9.0 Documentation
Tomcat 9.0 Configuration
Tomcat Wiki
Find additional important configuration information in:
$CATALINA_HOME/RUNNING.txt
Developers may be interested in:
Tomcat 9.0 Bug Database
Tomcat 9.0 JavaDocs
Tomcat 9.0 Git Repository at GitHub
Getting Help
FAQ and Mailing Lists
The following mailing lists are available:
tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).
tomcat-users
User support and discussion
taglibs-user
User support and discussion for Apache Taglibs
tomcat-dev
Development mailing list, including commit messages
Other Downloads
Tomcat Connectors
Tomcat Native
Taglibs
Deployer
Other Documentation
Tomcat Connectors
mod_jk Documentation
Tomcat Native
Deployer
Get Involved
Overview
Source Repositories
Mailing Lists
Wiki
Miscellaneous
Contact
Legal
Sponsorship
Thanks
Apache Software Foundation
Who We Are
Heritage
Apache Home
Resources
Copyright ©1999-2023 Apache Software Foundation. All Rights Reserved