**Create 3 self-signed certificates :**

keytool -alias root -dname "cn=RootCA, ou=Root\_CertificateAuthority, o=CertificateAuthority, c=IN" -genkeypair -storepass password -keyalg RSA -keystore c:/keyStore.keystore -validity 1000

keytool -alias intermediate -dname "cn=IntermediateCA, ou=Intermediate\_CertificateAuthority, o=CertificateAuthority, c=IN" -genkeypair -storepass password -keyalg RSA -keystore c:/keyStore.keystore -validity 1000

keytool -alias server -dname "cn=OAuth2Server, ou=Java, o=Oracle, c=IN" -genkeypair -storepass password -keyalg RSA -keystore c:/oauth2server.jks -validity 1000

**Create a certificate request (CSR) for CA\_Intermediate**

keytool -alias intermediate -certreq -storepass password -keystore c:/keyStore.keystore -keyalg RSA | keytool -alias root -gencert -ext san=dns:intermediate -storepass password -keystore c:/keyStore.keystore -keyalg RSA | keytool -alias intermediate -importcert -storepass password -keyalg RSA -keystore c:/keyStore.keystore -validity 1000

**Export the root certificate of CA\_Root and import it to oauth2server keystore**

keytool -export -alias root -storepass password -keystore c:/keyStore.keystore | keytool -import -alias root -keystore c:/oauth2server.jks -storepass password -noprompt –trustcacerts -validity 1000

**Create a certificate request (CSR) for your server certificate. Sign it with CA\_Intermediate using the gencert option.**

keytool -alias server -certreq -storepass password -keyalg RSA -keystore c:/oauth2server.jks | keytool -alias intermediate -gencert -ext san=dns:OAuth2Server -storepass password -keystore c:/keyStore.keystore -keyalg RSA | keytool -alias server -importcert -storepass password -keyalg RSA -keystore c:/oauth2server.jks -noprompt –trustcacerts –validity 1000

**Create a trust keystore with root and intermediate certificate:**

keytool -export -alias intermediate -storepass password -keystore c:/keyStore.keystore | keytool -import -alias intermediate -keystore c:/trust.jks -storepass password -trustcacerts -noprompt -validity 1000

keytool -export -alias root -storepass password -keystore c:/keyStore.keystore | keytool -import -alias root -keystore c:/trust.jks -storepass password -trustcacerts -noprompt -validity 1000

**List the oauth2server keystore :**

keytool -list -v -keystore c:/oauth2server.jks -storepass password

**Delete the root certificate of CA\_Root from oauth2server keystore. (We need it only for chaining)**

keytool -delete -alias root -keystore c:/oauth2server.jks -storepass password

**To extract certificates**

keytool -exportcert -alias root -keystore C:/trust.jks -file c:/rootCA.cer

keytool -exportcert -alias intermediate -keystore C:/trust.jks -file intermediateCA.cer

**To import rootCA into jvm’s truststore**

keytool -import -alias rootCA -file c:/rootCA.cer -keystore "C:\Program Files\Java\jdk-20\lib\security\cacerts" -storepass changeit

**Verifying import**

keytool -list -v -keystore "C:\Program Files\Java\jdk-20\lib\security\cacerts" -alias rootCA

To create a CA certificate execute the following steps:

**Create CA keystore (no change)**

keytool -genkeypair -alias root -dname "cn=RootCA, ou=Root\_CertificateAuthority, o=CertificateAuthority, c=IN" -keyalg RSA -keystore c:/caKeyStore.keystore -storepass password -keypass password -ext KeyUsage=digitalSignature,keyCertSign -ext BasicConstraints=ca:true,PathLen:3 -validity 1000

keytool -genkeypair -alias intermediate -dname "cn=IntermediateCA, ou=Intermediate\_CertificateAuthority, o=CertificateAuthority, c=IN" -keyalg RSA -keystore c:/caKeyStore.keystore -storepass password -keypass password -ext KeyUsage=digitalSignature,keyCertSign -ext BasicConstraints=ca:true,PathLen:3 -validity 1000

**Create client keystore**

keytool -genkeypair -alias client -dname "cn=OAuth2Client, ou=Java, o=Oracle, c=IN" -keyalg RSA -keystore c:/oauth2client.jks -storepass password -keypass password -validity 1000 -ext KeyUsage=digitalSignature,dataEncipherment,keyEncipherment,keyAgreement -ext ExtendedKeyUsage=serverAuth,clientAuth

**Create a certificate request (CSR) and certificate for IntermediateCA (no change)**

keytool -certreq -alias intermediate -storepass password -keystore c:/caKeyStore.keystore -keyalg RSA | keytool -alias root -gencert -ext san=dns:intermediate -storepass password -keystore c:/caKeyStore.keystore -keyalg RSA -validity 1000 -ext KeyUsage=digitalSignature,dataEncipherment,keyEncipherment,keyAgreement,keyCertSign -ext ExtendedKeyUsage=serverAuth,clientAuth -ext BasicConstraints=ca:true,PathLen:3 -rfc | keytool -alias intermediate -importcert -storepass password -keyalg RSA -keystore c:/caKeyStore.keystore

**Export the root certificate of RootCA and import it to oauth2client keystore (required for trust chain) before the client certificate is imported in client keystore**

keytool -exportcert -alias root -storepass password -keystore c:/caKeyStore.keystore -validity 1000 | keytool -importcert -alias root -keystore c:/oauth2client.jks -storepass password -noprompt –trustcacerts  
  
**Export the intermediateCA and import it to oauth2client keystore (required for trust chain) before the certificate is imported in the client keystore**keytool -exportcert -alias intermediate -storepass password -keystore c:/caKeyStore.keystore -validity 1000 | keytool -importcert -alias intermediate -keystore c:/oauth2client.jks -storepass password -noprompt -trustcacerts

**Create certificate signing request (CSR) for the client and later importing into client keystore**

keytool -certreq -alias client -storepass password -keystore c:/oauth2client.jks -keyalg RSA | keytool -alias intermediate -gencert -ext san=dns:OAuth2Client -storepass password -keystore c:/caKeyStore.keystore -keyalg RSA -validity 1000 -ext KeyUsage=digitalSignature,dataEncipherment,keyEncipherment,keyAgreement -ext ExtendedKeyUsage=serverAuth,clientAuth -rfc | keytool -alias client -importcert -storepass password -keyalg RSA -keystore c:/oauth2client.jks

**Delete the imported CA certificates from the client keystore**

keytool -delete -alias root -keystore c:/oauth2client.jks -storepass password

keytool -delete -alias intermediate -keystore c:/oauth2client.jks -storepass password

**Create a trust keystore with root and intermediate certificate: (first root then intermediate) (no change)**

keytool -exportcert -alias root -storepass password -keystore c:/caKeyStore.keystore | keytool -importcert -alias root -keystore c:/trust.jks -storepass password -trustcacerts -noprompt

keytool -exportcert -alias intermediate -storepass password -keystore c:/caKeyStore.keystore | keytool -importcert -alias intermediate -keystore c:/trust.jks -storepass password -trustcacerts -noprompt

**To extract certificates**

keytool -exportcert -alias root -keystore C:/trust.jks -file c:/rootCA.cer -storepass password

keytool -exportcert -alias client -keystore C:/oauth2client.jks -file c:/oauth2client.cer -storepass password

keytool -exportcert -alias intermediate -keystore C:/trust.jks -file c:/intermediateCA.cer -storepass password

**To import rootCA into jvm’s truststore**

keytool -import -trustcacerts -keystore "C:\Program Files\Java\jdk-20\lib\security\cacerts" -storepass changeit -alias rootCA -file “C:\rootCA.cer” (use this)

keytool -import -alias rootCA -file c:/rootCA.cer -keystore "C:\Program Files\Java\jdk-20\lib\security\cacerts" -storepass changeit

**To import intermediateCA into jvm’s truststore**

keytool -import -alias intermediateCA -file c:/intermediateCA.cer -keystore "C:\Program Files\Java\jdk-20\lib\security\cacerts" -storepass changeit

**Verifying import**

keytool -list -v -keystore "C:\Program Files\Java\jdk-20\lib\security\cacerts" -alias rootCA

**TO LIST THE CACERTS ENTRIES**

keytool -list -v -keystore "C:\Program Files\Java\jdk-20\lib\security\cacerts" -storepass changeit -alias rootCA  
  
  
**To delete the older entries of CACERTS of rootCA alias :**   
  
keytool -delete -alias rootCA -keystore "C:\Program Files\Java\jdk-20\lib\security\cacerts" -storepass changeit