**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

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

keytool -alias server -dname "cn=OAuth2Server, ou=Java, o=Oracle, c=IN" -genkeypair -storepass password -keyalg RSA -keystore c:/oauth2server.jks –validity 500

**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 500

**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

**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 500

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

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

**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 -storepass password

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

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

**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 -trustcacerts

**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 -trustcacerts

**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**

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 server keystore**

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

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 oauth2server keystore (required for trust chain) before the server certificate is imported in server keystore**

keytool -exportcert -alias root -storepass password -keystore c:/caKeyStore.keystore -validity 1000| keytool -importcert -alias root -keystore c:/oauth2server.jks -storepass password -noprompt –trustcacerts  
  
**Export the root certificate RootCA and intermediateCA and import them to oauth2server keystore (required for trust chain) before the server certificate is imported in server keystore**keytool -exportcert -alias intermediate -storepass password -keystore c:/caKeyStore.keystore -validity 1000 | keytool -importcert -alias intermediate -keystore c:/oauth2server.jks -storepass password -noprompt -trustcacerts

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

keytool -certreq -alias server -storepass password -keystore c:/oauth2server.jks -keyalg RSA | keytool -alias intermediate -gencert -ext san=dns:OAuth2Server -storepass password -keystore c:/caKeyStore.keystore -keyalg RSA -ext KeyUsage=digitalSignature,dataEncipherment,keyEncipherment,keyAgreement -ext ExtendedKeyUsage=serverAuth,clientAuth -rfc | keytool -alias server -importcert -storepass password -keyalg RSA -keystore c:/oauth2server.jks -validity 1000

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

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

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

**Create a trust keystore with root and intermediate certificate: (first root then intermediate)**

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 server -keystore C:/oauth2server.jks -file c:/oauth2server.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