**Keytool**

The java keytool is a key and certificate management tool that is used to manipulate the java keystore. The keystore is container for authorization certificates or public key certificates. and is often used by Java-based applications for encryption, authentication, and serving over HTTPS. Its entries are protected by a keystore password. A keystore entry is identified by an alias, and it consists of keys and certificates that form a trust chain.

**Creating and Importing Keystore Entries :**

This section covers Java Keytool commands that are related to generating key pairs and certificates, and importing certificates.

* **Generate Keys in New/Existing Keystore**

Use this method if you want to use HTTP (HTTP over TLS) to secure your Java application. This will create a new key pair in a new or existing Java Keystore, which can be used to create a CSR, and obtain an SSL certificate from a Certificate Authority.

This command generates a 2048-bit RSA key pair, under the specified alias (domain), in the specified keystore file (keystore.jks):

keytool -genkeypair -alias domain -keyalg RSA -keystore keystore.jks

* **Generate CSR(Certificate Signing Request) For Existing Private Key**

Use this method if you want to generate an CSR that you can send to a CA to request the issuance of a CA-signed SSL certificate. It requires that the keystore and alias already exist; you can use the previous command to ensure this.

This command creates a CSR (domain.csr) signed by the private key identified by the alias (domain) in the (keystore.jks) keystore.

keytool -certreq -alias domain -file domain.csr -keystore keystore.jks

After entering the keystore’s password, the CSR will be generated.

* **Import Signed/Root/Intermediate Certificate**Use this method if you want to import a signed certificate, e.g. a certificate signed by a CA, into your keystore; it must match the private key that exists in the specified alias. You may also use this same command to import root or intermediate certificates that your CA may require to complete a chain of trust. Simply specify a unique alias, such as root instead of domain, and the certificate that you want to import.

This command imports the certificate (domain.crt) into the keystore (keystore.jks), under the specified alias (domain). If you are importing a signed certificate, it must correspond to the private key in the specified alias:

keytool -importcert -trustcacerts -file domain.crt -alias domain -keystore keystore.jks

* **Generate Self-Signed Certificate in New/Existing Keystore**

Use this command if you want to generate a self-signed certificate for your Java applications. This is actually the same command that is used to create a new key pair, but with the validity lifetime specified in days.

This command generates a 2048-bit RSA key pair, valid for 365 days, under the specified alias (domain), in the specified keystore file (keystore.jks):

keytool -genkey -alias domain -keyalg RSA -validity 365 -keystore keystore.jks

**Viewing Keystore Entries** This section covers listing the contents of a Java Keystore, such   
 as viewing certificate information or exporting certificates.

keytool -list -keystore keystore.jk

**List Verbose Keystore Contents** This command lists verbose information about the entries a keystore (keystore.jks)   
 contains, including certificate chain length, fingerprint of certificates in the chain,   
 distinguished names, serial number, and creation/expiration date, under their   
 respective aliases:

keytool -list -v -keystore keystore.jks

**Use Keytool to View Certificate Information**

This command prints verbose information about a certificate file (certificate.crt),   
 including its fingerprints, distinguished name of owner and issuer, and the time   
 period of its validity:

keytool -printcert -file domain.crt

**Export Certificate**

This command exports a binary DER-encoded certificate (domain.der), that is associated   
 with the alias (domain), in the keystore (keystore.jks):

keytool -exportcert -alias domain -file domain.der -keystore keystore.jks

**Modifying Keystore**

This section covers the modification of Java Keystore entries,   
 such as deleting or renaming aliases.

**Change Keystore Password**

This command is used to change the password of a keystore  
 (keystore.jks):

keytool -storepasswd -keystore keystore.jks

then the new password. You may also specify the new password   
 in the command by using the -new newpass option, where   
 “newpass” is the password.

**Delete Alias**

This command is used to delete an alias (domain) in a keystore   
 (keystore.jks):

keytool -delete -alias domain -keystore keystore.jks

**Rename Alias**

This command will rename the alias (domain) to the destination   
 alias (newdomain) in the keystore (keystore.jks):

keytool -changealias -alias domain -destalias newdomain -keystore keystore.jks