

Java KeyTool

Juan Antonio León



¿Qué es Java KeyTool?

Java Keytool es un comando que permite generar pares de Private Key/Public Key y almacenarlos en un Java KeyStore.

Java keyStore (JKS) es un almacén de certificados y entidades de certificación usado por las aplicaciones Java para trabajar con SSL. Cada certificado contenido en la keyStore es identificado por un alias único.

Generar un certificado autofirmado

```
>> keytool -genkey -alias alias -keypass clave -validity 365 -storepass claveKeyStore
```

- -alias: es el nombre con el que haremos referencia al par de claves creado.
- -keypass: es la clave con la que podremos acceder a la clave privada del par de claves creado.
- -validity: es el tiempo de validez, en días. En nuestro ejemplo, un año.
- -storepass: clave para acceder a nuestro keystore.

Generar el CSR

```
>> keytool -certreq -alias alias -file certificado.csr -keypass clave -storepass claveDeKeyStore
```

- -file: nombre del fichero de salida, que luego mandaremos a la CA.

Crear nuestro propia CA

OpenSSL nos proporciona un script que nos facilita la tarea de crear la CA. Basta con hacer:

```
>> /usr/lib/ssl/misc/CA.sh -newca
```

- Nos pide el fichero con el certificado de la CA. Si pulsamos "enter" nos creará uno de forma automática (nosotros elegimos esta opción).
- Ahora nos pide la clave para acceder a la clave privada del nuevo certificado que está creando (el certificado de la CA).
- Nos pide el código de país, la provincia, la ciudad, la organización, la unidad organizativa, el nombre y la dirección de correo.

Firmar el CSR

Ya tenemos la CA, ahora firmamos el CSR que habíamos generado en el punto 4. Para esto también usaremos el script CA.sh. Este script trabaja con nombres fijos de ficheros, con lo que espera encontrar el CSR con el nombre "newreq.pem".

Lo que vamos a hacer es copiar nuestro CSR al directorio padre de "demoCA" con el nombre "newreq.pem". Es decir, nuestro CSR debe quedar a la misma altura que el directorio "demoCA", y no dentro de él.

Instalar el certificado de la CA

```
>> keytool -import -alias CAcert -keypass claveCAcert -file demoCA/cacert.pem -storepass claveDeKeyStore
```

Si quisiéramos instalar el certificado de la CA en el cacerts keystore podríamos hacer:

```
>> keytool -import -alias autentiaCaCert -keypass claveDeAutentiaCaCert -file demoCA/cacert.pem -keystore $JAVA_HOME/jre/lib/security/cacerts -storepass claveDeKeyStore
```

Instalar nuestro certificado firmado por la CA

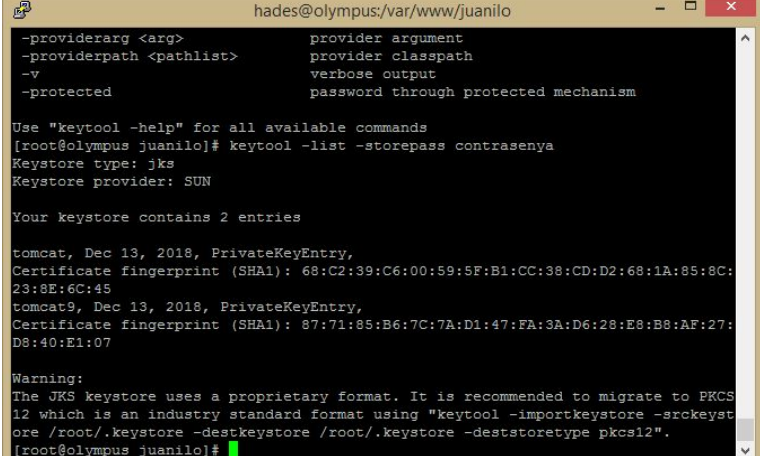
```
keytool -import -alias autentiaCert -keypass claveDeAutentiaCert -file autentiaCertFirmadoPorCA.pem -storepass claveDeKeyStore
```

Ahora podemos explorar nuestro keystore.

Explorar nuestro keystore

Mostrar el contenido actual del keystore

```
>> keytool -list -storepass claveKeyStore
```



```
hades@olympus:/var/www/juanilo
-providerarg <arg>           provider argument
-providerpath <pathlist>     provider classpath
-v                             verbose output
-protected                     password through protected mechanism

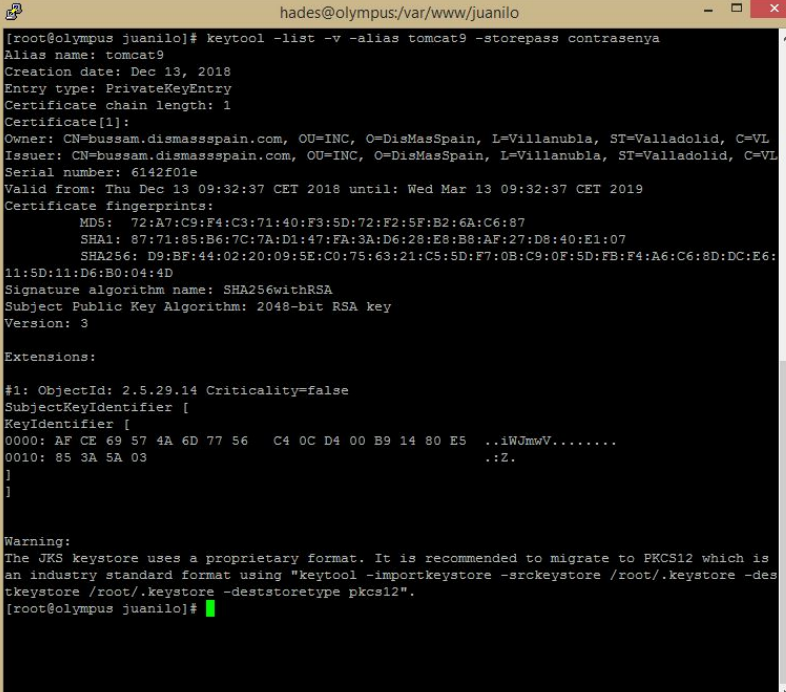
Use "keytool -help" for all available commands
[root@olympus juanilo]# keytool -list -storepass contrasenya
Keystore type: jks
Keystore provider: SUN

Your keystore contains 2 entries

tomcat, Dec 13, 2018, PrivateKeyEntry,
Certificate fingerprint (SHA1): 68:C2:39:C6:00:59:5F:B1:CC:38:CD:D2:68:1A:85:8C:
23:8E:6C:45
tomcat9, Dec 13, 2018, PrivateKeyEntry,
Certificate fingerprint (SHA1): 87:71:85:B6:7C:7A:D1:47:FA:3A:D6:28:E8:B8:AF:27:
D8:40:E1:07

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS
12 which is an industry standard format using "keytool -importkeystore -srckeyst
ore /root/.keystore -destkeystore /root/.keystore -deststoretype pkcs12".
[root@olympus juanilo]#
```

Mostrar el detalle del certificado elegido



```
hades@olympus:/var/www/juanilo
[root@olympus juanilo]# keytool -list -v -alias tomcat9 -storepass contrasenya
Alias name: tomcat9
Creation date: Dec 13, 2018
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=bussam.dissasspain.com, OU=INC, O=DisMasSpain, L=Villanubla, ST=Valladolid, C=VL
Issuer: CN=bussam.dissasspain.com, OU=INC, O=DisMasSpain, L=Villanubla, ST=Valladolid, C=VL
Serial number: 6142F01e
Valid from: Thu Dec 13 09:32:37 CET 2018 until: Wed Mar 13 09:32:37 CET 2019
Certificate fingerprints:
MD5: 72:A7:C9:F4:C3:71:40:F3:5D:72:F2:5F:B2:6A:C6:87
SHA1: 87:71:85:B6:7C:7A:D1:47:FA:3A:D6:28:E8:B8:AF:27:D8:40:E1:07
SHA256: D9:BF:44:02:20:09:5E:C0:75:63:21:CS:5D:F7:0B:C9:0F:5D:FB:F4:A6:C6:8D:DC:E6:
11:5D:11:D6:B0:04:4D
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: AF CE 69 57 4A 6D 77 56 C4 0C D4 00 B9 14 80 E5 ..iWjmwV.....
0010: 85 3A 5A 03 .:2.
]
]

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is
an industry standard format using "keytool -importkeystore -srckeystore /root/.keystore -des
tkeystore /root/.keystore -deststoretype pkcs12".
[root@olympus juanilo]#
```

```
>> keytool -list -v -alias
```

```
alias-storepass claveDeKeyStore
```

Gestionar nuestro keystore

Crear la keyStore

```
keytool -genkey -alias alias -keyalg RSA -keysize 2048 -keystore keystore.jks
```

Cambiar contraseña de la keyStore

```
keytool -storepasswd -keystore keystore.jks
```

Importar certificado

```
keytool -import -trustcacerts -alias alias -keystore keystore.jks
```

Borrar certificado

```
keytool -delete -alias alias -keystore keystore.jks
```