Generating and Converting Certificates with OpenSSL and Keytool

1. Generate RSA 4096-bit Private Key

Use OpenSSL to generate a 4096-bit RSA private key:

```
openssl genpkey -algorithm RSA -out private_key.pem -pkeyopt rsa_keygen_bits:4096
```

2. Generate the Public Key

Extract the public key from the private key:

```
openssl rsa -in private_key.pem -pubout -out public_key.pem
```

3. Generate a Self-Signed Certificate

Create a self-signed certificate valid for 1 year (365 days):

```
openssl req -new -x509 -key private_key.pem -out certificate.pem -days 36500
```

Convert Certificate to Different Formats

5. Convert PEM to PKCS#12 (.p12)

Convert the private key and certificate to a PKCS#12 file:

```
openssl pkcs12 -export -out certificate.p12 -inkey private_key.pem -in certificate.pem -name
"myalias"
```

You will be asked to set a password for the `.p12` file.

6. Convert PKCS#12 (.p12) to Java KeyStore (JKS)

Use the Java Keytool utility to convert `.p12` to `.jks`:

```
keytool -importkeystore -srckeystore certificate.p12 -srcstoretype PKCS12 -srcstorepass
password@123 -destkeystore mykeystore.jks -deststorepass changeit -destkeypass changeit -
alias myalias
```

Parameters:

- **-srckeystore**: The path to the `.p12` file (e.g., `certificate.p12`).
- **-srcstoretype**: Specifies the type of the source keystore, which is **PKCS12** in this case.
- **-srcstorepass**: The password for the source `.p12` file (in this example, `Pass@123`).
- **-destkeystore**: The path where the `.jks` file will be saved (e.g., `mykeystore.jks`).
- **-deststorepass**: The password for the destination `.jks` keystore (in this example, `changeit`).
- **-destkeypass**: The password for the key entry in the JKS keystore (in this example, `changeit`).
- **-alias**: The alias for the key entry in the keystore (e.g., `myalias`).
-

KeyStore Configuration for JWKs in ECRNow:

To configure the JWKS (JSON Web Key Set) with RSA keys stored in a Java KeyStore (`.jks`), you need to specify the file path to the keystore and the password to access it. This configuration is done in the `application.properties` or `application.yml` file.

Configuration Example:

In your `application.properties` or `application.yml`, specify the following properties:

```
# Location of the KeyStore file
jwks.keystore.location=pathtofile/ecrnow.jks

# Password for accessing the KeyStore

jwks.keystore.password=password
```

Summary

- **PEM Files**: Private key (`private_key.pem`), Public key (`public_key.pem`), Certificate (`certificate.pem`).
- **PKCS#12 File**: `.p12` format containing the private key and certificate.
- **Java KeyStore (JKS)**: `.jks` file for Java-based applications.

This document provides a structured approach to generating and converting certificates using OpenSSL and Keytool.