Enable https in spring boot

Enabling https using spring boot for java application running in an embedded tomcat required the following steps:

1. Generate a SSL certificate: generate a self-signed certificate using java keytool or get one from certificate authority.
2. Enable https in spring boot.

# Generate a SSL certificate

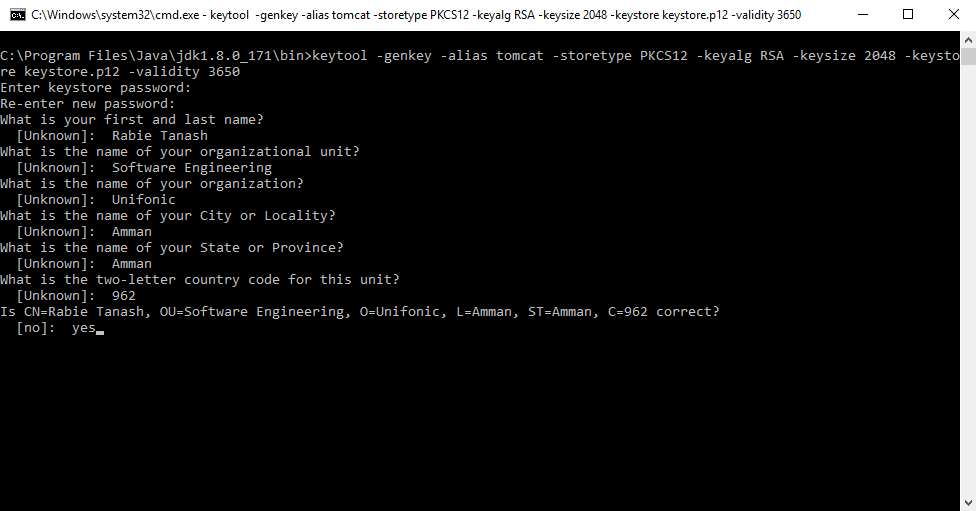
In this section a brief description on how to generate a self-signed certificate is presented.

Every java runtime environment comes bundled with a certificate management tool (keytool).

This can be used to generate our SSL certificate.

In order to start keytool first we open an cmd window and change directory to ${java.home}/jdk/bin.

Then we follow the steps in the below image.



This will generate a PKCS12 keystore called keystore.p12 with a ewly generate certificate in it, with certificate alias tomcat. We will need to reference keystore when we start to configure Spring Boot.

# Enable https in spring boot

By default, spring boot embedded tomcat container will have HTTP on port 8080 enabled. spring boot allow us to configure HTTP or HTTPS in the application.properties.

In order to configure https using spring boot we have to add the following commands to the default application.properties file under src/main/resources of our spring boot application.

server.port: 8443

server.ssl.key-store: keystore.p12

server.ssl.key-store-password: mypassword

server.ssl.keyStoreType: PKCS12

server.ssl.keyAlias: tomcat

That’s all we need to do to make your application accessible over HTTPS on https://localhost:8443.