# **SSL/TLS Configuration for Server-Client Communication**

## **Prerequisite:**

* Install Git Bash
* Java JDK (version 8 at least)

## **Procedure:**

### **Server Configuration:**

1. Open Git Bash in the Resources directory of the Server project.
2. Create a Certificate Authority Private Key:

keytool -genkeypair -alias ca\_alias -keyalg RSA -keysize 2048 -dname "CN=localhost, OU=FSOFT, O=FPT, L=CauGiay, S=HaNoi, C=VN" -ext bc:c -keystore server\_keystore.jks -storepass changeit -keypass changeit

1. Generate a CA certificate:

keytool -export -storepass changeit -alias ca\_alias -keystore server\_keystore.jks -file ca.crt -rfc

1. Create the Server Private Key:

keytool -genkey -alias server -keyalg RSA -keysize 2048 -keystore server\_keystore.jks -dname "CN=localhost, OU=FHN, O=FPT, L=CauGiay, S=HaNoi, C=VN" -storepass changeit -keypass changeit

1. Generate a Certificate Signing Request (CSR) Using Server Private Key:

keytool -certreq -alias server -file csr.txt -keystore server\_keystore.jks -storepass changeit

1. Generate self-signed SSL certificate with CA:

keytool -gencert -alias ca\_alias -infile csr.txt -outfile self\_signed\_server.crt -keystore server\_keystore.jks -storepass changeit -ext KU=digitalSignature,keyEncipherment -ext EKU=serverAuth -rfc

1. Import the self-signed certificate to the Server. The server will use this certificate and it will display on the browser:

keytool -import -alias server -keystore server\_keystore.jks -file self\_signed\_server.crt -storepass changeit

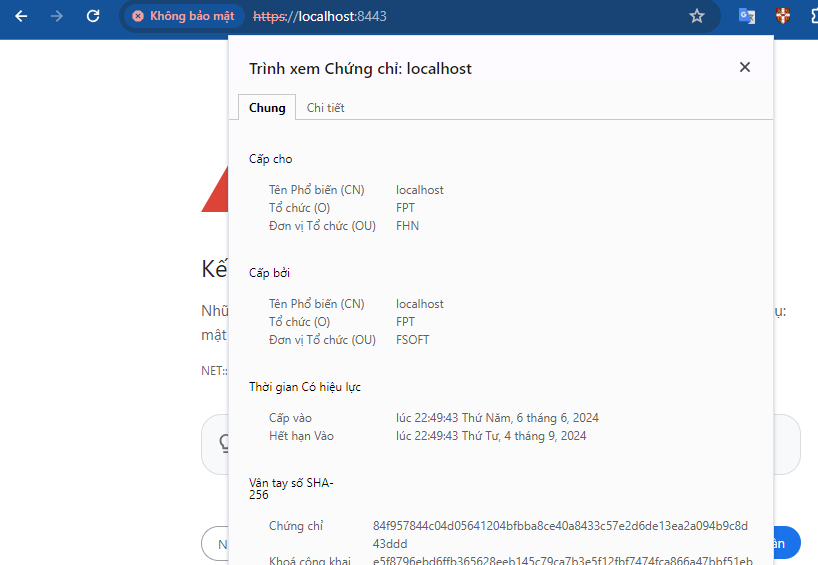
### **Client Configuration:**

1. Copy the CA certificate from the Resources directory of the Server project to the Client project.
2. Open Git Bash in the Resources directory of the Client project.
3. Import the CA certificate to the truststore of the client:

keytool -importcert -alias client -file ca.crt -keystore client\_truststore.jks -storepass changeit -noprompt

### **Running Applications:**

1. Start the Server application.
2. Start the Client application.
3. Access your Server application with URL: [https://localhost:8443](https://localhost:8443/) on the browser, verify if it displays an unsecure cert. Display the cert. The **issued to** should be **FHN**. The **issuer** should be **FSOFT**.



1. Access your Client application with URL: [http://localhost:8080](http://localhost:8080/) on the browser, verify if it shows "You have successfully configured SSL for your Java Application!". It indicates your Client has communicated successfully with your Server using SSL/TLS.

