Generating Sign-Signed Ceritificates

For creating certificates to test SSL on your development server.

Step-by-step guide

1. Run the following script: generate-certs.bat
    a. I got it from http://www.chesterproductions.net.nz/blogs/it/code/configuring-client-certificate-authentication-with-tomcat-and-java/537/
    b. It creates a "server" keystore and a "client" keystore.
        i. Later add the server keystore as the keystore and truststore in the SSLConnector in Tomcat's server.xml.
    c. Imports each into the other.
    d. Then uses the client keystore to generate a client cert.
        i. Later, you import that client cert into the browser as a personal cert.
2. Below is the SSL Connector config in Tomcat's server.xml:

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true" scheme="https" secure="true"
           clientAuth="want" sslProtocol="TLS"
           keystoreFile="certs/server.jks"
           keystoreType="JKS" keystorePass="password"
           truststoreFile="certs/server.jks"
           truststoreType="JKS" truststorePass="password"/>
```

3. I imported the client.p12 cert into the browser as a personal cert.
4. I imported the client and server keystores into the browser as Trusted Root Stores.
5. After restarting both the server and the browser, Spring Security using x509 worked as expected.

Related articles

- Generating Sign-Signed Ceritificates
- Spring Security with Certificates