Tomcat 和 android 双向 SSL 通信

一、预备

1. 基本的 JDK 环境（略）
2. 从 tomcat 官网下载 tomcat7，解压到指定的目录，我们以%TOMCAT_HOME%代替，执行命令%TOMCAT_HOME%/bin/start/startup.bat，确保首页 http://127.0.0.1:8080 可访问
3. Tomcat 的 SSL 支持，在%TOMCAT_HOME%/conf/server.xml 中去掉 connect 端口为 8443 的注释，代码如下：

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol"
           maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS" />
```

   重启 tomcat，访问 https://127.0.0.1:8443
4. 上面一切准备就绪，开始准备 Android 上的支持，在任意的 maven 项目中增加下面的依赖，确保后续生成 BKS 证书

```
<dependency>
    <groupId>org.bouncycastle</groupId>
    <artifactId>bcmail-jdk16</artifactId>
    <version>1.46</version>
</dependency>
```

5. 根据第四步自动下载的两个 jar 包，将其放置到%JAVA_HOME%/jre/lib/ext 目录下，避免在执行对应命令的时候加载特定的 jar 而执行过多的 classpath 参数
6. 修改 java 安全配置文件，在%JAVA_HOME%/jre/lib/security/java.security 中新增安全配置规则

```
#
security.provider.1=sun.security.provider.Sun
security.provider.2=sun.security.rsa.SunRsaSign
security.provider.3=sun.security.ec.SunEC
security.provider.4=com.sun.net.ssl.internal.ssl.Provider
security.provider.5=com.sun.crypto.provider.SunJCE
security.provider.6=sun.security.jgss.SunProvider
security.provider.7=com.sun.security.sasl.Provider
security.provider.8=org.jcp.xml.dsig.internal.dom.XMLDSigRI
security.provider.9=sun.security.smartcardio.SunPCSC
security.provider.10=sun.security.mscapi.SunMSCAPI
security.provider.11=org.bouncycastle.jce.provider.BouncyCastleProvider
#
```

   数字从上往下递增

   bks支持
7. 验证上述配置是否有效，编写 windows 下的批处理命令，详情如下

```
cmd /k keytool -genkey -v -alias yitaoKey  -dname "CN=XuYiTao" -keyalg RSA -keypass admin123 -storepass admin123
-keystore yitao.bks -validity 3600 -storetype BKS -provider org.bouncycastle.jce.provider.BouncyCastleProvider
```

   如果代码没有问题继续往下走，如果代码出现问题请直接进 oracle 官网 Java 下载页面，在右上角的搜索框中输入 JCE 7，找到和你版本对应的两个 jar 包 local_policy.jar 和 US_export_policy.jar，覆盖掉您自己安装的%JAVA_HOME%/jre/lib/security 路径下的文件，仍有问题，联系我。

到此，基本环境配置完毕

二、Tomcat 的 PC 版 SSL 双向配置

1、生成服务端的公钥和私钥

```
F:\authen>keytool -genkey -v -alias bosServerKey  -dname "CN=bosServerKey" -key
lg RSA -keypass chaseecho -storepass chaseecho -keystore server.keystore -valid
ty 3600
正在为以下对象生成 2,048 位RSA密钥对和自签名证书 (SHA256withRSA) (有效期为 3,60
 天):
        CN=bosServerKey
[正在存储server.keystore]
```

2、 生成客户端的公钥和私钥

```
F:\authen>keytool -genkey -v -alias clientKey  -dname "CN=bosClientKey"   -keyal
g RSA -keypass admin123 -storepass admin123 -keystore client.p12 -validity 3600
-storetype PKCS12
正在为以下对象生成 2,048 位RSA密钥对和自签名证书 (SHA256withRSA) (有效期为 3,600
 天):
        CN=bosClientKey
[正在存储client.p12]
```

3、 由客户端导出证书

```
F:\authen>keytool -export -alias clientKey -file clientKey.cer -keystore client.
p12 -storepass admin123 -storetype PKCS12
存储在文件 <clientKey.cer> 中的证书
```

4、 客户端的证书加入服务端信任列表，命令最后需要你输入一个字母【y】

```
F:\authen>keytool -import -v -alias clientKey -file clientKey.cer -keystore serv
er.keystore -storepass chaseecho
所有者: CN=bosClientKey
发布者: CN=bosClientKey
```

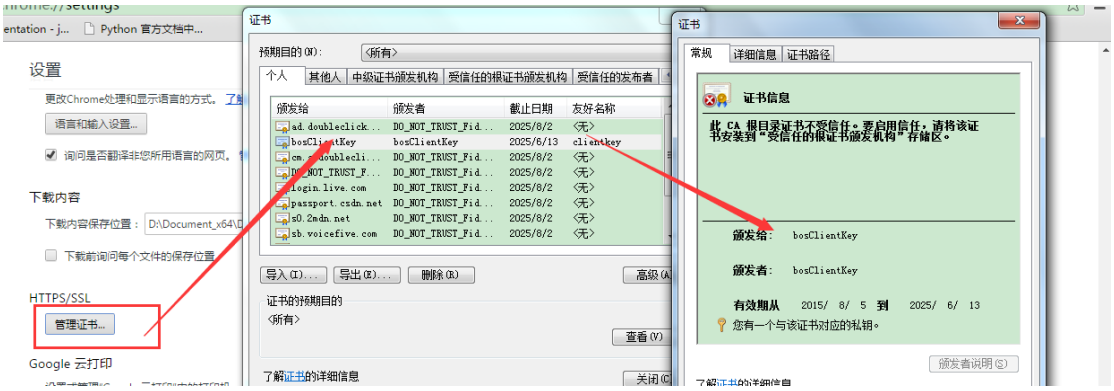
5、 将上文生成的 文件【server.keystore】放入目录%TOMCAT_HOME%/

6、 修改%TOMCAT_HOME%/conf/server.xml 如下所示，两个文件一样，互相信任

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol"
           maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS" keystoreFile="server.keystore" keystorePass="chaseecho"
           truststoreFile="server.keystore" truststorePass="chaseecho" />
```

重启 tomcat，访问 https://127.0.0.1:8443，发现是不可访问的

7、 双击上文生成的证书【client.p12】，进行安装，中间需要输入密码【admin123】，一路安装下去即可，建议安装到【个人】目录下，便于删除，在 chrome 浏览器选择【设置】，在高级配置里进行如下查看



8、 再次刷新页面就会跳出证书的选择页面，选择对应的证书之后即可进行正常访问。

9、 编写程序进行访问，基本代码如下：

加载证书

```
KeyStore ks = KeyStore.getInstance("PKCS12");
FileInputStream instream = new FileInputStream(new File(KS_FILE));
try {
    ks.load(instream, KS_PASSWORD.toCharArray());
} finally {
    instream.close();
}
SSLSocketFactory ssf = null;
SSLContext ctx = SSLContext.getInstance("TLS");
KeyManagerFactory kmf = KeyManagerFactory.getInstance("SunX509");
TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509");
kmf.init(ks, KS_PASSWORD.toCharArray());
KeyStore tks = KeyStore.getInstance("JKS");
tks.load(new FileInputStream(TRUST_FILE), TRUST_PASSWORD.toCharArray());
tmf.init(tks);
ctx.init(kmf.getKeyManagers(), tmf.getTrustManagers(), new SecureRandom());
```

握手访问

```
// //////////////////
ssf = ctx.getSocketFactory();
SSLSocket socket = (SSLSocket) ssf.createSocket(HOST_ADDR, 8443);
System.out.println("create socket success.");
// handshake
socket.startHandshake();
System.out.println("handshake success.");

// //////////////////////
URL url = new URL("https://" + HOST_ADDR + ":8443/taxi-restapi/sys/config");

HttpsURLConnection urlConnection = (HttpsURLConnection) url.openConnection();
urlConnection.setSSLSocketFactory(ctx.getSocketFactory());
InputStream input = urlConnection.getInputStream();

BufferedReader reader = new BufferedReader(new InputStreamReader(input, "UTF-8"));
StringBuffer result = new StringBuffer();
String line = "";
while ((line = reader.readLine()) != null) {
    result.append(line);
}
```

10、 以上是完整步骤，为客户端颁发更多的证书，将对应证书加入服务端信任列表即可。

三、Tomcat 的 Android 版 SSL 双向配置

1. 生成服务端的公钥和私钥

```
keytool -genkey -v -alias bosServerKey  -dname "CN=10.8.200.62" -keyalg RSA -keypass chaseecho -storepass chaseecho
-keystore server.keystore -validity 3600
```

2. 生成服务端证书

```
keytool -export -v -alias bosServerKey -keystore server.keystore -storepass chaseecho -rfc -file server.crt
```

3. 生成客户端公钥和私钥，这里是 BKS 格式的

```
keytool -genkey -v -alias yitaoKey  -dname "CN=XuYiTao" -keyalg RSA -keypass admin123 -storepass admin123
-keystore yitao.bks -validity 3600 -storetype BKS -provider org.bouncycastle.jce.provider.BouncyCastleProvider
```

4. 导出客户端证书

```
keytool -export -v -alias yitaoKey -keystore yitao.bks  -storepass admin123  -file yitao.crt -storetype BKS
-provider org.bouncycastle.jce.provider.BouncyCastleProvider
```

5. 服务端证书加入客户端信任列表

```
keytool -import -alias bosServerKey -keystore trust.bks -file server.crt -storepass chaseecho -storetype BKS
-provider org.bouncycastle.jce.provider.BouncyCastleProvider
```

6. 客户端证书导入服务端信任列表

```
keytool -import -v -alias yitaoKey -file yitao.crt -keystore server.keystore -storepass chaseecho
```

7. 将上述两个 bks 文件发给安卓客户端，听天由命。