应用系统体系结构作业4

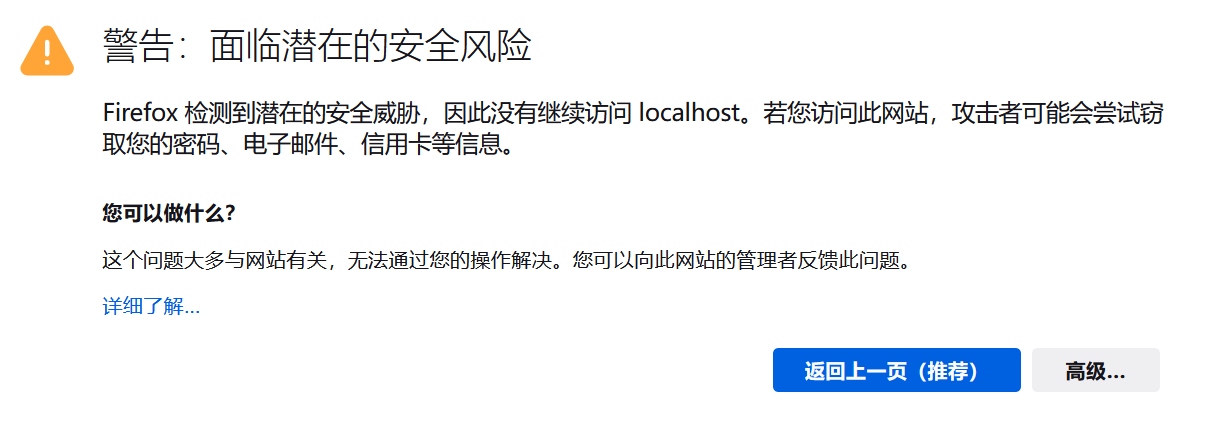
1. 我采用的是将**所有的访问都重构为HTTPS通信**的方式。
2. 首先我在Chrome浏览器搜索栏栏中输入实例请求:http://localhost:8080/login?username=Yhan&password=@abcABC12345:



之后按下回车，会将请求访问的接口重定向到https的8443接口，并会正确显示响应信息：(前提是证书已经被信任)：

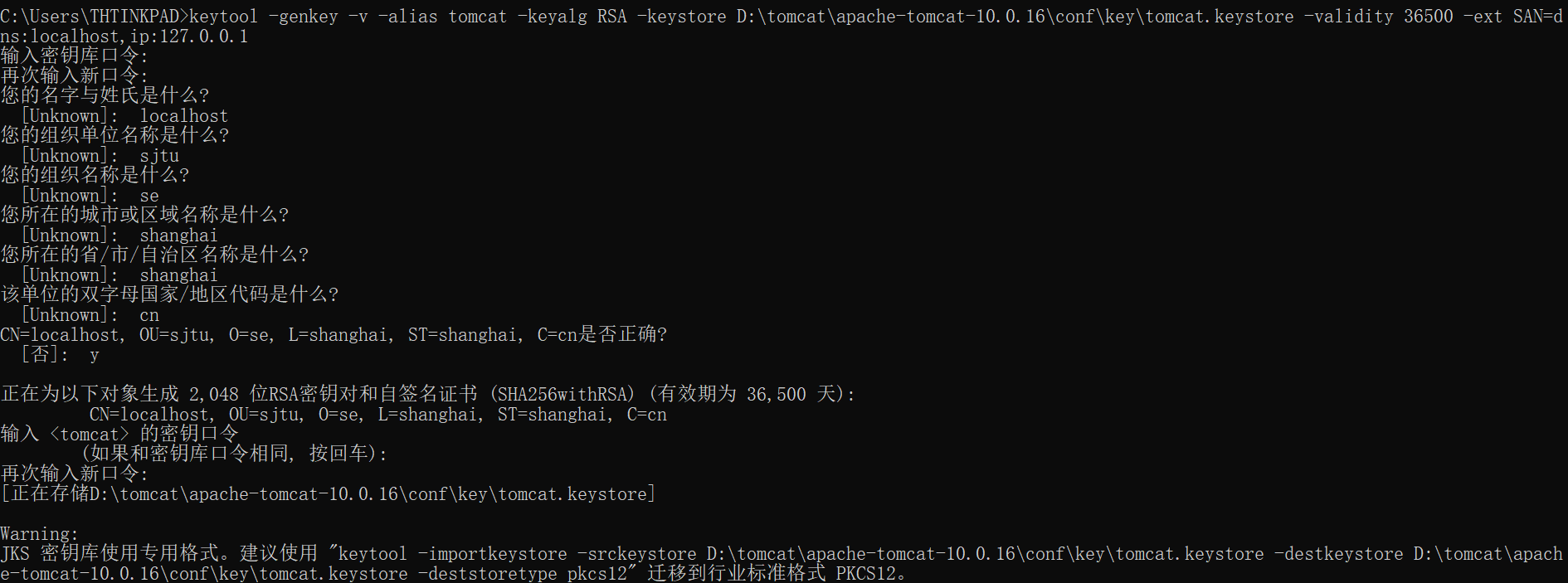


但是如果证书没有被信任，就会出现：

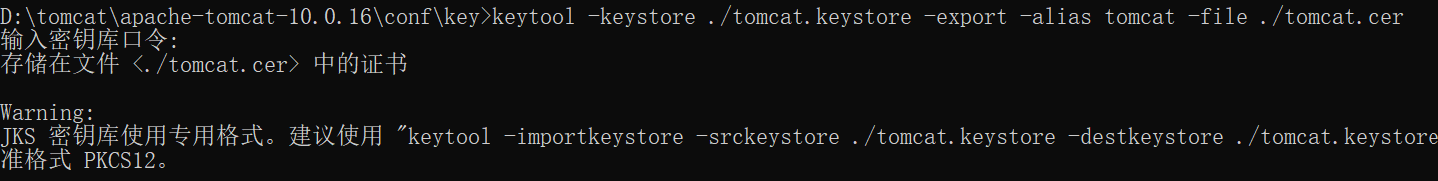


1. **关于HTTP与HTTPS运行时区别的解释**：
2. 发送http请求时页面响应速度要快于发送https请求时页面响应速度。这是因为https通信时需要进行额外的证书认证(即利用密钥进行加解密等操作)，进行额外的tcp握手等操作，从而使得https响应速度相对于http更慢。
3. http在tomcat中默认端口号为8080，https在tomcat中默认端口号为8443。
4. **为何一定要导入证书**：在进行HTTPS请求时，需要认证公有密钥的正确性。因而，服务器端会将公有密钥先发到第三方的数字认证机构，之后认证机构利用自己的私有秘钥对服务器端的公有密钥进行加密，之后将加密的公有密钥传递给客户端，这个加密之后的公钥就是证书。之后客户端使用数字认证机构的公有密钥进行解密来获取数字机构传输过来的服务器端的公有密钥，并通过比较来确定正确性。之后使用这个公有密钥加密信息来传输给服务器端。之后服务器端就可以使用自己的私有秘钥进行解密来获取传递过来的信息，这样就保证了信息传递的安全性，所以需要引入证书。
5. **证书相关操作流程**：

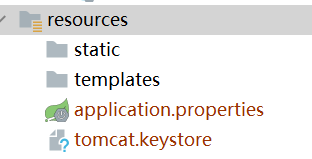
首先使用指令：keytool -genkey -v -alias (别名) -keyalg RSA -keystore (keystore路径) -validity 36500 -ext SAN=dns:localhost,ip:127.0.0.1来在指定路径下生成一个密钥，并将其保存在对应的keystore中。



之后使用指令keytool -keystore (第一步使用的keystore路径) -export -alias (第一步生成的别名) -file (证书路径)导出新生成的证书：



之后，将第一步生成的证书放入后端项目src文件夹下的resources文件夹下：



之后，在第二步生成的cer文件对应路径下，双击cer文件，进行证书的安装，将证书安装到“受信任的颁发者根目录”。



后端代码修改处截图：

在项目启动类中添加：



在application.properties中添加：

