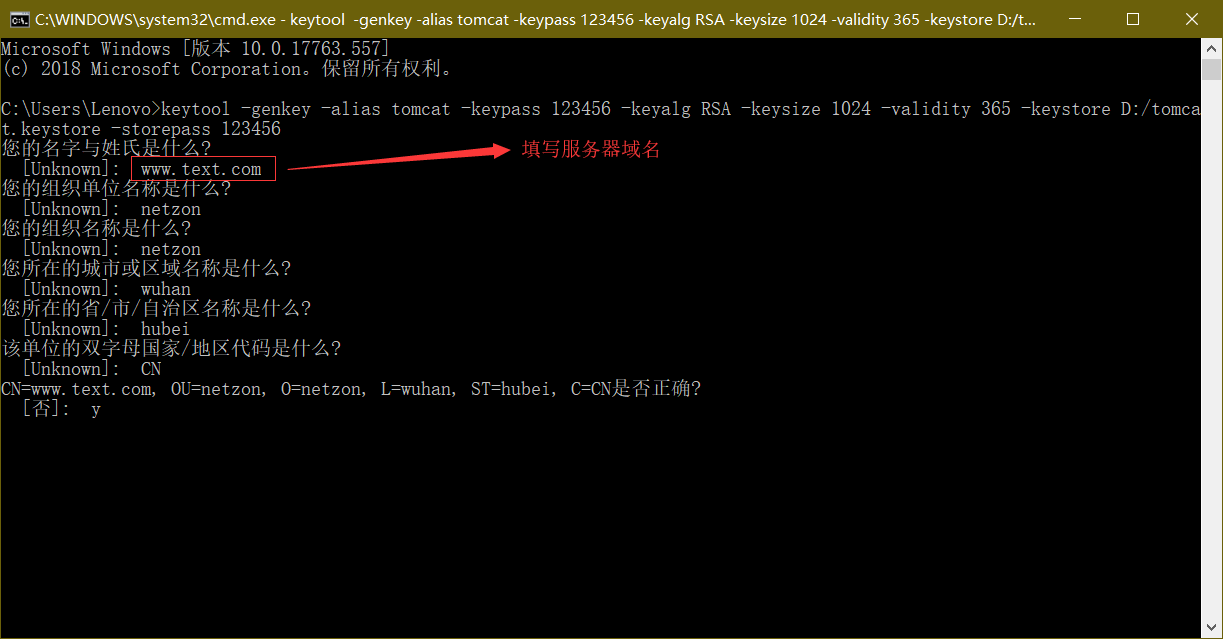
HTTPS配置

1. win + R打开cmd客户端，进入jdk的bin目录（如已配置jdk环境变量，则跳过改步骤）
2. 生成证书

-- 生成服务器证书

keytool -genkey -alias tomcat -keypass 123456 -keyalg RSA -keysize 1024 -validity 365 -keystore D:/tomcat.keystore -storepass 123456



**注：“您的名字与姓氏是什么？”请填写服务器域名**

-- 客户端生成证书

keytool -genkey -alias client -keypass 123456 -keyalg RSA -keysize 1024 -validity 365 -storetype PKCS12 -keystore D:/client.p12 -storepass 123456

-- 客户端导出为cer

keytool -export -alias client -keystore D:/client.p12 -storetype PKCS12 -keypass 123456 -file D:/client.cer

-- 导入到服务器的证书库

keytool -import -v -file D:/client.cer -keystore D:/tomcat.keystore -storepass 123456

-- 让客户端信任服务器证书

keytool -keystore D:/tomcat.keystore -export -alias tomcat -file D:/server.cer

1. tomcat配置HTTPS

复制D:/server.cer到tomcat的conf目录下，修改server.xml：

<Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol" SSLEnabled="true" maxThreads="150" scheme="https" secure="true"

clientAuth="false" sslProtocol="TLS" keystoreFile="conf/tomcat\_ali.keystore" keystorePass="123456"/>

1. 自动跳转https

在conf/web.xml添加

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

</welcome-file-list>

<security-constraint>

<web-resource-collection>

<web-resource-name>SSL</web-resource-name>

<url-pattern>/\*</url-pattern>

</web-resource-collection>

<user-data-constraint>

<transport-guarantee>CONFIDENTIAL</transport-guarantee>

</user-data-constraint>

</security-constraint>

如图：

