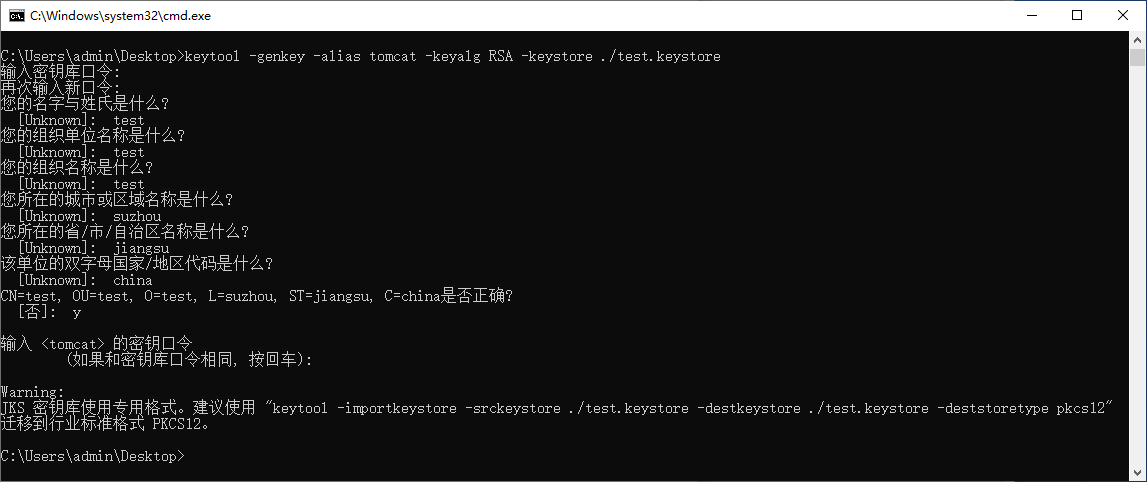
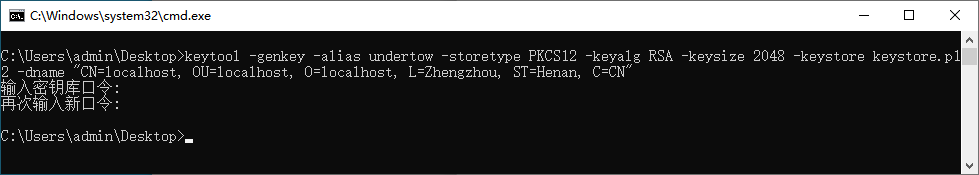
# HTTPS

keytool -genkey -alias tomcat -keyalg RSA -keystore ./test.keystore



# **HTTP2.0**

keytool -genkey -alias undertow -storetype PKCS12 -keyalg RSA -keysize 2048 -keystore keystore.p12 -dname "CN=localhost, OU=localhost, O=localhost, L=Zhengzhou, ST=Henan, C=CN"



**注：正式部署的话可能会报找不到证书的错误，所以一般推荐把证书放到与部署jar包所在的目录中去**

详细步骤：

步骤1：

以管理员方式打开cmd

步骤2：

keytool -genkey -alias server -keyalg RSA -storetype PKCS12 -keystore serverkeystore.p12 -ext SAN=dns:server,ip:127.0.0.1

密码：test123

确认密码：test123

余下全部：test

步骤3：

keytool -export -alias server -file servercert.cer -keystore serverkeystore.p12

步骤4：

keytool -import -alias server -keystore "%JAVA\_HOME%/jre/lib/security/cacerts" -file servercert.cer

密码：**changeit**

列出证书：keytool -list -keystore cacerts

删除证书：keytool -delete -alias server -keystore cacerts

步骤5：

eureka.instance.secure-port-enabled=true

eureka.instance.non-secure-port-enabled=false

eureka.instance.status-page-url=https://${eureka.instance.hostname}:${server.port}/info

eureka.instance.health-check-url=https://${eureka.instance.hostname}:${server.port}/health

eureka.instance.home-page-url=https://${eureka.instance.hostname}:${server.port}/

server.ssl.key-store=serverkeystore.p12

server.ssl.key-store-password=test123

server.ssl.key-alias=server

server.ssl.key-store-type=PKCS12

server.tomcat.uri-encoding=UTF-8

实际使用经验：多个微服务下尽量保持域名一样，端口不一样

**注：目前实际项目较常见的做法是：用户看到的是HTTPS请求，实际该请求是转到如nginx上，最后请求真正后端服务器走的还是HTTP。另外，目前主流还是HTTPS**