## 使用JDK自带的 keytool 工具生成公私钥证书库：

假如我们设置公钥库密码为：public\_password1234，私钥库密码为：private\_password1234，则生成命令如下：

#生成命令

keytool -genkeypair -keysize 1024 -validity 3650 -alias "privateKey" -keystore "privateKeys.keystore" -storepass "public\_password1234" -keypass "private\_password1234" -dname "CN=localhost, OU=localhost, O=localhost, L=SH, ST=SH, C=CN"

#导出命令

keytool -exportcert -alias "privateKey" -keystore "privateKeys.keystore" -storepass "public\_password1234" -file "certfile.cer"

#导入命令

keytool -import -alias "publicCert" -file "certfile.cer" -keystore "publicCerts.keystore" -storepass " public\_password1234"



上述命令执行完成之后，会在当前路径下生成三个文件，分别是：privateKeys.keystore、publicCerts.keystore、certfile.cer。其中文件certfile.cer不再需要可以删除，文件privateKeys.keystore用于当前的 授权服务端项目给客户生成license文件，而文件publicCerts.keystore则随应用代码部署到客户服务器，用户解密license文件并校验其许可信息。

## 服务端生成参数介绍

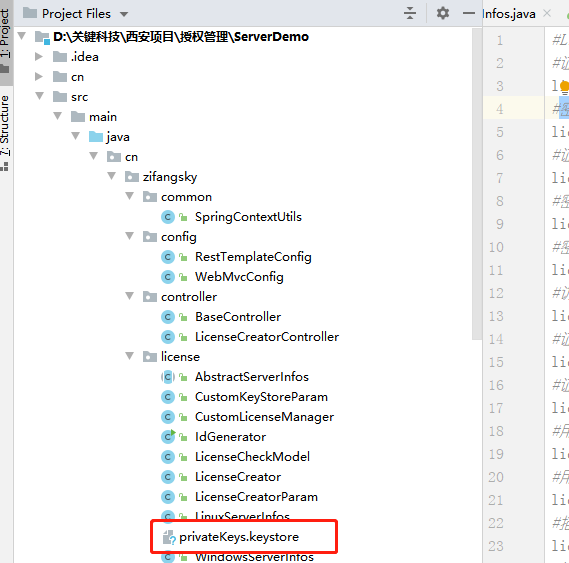
### license.licensePath

证书生成路径地址 如：license.licensePath=C:/shouquan/license.lic

### license.priPath

密钥库存储路径 如：license.priPath=privateKeys.keystore

下图



### license.subject

证书subject 相当于key 如：license.subject=license\_demo

注：当前参数可以作为我们不同项目的标识。可以利用他 进行区分

### license.privateAlias

密钥别称 license.privateAlias=privateKey

固定不变，他对应是下方如图



### license.keyPass

秘钥密码 license.keyPass=private\_password1234

对应jdk生成公私钥时设置的密码如：



注：一般*需要妥善保管，不能让使用者知道*；

### license.storePass

访问秘钥库的密码 license.storePass=public\_password1234

对应jdk生成时设置的密码如：



### license.startTime license.endTime

证书开始时间 license.startTime=2022-03-21 00:00:01

证书结束时间 license.endTime=2022-04-25 00:00:01

注：这个开始时间是给证书设置了一个范围，就是客户端使用时，当前时间在这个范围时，时间生效。是一层检验而已，配合后面的license.day参数使用。

### license.day

证书授权时常 以天为单位 license.day=10

注：就是我们服务端生成的lic文件后，拿着lic文件通过客户端导入，以当前上传lic的时间为准，根据license.day的天数计算，得到一个lic 失效的时间。从而控制。

### license.consumerType

用户类型：license.consumerType=User

注：这个值固定不动，用户标识，配合下方的 license.consumerAmount 用户个数（个数必须大于1 ） license.description 描述信息

### license.consumerAmount

用户个数：license.consumerAmount=1

注：这个是授权时，设置一个当前用户最多使用的个人。然后在客端根据这个用户个数：比如通过seesoon 判断当前用户在线人数，结合这个用户个数做一些事。在逻辑中，用户个数不能小于0，负责客户端检验就会报错。

### license.description

描述信息：license.description=描述信息

注：这个参数主要是在生成授权文件时，扩展的字段，如可以做一些授权信息的介绍，然后在客户端，解析这个描述信息，做个提示等。当前参数可以为空不写。

### license.cpuSerial

可被允许的CPU序列号 license.cpuSerial=

注：这个可以绑定机子的序列号，然后控制客户端机子，必须是为的当前序列号的机子，才可以操作。默认为空，根据实际需求填写

### license.macAddress

可被允许的MAC地址 license.macAddress=

注：同CPU序列号一样。

### license.mainBoardSerial

可被允许的主板序列号 license.mainBoardSerial=

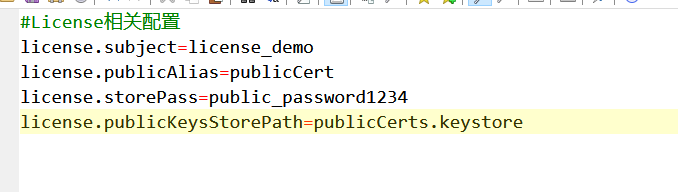
注：同CPU序列号一样。

### license.ips

可被允许的IP地址 license.ips=192.168.31.12

注：同CPU序列号一样

## 客户端解析参数介绍



### license.subject

证书subject 相当于key 如：license.subject=license\_demo

注：当前参数可以作为我们不同项目的标识。可以利用他 进行区分

### license.publicAlias

公钥别称 license.publicAlias=publicCert

固定不变，他对应是下方如图



### license.storePass

访问秘钥库的密码 license.storePass=public\_password1234

对应jdk生成时设置的密码如：



### license.publicKeysStorePath

密钥库存储路径 license.publicKeysStorePath=publicCerts.keystore

