秘钥生成文档

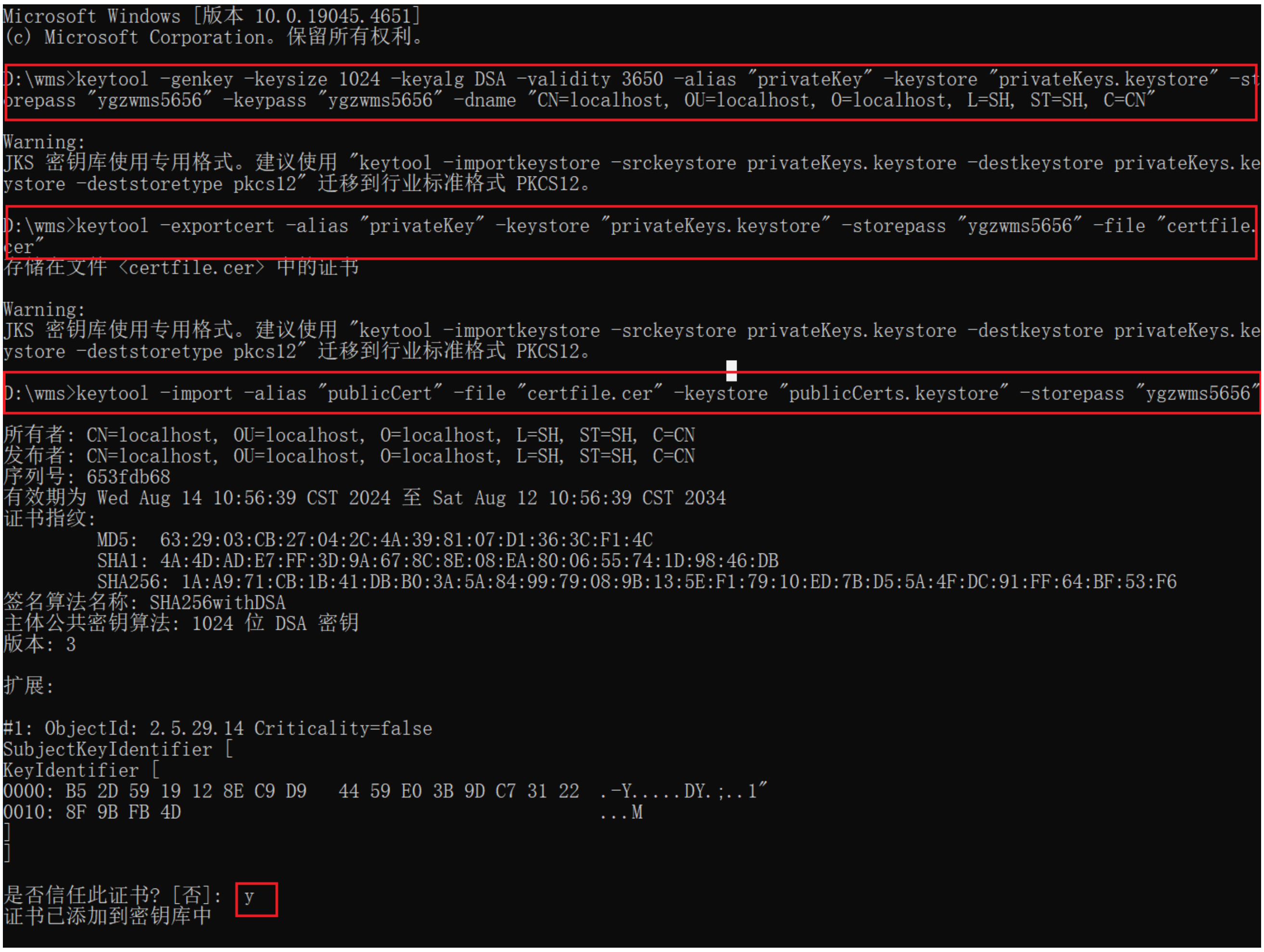
写在前面

> 1. 首先在系统中生成三个文件
> 2. 其次运行 `lisence` 项目生成 `lisence.lic` 证书
> 3. 集成主项目（这一步已经集成了），主要修改主项目文件路径，对比是否与前面生成的文件路径一样

例如在D:/wms/文件下 `cmd` 依次运行下面三行指令，生成三个文件

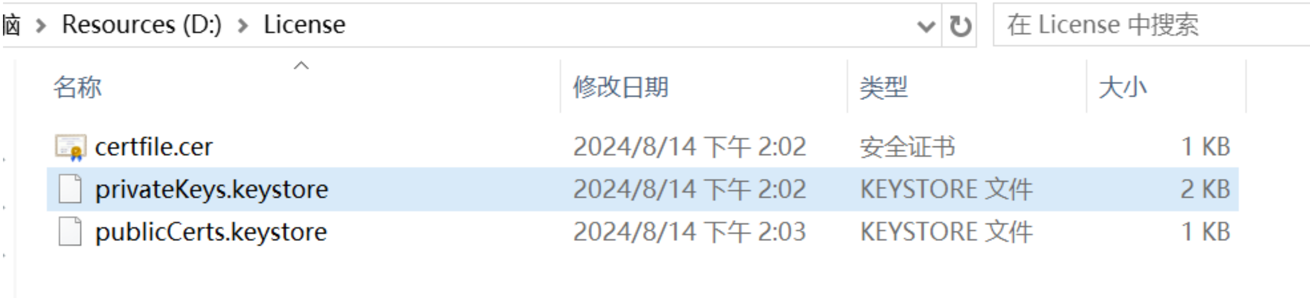
```
keytool -genkey -keysize 1024 -keyalg DSA -validity 3650 -alias "privateKey" -keystore "privateKeys.keystore" -storepass "ygzwms5656" -keypass "ygzwms5656" -dname "CN=localhost, OU=localhost, O=localhost, L=SH, ST=SH, C=CN"

keytool -exportcert -alias "privateKey" -keystore "privateKeys.keystore" -storepass "ygzwms5656" -file "certfile.cer"

keytool -import -alias "publicCert" -file "certfile.cer" -keystore "publicCerts.keystore" -storepass "ygzwms5656"
```

> 然后可以看到以下三个文件：
>
> `privateKeys.keystore` （私钥）提供给生成证书使用（自己保留）
>
> `publicCerts.keystore` （公钥）提供给证书认证使用（给客户使用）
>
> `certfile.cer` 后续步骤用不到，可以删除。



```
Microsoft Windows [版本 10.0.19045.4651]
(c) Microsoft Corporation。保留所有权利。

D:\wms>keytool -genkey -keysize 1024 -keyalg DSA -validity 3650 -alias "privateKey" -keystore "privateKeys.keystore" -st
orepass "ygzwms5656" -keypass "ygzwms5656" -dname "CN=localhost, OU=localhost, O=localhost, L=SH, ST=SH, C=CN"

Warning:
JKS 密钥库使用专用格式。建议使用 "keytool -importkeystore -srckeystore privateKeys.keystore -destkeystore privateKeys.ke
ystore -deststoretype pkcs12" 迁移到行业标准格式 PKCS12。

D:\wms>keytool -exportcert -alias "privateKey" -keystore "privateKeys.keystore" -storepass "ygzwms5656" -file "certfile.
cer"
存储在文件 <certfile.cer> 中的证书

Warning:
JKS 密钥库使用专用格式。建议使用 "keytool -importkeystore -srckeystore privateKeys.keystore -destkeystore privateKeys.ke
ystore -deststoretype pkcs12" 迁移到行业标准格式 PKCS12。

D:\wms>keytool -import -alias "publicCert" -file "certfile.cer" -keystore "publicCerts.keystore" -storepass "ygzwms5656"

所有者: CN=localhost, OU=localhost, O=localhost, L=SH, ST=SH, C=CN
发布者: CN=localhost, OU=localhost, O=localhost, L=SH, ST=SH, C=CN
序列号: 653fdb68
有效期为 Wed Aug 14 10:56:39 CST 2024 至 Sat Aug 12 10:56:39 CST 2034
证书指纹:
         MD5:  63:29:03:CB:27:04:2C:4A:39:81:07:D1:36:3C:F1:4C
         SHA1: 4A:4D:AD:E7:FF:3D:9A:67:8C:8E:08:EA:80:06:55:74:1D:98:46:DB
         SHA256: 1A:A9:71:CB:1B:41:DB:B0:3A:5A:84:99:79:08:9B:13:5E:F1:79:10:ED:7B:D5:5A:4F:DC:91:FF:64:BF:53:F6
签名算法名称: SHA256withDSA
主体公共密钥算法: 1024 位 DSA 密钥
版本: 3

扩展:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: B5 2D 59 19 12 8E C9 D9   44 59 E0 3B 9D C7 31 22  .-Y.....DY.;..1"
0010: 8F 9B FB 4D                                        ...M
]
]

是否信任此证书? [否]:  y
证书已添加到密钥库中
```

这是生成的三个文件



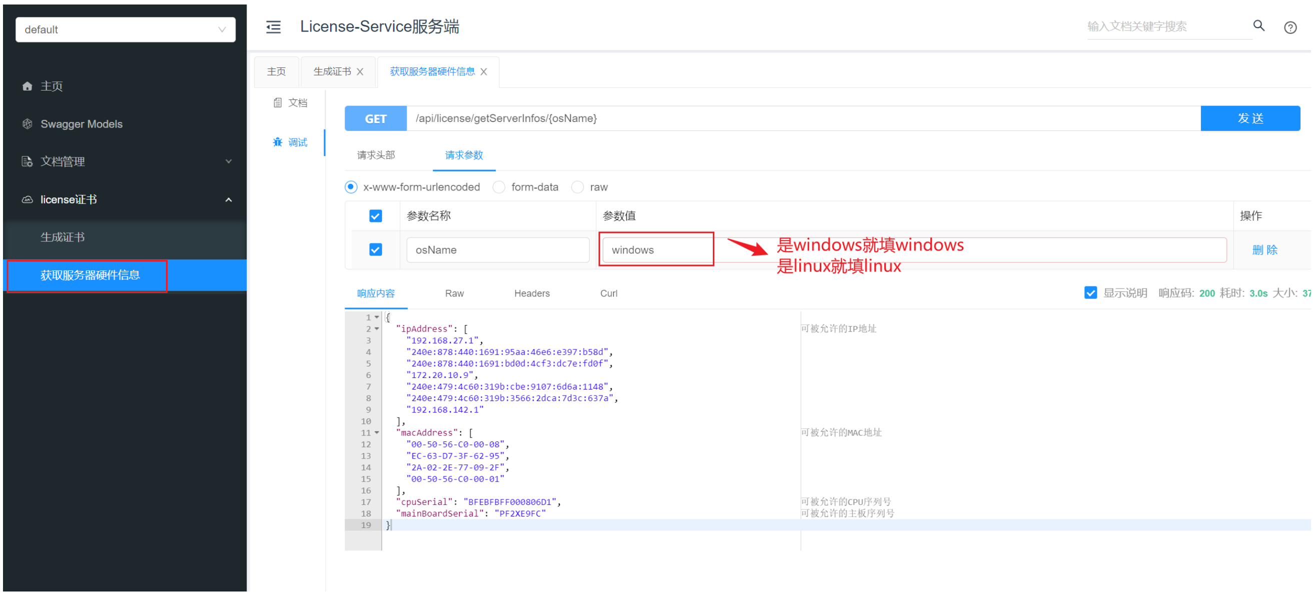
名称	修改日期	类型	大小
certfile.cer	2024/8/14 下午 2:02	安全证书	1 KB
privateKeys.keystore	2024/8/14 下午 2:02	KEYSTORE 文件	2 KB
publicCerts.keystore	2024/8/14 下午 2:03	KEYSTORE 文件	1 KB

打开 `springboot-license` 项目进行运行

修改 `application.yml` 文件中的文件路径

```
license:
  # 这是存放的路径， linux需要修改，例如改成/wms/**
  licensePath: D:/wms/license.lic
```

打开网址 `http://localhost:8099/api/doc.html` 获取服务器硬件信息



GET /api/license/getServerInfos/{osName}

● x-www-form-urlencoded ○ form-data ○ raw

☑	参数名称	参数值	操作
☑	osName	windows	删 除

是windows就填windows
是linux就填linux

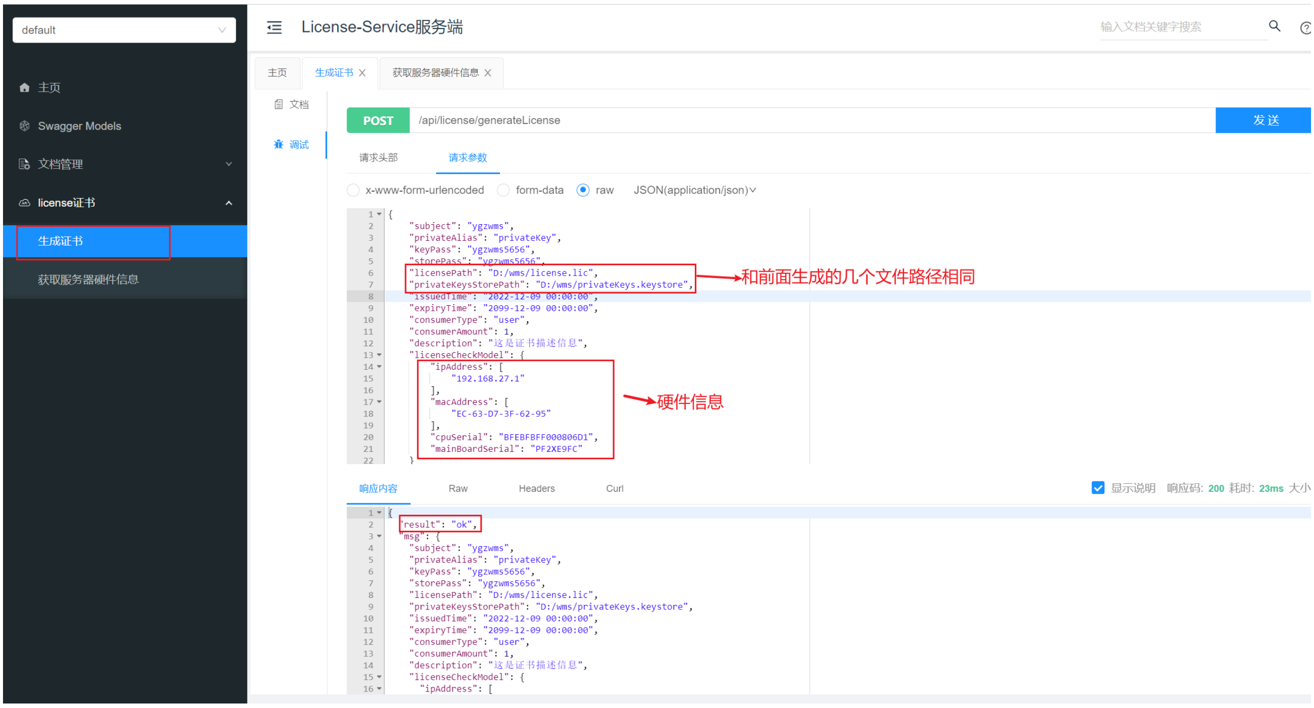
☑ 显示说明 响应码: 200 耗时: 3.0s 大小: 37

```
{
  "ipAddress": [                                   可被允许的IP地址
    "192.168.27.1",
    "240e:878:440:1691:95aa:46e6:e397:b58d",
    "240e:878:440:1691:bd0d:4cf3:dc7e:fd0f",
    "172.20.10.9",
    "240e:479:4c60:319b:cbe:9107:6d6a:1148",
    "240e:479:4c60:319b:3566:2dca:7d3c:637a",
    "192.168.142.1"
  ],
  "macAddress": [                                  可被允许的MAC地址
    "00-50-56-C0-00-08",
    "EC-63-D7-3F-62-95",
    "2A-02-2E-77-09-2F",
    "00-50-56-C0-00-01"
  ],
  "cpuSerial": "BFEBFBFF000806D1",                 可被允许的CPU序列号
  "mainBoardSerial": "PF2XE9FC"                    可被允许的主板序列号
}
```

根据获取的硬件信息生成 `license.lic` 证书

> 以下参数需修改
>
> 1. licensePath 和 privateKeysStorePath 位置
> 2. licenseCheckModel中四个参数的信息，必须修改

```
# windows环境
{
    "subject": "ygzwms",
    "privateAlias": "privateKey",
    "keyPass": "ygzwms5656",
    "storePass": "ygzwms5656",
    "licensePath": "D:/wms/license.lic",
    "privateKeysStorePath": "D:/wms/privateKeys.keystore",
    "issuedTime": "2022-12-09 00:00:00",
    "expiryTime": "2099-12-09 00:00:00",
    "consumerType": "user",
    "consumerAmount": 1,
    "description": "这是证书描述信息",
    "licenseCheckModel": {
        "ipAddress": [
            "192.168.27.1"
        ],
        "macAddress": [
            "EC-63-D7-3F-62-95"
        ],
        "cpuSerial": "BFEBFBFF000806D1",
        "mainBoardSerial": "PF2XE9FC"
    }
}

# linux环境
{
    "subject": "ygzwms",
    "privateAlias": "privateKey",
    "keyPass": "ygzwms5656",
    "storePass": "ygzwms5656",
    "licensePath": "/wms/license.lic",
    "privateKeysStorePath": "/wms/privateKeys.keystore",
    "issuedTime": "2022-12-09 00:00:00",
    "expiryTime": "2099-12-09 00:00:00",
    "consumerType": "user",
    "consumerAmount": 1,
    "description": "这是证书描述信息",
    "licenseCheckModel": {
        "ipAddress": [
            "192.168.27.1"
        ],
        "macAddress": [
            "EC-63-D7-3F-62-95"
        ],
        "cpuSerial": "BFEBFBFF000806D1",
        "mainBoardSerial": "PF2XE9FC"
    }
}
# linux
{
    "subject": "ygzwms",
    "privateAlias": "privateKey",
    "keyPass": "ygzwms5656",
    "storePass": "ygzwms5656",
    "licensePath": "/ygz/wms/license/license.lic",
    "privateKeysStorePath": "/ygz/wms/license/privateKeys.keystore",
    "issuedTime": "2022-12-09 00:00:00",
    "expiryTime": "2099-12-09 00:00:00",
    "consumerType": "user",
    "consumerAmount": 1,
    "description": "这是证书描述信息",
    "licenseCheckModel": {
        "ipAddress": [
            "172.24.11.55"
        ],
        "macAddress": [
            "00-16-3E-03-6A-60"
        ],
        "cpuSerial": "54 06 05 00 FF FB 8B 0F",
        "mainBoardSerial": "95fdcef8-8805-4035-9a89-bf53fbe22759"
    }
}
```



POST /api/license/generateLicense

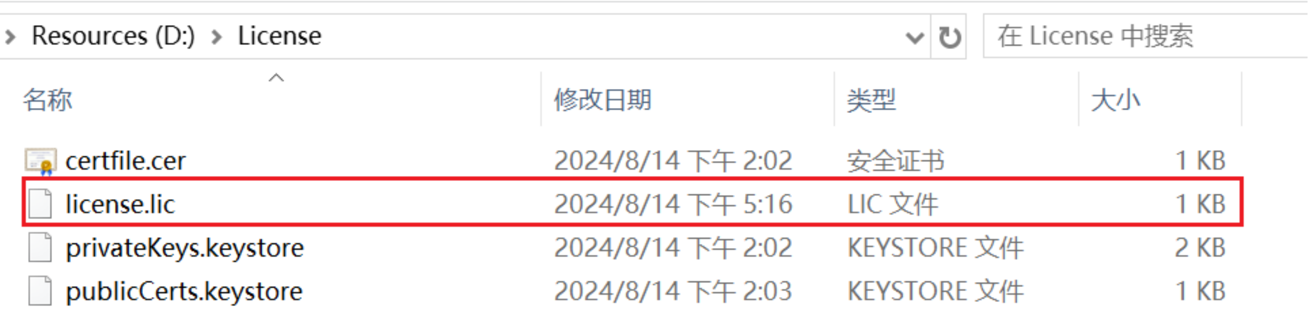
○ x-www-form-urlencoded ○ form-data ● raw JSON(application/json)

```
{
    "subject": "ygzwms",
    "privateAlias": "privateKey",
    "keyPass": "ygzwms5656",
    "storePass": "ygzwms5656",
    "licensePath": "D:/wms/license.lic",
    "privateKeysStorePath": "D:/wms/privateKeys.keystore",
    "issuedTime": "2022-12-09 00:00:00",
    "expiryTime": "2099-12-09 00:00:00",
    "consumerType": "user",
    "consumerAmount": 1,
    "description": "这是证书描述信息",
    "licenseCheckModel": {
        "ipAddress": [
            "192.168.27.1"
        ],
        "macAddress": [
            "EC-63-D7-3F-62-95"
        ],
        "cpuSerial": "BFEBFBFF000806D1",
        "mainBoardSerial": "PF2XE9FC"
    }
```

☑ 显示说明 响应码: 200 耗时: 23ms 大小

```
{
  "result": "ok",
  "msg": {
    "subject": "ygzwms",
    "privateAlias": "privateKey",
    "keyPass": "ygzwms5656",
    "storePass": "ygzwms5656",
    "licensePath": "D:/wms/license.lic",
    "privateKeysStorePath": "D:/wms/privateKeys.keystore",
    "issuedTime": "2022-12-09 00:00:00",
    "expiryTime": "2099-12-09 00:00:00",
    "consumerType": "user",
    "consumerAmount": 1,
    "description": "这是证书描述信息",
    "licenseCheckModel": {
      "ipAddress": [
```

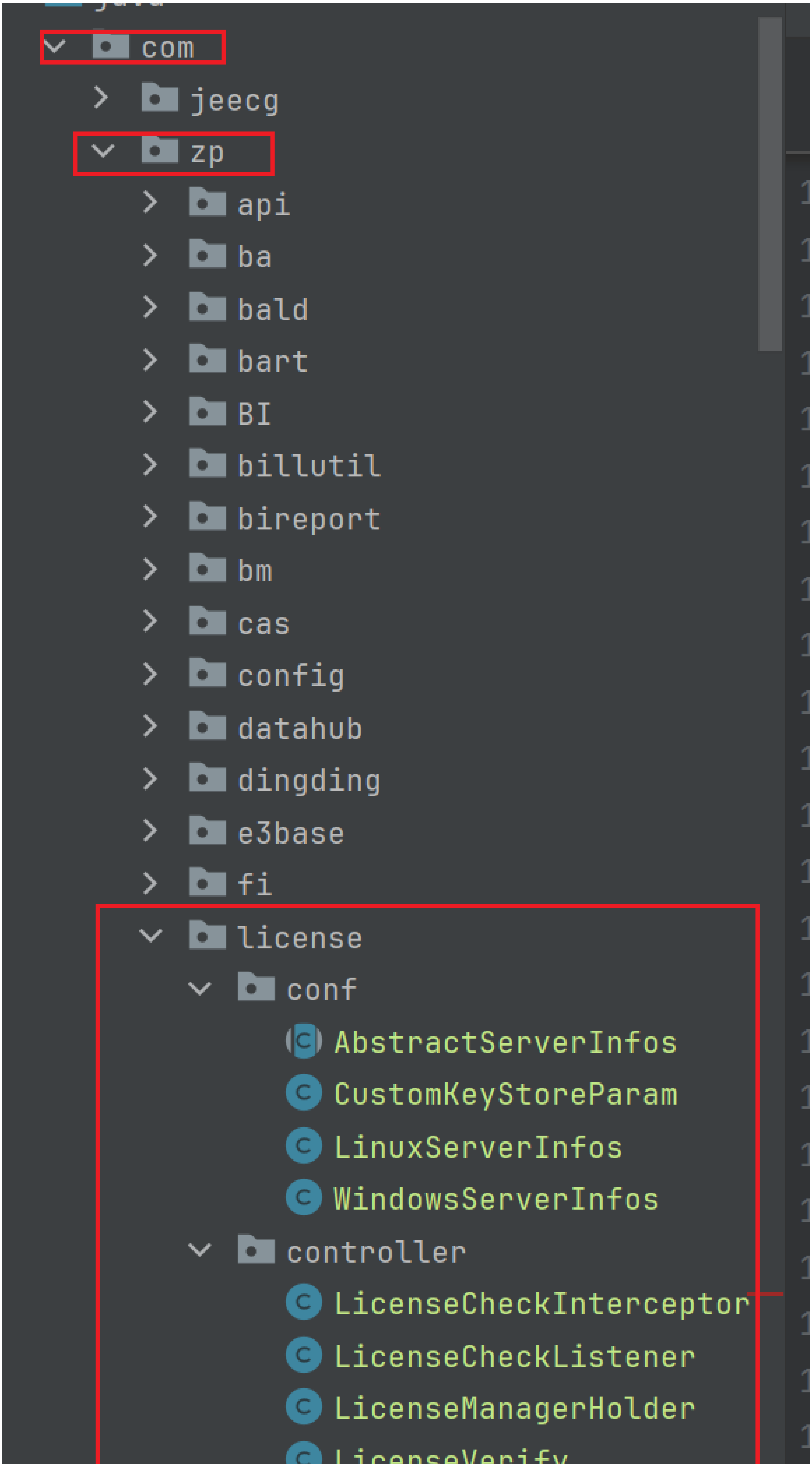
生成的文件和之前生成的保持在同一路径



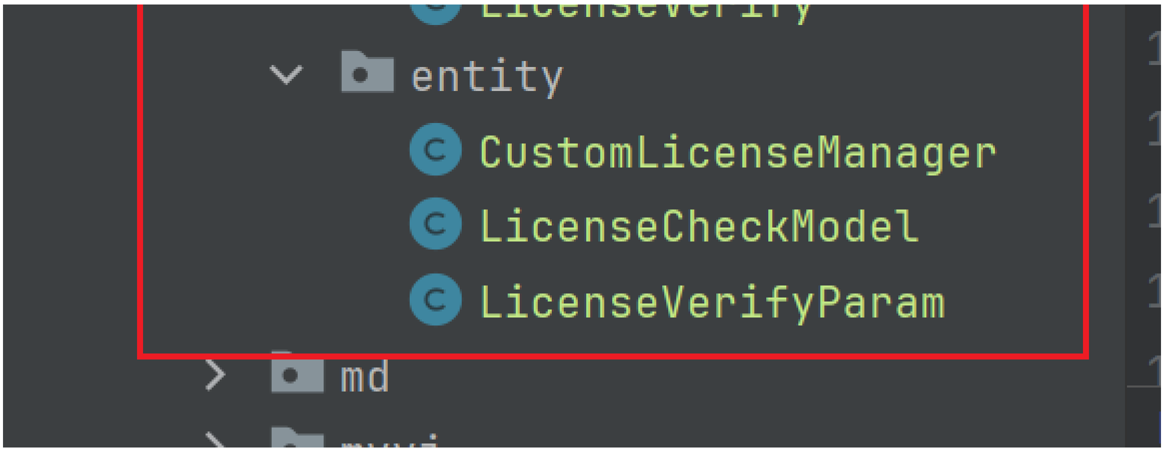
名称	修改日期	类型	大小
certfile.cer	2024/8/14 下午 2:02	安全证书	1 KB
license.lic	2024/8/14 下午 5:16	LIC 文件	1 KB
privateKeys.keystore	2024/8/14 下午 2:02	KEYSTORE 文件	2 KB
publicCerts.keystore	2024/8/14 下午 2:03	KEYSTORE 文件	1 KB

> 到这里在服务器生成授权文件已经完成，接下俩需要修改wms的sysConfig文件

集成到主项目



```
java
  com
    jeecg
    zp
      api
      ba
      bald
      bart
      BI
      billutil
      bireport
      bm
      cas
      config
      datahub
      dingding
      e3base
      fi
      license
        conf
          AbstractServerInfos
          CustomKeyStoreParam
          LinuxServerInfos
          WindowsServerInfos
        controller
          LicenseCheckInterceptor
          LicenseCheckListener
          LicenseManagerHolder
          LicenseVerify
```



entity

CustomLicenseManager

LicenseCheckModel

LicenseVerifyParam

md

修改 `sysConfig.properties` 中秘钥的路径

```
# windows ? linux
license.licensePath=D://License//license.lic
#license.licensePath=/wms/license/license.lic
license.publicKeysStorePath=D://License//publicCerts.keystore
#license.publicKeysStorePath=/wms/license/publicCerts.keystore


# 开启授权认证，关闭授权
# false
#app.token.flag.secretkey=PTNOkI+wLh0=
# true
app.token.flag.secretkey=aqvxTFL8q3o=
```

修改 `LicenseCheckListener` 个别关键字值

```
/**
     * 证书subject
     */
private String subject = "ygzwms";

/**
     * 公钥别称
     */
private String publicAlias = "publicCert";

/**
     * 访问公钥库的密码
     */
private String storePass = "ygzwms5656";

/**
     * 证书生成路径
     */
//    private String licensePath = "D://License//license.lic";
//    private String licensePath = "/wms/license/license.lic";
```

```
private String licensePath =
ResourceUtil.getConfigByName("license.licensePath");

/**
     * 密钥库存储路径
     */
//    private String publicKeysStorePath =
"D://License//publicCerts.keystore";
//    private String publicKeysStorePath =
"/wms/license/publicCerts.keystore";
private String publicKeysStorePath =
ResourceUtil.getConfigByName("license.publicKeysStorePath");
```

修改 `loginController.java` 的 `login` 接口

```
修改这个 public String login
    // 2022-07-29 吴超群 修改
    //          try {
    //              if
(StrUtil.isBlank(AlgorithmEncryptUtil.getCheckConfigFlag())) {
    //                  request.setAttribute("msg","授权码错误!");
    //                  return "login/login";
    //              }
    //
    //              if
("true".equalsIgnoreCase(AlgorithmEncryptUtil.getCheckConfigFlag())) {
    //                  AjaxJson j = new AjaxJson();
    //                  j = checkAccount();
    //                  if (!j.isSuccess()){
    //                      log.info("系统授权问题: " + j.getJsonStr());
    //                      if (j.getMsg()!=null){
    //                          request.setAttribute("msg",j.getMsg());
    //                          return "login/login";
    //                      }else {
    //                          return "login/loginError";
    //                      }
    //                  }else {
    //                      request.setAttribute("expireTime",j.getObj());
    //                  }
    //              }
    //          } catch (Exception e) {
    //              log.error("授权报错", e);
    //              request.setAttribute("msg","授权码错误!");
    //              return "login/login";
    //          }

    // 授权认证 2024-08-22
    if ("true".equalsIgnoreCase(AlgorithmEncryptUtil.getCheckConfigFlag())) {
        log.info("登录验证证书可使用性");
        LicenseVerify licenseVerify = new LicenseVerify();
        boolean verifyResult = licenseVerify.verify();

        if(verifyResult){
            log.info("验证成功，放行");
        }else{
            log.info("验证失败，拦截");
            request.setAttribute("msg","您的证书无效，请核查服务器是否取得授权或重新申请证书！");
            return "login/login";
        }
    }
```

`pom.xml` 文件增加授权依赖

```
<!--授权-->
<dependency>
    <groupId>de.schlichtherle.truelicense</groupId>
    <artifactId>truelicense-core</artifactId>
    <version>1.33</version>
</dependency>
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
    <version>3.7</version>
</dependency>
<dependency>
    <groupId>net.sourceforge.nekohtml</groupId>
    <artifactId>nekohtml</artifactId>
    <version>1.9.18</version>
</dependency>
```