**系统授权license授权认证**

**功能说明书**

|  |  |  |  |
| --- | --- | --- | --- |
| **文件版本：** |  | **编 制：** |  |
| **审 核：** |  | **批 准：** |  |

**XXXX有限责任公司 版权所有**

**内部资料 注意保密**

**修订记录**

| **修订版本号** | **修订人** | **修订日期** | **修订描述** |
| --- | --- | --- | --- |
| V0.1 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目 录**

1 简介 4

2 license.dat加密信息 4

3 运行步骤 4

4 主要功能代码说明 5

4.1 证书信息及生成 5

4.1.1 证书信息 5

4.1.2 证书信息添加 5

4.1.3 密钥生成 7

4.1.4 证书生成 7

4.2 监听器 7

4.2.1 实现逻辑 7

4.2.2 监听器注册 8

4.3 定时任务作用 8

4.4 访问控制实现 8

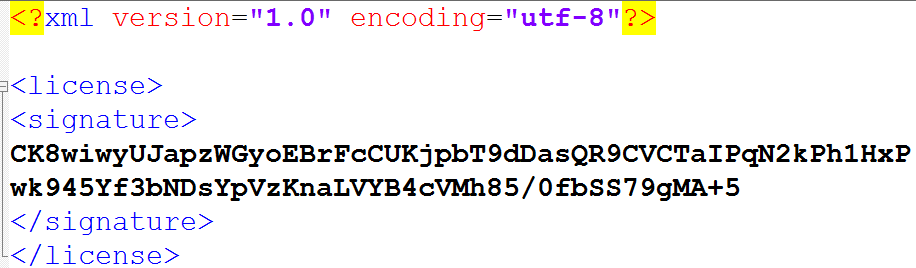
5 pom文件需要的包 9

# 简介

系统授权license认证，简单的实现了，在项目使用时，使用license.dat对项目进行认证，限制项目的访问。

# license.dat加密信息

license.dat中保存了签名信息，如图1所示。，签名信息采用DESede对“某段字符串”信息，用固定密钥进行的加密。“某段字符串”是由KEY.java中某几个固定写死的信息以逗号拼接构成。



**图1 license.dat内容**

# 运行步骤

1. 项目启动时，执行监听器SecurityContextListener
2. 监听器读取固定路径下“license.dat”文件，并对其中的签名信息进行解密。
3. 解密后并分解的基本信息和签名信息，保存在对象Lincense中
4. 监听器基本信息和签名信息的正确性，其证书是否过期，若验证不通过设置state=limit，验证通过设置state=run
5. 监听器获取ServletContext，将state信息保存在ServletContext
6. 实现定时任务，定时检查认证信息，若验证不通过设置state=limit，验证通过设置state=run
7. 定时任务将state信息保存在ServletContext
8. 在项目过滤器中添加对state判断，若未limit则限制访问

# 主要功能代码说明

## 证书信息及生成

### 证书信息

本代码基本信息包括：产品、公司名称、版本号，固定写死，保存的KEY.java中，对应License.java的属性。

基本信息可以根据自己的需要进行添加，添加方法及注意事项参考4.1.2

密钥和证书路径也保存在KEY.java中。如图2所示。

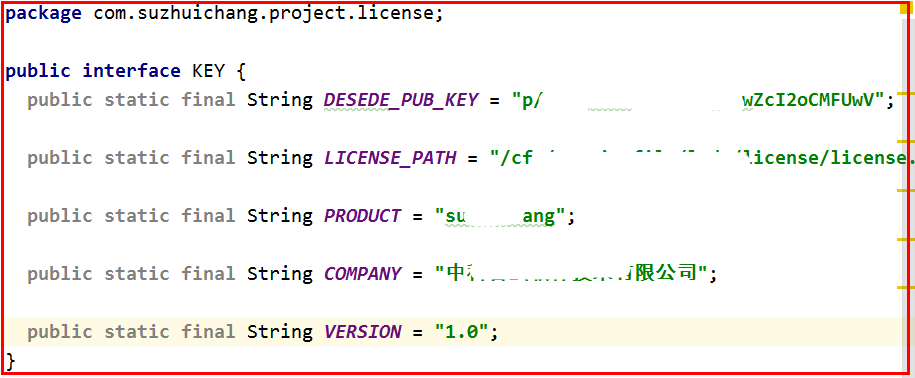
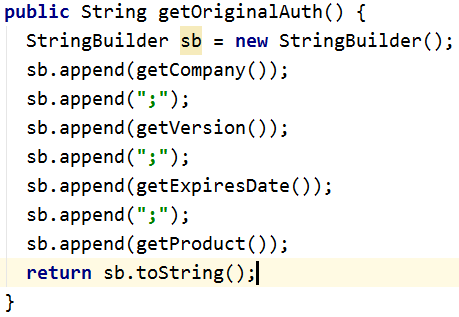


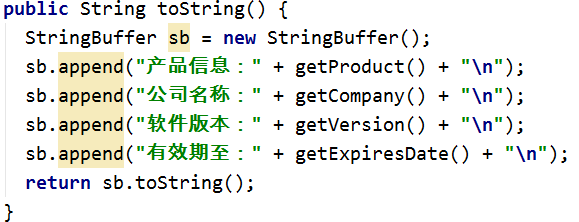
图2 KEY信息

### 证书信息添加

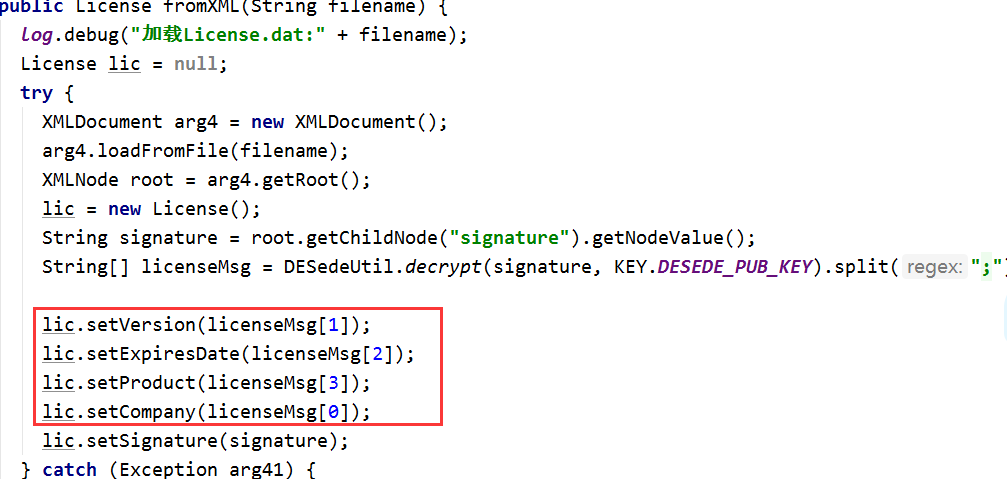
1. 在KEY.java中添加固定值的信息
2. 在License.java添加相应的属性
3. 在License.java的getOriginalAuth方法中，将新增的属性拼接在串中



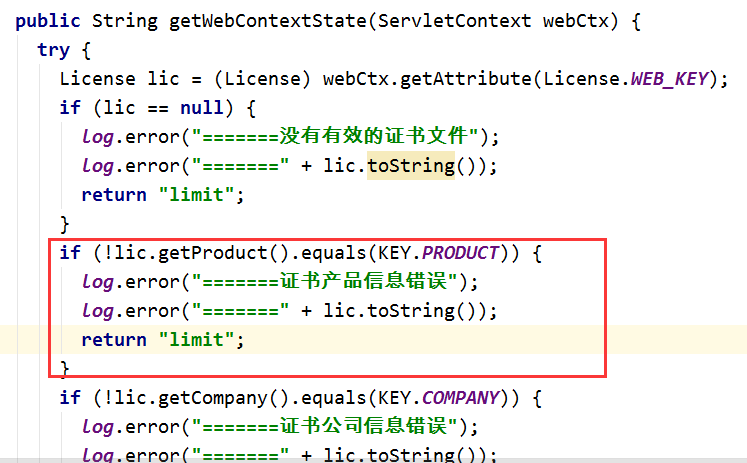
1. (可选)在License的toString方法中，添加新增的属性，此方法用于日志信息输出用



1. 在LicFactoryImpl.java的formXML方法中，将新加的属性按拼接在getOriginalAuth方法中的顺序取出，放置在Lincense中



1. 在LicFactoryImpl.java的getWebContextState方法中，判断新加属性的正确性

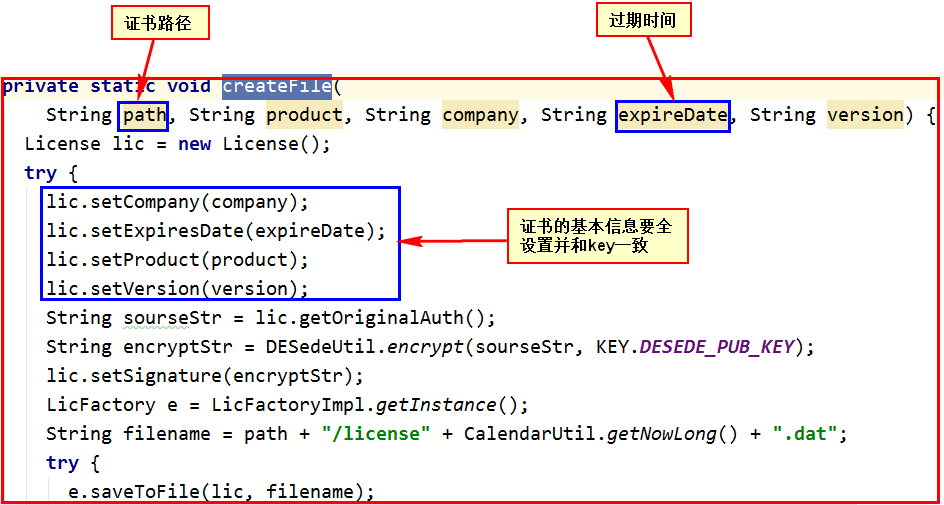


### 密钥生成

DESedeUtil.java中，generateKey()可以生成并返回密钥串，可以用写main方法调用，生成的密钥的替换KEY.java中的DESEDE\_PUB\_KEY

### 证书生成

CreateLicense.java的createFile方法生成证书.dat文件，生成的证书改名为license.dat,修改在KEY.java中指定的【LICENSE\_PATH】值，证书验证时会从此路径获取证书文件。



## 监听器

### 实现逻辑

实现监听器，在项目启动时，监听器执行contextInitialized方法。

初始化方法获取了KEY中【LICENSE\_PATH】指定路径中的证书文件，对签名signature进行解密，并对解密后的信息进行分解，值放置到License实体对象中，再将License实体对象保存到系统servletContext中。

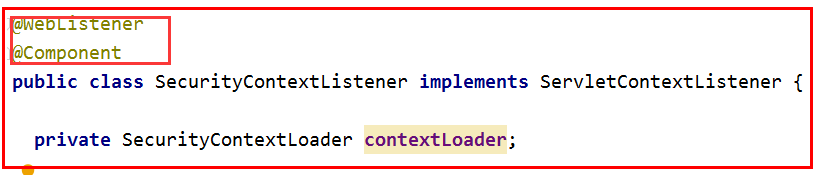
然后从servletContext获取License对象，进行验证，验证通过，返回信息“run”；验证未通过，返回信息“limit”，表示证书无效，限制访问。

验证返回的信息，将保存在servletContext中。

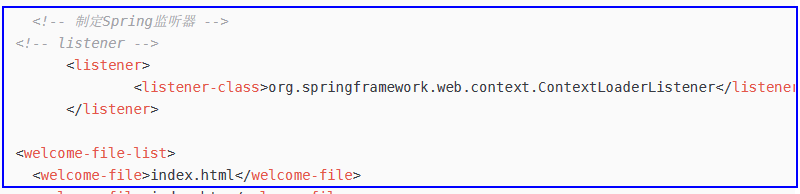
验证时，① 对License实体对象和KEY中保存的定值进行对比，验证信息是否一致；② 验证有效期是否过期；③ 验证解密后的拼接的信息是否一致。

### 监听器注册

maven项目监听器注册时，在类上添加注解@WebListener和@Component，如图3所示；如果是普通web项目，在web.xml中配置，如图4所示。



**图3 maven项目监听器注解**

****

**图4 普通web项目注册监听器**

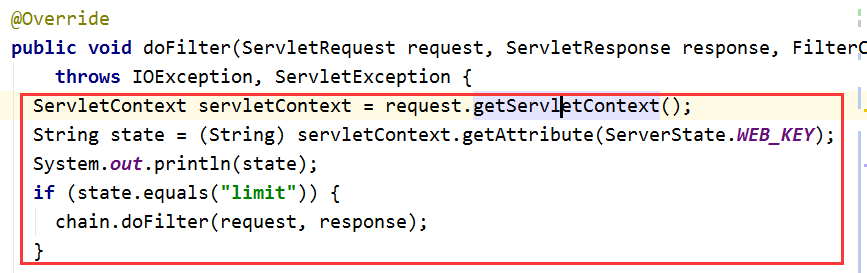
## 定时任务作用

定时任务的作用，是定时去检查项目的证书，验证信息和过期时间，判断证书是否支持有效，并重新更新验证信息limit或run至servletContext中。

定时任务类：LicenseTask.java

## 访问控制实现

在项目过滤器中，获取servletContext的验证信息(值为limit或run)，如果是run，继续访问，如果是limit，限制访问，代码如下。



# pom文件需要的包

<dependency>  
 <groupId>org.dom4j</groupId>  
 <artifactId>dom4j</artifactId>  
 <version>2.1.3</version>  
 </dependency>