**使用xjar对Spring-Boot JAR包加密**

**使用说明书**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 Xjar介绍 4

2 如何使用xjar v2.06 4

2.1 配置pom文件 4

2.2 手动下载maven包 5

2.3 运行加密 6

2.4 项目启动 7

2.4.1 windows启动 7

2.4.2 linux启动 7

2.5 错误处理 8

2.5.1 Exception in thread "main" java.lang.ClassNotFoundException: 8

3 参考网址 8

# Xjar介绍

Spring Boot JAR 安全加密运行工具，同时支持的原生JAR。 基于对JAR包内资源的加密以及拓展ClassLoader来构建的一套程序加密启动，动态解密运行的方案，避免源码泄露或反编译。 功能特性无需侵入代码，只需要把编译好的JAR包通过工具加密即可。 完全内存解密，杜绝源码以及字节码泄露或反编译。 支持所有JDK内置加解密算法。 可选择需要加解密的字节码或其他资源文件，避免计算资源浪费

# 如何使用xjar v2.06

## 配置pom文件

* **添加依赖**

<dependency>

<groupId>com.github.core-lib</groupId>

<artifactId>xjar</artifactId>

<version>v2.0.6</version>

</dependency>

<dependency>

<groupId>com.github.core-lib</groupId>

<artifactId>loadkit</artifactId>

<version>v1.0.0</version>

</dependency>

* **设置 jitpack.io 仓库**

<repositories>

<repository>

<id>jitpack.io</id>

<url>https://jitpack.io</url>

</repository>

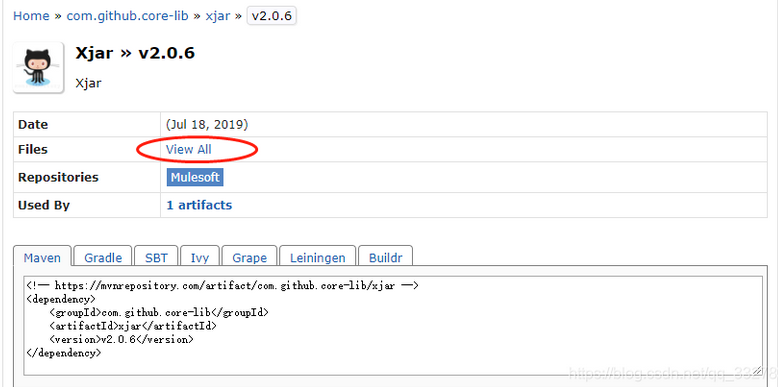
</repositories>

注：如果下载不了依赖包，可以进行手动下载

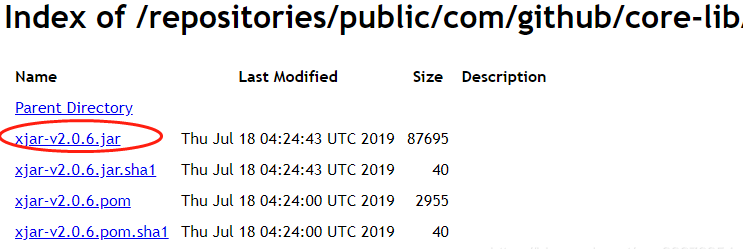
## 手动下载maven包

1. 打开maven的远程仓库下载：

<https://mvnrepository.com/artifact/com.github.core-lib/xjar/v2.0.6>



1. 选择xjar包

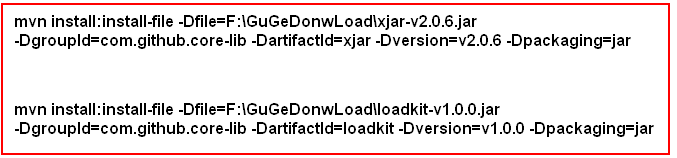


1. 下载loadkit，地址

https://mvnrepository.com/artifact/com.github.core-lib/loadkit/v1.0.0

1. 安装执行jar包

上面两个包下载到本地之后，启动cmd，执行下面两段命令（默认我们都有配置maven的环境变量），注意下面命令中的文件路劲改成自己的下载的对应路径



## 运行加密

在工程下随便创建一个类，加入下面的main方法，运行结束后会生成对应的加密jar包，然而你会发现加密后的jar包会比没有加密的大一倍。

import io.xjar.XConstants;

import io.xjar.XKit;

import io.xjar.boot.XBoot;

import io.xjar.key.XKey;

/\*\*

\* @Author zzl

\* @create 2020/5/22 0022 10:23

\*/

public class Test {

public static void main(String[] args) {

try {

// Spring-Boot Jar包加密

String password = "8888";

XKey xKey = XKit.key(password);

//参数1是我们打包的文件目录和名称，参数2是加密后的文件路径和名称，参数3是密码

XBoot.encrypt("E:\\GIT\\recess-manage-server\\target\\manage-server-1.0.0.jar",

"E:\\GIT\\recess-manage-server\\target\\manage-server-2-1.0.0-.jar", xKey);

System.out.println("Successfully generated encrypted jar");

}catch (Exception e){

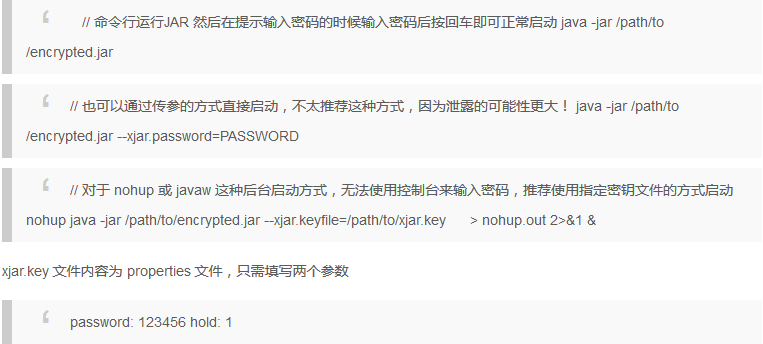
e.printStackTrace();

}

}

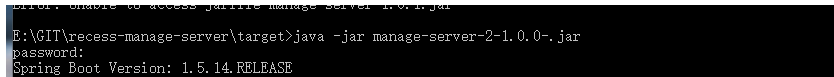
}

## 项目启动



### windows启动

windows启动或者当前窗口启动方式：cmd 到加密的jar下，启动jia包（java -jar xxxxxxx.jar），会提示输入密码，成功则启动。



### linux启动

linux后台启动，首先利用ieda创建一个properties文件，写入密码。然后放置到linux的某个目录下，运行即可。



可能在启动加密后的jar会提示：Exception in thread "main"

java.lang.ClassNotFoundException:，在pom加入下面代码，对应mainClass写自己的。

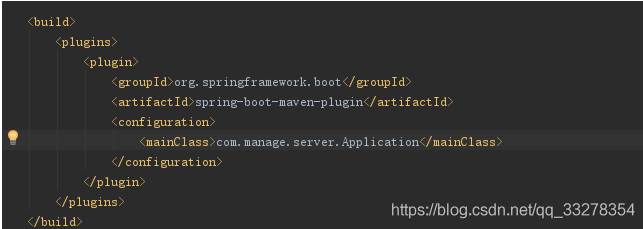


## 错误处理

### Exception in thread "main" java.lang.ClassNotFoundException:

可能在启动加密后的jar会提示：Exception in thread "main"

java.lang.ClassNotFoundException:，在pom加入下面代码，对应mainClass写自己的。



# 参考网址

* https://blog.csdn.net/qq\_33278354/article/details/106278539?utm\_medium=distribute.pc\_aggpage\_search\_result.none-task-blog-2~all~first\_rank\_v2~rank\_v25-8-106278539.nonecase&utm\_term=%E5%AF%B9jar%E8%BF%9B%E8%A1%8C%E5%8A%A0%E5%AF%86
* https://msd.misuland.com/pd/4146263639743012462