

手把手教你将 Java 项目推送到 Maven 中央仓库实践

超级小豆丁 SpringForAll社区 2月23日

点击上方👉[SpringForAll社区](#) 轻松关注!

及时获取有趣有料的技术文章

本文来源: <http://www.mydlq.club/article/15/>

- . 一、将项目推送到 Github
- . 二、注册 Sonatype 账户
- . 三、登录 Sonatype 创建工单
- . 四、确认域名
- . 五、配置 GPG
 - . 1、下载 GPG
 - . 2、配置密钥对
 - . 3、上传公钥到 GPS key-servers
- . 六、配置项目
 - . 1、配置 Strings.xml
 - . 2、配置 Pom.xml
 - . 3、设置编码
- . 七、发送项目到 Maven 中央仓库
 - . 1、登录 Nexus Repositories Manager
 - . 2、发布项目
 - . 3、查看发布过程
 - . 4、在 Issues 中回复已经 Release 完成
 - . 5、查看自己发布的项目

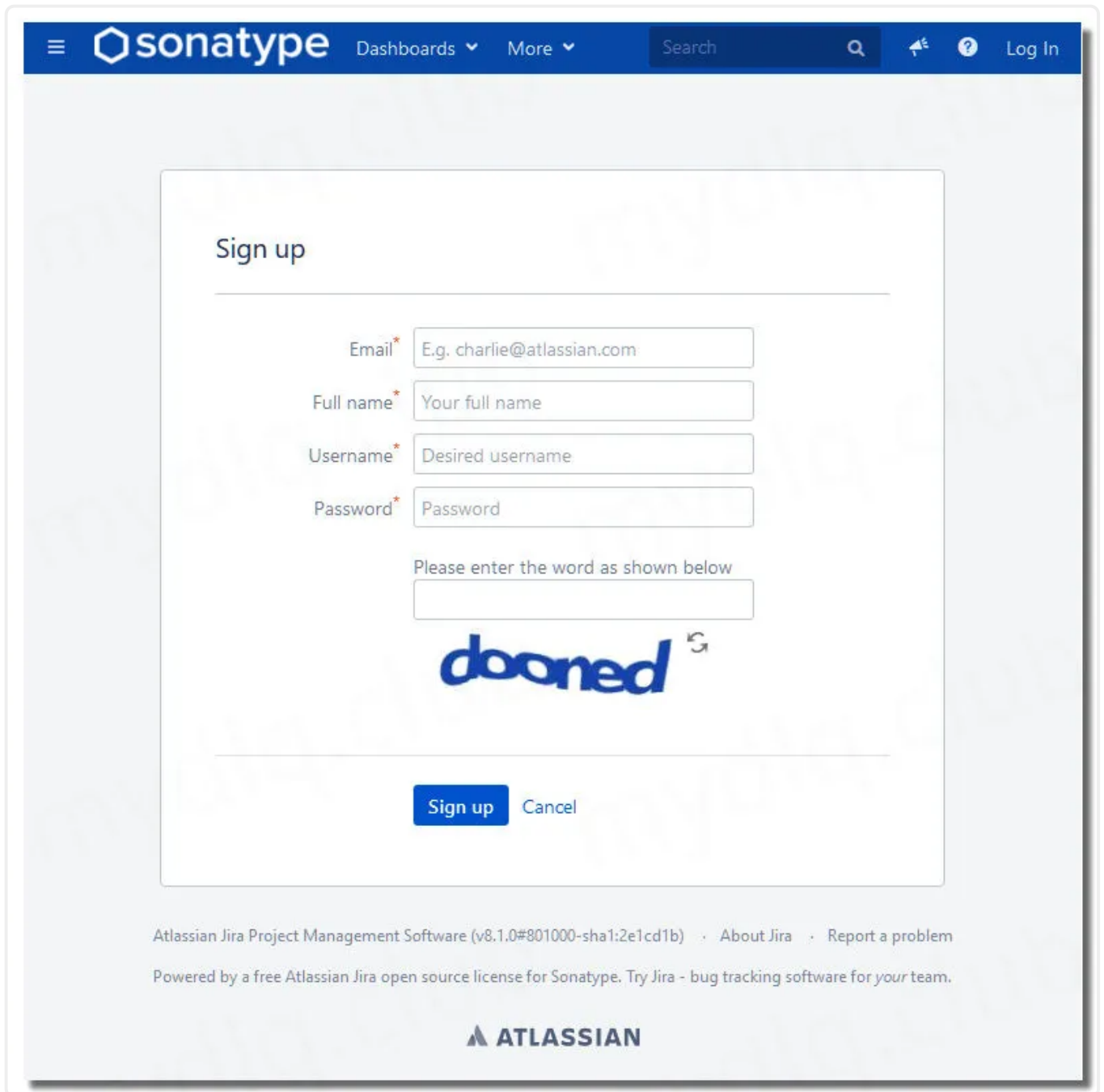
最新写了个开源项目 **Swagger Kubernetes**，考虑到发布到 **Maven** 中央仓库，发布成功，全球通用，在 **Maven** 拉取的过程中不存在 Jar 包不在无法下载的问题。由于第一次将项目提交到 **Maven** 中央仓库，网上知识参差不齐到最后遇见了很多坑，这里记录一下这个实践过程。

一、将项目推送到 **Github**

在 **GitHub** 上新建一个项目仓库，然后将要发送到 **Maven** 中央仓库的代码推送到 **Github**。这方面资料较多，这里不过多叙述。

二、注册 **Sonatype** 账户

进入 <https://issues.sonatype.org/secure/Dashboard.jspa> 注册JIRA账号，按提示完善个人信息。



The image shows a web browser window displaying the Sonatype sign-up page. The header is blue with the Sonatype logo, navigation links for 'Dashboards' and 'More', a search bar, and a 'Log In' link. The main content area is a white box titled 'Sign up'. It contains four input fields: 'Email' (placeholder: E.g. charlie@atlassian.com), 'Full name' (placeholder: Your full name), 'Username' (placeholder: Desired username), and 'Password' (placeholder: Password). Below these is a CAPTCHA section with the text 'Please enter the word as shown below' and a box containing the word 'dooned' in a stylized blue font. At the bottom of the form are two buttons: 'Sign up' (blue) and 'Cancel' (grey). Below the form, there is a footer section with text about Atlassian Jira Project Management Software (v8.1.0#801000-sha1:2e1cd1b), links for 'About Jira' and 'Report a problem', and a statement about being powered by a free Atlassian Jira open source license for Sonatype. The Atlassian logo is at the very bottom.

Sign up

Email*

Full name*

Username*

Password*

Please enter the word as shown below

Atlassian Jira Project Management Software (v8.1.0#801000-sha1:2e1cd1b) · [About Jira](#) · [Report a problem](#)

Powered by a free Atlassian Jira open source license for Sonatype. Try Jira - bug tracking software for your team.

ATLASSIAN

三、登录 Sonatype 创建工单

登录 <https://issues.sonatype.org/secure/Dashboard.jspa>，点击创建按钮来创建一个新的 issues 工单



内容如下，填写必填部分即可，然后等待审核。

5/18

```
* Add a TXT record to your DNS referencing this GitHub ticket: OSSRH-48958 (https://github.com/sonatype/docker-ossrh/issues/48958)
* Setup a redirect to your Github page (if it does not already exist)

If you do not own this domain, please read:
http://central.sonatype.org/pages/choosing-your-coordinates.html
You may also choose a groupId that reflects your project hosting, in this case, so

Swagger Kubernetes is a program for document aggregation of Swagger standardized s
-----

关键字：OSSRH-48958
URL：https://issues.sonatype.org/browse/OSSRH-48958
项目：Community Support – Open Source Project Repository Hosting
问题类型：New Project

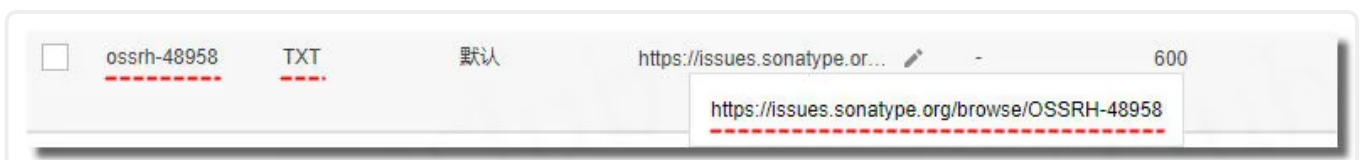
报告人：mydlq
经办人：Joel Orlina
优先级：重要

Swagger Kubernetes is a program for document aggregation of Swagger standardized s
```

意思就是询问你该 **GroupId** 设置的域名是否为你自己所拥有，如果是就需要证明一下，两种方法

- 1、设置github page当做域名。
- 2、在域名 DNS 解析中添加一条类型为 Txt 的记录。

由于本人有自己的域名，所以设置使用了第二种方法添加 NDS 解析记录，这里添加了记录为 OSSRH-48958，类型为 TxT，值为 <https://issues.sonatype.org/browse/OSSRH-48958> 的一条解析记录信息。



然后打开邮件中提到的地址 <https://issues.sonatype.org/browse/OSSRH-48958> 进入该 issues，在里面回复自己拥有该域名，且设置了 DNS 解析。

活动日志

全部

注释

工作日志

改动记录

活动日志

↑

mydlq 创建问题 - 3小时以前

Central OSSRH 添加了评论 - 3小时以前

Do you own the domain mydlq.club? If so, please verify ownership via one of the following methods:

Add a TXT record to your DNS referencing this JIRA ticket: [OSSRH-48958](#) (Fastest)

Setup a redirect to your Github page (if it does not already exist)

If you do not own this domain, please read:
<http://central.sonatype.org/pages/choosing-your-coordinates.html>
You may also choose a groupId that reflects your project hosting, in this case, something like io.github.my-dlq or com.github.my-dlq

Central OSSRH 进行了改变 - 3小时以前

域	原值	新值
状态	Open [1]	Waiting for Response [1]

备注

样式

B

I

U

A

°

🔗

🔒

☺

+

⤴

I own the domain mydlq.club and I have added a dns record TXT T record.

说明域名情况信息

可视化

文本

🔒

所有用户可见

添加

取消

五、配置 GPG

1、下载 GPG

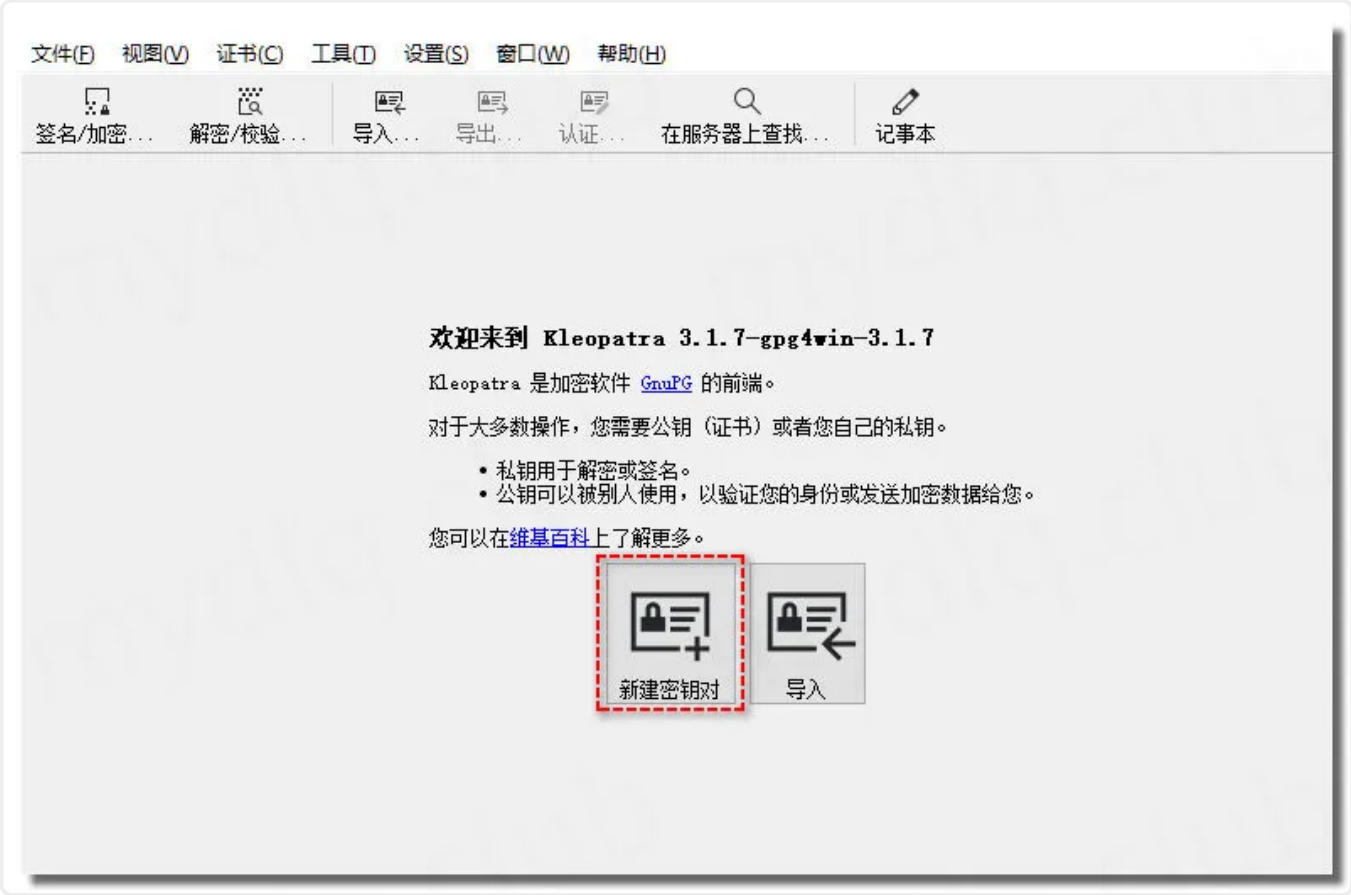
打开 <https://www.gpg4win.org/> 下载 GPG，然后一步步安装。

安装完成后会看到桌面上多了个 Kleopatra 图标，这个程序是 GnuPG 的前端程序，这里我们将它打开。

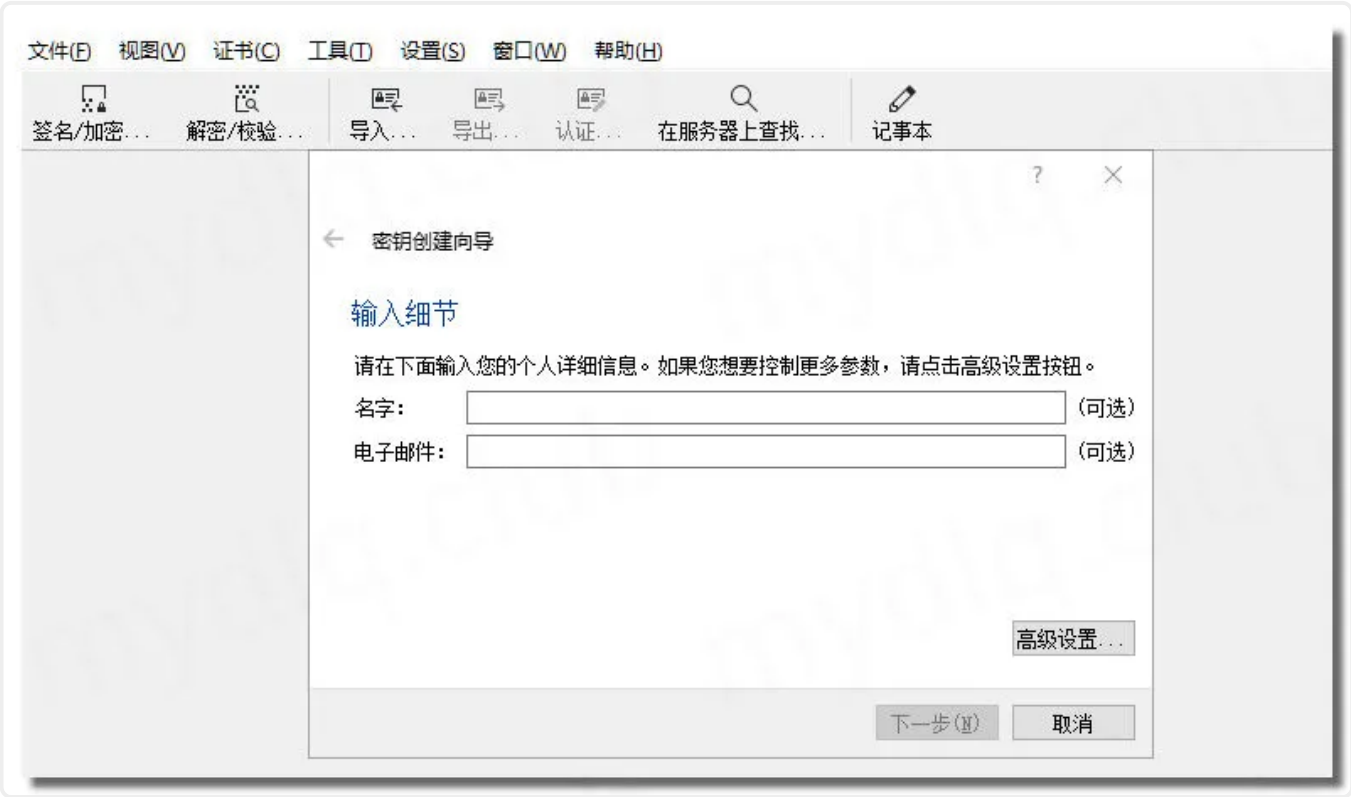
2、配置密钥对

打开 Kleopatra 后，点击新建密钥对

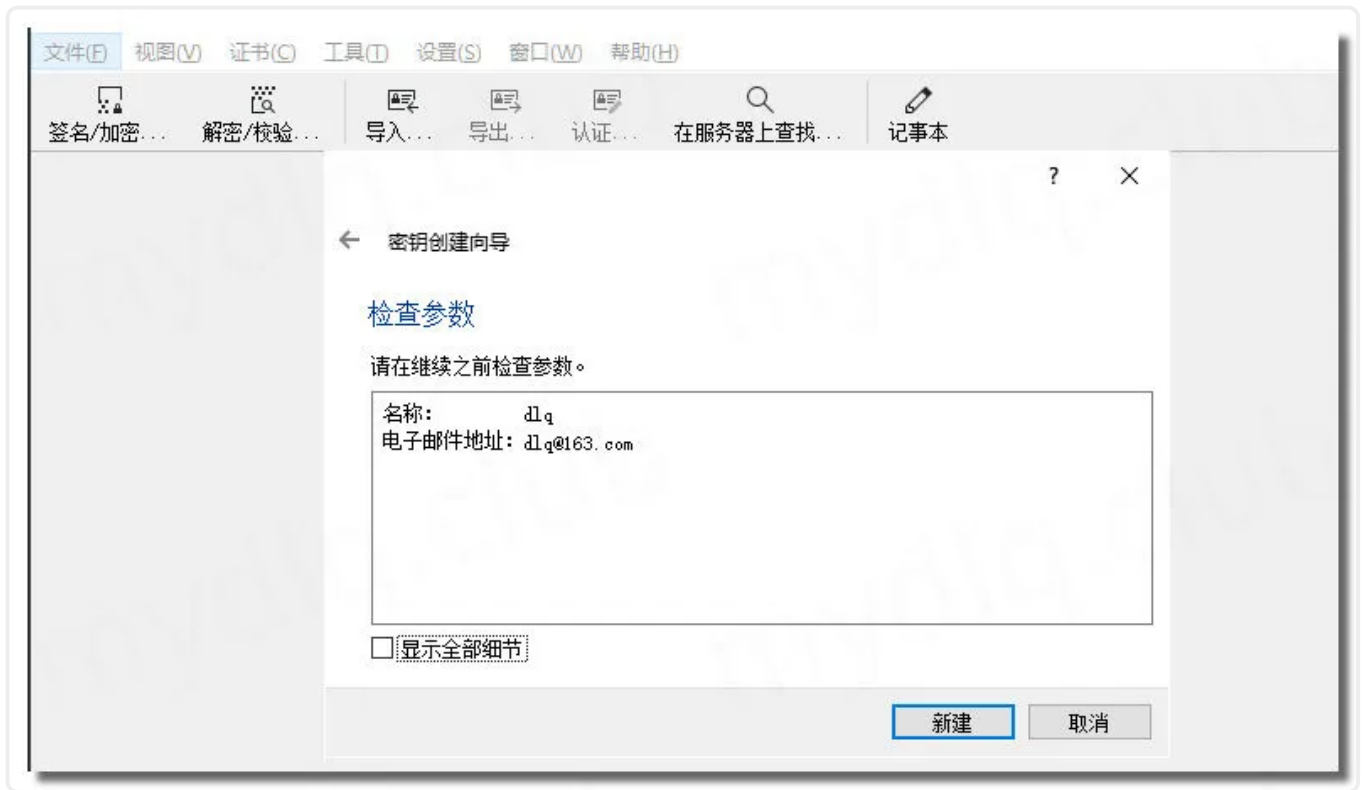
https://mp.weixin.qq.com/s?__biz=MzU0MDEwMjgwNA==&mid=2247486964&idx=1&sn=2b477595f44054dab9bdc16ae2925631&chksm=fb3f140fcc489d... 7/18



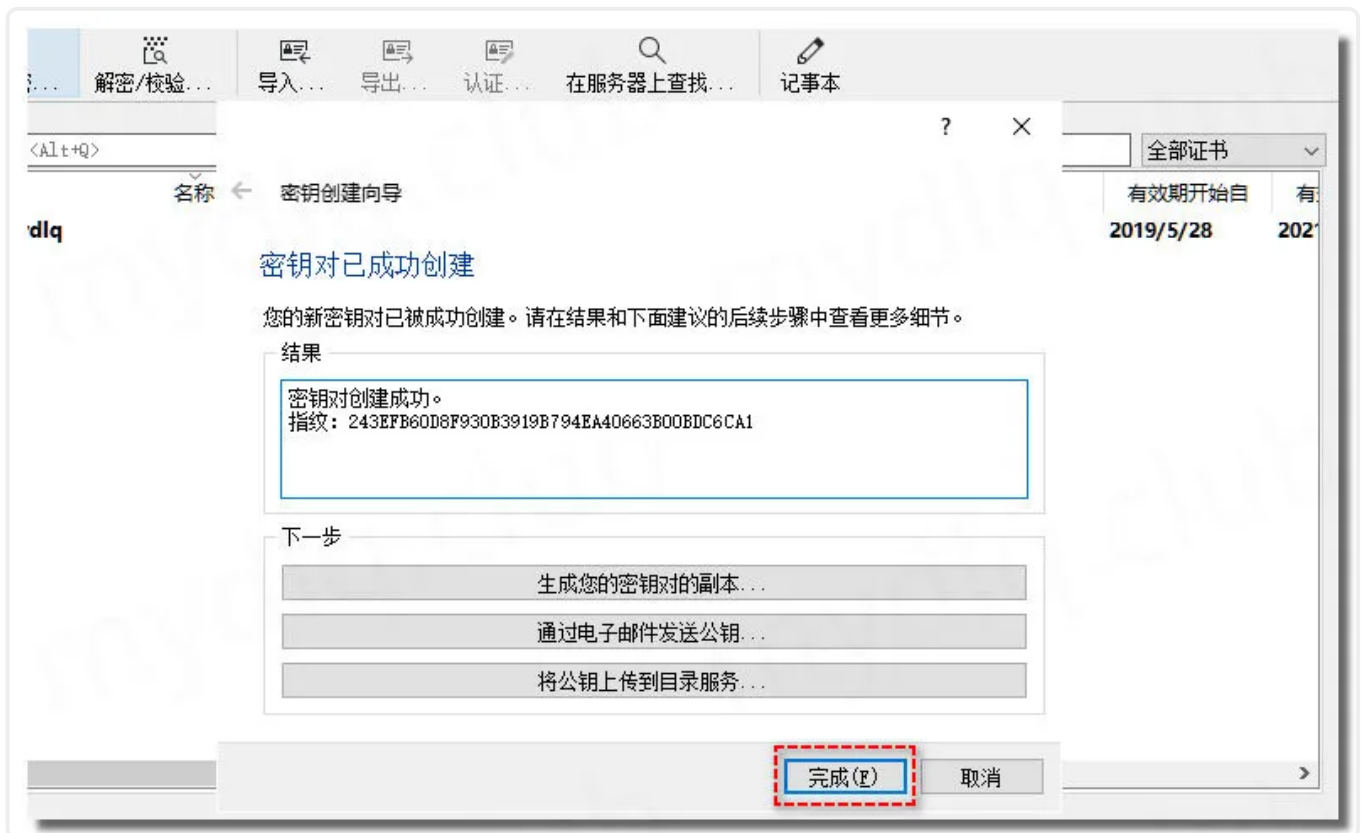
输入姓名和邮箱



最后确认新建



过程中需要自己输入一个**8**位以上的密码，请记住该密码，后面还需要用到。输入完成以后就会提示下面信息。



3、上传公钥到 GPS key-servers

打开命令提示符窗口，输入下面命令将证书发送到远程服务器

```
gpg --keyserver hkp://keyserver.ubuntu.com:11371 --send-keys 243EFB60D8F930B391CA194EA40663B00BDC6CA1
```



六、配置项目

要想将项目发送到 Maven 中央仓库，需要对项目进行一些设置来符合发布条件。

1、配置 Strings.xml

配置 maven 目录 conf 文件夹中的 settings.xml 文件，设置一个 server, 里面添加 JIRA 的账号和密码。

```
<servers>
  <server>
    <id>ossrh</id>
    <username>JIRA账号</username>
    <password>JIRA密码</password>
  </server>
</servers>
```

2、配置 Pom.xml

Pom.xml 配置一些必要的信息，例如 Maven 文档插件、打包插件、GPG验证插件等，如下：

可以查看我的项目的 GitHub 地址：Maven 示例项目，pom 参数照着修改即可，顺便点下 Start 星，Thanks。

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>club.mydlq</groupId>
  <artifactId>swagger-kubernetes</artifactId>
  <version>2.1.0</version>
  <packaging>jar</packaging>

  <name>swagger-kubernetes</name>
```

```

<name>swagger-kubernetes</name>
<url>https://github.com/my-dlq/swagger-kubernetes</url>
<description>swagger api doc for kubernetes discovery.</description>

<!--licenses信息-->
<licenses>
  <license>
    <name>The Apache Software License, Version 2.0</name>
    <url>http://www.apache.org/licenses/LICENSE-2.0.txt</url>
    <distribution>repo</distribution>
  </license>
</licenses>

<!--scm信息-->
<scm>
  <url>http://mydlq.club</url>
  <connection>scm:git:https://github.com/my-dlq/swagger-kubernetes.git</connection>
  <developerConnection>scm:git:https://github.com/my-dlq/swagger-kubernetes.git</developerConnection>
</scm>

<!--发布者信息-->
<developers>
  <developer>
    <name>mydlq</name>
    <email>mynamedlq@163.com</email>
    <organization>http://mydlq.club</organization>
    <organizationUrl>http://mydlq.club</organizationUrl>
  </developer>
</developers>

<properties>
  <java.version>1.8</java.version>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
    <version>2.1.0.RELEASE</version>
  </dependency>
</dependencies>

<build>
  <plugins>
    <!-- doc plugin,Maven API文档生成插件 -->
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-javadoc-plugin</artifactId>
      <version>3.1.0</version>
      <executions>
        <execution>
          <id>attach-javadocs</id>
          <goals>
            <goal>jar</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>

```

```

        </goals>
    </execution>
</executions>
</plugin>
<!-- resources plugin,Maven 资源插件 -->
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-source-plugin</artifactId>
    <version>3.1.0</version>
    <executions>
        <execution>
            <id>attach-sources</id>
            <goals>
                <goal>jar-no-fork</goal>
            </goals>
        </execution>
    </executions>
</plugin>
<!-- compiler plugin,Maven 编译插件 -->
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>3.3</version>
    <configuration>
        <source>${java.version}</source>
        <target>${java.version}</target>
        <showWarnings>true</showWarnings>
    </configuration>
</plugin>
<!-- gpg plugin,用于签名认证 -->
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-gpg-plugin</artifactId>
    <version>1.6</version>
    <executions>
        <execution>
            <id>sign-artifacts</id>
            <phase>verify</phase>
            <goals>
                <goal>sign</goal>
            </goals>
        </execution>
    </executions>
</plugin>
<!-- staging puglin,用于自动执行发布阶段(免手动)-->
<plugin>
    <groupId>org.sonatype.plugins</groupId>
    <artifactId>nexus-staging-maven-plugin</artifactId>
    <version>1.6.7</version>
    <extensions>true</extensions>
    <configuration>
        <serverId>ossrh</serverId>

```

```

        <nexusUrl>https://oss.sonatype.org/</nexusUrl>
        <autoReleaseAfterClose>true</autoReleaseAfterClose>
    </configuration>
</plugin>
<!-- release plugin,用于发布到release仓库部署插件 -->
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-release-plugin</artifactId>
    <version>2.4.2</version>
</plugin>
</plugins>
</build>

<!-- 这里引入 Settings.xml 中设置的用户名、密码 -->
<distributionManagement>
    <snapshotRepository>

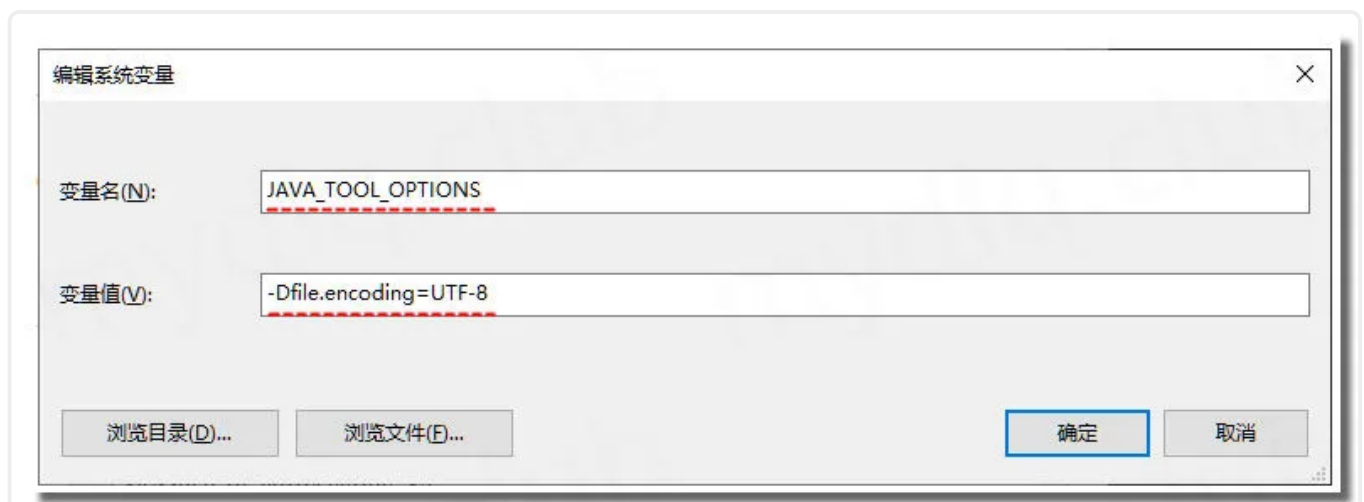
        <id>ossrh</id>
        <url>https://oss.sonatype.org/content/repositories/snapshots</url>
    </snapshotRepository>
    <repository>
        <id>ossrh</id>
        <url>https://oss.sonatype.org/service/local/staging/deploy/maven2/</url>
    </repository>
</distributionManagement>
</project>

```

3、设置编码

如果Maven编译doc因编码原因出错，可以在环境变量设置变量 JAVATOOL_OPTIONS，值为 UTF-8 来改变编码方式。

- JAVATOOL_OPTIONS
- -Dfile.encoding=UTF-8

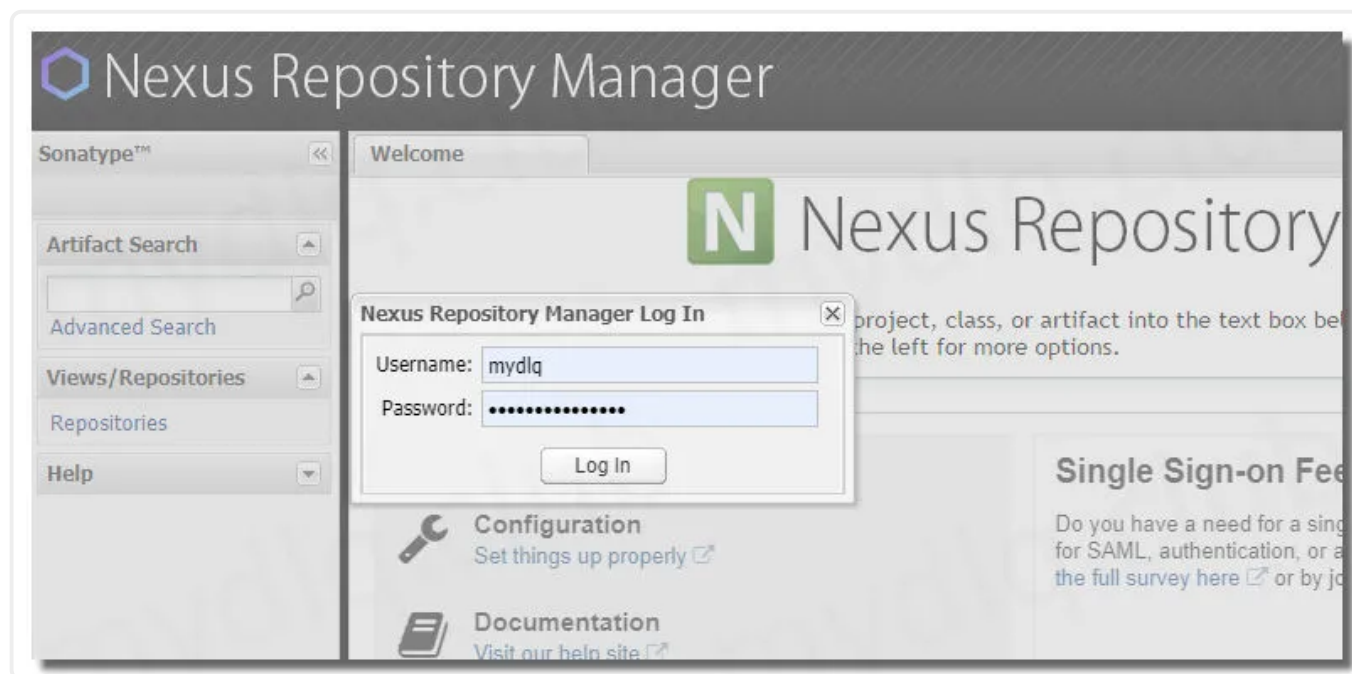


七、发送项目到 Maven 中央仓库

1、登录 Nexus Repositories Manager

在发布之前要求必须有登录 Nexus Repositories 的信息，所以发布前提前登录 Nexus Repositories

登录 <https://oss.sonatype.org> 其中用户名、密码就是 JIRA 的用户名、密码。



2、发布项目

执行 `mvn` 命令，部署项目到 Maven 仓库，过程中需要输入之前设置的 GPG 密码，输入一下即可。

```
$ mvn clean deploy
```

3、查看发布过程

回到 Nexus Repositories Manager，选择 **Staging Repositories** 暂存库，根据发布时间查找到你提交的 Jar 信息。然后查看 **Actives** 里面的执行信息，注意执行部署过程现在流行都交给 Maven 自动部署插件执行，网上一些老办法是进入这里面手动执行，非常不推荐这样做，因为这样非常容易部署发生问题。

Nexus Repository Manager

Sonatype™

Welcome | **Staging Repositories**

Refresh Close Promote Release Drop

Repository	Profile	Status	Updated
central_bundles-17615	Central Bundles	closed	2019-May-24 15:36:10
central_bundles-17616	Central Bundles	closed	2019-May-24 15:39:17
central_bundles-17618	Central Bundles	closed	2019-May-24 15:45:58
central_bundles-17619	Central Bundles	closed	2019-May-24 15:49:19
central_bundles-17620	Central Bundles	closed	2019-May-24 15:56:36
central_bundles-17621	Central Bundles	closed	2019-May-24 16:10:29
central_bundles-17623	Central Bundles	open	2019-May-26 00:34:43
central_bundles-17632	Central Bundles	open	2019-May-26 20:46:20
central_bundles-17634	Central Bundles	closed	2019-May-27 02:07:54
central_bundles-17636	Central Bundles	closed	2019-May-27 04:18:19

central_bundles-17636

Summary Activity Content

Activities

- open
- close
- Evaluating rules: Central Sync Requirement
- Evaluating rule: POM Validation
- Passed: POM Validation
- Evaluating rule: Profile target matcher
- Passed: Profile target matcher
- Evaluating rule: Archives must not contain insecure paths
- Passed: Archives must not contain insecure paths
- Evaluating rule: Checksum Validation
- Passed: Checksum Validation
- Evaluating rule: Sources Validation
- Passed: Sources Validation
- Evaluating rule: Javadoc Validation
- Passed: Javadoc Validation
- Evaluating rule: Signature Validation
- Passed: Signature Validation
- All rules passed: Central Sync Requirement
- Sending e-mail notifications
- Repository closed

可以看到执行过程中的信息

不要手动操作 这一切都应交给 Maven Release 插件来执行

注意：在 Maven 执行发布过程中才能看到，如果已经 release 完成，暂存库中可能找不到。

正常的部署流程应该为：**open**阶段—>**close**阶段—>**release**阶段 执行完成这三个阶段就会发布到中央仓库，等会会收到邮件通知

[<https://issues.sonatype.org/browse/OSSRH-48958?page=com.atlassian.jira.plugin.system:issuetype-description>]

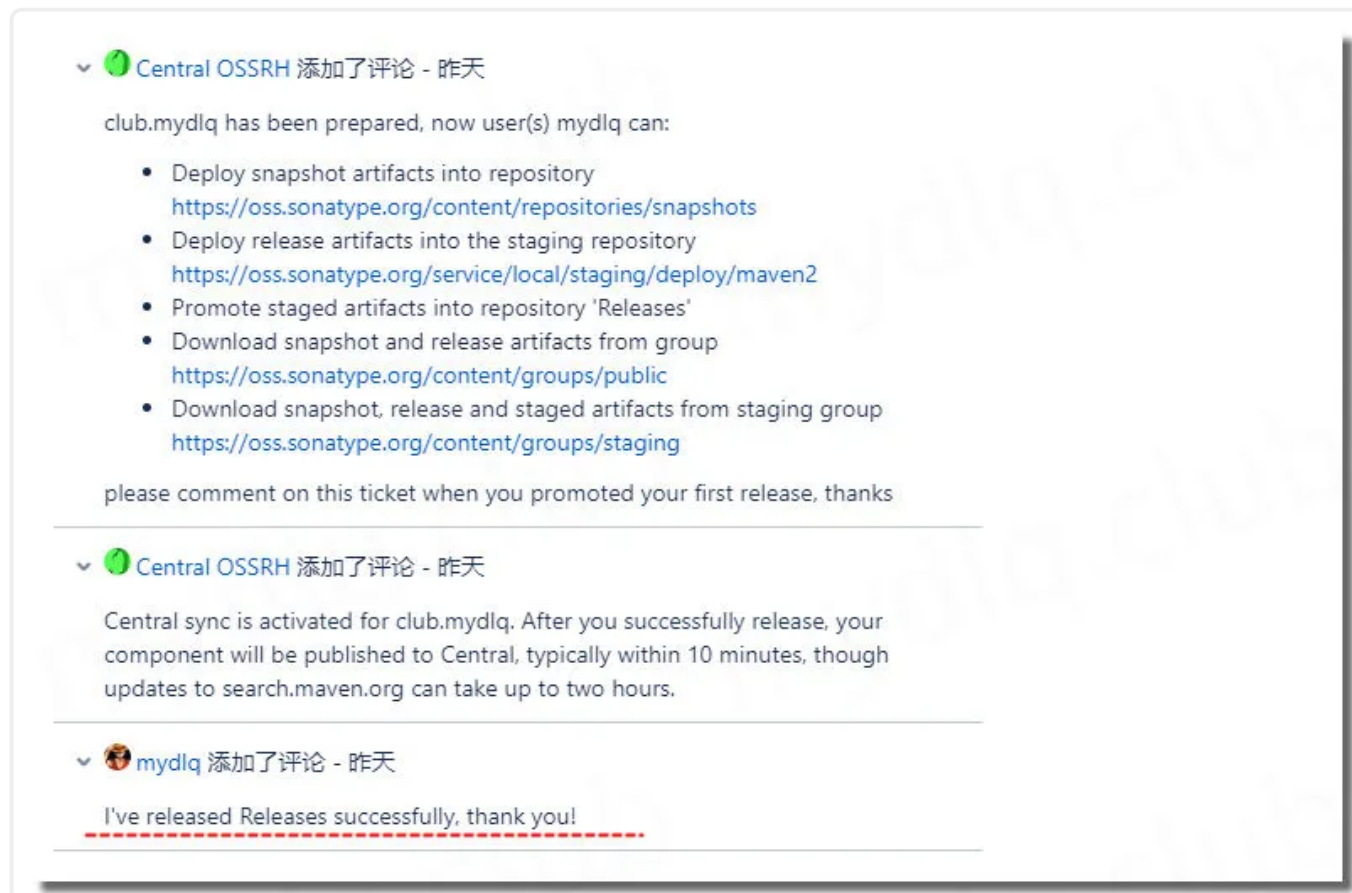
Central OSSRH 更新了 OSSRH-48958:

Central sync is activated for club.mydlq. After you successfully release, your com

意思为 Release 成功，大概等十分钟或者几个就能在中央仓库查询到该 Jar 信息，不过对 search.maven.org 的更新可能需要两个小时。

4、在 Issues 中回复已经 Release 完成

打开之前提交 Issues 地址，后面回复已经 Release 完成



5、查看自己发布的项目

发布成功后可以进入 <https://search.maven.org> 网址查询自己发布的 Jar，如果未查到等几个小时再此查询。

