

发布jar包到Maven中央仓库

当我们开发完一些java的jar的时候，可能想将它发布到远端的代码仓库，这样就可以让更多的人同样用到它。同样maven也是一个非常完备的项目，我们可以在上面下载到各种我们喜欢想要的jar包，下面将介绍一下如何将本地jar包发布到maven中央仓库。

发布的全部流程

注册Sonatype账号

需要注册一个Sonatype的账号，然后上面提交一个工单，[注册链接](#)，注册之后我们需要提交一个工单，点击Create就可以创建工单了，然后我们需要填写一些它的描述信息，描述信息类似下图：

这里面填写了一个Group Id，这个一般用公司的官网就可以了，然后可能工作人员会在工单的界面进行询问这个网站的所有权是否在你的手里，只需要回答一个yes就可以了，不过如果没有官网的话，用github它所在的网址也是非常不错的。

然后这个工单应该应该就已经提完了，然后需要等待工作人员的审核，如果审核完毕这个工单的状态就会变成RESOLVED状态，当这个状态的时候我们就可以提交审核了。

下载gpg对发布的文件进行签名

gpg应该是一个对称的加密工具，会生成公钥和私钥，[mac下载链接](#)，[windows下载链接](#)，然后就可以通过命令行生成密钥了。

```
gpg --gen-key
```

通过这个命令，会提示出一些东西，我们需要输入用户名，还有邮箱，然后还会弹出来一个对话框我们需要输入密码，其它的东西我们只需要敲回车就可以了，最后我们选择O，就可以了。如下图所示：

然后我们需要把这个gpg的key，上传到服务器，如下图所示：

配置Maven下面的setting.xml

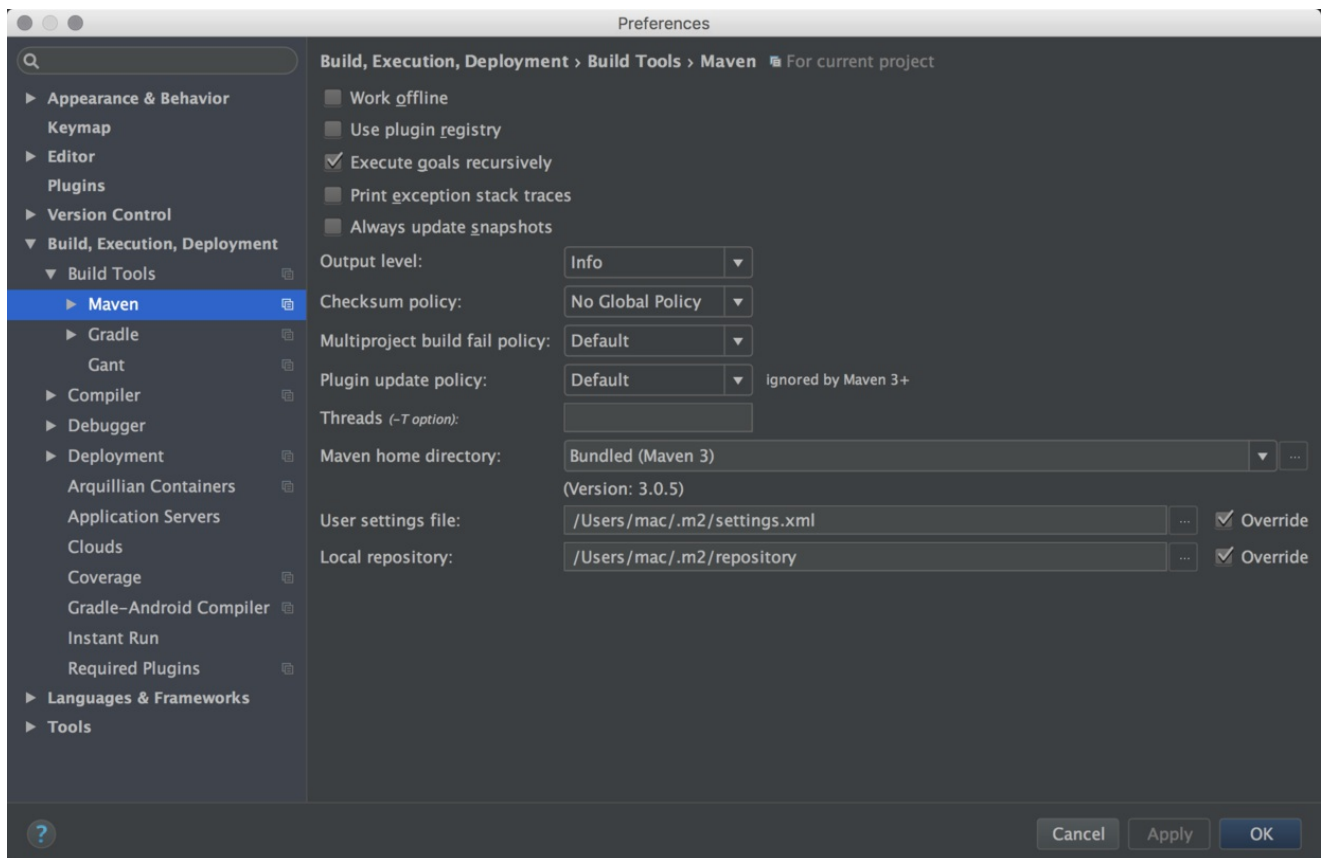
```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/settings-1.0.0.

  <servers>
    <server>
      <id>ossrh</id>
      <username>linSir</username>
      <password>8888888</password>
    </server>
    <server>
      <id>nexus-snapshots</id>
      <username>linSir</username>
      <password>Jx1523,,</password>
    </server>
  </servers>

</settings>
```

这里面有三个地方我们需要注意一下，分别是，，其中username和password需要和我们提交工单的网站的账号密码所对上，id可以随便写，但是需要和pom.xml里面对上。这里面有个小小的坑，我在这里说一下，我们需要配置的是Maven真正用到的setting.xml，我之前一直配置的是另一个Maven下面的xml，就导致一直没有权限做这件事情，希望大家不要掉进坑里。

如果用的是idea的话，可以查到Maven用到的setting.xml的位置，如下图：



配置pom.xml

```
<?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/xsd/maven-4.0.0"
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.github.dotengine</groupId>
  <artifactId>dotEngine-java-sdk</artifactId>
  <version>0.1.1</version>
  <packaging>jar</packaging>
  <name>dotEngine-java-sdk</name>
  <description>dot engine java sdk</description>
  <url>https://github.com/dotEngine/dotEngine-java-sdk</url>

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

  <scm>
    <connection>scm:git:https://github.com/dotEngine/dotEngine-java-sdk</connection>
    <developerConnection>scm:git:git@github.com:dotEngine/dotEngine-java-sdk</developerConnection>
    <url>git@github.com:dotEngine/dotEngine-java-sdk</url>
    <tag>0.1.0</tag>
  </scm>
  <issueManagement>
    <system>GitHub Issues</system>
    <url>https://github.com/dotEngine/dotEngine-java-sdk/issues</url>
  </issueManagement>
  <ciManagement>
    <system>Web site</system>
    <url>http://dot.cc</url>
  </ciManagement>

  <developers>
    <developer>
      <name>denghaizhu</name>
      <email>haizhu12345@gmail.com</email>
    </developer>
    <developer>
      <name>liulianxiang</name>
```

```

    <email>notedit@gmail.com</email>
  </developer>
</developers>

<properties>
  <github_account>dotEngine</github_account>
  <jwt.version>0.7.0</jwt.version>
  <jackson.version>2.8.1</jackson.version>
  <nimbus.jwt>4.13.1</nimbus.jwt>
  <jdk.version>1.7</jdk.version>
  <maven.jar.version>3.0.2</maven.jar.version>
  <maven.compiler.version>3.5.1</maven.compiler.version>

  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <buildNumber>${user.name}-${maven.build.timestamp}</buildNumber>
  <bouncycastle.version>1.55</bouncycastle.version>
  <easymock.version>3.4</easymock.version>
  <junit.version>4.12</junit.version>
  <powermock.version>1.6.5</powermock.version>
  <failsafe.plugin.version>2.19.1</failsafe.plugin.version>
</properties>
<dependencies>
  <dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt</artifactId>
    <version>${jwt.version}</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>${jackson.version}</version>
    <scope>provided</scope>
  </dependency>
</dependencies>

<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>

<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-source-plugin</artifactId>
      <version>2.2.1</version>
      <executions>
        <execution>
          <id>attach-sources</id>
          <goals>
            <goal>jar-no-fork</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-javadoc-plugin</artifactId>
      <version>2.9.1</version>
      <executions>
        <execution>
          <id>attach-javadocs</id>
          <goals>
            <goal>jar</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>

```

```

<artifactId>maven-gpg-plugin</artifactId>
<version>1.5</version>
<executions>
  <execution>
    <id>sign-artifacts</id>
    <phase>verify</phase>
    <goals>
      <goal>sign</goal>
    </goals>
  </execution>
</executions>
</plugin>

<!--<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>-->

</plugins>
</build>

</project>

```

这个是我配置的我的项目的pom.xml，大家可以参考一下，注意的就是要和setting里面的id对上，还有一些基本的配置，大家可以参考我的项目，改成自己项目需要的。

上传

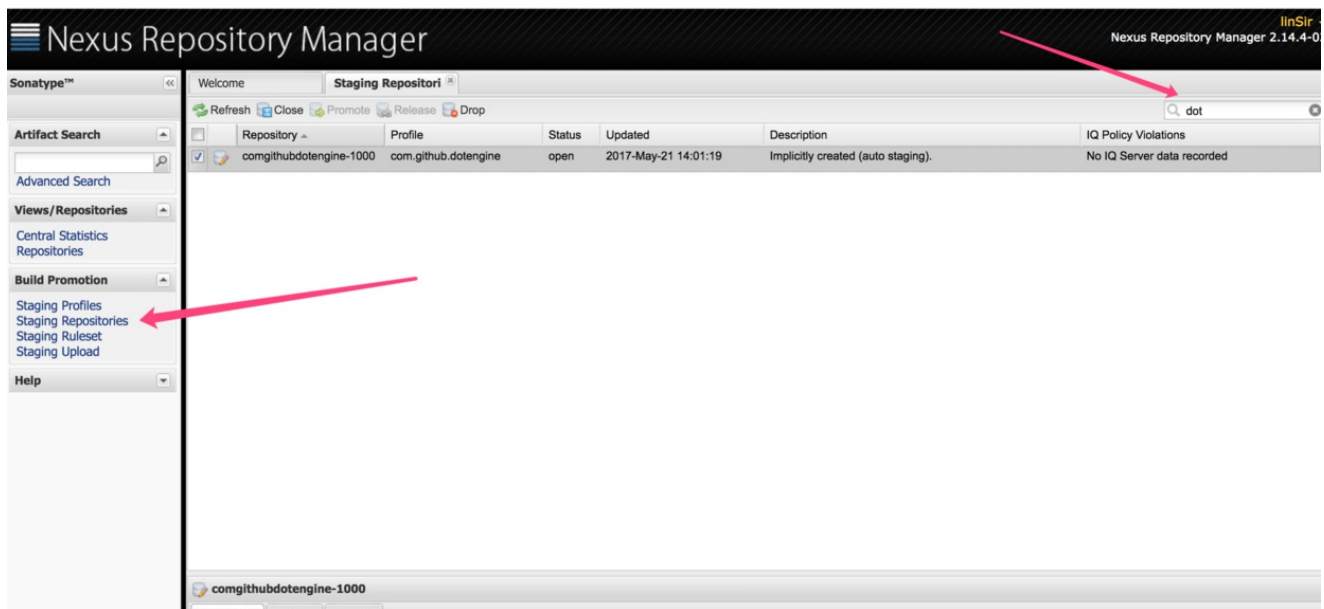
```
mvn clean deploy -X
```

```

Terminal
+ Uploading: https://oss.sonatype.org/service/local/staging/deploy/maven2/com/github/dotengine/dotEngine-java-sdk/0.1.1/dotEngine-java-sdk-0.1.1.jar.asc
Uploaded: https://oss.sonatype.org/service/local/staging/deploy/maven2/com/github/dotengine/dotEngine-java-sdk/0.1.1/dotEngine-java-sdk-0.1.1.jar.asc (801 B at 1.2 kB/s)
X [DEBUG] Using transporter WagonTransporter with priority -1.0 for https://oss.sonatype.org/service/local/staging/deploy/maven2/
[DEBUG] Using connector BasicRepositoryConnector with priority 0.0 for https://oss.sonatype.org/service/local/staging/deploy/maven2/ with username=linSir, password=***
Uploading: https://oss.sonatype.org/service/local/staging/deploy/maven2/com/github/dotengine/dotEngine-java-sdk/0.1.1/dotEngine-java-sdk-0.1.1.pom.asc
Uploaded: https://oss.sonatype.org/service/local/staging/deploy/maven2/com/github/dotengine/dotEngine-java-sdk/0.1.1/dotEngine-java-sdk-0.1.1.pom.asc (801 B at 1.1 kB/s)
[DEBUG] Using transporter WagonTransporter with priority -1.0 for https://oss.sonatype.org/service/local/staging/deploy/maven2/
[DEBUG] Using connector BasicRepositoryConnector with priority 0.0 for https://oss.sonatype.org/service/local/staging/deploy/maven2/ with username=linSir, password=***
Uploading: https://oss.sonatype.org/service/local/staging/deploy/maven2/com/github/dotengine/dotEngine-java-sdk/0.1.1/dotEngine-java-sdk-0.1.1-sources.jar.asc
Uploaded: https://oss.sonatype.org/service/local/staging/deploy/maven2/com/github/dotengine/dotEngine-java-sdk/0.1.1/dotEngine-java-sdk-0.1.1-sources.jar.asc (801 B at 940 B/s)
[DEBUG] Using transporter WagonTransporter with priority -1.0 for https://oss.sonatype.org/service/local/staging/deploy/maven2/
[DEBUG] Using connector BasicRepositoryConnector with priority 0.0 for https://oss.sonatype.org/service/local/staging/deploy/maven2/ with username=linSir, password=***
Uploading: https://oss.sonatype.org/service/local/staging/deploy/maven2/com/github/dotengine/dotEngine-java-sdk/0.1.1/dotEngine-java-sdk-0.1.1-javadoc.jar.asc
Uploaded: https://oss.sonatype.org/service/local/staging/deploy/maven2/com/github/dotengine/dotEngine-java-sdk/0.1.1/dotEngine-java-sdk-0.1.1-javadoc.jar.asc (801 B at 1.0 kB/s)
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 29.063 s
[INFO] Finished at: 2017-05-21T14:01:36+08:00
[INFO] Final Memory: 27M/303M
[INFO] -----

```

最后我们就能看到结果了,然后登陆[仓库的网站](#)网站，查看我们提交的代码



联系工作人员close掉这个issues

在工单的下面，就可以通知工作人员关闭这个工单的，同时我们需要在上图的界面中close掉这个，close的时候可能需要输入一些描述性信息，当关闭成功之后，再次选中这个，然后右侧还有一个release，点击之后同样需要输入描述信息，然后等几个小时，就可以在Maven的中央仓库中找到我们提交的库了。

[中央仓库地址](#)

搜索到的效果图：



以上便是上传的全部过程了，虽然有点折腾，但还是痛并快乐着的，等下一次，就不需要这么麻烦了~

下一次的操作，只需要，close掉这个，然后再重新release即可

