链滴

Gradle- 发布 jar 到中央仓库

作者：xjtushilei

原文链接：https://ld246.com/article/1527665929426

来源网站：链滴

用Gradle部署到OSSRH - 介绍

就像Gradle可以Central Repository中的jar一样，它也可以配置为发布到OSSRH，一样简单。

元数据和签名

为了使用Gradle将您的组件部署到OSSRH，您必须满足pom.xml中所有的东西，并提供所需的已签组件。

MavenGradle的插件可以处理元数据，生成所需的pom.xml文件，并负责将构建输出部署到存储库。signing插件允许您获取由标准Gradle任务创建的组件，并签名：

```
apply plugin: 'maven'

apply plugin: 'signing'
```

Jar文件

对于典型的Java项目，您可以添加javadocJar以及sourcesJar任务

```
task javadocJar(type: Jar) {
    classifier = 'javadoc'
    from javadoc
}
task sourcesJar(type: Jar) {
    classifier = 'sources'
    from sourceSets.main.allSource
}
```

并将它们与项目jar本身一起挂接到artifacts集合中：

```
artifacts {
    archives javadocJar, sourcesJar
}
```

签署文物

已定义的工件可以使用签名

```
signing {
    sign configurations.archives
}
```

元数据定义和上传

为了准备实际上传，您必须在maven插件的帮助下定义所有元数据。组和版本在顶层项目中设置，而artifactId为archiveTask配置。

```
group = "com.example.applications"
```

原文链接：Gradle- 发布 jar 到中央仓库

```
archivesBaseName = "example-application"
version = "1.4.7"
```

生成的pom文件必须进行签名，然后必须上传所有已签名的工件。所有这些都可以配置为配置的一部分ploadArchives。

```
uploadArchives {
  repositories {
    mavenDeployer {
      beforeDeployment { MavenDeployment deployment -> signing.signPom(deployment) }
      repository(url: "https://oss.sonatype.org/service/local/staging/deploy/maven2/") {
        authentication(userName: ossrhUsername, password: ossrhPassword)
      }
      snapshotRepository(url: "https://oss.sonatype.org/content/repositories/snapshots/") {
        authentication(userName: ossrhUsername, password: ossrhPassword)
      }
      pom.project {
        name 'Example Application'
        packaging 'jar'
        // optionally artifactId can be defined here
        description 'A application used as an example on how to set up
          pushing  its components to the Central Repository.'
        url 'http://www.example.com/example-application'
        scm {
          connection 'scm:svn:http://foo.googlecode.com/svn/trunk/'
          developerConnection 'scm:svn:https://foo.googlecode.com/svn/trunk/'
          url 'http://foo.googlecode.com/svn/trunk/'
        }
        licenses {
          license {
            name 'The Apache License, Version 2.0'
            url 'http://www.apache.org/licenses/LICENSE-2.0.txt'
          }
        }
        developers {
          developer {
            id 'manfred'
            name 'Manfred Moser'
            email 'manfred@sonatype.com'
          }
        }
      }
    }
  }
}
```

您的项目的依赖项将被插入到生成的项目中。

证书

用于签名和上传的凭证可以存储在gradle.properties用户主目录中的文件中。内容将如下所示

```
signing.keyId=YourKeyId
signing.password=YourPublicKeyPassword
```

```
signing.secretKeyRingFile=PathToYourKeyRingFile
ossrhUsername=your-jira-id
ossrhPassword=your-jira-password
```

部署

有了这个配置，就可以开始部署

```
gradle uploadArchives
```

将部署发布到中央存储库

部署完成后，您可以继续手动发布组件。

链接

- Gradle网站
- Gradle发布文档
- 使用Nexus在存储库管理中将Gradle与Nexus Staging Suite一起使用
- Nexus Repository Manager文档示例项目中的Gradle示例
- 关于部署到OSSRH的博客文章
- Gradle Nexus分段插件
- 关于在TheNexus上使用Gradle Nexus Staging Plugin的博客文章
- 用gradle-mvn-push发布Android档案（AAR）