1. **maven基本命令**

编译源代码

$ mvn compile

打包

$ mvn package

运行程序中的单元测试

$ mvn test

清除项目的maven产生代码

$ mvn clean

在本地Repository中安装jar包，install前会自动执行compile, package, test

$ mvn install

install一般会和clean一起使用，确保包是最新的

$ mvn clean install

若install时不想运行test，有两种方法

第一种，命令行方式

$ mvn install -Dmaven.test.skip=true

或者

$ mvn install -DskipTests

第二种，在pom.xml中添加

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.19.1</version>

<configuration>

<skipTests>true</skipTests>

</configuration>

</plugin>

1. **发布jar包到仓库**

在maven配置文件中添加账户密码

$ vim ~/.m2/settings.xml

添加如下信息

<?xml version="1.0" encoding="UTF-8"?>

<settings>

<mirrors>

<mirror>

<!-- This sends everything else to /public -->

<id>nexus</id>

<mirrorOf>\*</mirrorOf>

<url>http://nexus.d.xiaomi.net/nexus/content/groups/public</url>

</mirror>

</mirrors>

<profiles>

<profile>

<id>development</id>

<repositories>

<repository>

<id>central</id>

<url>http://nexus</url>

<releases><enabled>true</enabled></releases>

<snapshots><enabled>true</enabled></snapshots>

</repository>

</repositories>

<pluginRepositories>

<pluginRepository>

<id>central</id>

<url>http://nexus</url>

<releases><enabled>true</enabled></releases>

<snapshots><enabled>true</enabled></snapshots>

</pluginRepository>

</pluginRepositories>

</profile>

</profiles>

<activeProfiles>

<activeProfile>development</activeProfile>

</activeProfiles>

<servers>

<server>

<id>deploy</id>

<username>fanyuguang</username>

<password>Cloud666</password>

</server>

<server>

<id>archiva.internal</id>

<username>fanyuguang</username>

<password>Cloud666</password>

</server>

<server>

<id>archiva.snapshots</id>

<username>fanyuguang</username>

<password>Cloud666</password>

</server>

</servers>

</settings>

上传jar包到本地maven库

$ mvn install:install-file -Dfile=/home/fanyuguang/Project/tensorflow-serving-1.0.jar -DgroupId=com.xiaomi.tensorflow -DartifactId=tensorflow-serving -Dversion=1.0 -Dpackaging=jar -DrepositoryId=archiva.snapshots

发布jar包到私服

mvn -e deploy:deploy-file -Dfile=/home/fanyuguang/Project/tensorflow-serving-1.0.jar -DgroupId=com.xiaomi.tensorflow -DartifactId=tensorflow-serving -Dversion=1.0-SNAPSHOT -Dpackaging=jar -Durl=http://nexus.d.xiaomi.net/nexus/content/repositories/snapshots/ -DrepositoryId=archiva.snapshots

其中-Dfile表示需要上传的jar包的绝对路径，-DgroupId，-DartifactId和-Dversion对应该包上传后在maven中的坐标定位，-Dpackaging表示发布的文件的类型，-DrepositoryId为上传到的服务器的id

使用IntelliJ IDEA中maven发布jar包到私服

设置<packaging>标签为<packaging>jar</packaging>

在pom.xml中添加

<distributionManagement>

<snapshotRepository>

<id>archiva.snapshots</id>

<name>Archiva Snapshot Repository</name>

<url>http://nexus.d.xiaomi.net/nexus/content/repositories/snapshots/</url>

</snapshotRepository>

</distributionManagement>

1. **pom.xml详解**

<project>标签

<properties>标签

pom.xml中用到的一些属性

<dependencies>标签

项目中引入的maven依赖项

<build>标签

<extensions>标签

<plugins>标签

使用的插件列表