maven的学习

# maven 的介绍

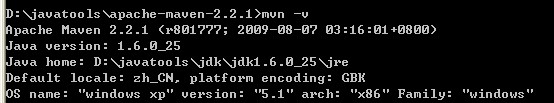
从字面解释maven是知识渊博、经验丰富的专家或怪才的意思。深入研究和使用maven，确实让人感到它确实沉淀了Java项目构建领域中的丰富知识和经验，并以一种高度可复用的形式出现在你的面前。maven的开发者在他们开发网站上指出，maven的目标是要使得项目的构建更加容易，它把编译、打包、测试、发布等开发过程中的不同环节有机的串联了起来，并产生一致的、高质量的项目信息，使得项目成员能够及时地得到反馈。maven有效地支持了测试优先、持续集成，体现了鼓励沟通，及时反馈的软件开发理念。如果说Ant的复用是建立在"拷贝--粘贴"的基础上的，那么Maven通过插件的机制实现了项目构建逻辑的真正复用。

# maven的环境搭建

首先去官方网站下载之：http://www.apache.org/dyn/closer.cgi/maven/binaries/apache-maven-2.2.1-bin.zip，我选择的是截至目前最新版本maven2.2.1版本

## 设置环境变量：

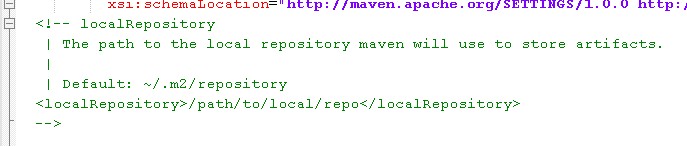
Maven2下载完后，我将其解压到我的计算机的D:\javatools\apache-maven-2.2.1下，目录结构如下：  


环境变量：  
        我的电脑----属性----高级----环境变量，点击“系统变量”下的新建，输入：变量名 MAVEN\_HOME; 变量值D:\javatools\apache-maven-2.2.1，在系统变量列表中找到系统变量path，打开之，在变量值中追加”;%MAVEN\_HOME%\bin”，至此环境变量设 置完毕。  
        检查一下是否已经完成安装，打开dos窗口，输入mvn –v，如果出现以下信息则表示maven2已经安装成功：  


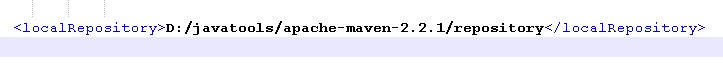
## 本地仓库配置：

D:\javatools\apache-maven-2.2.1\conf目录下的setting.xml文件中修改两处：

第一处：



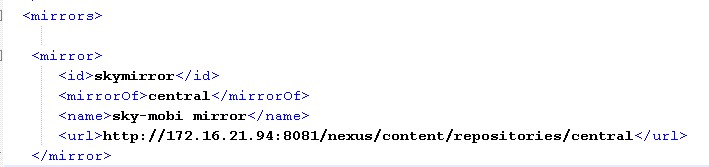
修改为自己对应的本地仓库目录：



第二处：



修改为：sky-mobi公司己经有架设一台mavan仓库，供大家使用



# Maven生命周期

# 常用的Maven命令

1. 创建Maven的普通java项目：

mvn archetype:create -DgroupId=packageName -DartifactId=projectName

2. 创建Maven的Web项目：

Mvn archetype:create -DgroupId=packageName -DartifactId=webappName –DarchetypeArtifactId = maven-archetype-webapp

3. 编译源代码： mvn compile

4. 编译测试代码：mvn test-compile

5. 运行测试：mvn test

6. 产生site：mvn site

7. 打包：mvn package

8. 清除产生的项目：mvn clean

9. 生成eclipse项目：mvn eclipse:eclipse， 生成idea项目 mvn idea:idea 这年头现在先进了，不用运行也能导入idea工具上了。

10. 编译测试的内容：mvn test-compile

11. 只打jar包: mvn jar:jar

12. 当开发一个带有很多失败单元测试的系统

mvn test -Dmaven.test.failure.ignore=true

13. 想要整个的跳过测试

mvn install -Dmaven.test.skip=true

14 maven命令把应用打包成相应的tar.gz包

mvn assembly:assembly

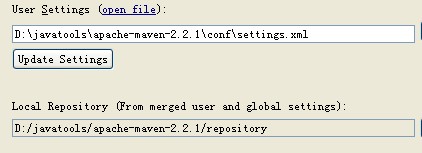
## 基本命令实战：

1 增加eclipse 的maven 插件：

maveneclipseplugin :http://m2eclipse.sonatype.org/sites/m2e

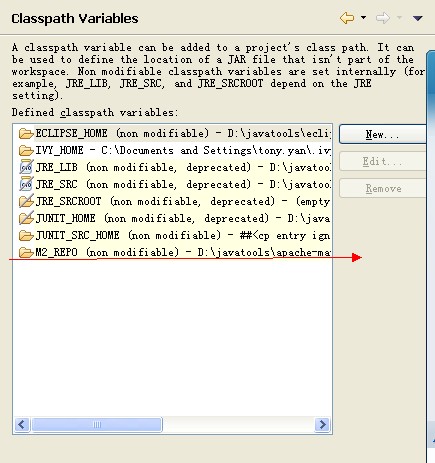
对插件做相应的配置：

windows-preferences-maven-user settings 设置maven本地的setting文件



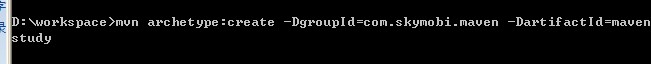
2 如果安装不了插件，则在:eclipse中的 windows-preferences-java-buildpath-classpath中增加：

Maven本地仓库:环境变量 M2\_REPO 指向具体的本地仓库



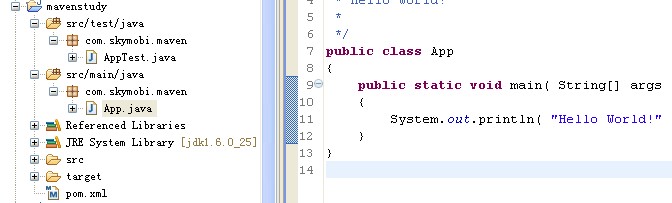
创建第一个java项目

mvn archetype:create -DgroupId=com.skymobi.maven -DartifactId=mavenstudy



成功会提示：build successful

自动创建了mavenstudy 项目目录，进入该目录执行 mvn eclipse:eclipse,执行完之后导入eclipse工具中,查看到的目录结构如下：

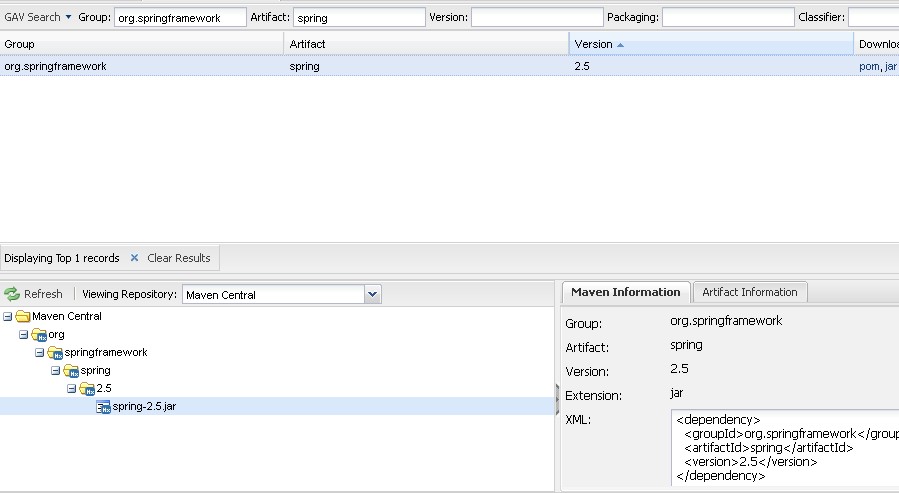


Java工程创建好了：

会涉及到增加依赖包：

以增加spring依赖包为例：(如果要增加其他依赖包同理)

1 打开公司的[http:// 172.16.21.94:8081/nexus/index.html#welcome](http://maven.sky-mobi.com:8081/nexus/index.html#welcome)搜索spring



把其中的 xml   
 <dependency>

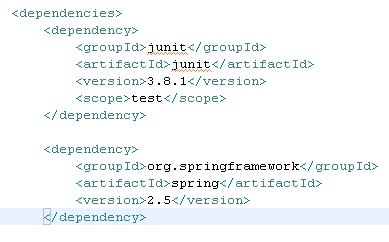
<groupId>org.springframework</groupId>

<artifactId>spring</artifactId>

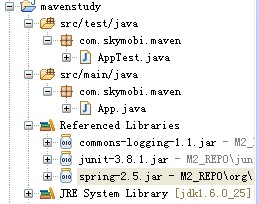
<version>2.5</version>

</dependency>

Copy 到工程应用下的 pom.xml文件里面



执行mvn install/eclipse:eclipse都可以下载相应的jar 包



**创建web应用：**mvn archetype:create -DgroupId=com.skymobi -DartifactId=mavenweb –DarchetypeArtifactId = maven-archetype-webapp

**Pom.xml中的<scope></scope>依赖范围说明：**

****

**常用的打包命令执行：**

**mvn package 要保证所有的单元测试通才能打包成功 ,否则打包不成功**

**mvn package -Dmaven.test.skip=true跳过测试进行打包 jar 或者 war**

**…….等等..**

# 常用Maven Plugin

maven-compiler-plugin

findbugs-maven-plugin

maven-war-plugin

maven-surefire-report-plugin

maven-assembly-plugin

maven-site-plugin

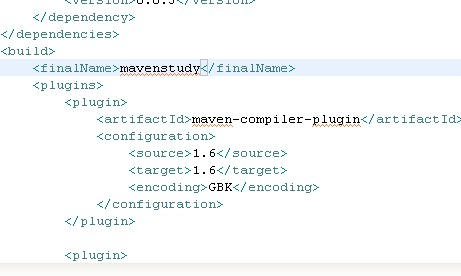
…….

<http://www.infoq.com/cn/news/2011/04/xxb-maven-7-plugin> 插件学习

讲解几个常用的：

如何增加插件：

在工程目录中的 pom.xml 中:



上面<build>节点中的<plugin>部分就是我们要添加的,就这么简单.

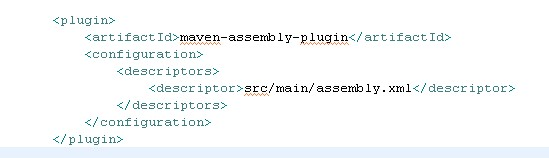
1 添加了maven-compiler-plugin：插件，此后mvn将使用jDK1.6编译程序，编码为GBK

2 maven-war-plugin插件的使用：

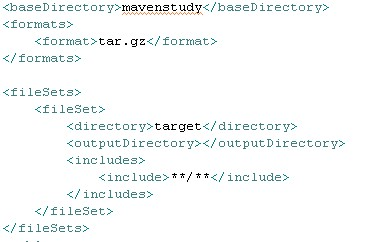


3 maven-assembly-plugin实现自定义打包，打包格式包括 format=zip, gz,tar,tar.gz,tar.bz2

配置如下：



Assemble.xml文件内容如下：



Directory: 打包之后存放的目录，

Basedirectory ：基目录，就是项目目录

打包具体的名字是　mavenstudy.tar.gz

4:…..

# 如何上传jar 致远程仓库中

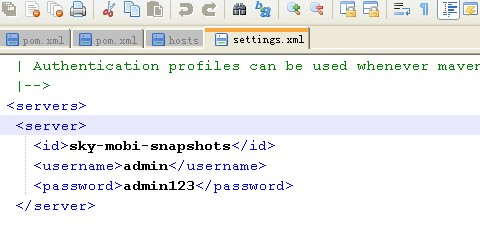
以第一个maven工程为例，上传mavenstudy.jar到远程开发仓库中（指不稳定的版本）

第一步在 工程的pom 文件中增加：



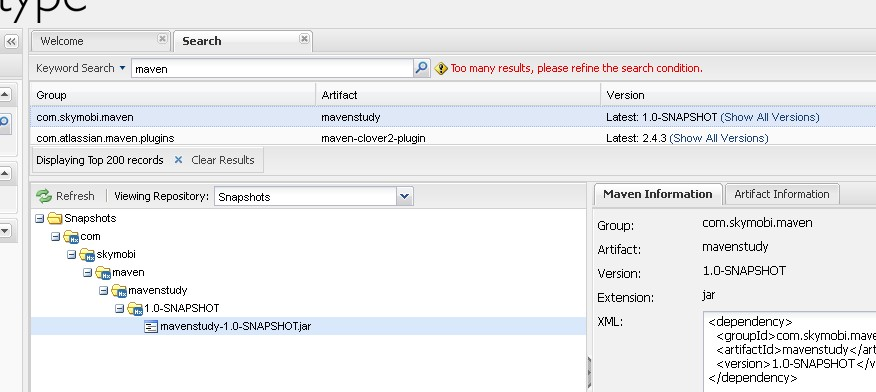
Id名字 和 本地maven 安装的setting 配置id 相同,如图

D:\javatools\apache-maven-2.2.1\conf





D:\workspace\mavenstudy> mvn deploy 命令，则会把应用打成的jar 上传至远程仓库中。这样另一个同事要用的，就可以通过仓库查询平台中看到：



2 上传本地仓库，就容易一些：执行D:\workspace\mavenstudy> mvn install

# 持续集成

1. 软件：TeamCity，Bamboo，Hudson，CruiseControl
2. 任务：打包部署（snapshot），集成测试，代码Check
3. UI简单，分布式，IDE整合

# Maven Repository管理

1. 私服：代理，artifact hosting，search，权限
2. Nexus: Maven Repository管理软件，proxy,hosing(site), user
3. Nexus Indexer: <http://www.mvnsearch.org/maven2/.index/>

内容： 各种Maven库代理、 第三方库、Release库和Snapshots库

备注：Releases库负责发布、Snapshots负责开发