Maven的使用手册

1 Maven的基础命令

1. **mvn compile** 编译命令，作用是将src/main/java下的文件编译为class文件输出到target目录下
2. **mvn test** 测试命令，会执行src/test/java下单元测试类
3. **mvn clean** 清理命令，执行clean会删除target目录的内容
4. **mvn package** 打包命令，对java工程打包成jar包，对web工程打包成war包
5. **mvn install** 安装命令，将工程打成jar包或war包并发布到本地仓库

2 pom基本配置

pom.xml是maven项目的核心配置文件，位于每个工程的根目录，基本配置如下：

1. ： 文件的根节点
2. ： pom.xml使用的对象模型版本
3. ： 项目名称，一般写项目的域名
4. ： 模块名称，子项目名或模块名称
5. ： 产品的版本号
6. ： 打包类型，一般有jar、war、pom等
7. ： 项目的显示名
8. ： 项目描述
9. ： 项目依赖构件配置，配置项目依赖构件的坐标
10. ： 项目构建配置，配置编译、运行插件等
11. ： 依赖的父模块
12. ： 声明依赖的版本信息
13. ： 项目所需插件
14. ： 版本锁定，版本锁定中的并没有真正的在项目中使用，想要使用还需另写，不必再写版本号，因为版本锁定了版本号
15. ： 排除依赖
16. ： 插件配置

3 三套生命周期

1. Clean生命周期 clean
2. Default生命周期 compile test package install deploy
3. Site生命周期 site

每个maven命令对应生命周期的某个阶段，例如：mvn clean命令对应生命周期是clean，所以会执行mvn clean命令。还比如mvn package命令对应的是Default生命周期，所以会依次执行compile、test、package命令。

注意：执行某个生命周期的某个阶段不会影响其他的生命周期，例如可以这样mvn clean package，那么将会依次执行clean、compile、test、package命令。

4 依赖范围

若A依赖B，B中有的依赖，A若需要不必再次依赖，这就叫做**依赖传递。**

若写入了两个依赖，这两个依赖都包含同一个包，这时maven就不知道具体该使用哪个包了，这样的话就会出现依赖冲突。maven为了解决依赖冲突，有一个依赖调节原则：1.谁先声明的就使用谁的。2.A依赖B，A和B中有一个包冲突了，就按照路径优先原则，使用A的。3.排除依赖，比如依赖struts2-spring-plugin,但不想使用它里面的spring-beans，可以这样：

```
        <dependency>
            <groupId>org.apache.struts</groupId>
            <artifactId>struts2-spring-plugin</artifactId>
            <version>2.3.24</version>
            <!--排除了spring-beans-->
            <exclusions>
              <exclusion>
                <groupId>org.springframework</groupId>
                <artifactId>spring-beans</artifactId>
              </exclusion>
            </exclusions>
        </dependency>
```

A依赖B，需要在A的pom.xml文件中添加B的坐标，添加坐标时需要指定依赖范围，默认为compile，依赖范围包括：

依赖范围	对于编译有效	对于测试有效	对于运行时有效	例子
compile	Y	Y	Y	spring-core
test	-	Y	-	junit
provided	Y	Y	-	servlet-api
runtime	-	Y	Y	jdbc驱动
system	Y	Y	-	maven仓库之外的类库