Maven：[https://blog.csdn.net/wfh6732/article/details/62089914/](https://blog.csdn.net/wfh6732/article/details/62089914/%20)

Maven打包插件：<https://blog.csdn.net/daiyutage/article/details/53739452>

# Maven

## 优点如下：

简化了项目依赖管理：

易于上手，对于新手可能一个"mvn clean package"命令就可能满足他的工作

便于与持续集成工具（jenkins）整合

便于项目升级，无论是项目本身升级还是项目使用的依赖升级。

有助于多模块项目的开发，一个模块开发好后，发布到仓库，依赖该模块时可以直接从仓库更新，而不用自己去编译。

maven有很多插件，便于功能扩展，比如生产站点，自动发布版本等

## 缺点如下：

maven是一个庞大的构建系统，学习难度大

maven采用约定优于配置的策略（convention over configuration），虽然上手容易，但是一旦出了问题，难于调试。

当依赖很多时，m2eclipse 老是搞得Eclipse很卡。

中国的网络环境差，很多repository无法访问，比如google code， jboss 仓库无法访问等。

## Maven常见的依赖范围有哪些?

**compile:**编译依赖，默认的依赖方式，在编译（编译项目和编译测试用例），运行测试用例，运行（项目实际运行）三个阶段都有效，典型地有spring-core等jar。

**test:**测试依赖，只在编译测试用例和运行测试用例有效，典型地有JUnit。

**provided:**对于编译和测试有效，不会打包进发布包中，典型的例子为servlet-api,一般的web工程运行时都使用容器的servlet-api。

**runtime:**只在运行测试用例和实际运行时有效，典型地是jdbc驱动jar包。

**system:** 不从maven仓库获取该jar,而是通过systemPath指定该jar的路径。

**import:** 用于一个dependencyManagement对另一个dependencyManagement的继承。

## 仓库

### 本地仓库

Maven会把我们项目所构建出来的jar包等等资源存放在本地仓库中。当我们需要jar包的时候，Maven第一时间也是去本地仓库中寻找jar包

### 中心仓库：

当Maven在本地仓库和私服找不到我们需要的jar包的时候，就去中心仓库中帮我们下载对应的jar包。那Maven怎么知道去哪里下载呢？？其实Maven已经配置好的了

apache-maven-3.2.1libmaven-model-builder-3.2.1orgapachemavenmodel下的POM.xml文件中已经配置好了

### 私有服务器

存储一些jar包的服务器，由于本地仓库不可能拥有大部分的jar包，一个团队要去开发，免不了总是去中心仓库下载，这就非常耗费时间了。如果私有服务器拥有这些jar包，那么每台电脑就不用去中心仓库中下载了...

## maven软件构建的生命周期

软件的生命周期就是以下

- 清除--> 编译-->测试-->报告-->打包（jarwar）-->安装-->部署

## maven坐标

坐标的组成： groupId + artifactId+ version

groupId：组id ,机构名，公司名：好比公司的id，或者是公司包名

alibaba ——-》高德--》5.01版本

artifactId：构建物id ，产品名或者产品的id

version ：版本号

jar包组成：

artifactId-version.jar