**关于MAVEN的补充介绍**

# 为什么要用maven

这好像是大家问得最多的问题，为什么要用maven？

## 1.1 用Maven构建项目

我在翻大家的体系结构作业的时候，看原型代码其实是很麻烦的，总不至于一个个组的代码都导入eclipse然后查看吧？更何况有些组的代码并不是用eclipse写的，项目结构也不一样。这时候我翻到一个组建立的maven项目，诶，这个方便，打开命令行，键入一个mvn install，便能知道项目是不是完整的，顺便能将所有的测试用例跑一遍（如果写了测试用例的话）。要想到这还是在我们院内部，大家都用eclipse的情况下。如果在开源项目里呢？全世界的人，各有各的习惯，各用各的IDE，甚至文本编辑器，如果没有一个统一标准的话，就没办法协同开发了。所以吧，这就是maven的用处，我随便从github上pull一个开源项目，如果是maven项目，那我拿来就可以mvn clean install。而且maven的目录结构都是固定的，也大大节约了看代码的成本。

上面都是大白话，我随便百度了一下，官话如下（摘自：<http://blog.csdn.net/mixiuali/article/details/8251487>）：

* Maven是一个构建工具,服务与构建。使用Maven配置好项目后,输入简单的命令,如:mvn clean install, Maven会帮我们处理那些繁琐的任务。
* Maven是跨平台的。
* Maven最大化的消除了构建的重复。
* Maven可以帮助我们标准化构建过程.所有的项目都是简单一致的,简化了学习成本。

## 1.2 Maven的依赖管理

不知道有没有同学体会过，项目开发的时候，有时候会遇到需要第三方jar包，比如这次有的同学会用到数据库，比如mysql吧，那么就会需要mysql-connector-java.jar。而小组又是共同开发的，则A同学下载了，再拷给B同学，再C同学……高级一点呢，A同学把jar包传到svn或者github上了，小组都下载一下。但是这里要提一句，svn，git等代码管理工具，只是用来管理代码的，jar包并不适合上传（它们又不管理二进制文件对吧），而且如果jar包很大，组员更新一次就要半死……

这时候用maven就很方便啦。在maven项目的配置文件pom.xml里的<dependencies>的标签里添加下面的话：

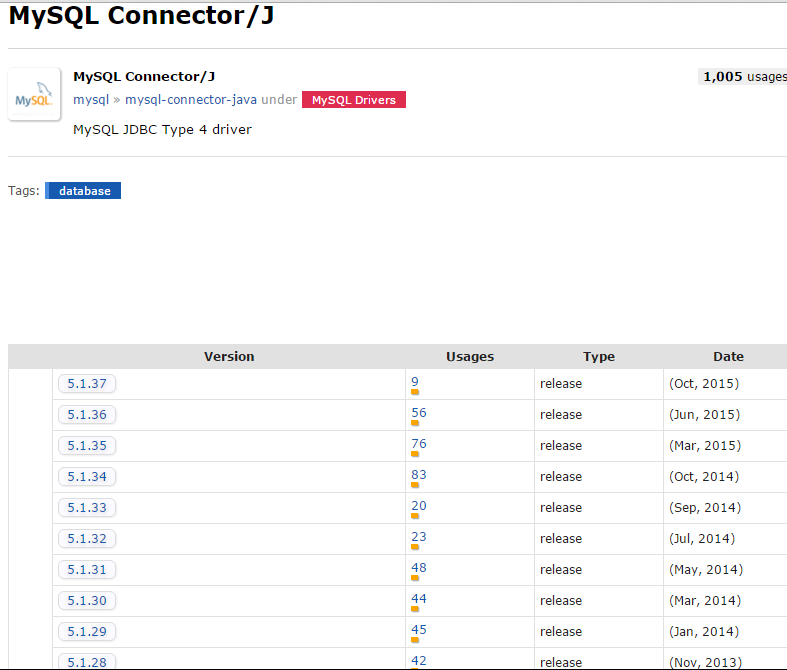
|  |
| --- |
| <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <version>5.1.37</version>  </dependency> |

这时候大家把改动pull下来，运行一下mvn install，就自动从maven的中心库把mysql-connector-java.jar下载下来，并且编译时自动链接，什么都不用操心了呢！

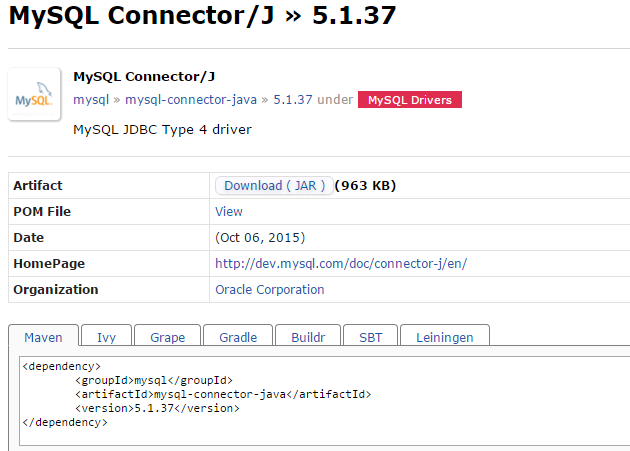
这时候肯定有人要问，上面那段话我怎么知道groupId，artifactId，version要写什么。这个嘛，百度一下……



第一条就是了，点进去，是maven库的网站，如下图，还会告诉你哪个版本用的人比较多呢：



随便点一个版本进去，如下图，直接告诉你怎么写了，多贴心！



## 1.3 第三点

从第一点延伸出来的，我们作为高智商灵长类动物，都难以看懂各种IDE结构的代码，你说Jenkins怎么看得懂并且自动构建的？所以要maven帮忙咯！有了maven固定结构的代码，以及其构建功能，Jenkins就能跑起来了。

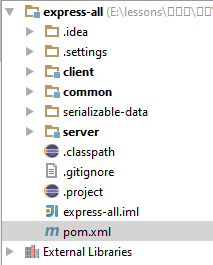
不过讲真，小项目用maven确实有点重了，但是其实也是为了让大家知道并且熟悉一下maven啦，以后再学习起gradle什么的就很快很容易了。

# Maven的配置

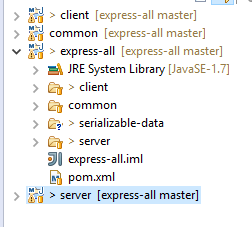
我记得有个同学跟我说大家其实还是不大懂maven的配置，那就在这里再讲一次吧。就以助教们在github上的原型为例好了。

## 2.1 父项目和子项目

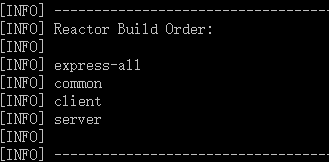
原型里有一个父项目express-all，以及三个子项目client, common, server。他们仨的目录结构如下图所示，也就是父目录express-all中有三个子目录client, common, server。



但在eclipse中的结构可能如下图所示，看起来像四个独立的项目，但是仔细看，express-all下面还是有三个子目录（client， common， server）的，只是这三个目录又分别与三个项目相对应。所以说eclipse对maven的支持不太好呢，看起来就很反人类……



这时候就有人问了，我为什么要建一个父项目，再建三个子项目，我直接建三个独立的项目不就好了吗？其实这里的client和server是依赖于common的，不知道大家有没有注意到，在父项目上跑mvn install的时候的输出：



可以看到，在目录里，三个文件夹的顺序是client、common、server，但是此时顺序被重构了，先编译common，然后才是client和server。其实maven会分析各个子项目的依赖关系，先编译依赖的最顶层。

所以……用maven管理各种子项目是很方便的。

## 2.2 父项目的配置

一般来说每个项目的一开始都有如下的配置：

|  |
| --- |
| <groupId>edu.nju.expressManageSystem</groupId>  <artifactId>express-all</artifactId>  <version>1.0-SNAPSHOT</version>  <packaging>pom</packaging> |

groupId是项目组织唯一的标识符，实际对应JAVA的包的结构，是main目录里java的目录结构。而artifactID就是项目的唯一的标识符，实际对应项目的名称，就是项目根目录的名称。Version就是版本号，这个大家随便填啦。值得一说的是packaging，packaging是打包方式，由于此处是父项目，没有源代码，也就是没有src文件，所以不需要打出jar包，所以此处是pom。

|  |
| --- |
| <modules>  <module>client</module>  <module>common</module>  <module>server</module>  </modules> |

由于是父项目，所以需要制定它有哪些子项目，分别是什么。记得之前有同学问我，为什么代码结构和我们提供的原型一样，但是运行mvn install只出现一个项目的success。我打开父项目的配置文件一看，并没有配置这一项，呃，maven很笨的，如果你不配置子项目是什么，它根本找不到的……

|  |
| --- |
| <dependencyManagement>  <dependencies>  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  <version>3.8.1</version>  <scope>test</scope>  </dependency>  </dependencies>  </dependencyManagement> |

这部分就是maven的依赖了，新建项目的话，都会默认添加junit依赖，默认是版本3，而我们现在一般用的是版本4，这个，自己改改就好了。同样地，可以添加项目的其他依赖，具体可以看看1.2节。

## 2.3 子项目的配置

由于是子项目，都要配<parent>这一项，如下所示：

|  |
| --- |
| <parent>  <groupId>edu.nju.expressManageSystem</groupId>  <artifactId>express-all</artifactId>  <version>1.0-SNAPSHOT</version>  </parent> |

这里要指明父项目是谁，需要填入父项目的groupId，artifactId和version，大家可以对照一下，和父项目里的配置是一样的。

对了，这里要提一下，由于client和server是依赖于common的，所以在它们的配置文件里都要有如下的配置：

|  |
| --- |
| <dependency>  <groupId>edu.nju.expressManageSystem</groupId>  <artifactId>common</artifactId>  <version>1.0-SNAPSHOT</version>  </dependency> |

必须显示声明对common的依赖，在<dependency>中填入common项目的groupId，artifactId和version，还是那句话，maven很笨的，你不说它不懂的。

# 运行Maven

这个有啥好讲的，最简单的一句话，mvn install。不管在命令行还是在eclipse还是在哪里，找到运行maven install的地方就好了。

但其实maven是有很多阶段的，install只是其中一个。阶段是有顺序的，运行后面的阶段会自动把前面的阶段一起运行了，感兴趣的同学可以自行查找maven生命周期的相关资料。

# 最后

其实maven可以用一本很厚的书来介绍，而这里，就先讲到这里吧。有什么不懂的可以发邮件问，随便发呗，助教也是学生，又不会因为你的问题太蠢就扣分的，顶多不回而已，啊哈哈……

以上。