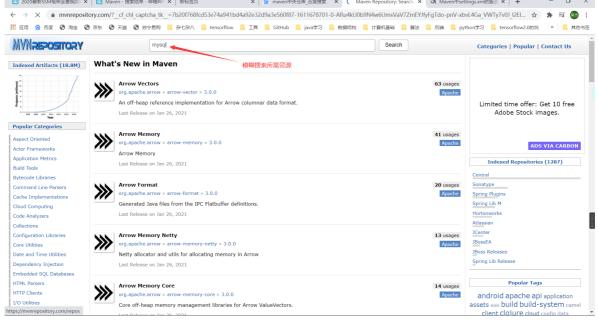
POM文件中的基本信息标签

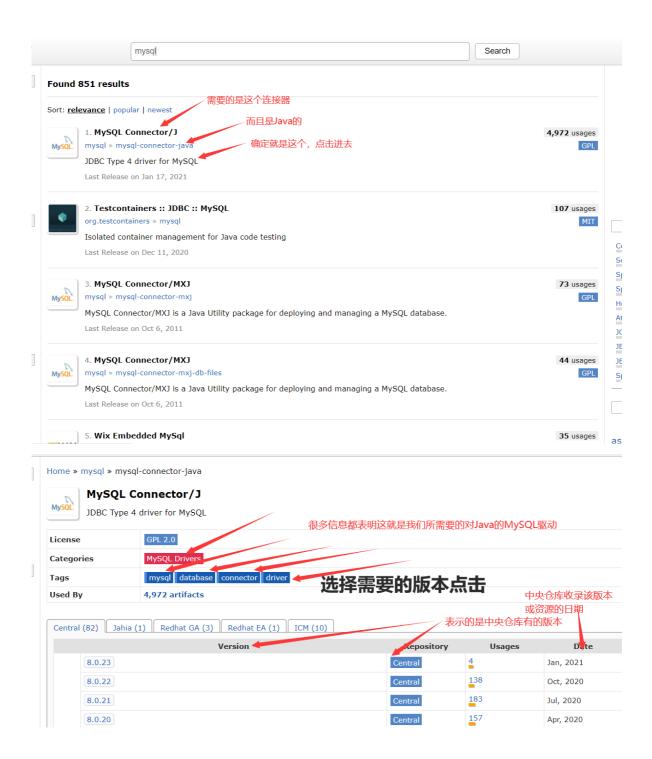
不止一个依赖

</denpendency>

Project Object Model项目对象模型,maven把一个项目的结构和内容抽象成一个这样的一个模型,然后 再xml文件中声明。所以pom.xml文件是maven的灵魂。 该文件中属性标签代表的信息如下: <modelversion> 前面说maven把项目抽象成一个模型,这个标签属性对应的就是模型的版本,对于maven版本的maven2和 maven3,该属性中模型版本号只能是4.0.0 <modelVersion>4.0.0</modelVersion> <groupId> 该属性代表组织id,一般都是公司域名倒写 1、域名倒写: com.baidu 2、域名倒写+项目名: com.baidu.appolo,这是百度的一个项目 该属性代表的是项目的名称,或是模块名称,对应groupId中项目中的子项目 <version> 该属性代表该项目的版本号,如果项目还是在开发中,是不稳定版本,在版本后带-SNAPSHOP(意思是快 照),版本号使用三位数标识,1.0.0 <!--上面三个属性合在一块称之为坐标,是该项目在互联网中的唯一标识。其中grouId,公司域名是互联网 中唯一的,倒写也是唯一的,使得该项目可以在互联网中唯一。而artifactId,是在公司内部项目中模块唯 一的。添加version版本号。就使得这个项目假如上传到互联网仓库中依据这个坐标唯一识别。 我们要使用到别人的资源,也是需要依靠第三方资源的坐标才能获取到的,这是一个项目中必须要有的。 那么如何知道其他资源(项目)的坐标呢? 访问www.mvnreppsitory.com: 中央仓库,在其中搜索对应资源,例如我们需要mysq1驱动,在其中 搜索"mysq1" 具体的操作下面粘贴截图 <groupId>com.studymyself <artifactId>maven-01</artifactId> <version>1.0-SNAPSHOT</version> <packaging></packaging> 该属性表示项目打包的类型,可以为jar、war、rar、ear、pom。默认不写的话是jar <dependencies> <!--该属性中是包含属性标签的<dependency>,表示可以有很多属性标签的<dependency> 因为项目需要很多依赖jar包 --> <denpendency> <!--一个该属性标签表示的一个依赖jar包,用该属性配置依赖,需要填到信息就是坐标 就是说maven把jar也当作一个项目了,依赖资源坐标的获取下面截图所示 --> </denpendency> <denpendency> <!--

```
</dependencies>
 <dependencies>
     <!--依赖,相当于Java代码中的import-->
    <dependency>
     <groupId>mysql</groupId>
     <artifactId>mysql-connector-java</artifactId>
     <version>5.1.37
    </dependency>
     <!--那么maven是怎么运作的呢?
         假设在有jar的情况,maven到本地仓库找mysql文件夹(对应的groupId),选择mysql-
                          夹,选择5.3.37版本文件夹,取出其中的mysql-connector-java-
 connector-java的文件
 5.3.37.jar, 配置到项目中
      <denpendency>
         . . .
     </denpendency>
 </dependencies>
 cproperties>
 该属性是定义一些配置属性的,如project.build.sourceEncoding定义项目构建源码编码方式,设置
 <build>
 该属性定义编译代码和测试代码时的信息配置,如设置编译插件的jdk版本等等
 </build>
 下面两个以后学
 <parent></parent>
 <modules></modules>
🗂 2020最新SSM框架金套視影 🗴 🛅 Maven-按数结果-時標形 🗴 新标签页 🗡 🥞 maven中央仓库-直接搜索 🗴 🕻 Maven Repository: Search. 🗴 🕄 Maven中settings.cml的配影 🗴 🕂
```







✓ Include comment with link to declaration