# spring boot特点

# 使用spring boot可以为微服务提供很好的支持，比如spring cloud

# 微框架，与spring4一起诞生，比如@RestController

# 很少的步骤就可以搭建一个框架，上手简单，节约开发者时间（spring全家桶）

# 可以独立运行而不依赖于容器

# 不需要打包成war包，可以在tomcat中直接运行

# 提供maven极简配置（缺点： 引用不需要的包导致应用臃肿）

# 可视化监控（性能，内存）

# 通过配置简化配置（xml）

# 为微服务铺路，比如dubbo，thrift等

# 使用场景

# 有spring都行

# j2ee/web项目

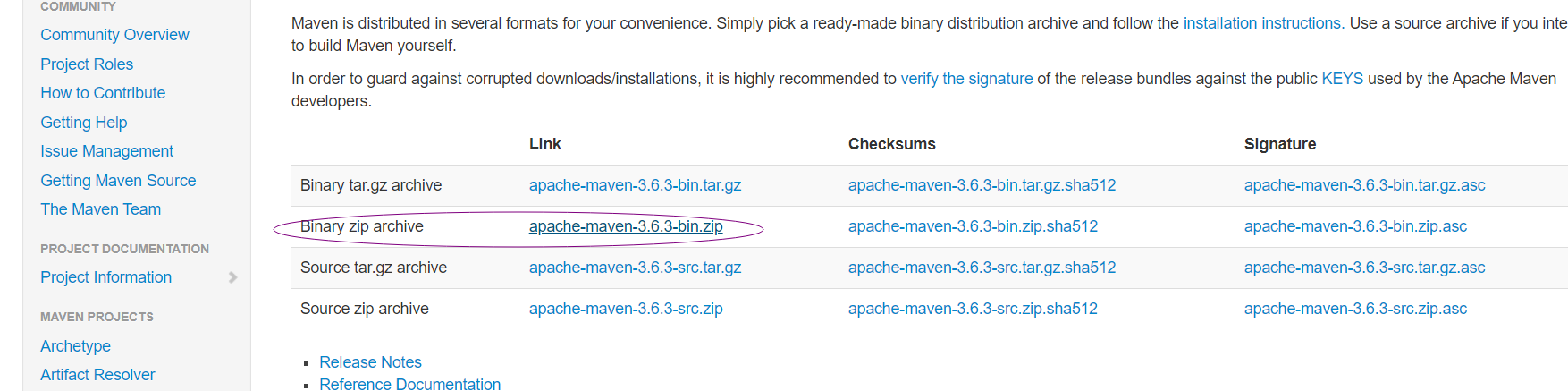
# 微服务（重中之重，spring boot是微服务的基础）

Spring boot搭建过程（eclipse）  
springboot版本，jdk版本，maven版本匹配关系：<https://blog.csdn.net/wj197927/article/details/79563081>

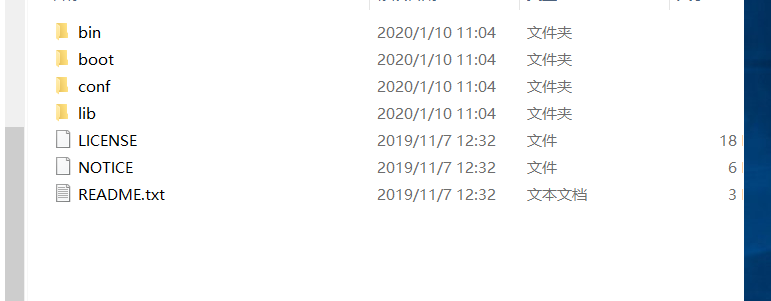
1.安装maven版本管理工具

Maven就是版本管理工具，可以自动下载对应版本依赖，管理版本等。

到官方网站下载maven包<http://maven.apache.org/download.cgi>



下载第二个并解压到某个磁盘中，比如到D盘，文件名为maven，解压后的文件如下：

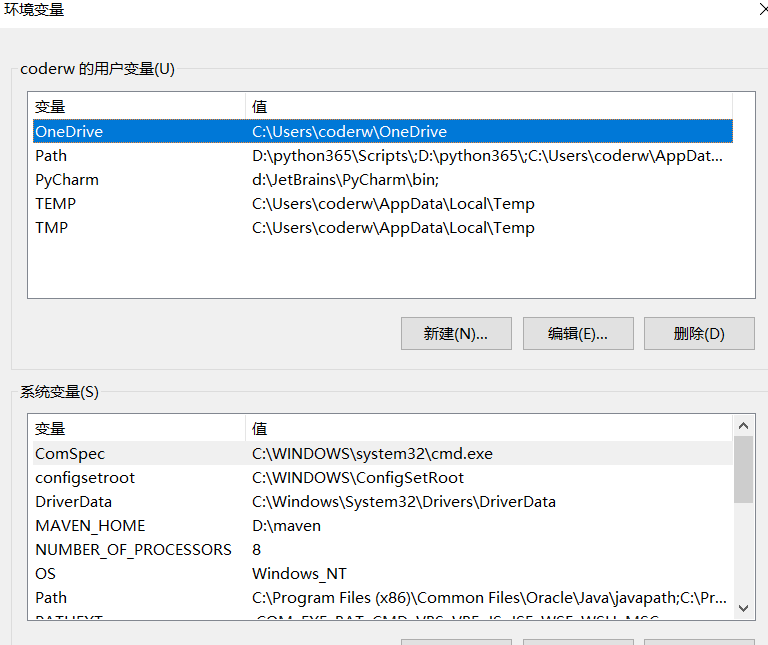


配置环境变量

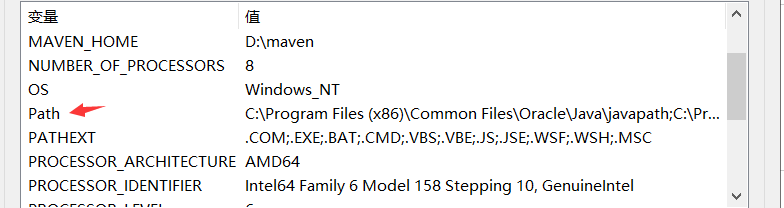
此电脑右键，点击属性弹出如下框

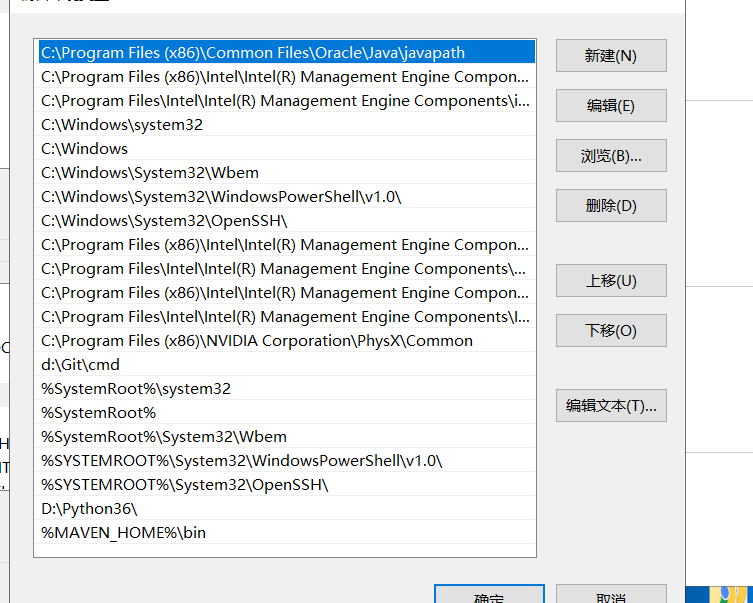


点击高级系统设置然后找到环境变量并点击



修改系统变量中的PAth，找到path并双击打开





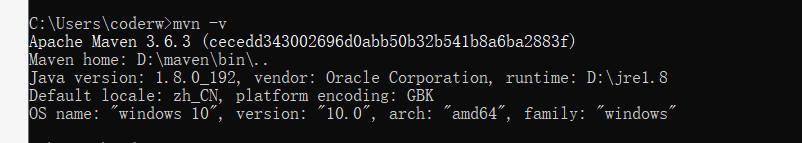
点击新建



在这里输入D:/maven/bin，具体的要看自己的安装目录在哪里，就是，安装目录/bin

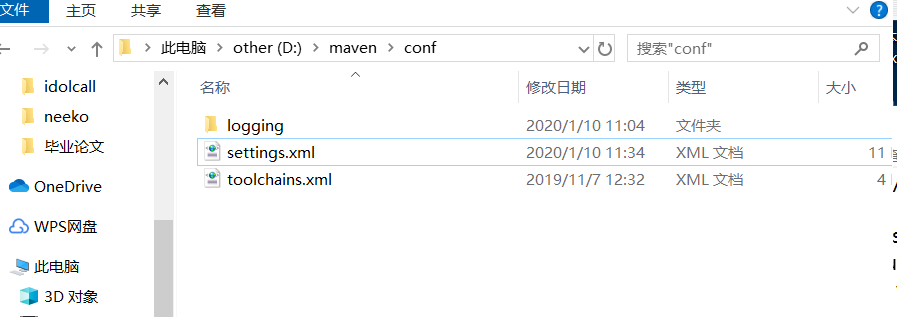
然后一路确定，

打开cmd窗口输入mvn -v

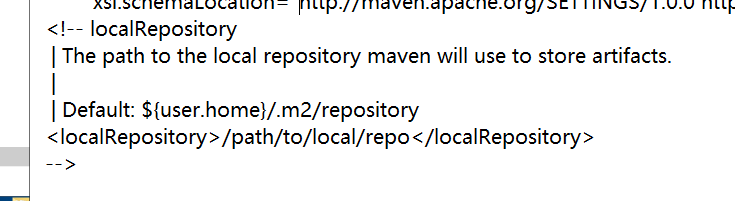


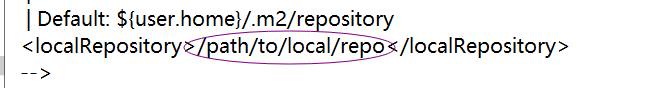
如果显示这样则表示完成。

接下来配置一些设置，打开maven下面的conf文件中的setting.xml文件

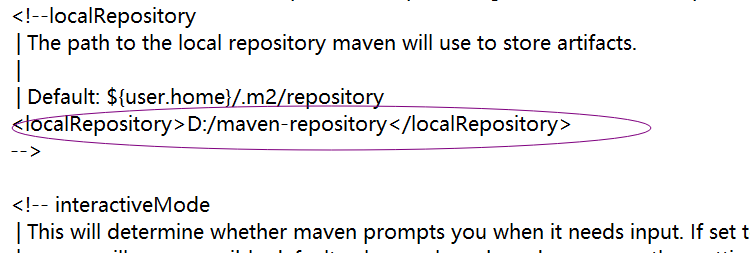


可以用记事本打开，找到localRepository这里

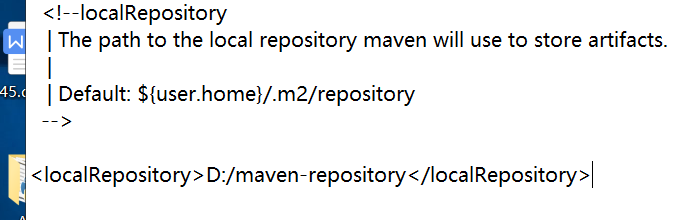




这里的意思是maven下载的包存放的位置，Default: ${user.home}/.m2/repository，是默认存放的位置，这里可以自己设置一个，然后将自己设置的去替换它

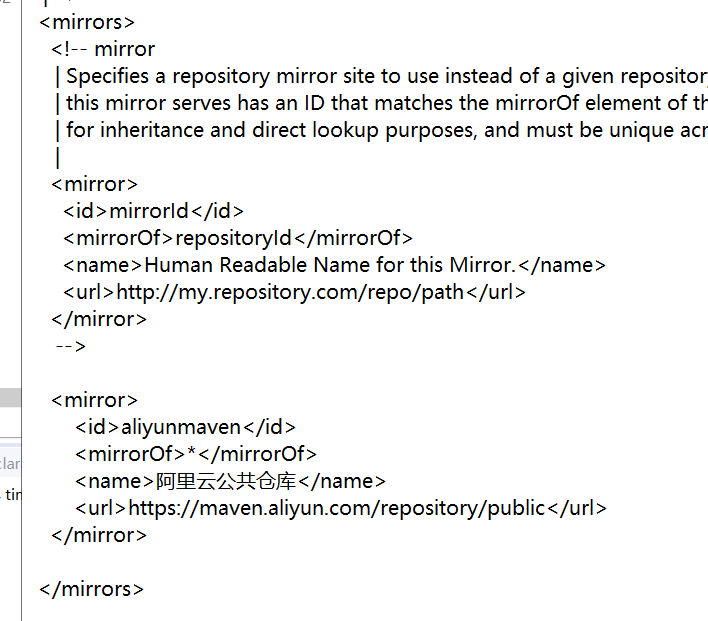


这样，包就可以下载到d盘的maven-repository文件中去了



注意，<!-- -->的意思是注释，所以这里的<localRepository>D:/maven-repository</localRepository>要单独拿出来，不要写到注释里面去了。

再配置mirror，配置mirror的意思是包从哪里下载，默认从外国的网站下载，比较慢，这里可以使用阿里云镜像



<mirrors>

<!-- mirror

| Specifies a repository mirror site to use instead of a given repository. The repository that

| this mirror serves has an ID that matches the mirrorOf element of this mirror. IDs are used

| for inheritance and direct lookup purposes, and must be unique across the set of mirrors.

|

<mirror>

<id>mirrorId</id>

<mirrorOf>repositoryId</mirrorOf>

<name>Human Readable Name for this Mirror.</name>

<url>http://my.repository.com/repo/path</url>

</mirror>

-->

<mirror>

<id>aliyunmaven</id>

<mirrorOf>\*</mirrorOf>

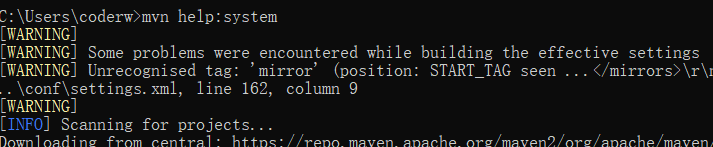
<name>阿里云公共仓库</name>

<url>https://maven.aliyun.com/repository/public</url>

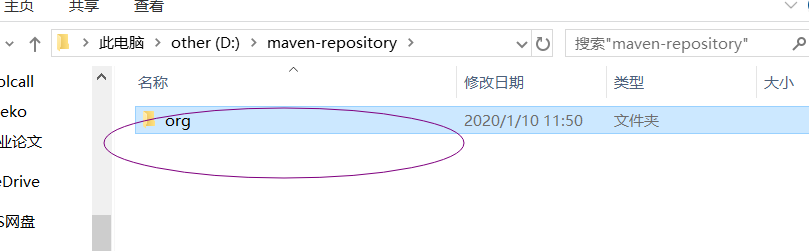
</mirror>

</mirrors>

执行mvn help:system



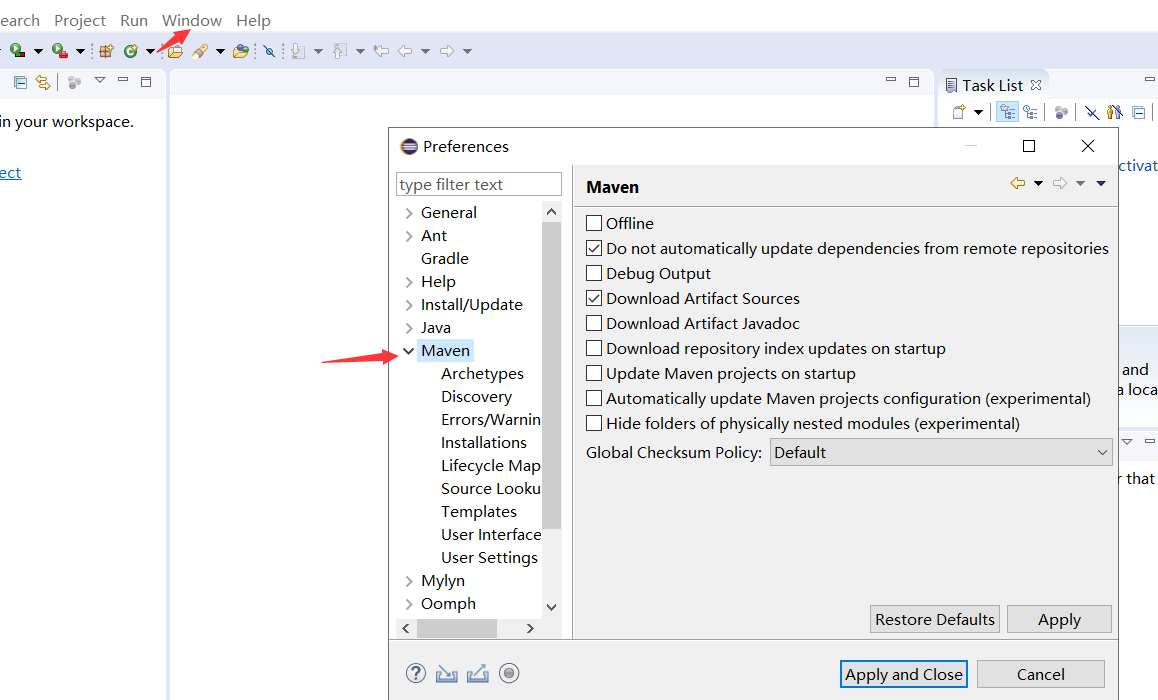
然后查看刚才配置的仓库会发现org包被下载到了该仓库中



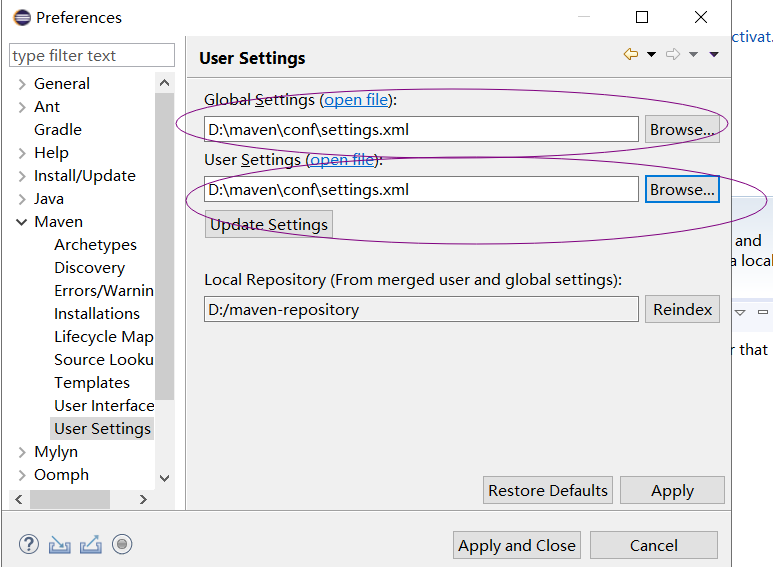
表示仓库配置正确。

打开eclipse

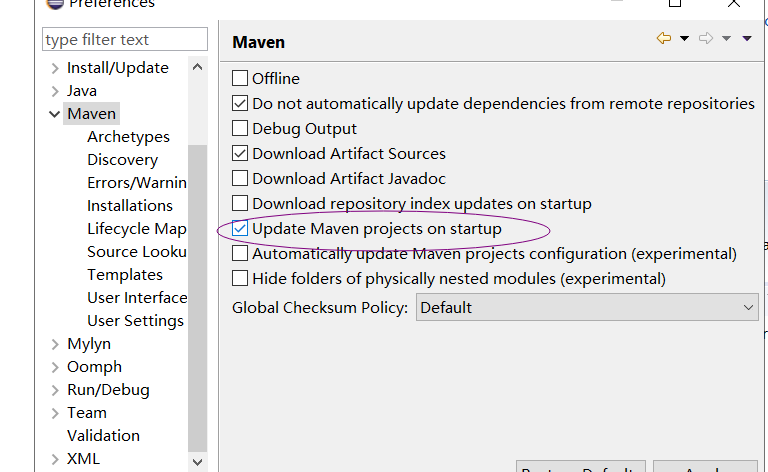
依次点击windows，preference，找到maven



点击User Settings然后将这里的两个文件都换成刚才修改过的setting.xml文件

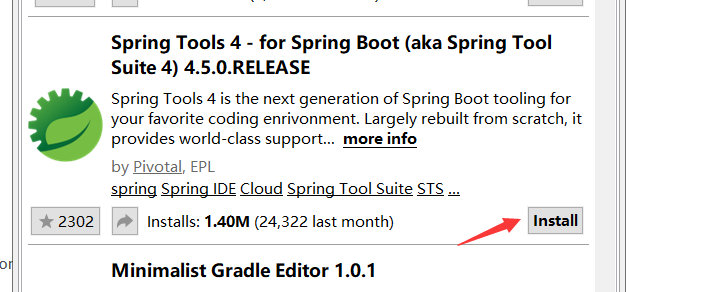


点击maven勾选这里，这样每次启动eclipse都会maven都会自动更新项目



2.安装springboot插件sts

点击eclipse的help，Eclise Marketspsce然后搜索STS，往下拉选择这个，install



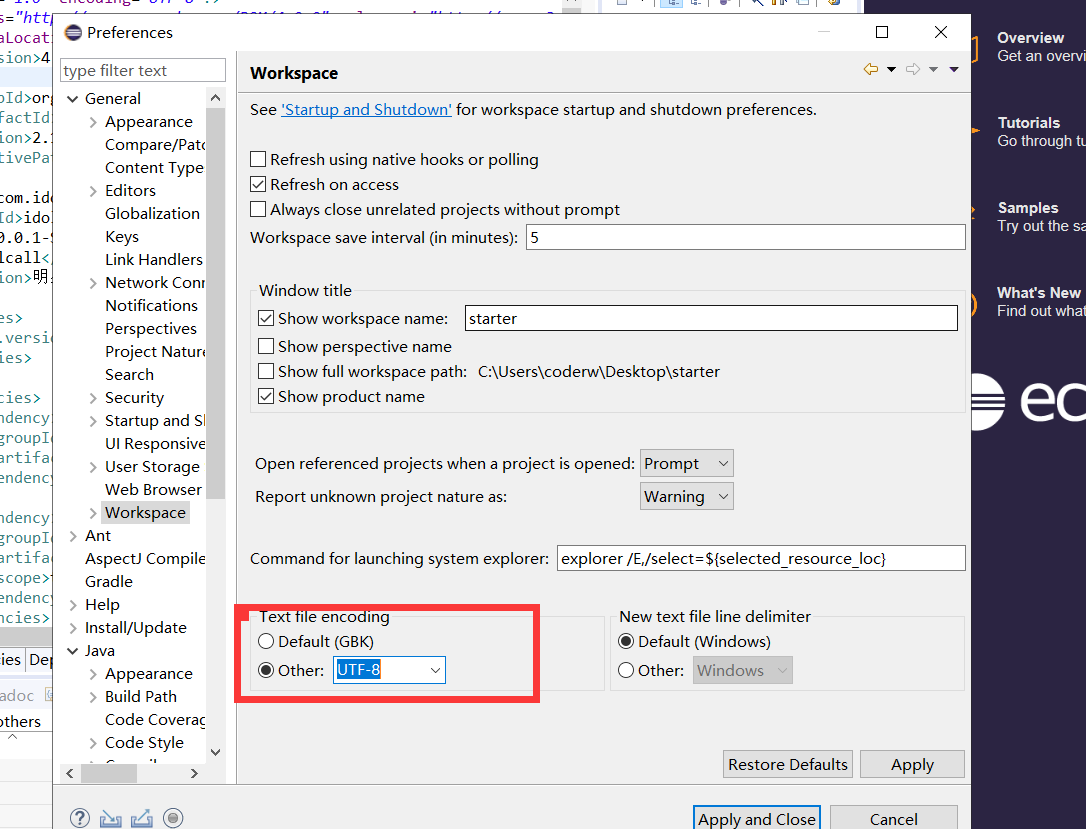
然后confirm，accept，等等，等待安装，估计需要好几分钟，等待安装完毕。安装完需要重启。

3.安装tomcat

使用spring boot不需要安装tomcat，spring boot自己集成了tomcat了

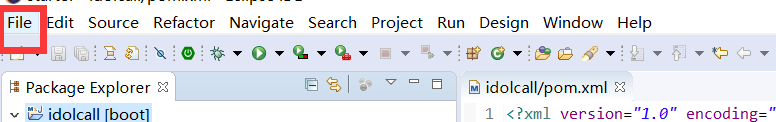
4.搭建spring boot项目

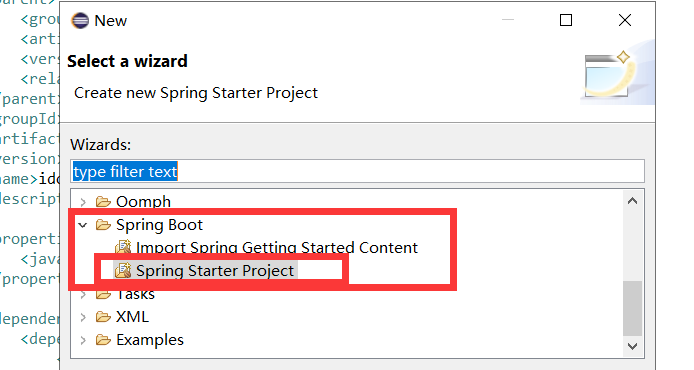
点击windows，preference，将这里编码改成utf-8



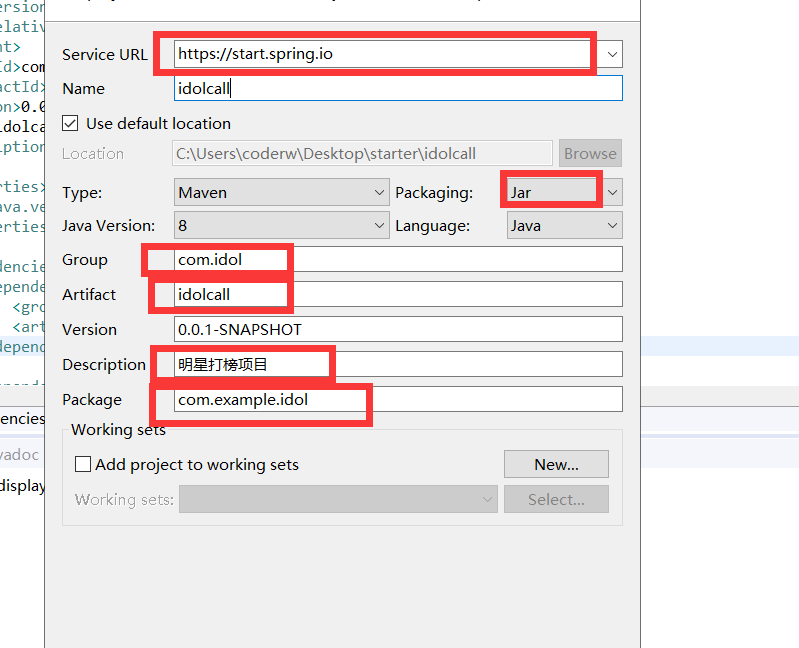
直接使用eclipse创建spring boot项目

打开eclipse ，点击file->new->other 选择





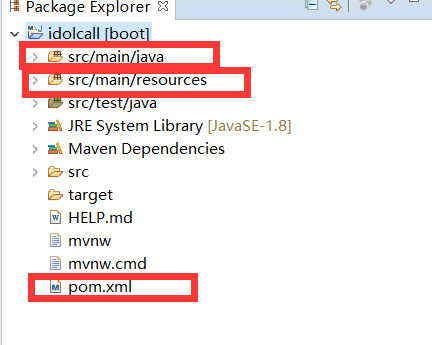
选择spring Starter Project进入一下界面



这里可能会因为网络问题造成无法访问start.spring.io这个网址，将后面的反斜杠去掉等一会儿，重复几次可能就好了。

选择jar包，group是公司名字，artifact是项目名称，package是包名字

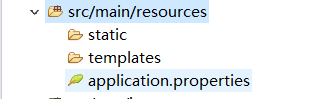
然后finish，finish后可能要等很久，需要下载很多包，等一会儿就好了。最后界面如下。



这里的pom是依赖管理文件，主要是maven用来处理依赖

Src/main/java是放置主要代码的

Src/main/resources存放资源文件

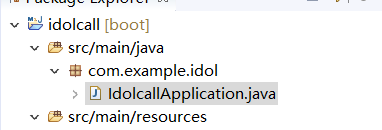


Src/test/java可以忽略，主要存放测试代码



这里的M和J分别表示这是个maven工程，Jar包

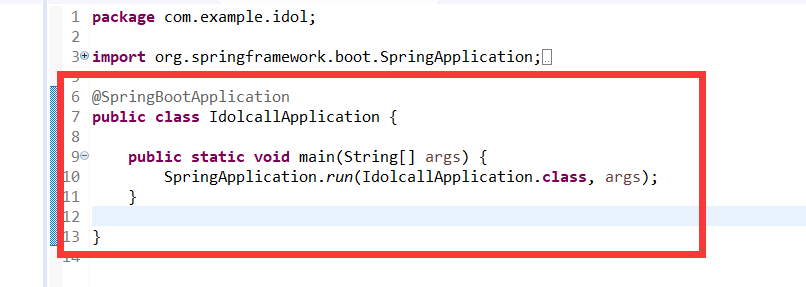
5.启动



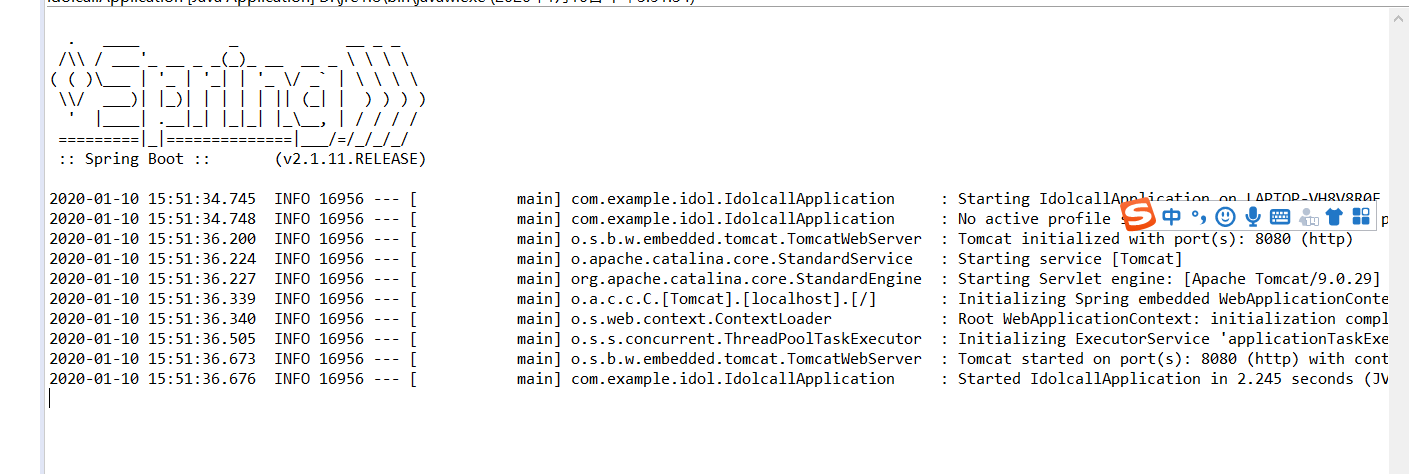
进入com.example.idol文件，进入IdolcallApplication.java然后右键，run as ，java application

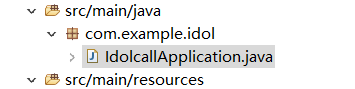
就可以启动项目了。

这里需要理解以下，spring boot的特点就是通过注解管理代码，所以方便。

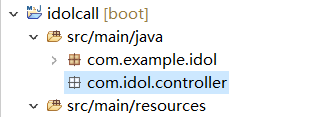


这里有个SpringBootApplication注解，这里就是整个springboot项目的启动入口（面试可能问），整个spring boot项目都是从这里启动的，而且spring boot不需要打包成war，就像一般的java程序一样可以启动，所以方便。启动后如下所示：

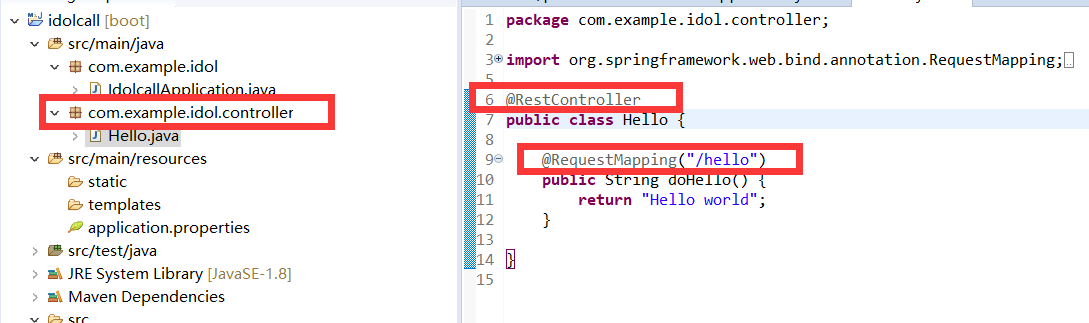


其实这里名字，包名可以随便改变，只要在启动入口的类加一个@SpringbootApplication的注解就可以了。

这里自己写一个简单的hello word的接口试一下这个项目。



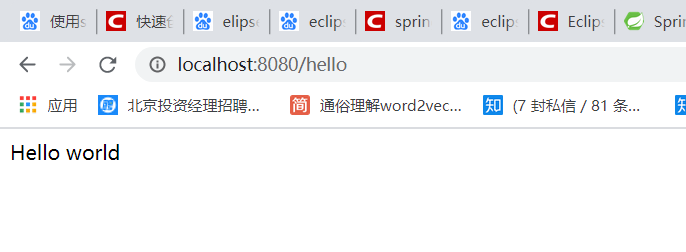
新建一个包，然后新建Hello文件如下。



这里的@RestController是告诉spring boot框架这是一个controller

这里的RequestMapping是告诉spring boot这个controller对应的访问地址为/hello

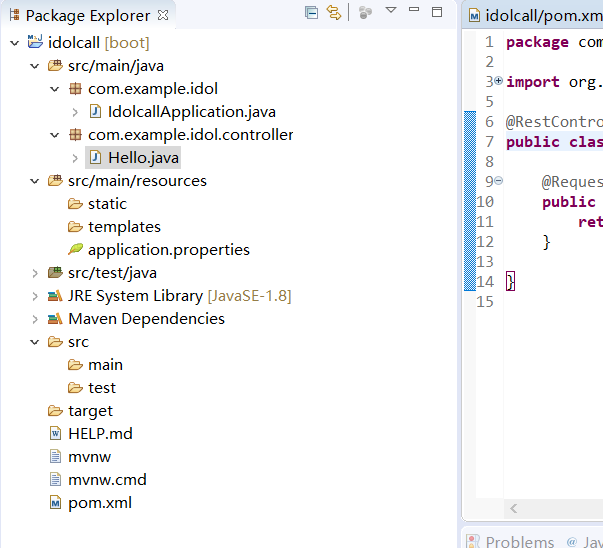
重新启动该项目，然后访问

得到结果如下

访问成功了。注意，这里的所有controller和service子类一定要在启动类所在包的子包下，因为spring boot启动的时候是依次访问其子包来发现各个组件的。

这里项目的结构还可以随便改，具体的等你把上面的弄好了再讲。

而且spring boot还有很多其它组件，很多其它使用方法，很多注解可以用，都是以后再讲。



各个显示的文件和文件夹，包的含义，先把spring boot启动起来。