# Maven是什么？

Maven中文释义为“专家”、“内行”，这里介绍的maven是Apache的一个顶级开源项目，是一个主要服务于java平台的项目自动化构建工具。

构建(可以想象一下 make 的过程)：

清理->编译->测试->打包->部署

Maven可以帮助我们自动化构建过程，我们只需要使用maven配置好项目，然后利用几个简单的maven命令即可完成这些琐碎、重复的工作。

Maven是一款优秀的项目构建工具，但其不仅仅是项目构建工具，它还是一个依赖管理工具和项目信息管理工具。

Maven为全球java开发者提供了一个免费的中央仓库，在中央仓库中基本上可以找到任何可信任的流行的开源类库。

“约定 > 配置 > 编码” ， 在 Maven 中，对项目的目录结构、测试用例命令等做了既有约定，只要遵循这些约定，用户便可畅享maven提供的服务，并可在遨游于各个不同项目之间时免去额外的学习成本。

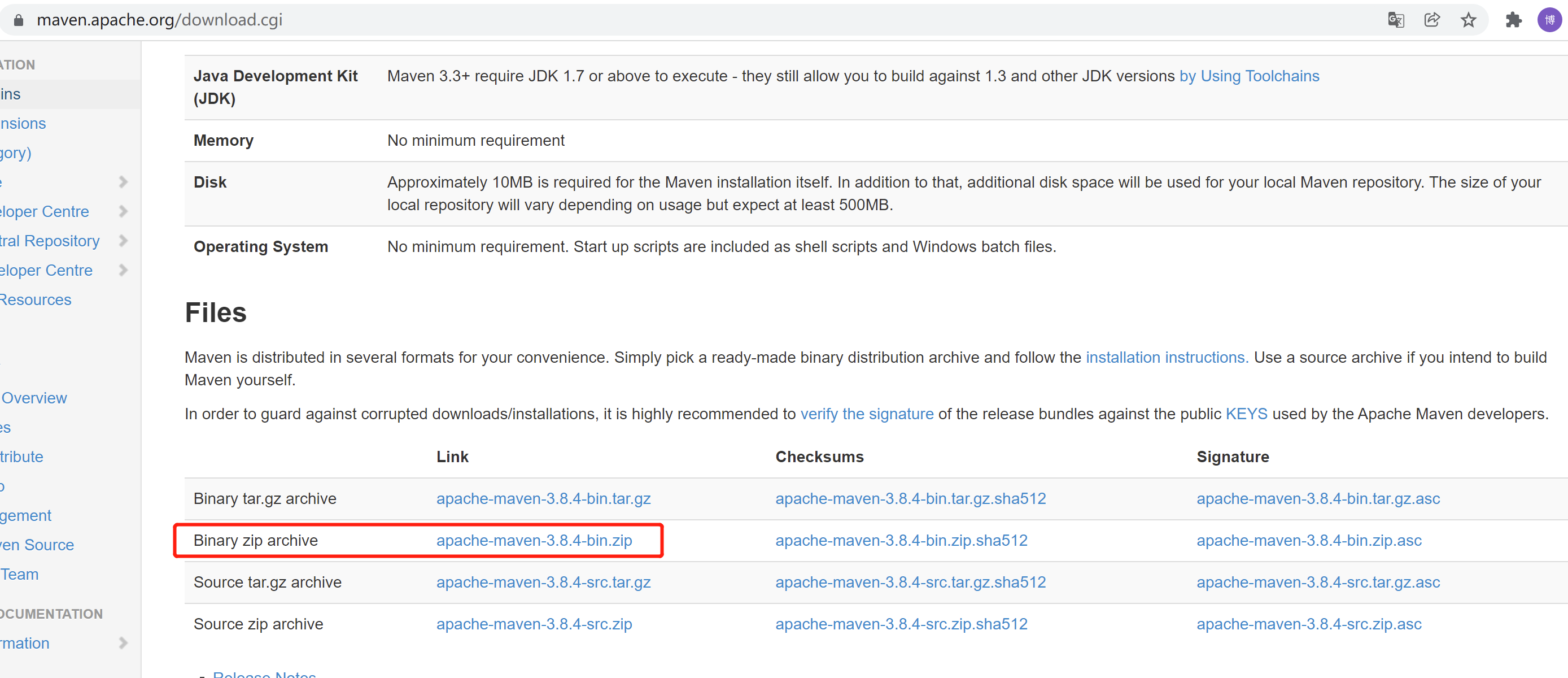
项目构建工具有很多： make、ant、maven、gradle

# Maven的安装和配置

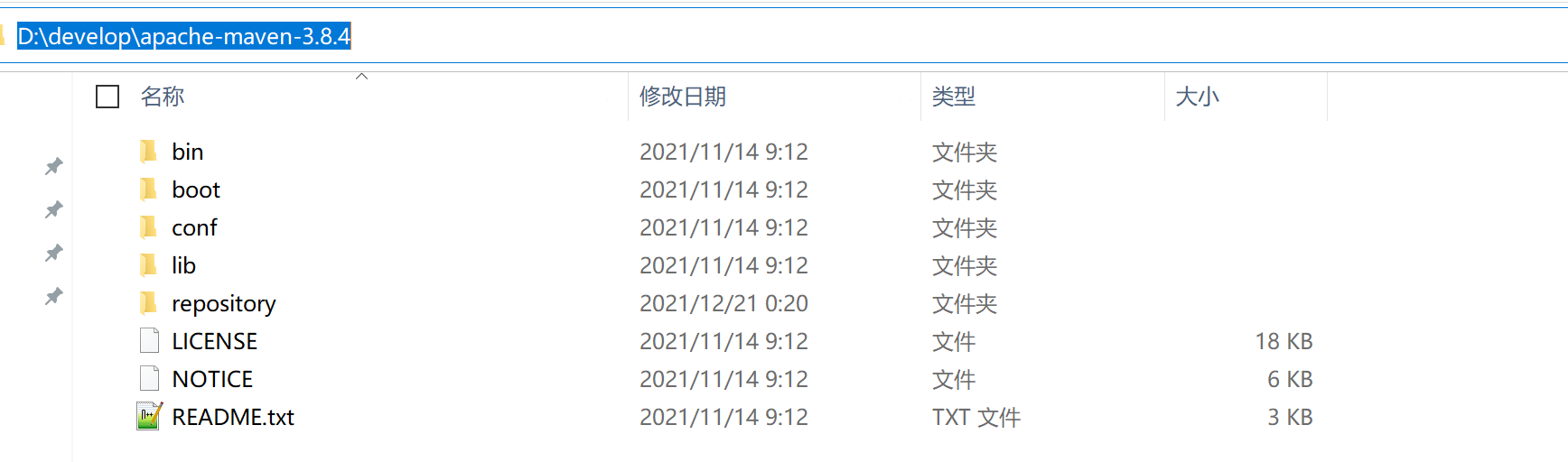
## 安装

**Maven的运行依赖jdk，所以在安装maven前，需要先安装jdk，这里不再赘述jdk安装方法。**

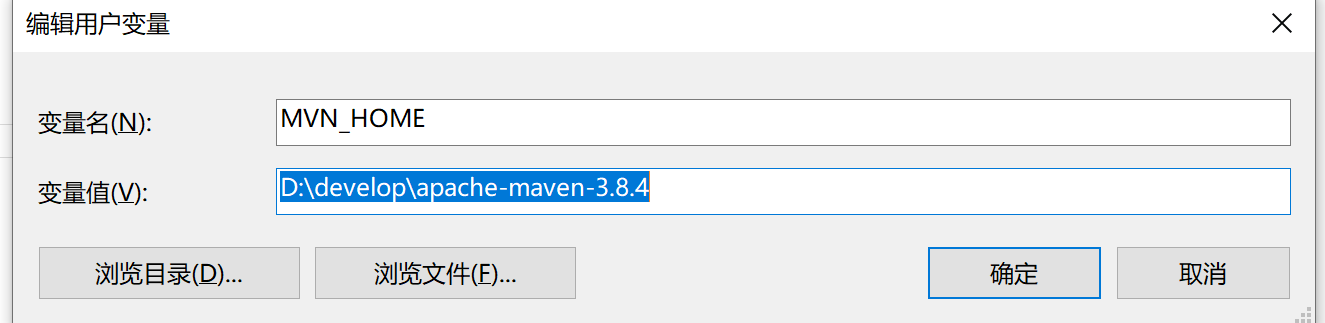
去 <https://maven.apache.org/> 找最新的安装包，下载 binary 版本即可，如果想要自己编译，可以下载 src 版本。



下载完成后，最好找一个简单的目录路径(全英文无空格)进行解压：



解压好后，将maven目录信息配置到环境变量中去，我这里是Windows环境，在用户环境中添加了个环境变量 MVN\_HOME

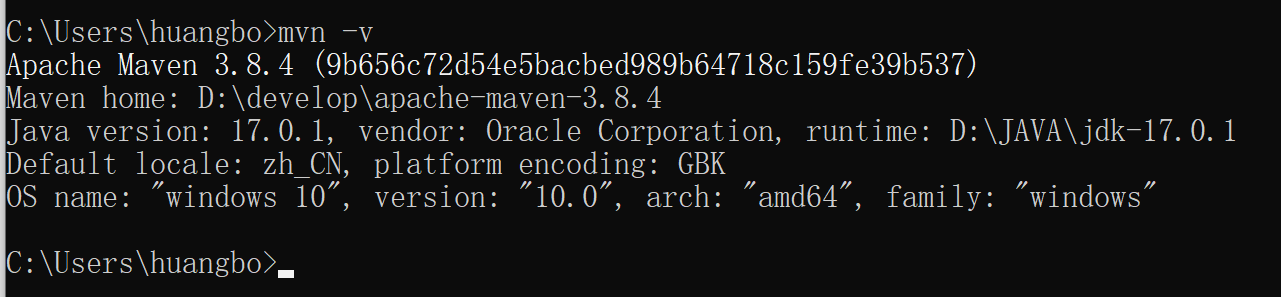


并在path中加入 %MVN\_HOME%\bin

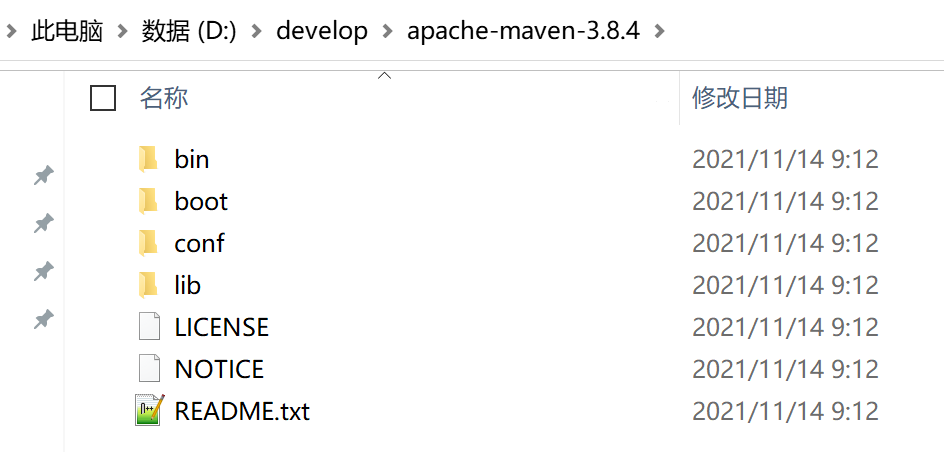
**需要升级maven版本时，只需要将 MVN\_HOME 的值修改为新版本所在目录即可**

若是**Unix/Linux 类型操作系统**，**最好对 maven\_home 创建软连接**， 这样在升级的时候只需要修改软连接的指向即可，都无需修改环境变量

配置好环境变量后可以运行 mvn 命令来检查是否成功(需要新开终端窗口)：



## Maven目录结构解析



* bin目录

主要包含maven的运行脚本文件和一个 classworlds的配置文件m2.conf

* boot目录

该目录中有一个 plexus-classworlds-2.6.0.jar 文件，这是一个类加载器框架，maven使用该框架来加载自己的类库。

* conf目录

这个目录中有个非常重要的文件 settings.xml ，这个配置文件决定了maven的全局行为。

* lib目录

主要是一些maven运行时所需的java类库

## 配置 settings.xml

可以通过修改 %MVN\_HOME%\conf 下的 settings.xml 文件来配置maven的全局行为，或者修改 ~/.m2/settings.xml 文件来配置用户范围内的maven行为

**注意：**修改%MVN\_HOME%\conf\settings.xml文件在maven升级时，要将该配置文件复制到新版本中去，因此最好是修改用户配置文件，这样升级的时候就可以不用管配置文件了

### Maven仓库

本地仓库：用来存储从远程仓库下载下来的公用构件

远程仓库：中央仓库、第三方公共仓库等

在 settings.xml 文件中，通过 <localRepository> 标签来配置本地仓库目录

|  |
| --- |
| <localRepository>D:\\develop\\repository</localRepository> |

Maven需要构件时，先在本地仓库中找，若找不到，就会从远程仓库将需要的构件下载至本地仓库。

下面是中央仓库配置，该配置既可配置在 settings.xml 中，也可以配置在 pom 文件中

|  |
| --- |
| <repositories>  <repository>  <id>central</id>  <name>Central Repository</name>  <url>https://repo.maven.apache.org/maven2</url>  <layout>default</layout>  <snapshots>  <!-- 表示不会从中央仓库下载快照版本的构件 -->  <enabled>false</enabled>  </snapshots>  </repository>  </repositories> |

### 代理服务配置

有些时候，本地机器无法直接访问公共仓库，需要配置代理服务器信息

假设有一IP地址为 218.14.227.197，端口号为3128的代理服务，在 settings.xml文件中可以添加如下配置信息：

|  |
| --- |
| <proxies>  <proxy>  <id>proxy-01</id>  <active>true</active>  <protocol>http</protocol>  <username>proxyuser</username>  <password>proxypass</password>  <host>218.14.227.197</host>  <port>3128</port>  <!-- nonProxyHosts 用来指定哪些主机名不需要代理服务，多个主机名用 | 分隔，支持通配符，如 \*.google.com 表示所有以 .google.com结尾的域名访问都不需要代理 -->  <nonProxyHosts>local.net|\*.google.com </nonProxyHosts>  </proxy>  </proxies> |

### 镜像配置

### 用户认证配置

# POM文件

Project Object Model ， POM

像make的 Makefile ，Ant的build.xml 文件一样， maven项目的核心是 pom.xml