# MAVEN

## Apache Maven là gì?

* **Apache Maven** là một **tool/công cụ/chương trình** giúp quản lí mọi tác vụ xoay quanh việc viết code trong các dự án Java, bao gồm: *tạo cấu trúc thư mục dự án, quản lí các thư viện đang sử dụng (được gọi là dependencies), biên dịch, kiểm thử, đóng gói sản phẩm*…
* Maven cũng là 1 style hay cách thức tổ chức code dùng trong các dự án Java.
* Mọi tác vụ cần làm cho dự án (ví dụ dùng thư viện nào, có kiểm thử khi đóng gói dự án hay không, đóng gói thế nào) đều được mô tả trong một file cấu hình gọi là **POM.XML**.
* Một dự án Maven chuẩn sẽ có các thông số cơ bản sau đây:
* **GroupId** – tên tổ chức/công ty, nơi sẽ tạo ra dự án/phần mềm
* **ArtifactId** – tên dự án, cũng chính là thư mục sẽ chứa code
* **Version** – đánh số phiên bản dự án, cũng là phiên bản phần mềm khi được phát hành

VD:

<!-- https://mvnrepository.com/artifact/com.microsoft.sqlserver/mssql-jdbc -->

<dependency>

<groupId>com.microsoft.sqlserver</groupId>

<artifactId>mssql-jdbc</artifactId>

<version>12.10.1.jre11</version>

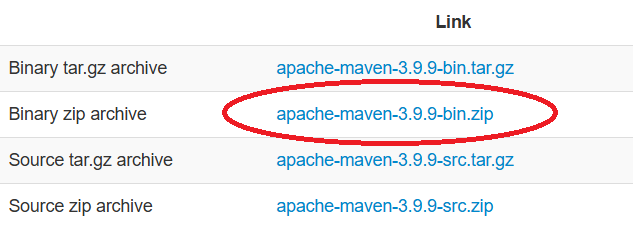
<scope>compile</scope>

</dependency>

* Maven sử dụng website ***https://mvnrepository.com/*** như là cái kho tổng để chứa toàn bộ các thư viện .jar cần sử dụng cho mọi dự án Java. Để sử dụng một thư viện bất kì, cần khai báo tên thư viện trong file POM.XML, và Maven sẽ tự động download thư viện từ kho tổng về máy (máy local) và add vào dự án. Thư viện trong Maven được gọi tên là **dependency**.
* Có 2 cách sử dụng Maven:
* Download Maven về cài riêng lẻ trên máy và gõ lệnh mvn kèm thêm các tham số
* Maven được tích hợp sẵn trong các IDE như Eclipse, IntelliJ, NetBeans, chỉ cần chọn loại dự án có chữ Maven

## Download Maven

Download Maven: ***https://maven.apache.org/download.cgi*** | chọn **Binary zip archive**

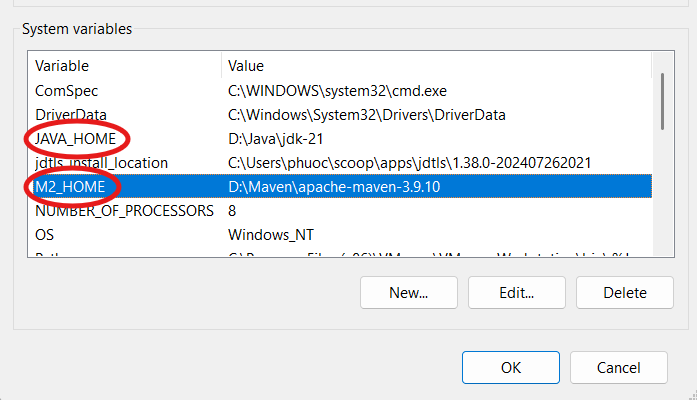


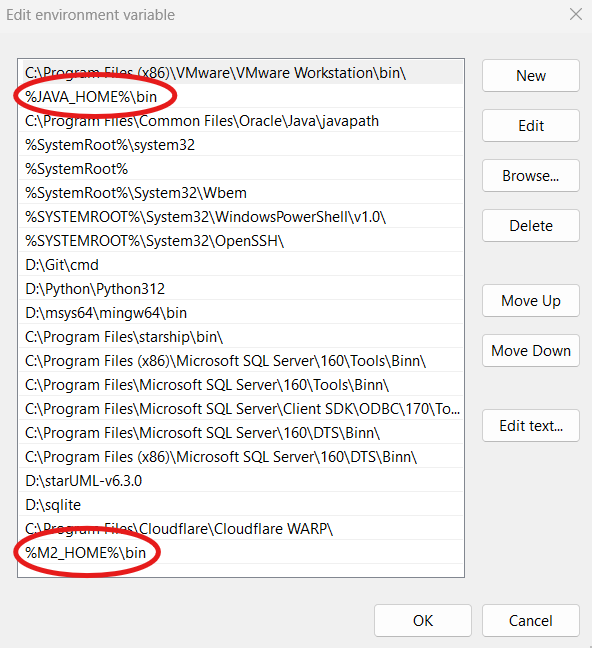
Trong ví dụ này chọn ***<https://dlcdn.apache.org/maven/maven-3/3.9.9/binaries/apachemaven-3.9.9-bin.zip>***

## Installing Apache Maven

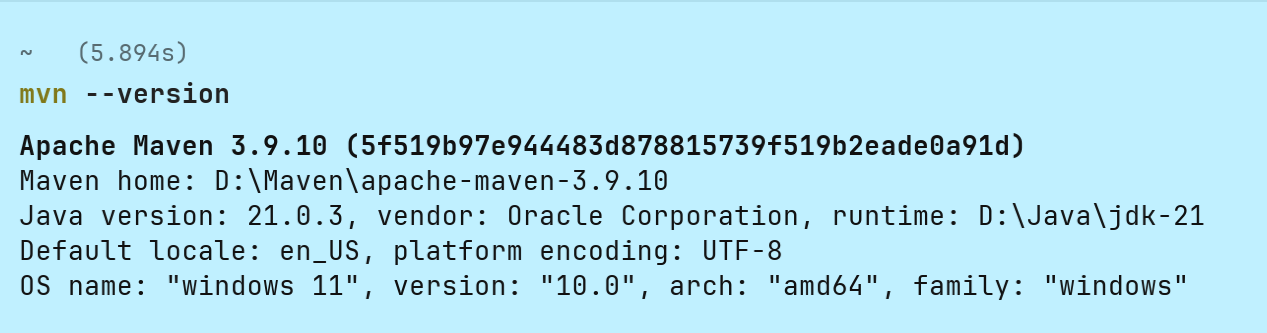
* Xả nén file .zip vừa tải về vào một thư mục trên ổ đĩa C, ví dụ C:\Program Files\apache-maven-3.9.9
* Thiết lập biến môi trường

|  |  |
| --- | --- |
| JAVA\_HOME | trỏ tới thư mục cài đặt JDK |
| M2\_HOME | trỏ tới thư mục cài đặt Maven |
| M2 | trỏ tới thư mục M2\_HOME\bin |
| PATH | bổ sung thêm đường dẫn trong biến M2 vào PATH |

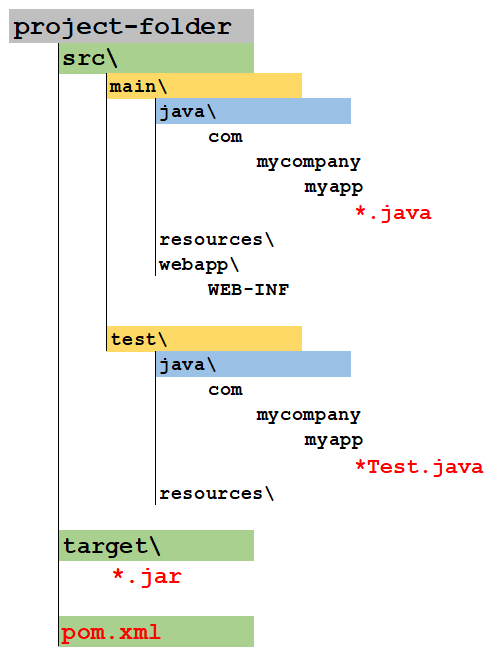




* Kiểm tra xem cài đặt Maven thành công hay chưa



## Standard Maven Directory Structure



## Creating a Project with Archetype

* Maven Archetype là những loại cấu trúc/dàn khung dự án Java, hay còn gọi là project template được cộng đồng Maven tạo sẵn cho các Java developer.
* Mỗi loại dự án Java sẽ có cấu trúc thư mục khác nhau (bên trong chứa code và các tài nguyên). Ví dụ dự án Java Web truyền thống sẽ khác dự án SpringBoot về cấu trúc thư mục.
* Maven cũng định nghĩa trước và sẵn có rất nhiều loại template/archetype dự án chúng ta sẽ triển khai và ta chỉ việc chọn loại dự án cần làm, Maven sẽ lo giúp phần tổ chức code cho ta.
* Nhờ Maven tạo dự án từ dự án mẫu có sẵn, gọi là generate dự án từ Archetype.
* Có sẵn hơn 2000??? loại dự án Java khác nhau để ta tái sử dụng (generate). Kho chứa chuẩn của các Archetype Maven: ***<https://maven.apache.org/archetypes/index.html>***
* Để tạo dự án, biên dịch, đóng gói, ta dùng câu lệnh **mvn** kèm thêm các tham số cần thiết.
* Để sử dụng/generate một template/archetype có sẵn trên mạng, ta phải:
* Chỉ ra các thông số **archetypeGroupId, archetypeArtifactId, archetypeVersion** của template cần sử dụng
* Chỉ ra **GroupId, ArtifactId, Version** của dự án riêng của mình
* Lệnh liệt kê danh sách các dàn khung dự án/template dự án có sẵn (hơn 2000 template).

**mvn archetype:generate**

**mvn archetype:generate -Dfilter=org.apache:web**

* Lệnh in danh sách các dàn khung dự án có sẵn ra file text.

**mvn archetype:generate > archetypes.txt**

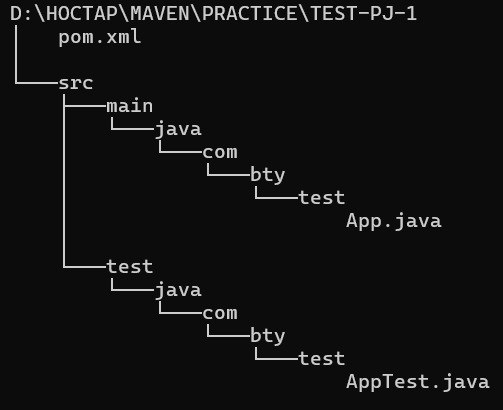
* Lệnh tạo dự án Java thuần OOP:

**mvn archetype:generate**

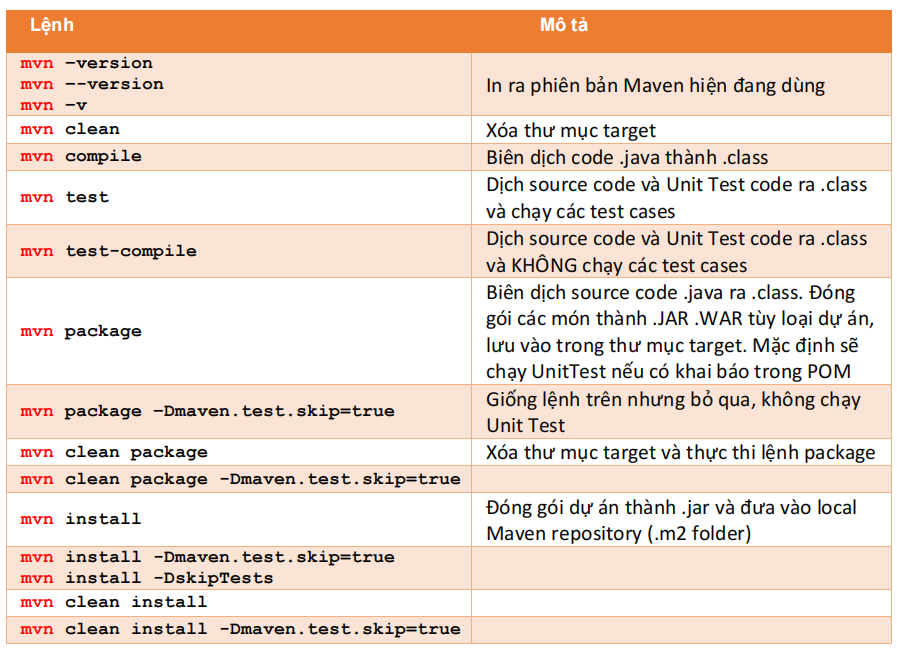
**-DarchetypeGroupId=org.apache.maven.archetypes**

**-DarchetypeArtifactId=maven-archetype-quickstart**

**-DarchetypeVersion=1.4**



## Common Maven Commands



## Application Execution

* Gõ lệnh sau để chạy app (JAR file)

Lưu ý: Phải đứng ở thư mục chứa **project**, tức là đứng ngoài thư mục **target\**

**java -jar target/tên-file-jar-của-bạn.jar**

* Nếu app không chạy, báo lỗi không tìm thấy hàm main(), lí do là file .jar này chưa được thiết

lập là file thực thi – execution, khi đó bắt buộc phải chỉ ra tên class chứa hàm main() trong

project của bạn khi gõ lệnh

**java -cp target/tên-file-jar-của-bạn.jar com.bty.healthy.Main**

* Hoặc để biến file JAR thành file thực thi, cần cấu hình thêm trong file **POM.XML** trước khi

đóng gói app.

**<plugin>**

**<!-- Build an executable JAR -->**

**<groupId>org.apache.maven.plugins</groupId>**

**<artifactId>maven-jar-plugin</artifactId>**

**<version>3.4.2</version>**

**<configuration>**

**<archive>**

**<manifest>**

**<mainClass>com.bty.heathy.Main**

**tên-package-và-tên-class-chứa-hàm-main**

**</mainClass>**

**</manifest>**

**</archive>**

**</configuration>**

**</plugin>**

Khi thực thi chỉ cần gọi tên file jar: **java -jar target/tên-file-jar-của-bạn.jar**

* Nếu file JAR chính lại cần hay phụ thuộc vào các file .jar khác, cần cấu hình thêm trong file

POM.XML để gom tất cả các dependency vào chung 1 file .jar cuối cùng (fat jar) trước khi

đóng gói app, lúc này file .jar sẽ có kích thước to hơn nhiều do phải chứa cả các dependency

khác bên trong.

**<build>**

**<plugins>**

**<!-- Tắt JAR mặc định -->**

**<plugin>**

**<artifactId>maven-jar-plugin</artifactId>**

**<version>3.4.2</version>**

**<configuration>**

**<skip>true</skip>**

**</configuration>**

**</plugin>**

**<!-- Assembly plugin -->**

**<plugin>**

**<groupId>org.apache.maven.plugins</groupId>**

**<artifactId>maven-assembly-plugin</artifactId>**

**<version>3.7.1</version>**

**<configuration>**

**<descriptorRefs>**

**<descriptorRef>jar-with-dependencies</descriptorRef>**

**</descriptorRefs>**

**<!-- Bỏ suffix "-jar-with-dependencies" -->**

**<appendAssemblyId>false</appendAssemblyId>**

**<!-- Tên file đầu ra -->**

**<finalName>${project.artifactId}-${project.version}</finalName>**

**<!-- Thêm Manifest với Main-Class -->**

**<archive>**

**<manifest>**

**<mainClass>com.bty.heathy.Main</mainClass>**

**</manifest>**

**</archive>**

**</configuration>**

**<executions>**

**<execution>**

**<id>make-assembly</id>**

**<phase>package</phase>**

**<goals>**

**<goal>single</goal>**

**</goals>**

**</execution>**

**</executions>**

**</plugin>**

**</plugins>**

**</build>**

Khi thực thi chỉ cần gọi tên file jar: **java -jar target/tên-file-jar-của-bạn.jar**