Maven in 5 Minutes

What is Apache Maven?

- 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 của bạn, nơi sẽ tạo ra dự án/phần mềm
  - **`ArtifactId`** – tên dự án của bạn, 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
- 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 bạn (máy local) và add vào dự án của bạn. 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, bạn chỉ cần chọn loại dự án có chữ Maven

Download Maven

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

	Link
Binary tar.gz archive	apache-maven-3.8.4-bin.tar.gz
Binary zip archive	apache-maven-3.8.4-bin.zip
Source tar.gz archive	apache-maven-3.8.4-src.tar.gz

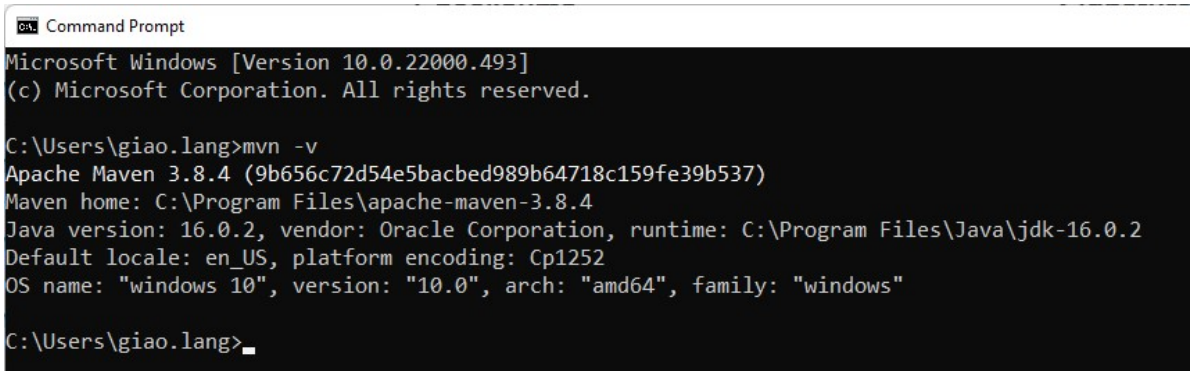
- Trong ví dụ này chọn **https://dlcdn.apache.org/maven/maven-3/3.8.4/binaries/apache-maven-3.8.4-bin.zip**

Installing Apache Maven

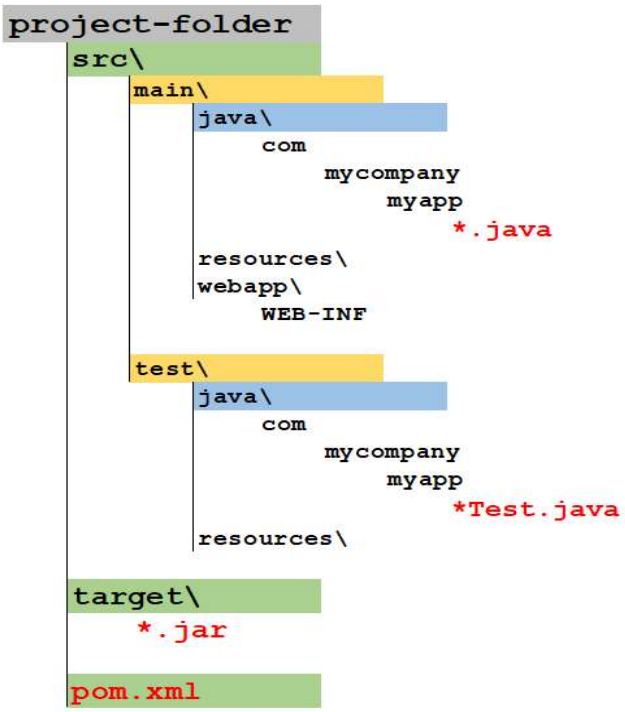
- Xả nén file .zip vừa tải về vào một thư mục trên ổ đĩa C, ví dụ C:\Program Files\apache-maven-3.8.4
- Thiết lập biến môi trường

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 bạn sẽ triển khai và bạn chỉ việc chọn loại dự án cần làm, Maven sẽ lo giúp phần tổ chức code cho bạn.
- 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.
- Một dự án Maven chuẩn sẽ có các thông số cơ bản sau đây:
  - **GroupId** – tên tổ chức/công ty của bạn
  - **ArtifactId** – tên dự án của bạn
  - **Version** – phiên bản dự án
- Để sử dụng/generate một template/archetype có sẵn trên mạng, bạn phải:
  - Chỉ ra các thông số **archetypeGroupId**, **archetypeArtifactId**, **archetypeVersion** của template bạn sử dụng
  - Chỉ ra **GroupId**, **ArtifactId**, **Version** của dự án riêng của bạn
- 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
```

- Tùy tham số đưa vào lệnh **mvn archetype:generate** mà bạn sẽ bị hỏi thêm các thông tin groupId, artifactId, package name.
- **Lưu ý các lệnh rất dài và có nhiều tham số và PHẢI gõ trên 1 dòng.**
- Gõ các lệnh dưới đây để tạo mới dự án Java.

  Lưu ý: trước khi gõ lệnh, phải chuyển đến một thư mục dùng để chứa project mới sẽ được tạo ra bởi Maven.

Lệnh tạo dự án Java thuần OOP:

```
mvn archetype:generate
-DarchetypeGroupId=org.apache.maven.archetypes
-DarchetypeArtifactId=maven-archetype-quickstart
-DarchetypeVersion=1.4
```

Bạn sẽ bị hỏi tên thương hiệu của bạn – groupId, tên dự án – artifactId cũng là tên thư mục của dự án, phiên bản dự án, tên package dự án

Lệnh tạo dự án Java thuần OOP:

```
mvn archetype:generate
-DgroupId=com.giaolang
-DartifactId=math-util
-DarchetypeArtifactId=maven-archetype-quickstart
-DarchetypeVersion=1.4
-DinteractiveMode=false
```

Bạn sẽ bị hỏi ít hơn. Chỉ bị hỏi version, tên package vì đã khai báo sẵn một vài thông tin của dự án rồi (groupId, artifactId)

Lệnh tạo dự án bất kì dựa trên template nào đó:

```
mvn archetype:generate
-DgroupId=tên-thương-hiệu-của-bạn
-DartifactId=tên-dự-án-của-bạn
-DarchetypeArtifactId=tên-template-có-sẵn-của-Maven
-DarchetypeVersion=phiên-bản-template-có-sẵn
-DinteractiveMode=false-làm-luôn-không-hỏi-từng-bước
```

Danh sách các template có sẵn khoảng 2000 template đã đề cập ở trên

Common Maven Commands

Lệnh	Mô tả

`mvn -version`
`mvn --version`
`mvn -v`
	In ra phiên bản Maven hiện đang dùng

`mvn clean`	Xóa thư mục target
`mvn compile`	Biên dịch code .java thành .class
`mvn test`	Dịch source code và Unit Test code ra .class và chạy các test cases
`mvn test-compile`	Dịch source code và Unit Test code ra .class và KHÔNG chạy các test cases

`mvn package`	Biên dịch source code .java ra .class. Đóng gói các món thành .JAR .WAR tùy loại dự án, lưu vào trong thư mục target. Mặc định sẽ chạy UnitTest nếu có khai báo trong POM
`mvn package -Dmaven.test.skip=true`	Giống lệnh trên nhưng bỏ qua, không chạy Unit Test
`mvn clean package`	Xóa thư mục target và thực thi lệnh package
`mvn clean package -Dmaven.test.skip=true`	
`mvn install`	Đóng gói dự án thành .jar và đưa vào local Maven repository (.m2 folder)
`mvn install -Dmaven.test.skip=true` `mvn install -DskipTests`	
`mvn clean install`	
`mvn clean install -Dmaven.test.skip=true`	

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/my-app-1.0-SNAPSHOT.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() khi gõ lệnh.

  ```
  java -cp target/my-app-1.0-SNAPSHOT.jar com.mycompany.myapp.AppMain
  ```

- 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>3.1.0</version>
   <configuration>
      <archive>
         <manifest>
            <mainClass>com.mycompany.myapp.AppMain
                       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/my-app-1.0-SNAPSHOT.jar`**

References

- https://maven.apache.org/guides/getting-started/index.html