

# Maven in 5 Minutes

© 2021 GIÁO.LÀNG | ver. 21.1011

## Download Maven

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

Link	
Binary tar.gz archive	<a href="#">apache-maven-3.8.3-bin.tar.gz</a>
Binary zip archive	<a href="#">apache-maven-3.8.3-bin.zip</a>
Source tar.gz archive	<a href="#">apache-maven-3.8.3-src.tar.gz</a>
Source zip archive	<a href="#">apache-maven-3.8.3-src.zip</a>

- Trong ví dụ này chọn <https://dlcdn.apache.org/maven/maven-3/3.8.3/binaries/apache-maven-3.8.3-bin.zip>

## Installing Apache Maven

- Xả nén file .zip vừa tải về một thư mục trên ổ đĩa C, ví dụ C:\Program Files\apache-maven-3.8.3
- Thiết lập biến môi trường

<b>JAVA_HOME</b>	trỏ tới thư mục cài đặt JDK
<b>M2_HOME</b>	trỏ tới thư mục cài đặt Maven
<b>M2</b>	trỏ tới thư mục M2_HOME\bin
<b>PATH</b>	bổ sung thêm đường dẫn trong biến M2 vào PATH

- Kiểm tra xem cài đặt Maven thành công hay chưa?

```
C:\Windows\System32\cmd.exe

C:\Users\giao.lang\Desktop\_code.truong.lang>mvn -v
Apache Maven 3.8.3 (ff8e977a158738155dc465c6a97ffaf31982d739)
Maven home: C:\Program Files\apache-maven-3.8.3
Java version: 16.0.2, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-16.0.2
Default locale: en_US, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Users\giao.lang\Desktop\_code.truong.lang>
```

## Standard Maven Directory Structurer

```
project-dir
|-- src\
|-- main\
|   |-- java\
|   |   |-- com
|   |   |   |-- mycompany
|   |   |   |   |-- myapp
|   |   |   |       |-- *.java
|   |-- resources\
|
|   |-- webapp\
|   |   |-- WEB-INF
|
|-- test\
|   |-- java\
|   |   |-- com
|   |   |   |-- mycompany
|   |   |   |   |-- myapp
|   |   |   |       |-- *Test.java
|   |-- resources\
|
|-- target\
|-- pom.xml
```

## Creating a Project with Archetype

- Maven Archetype là những loại cấu trúc 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 chứa code và các tài nguyên khác nhau. Ví dụ dự án Java Web truyền thống sẽ khác dự án SpringBoot về cấu trúc thư mục. Maven cũng định nghĩa sẵn rất nhiều loại dự án bạn sẽ triển khai và bạn chỉ việc chọn loại dự án cần làm, Maven sẽ giúp phần còn lại cho bạn.
- 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>
- Liệt kê danh sách các dàn khung dự án có sẵn

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

- Liệt kê 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 sau đây để tạo mới dự án Java. Lưu ý: chuyển đến một thư mục dùng để chứa chứa project mới sẽ được tạo ra bởi Maven trước khi gõ lệnh.
- 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

```
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
```

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

Bạn chỉ bị hỏi xác nhận vài điều: version, tên package vì đã khai báo sẵn một vài thông tin của dự án rồi

## Common Maven Commands

Lệnh	Mô tả
<code>mvn -version</code> <code>mvn -version</code> <code>mvn -v</code>	In ra phiên bản Maven hiện đang dùng
<code>mvn clean</code>	Xóa thư mục target
<code>mvn compile</code>	
<code>mvn test</code>	Dịch source code và Unit Test code ra .class và chạy các test cases
<code>mvn test-compile</code>	Dịch source code và Unit Test code ra .class và KHÔNG chạy các test cases
<code>mvn package</code>	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

<b>mvn package -Dmaven.test.skip=true</b>	Giống lệnh trên nhưng bỏ qua, không chạy Unit Test
<b>mvn clean package</b>	Xóa thư mục target và thực thi lệnh package
<b>mvn clean package -Dmaven.test.skip=true</b>	
<b>mvn install</b>	Đóng gói dự án thành .jar và đưa vào local Maven repository (.m2 folder)
<b>mvn install -Dmaven.test.skip=true</b> <b>mvn install -DskipTests</b>	
<b>mvn clean install</b>	
<b>mvn clean install -Dmaven.test.skip=true</b>	

## Application Execution

- Gõ lệnh sau để chạy app (khi đang đứng ở thư mục chứa project). Bắt buộc phải chỉ ra tên class chứa hàm main() khi file JAR không được thiết lập là file execution

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

- Biến file JAR thành file thực thi, cần cấu hình thêm trong file POM 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</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>