

并发编程网 - ifeve.com

让天下没有难学的技术

[HOME](#)[JAVA](#)[《maven官方文档》5分钟开始Maven](#)

SEARCH



APR 2016

7,913 人阅读

司余

JAVA

★★★★★ (2 votes, average: 5.00 out of 5)

Write comment

《maven官方文档》5分钟开始Maven

[原文地址](#)

前提

你必须明白如何在电脑上安装软件。如果你不知道如何做，请向你学校、办公室里等的人请教下，或者付费给他人让他们解释给你。Maven邮件组不是寻求这个建议的最好地方。

安装

Maven是个Java工具，所以你必须安装了Java才能继续。 首先，[下载Maven](#)，然后按照[安装指南](#)。完成后，在终端或命令行中键入如下指令：

```
mvn --version
```

应该会打印已安装Maven的版本。例如：

```
Apache Maven 3.0.5 (r01de14724cdef164cd33c7c8c2fe155faf9602da; 2013-02-19 14:51:28+0100)

Maven home: D:\apache-maven-3.0.5\bin\..

Java version: 1.6.0_25, vendor: Sun Microsystems Inc.

Java home: C:\Program Files\Java\jdk1.6.0_25\jre

Default locale: nl_NL, platform encoding: Cp1252

OS name: "windows 7", version: "6.1", arch: "amd64", family: "windows"
```



RECENT POSTS

[《RabbitMQ官方指南》主题](#)

[《Nginx官方文档》关于nginxScript](#)

[《RabbitMQ官方指南》路由](#)

[《RabbitMQ官方指南》RPC](#)

[《Nginx官方文档》用DTrace pid提供程序调试nginx](#)

[《Nginx官方文档》转换重写规则](#)

[《Nginx官方文档》WebSocket代理](#)

基于你的网络设置，你需要额外的配置。如果有必要的话看看[Maven配置指南](#).如果你在使用Windows，你应该查看[Windows前提](#)以确保你已准备好在Windows上使用Maven。

创建一个项目

你需要一个给你项目保存的地方。在某处创建一个目录，然后在这个目录里开始shell。在你的命令行上，执行如下Maven语言：

```
mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app -DarchetypeArtifactId=maven-archetype-quickstart -
```

如果你已安装Maven,第一次执行这个需要一定时间。这是因为Maven在下载最新的物件（jar插件和其他文件）到你的本地仓库。在成功前，也需要你执行几次命令。这是因为在下载完成前，远程服务器可能超时。别担心，有几种解决这个的方法。

你讲看见生成语言创建了一个和物件ID相同名字的目录。变更到这个目录：

```
cd my-app
```

使用这个目录，你讲看到如下[标准目录结构](#)：

```
my-app
|-- pom.xml
`-- src
    |-- main
    |   |-- java
    |       |-- com
    |           |-- mycompany
    |               |-- app
    |                   |-- App.java
    |-- test
    |   |-- java
    |       |-- com
    |           |-- mycompany
    |               |-- app
    |                   |-- AppTest.java
```

src/main/java目录包含项目源代码，src/test/java目录包含检查源代码，pom.xml是项目的项目对象模型，或POM。

POM

[《RabbitMQ官方指南》翻译邀请](#)

[JAVA8 stream 中Spliterator的使用\(二\)](#)

[JAVA8 stream 中Spliterator的使用\(一\)](#)

[Spring Boot 整合 Thymeleaf 完整 Web 案例](#)

[实战Spring事务传播性与隔离性](#)

[nginx如何处理TCP / UDP会话](#)

[《Nginx官方文档》配置HTTPS服务器](#)

[《TensorFlow官方文档》快速入门](#)

[《Nginx官方文档》Nginx之Server names](#)

[Spring事务配置解惑](#)

[《Spring 5 官方文档》39. 创建可扩展的XML](#)

[实战解析—论三年内快速成长为一名技术专家](#)

[软件架构模式-第二章事件驱动架构\(上\)](#)

[《阿里感悟》如何在三年内成长为一名技术专家](#)

[软件架构模式-第一章分层架构（下）](#)

[《Maven官方文档》配置默认Mojo扩展](#)

[《Hibernate快速开始 – 4 – 使用JAVA持久层 API \(JPA\) 教程》](#)

[《Hibernate快速开始》翻译邀请](#)

[Java面试题-基础知识](#)

[JDK8中新增原子性操作类LongAdder](#)

[《Maven官方文档》Maven 文档风格指南](#)

[《Maven官方文档》HttpClient HTTP提供者的先进配置](#)

[应届生JAVA技术面试题](#)

在Maven里，pom.xml文件是项目配置的核心。它是一个独立的配置文件，包含了你想要的方式下建立项目所必须的主要信息。POM是巨大的，其复杂度使人苦恼。但你只需要有效地使用它，没有必要明白所有的错综复杂。项目的POM是：

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>0.0</modelVersion>

  <groupId>mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <version>0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>Maven Quick Start Archetype</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>8.2</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

只需要我做的是么？

你执行了Maven语言原型：generate,向这个语言传了许多参数。前缀原型是包含这个语言的插件。如果你熟悉Ant，你可以设想这个是相同的任务。基于一个原型，这个语言创建了一个简单的项目。现在可以说，插件是个有通用目的的语言的集合。例如jboss-maven-plugin，目的是“处理大量jboss条目”。

建设项目

```
mvn package
```

命令行会打印出各种指令，并具有以下结尾：

热门文章

- [Google Guava官方教程（中文版）](#) 516,108 人阅读
- [Java NIO系列教程（一）Java NIO 概述](#) 333,121 人阅读
- [Java并发性和多线程介绍目录](#) 227,351 人阅读
- [Java NIO 系列教程](#) 221,417 人阅读
- [Java NIO系列教程（十二）Java NIO与IO](#) 189,932 人阅读
- [Java NIO系列教程（六）Selector](#) 170,555 人阅读
- [Java NIO系列教程（二）Channel](#) 165,322 人阅读
- [Java NIO系列教程（三）Buffer](#) 164,634 人阅读
- [《Storm入门》中文版](#) 163,136 人阅读
- [Java8初体验（二）Stream语法详解](#) 162,204 人阅读
- [69道Spring面试题和答案](#) 145,938 人阅读
- [Netty 5用户指南](#) 135,326 人阅读
- [Java 7 并发编程指南中文版](#) 126,965 人阅读
- [并发框架Disruptor译文](#) 121,530 人阅读
- [Java NIO系列教程（八）SocketChannel](#) 110,326 人阅读
- [\[Google Guava\] 2.3-强大的集合工具类：ja...](#) 100,347 人阅读
- [\[Google Guava\] 3-缓存](#) 99,885 人阅读
- [\[Google Guava\] 1.1-使用和避免null](#) 99,306 人阅读
- [Java NIO系列教程（七）FileChannel](#) 98,284 人阅读

```
...
[INFO] -----
[INFO] BUILD SUCCESSFUL
[INFO] -----[INFO] Total time: 2 seconds
[INFO] Finished at: Thu Jul 07 21:34:52 CEST 2011
[INFO] Final Memory: 3M/6M
[INFO] -----
```

不像第一个执行的命令（ *archetype:generate* ），你可能注意到第二个命令是简单的一个词-package。这是个阶段实行而不是语言。实行阶段是[建设生命周期](https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html)里的而一个步骤,是一系列有序的阶段。当给出一个实行阶段，Maven将执行序列里的每一个阶段，并且包括定义的阶段。例如，如果我们执行 *编译* 阶段，实际上得到执行的阶段是：

1. 验证
2. 编译代码
3. 处理代码
4. 生成资源文件
5. 处理资源文件
6. 编译

您可以使用以下命令测试新编译和打包的JAR：

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

将打印经典语句：

```
Hello World!
```

运行Maven工具

Maven阶段

尽管几乎没有一个全面的清单，这些都是执行最常用的默认生命周期阶段。

验证：验证项目是正确的，所有必要的信息

编译：编译项目的源代码

测试：使用合适的单元测试框架测试编译的源代码。 这些测试应该不需要代码被打包或部署

TAGS

actor Basic book classes collections
concurrency Concurrent concurrent
data structure Customizing Executor
Executor framework False Sharing faq fork
Fork/Join fork join Framework Functional
Programming Guava IO **JAVA** java8
jmm join JVM lock Memory Barriers Netty
NIO OAuth 2.0 pattern-matching RingBuffer Scala
slf4j spark spark官方文档 stm **Storm**
synchronization Synchronized **thread**
tomcat volatile 多线程 **并发译文，Java**
，Maven

- 包**：把编译后的代码，并在其分发格式打包，例如一个JAR。
- 集成测试**：如果需要，处理和部署包到集成测试可以运行的环境
- 验证**：运行任何检查，以验证包是有效的，同时符合质量标准
- 安装**：包本地安装到本地存储库中，用作本地其他项目的依赖
- 部署**：在集成或发布环境中进行，复制最终的包到远程仓库与其他开发者和项目共享。

还有值得注意的另外两个Maven的生命周期超出上面的默认列表。 他们是

- 清洁**：清理之前建立创建的工件
- 网站**：生成此项目的站点文档，

实行阶段实际上映射到底层的语言。 每阶段中执行的具体语言是取决于项目的包类型。 例如， *package*执行的*jar : jar*，如果项目类型为JAR和*war : war*，如果项目类型是-你猜对了-WAR。

需要注意的一个有趣的事情是，阶段和语言，可以按顺序执行。

```
mvn clean dependency:copy-dependencies package
```

此命令将清理项目，复制依赖关系，并打包项目（当然执行各个阶段到包）。

生成站点

```
mvn site
```

这个阶段产生基于对项目的POM信息的站点。你可以看一下在target/site生成的文档。

结论

我们希望这一快速浏览已经激起了你的Maven中的通用性兴趣。请注意，这是一个非常截断快速入门指南。现在，你已经准备好关于你刚才执行的动作更全面的信息。 看看[Maven的入门指南](#)。

原创文章，转载请注明： 转载自[并发编程网 – ifeve.com](#) **本文链接地址：** [《maven官方文档》5分钟开始Maven](#)



司余

积少成多, 聚沙成塔

ARCHIVES

[October 2017](#) (16)

[September 2017](#) (10)

[August 2017](#) (25)

[July 2017](#) (20)

[June 2017](#) (38)

[May 2017](#) (59)

[April 2017](#) (17)

[March 2017](#) (30)

[February 2017](#) (9)

[January 2017](#) (7)

[December 2016](#) (12)

[November 2016](#) (27)

[October 2016](#) (16)

[September 2016](#) (11)

[August 2016](#) (6)

[July 2016](#) (9)

[June 2016](#) (7)

[May 2016](#) (20)

[April 2016](#) (28)

[March 2016](#) (8)

[February 2016](#) (7)

友情链接

[coolshell](#)

[Programer. 大猫](#)

[一粟的博客](#)

[志俊的博客](#)

[最代码](#)

[点点折](#)

[领悟书生](#)

CATEGORIES

[akka](#) (20)

[Android](#) (3)

[C++](#) (12)

[CPU](#) (2)

[Framework](#) (54)

[GO](#) (6)

[groovy](#) (6)

[guava](#) (23)

[JAVA](#) (724)

[JVM](#) (37)

[linux](#) (8)

★[添加本文到我的收藏](#)

Related Posts:

1. [《Maven官方文档》创建Archetype](#)
2. [《Maven官方文档》什么是原型\(Archrtype\)?](#)
3. [《Maven官方文档》POM文件](#)
4. [《Maven官方文档》-Maven依赖机制简介](#)
5. [《Maven官方指南》可选的依赖和依赖排除](#)
6. [Maven的Java插件开发指南](#)
7. [spring boot集成mongodb最简单版](#)
8. [Apache Storm 官方文档 —— 使用 Maven 构建 Storm 应用](#)
9. [《Maven官方指南》翻译邀请](#)
10. [《Mavan官方文档》构建生命周期介绍](#)
11. [Maven入门指南（二）](#)
12. [Maven仓库介绍](#)
13. [Springboot 整合 Dubbo/ZooKeeper 详解 SOA 案例](#)
14. [《KAFKA官方文档》5.2 APIs](#)
15. [《Spring Data 官方文档》Reference Documentation至5.2. Examples Repository](#)

Write comment

Comments RSS

Trackback are closed

Comments (0)

No comments yet.

You must be [logged in](#) to post a comment.

[《SLF4J官方文档》SLF4J迁移工具](#)

[《Maven官方文档》POM文件](#)

[January 2016](#) (10)

[Netty](#) (31)

[December 2015](#) (15)

[react](#) (6)

[November 2015](#) (24)

[redis](#) (22)

[October 2015](#) (17)

[Scala](#) (11)

[September 2015](#) (29)

[spark](#) (19)

[August 2015](#) (44)

[Spring](#) (18)

[July 2015](#) (20)

[storm](#) (44)

[June 2015](#) (21)

[thinking](#) (3)

[May 2015](#) (15)

[Velocity](#) (10)

[April 2015](#) (27)

[Web](#) (18)

[March 2015](#) (13)

[zookeeper](#) (1)

[February 2015](#) (11)

[公告](#) (5)

[January 2015](#) (13)

[大数据](#) (33)

[December 2014](#) (26)

[好文推荐](#) (31)

[November 2014](#) (61)

[并发书籍](#) (96)

[October 2014](#) (33)

[并发译文](#) (398)

[September 2014](#) (47)

[感悟](#) (3)

[August 2014](#) (27)

[技术问答](#) (12)

[July 2014](#) (13)

[敏捷管理](#) (6)

[June 2014](#) (32)

[本站原创](#) (86)

[May 2014](#) (45)

[架构](#) (29)

[April 2014](#) (41)

[活动](#) (6)

[March 2014](#) (34)

[网络](#) (6)

[February 2014](#) (38)

[January 2014](#) (19)

[December 2013](#) (10)

[November 2013](#) (4)

[October 2013](#) (20)

[September 2013](#) (38)

[August 2013](#) (48)

[July 2013](#) (26)

[June 2013](#) (16)

[May 2013](#) (9)

[April 2013](#) (17)

[March 2013](#) (41)

[February 2013](#) (25)

[January 2013](#) (57)

[December 2012](#) (9)

[October 2012](#) (1)

[August 2012](#) (1)

