

POM是Maven工程的基本工作单元，是一个XML文件。执行任务时，Maven会在当前目录找POM，获取配置信息。

POM文件需要project元素和三个必需字段：groupId，artifactId，version。

节点	描述
project	工程的根标签。
modelVersion	模型版本需要设置为 4.0。
groupId	这是工程组的标识。它在一个组织或者项目中通常是唯一的。例如，一个银行组织 com.companyname.project-group 拥有所有的和银行相关的项目。
artifactId	这是工程的标识。它通常是工程的名称。例如，消费者银行。groupId 和 artifactId 一起中的位置。
version	这是工程的版本号。在 artifact 的仓库中，它用来区分不同的版本。例如： com.company.bank:consumer-banking:1.0 com.company.bank:consumer-banking:1.1

```
<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">

  <!-- 模型版本 -->
  <modelVersion>4.0.0</modelVersion>

  <!-- 公司或者组织的唯一标志，并且配置时生成的路径也是由此生成， 如
  com.companyname.project-group，maven会将该项目打成的jar包放本地路
  径：/com/companyname/project-group -->
  <groupId>com.companyname.project-group</groupId>

  <!-- 项目的唯一ID，一个groupId下面可能多个项目，就是靠artifactId来区分的 --
  >
  <artifactId>project</artifactId>

  <!-- 版本号 -->
  <version>1.0</version>
</project>
```

## Super POM

父（Super）POM是 Maven 默认的 POM。所有的 POM 都继承自一个父 POM（无论是否显式定义了这个父 POM）。父 POM 包含了一些可以被继承的默认设置。因此，当 Maven 发现需要下载 POM 中的 依赖时，它会到 Super POM 中配置的默认仓库 <http://repo1.maven.org/maven2> 去下载。

Maven 使用 effective pom（Super pom 加上工程自己的配置）来执行相关的目标，它帮助开发者在 pom.xml 中做尽可能少的配置，当然这些配置可以被重写。

- 父pom配置：将需要继承的Jar包的坐标放入标签即可。

1. `<dependencyManagement>`
2. `<dependencies>`
3. `<dependency>`
4. `<groupId>cn.missbe.web.search</groupId>`
5. `<artifactId>resource-search</artifactId>`
6. `<packaging>pom</packaging>`
7. `<version>1.0-SNAPSHOT</version>`
8. `</dependency>`
9. `</dependencies>`
10. `</dependencyManagement>`

- 子pom配置：

1. `<parent>`
2. `<groupId>父pom所在项目的groupId</groupId>`
3. `<artifactId>父pom所在项目的artifactId</artifactId>`
4. `<version>父pom所在项目的版本号</version>`

5. `</parent>`
6. `<parent>`
7. `<artifactId>resource-search</artifactId>`
8. `<groupId>cn.missbe.web.search</groupId>`
9. `<version>1.0-SNAPSHOT</version>`
10. `</parent>`