

maven 中 scope 标签详解

作者: [ibut](#)

原文链接: <https://ld246.com/article/1530002896505>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

最近在做pom优化工作，发现对于maven依赖管理中的scope标签还是有不明白的地方，所以今天就总结一下这方面的知识，scope在maven的依赖管理中主要负责项目的部署

maven的哲学在上次技术分享的时候也提到了：约定大于配置，所以在maven中，很多内容都有默认值，scope的默认值是compile，那么scope还能有哪些选项呢？

scope的分类

- 1.compile：默认值 他表示被依赖项目需要参与当前项目的编译，还有后续的测试，运行周期也参与中，是一个比较强的依赖。打包的时候通常需要包含进去
- 2.test：依赖项目仅仅参与测试相关的工作，包括测试代码的编译和执行，不会被打包，例如：junit
- 3.runtime：表示被依赖项目无需参与项目的编译，不过后期的测试和运行周期需要其参与。与compile相比，跳过了编译而已。例如JDBC驱动，适用运行和测试阶段
- 4.provided：打包的时候可以不用包进去，别的设施会提供。事实上该依赖理论上可以参与编译，测试，运行等周期。相当于compile，但是打包阶段做了exclude操作
- 5.system：从参与度来说，和provided相同，不过被依赖项不会从maven仓库下载，而是从本地文件系统拿。需要添加systemPath的属性来定义路径

scope的依赖传递

A依赖B，B依赖C。当前项目为A，只当B在A项目中的scope，那么c在A中的scope是如何得知呢？

当C是**test**(注意test)或者**provided**时，C直接被**丢弃**，A不依赖C；（排除传递依赖）

否则A依赖C，C的scope继承与B的scope

所以在优化过程中，我们把项目中的一部分依赖，加上了scope->provided标签，也就是说，避免了后打包阶段把一些可以从ear中已经提供的包排除在外，去掉重复的打包过程。