



Maven——构建工具

回顾日常的工作中，我们能发现每天都有相当一部分时间花费在了项目的编译、单元测试、生成文档、打包和部署等繁琐的工作上，这就是手动构建项目所需要的代价。而通过Maven，我们需要做的只是配置好项目的POM，然后输入简单的Maven命令，然后所有繁琐的步骤都能够自动完成。

Maven最大化的消除了构建的重复，抽象了构建生命周期，并且为绝大部分的构建任务提供了已实现的插件，我们需要做的就是遵循Maven定义的生命周期过程，调用相关的命令。例如，我们只需要按照Maven的约定编写好测试用例，当我们运行构建的时候，测试用例会自动执行。此外，如果有特殊的需求，我们还可以实现自定义插件。

Maven还有一个优点。它能够帮助我们标准化构建过程。不使用Maven的情况下，不同的项目会存在不同的构建方法；有了Maven之后，所有项目的构建命令都是简单一致的，有利于促进团队的标准化。

Maven——依赖管理工具

在日常的项目管理中，对于项目所引用的第三方类库（如jar包）的管理无疑是一个令人头疼的问题，手工解决依赖的版本冲突、依赖臃肿等问题是十分枯燥的。而Maven通过坐标系统唯一的定位了每一个依赖，并提供了一系列简单的依赖管理方法，因此借助Maven我们能够轻松的对项目的依赖进行管理。

Maven为全世界的Maven用户提供了免费的中央仓库，在其中几乎可以找到任何流行的开源类库。通过一些Maven的衍生工具，我们能够对其进行快速的搜索，获取其Maven坐标。只要定位了坐标，Maven就能够帮助我们自动下载，省却了手工劳动。

Maven——项目信息管理工作

Maven还能帮助我们管理项目的项目信息，包括项目描述、开发者信息、版本控制信息、许可证和缺陷管理信息等，除了直接的项目信息，通过Maven的插件，我们还能够轻松的获取项目文档、测试报告、源码版本日志等项目信息。

最后我们来看看Maven的整体架构图，在学习了后面的篇章后相信你会对该图有更深刻的理解。