



链滴

Gradle Wrapper 介绍

作者: [xiegangd](#)

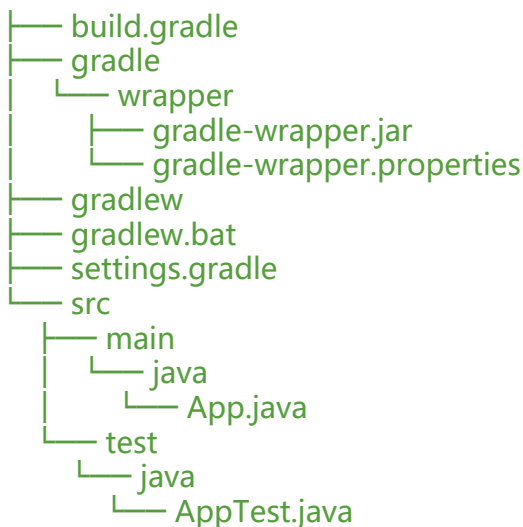
原文链接: <https://ld246.com/article/1536806027914>

来源网站: [链滴](#)

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

回顾

上一篇文章介绍了如何使用 [Gradle 快速创建 Java 项目](#)，开始讲 Gradle Wrapper 之前先来重温一下通过 `gradle init` 命令创建的 Java 项目目录结构



当时说过：有 2 种方式来编译这个示例项目

1. 通过本机安装好的 `gradle` 命令编译
2. 通过当前项目根目录下的 `gradlew` 命令编译（官方推荐的使用方式）

这篇文章的目的就是要介绍为什么官方推荐使用通过当前项目根目录下的 `gradlew` 命令编译项目

Gradle Wrapper 要解决什么问题

Gradle Wrapper 的目标是：任何人、任何时候、在任何电脑上拿到 `gradle` 创建的项目都能被准确的编译、执行

想象以下几个场景：

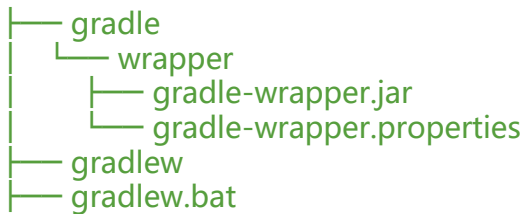
- 把自己写好的代码共享给其他人使用，可能出现如下情况
 - 对方电脑没有安装 `gradle`
 - 对方电脑安装过 `gradle`，但是版本太旧了
- 自己换电脑了，同样也会遇到上一种情况
- 翻出了自己几年前写的代码，但是当前电脑的 `gradle` 版本太新了，无法直接编译几年前的旧代码

如何让上面的问题不再是问题，无论是哪种异常情况都能得到正确的编译？

今天的主角 Gradle Wrapper 就是为了解决以上问题而存在的，这也是官方推荐使用它的原因

如何解决

项目根目录下与 Gradle Wrapper 相关的几个配置文件



- gradlew 是 Linux, Mac, 类 unix 系统上的可执行脚本
- gradlew.bat 是 Windows 系统上的可执行脚本
- gradle-wrapper.properties 文件记录着以下信息
 - 当前项目想使用的 gradle 版本
 - 去哪里下载这个版本的 gradle
 - 下载之后的压缩包保存在哪里
 - 压缩包解压到哪里

当我们第一次执行 `./gradlew build` 命令的时候, gradlew 会读取 `gradle-wrapper.properties` 文件配置信息, 准确的将指定版本的 gradle 下载并解压到指定的位置。之后执行的 `./gradlew` 所有命令是使用指定的 gradle 版本, 完美的解决了上面提出的问题

疑问

到底什么时候使用自己安装的 gradle, 什么时候使用当前项目下的 gradlew

- 从其他人获得的代码, 使用当前项目下的 gradlew
- 使用自己安装的 gradle 新建项目。这样就等于给当前项目指定了希望其他人拿到项目之后使用的 gradle 版本。这种情况下由于我们自己安装过的 gradle 版本就是当前项目 gradlew 定义的版本, 所以可以直接用自己安装的 gradle 执行各种构建命令, 因为跟 gradlew 执行的效果一模一样