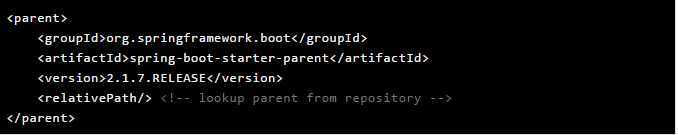
https://mp.weixin.qq.com/s/E\_hAnukSLqDEv78gP7ThOg

# 理解 spring-boot-starter-parent

前面和大伙聊了 Spring Boot 项目的三种创建方式，这三种创建方式，无论是哪一种，创建成功后，pom.xml 坐标文件中都有如下一段引用：



对于这个 parent 的作用，你是否完全理解？有小伙伴说，不就是依赖的版本号定义在 parent 里边吗？是的，没错，但是 parent 的作用可不仅仅这么简单哦！本文松哥就来和大伙聊一聊这个 parent 到底有什么作用。

# 基本功能

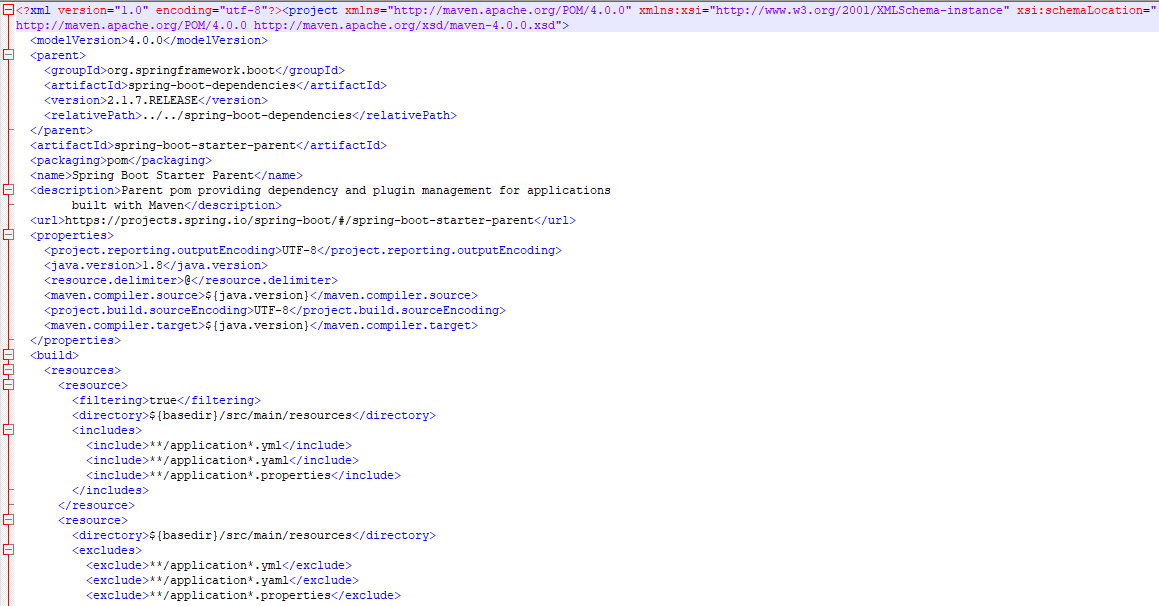
当我们创建一个 Spring Boot 工程时，可以继承自一个 spring-boot-starter-parent ，也可以不继承自它，我们先来看第一种情况。先来看 parent 的基本功能有哪些？

1. 定义了 Java 编译版本为 1.8 。
2. 使用 UTF-8 格式编码。
3. 继承自 spring-boot-dependencies，这个里边定义了依赖的版本，也正是因为继承了这个依赖，所以我们在写依赖时才不需要写版本号。
4. 执行打包操作的配置。
5. 自动化的资源过滤。
6. 自动化的插件配置。
7. 针对 application.properties 和 application.yml 的资源过滤，包括通过 profile 定义的不同环境的配置文件，例如 application-dev.properties 和 application-dev.yml。

**请注意，由于application.properties和application.yml文件接受Spring样式占位符** ${...} **，因此 Maven 过滤更改为使用** @..@ **占位符，当然开发者可以通过设置名为 resource.delimiter 的Maven 属性来覆盖** @..@**占位符。**

## 源码分析

当我们创建一个 Spring Boot 项目后，我们可以在本地 Maven 仓库中看到这个具体的 parent 文件，以 2.1.7 这个版本为例，松哥 这里的路径是 C:\Users\sang\.m2\repository\org\springframework\boot\spring-boot-starter-parent\2.1.7.RELEASE\spring-boot-starter-parent-2.1.7.RELEASE.pom ,打开这个文件，快速阅读文件源码，基本上就可以证实我们前面说的功能，如下图：



我们可以看到，它继承自 spring-boot-dependencies ，这里保存了基本的依赖信息，另外我们也可以看到项目的编码格式，JDK 的版本等信息，当然也有我们前面提到的数据过滤信息。最后，我们再根据它的 parent 中指定的 spring-boot-dependencies 位置，来看看 spring-boot-dependencies 中的定义：

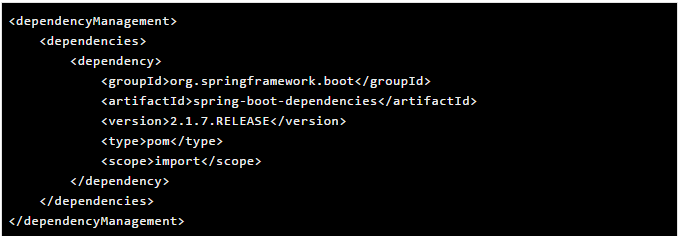


在这里，我们看到了版本的定义以及 dependencyManagement 节点，明白了为啥 Spring Boot 项目中部分依赖不需要写版本号了。

# 不用 parent

但是并非所有的公司都需要这个 parent ，有的时候，公司里边会有自己定义的 parent ，我们的 Spring Boot 项目要继承自公司内部的 parent ，这个时候该怎么办呢？

一个简单的办法就是我们自行定义 dependencyManagement 节点，然后在里边定义好版本号，再接下来在引用依赖时也就不用写版本号了，像下面这样：



这样写之后，依赖的版本号问题虽然解决了，但是关于打包的插件、编译的 JDK 版本、文件的编码格式等等这些配置，在没有 parent 的时候，这些统统要自己去配置。