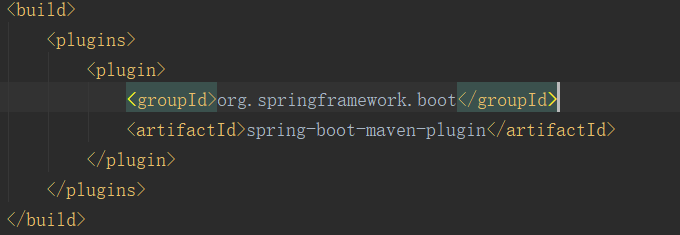
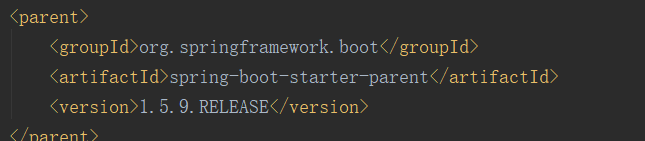
# Springboot

1. 简化spring开发，约定大于配置
2. Starters自动依赖于版本控制

Springboot可以添加maven插件后，直接把其打包成jar包，然后java –jar jar包名就可以直接运行



## Pom文件探究



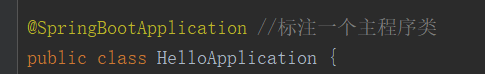
他来真正管理springboot版本管理依赖，以后导入的jar包就不用写版本了

Spring-boot-starter：springboot场景启动器

Springboot将所有功能都抽取出来，做成一个个starter，只需要在相应项目里导入对应的starter就可以了

## 原理

### 启动类



该注解有@SpringbootConfiguration和@EnableAutoConfiguration

1、@SpringbootConfiguration底层是@Configuration配置类

2、@ EnableAutoConfiguration：开启自动配置，他又包含下面这两个

@AutoConfigurationPackage  
@Import({EnableAutoConfigurationImportSelector.class})

而其中的@AutoConfiurationPackage有导入了一个Registry，该类用来扫描SpringbootApplication所在包的所有组件，并注册到spring容器

又导入了EnableAutoConfiurationImportSelector类，它将选择要导入的组件（导入很多的自动配置类）注册到容器。

那他是怎么找到这些组件的，启动的时候去类路径下的/META-INF/spring-factories获取自动配置类

以前javaee配置文件都有org.springframework.boot.autoconfigure下面的类来代替了

### 配置文件

推荐使用Application.yml文件

要读取配置文件中的内容

1、添加pom

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-configuration-processor</artifactId>  
 <optional>true</optional>  
</dependency>

1. 添加注解

添加@ConfigurationProperties(prefix = "person")

1. 把该类注册到容器中

### 自动配置

1. springboot启动时候加载主配置类，开启了自动配置功能
2. 自动配置功能导入了许多的自动配置组件到容器中