# Spring Boot入门

## 简介

背景：J2EE有太多复杂的配置以及部署流程，第三方技术集成难度大。

解决：Spring 全家桶；**Spring Boot ->一站式解决方案；**Spring Cloud->分布式整体解决方案。

## 微服务

是一种架构风格，即服务微化。

一个应用应该是一组小型服务；可以通过HTTP的方式进行互通。

每一个功能元素其实是一个**可独立替换**和**可独立升级**的软件单元。

## 环境准备

环境约束：

jdk 1.8

Maven3.x：maven安装的目录下的settings.xml文件中加入以下内容。

<profiles>

        <profile>

            <id>jdk-1.8</id>

            <activation>

                <activeByDefault>true</activeByDefault>

                <jdk>1.8</jdk>

            </activation>

            <properties>

                <maven.compiler.source>1.8</maven.compiler.source>

                <maven.compiler.target>1.8</maven.compiler.target>

                <maven.compiler.compilerVersion>1.8</maven.compiler.compilerVersion>

            </properties>

        </profile>

    </profiles>

Idea2017

Spring boot 1.5.9

## 四、简单的Spring Boot项目

1.创建maven项目后引入相关的spring-boot依赖

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.3.1.RELEASE</version>

</parent>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

</dependencies>

1. 编写主程序

@SpringBootApplication

@RestController

public class **MainApplication** {

public static void **main**(String[] args) {

*//Spring应用启动*

SpringApplication.**run**(MainApplication.class, args);

}

@GetMapping("/hello")

public String **hello**(@RequestParam(value = "name", defaultValue = "World") String name) {

return String.**format**("Hello %s!", name);

}

}

1. 简化部署

pom.xml中加入打包的相关依赖

<build>

    <plugins>

        <plugin>

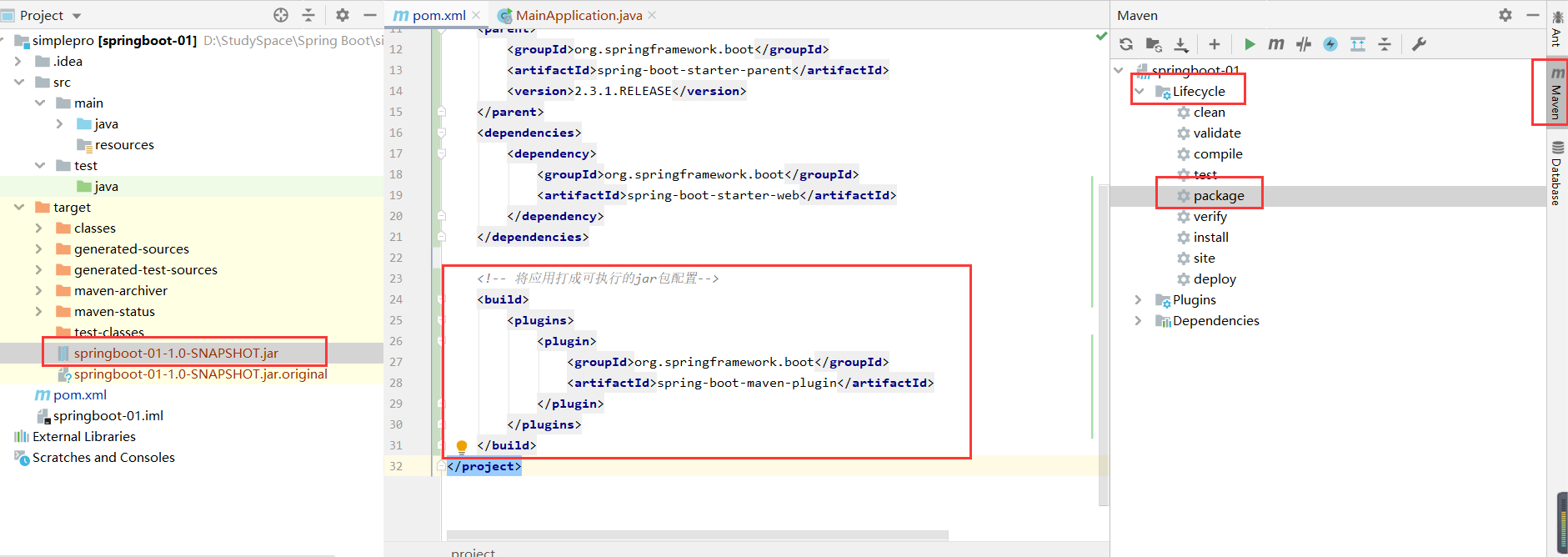
            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-maven-plugin</artifactId>

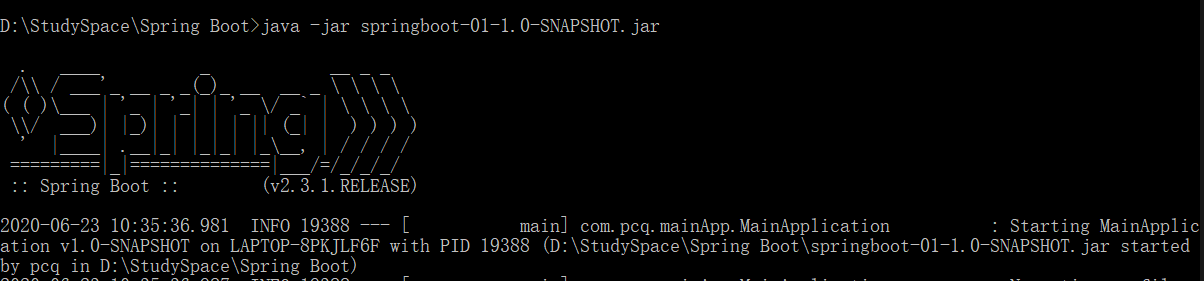
        </plugin>

    </plugins>

</build>

通过package程序打包，在target目录下会生成可执行jar包。

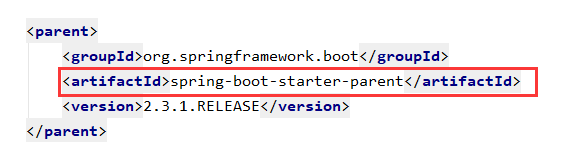
通过java -jar jar包的命令方式可运行。



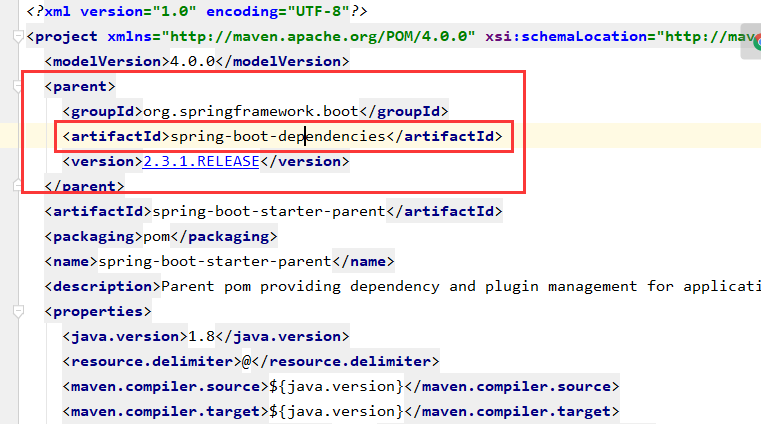
## 五、SpringBoot程序启动说明

### 1.pom.xml文件

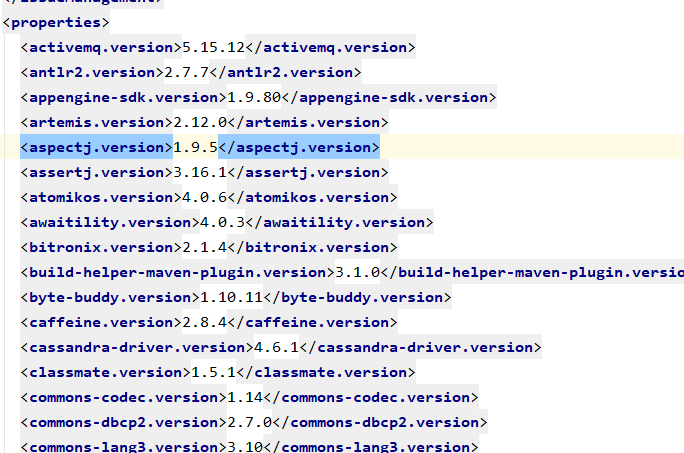
#### 父项目



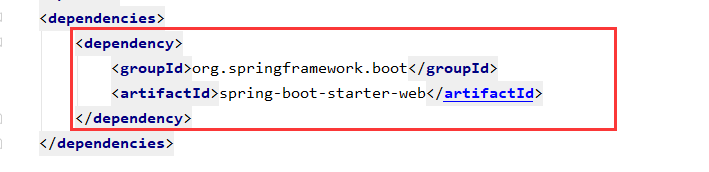
点进去发现它还有一个父项目



该父项目spring-boot-dependencies下的properties属性配置了大量的依赖版本号。它就是springboot的仲裁中心，以后导入的依赖都无需声明版本，但该仲裁中心之外定义的依赖仍然需要声明版本。



#### 导入的依赖包

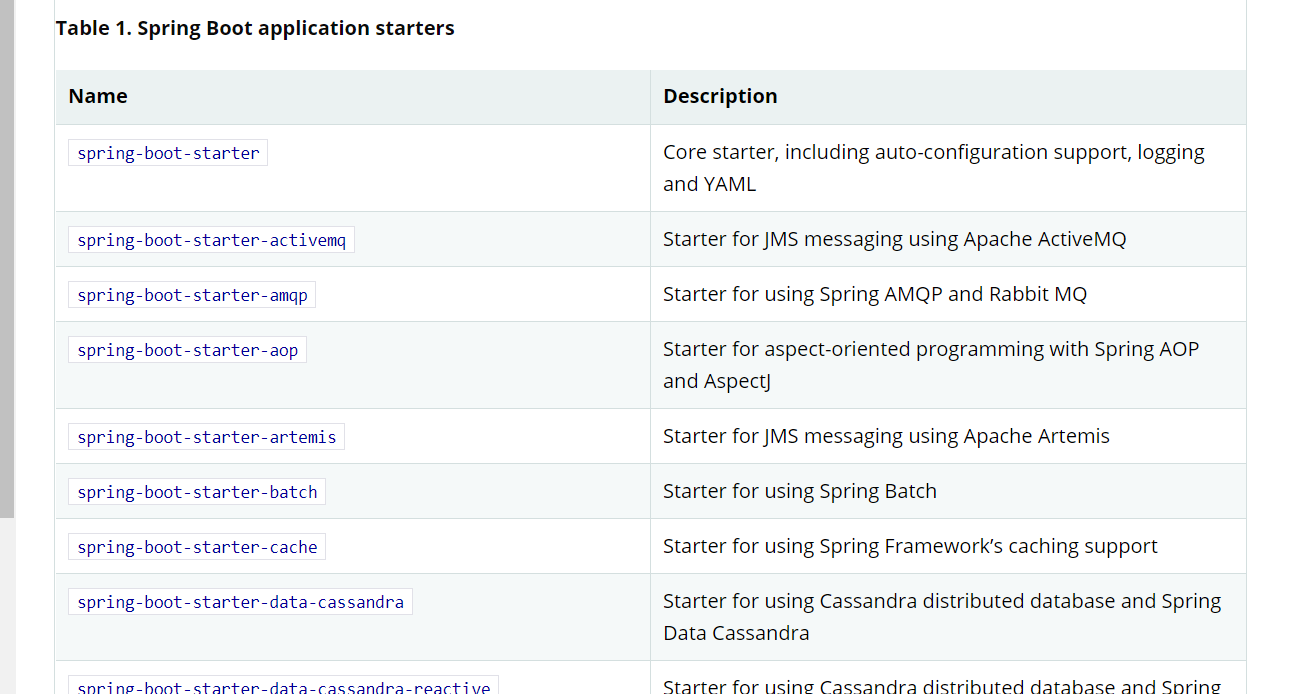


spring-boot-starter：springboot场景启动器。

spring-boot-starter-web：web应用场景启动器。它导入了web模块正常运行所需要依赖的组件。



**Spring Boot将所有的功能场景都抽取出来，做成一个个starters（启动器），项目中需要什么样的业务场景，就引入什么样的场景启动器，这些启动器会自动引入相关的依赖包。**



### 主程序类