SpringBoot打包与引入

利用spi机制注入

参考：

spring boot项目打jar包，其他项目调用_大大瓜先生的博客-CSDN博客_springb...

;springboot 生成第三方可用jar包；spring boot项目打jar包，其他项目调用；SpringBoot 项目打成jar...

https://blog.csdn.net/qazxcvbnm_/article/details/123274866?spm=1001.2101.3001.6650.1&utm_...

无标题

_

1. 创建一个springboot项目，实现自己需要的功能
2. **不能使用springboot默认的打包方式，将pom文件里面的 `<build>` 标签下的内容进行替换**

XML | 复制代码

```xml
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <configuration>
        <source>1.8</source> <!--指明源码用的Jdk版本-->
        <target>1.8</target> <!--指明打包后的Jdk版本-->
      </configuration>
    </plugin>
  </plugins>
  <resources>
    <resource>
      <directory>src/main/resources</directory>
      <excludes>
        <!--打包时需要被排除的文件-->
        <exclude>application.yml</exclude>
      </excludes>
    </resource>
  </resources>
</build>
```

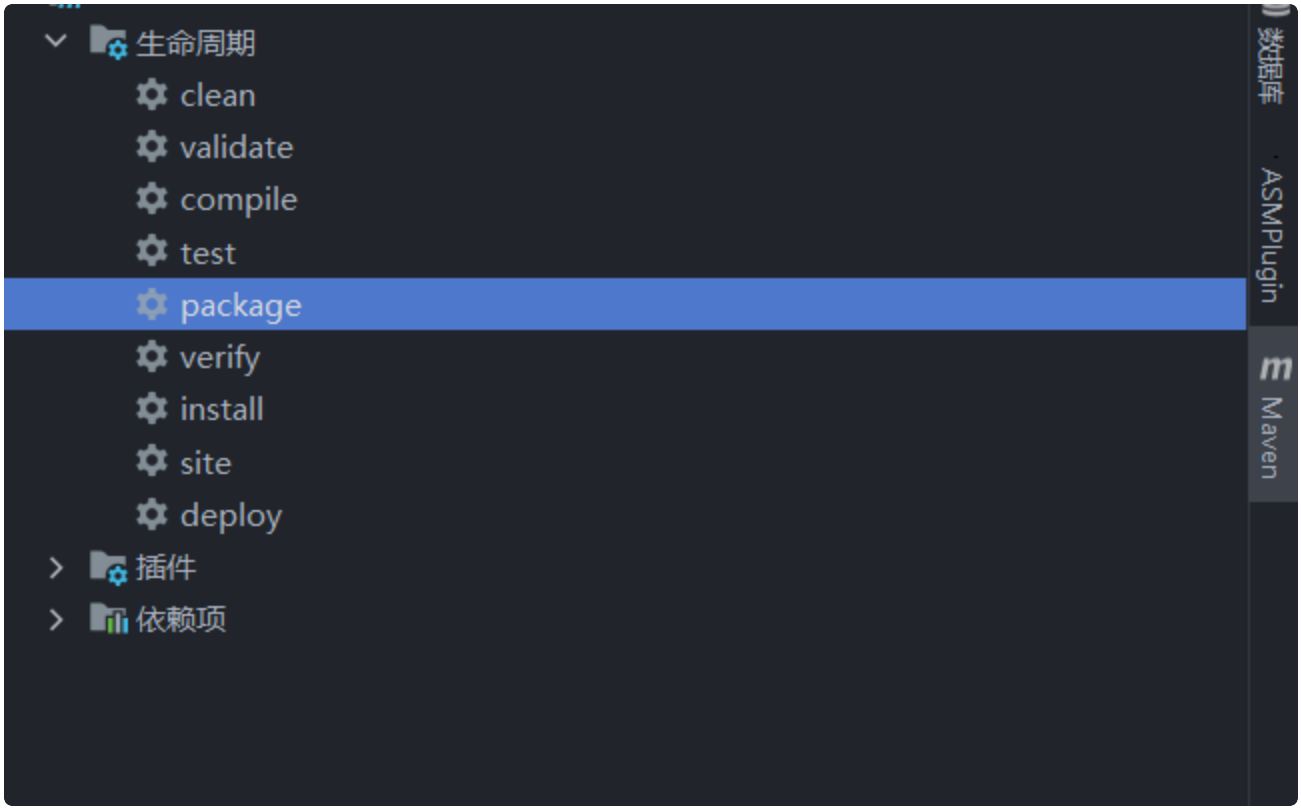
3. 可以剔除一些不需要的文件：application.yml、测试类等
4. 新增一个配置目录config，在该目录下新建一个配置类xxxConfig.java，basePackages中填入项目src目录下的包的根路径，用于指定需要扫描的beans有哪些

```java
@Configuration
@ComponentScan(basePackages = {"xxx.xxxx.xxx.**"})
public class xxxConfig {
}
```

5. **在resources目录下新建META-INF/spring.factories文件，写入以下内容。**
   最关键的就是 resources/META-INF/spring.factories 文件，当项目引入该jar包启动时，Spring会扫描所有jar包下面的 spring.factories 文件，进行相应的自动配置处理。
   其中org.springframework.boot.autoconfigure.EnableAutoConfiguration 代表自动配置的key，即代表需要自动配置哪些类，\ 可以理解为一个换行符，则该行下面的每行当做一个参数。 `xxxxx` 填入第四步中生成的对应的配置类的全限定名

```xml
org.springframework.boot.autoconfigure.EnableAutoConfiguration=\
xxxxxx
```

6. 使用idea右侧的mvn package打包生成jar包



7. 在需要使用该jar包的项目中的project structure中的libraries中引入jar包即可使用