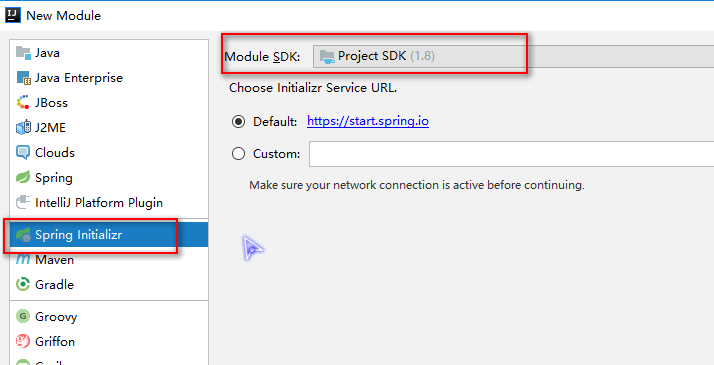
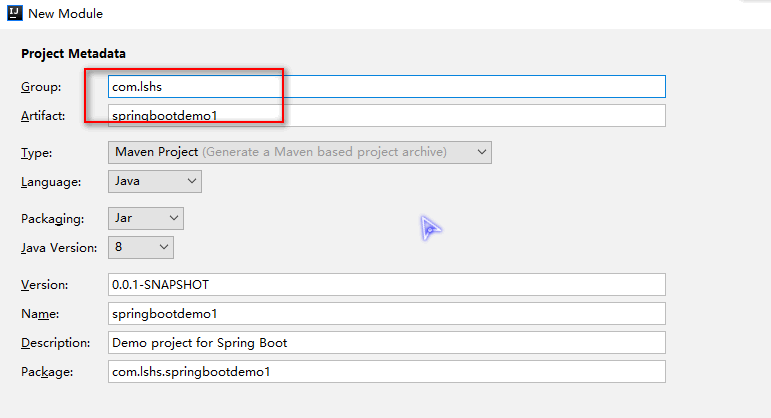


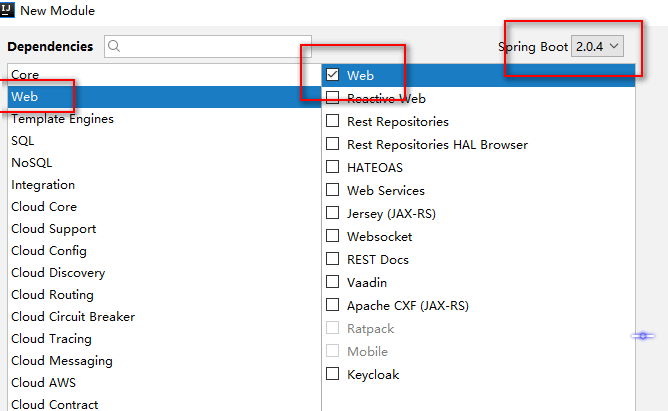
1.打开IDEA,创建新项目，选择Spring Initializr



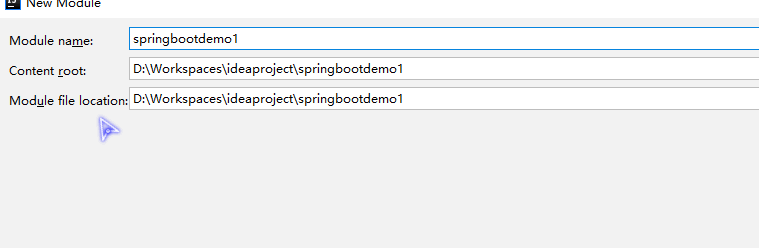
2.输入Artifact



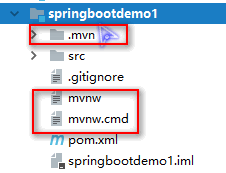
3.勾选Web



4.点击finish完成



5.进入项目，可以将以下内容删除



pom.xml文件：

**[html]** [view plain](https://blog.csdn.net/lom9357bye/article/details/69677120) [copy](https://blog.csdn.net/lom9357bye/article/details/69677120)

1. <code class="language-html"><?xml version="1.0" encoding="UTF-8"?>
2. <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3. xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4. <modelVersion>4.0.0</modelVersion>
6. <groupId>com.example</groupId>
7. <artifactId>springbootdemo</artifactId>
8. <version>0.0.1-SNAPSHOT</version>
9. <packaging>jar</packaging>
11. <name>springbootdemo</name>
12. <description>Demo project for Spring Boot</description>
14. <!--起步依赖-->
15. <parent>
16. <groupId>org.springframework.boot</groupId>
17. <artifactId>spring-boot-starter-parent</artifactId>
18. <version>1.5.2.RELEASE</version>
19. <relativePath/> <!-- lookup parent from repository -->
20. </parent>
22. <properties>
23. <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
24. <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
25. <java.version>1.8</java.version>
26. </properties>
28. <dependencies>
29. <!--开发web项目相关依赖-->
30. <dependency>
31. <groupId>org.springframework.boot</groupId>
32. <artifactId>spring-boot-starter-web</artifactId>
33. </dependency>
34. <!--springboot单元测试-->
35. <dependency>
36. <groupId>org.springframework.boot</groupId>
37. <artifactId>spring-boot-starter-test</artifactId>
38. <scope>test</scope>
39. </dependency>
40. </dependencies>
42. <!--maven构建-->
43. <build>
44. <plugins>
45. <plugin>
46. <groupId>org.springframework.boot</groupId>
47. <artifactId>spring-boot-maven-plugin</artifactId>
48. </plugin>
49. </plugins>
50. </build>

53. </project>
54. </code>

6.创建一个HelloController

1. package com.example;
3. import org.springframework.web.bind.annotation.RequestMapping;
4. import org.springframework.web.bind.annotation.RestController;
6. @RestController
7. public class HelloController {
9. @RequestMapping("/hello")
10. public String hello() {
11. return "hello,this is a springboot demo";
12. }
13. }

7.程序自动生成的SpringbootdemoApplication，会有一个@SpringBootApplication的注解，这个注解用来标明这个类是程序的入口

1. package com.example;
3. import org.springframework.boot.SpringApplication;
4. import org.springframework.boot.autoconfigure.SpringBootApplication;
6. //入口
7. @SpringBootApplication
8. public class SpringbootdemoApplication {
10. public static void main(String[] args) {
11. SpringApplication.run(SpringbootdemoApplication.class, args);
12. }
13. }

@SpringBootApplication开启了Spring的组件扫描和springboot的自动配置功能，相当于将以下三个注解组合在了一起

（1）@Configuration：表名该类使用基于Java的配置,将此类作为配置类

（2）@ComponentScan：启用注解扫描

（3）@EnableAutoConfiguration：开启springboot的自动配置功能

8.运行SpringbootdemoApplication类

测试：

在地址栏中输入http://localhost:8080/hello

9.使用启动jar包的方式启动

（1）首先进入项目所在目录，如果是mac系统在项目上右键，选择Reveal in Finder,Windows系统在项目上右键选择Show in Explorer，即可打开项目所在目录

（2）打开终端，进入项目所在目录

     cd /Users/shanml/IdeaProjects/SpringbootDemo

     输入mvn install,构建项目

（3）构建成功后，在项目target文件夹下会多出一个jar包

（4）使用java -jar springbootdemo-0.0.1-SNAPSHOT.jar

     启动jar包即可