**Spring boot快速搭建**

**一、spring boot简介**

Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。通过这种方式，Spring Boot致力于在蓬勃发展的快速应用开发领域(rapid application development)成为领导者。（出自百度百科）

**二、特点**

* 创建独立的Spring应用程序
* 嵌入的Tomcat，无需部署WAR文件
* 简化Maven配置
* 自动配置Spring
* 提供生产就绪型功能，如指标，健康检查和外部配置
* 绝对没有代码生成并且对XML也没有配置要求

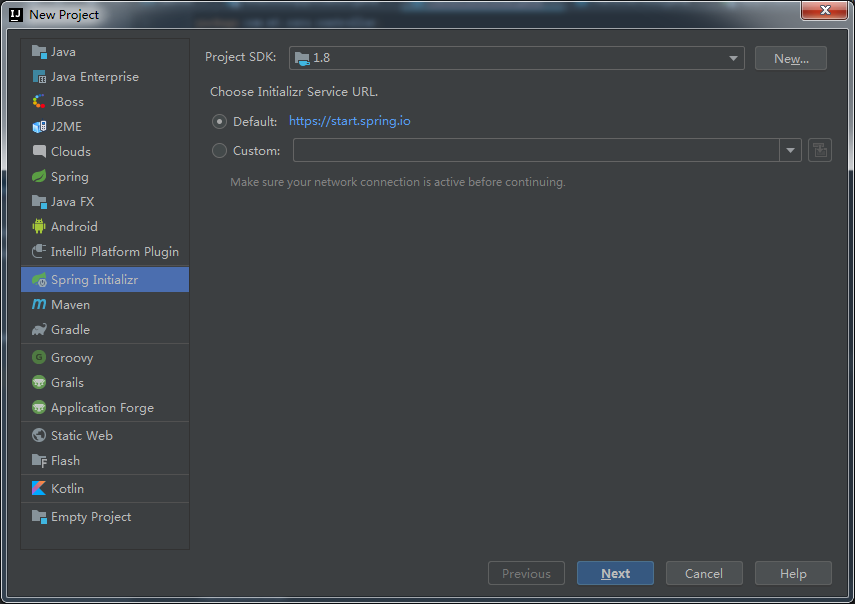
**三、搭建准备**

基础环境 maven + JDK1.8

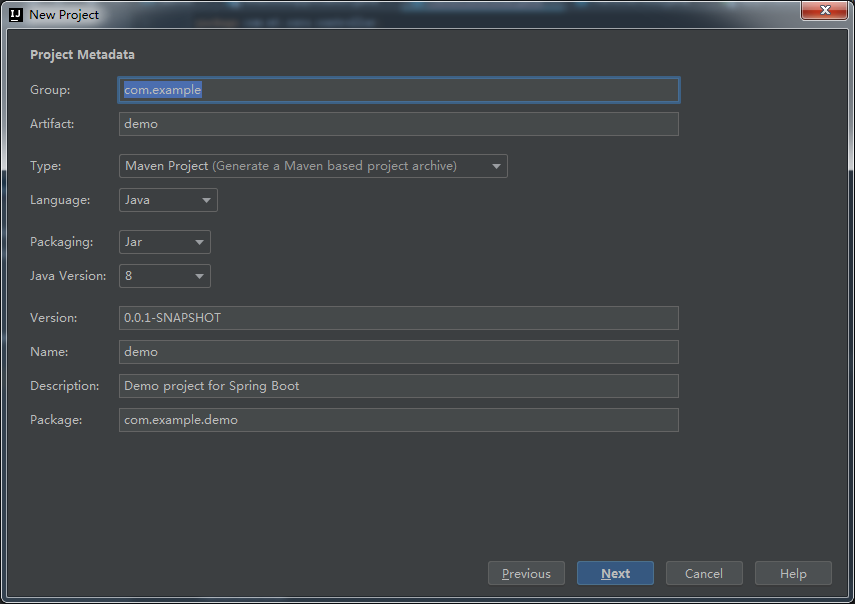
开发工具 idea

**四、开始搭建**

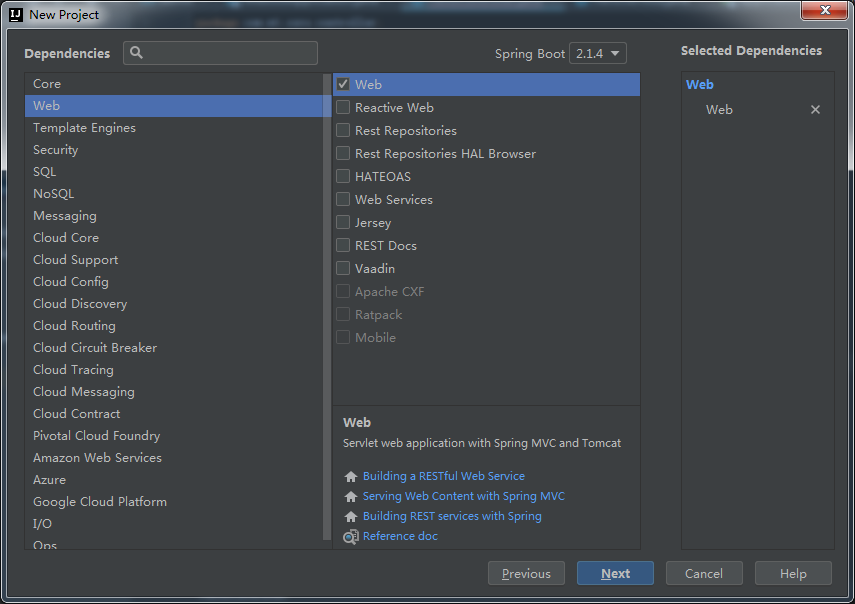
1、在idea中创建新的project，选择Spring initializr，然后点击next，如下图：



进入如下界面,点击next



然后选择 web中的web，点击next，然后点击finish，如下图：



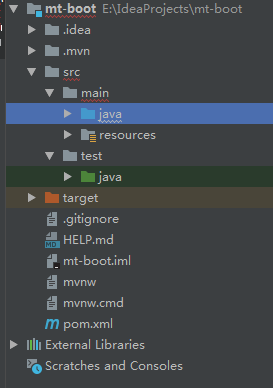
完成项目的创建。

2、项目基本结构

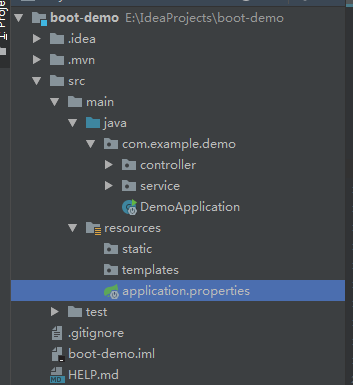
src.main.java 主程序的入口，核心代码的存放地

src.main.resource 存放配置文件

src.test.java 测试单元



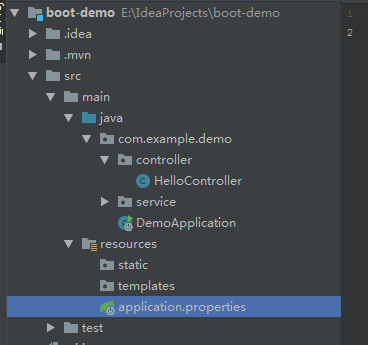
3、配置application.properties文件，如下图位置：



配置文件内容，配置端口：

##项目占用端口  
server.port= 81

4、编写controller接口，在main.java.com.example.demo包下创建controller包，然后创建HelloController对象，如下图：

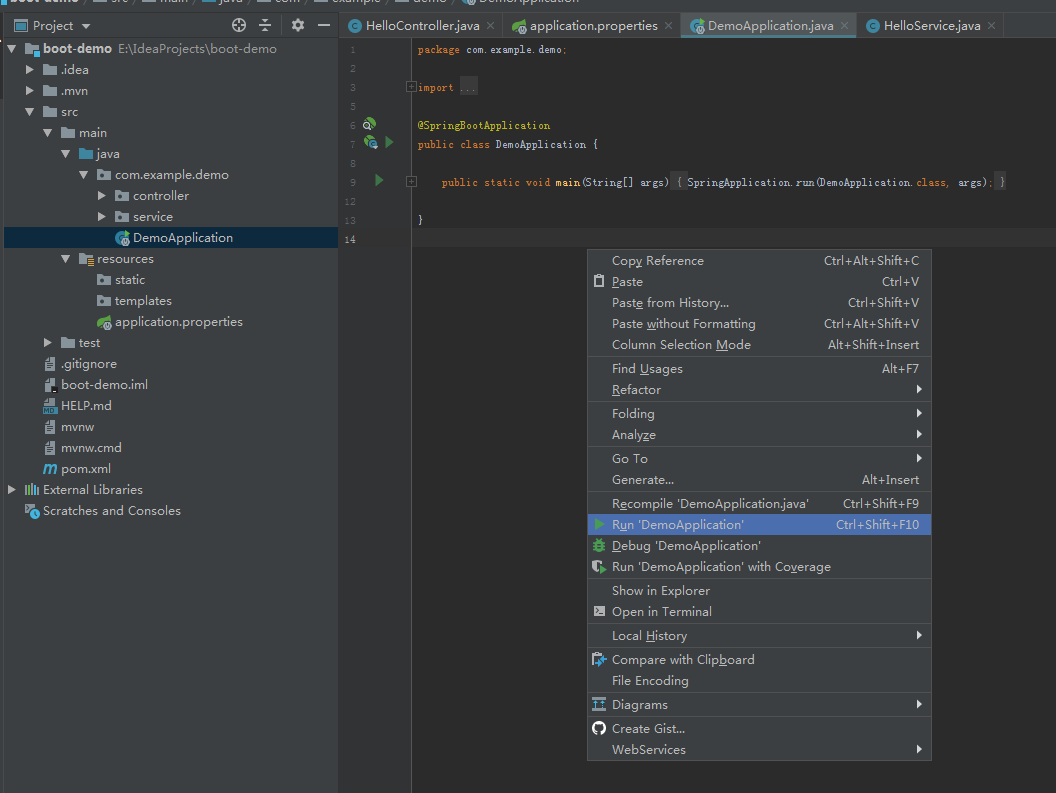


HelloController 代码如下：

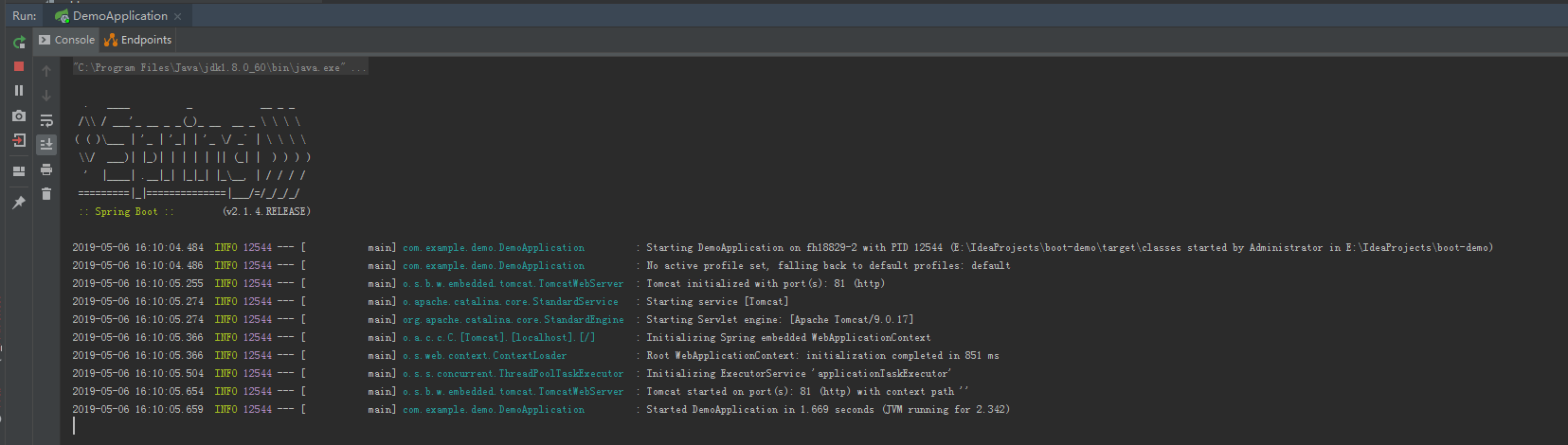
@RestController  
public class HelloController {  
  
 @RequestMapping("/hello")  
 public String hello(@RequestParam("name") String name)  
 {  
 return "hello "+name;  
 }

}

现在可以尝试第一次运行程序了，打开DemoApplication类，右键运行，如下图

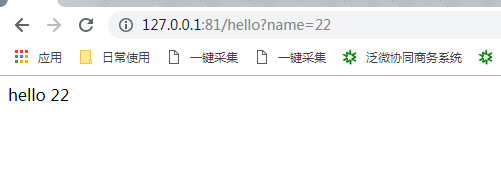


控制台有如下输出，说明完成程序启动



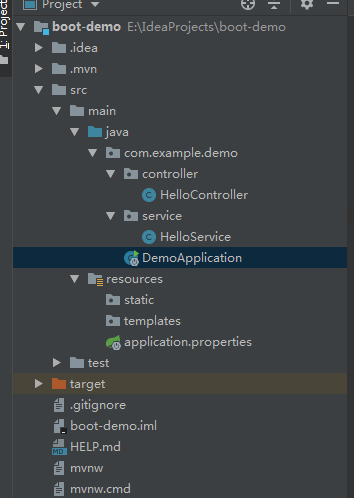
在浏览器中输入url ：<http://127.0.0.1:81/hello?name=22>

显示如下内容，接口类编写成功



5、进行注入操作

在main.java.com.example.demo包下创建service包，然后创建HelloService对象，如下图：



HelloService代码如下：

@Component  
public class HelloService {  
  
 public String getMessage()  
 {  
 return "I am from srvice ";  
 }  
}

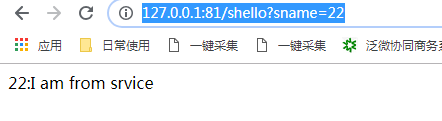
修改HelloController代码，如下：

@RestController  
public class HelloController {  
  
 @RequestMapping("/hello")  
 public String hello(@RequestParam("name") String name)  
 {  
 return "hello "+name;  
 }  
  
 @Autowired  
 HelloService service;  
 @RequestMapping("/shello")  
 //url中sname映射为name参数  
 public String sHello(@RequestParam("sname") String name)  
 {  
 return name +":" +service.getMessage();  
 }  
}

编写完成，进行第二次测试，运行程序（DemoApplication 同上一次）

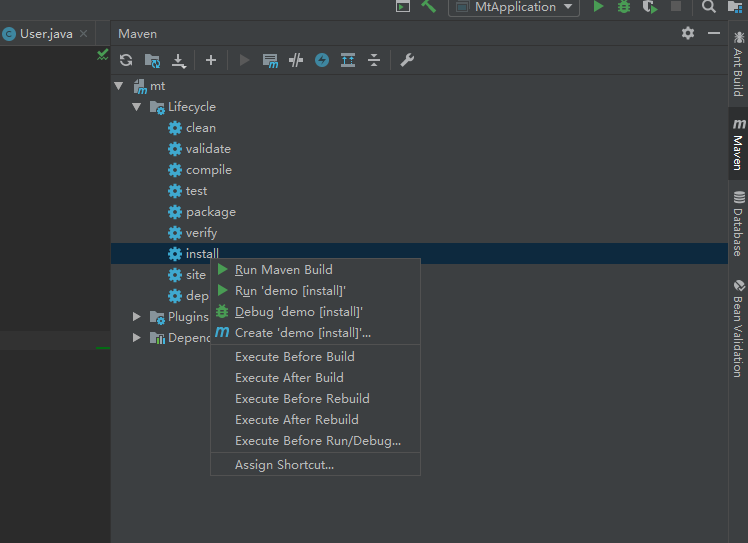
在浏览器中输入url：<http://127.0.0.1:81/shello?sname=22>

显示如下内容：

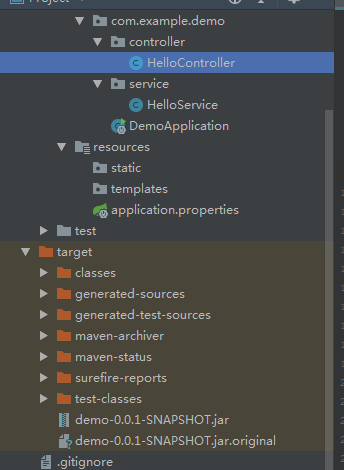


**五、生成jar包，并运行**

运行maven build，如下图：



打开target文件，找到demo-0.0.1-SNAPSHOT.jar，如下图



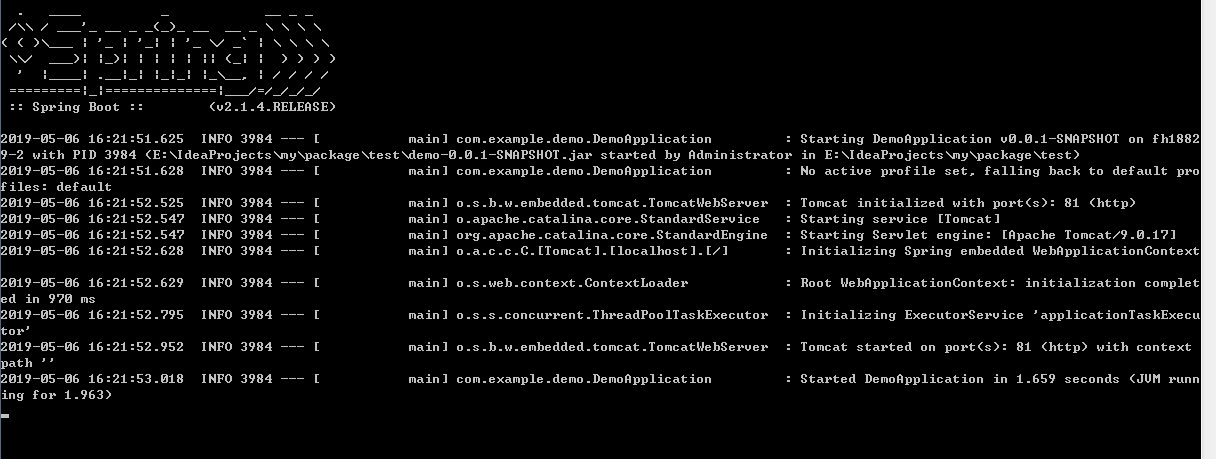
拷贝jar包到要部署的位置，打开dos窗口 输入命令（已经配置JavaHome）

java -jar demo-0.0.1-SNAPSHOT.jar

如果没有配置 Javahome的话，运行如下命令：

D:/jdk/bin/java -jar demo-0.0.1-SNAPSHOT.jar(以实际的JDK位置为准)

显示如下内容，运行成功：



可以在浏览器中输入 <http://127.0.0.1:81/shello?sname=22>试下。