20%的核心功能，学会，有些同学，也许之前看过一些书，或者是看过一些视频课程，那么其实可能已经按照我说的那种大而全的方式先学了一遍，可能都会了，那就更好。建议大家说就当是复习，跟着我走一遍，其实就花几个小时的时间，带着大家快速学会这个技术。而且我认为，如果只是学习spring boot的功能的话，不光是学习20%的核心功能，你只要花点时间，把spring boot挑一本书，或者挑一个视频课程，整个看一遍，跟着走一遍，只要不是傻子，都能学会。但是我这里的重点，集中在做完项目之后的spring boot核心源码剖析那块，那是有技术难度和挑战的亮点。

1、创建一个空的maven工程

2、在pom.xml中引入spring boot的依赖

<!-- 继承spring boot的父工程 -->

<!-- spring boot父工程直接约束了常用依赖的版本 -->

<parent>

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

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

   <version>1.5.9.RELEASE</version>

</parent>

<!-- 添加一个支持web应用的spring boot依赖 -->

<!-- spring boot会根据我们引入的依赖，判断出来我们要开发一个web工程 -->

<!-- 接着会对web工程需要的东西，比如tomcat服务器，自动进行配置 -->

<dependencies>

   <dependency>

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

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

   </dependency>

</dependencies>

<!-- 引入一个spring boot插件，可以支持我们打包程序 -->

<!-- 打包时是需要将所有依赖的第三方jar包都打进来的，spring boot这个插件可以支持 -->

<build>

   <plugins>

       <plugin>

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

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

       </plugin>

   </plugins>

</build>

3、编写代码

import org.springframework.boot.\*;

import org.springframework.boot.autoconfigure.\*;

import org.springframework.stereotype.\*;

import org.springframework.web.bind.annotation.\*;

// @RestController会告诉spring boot，这是一个spring mvc的controller

// 同时@RestController默认就是支持将返回值直接给浏览器的，而不是去渲染视图

@RestController

// auto configuration，是spring boot最最重要和核心的功能之一

// spring boot的核心思想，就是不要去做太多的xml配置，全部基于约定的一些规则，自动完成一些配置

// auto configuration，就会根据我们引入的一些依赖，比如引入spring-boot-starter-web，就会根据我们要开发web程序的特点，自动完成tomcat服务器等相关的web配置

@EnableAutoConfiguration

public class HelloWorldApplication {

   // @RequestMapping，就是做http请求的路由

   @RequestMapping("/hello")

   public String sayHello() {

       return "hello world";

  }

   public static void main(String[] args) throws Exception {

       SpringApplication.run(HelloWorldApplication.class, args);

  }

}

4、直接运行spring boot应用

直接在eclipse中执行main方法，就可以启动spring boot应用

访问浏览器：http://localhost:8080/hello，可以看到浏览器下显示：

hello world

5、将spring boot应用打包成可以直接执行的jar包

因为我们之前使用了spring-boot-maven-plugin插件，所以可以直接打包，就会包含依赖的所有第三方jar包，成为一个可以执行的jar包

mvn pakcage

然后执行这个jar包：java -jar target\springboot-demo-1.0.0.jar

就会启动spring boot应用，访问浏览器可以看到响应结果

853769620