**springboot的环境搭建**

环境要求:

1. MAVEN 3.x+
2. Spring FrameWork 4.x+
3. JDK7.x +
4. Spring Boot 1.5.x+

**3.1 项目中引入依赖**

```xml org.springframework.boot spring-boot-starter-parent 1.5.7.RELEASE

<dependencies>

<!--引入springboot的web支持-->

<dependency>

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

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

</dependency>

</dependencies>

```

**3.2 引入配置文件**

​ 项目中src/main/resources/application.yml

**3.3 建包并创建控制器**

```java //在项目中创建指定的包结构 /\* com +| baizhi +| controller \*/ @Controller @RequestMapping("/hello") public class HelloController { @RequestMapping("/hello") @ResponseBody public String hello(){ System.out.println("======hello world======="); return "hello"; } }

```

**3.4 编写入口类**

java //在项目中如下的包结构中创建入口类 Application /\* com +| baizhi \*/ @SpringBootApplication public class Application { public static void main(String[] args) { SpringApplication.run(Application.class,args); } }

**3.5 运行main启动项目**

```java o.s.j.e.a.AnnotationMBeanExporter : Registering beans for JMX exposure on startup s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 8989 (http) com.baizhi.Application : Started Application in 2.152 seconds (JVM running for 2.611)

//说明: 出现以上日志说明启动成功 ```

**3.6 访问项目**

java //注意: springboot的项目默认没有项目名 //访问路径: http://localhost:8080/hello/hello

**4. 启动tomcat端口占用问题**

yml server: port: 8989 #用来指定内嵌服务器端口号 context-path: /springboot #用来指定项目的访问路径

**5. springboot相关注解说明**

```java /\*说明:

spring boot通常有一个名为 xxxApplication的类,入口类中有一个main方法, 在main方法中使用SpringApplication.run(xxxApplication.class,args)启动springboot应用的项目。

@RestController: 就是@Controller+@ResponseBody组合，支持RESTful访问方 式，返回结果都是json字符串。

@SpringBootApplication 注解等价于: @Configuration 项目启动时自动配置spring 和 springmvc 初始搭建 @EnableAutoConfiguration 自动与项目中集成的第三方技术进行集成 @ComponentScan 扫描入口类所在子包以及子包后代包中注解

\*/ ```

**6. springboot中配置文件的拆分**

```yml

**说明: 在实际开发过程中生产环境和测试环境有可能是不一样的 因此将生产中的配置和测试中的配置拆分开,是非常必要的在springboot中也提供了配置文件拆分的方式. 这里以生产中项名名称不一致为例:**

生产中项目名为: cmfz

测试中项目名为: springboot

端口同时为: 8080

拆分如下: #主配置文件: application.yml #用来书写相同的的配置 server: port: 8080 #生产和测试为同一个端口

#生产配置文件:

application-pord.yml

server:

context-path: /cmfz

#测试配置文件:

application-dev.yml

server:

context-path: /springboot

```