今日任务完成9

SpringBoot:

实现步骤：1.创建Maven项目2.导入SpringBoot起步依赖3.定义Contorller

4.编写引导类5.启动测试

SpringBoot在创建项目时，使用jar的打包方式。

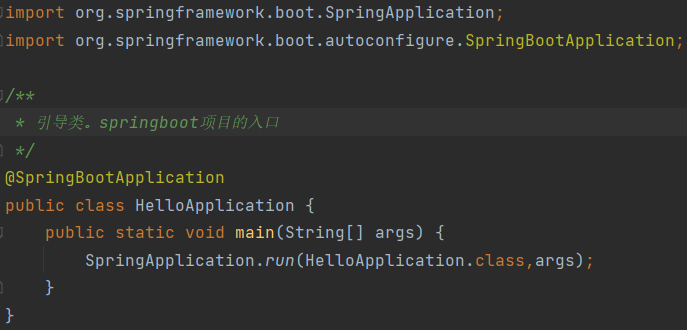
SpringBoot的引导类，是项目入口，运行main方法就可以启动项目

使用SpringBoot和Spring构建的项目，业务代码编写方式完全一样

案例：搭建SpringBoot工程，定义HelloController.hello()方法，返回“Hello Spring Boot!”







访问路径localhost:8080\hello

案例:使用idea快速构建SpringBoot工程。定义HelloController.hello()方法，返回Hello Spring Boot!

SpringBoot起步依赖原理分析

在Spring-boot-start-parent中定义了各种技术的信息，组合了一套最优搭配的技术版本。

在各种starter中，定义了完成该功能需要的坐标集合，其中大部分版本信息来自于父工程。

我的工程继承parent，引入starter后，通过依赖传递，就可以简单方便获的需要的jar包，并且不存在版本冲突等问题。

SpringBoot配置

SpringBoot是基于约定的，所有很多配置都有默认值，但如果想使用自己的配置替换默认配置的话，就可以使用application.properties或者application.yml(application.yaml)进行配置

properties: server.port=8080

yml: server:

prot:8080

SpringBoot提供了2中配置文件类型：properteis和yml/yaml

默认配置文件名称：application

在同一级目录下优先级：properties>yml>yaml

YAML

YAML是一种直观的能够被电脑识别的数据序列化格式，以数据为核心的，比传统的xml方式更加简洁。

YAML文件的扩展名可以使用.yml或者yaml。

YAML:基本语法

1. 大小写敏感
2. 数据值前边必须有空格，作为分隔符
3. 使用缩进表示层级关系
4. 缩进时不允许使用Tab键，只允许使用空格(各个系统Tab对应的空格数目可能不同，导致层次混乱)
5. 缩进的空格数目不重要，只要相同层级的元素左边对齐即可
6. #表示注释，从这个字符一直到行尾，都会被解析器忽略

server:  
 port: 8080  
 address: 127.0.0.1  
name: abc

YAML:数据格式

对象(map):键值对的集合

person:  
 name: zhangsan  
 *#行内写法*person1: {name: zhangsan}

数组：一组按次序排列的值

address:  
 - beijing  
 - shanghai  
 *#行内写法*address1: [bejing,shanghai]

纯量：单个的、不可再分的值

msg1: 'hello \n world' *#单引忽略转义字符*msg2: "hello \n world" *#双引识别转义字符*

YAML：参数引用

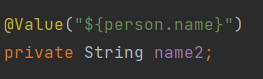
name: lisi  
person:  
 name: ${name} *#引用上边定义的name值*

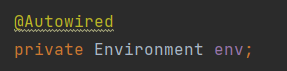
读取配置内容

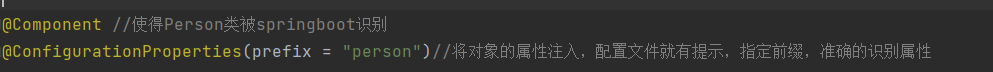
@Value

Environment

@ComfigurationProperties



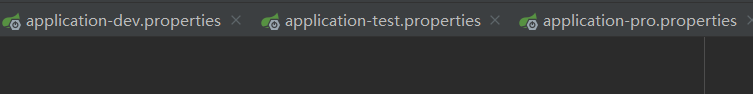


profile

我们在开发SpringBoot应用时，通常同一套程序会被安装到不同环境，比如开发、测试、生产等。其中数据库地址、服务器端口等配置不同，如果每次打包时，都要修改配置文件，那么非常麻烦。profile功能就是来进行动态配置切换的

1. profile配置方式

多profile文件方式



yml多文档方式



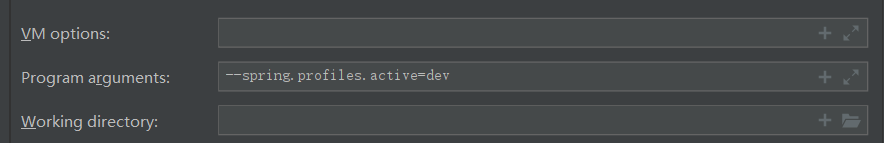
1. profile激活方式

配置文件

虚拟机参数



命令行参数



1. profile是用来完成不同环境下，配置动态功能的
2. profiles配置方式

多profiles文件方式：提供多个配置文件，每个代表一种环境

.application-div.properties/yml开发环境

.application-test.properties/yml测试环境

.application-pro.properties/yml生产环境

yml多文档方式：

.在yml中使用---分隔不同的配置

1. profile激活方式

.配置文件：在配置文件中配置：spring.profiles.active=dev

.虚拟机参数：在VM options指定：-Dspring.profiles.active=dev

.命令行参数：java-jarxxx.jar –spring.profiles.active=div

内部配置加载顺序

Springboot程序启动时，会从以下位置加载配置文件：

1. file：./config/:当前项目下的/config目录下
2. files:./ :当前项目的根目录
3. classpath:/config/:classpath的/config目录
4. classpath:/ :classpath的根目录

加载顺序为上文的排列顺序，高优先级配置的属性会生效