IDEA开发工具

# 修改项目端口号

需要在项目目录 resources-appicaiion.properties文件中声明，server.port=8081

# 接口control注解方式

1、通过localhost:8080/hello访问

@RestController  
public class HelloController {  
 @RequestMapping("/hello")  
 public String hellow() {  
 return "hello world!";  
 }  
}

2、通过localhost:8080/hello/hello访问

@Controller  
@RequestMapping("/hello")  
public class ControllerHello {  
 @GetMapping("/hello")  
 @ResponseBody  
 public String hello() {  
 return "hello world!";  
 }  
}

# SpringBoot 自动配置

pom.xml

spring-boot-dependencies:核心依赖在父工程中

我们再写或者引入一些springBoot一来的时候，不需要指定版本，就是因为有这些版本仓库

启动器

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
</dependency>

启动器说白了就是springBoot的启动场景

比如spring-boot-starter-web，她就会帮我们自动导入web环境的所有毅力啊

Springboot会将所有的功能场景，都变成一个个的启动器

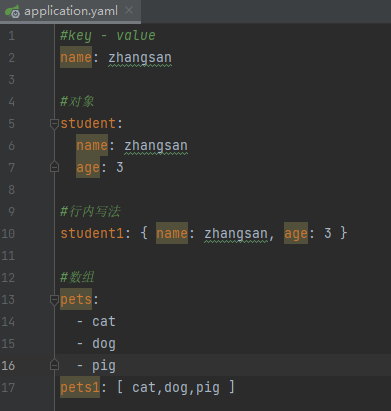
我们要使用什么功能，就只需要找到对应的启动器就可以了 starter

# 配置文件

官方不推荐使用application.properties文件写配置，推荐使用application.yaml

properties只能存键值对

yaml的配置文件可以存对象（可以注入到我们的配置类中，可以直接给实体类赋值）

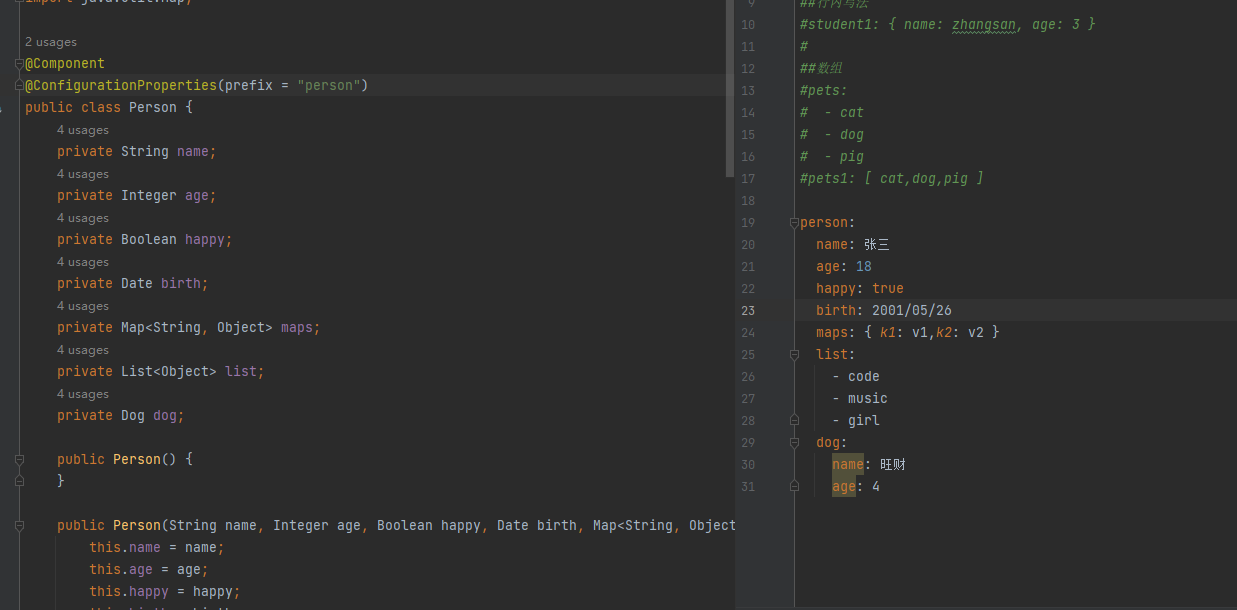


# Component注解

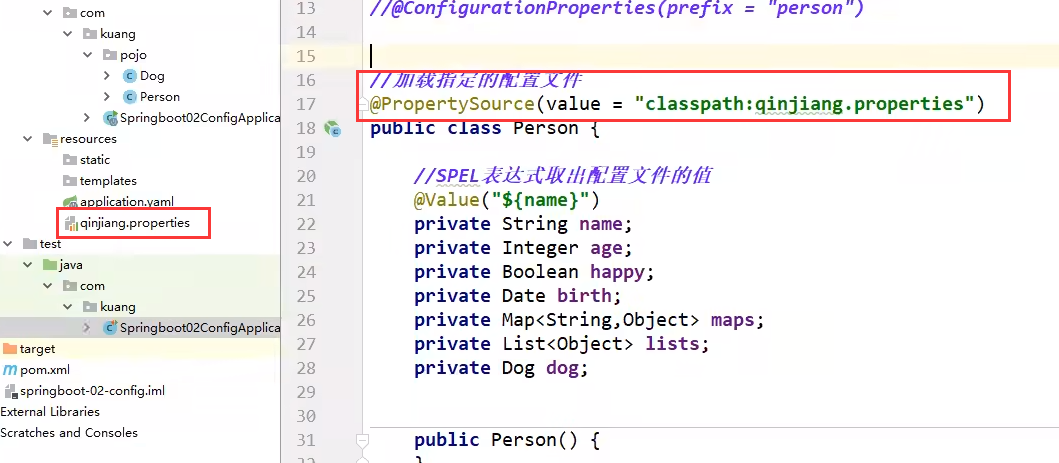
可以给类文件添加@Component注解，把类注册到spring组件中，试其可以被扫描到

变量可以添加@Value注解赋值

# 通过application.yaml给类赋值



加载指定的配置文件



yaml绑定数据支持松散绑定、支持JSR-303格式校验（需要用 @Validated 注解）

# application配置文件可添加的路径

根目录/config

根目录

classpath/config（项目src/main/resources/config）

classpath目录（项目src/main/resources）

四个路径都有配置文件的话，优先级也是上面的从上到下

# 多环境配置

1. properties类配置文件

创建对应环境的properties文件

application.properties

application-dev.properties

application-test.properties

在application.properties里设置当前激活那个配置文件即可

*#springboot的多环境配置：可以选择激活哪一个配置文件*spring:  
 profiles:  
 active: dev

1. yaml类配置文件

所有配置都在一个文件里写

application.yaml

*#springboot的多环境配置：可以选择激活哪一个配置文件*spring:  
 profiles:  
 active: dev  
---  
spring:  
 config:  
 activate:  
 on-profile: dev  
server:  
 port: 8081  
---  
spring:  
 config:  
 activate:  
 on-profile: test  
server:  
 port: 8082

使用---符号可以将当前文件分割为三个文档块，分别写不同配置

# 指定位置加载配置文件

