https://www.cnblogs.com/junyang/p/8151802.html

https://blog.csdn.net/u013248535/article/details/55100979

使用Spring boot ，可以轻松的创建独立运行的程序，非常容易构建独立的服务组件，是实现分布式架构、微服务架构利器。Spring boot简化了第三方包的引用，通过提供的starter，简化了依赖包的配置。

# Spring boot的优点

轻松创建独立的Spring应用程序。

内嵌Tomcat、jetty等web容器，不需要部署WAR文件。

提供一系列的“starter” 来简化的Maven配置。

开箱即用，尽可能自动配置Spring。

SpringBoot集成的插件更多，从而使用很多服务，都只是引入一个依赖，几个注解和Java类就可以用了

在SpringBoot下更常见的形式是将SpringBoot应用打包成可执行jar包文件。之所以这么做，源于你可以直接将SpringBoot应用看成是一个Java Application

# 使用步骤

pom 中引入 spring-boot 依赖，创建一个包含 main 方法的类，main 中 SpringApplication.run() 即可启动 spring boot 项目，在浏览器输入 localhost:8080 即可访问。

## 启动方式

1. Idea创建项目，运行mainfangf
2. mvn spring-boot:run
3. mvn 编译打包项目，java -jar运行项目

## 配置

配置文件使用 .yml 格式，更加简洁清晰

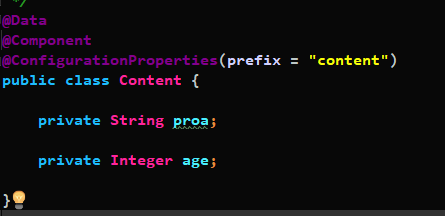
在配置文件中引用配置



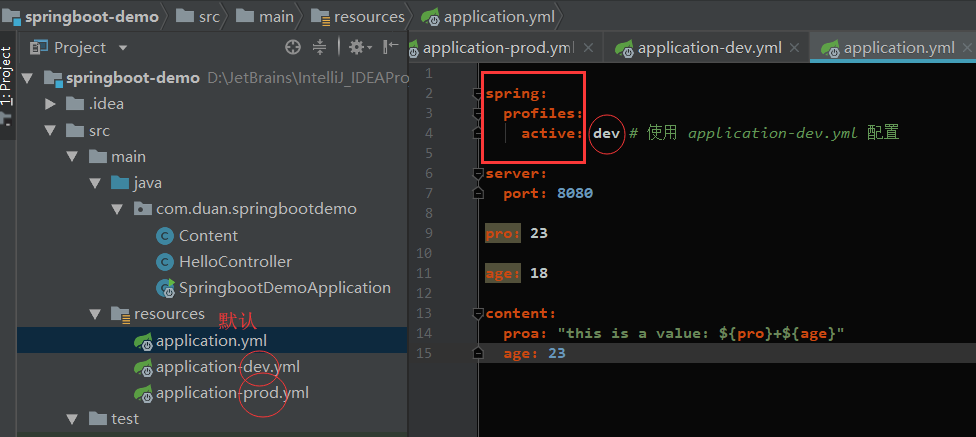
在java代码中通过@Value注解注入配置



将注解绑定到实体类，需要提供setter，getter方法。



在不同环境中使用不同的配置：



# 注解

RequestMapping不限定请求方式时，get和post都能请求，应显示加以限定。

@GetMapping @PostMappting 简化 @RequestMapping 书写方式

# 监听器

## SpringApplicationRunListener

<https://www.cnblogs.com/ning-blogs/archive/2017/03/10/6529710.html>

可用于监听一个 spring boot 项目的 SpringApplication 的 run 方法，从而在 spring boot 项目启动前进行一些操作。

该接口实现了需要在 META-INF/spring.factories 中进行声明，同时实现类要提供一个 public 且接收 SpringApplication 和 String[] 的构造器。

## ApplicationContextInitializer

在 SpringApplication 的 run 方法中会从 META-INF/spring.factories 查找并回调方法