# application.properties配置文件

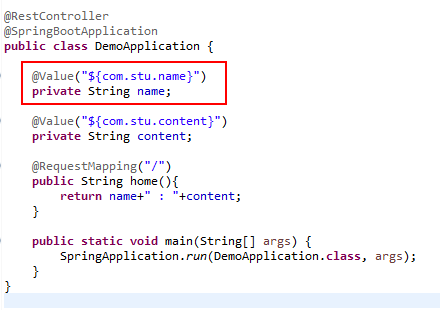
1. 作用：Spring Boot使用了一个全局的配置文件application.properties，放在src/main/resources目录下或者类路径的/config下。Sping Boot的全局配置文件的作用是对一些默认配置的配置值进行修改。
2. 自定义属性（application.properties）

2.1 普通属性配置 – 字符串

(1) 配置

C:\Users\HP\AppData\Local\Temp\1514459335(1).png

(2) 注解@Value将配置的属性和声明的变量绑定



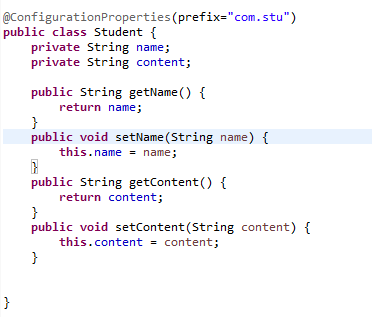
* 1. 配置到Bean对象

(1) 配置

C:\Users\HP\AppData\Local\Temp\1514459335(1).png

(2) 绑定到Bean

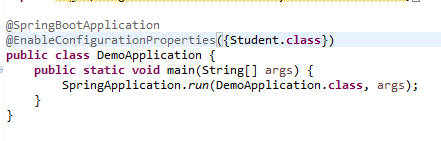
**注解@ConfigurationProperties(prefix=”com.stu”)**



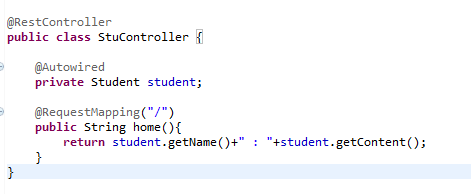
(3) 在main入口类指定加载的Bean

**注解EnableConfigurationProperties({Student.class})**

或 在实体类加上@Configuration

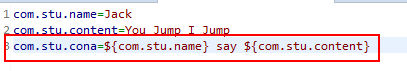


(4) 在Controller注入Bean



3. 其他特性

* 1. 在配置文件中属性间相互调用



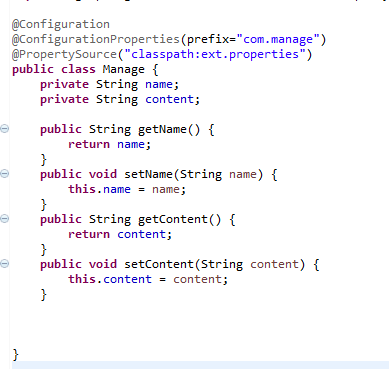
3.2使用自定义配置文件

1. 配置文件ext.properties

C:\Users\HP\AppData\Local\Temp\1514461447(1).png

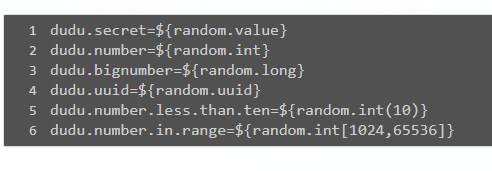
1. 绑定到Bean – 不需要在启动类加载class，已经声明了加载的文件

注解@Configuration @PropertySource(“classpath:ext.properties”)



* 1. 随机值配置

配置文件中${random} 可以用来生成各种不同类型的随机值，从而简化了代码生成的麻烦，例如 生成 int 值、long 值或者 string 字符串。



* 1. 配置文件的优先级

1. src/main/resources/config下application.properties覆盖src/main/resources下application.properties中相同的属性
2. 在相同优先级位置同时有application.properties和application.yml，那么application.properties里的属性里面的属性就会覆盖application.yml。

# 配置文件设置其他特性

1. 项目名配置
2. 1.x版本 server.context-path = / free
3. 2.x 版本 server.servlet.context-path = /free
4. 端口配置
5. server.port = 8084

# 三、 配置文件的加载

## 1. 初始化加载

1.1 默认加载地址

(1) file:/config ：当前jar包加载目录，config文件夹

(2) file: ：当前jar包加载目录

(3) classpath:/config ：jar包解压根目录，config文件夹

(4) classpath: ：jar包解压根目录

1.2 自定义加载地址

(1) 通过命令行参数指定spring.config.location

java -jar （jar包位置）--spring.config.location=（配置文件位置）

1. java -classpath （jar包加载地址，用冒号linux/分号win隔开） 启动类

springboot会自动从该目录加载默认配置文件

## 2. 配置文件的区别

在springboot有两种默认配置文件

1. bootstrap (.yml 或者 .properties)
2. application (.yml 或者 .properties)
   1. bootstrap/ application 的区别
3. Spring Cloud 构建于 Spring Boot 之上，在 Spring Boot 中有两种上下文，一种是 bootstrap, 另外一种是 application, bootstrap 是应用程序的父上下文，也就是说 bootstrap 加载优先于 applicaton。
4. bootstrap 主要用于从额外的资源来加载配置信息，还可以在本地外部配置文件中解密属性。这两个上下文共用一个环境，它是任何Spring应用程序的外部属性的来源。bootstrap 里面的属性会优先加载，它们默认也不能被本地相同配置覆盖。
5. 对比 application 配置文件，boostrap 由父 ApplicationContext 加载，比 applicaton 优先加载，boostrap 里面的属性不能被覆盖。
   1. bootstrap/ application 的应用场景
6. application 配置文件主要用于 Spring Boot 项目的自动化配置。

bootstrap应用场景

1. 使用 Spring Cloud Config 配置中心时，这时需要在 bootstrap 配置文件中添加连接到配置中心的配置属性来加载外部配置中心的配置信息；
2. 一些固定的不能被覆盖的属性
3. 一些加密/解密的场景；