**第二篇：Spring Boot配置文件详解**

参考：<http://blog.csdn.net/zwfmu/article/details/71156261>

转载请标明出处：   
<http://blog.csdn.net/forezp/article/details/70437576>   
本文出自[方志朋的博客](http://blog.csdn.net/forezp)

springboot采纳了建立生产就绪[spring](http://lib.csdn.net/base/javaee)应用程序的观点。 Spring Boot优先于配置的惯例，旨在让您尽快启动和运行。在一般情况下，我们不需要做太多的配置就能够让spring boot正常运行。在一些特殊的情况下，我们需要做修改一些配置，或者需要有自己的配置属性。

## 一、自定义属性

当我们创建一个springboot项目的时候，系统默认会为我们在src/main/[Java](http://lib.csdn.net/base/javase)/resources目录下创建一个application.properties。个人习惯，我会将application.properties改为application.yml文件，两种文件格式都支持。

在application.yml自定义一组属性：

my:

name: forezp

age: 12

如果你需要读取配置文件的值只需要加@Value(“${属性名}”)：

@RestController

public class MiyaController {

@Value("${my.name}")

private String name;

@Value("${my.age}")

private int age;

@RequestMapping(value = "/miya")

public String miya(){

return name+":"+age;

}

}

启动工程，访问：localhost:8080/miya,浏览器显示：

forezp:12

## 二、将配置文件的属性赋给实体类

当我们有很多配置属性的时候，这时我们会把这些属性作为字段来创建一个javabean，并将属性值赋予给他们,比如：

my:

name: forezp

age: 12

number: ${random.int}

uuid : ${random.uuid}

max: ${random.int(10)}

value: ${random.value}

greeting: hi,i'm ${my.name}

其中配置文件中用到了${random} ，它可以用来生成各种不同类型的随机值。

怎么将这些属性赋于给一个javabean 呢，首先创建一个javabean ：

@ConfigurationProperties(prefix = "my")

@Component

public class ConfigBean {

private String name;

private int age;

private int number;

private String uuid;

private int max;

private String value;

private String greeting;

省略了getter setter....

需要加个注解@ConfigurationProperties，并加上它的prrfix。另外@Component可加可不加。另外spring-boot-configuration-processor依赖

org.springframework.boot

spring-boot-configuration-processor

true

另外需要在应用类或者application类，加EnableConfigurationProperties注解。

@RestController

@EnableConfigurationProperties({ConfigBean.class})

public class LucyController {

@Autowired

ConfigBean configBean;

@RequestMapping(value = "/lucy")

public String miya(){

return configBean.getGreeting()+" >>>>"+configBean.getName()+" >>>>"+ configBean.getUuid()+" >>>>"+configBean.getMax();

}

启动工程，访问localhost:8080/lucy,我们会发现配置文件信息读到了。

## 三、自定义配置文件

上面介绍的是我们都把配置文件写到application.yml中。有时我们不愿意把配置都写到application配置文件中，这时需要我们自定义配置文件，比如test.properties:

com.forezp.name=forezp

com.forezp.age=12

怎么将这个配置文件信息赋予给一个javabean呢？

@Configuration

@PropertySource(value = "classpath:test.properties")

@ConfigurationProperties(prefix = "com.forezp")

public class User {

private String name;

private int age;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

}

在最新版本的springboot，需要加这三个注解。

@Configuration   
@PropertySource(value = “classpath:test.properties”)   
@ConfigurationProperties(prefix = “com.forezp”);在1.4版本需要   
PropertySource加上location。

@RestController

@EnableConfigurationProperties({ConfigBean.class,User.class})

public class LucyController {

@Autowired

ConfigBean configBean;

@RequestMapping(value = "/lucy")

public String miya(){

return configBean.getGreeting()+" >>>>"+configBean.getName()+" >>>>"+ configBean.getUuid()+" >>>>"+configBean.getMax();

}

@Autowired

User user;

@RequestMapping(value = "/user")

public String user(){

return user.getName()+user.getAge();

}

}

启动工程，打开localhost:8080/user;浏览器会显示：

forezp12

## 四、多个环境配置文件

在现实的开发环境中，我们需要不同的配置环境；格式为application-{profile}.properties，其中{profile}对应你的环境标识，比如：

* application-test.properties：测试环境
* application-dev.properties：开发环境
* application-prod.properties：生产环境

怎么使用？只需要我们在application.yml中加：

spring:

profiles:

active: dev

其中application-dev.yml:

server:

port: 8082

启动工程，发现程序的端口不再是8080,而是8082。

源码下载：<https://github.com/forezp/SpringBootLearning>

## 五、参考文献

[spring-boot-reference-guide-zh](https://www.gitbook.com/book/qbgbook/spring-boot-reference-guide-zh/details)

[pring Boot干货系列：（二）配置文件解析](http://tengj.top/2017/02/28/springboot2/)

[Spring Boot属性配置文件详解](http://blog.didispace.com/springbootproperties/)