<http://blog.csdn.net/qq_32786873/article/details/52840745>

了解过**[spring](http://lib.csdn.net/base/javaee" \o "Java EE知识库" \t "_blank)**-Boot这个技术的，应该知道Spring-Boot的核心配置文件application.properties，当然也可以通过注解自定义配置文件的信息。

Spring-Boot读取配置文件的方式：

一.读取核心配置文件信息application.properties的内容

     核心配置文件是指在resources根目录下的application.properties或application.yml配置文件，读取这两个配置文件的方法有两种，都比较简单。

核心配置文件application.properties内容如下：

**[java]** [view plain](http://blog.csdn.net/qq_32786873/article/details/52840745) [copy](http://blog.csdn.net/qq_32786873/article/details/52840745)

1. test.msg=Hello World SpringBoot

方式一：使用@Value方式（常用）

**[java]** [view plain](http://blog.csdn.net/qq_32786873/article/details/52840745) [copy](http://blog.csdn.net/qq_32786873/article/details/52840745)

1. <span style="font-size:18px;">**package** Solin.controller;
3. **import** org.springframework.beans.factory.annotation.Value;
4. **import** org.springframework.web.bind.annotation.RequestMapping;
5. **import** org.springframework.web.bind.annotation.RestController;
7. @RestController
8. **public** **class** WebController {
9. @Value("${test.msg}")
10. **private** String msg;
12. @RequestMapping("/index1")
13. **public** String index1(){
14. **return** "方式一:"+msg;
15. }
16. }</span>

注意：在@Value的${}中包含的是核心配置文件中的键名。在Controller类上加@RestController表示将此类中的所有视图都以JSON方式显示，类似于在视图方法上加@ResponseBody。  
访问：http://localhost:8088/index1时得到："方式一:Hello World SpringBoot"

方式二：使用Environment方式

**[java]** [view plain](http://blog.csdn.net/qq_32786873/article/details/52840745) [copy](http://blog.csdn.net/qq_32786873/article/details/52840745)

1. <span style="font-size:18px;">**package** Solin.controller;
3. **import** org.springframework.beans.factory.annotation.Autowired;
4. **import** org.springframework.beans.factory.annotation.Value;
5. **import** org.springframework.core.env.Environment;
6. **import** org.springframework.web.bind.annotation.RequestMapping;
7. **import** org.springframework.web.bind.annotation.RestController;
9. @RestController
10. **public** **class** WebController {
11. @Autowired
12. **private** Environment env;
14. @RequestMapping("/index2")
15. **public** String index2(){
16. **return** "方式二:"+env.getProperty("test.msg");
17. }
18. }
19. </span>

注意：这种方式是依赖注入Evnironment来完成，在创建的成员变量private Environment env上加上@Autowired注解即可完成依赖注入，然后使用env.getProperty("键名")即可读取出对应的值。  
访问：http://localhost:8088/index2时得到："方式二:Hello World SpringBoot"

二.读取自定义配置文件信息，例如：**author**.properties

为了不破坏核心文件的原生态，但又需要有自定义的配置信息存在，一般情况下会选择自定义配置文件来放这些自定义信息，这里在resources目录下创建配置文件author.properties

resources/author.properties内容如下：

**[java]** [view plain](http://blog.csdn.net/qq_32786873/article/details/52840745) [copy](http://blog.csdn.net/qq_32786873/article/details/52840745)

1. author.name=Solin
2. author.age=22

#### 创建管理配置的实体类：

**[java]** [view plain](http://blog.csdn.net/qq_32786873/article/details/52840745) [copy](http://blog.csdn.net/qq_32786873/article/details/52840745)

1. **package** Solin.controller;
3. **import** org.springframework.boot.context.properties.ConfigurationProperties;
4. **import** org.springframework.context.annotation.Configuration;
5. **import** org.springframework.stereotype.Component;
7. //加上注释@Component，可以直接其他地方使用@Autowired来创建其实例对象
8. @Component
9. @ConfigurationProperties(prefix = "author",locations = "classpath:author.properties")
10. **public** **class** MyWebConfig{
11. **private** String name;
12. **private** **int** age;
13. **public** String getName() {
14. **return** name;
15. }
16. **public** **void** setName(String name) {
17. **this**.name = name;
18. }
19. **public** **int** getAge() {
20. **return** age;
21. }
22. **public** **void** setAge(**int** age) {
23. **this**.age = age;
24. }
25. }

注意：  
    在@ConfigurationProperties注释中有两个属性：  
locations：指定配置文件的所在位置  
prefix：指定配置文件中键名称的前缀（我这里配置文件中所有键名都是以author.开头）  
    使用@Component是让该类能够在其他地方被依赖使用，即使用@Autowired注释来创建实例。  
  
创建测试Controller

**[java]** [view plain](http://blog.csdn.net/qq_32786873/article/details/52840745) [copy](http://blog.csdn.net/qq_32786873/article/details/52840745)

1. **package** Solin.controller;
3. **import** org.springframework.beans.factory.annotation.Autowired;
4. **import** org.springframework.stereotype.Controller;
5. **import** org.springframework.web.bind.annotation.RequestMapping;
6. **import** org.springframework.web.bind.annotation.ResponseBody;
8. @Controller
9. **public** **class** ConfigController {
10. @Autowired
11. **private** MyWebConfig conf;
13. @RequestMapping("/test")
14. **public** @ResponseBody String test() {
15. **return** "Name:"+conf.getName()+"---"+"Age:"+conf.getAge();
16. }
17. }

注意：由于在Conf类上加了注释@Component，所以可以直接在这里使用@Autowired来创建其实例对象。

访问：http://localhost:8088/test时得到："Name:Solin---Age:22"