

SpringBoot核心配置与注解

案例2-1全局配置信息获取

一、实验目标

- 1、进一步熟悉IDEA开发环境的使用
- 2、使用Spring Initializr方式构建Spring Boot项目
- 3、掌握Spring Boot全局配置文件与注解的使用
- 4、自定义配置和多环境配置的使用

二、实验环境

Windows操作系统+JDK1.8+Maven3.6+IDEA2019

三、实验要求

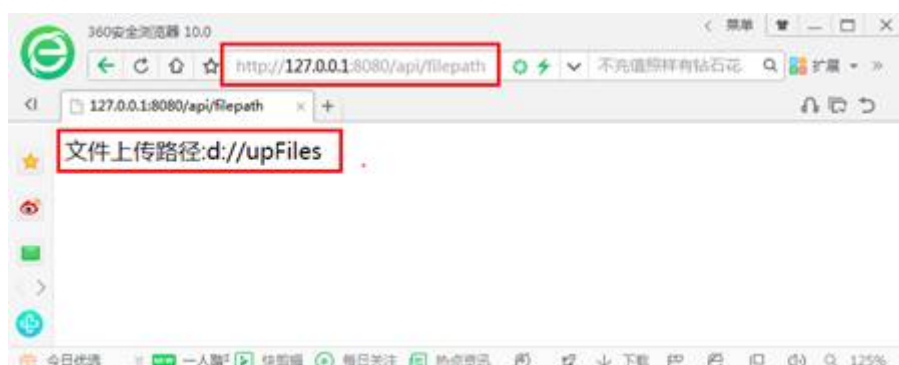
- 1、使用IDEA的Spring Initializr构建Spring Boot项目
- 2、在全局配置文件中设置配置信息如：

文件上传路径（uploadfile.path=d://upfiles）

- 3、在类中获取在全局配置文件设置的信息。



- 4、在浏览器上访问 <http://localhost:8080/api/filepath>时输出显示上面设置的参数信息。



- 5、采用多环境配置的方法配置开发环境和生产环境分别运行在不同的端口和设置不同文件上传路径

四、实验关键步骤

- 1、使用Spring Initializr方式构建Spring Boot项目并选择 spring-boot-starter-web框架。
- 2、创建控制器: ApiController

```
pom.xml x application.properties x ApiController.java x
1 package com.ljxy.ch21.controller;
2
3 import org.springframework.beans.factory.annotation.Value;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RestController;
7
8 @RestController
9 @RequestMapping("/api")
10 public class ApiController {
11     @Value("${uploadfile.path}")
12     String filepath;
13
14     @GetMapping("/test")
15     public String test(){
16         return "test,ok!";
17     }
18     @GetMapping("/filepath")
19     public String filepath(){
20         return "文件上传路径:"+filepath;
21     }
22 }
```

3、使用yaml格式配置

```
ApiController.java x my.properties x application.yaml x
1 uploadfile:
2   path: d:/upfiles2
3
```

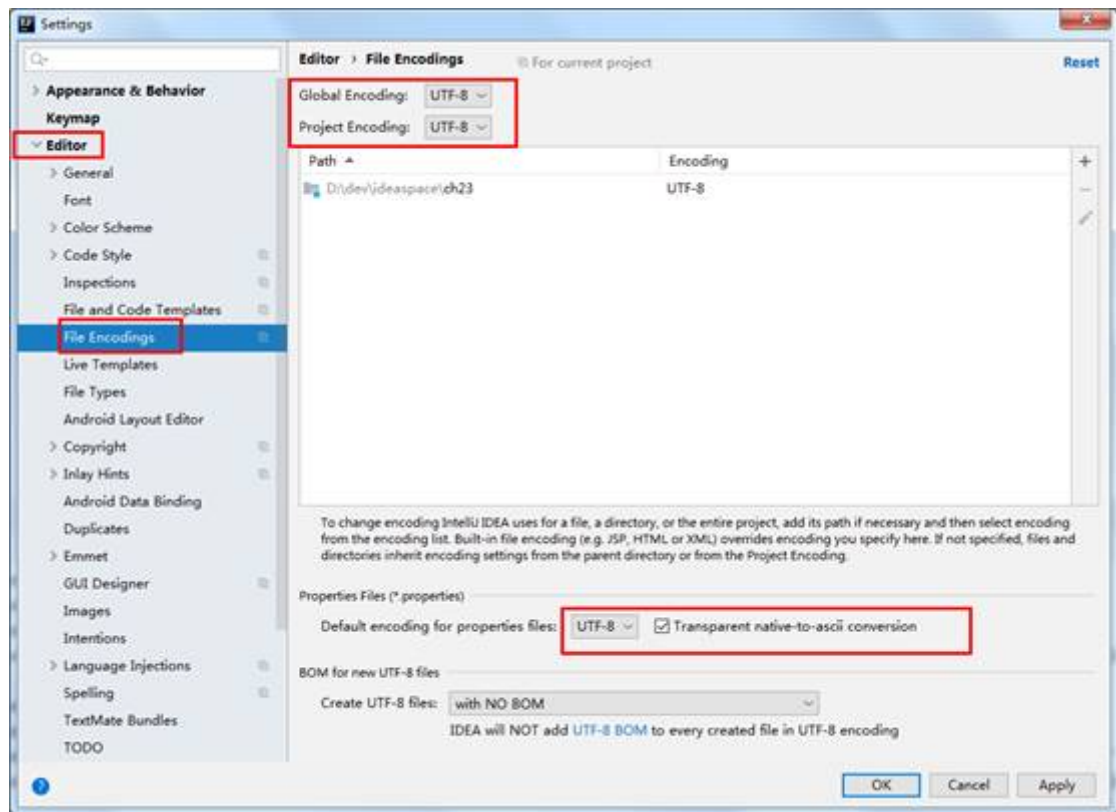
4、使用@PropertySource加载自定义配置文件

```
ApiController.java x my.properties x
1 uploadfile.path2=d:/upfiles2

@RestController
@RequestMapping("/api")
@PropertySource("classpath:my.properties")
public class ApiController {
    @Value("${uploadfile.path}")
    private String filepath;
    @Value("${uploadfile.path2}")
    private String filepath2;

    @GetMapping("/filepath")
    public String filePath(){
        return "上传文件路径："+filepath+",filepath2="+filepath2;
    }
}
```

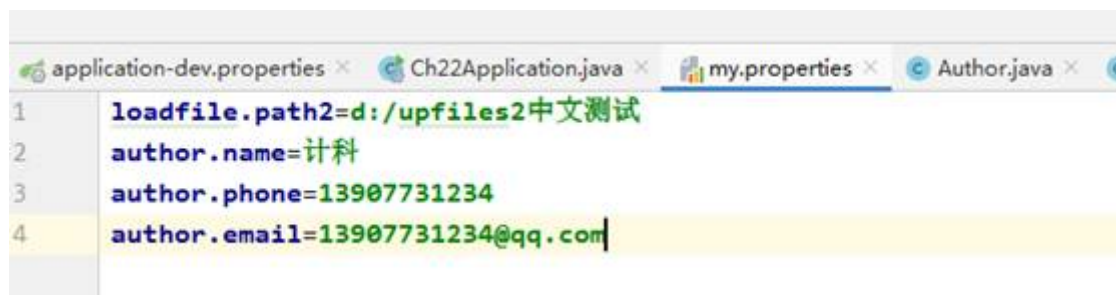
idea开发使用@Value注解注入.properties文件中中文乱码问题解决 @PropertySource(value = "classpath:my.properties",encoding = "UTF-8")



5、多环境配置



6、使用@Configuration指定自定义配置类



```

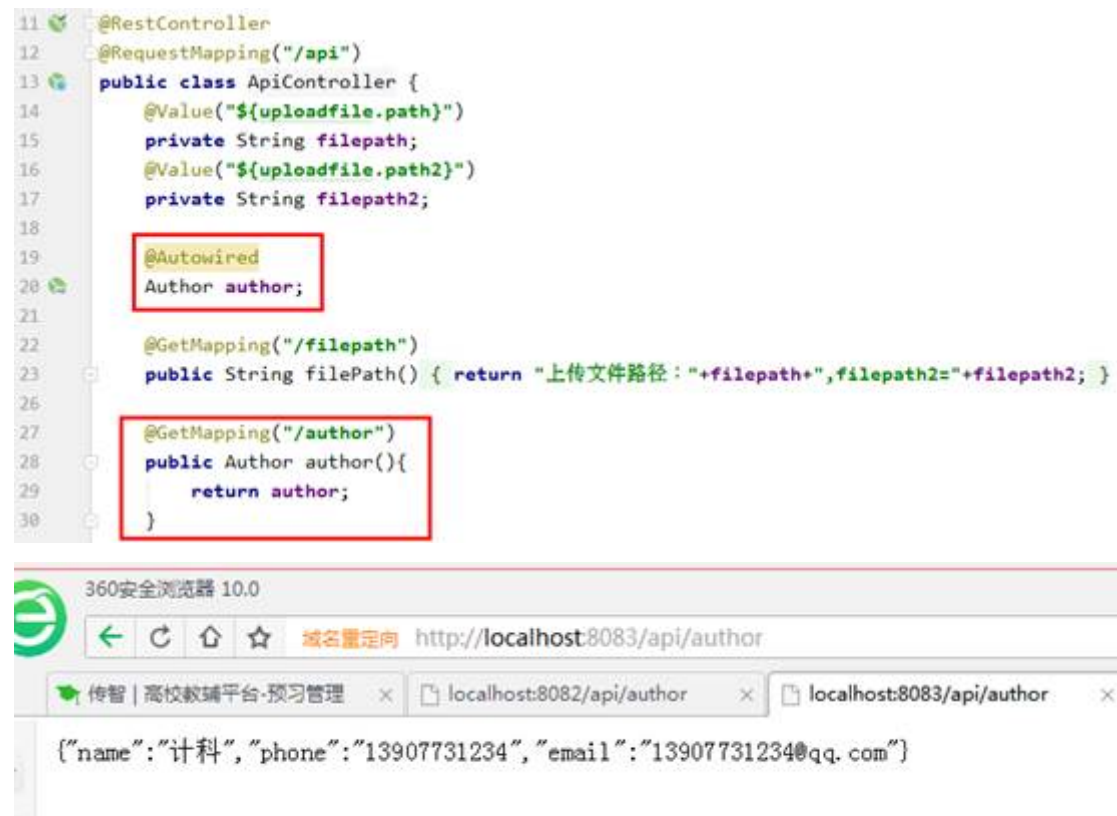
9  @Configuration
10 public class MyConfig {
11     @Value("${author.name}")
12     private String name;
13     @Value("${author.phone}")
14     private String phone;
15     @Value("${author.email}")
16     private String email;
17     @Bean
18     public Author getAuthor(){
19         System.out.println("name = " + name);
20         return new Author(name,phone,email);
21     }
22 }

```

```

3  public class Author {
4      String name;
5      String phone;
6      String email;
7
8      public Author(String name, String phone, String email) {
9          this.name = name;
10         this.phone = phone;
11         this.email = email;
12     }
13
14     public String getName() { return name; }
15
16     public void setName(String name) { this.name = name; }
17
18     public String getPhone() { return phone; }
19
20     public void setPhone(String phone) { this.phone = phone; }
21
22     public String getEmail() { return email; }
23
24     public void setEmail(String email) { this.email = email; }
25
26     @Override
27     public String toString() {

```



五、实验结果及测试效果截图