



链滴

Spring boot 中使用工具类 无需注入获取.yml 中的值

作者: [HuberyYan](#)

原文链接: <https://ld246.com/article/1609384920488>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

转载自: https://blog.csdn.net/xiao____xin/article/details/86700939

之前在一篇文章中写到在spring boot中使用工具类方式获取.yml文件中值的问题 (文章参考: https://blog.csdn.net/xiao____xin/article/details/73274830) , 后续考虑了下还是不太优雅。后来通过看源码发现了新大陆, 即通过: `YamlPropertiesFactoryBean`, 具体实现如下:

1. 新建 `BeanConfiguration` 类, 用于项目启动构造我们的 工具类:

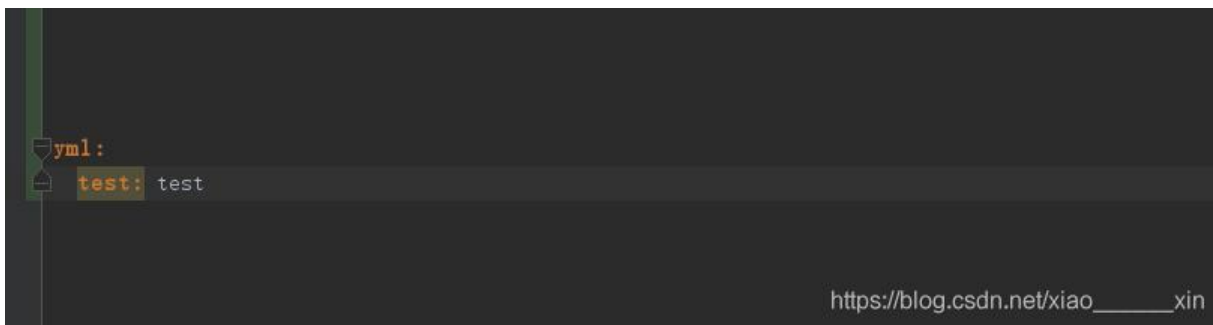
```
@Configurationpublic class BeanConfiguration {    @Bean    public YamlConfigurerUtil yamlConfigurerUtil() {        //1:加载配置文件        Resource app = new ClassPathResource("application.yml");        Resource appDev = new ClassPathResource("application-dev.yml");        Resource appLocalDev = new ClassPathResource("application-prod.yml");        Resource appPre = new ClassPathResource("application-test.yml");        YamlPropertiesFactoryBean yamlPropertiesFactoryBean = new YamlPropertiesFactoryBean();        // 2:将加载的配置文件交给 YamlPropertiesFactoryBean        yamlPropertiesFactoryBean.setResources(app, appDev, appLocalDev, appPre);        // 3        将yaml转换成 key: val        Properties properties = yamlPropertiesFactoryBean.getObject();        // 4: 将Properties 通过构造方法交给我们写的工具类        YamlConfigurerUtil yamlConfigurerUtil = new YamlConfigurerUtil(properties);        return yamlConfigurerUtil;    }}
```

2. 新建我们的工具类 `YamlConfigurerUtil`:

```
public class YamlConfigurerUtil {    private static Properties yamlProperties = new Properties();    public YamlConfigurerUtil(Properties properties){        yamlProperties = properties;    }    public static String getStrYmlVal(String key){        return yamlProperties.getProperty(key);    }    public static Integer getIntegerYmlVal(String key){        return Integer.valueOf(yamlProperties.getProperty(key));    }}
```

接下来我们进行测试:

在yml中新增测试节点



```
@RunWith(SpringRunner.class)@SpringBootTestpublic class ConfigTest {    @Test    public void test() {        String val = YamlConfigurerUtil.getStrYmlVal("yaml.test");        System.out.println(val);    }}
```