



链滴

# spring boot 读取项目参数配置

作者: [wenyl](#)

原文链接: <https://ld246.com/article/1632708760830>

来源网站: [链滴](#)

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



## 1、简介

spring boot项目启动时可以配置一些参数，我们可以通过实现ApplicationRunner或CommandLineRunner接口，并重写run方法来获取这些配置，如果项目内定义了多个类实现了这两个类的话，通过@Order注解来指定他们的执行顺序。

spring boot项目可以在启动的时候指定项目运行参数，eg: java -jar --user=wenyl

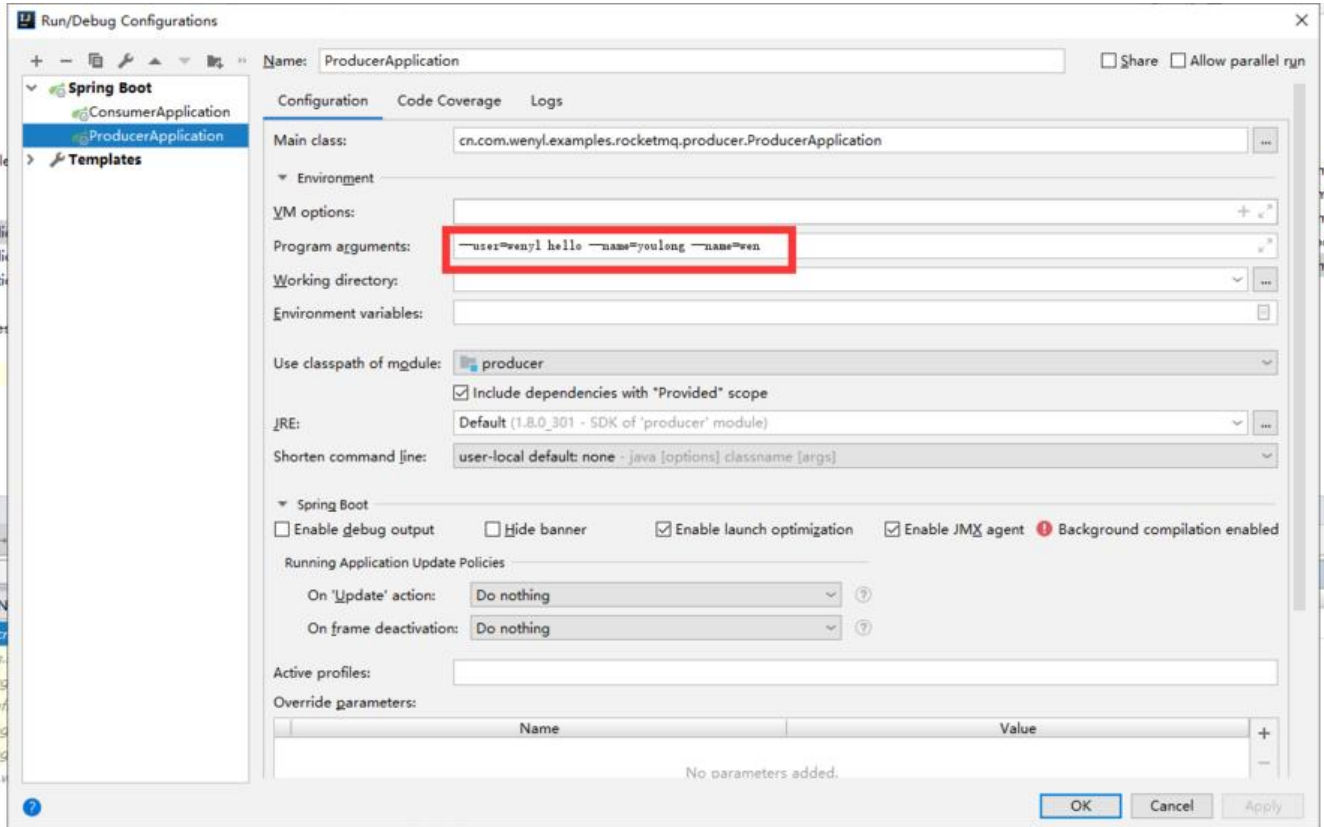
## 2、ApplicationRunner

### 2.1、源码解析

ApplicationRunner的run方法参数是ApplicationArguments,内部共有五个方法

```
public interface ApplicationArguments {  
    String[] getSourceArgs();  
    Set<String> getOptionNames();  
    boolean containsOption(String name);  
    List<String> getOptionValues(String name);  
    List<String> getNonOptionArgs();  
}
```

我们添加启动参数如下



getSourceArgs()以空格为分隔符，获取所有配置，并且以字符串形式返回

```

sourceArgs = {String[4]@6138}
  0 = "--user=wenyl"
  1 = "hello"
  2 = "--name=youlong"
  3 = "--name=wen"

```

getOptionNames()方法用于获取配置项（配置项以--开头）的名称，这个方法返回的值是set集合，为一个配置项的值，可能是数组，需要分开定义但是name都是同一个

```

optionNames = {Collections$UnmodifiableSet@6137} size = 2
  0 = "name"
  1 = "user"

```

getOptionValues()方法用于获取配置项的值，返回的是一个数组，意味着我们可以给同一个配置项义不同的值，如我们添加的name属性，定义了两个值

```

参数-name  值:[youlong, wen]
参数-user  值:[wenyl]

```

getNonOptionArgs()方法用于获取非配置项参数，在测试用的配置中，有个hello参数是没有--前缀获取到的也只有它

```

nonOptionArgs = {Collections$UnmodifiableRandomAccessList@6136} size = 1
  0 = "hello"

```

## 2.2、案例

@Component

```

public class MessageApplicationRunner1 implements ApplicationRunner {

    @Override
    public void run(ApplicationArguments args) throws Exception {
        System.out.println("开始加载");
        List<String> nonOptionArgs = args.getNonOptionArgs();
        Set<String> optionNames = args.getOptionNames();
        for(String optionName:optionNames){
            System.out.println("参数-"+optionName+" 值:"+ args.getOptionValues(optionName))
        }
        String[] sourceArgs = args.getSourceArgs();
        System.out.println("结束加载");
    }
}

```

### 3、CommandLineRunner

CommandLineRunner没有ApplicationRunner那么多操作，只是以空格为分隔，截取配置，2.1中样的配置，获取情况如下

```

args = {String[4]@6138}
> 0 = "--user=wenyl"
> 1 = "hello"
> 2 = "--name=yulong"
> 3 = "--name=wen"

```

```

@Component
public class MessageApplicationRunner implements CommandLineRunner {
    @Override
    public void run(String... args) throws Exception {
        System.out.println(args);
    }
}

```

### 4、ApplicationRunner和CommandLineRunner的区别

ApplicationRunner的run方法的参数是ApplicationArguments，而CommandLineRunner的run方的参数是String类型

```

@FunctionalInterface
public interface ApplicationRunner {

    /**
     * Callback used to run the bean.
     * @param args incoming application arguments
     * @throws Exception on error
     */
    void run(ApplicationArguments args) throws Exception;
}

```

```
@FunctionalInterface
public interface CommandLineRunner {

    /**
     * Callback used to run the bean.
     * @param args incoming main method arguments
     * @throws Exception on error
     */
    void run(String... args) throws Exception;
}
```