https://mp.weixin.qq.com/s/yF-lLZkgZAbJ6zKU3\_9FKw

# 定义系统启动任务的两种方式

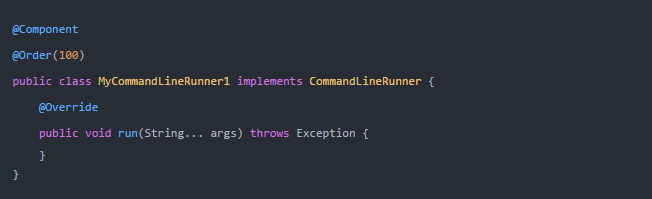
在 Servlet/Jsp 项目中，如果涉及到系统任务，例如在项目启动阶段要做一些数据初始化操作，这些操作有一个共同的特点，只在项目启动时进行，以后都不再执行，这里，容易想到web基础中的三大组件（ Servlet、Filter、Listener ）之一 Listener ，这种情况下，一般定义一个 ServletContextListener，然后就可以监听到项目启动和销毁，进而做出相应的数据初始化和销毁操作，例如下面这样：



当然，这是基础 web 项目的解决方案，如果使用了 Spring Boot，那么我们可以使用更为简便的方式。Spring Boot 中针对系统启动任务提供了两种解决方案，分别是 CommandLineRunner 和 ApplicationRunner，分别来看。

## CommandLineRunner

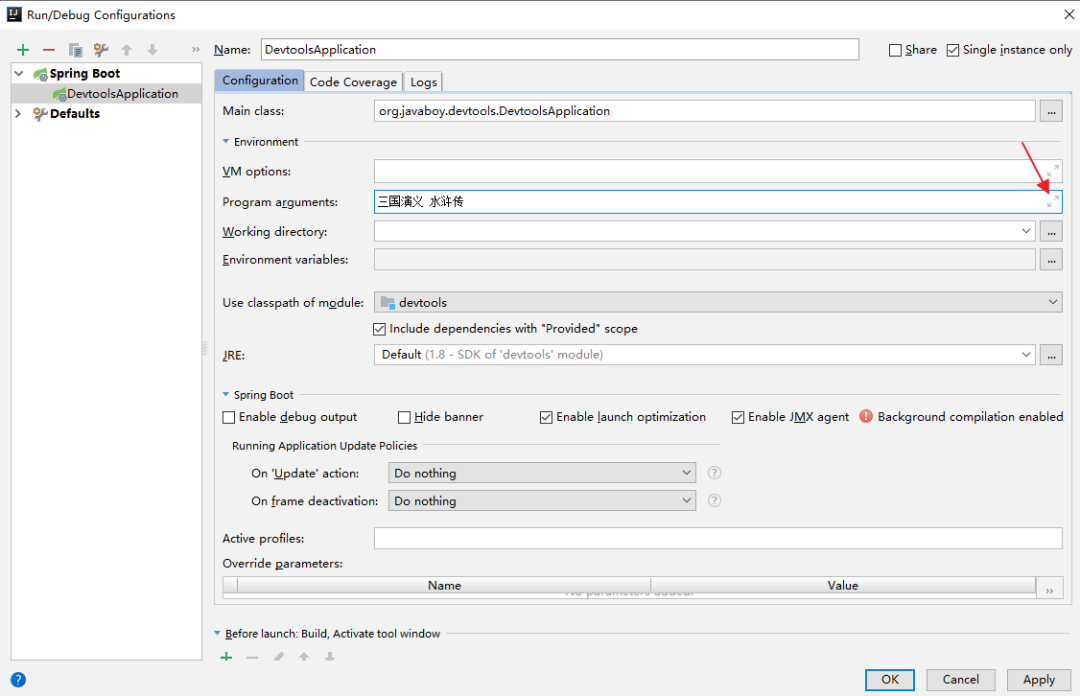
使用 CommandLineRunner 时，首先自定义 MyCommandLineRunner1 并且实现 CommandLineRunner 接口：



关于这段代码，我做如下解释：

1. 首先通过 @Compoent 注解将 MyCommandLineRunner1 注册为Spring容器中的一个 Bean。
2. 添加 @Order注解，表示这个启动任务的执行优先级，因为在一个项目中，启动任务可能有多个，所以需要有一个排序。@Order 注解中，数字越小，优先级越大，默认情况下，优先级的值为 Integer.MAX\_VALUE，表示优先级最低。
3. 在 run 方法中，写启动任务的核心逻辑，当项目启动时，run方法会被自动执行。
4. run 方法的参数，来自于项目的启动参数，即项目入口类中，main方法的参数会被传到这里。

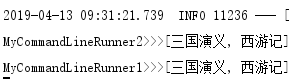
此时启动项目，run方法就会被执行，至于参数，可以通过两种方式来传递，如果是在 IDEA 中，可以通过如下方式来配置参数：



另一种方式，则是将项目打包，在命令行中启动项目，然后启动时在命令行传入参数，如下：

java -jar devtools-0.0.1-SNAPSHOT.jar 三国演义 西游记

注意，这里参数传递时没有 key，直接写 value 即可，执行结果如下：



## ApplicationRunner

ApplicationRunner 和 CommandLineRunner 功能一致，用法也基本一致，唯一的区别主要体现在对参数的处理上，ApplicationRunner 可以接收更多类型的参数（ApplicationRunner 除了可以接收 CommandLineRunner 的参数之外，还可以接收 key/value 形式的参数）。

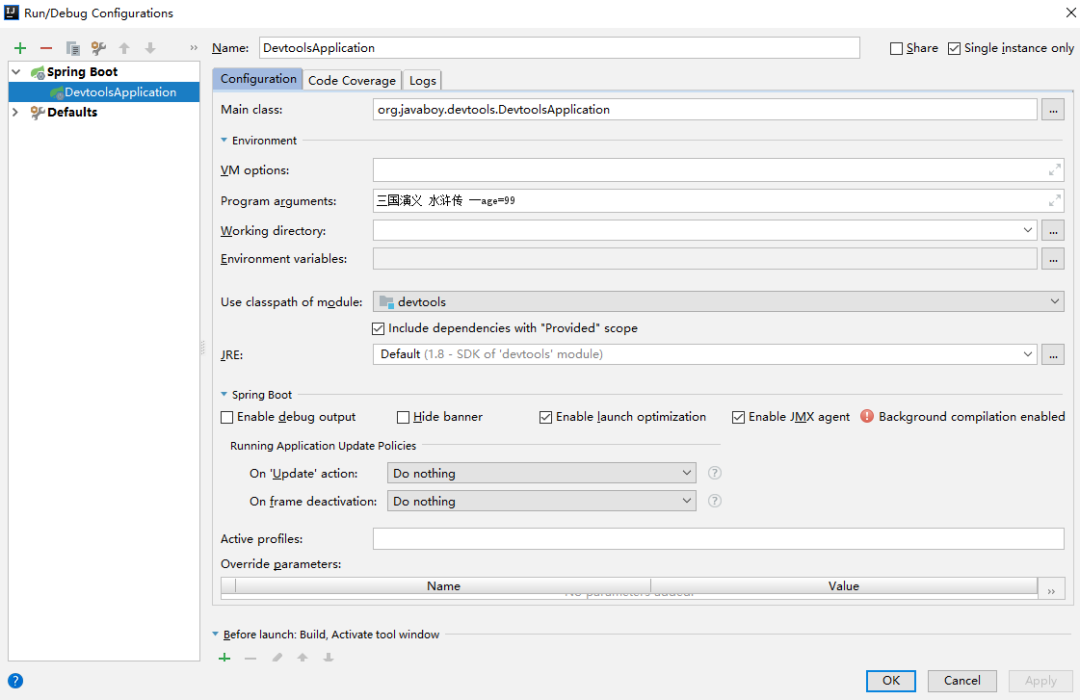
使用 ApplicationRunner ，自定义类实现 ApplicationRunner 接口即可，组件注册以及组件优先级的配置都和 CommandLineRunner 一致，如下：



当项目启动时，这里的 run 方法就会被自动执行，关于 run 方法的参数 ApplicationArguments ，我说如下几点：

1. args.getNonOptionArgs();可以用来获取命令行中的无key参数（和CommandLineRunner一样）。
2. args.getOptionNames();可以用来获取所有key/value形式的参数的key。
3. args.getOptionValues(key));可以根据key获取key/value 形式的参数的value。
4. args.getSourceArgs(); 则表示获取命令行中的所有参数。

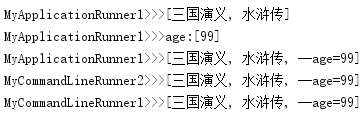
ApplicationRunner 定义完成后，传启动参数也是两种方式，参数类型也有两种，第一种和 CommandLineRunner 一致，第二种则是 --key=value 的形式，在 IDEA 中定义方式如下：



或者使用 如下启动命令：

java -jar devtools-0.0.1-SNAPSHOT.jar 三国演义 西游记 --age=99

运行结果如下：



## 总结

整体来说 ，这两种的用法的差异不大 ，主要体现在对参数的处理上，小伙伴可以根据项目中的实际情况选择合适的解决方案。