SpringBoot之logback-spring.xml不生效

**一、前言**

做新应用就是这样，会遇到各种问题，昨天刚解决了加载某一个类时候抛出了class is not visible from class loader的问题，今天就有遇到了日志文件找不到的问题，还是和二方库有关的，下面就一一道来。

**二、问题产生**

* 正常情况下在 src/main/resources目录放下 logback-spring.xml的配置文件（使用logback日志系统），如下图

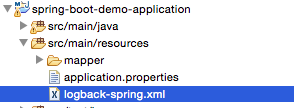


image.png

* application.properties里面设置 spring.application.name=spring-boot-demo-application
* 引入了一个二方包，二方包里面有logback.xml

按照上面配置，运行后正常情况下我们希望在 user.home/spring-boot-demo-application/logs目录应该有applicaiton.log日志文件，然而并没有，连spring-boot-demo-application这个文件夹都没有生成。

**三、问题分析**

那么我们就去看看日志系统是如何查找并解析日志配置文件的，SpringBoot中是使用LoggingApplicationListener这个类来进行日志系统的初始化的。LoggingApplicationListener实现了ApplicationListener接口，那么我们通过时序图看LoggingApplicationListener的onApplicationEvent方法做了啥：

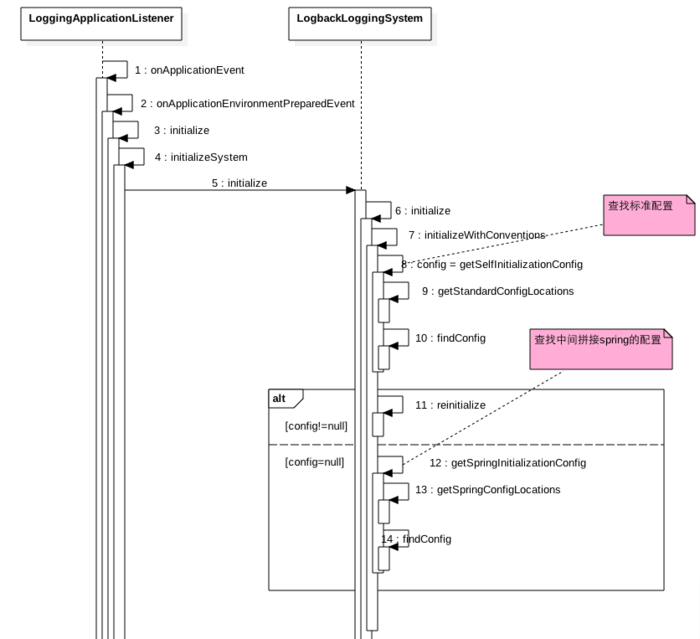


image.png

* 代码（8）查找标准日志配置文件，什么是标准那，那么就看代码（9）的代码：

protected String[] getStandardConfigLocations() {

return new String[] { "logback-test.groovy", "logback-test.xml", "logback.groovy",

"logback.xml" };

}

像 "logback-test.groovy", "logback-test.xml", "logback.groovy","logback.xml"这些是标准的。  
那么具体怎么查找那，要看代码（10):

private String findConfig(String[] locations) {

for (String location : locations) {

ClassPathResource resource = new ClassPathResource(location,

this.classLoader);

if (resource.exists()) {

return "classpath:" + location;

}

}

return null;

}

可知使用ClassPathResource类去查找，下面看ClassPathResource的exists方法：

public boolean exists() {

return (resolveURL() != null);

}

protected URL resolveURL() {

if (this.clazz != null) {

return this.clazz.getResource(this.path);

}

else if (this.classLoader != null) {

return this.classLoader.getResource(this.path);

}

else {

return ClassLoader.getSystemResource(this.path);

}

}

可知是使用 this.classLoader.getResource(this.path);去查找这里classLoader为AppClassloader。

* 如果代码（8）没有查找到配置，则执行点（12），代码12逻辑和代码（8）类似只是查找文件名字不一样，下面看下：

protected String[] getSpringConfigLocations() {

String[] locations = getStandardConfigLocations();

for (int i = 0; i < locations.length; i++) {

String extension = StringUtils.getFilenameExtension(locations[i]);

locations[i] = locations[i].substring(0,

locations[i].length() - extension.length() - 1) + "-spring."

+ extension;

}

return locations;

}

可知是在getStandardConfigLocations的文件名上拼接spring，拼接后的文件名为：  
“` “logback-test-spring.groovy”, “logback-test-spring.xml”, “logback-spring.groovy”,”logback-spring.xml” “

综上所述SpringBoot首先去查找标准的日志配置文件，如果找不到在去找拼接Spring的配置的文件。  
那么上面我们说了应用中是引入了一个含有logback.xml的jar包，而这个jar包也是使用appclassloader加载的，所以在执行步骤（8）的时候找到了jar包里面的logback.xml，所以就不会再去执行步骤（12）来找我们自定义的logback-spring.xml了。

**四、问题解决**

* 方案一，修改我们的配置文件为logback.xml，这样在步骤（8）的时候会首先查找logback.xml，应该是可以找到的。
* 方案二、避免二方包里面含有logback.xml，这种情况下，无论我们自己的配置是logback-spring.xml还是logback.xml都不会有问题。

**五、总结**

日常开发中二方包里面不要带有日志配置文件，二方库中使用日志一般都是使用代码创建的方式。  
欢迎关注微信公众号 ‘技术原始积累’



***原创文章，转载请注明：****转载自[并发编程网 – ifeve.com](http://ifeve.com/)****本文链接地址:****[SpringBoot之logback-spring.xml不生效](http://ifeve.com/36583-2/)*