

# 原创 springBoot完美配置log4j2

2018-03-01 10:19:19 杂七杂八的杂 阅读数 28478 更多

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：[https://blog.csdn.net/V\\_Come\\_On/article/details/79408773](https://blog.csdn.net/V_Come_On/article/details/79408773)

## 1、pom.xml加入log4j2，并同时把spring boot默认的logging去掉

```
1 <dependencies>
2     <dependency>
3         <groupId>org.springframework.boot</groupId>
4         <artifactId>spring-boot-starter-web</artifactId>
5         <exclusions><!-- 去掉默认配置 -->
6             <exclusion>
7                 <groupId>org.springframework.boot</groupId>
8                 <artifactId>spring-boot-starter-logging</artifactId>
9             </exclusion>
10        </exclusions>
11    </dependency>
12    <dependency> <!-- 引入log4j2依赖 -->
13        <groupId>org.springframework.boot</groupId>
14        <artifactId>spring-boot-starter-log4j2</artifactId>
15    </dependency>
16 </dependencies>
```

## 2.在src/main/resource加入log4j2-spring.xml配置文件，取名一定是log4j2-spring，这样省的在application.yml配置

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--设置log4j2的自身log级别为warn-->
3 <!--日志级别以及优先级排序: OFF > FATAL > ERROR > WARN > INFO > DEBUG > TRACE > ALL -->
4 <!--Configuration后面的status, 这个用于设置log4j2自身内部的信息输出, 可以不设置,
5     当设置成trace时, 你会看到log4j2内部各种详细输出-->
6 <!--monitorInterval: Log4j能够自动检测修改配置 文件和重新配置本身, 设置间隔秒数-->
7 <configuration status="warn" monitorInterval="30">
8     <!--先定义所有的appender-->
9     <appenders>
10         <!--这个输出控制台的配置-->
11         <console name="Console" target="SYSTEM_OUT">
12             <!--输出日志的格式-->
13             <PatternLayout pattern="%d{HH:mm:ss:SSS} [%p] - %l - %m%n"/>
14         </console>
15         <!--文件会打印出所有信息, 这个log每次运行程序会自动清空, 由append属性决定, 这个也挺有用的, 适合临时测试用-->
16         <File name="log" fileName="log/test.log" append="false">
17             <PatternLayout pattern="%d{HH:mm:ss.SSS} %-5level %class{36} %L %M - %msg%xEx%n"/>
18         </File>
19         <!-- 这个会打印出所有的info及以下级别的信息, 每次大小超过size,
20             则这size大小的日志会自动存入按年份-月份建立的文件夹下面并进行压缩, 作为存档-->
21         <RollingFile name="RollingFileInfo" fileName="${sys:user.home}/logs/hpaasvc/info.log"
22             filePattern="${sys:user.home}/logs/hpaasvc/${date:yyyy-MM}/info-%d{yyyy-MM-dd}-%i.log">
23             <Filters>
24                 <!--控制台只输出level及以上级别的信息 (onMatch), 其他的直接拒绝 (onMismatch) -->
25                 <ThresholdFilter level="INFO" onMatch="ACCEPT" onMismatch="DENY"/>
26                 <ThresholdFilter level="WARN" onMatch="DENY" onMismatch="NEUTRAL"/>
27             </Filters>
28             <PatternLayout pattern="%d{HH:mm:ss:SSS} [%p] - %l - %m%n"/>
29             <Policies>
30                 <TimeBasedTriggeringPolicy/>
31                 <SizeBasedTriggeringPolicy size="100 MB"/>
32             </Policies>
33         </RollingFile>
34
35         <RollingFile name="RollingFileWarn" fileName="${sys:user.home}/logs/hpaasvc/warn.log"
36             filePattern="${sys:user.home}/logs/hpaasvc/${date:yyyy-MM}/warn-%d{yyyy-MM-dd}-%i.log">
37             <Filters>
38                 <ThresholdFilter level="WARN" onMatch="ACCEPT" onMismatch="DENY"/>
39                 <ThresholdFilter level="ERROR" onMatch="DENY" onMismatch="NEUTRAL"/>
40             </Filters>
```

2

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

8

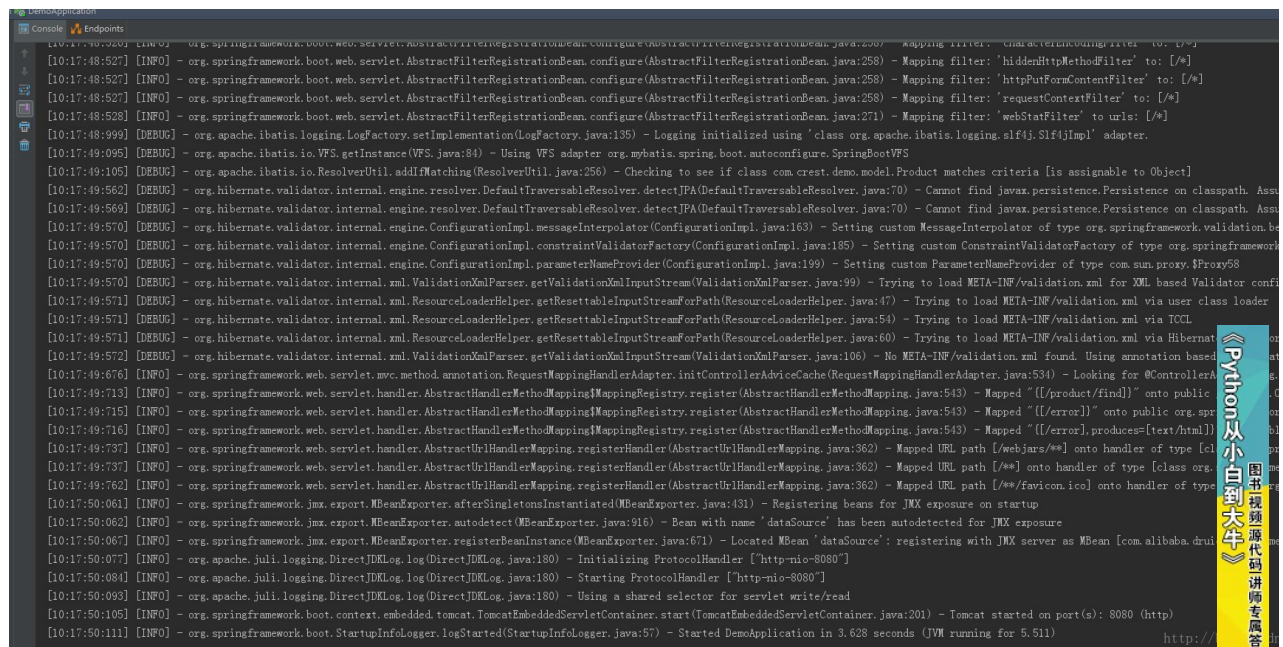
8

8

8

```
41 <PatternLayout pattern="%d{HH:mm:ss:SSS} [%p] - %l - %m%n"/>
42 <Policies>
43 <TimeBasedTriggeringPolicy/>
44 <SizeBasedTriggeringPolicy size="100 MB"/>
45 </Policies>
46 <!-- DefaultRolloverStrategy属性如不设置，则默认为最多同一文件夹下7个文件，这里设置了20 -->
47 <DefaultRolloverStrategy max="20"/>
48 </RollingFile>
49
50 <RollingFile name="RollingFileError" fileName="${sys:user.home}/logs/hpaasvc/error.log"
51     filePattern="${sys:user.home}/logs/hpaasvc/${date:yyyy-MM}/error-%d{yyyy-MM-dd}-%i.log">
52     <ThresholdFilter level="ERROR"/>
53     <PatternLayout pattern="%d{HH:mm:ss:SSS} [%p] - %l - %m%n"/>
54     <Policies>
55         <TimeBasedTriggeringPolicy/>
56         <SizeBasedTriggeringPolicy size="100 MB"/>
57     </Policies>
58 </RollingFile>
59
60 </appenders>
61 <!-- 然后定义logger，只有定义了logger并引入的appender，appender才会生效-->
62 <loggers>
63     <!-- 过滤掉spring和hibernate的一些无用的debug信息-->
64     <logger name="org.springframework" level="INFO"/>
65     </logger>
66     <logger name="org.mybatis" level="INFO"/>
67     </logger>
68     <root level="all">
69         <appender-ref ref="Console"/>
70         <appender-ref ref="RollingFileInfo"/>
71         <appender-ref ref="RollingFileWarn"/>
72         <appender-ref ref="RollingFileError"/>
73     </root>
74 </loggers>
75
76 </configuration>
```

### 3.启动项目，查看日志



The screenshot displays a log window with the following content:

```
[10:17:49:522] [INFO] - org.springframework.boot.web.servlet.AbstractFilterRegistrationBean.configure(AbstractFilterRegistrationBean.java:250) - Mapping filter: 'characterEncodingFilter' to: [/]
[10:17:49:527] [INFO] - org.springframework.boot.web.servlet.AbstractFilterRegistrationBean.configure(AbstractFilterRegistrationBean.java:258) - Mapping filter: 'hiddenHttpMethodFilter' to: [/]
[10:17:49:527] [INFO] - org.springframework.boot.web.servlet.AbstractFilterRegistrationBean.configure(AbstractFilterRegistrationBean.java:258) - Mapping filter: 'httpPutFormContentFilter' to: [/]
[10:17:49:528] [INFO] - org.springframework.boot.web.servlet.AbstractFilterRegistrationBean.configure(AbstractFilterRegistrationBean.java:258) - Mapping filter: 'requestContextFilter' to: [/]
[10:17:49:528] [INFO] - org.springframework.boot.web.servlet.AbstractFilterRegistrationBean.configure(AbstractFilterRegistrationBean.java:271) - Mapping filter: 'webStatFilter' to: urls: [/]
[10:17:49:999] [DEBUG] - org.apache.ibatis.logging.LogFactory.setImplementation(LogFactory.java:135) - Logging initialized using 'class org.apache.ibatis.logging.slf4j.Slf4jImpl' adapter.
[10:17:49:095] [DEBUG] - org.apache.ibatis.io.VFS.getInstance(VFS.java:84) - Using VFS adapter org.mybatis.spring.boot.autoconfigure.SpringBootVFS
[10:17:49:105] [DEBUG] - org.apache.ibatis.io.ResolverUtil.addIfMatching(ResolverUtil.java:256) - Checking to see if class com.crest.demo.model.Product matches criteria [is assignable to Object]
[10:17:49:562] [DEBUG] - org.hibernate.validator.internal.engine.resolver.DefaultTraversableResolver.detectJPA(DefaultTraversableResolver.java:70) - Cannot find javax.persistence.Persistence on classpath. Assuming it is not present.
[10:17:49:569] [DEBUG] - org.hibernate.validator.internal.engine.resolver.DefaultTraversableResolver.detectJPA(DefaultTraversableResolver.java:70) - Cannot find javax.persistence.Persistence on classpath. Assuming it is not present.
[10:17:49:570] [DEBUG] - org.hibernate.validator.internal.engine.ConfigurationImpl.messageInterpolator(ConfigurationImpl.java:163) - Setting custom MessageInterpolator of type org.springframework.validation.beanvalidation.Messages
[10:17:49:570] [DEBUG] - org.hibernate.validator.internal.engine.ConfigurationImpl.constraintValidatorFactory(ConfigurationImpl.java:185) - Setting custom ConstraintValidatorFactory of type org.springframework
[10:17:49:570] [DEBUG] - org.hibernate.validator.internal.engine.ConfigurationImpl.parameterNameProvider(ConfigurationImpl.java:199) - Setting custom ParameterNameProvider of type com.sun.proxy.$Proxy58
[10:17:49:570] [DEBUG] - org.hibernate.validator.internal.xml.ValidationXmlParser.getValidationXmlInputStream(ValidationXmlParser.java:99) - Trying to load META-INF/validation.xml for XML based Validator confi
[10:17:49:571] [DEBUG] - org.hibernate.validator.internal.xml.ResourceLoaderHelper.getResettableInputStreamForPath(ResourceLoaderHelper.java:47) - Trying to load META-INF/validation.xml via user class loader
[10:17:49:571] [DEBUG] - org.hibernate.validator.internal.xml.ResourceLoaderHelper.getResettableInputStreamForPath(ResourceLoaderHelper.java:54) - Trying to load META-INF/validation.xml via TCCL
[10:17:49:571] [DEBUG] - org.hibernate.validator.internal.xml.ResourceLoaderHelper.getResettableInputStreamForPath(ResourceLoaderHelper.java:60) - Trying to load META-INF/validation.xml via Hibernate
[10:17:49:572] [DEBUG] - org.hibernate.validator.internal.xml.ValidationXmlParser.getValidationXmlInputStream(ValidationXmlParser.java:106) - No META-INF/validation.xml found. Using annotation based
[10:17:49:676] [INFO] - org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.initControllerAdviceCache(RequestMappingHandlerAdapter.java:534) - Looking for @ControllerA
[10:17:49:713] [INFO] - org.springframework.web.servlet.handler.AbstractHandlerMethodMapping$MappingRegistry.register(AbstractHandlerMethodMapping.java:543) - Mapped "([/product/find])" onto public
[10:17:49:715] [INFO] - org.springframework.web.servlet.handler.AbstractHandlerMethodMapping$MappingRegistry.register(AbstractHandlerMethodMapping.java:543) - Mapped "([/error])" onto public org.spr
[10:17:49:716] [INFO] - org.springframework.web.servlet.handler.AbstractHandlerMethodMapping$MappingRegistry.register(AbstractHandlerMethodMapping.java:543) - Mapped "([/error], produces=[text/html])"
[10:17:49:737] [INFO] - org.springframework.web.servlet.handler.AbstractUrlHandlerMapping.registerHandler(AbstractUrlHandlerMapping.java:362) - Mapped URL path [/webjars/**] onto handler of type [cl
[10:17:49:737] [INFO] - org.springframework.web.servlet.handler.AbstractUrlHandlerMapping.registerHandler(AbstractUrlHandlerMapping.java:362) - Mapped URL path [/**] onto handler of type [class org
[10:17:49:762] [INFO] - org.springframework.web.servlet.handler.AbstractUrlHandlerMapping.registerHandler(AbstractUrlHandlerMapping.java:362) - Mapped URL path [/**/favicon.ico] onto handler of type
[10:17:50:061] [INFO] - org.springframework.jmx.export.MBeanExporter.afterSingletonsInstantiated(MBeanExporter.java:431) - Registering beans for JMX exposure on startup
[10:17:50:062] [INFO] - org.springframework.jmx.export.MBeanExporter.autodetect(MBeanExporter.java:916) - Bean with name 'dataSource' has been autodetected for JMX exposure
[10:17:50:067] [INFO] - org.springframework.jmx.export.MBeanExporter.registerBeanInstance(MBeanExporter.java:671) - Located MBean 'dataSource': registering with JMX server as MBean [com.alibaba.druid
[10:17:50:077] [INFO] - org.apache.juli.logging.DirectJDKLog.log(DirectJDKLog.java:180) - Initializing ProtocolHandler ["http-nio-8080"]
[10:17:50:084] [INFO] - org.apache.juli.logging.DirectJDKLog.log(DirectJDKLog.java:180) - Starting ProtocolHandler ["http-nio-8080"]
[10:17:50:093] [INFO] - org.apache.juli.logging.DirectJDKLog.log(DirectJDKLog.java:180) - Using a shared selector for servlet write/read
[10:17:50:105] [INFO] - org.springframework.boot.context.embedded.tomcat.TomcatEmbeddedServletContainer.start(TomcatEmbeddedServletContainer.java:201) - Tomcat started on port(s): 8080 (http)
[10:17:50:111] [INFO] - org.springframework.boot.StartupInfoLogger.logStarted(StartupInfoLogger.java:57) - Started DemoApplication in 3.628 seconds (JVM running for 5.511)
```

文章最后发布于: 2018

有 0 个人打赏

### SpringBoot 项目中使用Log4j2详细（避坑）

阅读量 2万+

以下大部分内容转载整理自 <https://blog.csdn.net/vbirdbest/article/details/71751835>，感谢 vbirdbest 的相关... 博文 来自: RyanDon的博客