



大家都在搜...



下载APP

开源软件

问答

动弹

博客

协作翻译 &gt; Web/WAP应用开发 &gt; 正文

## 使用 Spring Boot 配置日志

已翻译 100%

👍 顶

2

oschina 投递于 2017/09/09 08:16 (共 19 段, 翻译完成于 09-19) 阅读 2463 收藏 130

💬 评论 3

英文原文: Configuring Logback With Spring Boot

## 亿速云香港服务器CN2高速仅29元

亿速云, 前10强游戏企业在都用的云服务器性能稳定速度快免备案24小时售后在线 亿速云

参与翻译 (6人): 亚林瓜子, Tocy, vampire88, 回忆悲哀烦恼, 风马少年, aaronday

🖨 打印

仅中文

中英文对照

仅英文

继之前的设置了 logging.path 的 application.properties 配置片段, 如果没有其他配置, 会导致将日志输出到文件 (以及控制台)。因此, 您可以按那样做, 但这样也会导致写入的文件和文件名不受您的控制。下面的示例将演示与上述 SAVE-TO-FILE

您还没有登录, 点击[此处登录](#) ✕

```
logging.pattern.console=
logging.path=logs
logging.file=${logging.path}/log.log
logging.pattern.file=%d{dd-MM-yyyy HH:mm:ss.SSS} [%thread] %-5level %logger{36}.%M - %msg%n
```

这和 XML 配置有所不同的是, 该示例展示了在 MyServiceImpl 的日志记录器中引用的 appender, 但上述的 application.properties 代码片段还包括将所有日志消息输出到文件的根日志记录器。

logging.pattern.console 被添加来阻止输出日志到控制台, 以使其与上述的 XML 配置保持一致 (这似乎不是一个很好的方式, 但是我还没有看到另一个更好的解决方案)。

**亚林瓜子**翻译于 2017/09/13  
09:01

👍 顶

0

RollingFileAppender 将根据日志滚动策略将日志保存到不同的文件。它允许将日志输出分割成您定制的各种形式, 这样做会方便很多。例如, 您可以根据日期分隔日志文件, 以便查看过去在特定日期发生



大家都在搜...



下载APP

开源软件

问答

动弹

博客

TimeBasedRollingPolicy 将根据日期创建一个新的文件。以下配置将每天创建一个新文件，并使用 %d 符号将日期附加到日志文件的名称上。 %d 符号的格式很重要，因为要根据它来推断日志滚动的时间段。下面的示例是每天滚动，如果要每月滚动，可以使用 %d {MM-yyyy} 模式来包含日期部分。只要 %d 符号中的格式符合 SimpleDateFormat 类允许的格式，就可以使用不同的日志滚动周期 - 不局限于每天或每月。

```
<appender name="SAVE-TO-FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>${LOG_PATH}/log.log</file>
  <encoder class="ch.qos.logback.classic.encoder.PatternLayoutEncoder">
    <Pattern>%d{dd-MM-yyyy HH:mm:ss.SSS} [%thread] %-5level %logger{36}.%M - %msg%n</Pattern>
  </encoder>
  <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <fileNamePattern>${LOG_PATH}/archived/log_%d{dd-MM-yyyy}.log</fileNamePattern>
    <maxHistory>10</maxHistory>
    <totalSizeCap>100MB</totalSizeCap>
  </rollingPolicy>
</appender>
```

**亚林瓜子**翻译于 2017/09/14  
08:23

👍 顶

0

您还没有登录，[点击此处登录](#) ✕

我已经在上面的例子中包含了 TimeBasedRollingPolicy 可用的一些属性。maxHistory 用于设置归档日志文件在被自动删除之前将被保留多久。保留的时间取决于文件名中指定的回转时间，因此在上述示例中，回转周期为一天，允许在删除之前存储最多 10 天的归档日志。totalSizeCap 用于限制所有归档日志文件的最大大小。它需要使用 maxHistory 锁设置的 maxHistory 属性，在删除归档文件时优先级高于 totalSizeCap。

此配置超出了在 application.properties 文件内就可以完成的范围。对于以下示例也可以这样说，尽管默认配置将允许日志文件在达到 10 MB 时进行回转，并支持最多 7 个归档日志文件。

**Tocy**翻译于 2017/09/10  
10:21

👍 顶

0





大家都在搜...



下载APP

开源软件

问答

动弹

博客

此策略，我并没有保存它们，因为较小的文件大小使得分离的日志更容易遍历。

```
<appender name="SAVE-TO-FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>${LOG_PATH}/log.log</file>
  <encoder class="ch.qos.logback.classic.encoder.PatternLayoutEncoder">
    <Pattern>%d{dd-MM-yyyy HH:mm:ss.SSS} [%thread] %-5level %logger{36}.%M - %msg%n</Pattern>
  </encoder>
  <rollingPolicy class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
    <fileNamePattern>${LOG_PATH}/log_%i.log</fileNamePattern>
    <minIndex>2</minIndex>
    <maxIndex>3</maxIndex>
  </rollingPolicy>
  <triggeringPolicy class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
    <maxFileSize>1KB</maxFileSize>
  </triggeringPolicy>
</appender>
```

在 FixedWindowRollingPolicy 中找到的 minIndex 和 maxIndex 的可选属性指定了在日志文件名中可以使用的%i的最小值和最大值。因此，在上述示例中，当日志滚动时，它们可以使用名称 log\_2.log 和 log\_3.log（尽管从2开始是奇怪的，[您还没有登录，点击此处登录](#) 台）。生成日志文件的过程如下（以上述代码片段为例）：

- log.log 文件达到最大文件大小时--> log.log 重命名为 log\_2.log 并新生成一个 log.log 文件
- log\_2.log 文件达到最大文件大小时-->log\_2.log 重命名为 log\_3.log, log.log 重命名为 log\_2.log 并生成一个新的 log.log 文件
- log\_3.log 文件达到最大文件大小时-->log\_3.log 删除, log\_2.log 重命名为 log\_3.log, log.log 重命名为 log\_2.log 并新生成一个 log.log 文件

A

**aaronday**翻译于 2017/09/14  
17:44

👍 顶

0

如果我仍然没有做好给你解释这个流程的工作，那么看看 [FixedWindowRollingPolicy](#) 文档，在我没解释清楚的情况下，希望它能让你理解清楚。





大家都在搜....



下载APP

开源软件

问答

动弹

博客

```
<appender name="SAVE-TO-FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>${LOG_PATH}/log.log</file>
  <encoder class="ch.qos.logback.classic.encoder.PatternLayoutEncoder">
    <Pattern>%d{dd-MM-yyyy HH:mm:ss.SSS} [%thread] %-5level %logger{36}.%M - %msg%n</Pattern>
  </encoder>
  <rollingPolicy class="ch.qos.logback.core.rolling.SizeAndTimeBasedRollingPolicy">
    <fileNamePattern>${LOG_PATH}/archived/log_%d{dd-MM-yyyy}_%i.log</fileNamePattern>
    <maxFileSize>10MB</maxFileSize>
    <maxHistory>10</maxHistory>
    <totalSizeCap>100MB</totalSizeCap>
  </rollingPolicy>
</appender>
```

如你所看到的，它包含 `maxFileSize`、`maxHistory` 和 `totalSizeCap`，可以控制单个文件以及文件集合的大小。因此，上述示例将把 10 天的历史记录分割为 10MB 的文件，当所有文件的总大小达到 100MB 时，最旧的文件将被删除。

**Tocy**翻译于 2017/09/10  
10:46

👍 顶

0

您还没有登录，[点击此处登录](#) ✕

现在我们已经看过如何定义可以输出到控制台或文件的多个 `appender`，我们可以将它们组合一次输出到两种格式，只需在 `logger` 中引用多个 `appender` 即可：

```
<logger name="com.lankydan.service.MyServiceImpl" additivity="false" level="debug">
  <appender-ref ref="STDOUT" />
  <appender-ref ref="SAVE-TO-FILE" />
</logger>
```

所以现在这个 `logger` 将输出到控制台，使用 `STDOUT` 以及 `SAVE-TO-FILE` `appender` 输出到文件。

可以通过 `application.properties` 实现类似的配置。如果你回到该页面，你可能可以弄清楚自己要如何配置，因为上一个例子有一个额外添加的行，以阻止它输出到控制台和文件中。同样，这将包含来自根 `logger` 的日志消息，而不仅仅像 `MyServiceImpl` 上面的代码段一样。







大家都在搜....



下载APP

开源软件

问答

动弹

博客

在 application.properties 设置或作为环境变量设置：

```
spring.profiles.active=dev
```

或作为 VM 选项设置：

```
-Dspring.profiles.active=dev
```

现在当应用程序运行时，将使用用于 dev 的 springProfile，导致日志输出到控制台和文件。如果这被推送到生产环境中，则该属性需要设置为 prod，这会将配置匹配合适的位置，例如仅将日志写入文件，并可能更改所有或某些类/包的日志记录级别。

也可以通过 application.properties 来提供类似的配置。那么，实际上并不是 application.properties，而是 application-dev.properties 和 application-prod.properties - 每个环境的单独的配置文件。遵循 application-{environment}.properties 的命名约定，其中 {environment} 替换为环境名称。根据您的 VM 选项或环境变量，可以选择其中之一。您还没有登录，[点击此处登录](#) 的 springProfile 配置一样。以下是上面描述片段的等效配置。

application-dev.properties:

```
logging.level.root=info
logging.level.com.lankydan.service=debug
logging.path=logs
logging.file=${logging.path}/log.log
logging.pattern.file=%d{dd-MM-yyyy HH:mm:ss.SSS} [%thread] %-5level %logger{36}.%M - %msg%n
logging.pattern.console=%d{dd-MM-yyyy HH:mm:ss.SSS} %magenta([%thread]) %highlight(%-5level) %logger.%M - %msg%n
```

application-prod.properties:

```
logging.level.root=info
logging.level.com.lankydan.service=error
logging.path=logs
logging.file=${logging.path}/log.log
logging.pattern.file=%d{dd-MM-yyyy HH:mm:ss.SSS} [%thread] %-5level %logger{36}.%M - %msg%n
logging.pattern.console=
```





大家都在搜...



下载APP

开源软件

问答

动弹

博客

08:44

👍 顶

0

我想我应该把这篇文章发布出来，虽然篇幅比我预想中的还要长。此外，还有很多可以使用 Logback 和 Spring Boot 来完成的工作，我在这里并没有涉及到。从本教程总结来看，你应该已经掌握了如何使用 Spring Boot 中的 Logback，包括如何使用属性文件来更改 Spring Boot 所提供的默认设置，以及如何进一步使用 Logback（通过 logback.xml 和 logback-spring.xml）创建自定义配置。你还应该了解了属性文件所提供的配置中的限制，以便你知道何时可以直接切换到使用 Logback 来实现你的目标。

这些示例中使用的代码可以在我的 [GitHub](#) 上找到。

**Tocy**翻译于 2017/09/10  
11:49

👍 顶

0

您还没有登录，[点击此处登录](#) ✕

本文中的所有译文仅用于学习和交流目的，转载请务必注明文章译者、出处、和本文链接。  
我们的翻译工作遵照 [CC 协议](#)，如果我们的工作有侵犯到您的权益，请及时联系我们。

## 评论(3)

**jaychang** 2017/10/06 09:31

翻译不错

🚫 举报

**哈哈爱兮爱兮乎乎** 2017/10/05 16:36

看不懂。

🚫 举报

