

[logback](#)

博客分类：

- [logback](#)

[log4j](#) [iBATIS](#) [DAO](#) [JMS](#) [ActiveMQ](#)

新的公司日志管理统一用logback，由于之前一直在用log4j，所以其实对logback这个由log4j长出来的家伙还不算陌生。

首先介绍下logback的前世今生。

slf4j由log4j作者Ceki开发，逐步取代apache commons logging。

logback由log4j作者Ceki开发，逐步取代log4j。

slf4j等于commons-logging，是各种日志实现的通用入口，会根据classpath中存在下面哪一个Jar来决定具体的日志实现库。

logback相比较log4j的优势

slf4j支持参数化的logger.error("帐号ID: {}不存在", userId);

告别了if(logger.isDebugEnabled()) 时代。

另外logback的整体性能比log4j也较佳，hibernate等项目已经采用了slf4j。

slf4j和logback的使用

- 1.如果日志的参数超过3个，需要写成

Java代码



```
1. Object[] params = {newVal, below, above};
2. logger.debug("Value {} was inserted between {} and {}. ",
params);
```

- 2.因为内部已优化，作者认为slf4j的logger不需要定义为static。

- 3.可设置缓存后批量写日志文件(但服务器如果重启，可能会丢失未写到磁盘的记录)

- 4.MDC，用Filter，将当前用户名等业务信息放入MDC中，在日志format定义中即可使用该变量。

- 5.JMS Appender用于告警，DB Appender用于业务日志等可以使用插件，如生成Log代码的Eclipse插件Log4E。

6.tomcat和glassfish中，设定日志路径为../logs/xxxx.log 都能将日志放入应用服务器本身的logs目录。

最后把最近完善的一个logback.xml贴上，毕竟实际项目中的文件最能说明问题。

Xml代码



```
1. <?xml version="1.0" encoding="UTF-8"?>
2.
3. <configuration>
4.     <substitutionProperty name="log.base"
value="d:\\logback\\logback" />
5.     <jmxConfigurator />
6.     <appender name="stdout"
class="ch.qos.logback.core.ConsoleAppender">
7.         <filter
class="ch.qos.logback.core.filter.EvaluatorFilter">
8.             <evaluator name="myEval">
9.                 <expression>message.contains("dao")
</expression>
10.            </evaluator>
11.            <onMatch>ACCEPT</onMatch>
12.            <onMismatch>DENY</onMismatch>
13.        </filter>
14.        <layout class="ch.qos.logback.classic.PatternLayout">
15.            <pattern>%date [%thread] %-5level %logger{80} -
%msg%n</pattern>
16.        </layout>
17.    </appender>
18.    <appender name="logfile-dao"
class="ch.qos.logback.core.rolling.RollingFileAppender">
19.        <filter
class="ch.qos.logback.core.filter.EvaluatorFilter">
20.            <evaluator name="myEval_dao">
21.                <expression>message.contains("dao")
</expression>
22.            </evaluator>
23.            <onMatch>ACCEPT</onMatch>
24.            <onMismatch>DENY</onMismatch>
25.        </filter>
26.        <Encoding>UTF-8</Encoding>
27.
```

```

28.         <File>${log.base}_dao.log</File>
29.         <rollingPolicy
class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
30.             <FileNamePattern>${log.base}.%d{yyyy-MM-
dd}_dao.log.zip
31.         </FileNamePattern>
32.         </rollingPolicy>
33.         <layout class="ch.qos.logback.classic.PatternLayout">
34.             <pattern>%date [%thread] %-5level %logger{80} -
%msg%n</pattern>
35.         </layout>
36.     </appender>
37.     <appender name="logfile-service"
38.
class="ch.qos.logback.core.rolling.RollingFileAppender">
39.         <filter
class="ch.qos.logback.core.filter.EvaluatorFilter">
40.             <evaluator name="myEval_service">
41.                 <expression>message.contains("service.impl")
</expression>
42.             </evaluator>
43.             <onMatch>ACCEPT</onMatch>
44.             <onMismatch>DENY</onMismatch>
45.         </filter>
46.         <Encoding>UTF-8</Encoding>
47.         <File>${log.base}_service.log</File>
48.         <rollingPolicy
class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
49.             <FileNamePattern>${log.base}.%d{yyyy-MM-
dd}_service.log.zip
50.         </FileNamePattern>
51.         </rollingPolicy>
52.         <layout class="ch.qos.logback.classic.PatternLayout">
53.             <pattern>%date [%thread] %-5level %logger{80} -
%msg%n</pattern>
54.         </layout>
55.     </appender>
56.     <appender name="jms_dao"
class="ch.qos.logback.classic.net.JMSQueueAppender">
57.         <InitialContextFactoryName>
58.
org.apache.activemq.jndi.ActiveMQInitialContextFactory
59.     </InitialContextFactoryName>
60.         <ProviderURL>tcp://192.168.1.120:61616</ProviderURL>
61.         <filter

```

```

class="ch.qos.logback.core.filter.EvaluatorFilter">
62.     <evaluator name="myEval_service">
63.         <expression>message.contains("dao")
</expression>
64.     </evaluator>
65.     <onMatch>ACCEPT</onMatch>
66.     <onMismatch>DENY</onMismatch>
67. </filter>
68.     <QueueConnectionFactoryBindingName>ConnectionFactory
69. </QueueConnectionFactoryBindingName>
70.     <QueueBindingName>cms_dao_log</QueueBindingName>
71. </appender>
72. <appender name="jms_service"
class="ch.qos.logback.classic.net.JMSQueueAppender">
73.     <InitialContextFactoryName>
74.     org.apache.activemq.jndi.ActiveMQInitialContextFactory
75. </InitialContextFactoryName>
76.     <ProviderURL>tcp://192.168.1.120:61616</ProviderURL>
77.     <filter
class="ch.qos.logback.core.filter.EvaluatorFilter">
78.         <evaluator name="myEval_service">
79.             <expression>message.contains("service.impl")
</expression>
80.         </evaluator>
81.         <onMatch>ACCEPT</onMatch>
82.         <onMismatch>DENY</onMismatch>
83.     </filter>
84.     <QueueConnectionFactoryBindingName>ConnectionFactory
85. </QueueConnectionFactoryBindingName>
86.     <QueueBindingName>cms_service_log</QueueBindingName>
87. </appender>
88. <logger name="com.cms5.cmsservice.jms">
89.     <level value="DEBUG" />
90. </logger>
91. <logger name="java.sql.PreparedStatement">
92.     <level value="DEBUG" />
93. </logger>
94. <logger name="java.sql.Connection">
95.     <level value="DEBUG" />
96. </logger>
97. <logger name="java.sql.Statement">
98.     <level value="DEBUG" />
99. </logger>
100. <logger name="com.ibatis">

```

```

101.     <level value="DEBUG" />
102. </logger>
103. <logger name="com.ibatis.common.jdbc.SimpleDataSource">
104.     <level value="DEBUG" />
105. </logger>
106. <logger name="com.ibatis.common.jdbc.ScriptRunner">
107.     <level value="DEBUG" />
108. </logger>
109. <logger
name="com.ibatis.sqlmap.engine.impl.SqlMapClientDelegate">
110.     <level value="DEBUG" />
111. </logger>
112. <logger name="com.danga.MemCached">
113.     <level value="INFO" />
114. </logger>
115. <logger name="org.springframework.test">
116.     <level value="DEBUG" />
117. </logger>
118. <logger name="org.apache.struts2">
119.     <level value="DEBUG" />
120. </logger>
121. <root>
122.     <level value="INFO" />
123.     <!--<appender-ref ref="stdout" />
124.     -->
125.     <appender-ref ref="logfile-dao" />
126.     <appender-ref ref="logfile-service" />
127.     <appender-ref ref="jms_dao" />
128.     <appender-ref ref="jms_service" />
129. </root>
130. </configuration>

```

就先不做说明了，里面的配置还是比较明确的，关于jmsappender那部分，之前也做过简单说明。

项目中用到的logback列子

分类： [Web开发](#) 2011-09-21 13:59 908人阅读 [评论\(1\)](#) [收藏](#) [举报](#)

[class](#) [thread](#) [layout](#)

莫个银行项目中实际引用的Llogback实例，禁提供大家参考！！！！

[\[html\]](#) [view plain copy](#)

```
1. <!-- if debug set to true, context info always print
2. otherwise the contex info will print when error occour -->
3. <configuration scan="true" scanPeriod="60 seconds"
debug="false">
4.
5.   <appender name="stdout"
class="ch.qos.logback.core.ConsoleAppender">
6.     <layout class="ch.qos.logback.classic.PatternLayout">
7.       <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level
%logger{40} - %msg%n</pattern>
8.     </layout>
9.   </appender>
10.
11.   <!-- 所有日志文件-->
12.   <appender name="AllLog"
class="ch.qos.logback.core.rolling.RollingFileAppender">
13.     <rollingPolicy
class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
14.       <fileNamePattern>log/abcchina/all/%d{yyyy-MM-
dd}.log</fileNamePattern>
15.     <maxHistory>30</maxHistory>
16.   </rollingPolicy>
17.   <encoder>
18.     <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{40} -
%msg%n</pattern>
19.   </encoder>
20. </appender>
21.
22.   <appender name="TaskLog"
class="ch.qos.logback.core.rolling.RollingFileAppender">
23.     <rollingPolicy
class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
24.       <fileNamePattern>log/abcchina/task/%d{yyyy-MM-
dd}.log</fileNamePattern>
25.     <maxHistory>30</maxHistory>
26.   </rollingPolicy>
27.   <encoder>
28.     <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{40} -
%msg%n</pattern>
29.   </encoder>
30. </appender>
31.   <appender name="SendLog"
```

```
class="ch.qos.logback.core.rolling.RollingFileAppender">
32.   <rollingPolicy
class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
33.     <fileNamePattern>log/abcchina/send/%d{yyyy-MM-
dd}.log</fileNamePattern>
34.     <maxHistory>30</maxHistory> 项目中用到的logback例子
35.   </rollingPolicy>
36.   <encoder>
37.     <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{40} -
%msg%n</pattern>
38.   </encoder>
39. </appender>
40. <!-- 为某个包下的所有类指定Appender 这里也可以指定类名称例如:
com.aa.bb.ClassName-->
41. <logger name="com.aa.bb" >
42.   <appender-ref ref="TaskLog" />
43. </logger>
44.
45. <logger name="com.aa.cc" >
46.   <appender-ref ref="SendLog" />
47. </logger>
48.
49. <root level="INFO">
50.   <appender-ref ref="AllLog" />
51. </root>
52. </configuration>
```