

# Personalização do Log

## Introdução

O Quality Manager faz uso da biblioteca Apache log4j 2 para registro e auditoria das mensagens geradas pela aplicação tanto para visualização de possíveis falhas quanto para exibição de mensagens informativas. O sistema conta com uma configuração específica de logs **log4j2.xml** como visto abaixo (versão 2.5.8 em diante):

```
<?xml version="1.0" encoding="UTF-8"?>

<Configuration status="INFO" monitorInterval="60">

    <Appenders>
        <RollingFile name="ROLLING_FILE_APPENDER"
            fileName="${sys:catalina.home}/logs/md2-quality-
manager.log"
            filePattern="${sys:catalina.home}/logs/md2-quality-manager.%d{yyyy-MM-
dd}-%i.log">
            <PatternLayout charset="UTF-8"
                pattern="[%-5level] %d{yyyy-MM-dd HH:mm:ss.SSS} - %c{1} - %msg%n%xEx{
filters(jdk.internal.reflect,
java.lang.reflect,org.jboss.weld,org.hibernate.internal,org.hibernate.hql.internal,
org.hibernate.engine.jdbc.internal, org.hibernate.loader.Loader,
org.hibernate.engine.query, org.hibernate.jpa,
org.apache.catalina, org.apache.coyote, org.apache.tomcat, javax.faces,
com.sun.faces,
org.omnifaces, javax.el, org.apache.el)}"/>
            <Policies>
                <OnStartupTriggeringPolicy/>
                <TimeBasedTriggeringPolicy/>
                <SizeBasedTriggeringPolicy size="50 MB"/>
            </Policies>
            <DefaultRolloverStrategy max="20"/>
        </RollingFile>

        <Console name="CONSOLE_APPENDER" target="SYSTEM_OUT">
            <PatternLayout charset="UTF-8"
```

```

        pattern "[%5level] %d{yyyy-MM-dd HH:mm:ss.SSS} - %c{1} - %msg%n%xEx{
        filters(jdk.internal.reflect,
java.lang.reflect,org.jboss.weld,org.hibernate.internal,org.hibernate.hql.internal,
        org.hibernate.engine.jdbc.internal, org.hibernate.loader.Loader,
org.hibernate.engine.query, org.hibernate.jpa,
        org.apache.catalina, org.apache.coyote, org.apache.tomcat, javax.faces,
com.sun.faces,
        org.omnifaces, javax.el, org.apache.el)}/>
    </Console>
</Appenders>

<Loggers>
    <!-- Root Logger -->
    <Root level="INFO" additivity="false">
        <appender-ref ref="CONSOLE_APPENDER" />
        <appender-ref ref="ROLLING_FILE_APPENDER" />
    </Root>

    <!-- Quality Manager Logger -->
    <Logger name="com.md2net" level="INFO" additivity="false">
        <appender-ref ref="CONSOLE_APPENDER" />
        <appender-ref ref="ROLLING_FILE_APPENDER" />
    </Logger>

    <!-- Quartz -->
    <Logger name="org.quartz" level="ERROR" additivity="false">
        <appender-ref ref="CONSOLE_APPENDER" />
        <appender-ref ref="ROLLING_FILE_APPENDER" />
    </Logger>

    <!-- Flyway -->
    <Logger name="org.flywaydb" level="INFO" additivity="false">
        <appender-ref ref="CONSOLE_APPENDER" />
        <appender-ref ref="ROLLING_FILE_APPENDER" />
    </Logger>

    <!-- Hikari -->
    <Logger name="com.zaxxer.hikari" level="ERROR" additivity="false">
        <appender-ref ref="CONSOLE_APPENDER" />
        <appender-ref ref="ROLLING_FILE_APPENDER" />
    </Logger>

```

```

</Logger>

<!-- RESTEASY -->
<logger name="org.jboss.resteasy" level="ERROR" additivity="false">
    <appender-ref ref="CONSOLE_APPENDER" />
    <appender-ref ref="ROLLING_FILE_APPENDER" />
</logger>

<!-- WELD -->
<logger name="org.jboss.weld" level="ERROR" additivity="false">
    <appender-ref ref="CONSOLE_APPENDER" />
    <appender-ref ref="ROLLING_FILE_APPENDER" />
</logger>

<!-- Hibernate -->
<Logger name="org.hibernate" level="ERROR" additivity="false">
    <appender-ref ref="CONSOLE_APPENDER" />
</Logger>

<!-- Hibernate Query -->
<Logger name="org.hibernate.SQL" level="OFF" additivity="false">
    <appender-ref ref="CONSOLE_APPENDER" />
</Logger>

<!-- Hibernate Query Params -->
<Logger name="org.hibernate.type.descriptor.sql" level="OFF"
additivity="false">
    <appender-ref ref="CONSOLE_APPENDER" />
</Logger>

</Loggers>
</Configuration>

```

Acima, pode-se ver que a aplicação dispõe de dois appenders que propagam o registro do log de maneiras distintas: Pela rotação de arquivos e pelo console (stdout)

- **Rotação em arquivo:** Nesta modalidade de log, a aplicação gera uma série de arquivos com prefixo md2-quality-manager-\*.log que irá conter todas as mensagens registradas pelos loggers. Este arquivo é regido por políticas quanto ao seu tamanho, tempo e propagação. Este arquivo é disponibilizado pela tela de Interface e Administração para Diagnóstico como mecanismo de diagnóstico de problemas ocorridos em execução.

- **Console:** Nesta modalidade de log, a aplicação gera o registro do log no fluxo de saída (stdout) para visualização no console.

Existem nove Loggers explicitos no arquivo (excluindo o root) para implementação dos appenders sendo eles:

- **Quality Manager Logger:** Controla as mensagens de log geradas pela aplicação.
- **Quartz Logger:** Controla as mensagens exibidas pelo Quartz (Scheduler)
- **Flyway Logger:** Controla as mensagens exibidas pelo Flyway (Migração de Banco)
- **Hikari Logger:** Controla as mensagens exibidas pelo Hikari (Pool de Conexões)
- **RestEasy Logger:** Controla as mensagens exibidas pelo RestEasy (API rest)
- **Weld Logger:** Controla as mensagens exibidas pelo Weld (Injeção de Dependências)
- **Hibernate Logger:** Controla as mensagens exibidas pelo Hibernate (ORM)
- **Hibernate (SQL) Logger:** Controla a exibição das queries geradas pelo Hibernate (Apenas em nível DEBUG)
- **Hibernate (Type Descriptor) Logger:** Controla a exibição dos parâmetros das queries (Apenas em nível TRACE)

Por padrão, a aplicação utiliza o **PatternLayout** para especificar o formato geral das mensagens como ilustrado abaixo. Todas as mensagens iniciam-se pelo seu nível, data/hora do evento, classe responsável, a mensagem e a stacktrace da exceção (se houver)

```
INFO ] 2022-06-21 23:25:51.579 - HikariDataSource - HikariPool-1 - Starting...
INFO ] 2022-06-21 23:25:51.954 - HikariDataSource - HikariPool-1 - Start completed.
INFO ] 2022-06-21 23:25:56.671 - HikariDataSource - HikariPool-2 - Starting...
INFO ] 2022-06-21 23:25:56.757 - HikariDataSource - HikariPool-2 - Start completed.
INFO ] 2022-06-21 23:26:02.859 - HikariDataSource - HikariPool-3 - Starting...
INFO ] 2022-06-21 23:26:02.925 - HikariDataSource - HikariPool-3 - Start completed.
INFO ] 2022-06-21 23:26:03.917 - VersionPrinter - Flyway-Community Edition 7.12.1 by Redgate
INFO ] 2022-06-21 23:26:03.922 - BaseDatabaseType - Database: jdbc:postgresql://localhost:5432/qualitymanager (PostgreSQL 12.11)
INFO ] 2022-06-21 23:26:04.349 - DbValidate - Successfully validated 76 migrations (execution time 00:00.328s)
INFO ] 2022-06-21 23:26:04.382 - DbMigrate - Current version of schema "public": 0076
WARN ] 2022-06-21 23:26:04.383 - DbMigrate - outOfOrder mode is active. Migration of schema "public" may not be reproducible.
INFO ] 2022-06-21 23:26:04.384 - DbMigrate - Schema "public" is up to date. No migration necessary.
```

## Classificação quanto ao nível

O log4j faz uso de diferentes níveis de acordo com a criticidade do evento através dos 6 primeiros níveis (TRACE, DEBUG, INFO, WARN, ERROR e FATAL). Os níveis restantes (ALL e OFF) apenas determinam se a captura dos logs está ativada ou não. A exibição das mensagens quanto ao nível segue abaixo:

| Exibição por nível | TRACE | DEBUG | INFO | WARN | ERROR | FATAL |
|--------------------|-------|-------|------|------|-------|-------|
| TRACE              | Sim   | Sim   | Sim  | Sim  | Sim   | Sim   |
| DEBUG              | Não   | Sim   | Sim  | Sim  | Sim   | Sim   |
| INFO               | Não   | Não   | Sim  | Sim  | Sim   | Sim   |
| WARN               | Não   | Não   | Não  | Sim  | Sim   | Sim   |
| ERROR              | Não   | Não   | Não  | Não  | Sim   | Sim   |

|              |     |     |     |     |     |     |
|--------------|-----|-----|-----|-----|-----|-----|
| <b>FATAL</b> | Não | Não | Não | Não | Não | Sim |
| <b>ALL</b>   | Sim | Sim | Sim | Sim | Sim | Sim |
| <b>OFF</b>   | Não | Não | Não | Não | Não | Não |

O Quality Manager faz uso dos níveis de alguns níveis acima para registrar os diferentes eventos ocasionados pelo sistema de acordo com a relação abaixo:

| Exibição por nível | Tipo de mensagem   |
|--------------------|--|
| <b>TRACE</b>       | <p>Mensagens para o contexto de depuração. Irá registrar a modificação sofrida pelas entidades no banco de dados juntamente com sua chave primária no banco de dados. O formato padrão da mensagem que será exibida é:</p> <p><b>"Uma instância da entidade [X] foi &lt;inserida   atualizada   removida&gt;. ID [Y]"</b></p> <pre>- SessionHibernateListener - Uma instância da entidade [LogSistema] foi inserida. ID [787] - SessionHibernateListener - Uma instância da entidade [Usuario] foi atualizada. ID [193]</pre>  |
| <b>DEBUG</b>       | <p>Mensagens para o contexto de depuração. Irá registrar as ações realizadas no sistema por usuário e por módulo (Navegação, criação ou modificação de registros e etc). O formato padrão desta mensagem para navegação é <b>"Usuário [X] acessou o módulo [Y]"</b> e para ações é <b>"Usuário [X] no módulo [Y] realizou ação Z"</b></p> <pre>- UsuarioLogado - Usuário [admin_modelo] acessou o módulo [Setor] - SetorBean - Usuário [admin_modelo] no módulo [Setor] abriu a edição do setor [MeuNovoSetor] - SetorBean - Usuário [admin_modelo] no módulo [Setor] cancelou a edição do setor</pre> |
| <b>INFO</b>        | Mensagens com caráter informativo. Irá registrar as mensagens de inicialização do Quality Manager e a execução das tarefas automatizadas (notificações, envio de emails, expurgos e etc)   |
| <b>WARN</b>        | Mensagens para indicar avisos. Irá registrar de falhas em teste de conexão (LDAP, MDM) e informações quanto ao limite gerenciado pelo Licenciamento de Arquivos  |
| <b>ERROR</b>       | Mensagens para indicar erros. Irá registrar falha na execução de alguma ação que pode representar um problema com a aplicação  |

Para controlar a exibição das mensagens, o usuário pode colocar uma cópia do arquivo **log4j2.xml** na pasta **conf** do Tomcat antes da inicialização do Quality Manager. Para personalização do nível configurado no nível TRACE, por exemplo, basta alterar a propriedade **level** em **Logger** conforme abaixo:

```
<!-- Quality Manager Logger -->
<Logger name="com.md2net" level="TRACE" additivity="false">
  <appender-ref ref="CONSOLE_APPENDER" />
  <appender-ref ref="ROLLING_FILE_APPENDER" />
</Logger>
```

Por padrão, o arquivo monitora as alterações no arquivo em tempo de execução a cada um minuto. Para modificar esta configuração em 30 segundos, por exemplo, altere a propriedade **monitorInterval** em **Configuration** com o valor em segundos:

```
<Configuration status="INFO" monitorInterval="30">
```

---

Revision #6

Created 4 August 2022 17:22:17

Updated 12 December 2022 23:52:05