# Log4j Tutorial

Log4j是一个简单灵活的日志框架

Log4j的Maven依赖

|  |
| --- |
| <properties>  <log4j-version>1.2.7</log4j-version> </properties> |
| <dependencies>  <dependency>  <groupId>log4j</groupId>  <artifactId>log4j</artifactId>  <version>${log4j-version}</version>  </dependency> </dependencies> |

Properties配置

log4j.properties配置文件

|  |  |
| --- | --- |
| 文件名 | log4j.properties |
| 配置文件 | log4j.rootLogger=DEBUG, consoleAppender, fileAppender    log4j.appender.consoleAppender=org.apache.log4j.ConsoleAppender  log4j.appender.consoleAppender.layout=org.apache.log4j.PatternLayout  log4j.appender.consoleAppender.layout.ConversionPattern=[%t] %-5p %c %x - %m%n    log4j.appender.fileAppender=org.apache.log4j.RollingFileAppender  log4j.appender.fileAppender.layout=org.apache.log4j.PatternLayout  log4j.appender.fileAppender.layout.ConversionPattern=[%t] %-5p %c %x - %m%n  log4j.appender.fileAppender.File=demoApplication.log |

测试代码

|  |
| --- |
| import org.apache.log4j.Logger; import org.apache.log4j.PropertyConfigurator;  public class Log4j1PropertiesDemo {   static Logger logger = Logger.getLogger(Log4j1PropertiesDemo.class);   /\*\*  \* 加载配置文件  \* 打印Log  \*  \* @param args  \*/  public static void main(String[] args) {  PropertyConfigurator.configure("src/main/resources/log4j.properties");  logger.debug("Log4j appender configuration is successful !!");  } } |

代码执行结果

|  |
| --- |
| [main] DEBUG com.log4j1.Log4j1PropertiesDemo - Log4j appender configuration is successful !! |

将日志输出到控制台

|  |
| --- |
| # Root logger  log4j.rootLogger=INFO, console    log4j.appender.console=org.apache.log4j.ConsoleAppender  log4j.appender.console.Target=System.out  log4j.appender.console.layout=org.apache.log4j.PatternLayout  log4j.appender.consoleAppender.layout.ConversionPattern=[%t] %-5p %c %x - %m%n |

将日志输出到文件

|  |
| --- |
| # Root logger  log4j.rootLogger=INFO, file    # Direct log messages to a log file  log4j.appender.file=org.apache.log4j.RollingFileAppender    log4j.appender.file.File=C:\\temp\info.log  log4j.appender.file.MaxFileSize=10MB  log4j.appender.file.MaxBackupIndex=10  log4j.appender.file.layout=org.apache.log4j.PatternLayout  log4j.appender.file.layout.ConversionPattern=[%t] %-5p %c %x - %m%n |

Xml配置文件

log4j.xml配置文件

|  |
| --- |
| **<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd"*>*** <**log4j:configuration debug="true" xmlns:log4j='http://jakarta.apache.org/log4j/'**>   <**appender name="console" class="org.apache.log4j.ConsoleAppender"**>  <**param name="Target" value="System.out"**/>  <**layout class="org.apache.log4j.PatternLayout"**>  <**param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n"** />  </**layout**>  </**appender**>   <**appender name="fileAppender" class="org.apache.log4j.RollingFileAppender"**>  <**param name="File" value="demoApplication.log"**/>  <**layout class="org.apache.log4j.PatternLayout"**>  <**param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n"** />  </**layout**>  </**appender**>   <**root**>  <**priority value ="debug"**></**priority**>  <**appender-ref ref="console"**></**appender-ref**>  <**appender-ref ref="fileAppender"**></**appender-ref**>  </**root**>  </**log4j:configuration**> |

测试代码

|  |
| --- |
| **import** org.apache.log4j.Logger; **import** org.apache.log4j.xml.DOMConfigurator;  **public class** Log4j1XmlDemo {  **static** Logger *logger* = Logger.*getLogger*(Log4j1XmlDemo.**class**);   */\*\*  \* 加载配置文件  \* 打印Log  \** ***@param args*** *\*/* **public static void** main(String[] args) {   DOMConfigurator.*configure*(**"src/main/resources/log4j.xml"**);  *logger*.debug(**"Log4j appender configuration is successful !!"**);  } } |

代码执行结果

|  |
| --- |
| DEBUG Log4j1XmlDemo:17 - Log4j appender configuration is successful !! |

通过log4j.xml将日志输出到控制台

|  |
| --- |
| <!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">  <log4j:configuration debug="true" xmlns:log4j='<http://jakarta.apache.org/log4j/>'>      <appender name="console" class="org.apache.log4j.ConsoleAppender">      <param name="Target" value="System.out"/>      <layout class="org.apache.log4j.PatternLayout">      <param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n"/>      </layout>    </appender>      <root>      <priority value ="debug"></priority>      <appender-ref ref="console"></appender-ref>    </root>    </log4j:configuration> |

通过log4j.xml将日志输出到文件中

|  |
| --- |
| <!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">  <log4j:configuration debug="true" xmlns:log4j='<http://jakarta.apache.org/log4j/>'>      <appender name="fileAppender" class="org.apache.log4j.RollingFileAppender">      <param name="File" value="demoApplication.log"/>      <layout class="org.apache.log4j.PatternLayout">      <param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n"/>      </layout>    </appender>      <root>      <priority value ="debug"></priority>      <appender-ref ref="fileAppender"></appender-ref>    </root>    </log4j:configuration> |

log4j默认配置信息

测试代码

|  |
| --- |
|  |

代码执行结果

|  |
| --- |
|  |

Log4j的日志级别

|  |  |
| --- | --- |
| 日志级别 | 描述 |
| **ALL** |  |
| **TRACE** |  |
| **DEBUG** |  |
| **WARN** |  |
| **ERROR** |  |
| **FATAL** |  |
| **OFF** |  |

如何在log4j.properties配置日志级别

|  |
| --- |
| log4j.rootLogger=DEBUG, consoleAppender    log4j.appender.consoleAppender=org.apache.log4j.ConsoleAppender  log4j.appender.consoleAppender.layout=org.apache.log4j.PatternLayout  log4j.appender.consoleAppender.layout.ConversionPattern=[%t] %-5p %c %x - %m%n    #Log info messages for package 'com.howtodoinjava.web.controller'  log4j.logger.com.howtodoinjava.web.controller=INFO, consoleAppender |

如何在log4j.xml配置日志级别

|  |
| --- |
| <log4j:configuration debug="true" xmlns:log4j='<http://jakarta.apache.org/log4j/>'>      <appender name="console" class="org.apache.log4j.ConsoleAppender">      <param name="Target" value="System.out"/>      <layout class="org.apache.log4j.PatternLayout">      <param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n"/>      </layout>    </appender>      <logger name="com.howtodoinjava.web.controller">      <level value="INFO" />      <appender-ref ref="console" />    </logger>      <root>      <level value="DEBUG" />      <appender-ref ref="console"></appender>    </root>    </log4j:configuration> |

日志级别如何工作的

|  |
| --- |
| ALL < TRACE < DEBUG < INFO < WARN < ERROR < FATAL < OFF |

trace designates finer-grained informational events than the DEBUG.Since:1.2.12

日志级别测试代码

|  |
| --- |
|  |

代码执行结果

|  |
| --- |
|  |

Log4j的Appender

# Log4j ConsoleAppender配置示例

|  |  |
| --- | --- |
| 属性 | 描述 |
| immediateFlush | 设置是否使用每个日志记录输出请求刷新控制台流。 |
| encoding | 覆盖默认的字符编码方案。 |
| threshold | 任何级别低于阈值的日志记录请求都将被忽略。 |
| target |  |