# Log4j专题

## Log4j简介

Log4j就是帮助开发人员进行日志输出管理的API类库。它最重要的特点就可以配置文件灵活的设置日志信息的优先级Loggers、日志信息的输出目的地Appenders以及日志信息的输出格式Layouts。综合使用这三个组件可以轻松地记录信息的类型和级别，并可以在运行时控制日志输出的样式和位置。

## 为什么要使用log4j输出日志

记录日志可以作为日后处理问题的一个追溯，方便开发者根据日志来统计查询处理问题。此外，查阅日志内容可以了解项目的运行状况，发现项目存在的一些隐藏的bug，大多数人会选择使用System.out.println语句输出某个变量值的方法进行调试。这样会带来一个非常麻烦的问题：一旦哪天程序员决定不要显示这些System.out.println的东西了就只能一行行的把这些垃圾语句注释掉。若哪天又需调试变量值，则只能再一行行去掉这些注释恢复System.out.println语句，但是使用log4j只需要设置配置文件就可以。

## 怎么使用log4j输出日志

关键是log4j三个组件的使用

#### 1、Loggers

Loggers组件在此系统中被分为五个级别：DEBUG、INFO、WARN、ERROR和FATAL。这五个级别是有顺序的，DEBUG < INFO < WARN < ERROR < FATAL，分别用来指定这条日志信息的重要程度，明白这一点很重要，Log4j有一个规则：只输出级别不低于设定级别的日志信息，假设Loggers级别设定为INFO，则INFO、WARN、ERROR和FATAL级别的日志信息都会输出，而级别比INFO低的DEBUG则不会输出。

#### 2、Appenders

禁用和使用日志请求只是Log4j的基本功能，Log4j日志系统还提供许多强大的功能，比如允许把日志输出到不同的地方，如控制台（Console）、文件（Files）等，可以根据天数或者文件大小产生新的文件，可以以流的形式发送到其它地方等等。

常使用的类如下：

org.apache.log4j.ConsoleAppender（控制台）

org.apache.log4j.FileAppender（文件）

org.apache.log4j.DailyRollingFileAppender（每天产生一个日志文件）

org.apache.log4j.RollingFileAppender（文件大小到达指定尺寸的时候产生一个新的文件）

org.apache.log4j.WriterAppender（将日志信息以流格式发送到任意指定的地方）

#### 3、Layouts

有时用户希望根据自己的喜好格式化自己的日志输出，Log4j可以在Appenders的后面附加Layouts来完成这个功能。Layouts提供四种日志输出样式，如根据HTML样式、自由指定样式、包含日志级别与信息的样式和包含日志时间、线程、类别等信息的样式。

常使用的类如下：

org.apache.log4j.HTMLLayout（以HTML表格形式布局）

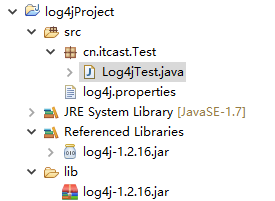
org.apache.log4j.PatternLayout（可以灵活地指定布局模式）

org.apache.log4j.SimpleLayout（包含日志信息的级别和信息字符串）

org.apache.log4j.TTCCLayout（包含日志产生的时间、线程、类别等信息）

## 代码示例

#### 1、代码结构



#### 2、配置文件说明

先看一个常见的log4j.properties文件，它是在控制台和myweb.log文件中记录日志：

#设置日志的输出级别、控制台的数据、文件的输出

#此处级别设置的是DEBUG，是最小级别

log4j.rootLogger=DEBUG, stdout, logfile

log4j.category.org.springframework=ERROR

log4j.category.org.apache=INFO

#控制台的输出控制

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

#每行日志的格式

log4j.appender.stdout.layout.ConversionPattern=%d %p [%c] - %m%n

#文件大小到达指定尺寸的时候产生一个新的文件

log4j.appender.logfile=org.apache.log4j.RollingFileAppender

#设置文件的输入路径

log4j.appender.logfile.File=d:\\log\\myweb.log

#每个文件的大小 ，为了测试设置的值小

log4j.appender.logfile.MaxFileSize=1KB

#最多产生的文件的索引数

log4j.appender.logfile.MaxBackupIndex=5

log4j.appender.logfile.layout=org.apache.log4j.PatternLayout

log4j.appender.logfile.layout.ConversionPattern=%d %p [%c] - %m%n

#### 3、测试代码

**public** **class** Log4jTest {

**static** Logger *logger* = Logger.*getLogger*(Log4jTest.**class**);

**public** **static** **void** main(String[] args) {

**for** (**int** i = 0; i < 100; i++) {

*logger*.debug("这是个debug级别的输出，会出现在控制台和一个log日志文件中");

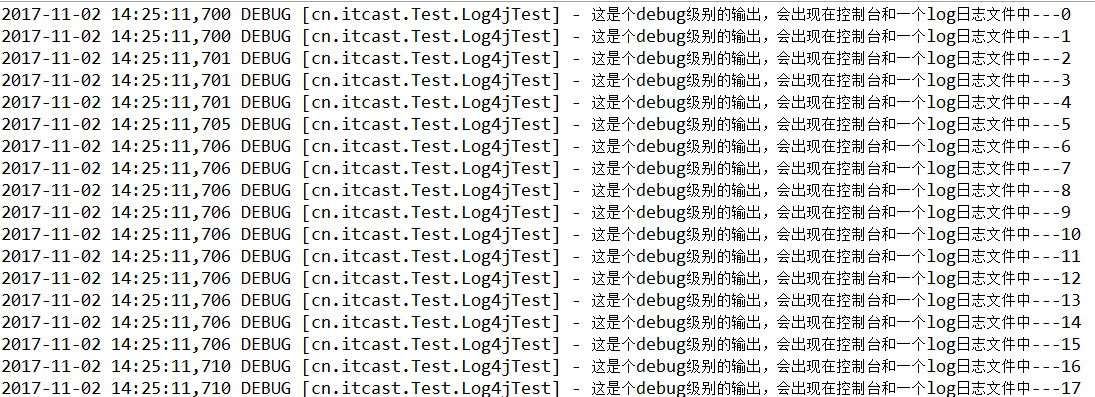
}

}

}

#### 4、执行后产生的结果

控制台：



生成文件：



如果把log4j.rootLogger=DEBUG, stdout, logfile中的DEBUG改为ERROR则不会再控制台输出日志并且不会产生新的log文件

自己测试发现：生成的6个文件，数字越大日志的时间越早，如果6个文件都达到最大容量，那么没有数字的日志文件会被清空并写入日志。

报错日志到文件的关键配置是log4j.appender.logfile.File=d:\\log\\myweb.log，去掉这句话就不会保存了。