# Log4j

Apache开源(http://logging.apache.org/log4j/),轻量的日志管理框架,

控制日志信息输出地:控制台,文件,GUI,套接字,NT事件记录器,Unix Syslog

守护进程等,控制输出格式,定义日志级别,无需修改代码,配置文件修改

## 日志用途分类:

开发调试日志

系统运行日志

## log4j对比System.out.println优势

1.分级别控制日志的输出与开关

调试阶段写了100system语句,上线时需不需要这些语句?

(不需要手动删除,自己就能不输出)

2.按不同的类和包进行输出控制

很多模块都有输出语句,但是不能很好区分是哪一个输出?

(不需要区分,可以根据不同的包,不同的类分情况输出)

3.控制日志不同的输出位置

# Log4j 组成(三大核心组件)

//log4j关键字 必须要有 后面的内容通过.来连接 (大小写敏感)

**//rootLogger:定义日志的根节点(=左边是定义内容 ＝右边定义内容值)**

//info:日志级别

//e.g info,stdou1,stdou2 表示info级别日志有两个stdou1和stdou2

//stdou1:定义的日志文件的名称

//appender:表示日志输入的目的地

e.g log4j.appender.stdou1 表示stdou1日志文件输出到 ＝ 右边制定的位置

//File:表示如果日志是输出到文件中,需要制定该文件的名称

=右边制定文件名称,需要注意该名称包括path.fileName.如果不制定path.默认工程下

//Append:是否追加 true表示追加 false表示覆盖

//layout:表示日志输出的样式

如果输出的样式是PatternLayout需要自定义输出模版.

该模版可以通过ConversionPattern关键字来获取

log4j.rootLogger=info,stdou1,stdou2

log4j.appender.stdou1=org.apache.log4j.FileAppender

log4j.appender.stdou1.File=test.log

log4j.appender.stdou1.Append=true

log4j.appender.stdou1.layout=org.apache.log4j.PatternLayout

log4j.appender.stdou1.layout.ConversionPattern=%d{yyyy MM dd HH:mm:ss,SSS}-%m%n

log4j.appender.stdou2=org.apache.log4j.ConsoleAppender

log4j.appender.stdou2.layout=org.apache.log4j.PatternLayout

log4j.appender.stdou2.layout.ConversionPattern=%d{yyyy-MM-dd HH\:mm\:ss} %p [%c] %m%n

## 1.Logger

代表了Log4J的日志记录器,它能够对日志信息进行分类筛选,控制日志级别,完成日志信息的具体输出操作Log4j允许定义多个日志记录器,有且只有一个根记录器，根记录器由org.apache.log4j.Logger支持根日志记录器不能通过名字获取，只能通过Logger.getRootLogger();获取其他日志记录器 Logger.getLogger(String name)

日志级别：

debug(调试)<info(信息)<warn(警告)<error(错误)<fatal(奔溃)

主要方法

//得到根Logger

public static Logger getRootLogger();

//得到名字为name的Logger

public static Logger getLogger(String name);

//相当于Logger.getLogger(clazz.getName())

public static Logger getLogger(java.lang.Class clazz);

public void debug(Object message); //输出debug级别的日志信息

public void info(Object message); //输出info级别的日志信息

public void warn(Object message); //输出warn级别的日志信息

public void error(Object message); //输出error级别的日志信息

public void fatal(Object message); //输出fatal级别的日志信息

public void trace(Object message); //输出trace级别的日志

## 2.Appender

控制日志的输出位置。管理日志语句的输出结果,接收Logger发送来的日志,决定日志输出目的地,log4j允许将日志请求输出到多个目的地

在Log4j中,输出目的地被称为Appender,支持一下目的地:

Console(控制台),File(文件),GUI components(GUI组件),

Remote socket servers(套接字服务器),JMS,NT Event Logger(NT事件记录器),

Remote UNIX Syslog daemon(UNIX Syslog守护进程)

Log4j提供的Appender有以下几种：

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 //文件大小达到指定尺寸时产生一个新文件

org.apache.log4j.WriterAppender //将日志以流格式发送到任意指定位置

org.apache.log4j.JDBCAppender //把日志用JDBC记录到数据库

## 3.Layout

配置日志的输出格式。决定Appender将日志语句写入日志目的地采用的格式。输出信息的格式,Log4j中称为Layout

**常用Layout：**

org.apache.log4j.HTMLLayout //HTML表格形式布局

org.apache.log4j.PatternLayout //根据模板布局

org.apache.log4j.SimpleLayout //包含日志信息级别和信息字符串

org.apache.log4j.TTCCLayout //包含日志产生的线程,类别,内容等信息

一个Logger可以有多个Appender,即日志信息可以同时输出到多个设备上,

一个Appender可以被多个Logger使用

每个Appender对应一种Layout

Logger

\

Appender1 --> Layout

/

Logger

\

Appender2 --> Layout

# 二、Log4j的基本使用步骤

1.定义配置文件

properties配置文件

2.日志记录器的配置

log4j.rootLogger = [priority],appenderName,appenderName, ...

eg:

log4j.rootLogger=info,stdou1,stdout2

3.Appender的配置

log4j.appender.stdou1=org.apache.log4j.FileAppender

log4j.appender.stdou1.File=test.log

log4j.appender.stdou1.Append=false

4.Layout的配置

log4j.appender.stdout1.layout=org.apache.log4j.HTMLLayout

# 三.常用Layout类型ConversionPattern

PatternLayout让开发者依照ConversionPattern定义输出格式

ConversionPattern中一些指定日志内容和格式的预定义符号如下:

%m 输出代码中指定的消息

%p 输出优先级，即DEBUG，INFO，WARN，ERROR，FATAL

%r 输出自应用启动到输出该log信息耗费的毫秒数

%c 输出root用户

%t 输出产生该日志事件的线程名

%n 输出一个回车换行符，Windows平台为“\r\n”，Unix平台为“\n”

%d 输出日志时间点的日期或时间，默认格式为ISO8601，也可以在其后指定格式

如：%d{yyy MMM dd HH:mm:ss,SSS}

输出类似：2002年10月18日22:10:28，921

%l 输出日志事件的发生位置，包括类目名、发生的线程，以及在代码中的行数

-X号: X信息输出时左对齐

%x: 输出和当前线程相关联的NDC(嵌套诊断环境),尤其用到像java servlets这样的多客户多线程的应用中。

%%: 输出一个"%"字符

%F: 输出日志消息产生时所在的文件名称

%L: 输出代码中的行号

可以在%与模式字符之间加上修饰符来控制其最小宽度、最大宽度、和文本的对齐方式。

如：

1) %20c：指定输出category的名称，最小的宽度是20，如果category的名称小于20的话，默认的情况下右对齐。

2) %-20c:指定输出category的名称，最小的宽度是20，如果category的名称小于20的话，"-"号指定左对齐。

3) %.30c:指定输出category的名称，最大的宽度是30，如果category的名称大于30的话，就会将左边多出的字符截掉，但小于30的话也不会有空格。

4) %20.30c:如果category的名称小于20就补空格，并且右对齐，如果其名称长于30字符，就从左边较远输出的字符截掉