

# Log4j

张海宁

贵州大学

*hnzhang1@gzu.edu.cn*

May 11, 2018

# Overview

- 1 Log4j
- 2 Log4j 的配置
- 3 Logger
- 4 Appender
- 5 Layout
- 6 Appendix
  - 相关资源

# Log4j 简介

Apache log4j is a logging library for Java.

With log4j it is possible to enable logging at runtime without modifying the application binary. Logging behavior can be controlled by editing a configuration file, without touching the application binary.

The target of the log output can be a file, an OutputStream, a java.io.Writer, a remote log4j server, a remote Unix Syslog daemon, or many other output targets.

# Log4j 的组成

Log4j 主要由以下 3 大组件构成：

- ① **Logger**  
可以定义多个 Logger，每个 Logger 实现不同的功能
- ② **Appender**  
指定 log 信息存储的目标地址
- ③ **Layout**  
格式化 log 信息的输出样式

# Log4j 的配置

# Log4j 的配置

## ① 下载 jar 文件

`http://www.apache.org/dyn/closer.cgi/logging/log4j/1.2.17/log4j-1.2.17.zip`

## ② 添加到 build path

将上一步下载的压缩包解压，将其中的 jar 文件放到项目的 lib 路径中，比如：`/Users/hainingzhang/git/javaweb/lab-java/WebContent/WEB-INF/lib`，然后右键点击 log4j 的 jar 文件，选择 build path-add to build path.

# Logger



# 日志级别

Logger 是 Log4j 的日志记录器，是 Log4j 的核心组件。Log4j 将输出的日志信息定义了 5 种级别，如表1。

name	type	desc	level
DEBUG	Object	输出调试级别的日志信息	1
INFO	Object	输出消息日志	2
WARN	Object	输出警告级别的日志信息	3
ERROR	Object	输出错误级别的日志信息	4
FATAL	Object	输出致使错误级别的日志信息	5

Table: Log4j 的日志级别

只有高过配置中定义级别的日志信息才会被输出。

# 日志输出

在程序中，可以使用 `Logger` 类的不同方法来输出对应级别的日志信息。

## 日志输出方法

```
logger.debug("debug log");  
logger.info("info log");  
logger.warn("warn log");  
logger.error("数据库连接失败");  
logger.fatal("内存不足");
```

在配置文件中配置 Logger 日志时，可以定义日志的级别、输出目标等信息。

## 配置日志级别

```
log4j.logger.[loggerName]=[loggerLevel], appenderName, ...
```

- loggerName  
日志的名称
- loggerLevel  
日志的级别
- appenderName  
日志的输出目的地

```
log4j.logger.myLogger=debug,file
```

# Logger

# Appender

# Appender 的种类

Appender 可以指定日志文件的目的地，几种常用的目的地如表2所示：

Appender 接口的实现类	Description
org.apache.log4j.ConsoleAppender	输出到控制台
org.apache.log4j.FileAppender	输出到日志文件
org.apache.log4j.RollingFileAppender	当文件大小超出限制时，重新生成新的日志文件
org.apache.log4j.DailyRollingAppender	每天只生成一个对应的日志文件

Table: Appender 的几种常用目的地

# 配置日志

在配置文件中配置 Logger 日志时，可以定义日志的级别、输出目标等信息。

## 配置输出目标

`log4j.appender.[dstName]=[Appender接口的实现类]`

`#appender`

`log4j.appender.file=org.apache.log4j.RollingFileAppender`

`log4j.appender.file.File=/root/log.htm`

`log4j.appender.file.MaxFileSize=10KB`

`log4j.appender.file.MaxBackupIndex=3`

`log4j.logger.myLogger=debug,file`

例子给出的 10KB 和 3 代表的意思是：日志文件增长到 10KB 后就会生成一个新的日志文件，并将此日志文件转为一个备份日志文件，当备份日志文件超出 3 个后，则最早的那个备份文件会被覆盖。

# Layout



# Layout 的种类

Appender 必须使用一个与之相关联的 Layout 附加在 Appender 上, 它可以根据用户的个人习惯格式化日志的输出格式, 例如: 文本文件、HTML 文件等。

layout 的子类	description
org.apache.log4j.HTMLLayout	将日志以 HTML 格式输出
org.apache.log4j.PatternLayout	将日志以指定的转换模式格式化输出
org.apache.log4j.SimpleLayout	将日志以简单的格式输出
org.apache.log4j.TTCCLayout	这种日志包含线程等信息

Table: Layout 的种类

为 appender 定义 Layout.

## 定义 Layout

```
log4j.appender.[dstName].layout=[Layout的子类]
```

```
#layout
```

```
log4j.appender.file.layout=org.apache.log4j.HTMLLayout
```

```
log4j.appender.console.layout=org.apache.log4j.PatternLayout
```

```
log4j.appender.console.layout.ConversionPattern=%d%F - %m%n
```

# 部分转换字符表

转换字符	描述
%c	日志名称
%d	产生日志的时间和日期
%l	日志操作代码所在的类名及方法名，还会包含行号
%p	以大写格式输出日志级别
%F	日志操作所在类的源文件名称
%m	除输出日志信息外，不包含任何信息
%n	日志信息中的换行符

Table: Layout 的部分转换字符对照表

# 实例



# The End

# Appendix

## ppt、项目源代码及实验指导书的地址

- ① ppt  
<https://github.com/gmsft/ppt/tree/master/javaweb>
- ② lab-java 项目  
<https://github.com/gmsft/javaweb>
- ③ 实验指导书  
放在 ppt 的仓库里