

Log4j

主讲：崔译

一、简介

logger for java

是一个用于记录系统日志的第三方jar

使用log4j 记录日志（访问日志、执行日志、异常日志....）

简化了 基础操作（IO、Sysout、jdbc），提供统一方法，可以同时将日志记录到不同位置

二、properties 配置方式

1、导入jar

```
<dependency>
  <groupId>org.log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.9</version>
</dependency>
```

2、导入配置文件

log4j.properties

```
log4j.rootLogger=INFO,cons,myFile

log4j.appender.cons=org.apache.log4j.ConsoleAppender
log4j.appender.cons.Target=System.out
log4j.appender.cons.layout=org.apache.log4j.PatternLayout
log4j.appender.cons.layout.ConversionPattern=%m%n

log4j.appender.myFile=org.apache.log4j.FileAppender
log4j.appender.myFile.file=./log.txt
log4j.appender.myFile.layout=org.apache.log4j.PatternLayout
log4j.appender.myFile.layout.ConversionPattern=%m%n
```

3、编写日志代码

```

package com;

import org.apache.log4j.Logger;

public class HelloWorld {
    public static void main(String[] args) {
        // 获取Logger 日志的记录器
        Logger logger = Logger.getLogger(HelloWorld.class);
        // 记录日志
        logger.error("这是日志内容");
    }
}

```

三、XML 配置方式

导入dtd文件

log4j.jar ---> org --> apache --> log4j --> xml --> log4.dtd

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE log4j:configuration PUBLIC "log4j" "log4j.dtd" >
<log4j:configuration>

    <appender name="cons" class="org.apache.log4j.ConsoleAppender">
        <param name="target" value="System.out"/>
        <layout class="org.apache.log4j.SimpleLayout"></layout>
    </appender>

    <appender name="myFile" class="org.apache.log4j.FileAppender">
        <param name="file" value="./newLog.txt"/>
        <layout class="org.apache.log4j.SimpleLayout"></layout>
    </appender>

    <root>
        <level value="INFO"></level>
        <appender-ref ref="cons"/>
        <appender-ref ref="myFile"/>
    </root>

</log4j:configuration>

```

四、java代码方式

```

Logger logger = Logger.getLogger(HelloWorld.class);
logger.setLevel(Level.INFO);

ConsoleAppender conAppender = new ConsoleAppender();

```

```
conAppender.setTarget(ConsoleAppender.SYSTEM_OUT);
conAppender.activateOptions();

conAppender.setLayout(new SimpleLayout());

FileAppender fileAppender = new FileAppender();
fileAppender.setName("myFile");
fileAppender.setFile("./f.txt");
fileAppender.activateOptions();
fileAppender.setLayout(new SimpleLayout());

logger.addAppender(conAppender);
logger.addAppender(fileAppender);

logger.error("xxxxx");
```

五、关于文件名

在没有覆盖系统变量的前提下，文件名只能是 `log4j.xml` 或者 `log4j.properties`

```
static public final String DEFAULT_CONFIGURATION_FILE = "log4j.properties";
static final String DEFAULT_XML_CONFIGURATION_FILE = "log4j.xml";
if(configurationOptionStr == null) {
    url = Loader.getResource(DEFAULT_XML_CONFIGURATION_FILE);
    if(url == null) {
        url = Loader.getResource(DEFAULT_CONFIGURATION_FILE);
    }
}
```

六、rootLogger

- 值 是以 `,` 分割的字符串
- 第一个值是 日志等级（Level），后面的所有的值是 日志记录器（Appender）的名字
- 只有写在rootLogger中的日志记录器才会有效

七、日志等级

log4j 的日志 分为以下5个等级，每个等级具有特定意义，对应一个同名方法

- debug 调试日志
- info 普通消息日志
- warn 警告消息日志
- error 错误消息日志
- fatal 灾难消息日志

当且仅当，方法等级 大于等于 rootLogger中的Level 时，日志才会被记录

八、Appender

1、JDBCAppender

将日志记录在数据中

```
# 选择jdbc记录器
log4j.appender.jdbc=org.apache.log4j.jdbc.JDBCAppender
# 驱动
log4j.appender.jdbc.driver=com.mysql.jdbc.Driver
# 密码
log4j.appender.jdbc.password=
# url
log4j.appender.jdbc.URL=jdbc:mysql://localhost:3306/log
# 用户名
log4j.appender.jdbc.user=root
# 数据库插入语句
log4j.appender.jdbc.sql=insert into t_log values ('%m')
# 布局
log4j.appender.jdbc.layout=org.apache.log4j.PatternLayout
log4j.appender.jdbc.layout.ConversionPattern=%m%n
```

2、FileAppender

将日志记录在文件中

```
log4j.appender.myFile=org.apache.log4j.FileAppender
# 文件路径
log4j.appender.myFile.file=./log.txt
log4j.appender.myFile.layout=org.apache.log4j.PatternLayout
log4j.appender.myFile.layout.ConversionPattern=%m%n
```

3、ConsoleAppender

将日志记录在控制台

```
log4j.appender.cons=org.apache.log4j.ConsoleAppender
log4j.appender.cons.Target=System.out
log4j.appender.cons.layout=org.apache.log4j.PatternLayout
log4j.appender.cons.layout.ConversionPattern=%m%n
```

4、RollingFileAppender

将日志记录在文件中, 当文件达到一定大小之后, 会创建新文件

```
log4j.appender.myrFile=org.apache.log4j.RollingFileAppender
log4j.appender.myrFile.file=./log.txt
# 每个文件的最大容量 单位是b
log4j.appender.myrFile.maxFileSize=1024
# 最多备份文件的个数
log4j.appender.myrFile.maxBackupIndex=5
log4j.appender.myrFile.layout=org.apache.log4j.PatternLayout
log4j.appender.myrFile.layout.ConversionPattern=%m%n
```

5、DailyRollingFileAppender

每天一个备份文件

```
log4j.appender.mydFile=org.apache.log4j.DailyRollingFileAppender
# 前一天日志的备份文件的后缀
log4j.appender.mydFile.datePattern=yyyyMMdd
# 当天的日志的记录文件
log4j.appender.mydFile.file=./nl.txt
log4j.appender.mydFile.layout=org.apache.log4j.PatternLayout
log4j.appender.mydFile.layout.ConversionPattern=%m%n
```

九、Layout

为日志设置输出格式

1、SimpleLayout

格式

```
日志等级 - 日志内容
```

2、TTCCLayout

```
[main] ERROR com.HelloWorld - error
[线程] 日志等级 记录日志的类的包名.类名 - 日志内容
```

3、XMLLayout

```
<log4j:event logger="com.HelloWorld" timestamp="1529999963014" level="FATAL"
thread="main">
  <log4j:message><![CDATA[fatal]]></log4j:message>
</log4j:event>
```

4、HTMLLayout

HTML表格数据

5、PatternLayout

最复杂，功能最强大

使用 `ConversionPattern` 自定义输出格式

格式	作用
%p	日志等级
%d	日期 2018-6-26 16:05:19,555
%d{ymdhms自定义日期格式}	%d{yyyyMMddHHmmss}
%r	记录日志所消耗的时间
%C	产生日志的包名.类名
%t	产生日志的线程名
%m	日志消息
%l	产生日志的 包名.类名.方法名（类名:行数）
%L	行号

十、路径问题

- 对于J2SE项目（jar包）
 - 相对于项目目录
- 对于J2EE项目（war包）
 - 相对于 `TOMCAT_HOME/bin`