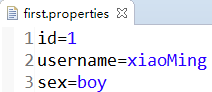
# 日志管理

## 一、Log4J

Log4J是java的日志管理器，用来记录应用系统的运行状态。

properties类型的文件存放的内容是key-value形式的。

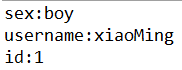


java.util包中的Properties类继承自Hashtable类，也是线程安全的， Properties也是Map中的一种，是Hashtable的子类。一般当key和value是文本类型，如字符串时使用Properties类，当key和value都是Object类型时，使用其它类型的Map。

**遍历properties文件的内容：**

**public** **class** Properties **extends** Hashtable<Object,Object>{}





对于Log4J来说，默认是在src目录下读取配置文件的。

### 1.定义Log4J的配置文件

#### ⑴配置根Logger：

**log4j.rootLogger=**[level],appenderName1,appenderName2....

其中level 是日志记录的优先级，从低到高分别为ALL、**DEBUG、INFO、WARN、ERROR**、FATAL、OFF。level越低，表示该logger越详细。定义了高级别的优先级，则不能打印出低级别的信息，程序会打印高于或等于所设置级别的日志，设置的日志等级越高，打印出来的日志就越少。

All：最低等级的，用于打开所有日志记录

Trace：是追踪，就是程序推进一下，可以用trace输出

Debug：指出细粒度信息事件对调试应用程序是非常有帮助的

Info：消息在粗粒度级别上突出强调应用程序的运行过程

Warn：输出警告及warn以下级别的日志

Error：输出错误信息日志

Fatal：输出每个严重的错误事件将会导致应用程序的退出的日志

OFF：最高等级的，用于关闭所有日志记录

appenderName表示日志信息输出到哪个地方。这里一般是写输出目的地的别名。

#### ⑵配置日志输出目的地Appender

**log4j.appender.**appenderName=目的地的全名称

log4j输出的目的地有以下几种：

**①org.apache.log4j.ConsoleAppender：控制台**

**②**org.apache.log4j.FileAppender：文件

**③**org.apache.log4j.DailyRollingFileAppender：每天产生一个日志文件

**④**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

**⑤**org.apache.log4j.WriterAppender：将日志信息以流格式发送到任意指定的地方

#### ⑶配置日志信息的布局

**log4j.appender.**appenderName**.layout**=布局的全名称

log4j有以下几种布局：

**①**org.apache.log4j.HTMLLayout：以HTML表格形式布局

**②org.apache.log4j.PatternLayout**：可灵活地指定布局模式

**③**org.apache.log4j.SimpleLayout：包含日志信息的级别和信息字符串

**④**org.apache.log4j.TTCCLayout：包含日志产生的时间、线程、类别等等信息

#### ⑷配置日志信息的输出格式

**log4j.appender.**appenderName**.layout.ConversionPattern**=指定输出格式

**%m** 输出代码中指定的消息

**%p** 输出优先级，即DEBUG，INFO，WARN，ERROR，FATAL

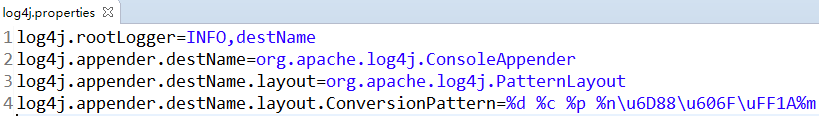
**%r** 输出自应用启动到输出该log信息耗费的毫秒数

**%c** 输出所属的类目，通常就是所在类的全名

**%t** 输出产生该日志事件的线程名

**%n** 输出一个回车换行符

**%d** 输出日志时间点的日期或时间。也可以指定输出的时间格式，如%d{yyy MMM dd HH:mm:ss,SSS}。



### 2.使用Log4J的步骤

**①**添加Log4J的项目到项目中，对于javaSE的项目，右键build Path-->Add to build path，对于javaWeb项目，直接放在项目的WEB-INFO目录下的lib包中即可。

**②**在src目录下面写一个log4j.properties的配置文件，不管在se项目还是web项目，都是放在src目录下的。

**③**给指定的类添加一个日志管理器，并添加相应的日志

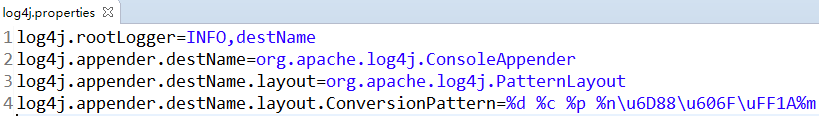
**Logger log=Logger.getLogger(指定类.class)**;

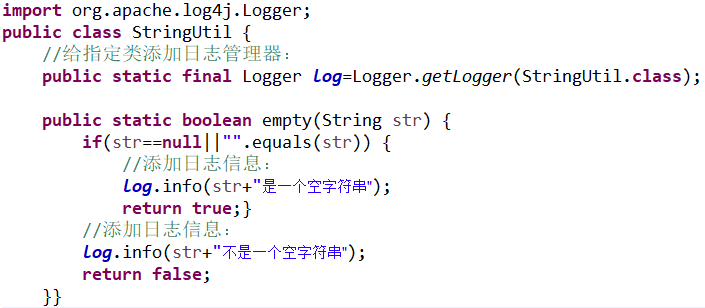
注意这个日志管理器是org.apache.log4j.Logger包下的，而不是java.util包下的。

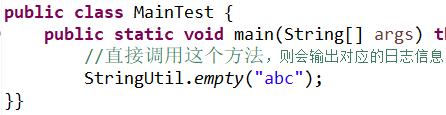
log**.info(“要显示的日志信息”)**、log**.debug(...)**，注意这里调用的方法一般和定义的日志级别对应，如日志级别为INFO，则调用info()方法。

### 3.例子

**javaSE例子：**



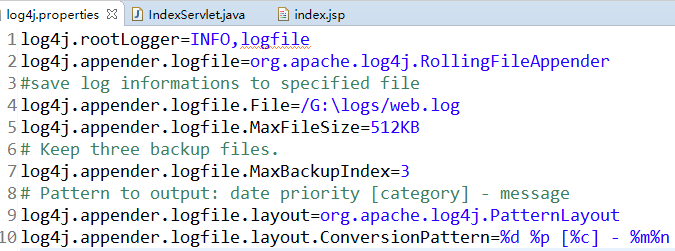


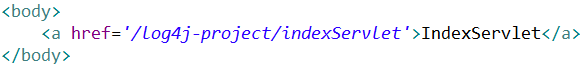




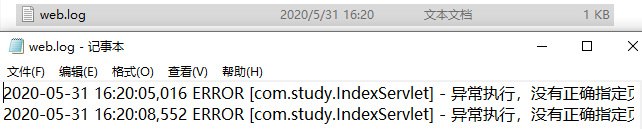
**javaWeb例子：**

将日志信息保存到指定的文件中：









## 二、log4j2

Apache Log4j 2是Log4j的升级版。在稳态日志记录期间，Log4j2 在独立应用程序中是无垃圾的，而在Web应用程序中是低垃圾的。Log4j 2中，可以通过代码或配置轻松定义自定义日志级别。支持Lambda表达式。

### 1.使用log4j2的步骤

**⑴**添加jar包。javaSE中直接添加log4j-api、log4j-core这两个jar包就可以，然后build Path。javaWeb中将log4j-web、log4j-api、log4j-core这三个包放在WEB-INFO目录下的lib目录中。

**⑵**在src目录下添加log4j2的配置文件。log4j2支持**xml、json、yaml**格式的配置文件。如配置文件为**log4j2.xml**。

**⑶**在指定类中添加日志监视器，并添加日志信息。

注意导包是导下面这两个包：

**import** org.apache.logging.log4j.LogManager;

**import** org.apache.logging.log4j.Logger;

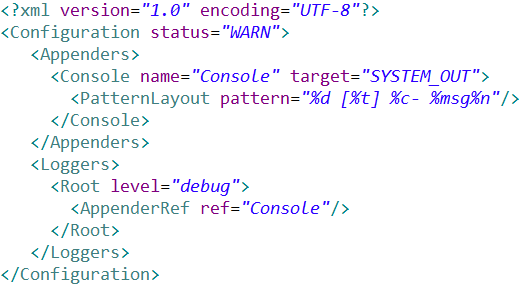
注意**log4j2**创建日志监视器采用的是下面这种方法：

**Logger log=LogManager.getLogger(指定类.class)**;

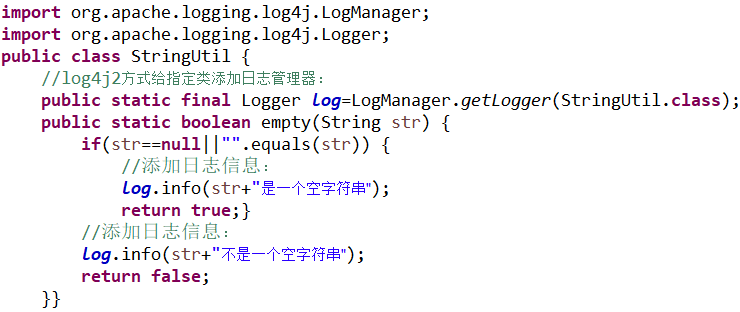
而**log4j**采用下面这种方式创建：

**Logger log=Logger.getLogger(指定类.class)**;

**一个简单的log4j2.xml文件：**



**log4j2方式创建日志监视器：**



### 2.log4j2.xml文件

**⑴<?xml version="1.0" encoding="UTF-8"?>**

**⑵**根节点**Configuration**：两个属性(**status**、**monitorinterval**)，两个子节点(**Appenders**、**Loggers**)

其中status用来指定log4j本身的打印日志的级别，monitorinterval用于指定log4j自动重新配置的监测间隔时间，单位是s，最小是5s。

**Appenders**节点，常见的有三种子节点：**Console**、**RollingFile**、**File**。这三种节点都有name属性和PatternLayout子节点。

name属性：指定 Appender 的名字；

PatternLayout节点：输出格式，不设置默认为%m%n；

**①Console**节点用来定义输出到控制台的Appender。

target属性：SYSTEM\_OUT 或SYSTEM\_ERR，一般只设置默认SYSTEM\_OUT。

**②File**节点用来定义输出到指定位置的文件的Appender。

fileName属性：指定输出日志的目的文件带全路径的文件名

append属性：false表示每次运行程序时，原来文件中的内容就会被清空。true则表示在原来文件的末尾添加日志信息。

**③RollingFile**节点用来定义超过指定大小自动删除旧的，创建新的Appender。

fileName属性：指定输出日志的目的文件带全路径的文件名

filePattern：指定新建日志文件的名称格式

**Loggers**节点，常见的有两种Root和Logger。

Root 节点用来指定项目的根日志，如果没有单独指定 Logger，那么就会默认使用该 Root日志输出。其level属性表日志输出级别。

**一个简单的log4j2.xml文件：**



一个比较完整的log4j2.xml文件：[log4j2.xml](file:///F:\1.java笔记（和盈）\4.javaEE基础\log4j2.xml)