## log4j使用

### 导包，log4j-1.2.17.jar。

### 创建配置文件log4j.properties

一般位于classpath下，配置Log4jContextListenner时，可为任意路径。

根Logger：log4j.rootLogger = [ level ] , appenderName, appenderName, …

level，日志信息的优先级：OFF，FATAL，ERROR，WARN，INFO，DEBUG，ALL。Log4j建议只使用四个级别，优先级从高到低分别是ERROR、WARN、INFO、DEBUG。通过在这里定义的级别，您可以控制到应用程序中相应级别的日志信息的开关。比如在这里定 义了INFO级别，则应用程序中所有DEBUG级别的日志信息将不被打印出来。

appenderName，日志信息的输出目的地，可以输出到多个地方。

日志信息的输出格式Appender：

##输出到控制台##

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

log4j.appender.stdout.Target = System.out

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

log4j.appender.stdout.layout.ConversionPattern = %d{ABSOLUTE} %5p %c{ 1 }:%L - %m%n

##输出到文件##

log4j.appender.E = org.apache.log4j.DailyRollingFileAppender

log4j.appender.E.File =E://logs/error.log ##可配为相对路径##

log4j.appender.E.Append = true ##追加##

log4j.appender.E.Threshold = ERROR ##阈值##

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

log4j.appender.E.layout.ConversionPattern = %-d{yyyy-MM-dd HH:mm:ss} [ %t:%r ] - [ %p ] %m%n

### 配置输出文件路径

1. 配置为绝对路径，如：E://logs/error.log
2. 通过jvm环境变量，配置相对路径。

a，spring的Log4jConfigListener。不配置webApp.root时，默认为webapp.root。

<context-param>

<param-name>webAppRootKey</param-name>

<param-value>webApp.root</param-value>

</context-param>

<context-param>

<param-name>log4jConfigLocation</param-name>

<param-value>classpath:log4j.properties</param-value>

</context-param>

<listener>

<listener-class>org.springframework.web.util.Log4jConfigListener</listener-class>

</listener>

b，使用已有jvm变量：例如：log4j.appender.logfile.File=${user.home}/logs/app.log

c，在项目启动时通过System.setProperty设置，通过实现ServletContextListener来解决。

### 使用log4j输出日志。？tomcat信息输出？

项目中通过logger输出指定级别信息；框架同理？

### Appender配置

Appender类型

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）

layout布局

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、类别等等信息）

格式化打印参数

%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，Windows平台为“rn”，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，比如：%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。举例：Testlog4.main(TestLog4.java:10)

### 链接

Log4j使用教程：<https://blog.csdn.net/yucaifu1989/article/details/70229909>

Log4j 日志文件存放位置设置：<https://blog.csdn.net/zwj1030711290/article/details/51890825>

多重web应用中webapp. root重用的问题：<https://blog.csdn.net/sdx1237/article/details/51836489>