https://blog.csdn.net/kesalin/article/details/556080

**log4j的配置笔记**

2005年12月19日 12:44:00 [飘飘白云](https://me.csdn.net/kesalin) 阅读数：1812

来源：[http://blogger.org.cn/blog/more.asp?name=zhanglincon&id=10457](http://blogger.org.cn/blog/more.asp?name=zhanglincon&id=10457" \t "_blank)

[zhanglincon](http://blogger.org.cn/blog/blog.asp?name=zhanglincon) 发表于 2005-12-8 9:57:52

|  |
| --- |
| (1)首先需要一个log4j-1.2.\*.jar包，jar包的位置：webapps/web/WEB-INF/lib/下 在log4j Web站点下载ZIP(Windows)或gzipped(基于Unix系统)格式的发行版本：[http://jakarta.apache.org/log4j/docs/download.html](http://jakarta.apache.org/log4j/docs/download.html" \t "_blank) 下载，将发行版解包，将创建logging-log4j-1.2.13目录，在此顶层目录dist/lib/目录下有log4j-1.2.13.jar文件。      (2)在web/WEB-INF/web.xml文件中做如下配置：配置一个Servlet映射，用来初始化log4j. <servlet>     <servlet-name>log4jInitServlet</servlet-name>     <servlet-class>com.airingbox.kzSuggestion.Log4jInitServlet</servlet-class>     <init-param>         <param-name>Log4j-init-file</param-name>         <param-value>/WEB-INF/classes/log4j.properties</param-value>     </init-param>     <load-on-startup>1</load-on-startup> </servlet> <servlet-mapping>     <servlet-name>log4jInitServlet</servlet-name>     <url-pattern>/log4jinitservlet</url-pattern> </servlet-mapping> *<load-on-startup>1</load-on-startup>是启动顺序，0：表示什么时候用什么时候启动，1：表示自动启动*   (3)编写初始化的servlet的类文件 package com.airinbox.kzSuggestion;  import org.apache.log4j.PropertyConfigurator; import javax.servlet.http.HttpServlet; import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse; import javax.servlet.ServletContext; import java.io.PrintWriter; import java.io.IOException;  /\*\*  \* 初始化log4j模块  \* <p>Title: init</p>  \* <p>Description: </p>  \* <p>Copyright: Copyright (c) 2005</p>  \* <p>Company: </p>  \* @author zhanglipeng  \* @version 1.0  \*/ public class Log4jInitServlet extends HttpServlet {    public  void init() {     String prefix =  getServletContext().getRealPath("/");     String file = getInitParameter("Log4j-init-file");     if(file != null) {       /\*\*        \*读取配置文件的方法        \*BasicConfigurator.configure ()： 自动快速地使用缺省Log4j环境。        \*PropertyConfigurator.configure ( String configFilename) ：读取使用Java        \*的特性文件编写的配置文件。        \*DOMConfigurator.configure ( String filename ) ：读取XML形式的配置文件。        \*/       PropertyConfigurator.configure(prefix+file);     }     System.out.println("LOG4J模块初始化成功!");   }    public  void doGet(HttpServletRequest req, HttpServletResponse res) {   } }   (4)编写要读取的属性文件 # 将信息输出到控制台 log4j.rootLogger=INFO,A1,A4 log4j.appender.A1=org.apache.log4j.ConsoleAppender log4j.appender.A1.layout=org.apache.log4j.PatternLayout log4j.appender.A1.layout.ConversionPattern=%-4r %-5p [%t] %37c %3x - %m%n # Available levels are DEBUG, INFO, WARN, ERROR, FATAL 他们级别最高的是fatal,就是说如果你在log4j.rootLogger=fatal,A1,A4 那么在日志中就不会出现比他级别低的debug,info,warn,error的日志信息，如果用info,就只是不会出现debug信息 这里的%p是指优先级，%m是指message本身，%n是指在新的一行输出    # 将信息输出到mylog.txt文件中  log4j.appender.A4=org.apache.log4j.FileAppender  log4j.appender.A4.file=/home/kzSuggestion/log/mylog.txt  log4j.appender.A4.layout=org.apache.log4j.PatternLayout  log4j.appender.A4.layout.ConversionPattern=[%-5p] %d{yyyy-MM-dd HH:mm:ss,SSS} method:%l%n%m%n     a.rootLogger：log书写器，A1,A4,A6,A8是日志存储器的名字，就是说你要将日志输出到控制台，输出到文件，还是每天产生一个文件，还是通过UDP输出到文件 log4j.appender.A4.file=/home/kzSuggestion/log/mylog.txt是要输出到文件的绝对路径    b.Appender - 日志目的地，把格式化好的日志信息输出到指定的地方去     c.ConsoleAppender - 目的地为控制台的Appender     d.FileAppender - 目的地为文件的Appender     e.RollingFileAppender - 目的地为大小受限的文件的Appender      f.Layout - 日志格式化器，用来把程序员的logging request格式化成字符串        PatternLayout - 用指定的pattern格式化logging request的Layout      g.Log4j提供的layout有以下几种：  　　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）  　　Log4J采用类似C语言中的printf函数的打印格式格式化日志信息，打印参数如下： %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“/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，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)   (5)这样就可以试一下了，在JSP或JAVA中输出log     a:导入org.apache.log4j.Logger和org.apache.log4j.propertyConfigurator包     b:调用方法:Logger logger = Logger.getLogger(你的类名.class.getName());        或者：Logger logger = Logger.getLogger(\*.jsp);        logger.info("come in \*\*\* "); //输出log     c:插入log信息的４种方法 　　Logger.debug ( Object message ) ; 　　Logger.info ( Object message ) ; 　　Logger.warn ( Object message ) ; 　　Logger.error ( Object message ) ;  (6)这时可以在控制台或在应用下/log/mylog.txt中看到log的输出信息  资料下载可以到log4j的主页:  Log4j主页：<http://jakarta.apache.org/log4j> |