

Log4J

Log4j es una biblioteca de **código abierto** desarrollada en **Java** por la **Apache Software Foundation** que ayuda a las **aplicaciones** de software a realizar un **seguimiento** de sus actividades pasadas. Básicamente, añades la librería de log4j y al utilizarla, escribes **trazas** de lo que está haciendo el programa en ese momento. Esto es especialmente útil para los programas ubicados en servidores – que no se detienen nunca – o en aquellos que no tienen un interfaz con el usuario. Si ocurre un error, basta con que leas el **fichero de trazas** y te enterarás de lo que estaba haciendo el programa cuando falló.

Instalado en log4J

Para que funcione, lo primero es instalar en nuestro proyecto las librerías del log4j correspondientes, como cualquier otro jar. Una vez hecho, hay que incluir también un fichero de configuración (un **properties** o un **xml**) para que el log4j sepa lo que tiene que hacer con las trazas. Estos ficheros deben de colocarse en la carpeta src, donde el propio log4j los buscará automáticamente. Un ejemplo sencillo de un **log4j.properties**:

```
1 # Define the root logger with appender file
2 log = c://trastero
3 log4j.rootLogger = DEBUG, FILE
4
5 # Define the file appender
6 log4j.appender.FILE=org.apache.log4j.FileAppender
7 log4j.appender.FILE.File=${log}//log.log
8
9 # Define the layout for file appender
10 log4j.appender.FILE.layout=org.apache.log4j.PatternLayout
11 log4j.appender.FILE.layout.conversionPattern=%m%n
```

Cosas a fijarse:

- 1) La ruta donde se colocará el fichero de trazas
- 2) Appender. Puedes definir varios. Son los ‘actualizadores’ de los ficheros de trazas y los hay de diferentes tipos. Incluso puedes escribir trazas de log en consola.
- 3) Conversion Pattern: Cómo quieres que aparezcan las trazas escritas en el fichero. Por ejemplo, si pones %d aparecerá con cada traza la fecha en la que se inscribió en el fichero, útil para saber cuándo se produjo un error.
- 4) El nivel del Log: está puesto a DEBUG, luego todas las trazas de prioridad DEBUG o más urgentes se mostrarán en el fichero de trazas, pero no las menos urgentes. Cuando no necesitemos las trazas, podemos subir el nivel de prioridad para que se ignoren ciertos mensajes.

Usando el log4j

Para usar el log4j, basta con crear una variable **log**.

```
private static final Logger log = LogManager.getLogger(Excepciones.class);
```

A partir de ahora, siempre que queramos tirar una traza, llamamos a sus métodos correspondientes. Este es un uso habitual de las trazas:

```
} catch (NullPointerException e) {  
    // Se ejecuta esto si se genera una NullPointerException  
    log.error("Se ha producido una NullPointerException");  
} catch (Exception e) {  
    // Se ejecuta esto si se genera cualquier otra Exception  
    log.error("Se ha producido una Exception");  
} finally {  
    // Esto se va a mostrar tanto si hay Excepcion como si no...  
    System.out.println("Volvemos al menu... ");  
    System.out.println(" ");  
}
```

Existen varios **niveles de prioridad** de las trazas:

- TRACE: El menos prioritario, se usa para trazas triviales. Se suelen borrar al final.
- DEBUG: Se usa para cuando estás aun haciendo pruebas del código. A veces se dejan en el código, pero se sube el nivel del log4j a ERROR para que las ignore.
- INFO: Se usa para dejar constancia de cosas importantes.
- WARN: Un aviso de que algo podría ser un error...
- ERROR: A ocurrido un error. Es el que se suele poner en las Excepciones. Se usa cuando se sabe a ciencia cierta que el programa no va a fallar.
- FATAL: Es el de mayor prioridad. Si te aparece esta traza, es que el programa ha fallado de forma catastrófica.
- ALL: Se muestran todas las trazas, sean las que sean-
- OFF: No se muestra ninguna.

AVISO

Log4j es muy, muy, muy ineficiente en cuanto a rendimiento. Incluso cuando lo pones a que no escriba nada. La mayoría de los equipos de desarrollo no quieren verlo ni en pintura, por lo que es habitual eliminarlo cuando se entrega al cliente.

... además, siempre hay algún espabilado que se pone a escribir claves de acceso y cosas así en los ficheros de trazas...

... para matarlo...