LOG4J JAVA

minsait
by Indra

LOG4J

- El registro de eventos en una aplicación, es un proceso empleado en todas las aplicaciones.
- Es una práctica que permite descubrir por qué instrucciones está pasando nuestra aplicación y poder deducir así si ha habido algún problema o no

LOG4J

- LOG4j, es una librería para JAVA, que se ha convertido en un estándar cuando necesitamos registrar algún evento, dada su potencia y facilidad de uso

LOG4J-Properties

- Para parametrizar el comportamiento del sistema de log, se emplea un fichero de propiedades, que indica dónde, cómo y cuando se comportará nuestra LOG

Patrón Singleton

- El patrón de diseño Singleton en POO, es un modelo de diseño por el cual, una clase, sólo va a tener una única instancia
- Es especialmente útil y aplicable al caso de Logs, ya que varios procesos podrían intentar escribir en el mismo archivo a la vez

Log Properties

- Hay 3 conceptos que manejar a la hora de definir el log:
  - Logger
  - Appender
  - Layout

Logger

- Podemos tener definidos varias formas de log, según nuestro entorno o necesidad o destino de nuestro registro
- Para cada uno, definimos un logger
- Además, existe un logger principal o root

Appender

- Un appender es una salida del destino de nuestro mensaje. Podemos definir un fichero local, uno remoto, una base de datos
- Sólo tendremos que elegir la clase apropiada

Appender

- ConsoleAppender: Indicaremos esa clase para mostrar los mensajes por pantalla
- FileAppender: Redirige los mensajes a un archivo

Layout

- Mediante Layout, definimos el estilo que queremos para imprimir nuestros mensajes.

Layout

- Los estilos principales son

%m: muestra el mensaje.
%p: muestra el nivel de prioridad.
%t: muestra el nombre del thread que loguel el evento.
%n: salto de línea
%d: muestra la fecha del vento, que se le puede dar un formato determinado, por ejemplo %d{HH:mm:ss,SSS} o %d{dd MMM yyyy HH:mm:ss,SSS}.
%C: muestra el nombre de la clase que logueo el evento.
%M: muestra el método de la clase que logueo el evento.
%L: muestra el número de línea donde se logueo el evento.

NIVELES de prioridad

EL SISTEMA DE LOG, NOS OFRECE DISTINTOS NIVELES DE PRIORIDAD, QUE SE CORRESPONDEN CON LA URGENCIA O IMPORTANCIA DEL MENSAJE QUE VAYAMOS A REGISTRAR

NIVELES de prioridad

- Hay definidos 6 niveles. De menor a mayor importancia, son:
  - TRACE: Para mostrar mensajes sin importancia
  - DEBUG: Para la fase de pruebas
  - INFO: Informamos del paso de la aplicación
  - WARN: Aviso, va bien, pero no del todo
  - ERROR: Algo grave pasa
  - FATAL: Parada de programa

Setting

- Debemos descargar la librería apropiada de la versión y añadirla a nuestro classpath
- La ubicación por defecto del fichero log4j.properties es la carpeta /bin

Práctica

- Dada la clase de ejemplo, crear una nueva clase log con el patrón singleton que tenga un appender que nos permita escribir en fichero

Ayudarse de la documentación oficial para esta versión