




Archivos Log

Jyoti Rospigliosi
Miguel Angel Herrera
Adrian Miranda



Agenda

- *Qué son Archivos Log y para qué sirven.*
- *Archivos Log en java*
- *Frameworks Log*
- *Ejemplo*



¿Qué son y para qué sirven los log files o los archivos de registro?

Un fichero o archivo log tiene como principal objetivo definir el comportamiento de los programas. Tener un log donde se reporten los eventos y los errores nos permite desarrollar un mayor análisis de problemas y nos ahorra tiempo en caso estos se presenten. Estos “logs” son gestionados por prácticamente todas las aplicaciones, servidores, bases de datos y sistemas de manera automática y permiten controlar (de forma centralizada) todos los procesos relevantes.

Archivos Log en Java

Cómo conseguir los archivos log:

Windows 10:

En el campo de búsqueda, escriba %TEMP%

Windows 8:

Vaya a inicio > escriba: %TEMP% en el campo Iniciar Búsqueda

Windows Vista y Windows 7

Vaya a INICIO y escriba %Temp% o en la siguiente ruta:
C:\Usuarios\{nombre de usuario}\AppData\Local\Temp

Windows XP:

Inicio > Ejecutar> Escribir %TEMP%

Buscar los siguientes archivos en la carpeta Temp:

jreMSI.log

JavaDeployReg.log

java_install.log

java_install_reg.log

jusched.log



Frameworks


tinylog

Simplifique su registro con tinylog. Este marco de registro ligero se puede utilizar con Java, Kotlin, Scala y cualquier otro lenguaje JVM. tinylog es de código abierto y se ejecuta en cualquier plataforma Java (Java 6 o posterior), así como en Android (API Nivel 1 o superior).

[Empezando](#)[Descarga tinylog](#)

Última versión: 2.2.1 / 2.3.0-M1

Tiny Log




```
import org.tinylog.Logger;

public class Application {

    public static void main(String[] args) {
        Logger.info("Hello World!");
    }


}
```

Evita el código repetitivo



```
writer      = file  
writer.file = application.log  
writer.format = {class}.{method}() {message}  
level       = info
```

Configurable



Como su nombre lo indica, tinylog es un marco de registro ligero. tinylog en sí consta de solo dos archivos JAR (uno para la API y el otro para la implementación) sin dependencias externas. Ambos archivos JAR tienen un tamaño combinado de solo 157 KB.



157 KB

Fuente abierta

tinylog es de código abierto y tiene licencia Apache License 2.0 . Esto significa que el marco de registro se puede utilizar de forma gratuita y sin restricciones, incluso en software comercial.





LOG4J

Log4J



Log4J tiene 3 componentes principales

Loggers

Appenders

Layout

LOGGERS

Niveles	Descripción
ALL	Tiene el rango más bajo posible y está destinado a activar todos los registros.
DEBUG	Designa eventos informativos muy detallados que son más útiles para depurar una aplicación.
INFO	Designa mensajes informativos que destacan el progreso de la aplicación en el nivel superior.
WARN	Designa información sobre comportamiento del sistema defectuoso e inesperado.
ERROR	Designa eventos de error que aún podrían permitir que la aplicación continúe ejecutándose.
FATAL	Designa eventos de error muy graves que presumiblemente llevarán a la aplicación a abortar.
OFF	Tiene el rango más alto posible y está diseñado para desactivar el registro.

Jerarquía de Salida del LOGGER

Mensajes Desplegados

Nivel de Registro	DEBUG	INFO	WARN	ERROR	FATAL
	DEBUG				
	INFO				
	WARN				
	ERROR				
	FATAL				
	ALL				
OFF					

APPENDERS

Tipos	Descripción
Console	Agrega eventos de registro a System.out o System.err utilizando un diseño especificado por el usuario. El objetivo predeterminado es System.out.
File	Agrega eventos de registro a un archivo.
DailyRollingFile	Extiende FileAppender para que el archivo subyacente se desplace a una frecuencia elegida por el usuario.
RollingFile	Extiende FileAppender para hacer una copia de seguridad de los archivos de registro cuando alcanzan un tamaño determinado.
Writer	Agrega eventos de registro a un Writer o un OutputStream según la elección del usuario.
SMTP	Envía un correo electrónico cuando ocurre un evento de registro específico, típicamente en errores o errores fatales.
Socket	Envía objetos LoggingEvent a un servidor de registro remoto, generalmente un Nodo Socket
SocketHub	Envía objetos LoggingEvent a un conjunto de servidores de registro remoto, generalmente un Nodo Socket
SysLog	Envía mensajes a un proceso demonio remoto
Telnet	Es un log4j appender que se especializa en escribir en un socket de solo lectura

LAYOUTS

Tipo	Descripción
CSV	Este diseño crea registros de valores separados por comas (CSV).
GELF	Presenta eventos en el formato de registro ampliado de Graylog.
HTML	HtmlLayout genera una página HTML y agrega cada LogEvent a una fila en una tabla.
JSON	Anexa una serie de eventos como cadenas JSON serializado como bytes.
Pattern	Un diseño flexible configurable con cadena de patrones. El objetivo de esta clase es formatear un LogEvent y devolver los resultados. El formato del resultado depende del patrón de conversión.

↑ APACHE LOG4J™ 2

Acerca de

Descargar

Javadoc

Artefactos de Maven,

Ivy y Gradle

Dependencias de tiempo
de ejecución

Registro de cambios

Preguntas más
frecuentes

Actuación

Artículos y tutoriales

Seguridad

Apoyo

Apache Log4j 2

Apache Log4j 2 es una actualización de Log4j que proporciona mejoras significativas sobre su predecesor, Log4j 1.x, y proporciona muchas de las mejoras disponibles en Logback al tiempo que soluciona algunos problemas inherentes a la arquitectura de Logback.



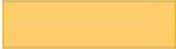





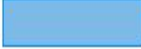

Características

Separación de API

La API para Log4j es independiente de la implementación, lo que deja en claro para los desarrolladores de aplicaciones qué clases y métodos pueden utilizar, al tiempo que garantizan la compatibilidad con versiones posteriores. Esto permite al equipo de Log4j mejorar la implementación de forma segura y compatible.

La API de Log4j es una fachada de registro que, por supuesto, se puede usar con la implementación de Log4j, pero también se puede usar frente a otras implementaciones de registro, como Logback. La API de Log4j tiene varias ventajas sobre SLF4J: 1. La API de Log4j admite [mensajes de registro](#) en lugar de solo Strings. 2. La API de Log4j admite expresiones lambda. 3. La API de Log4j proporciona muchos más métodos de registro que SLF4J. 4. Además del formato de "registro parametrizado" admitido por SLF4J, la API de Log4j también admite eventos que utilizan la sintaxis `java.text.MessageFormat` y mensajes de estilo `printf`. 5. La API de Log4j proporciona un método `LogManager.shutdown()`. La implementación de registro subyacente debe implementar la interfaz `Terminable` para que el método tenga efecto. 6. Otras construcciones como `Marcadores`, `Niveles de registro` y `ThreadContext` (también conocido como `MDC`) son totalmente compatibles.

Log4J 2

Marco de referencia		Entradas de registro procesadas por segundo
tinylog 2.1.0		85,115
with writing thread		427,774
tinylog 1.3.6		89,945
with writing thread		380,230
Log4j 2.12.1		46,490
with async logger		94,692
Log4j 1.2.17		43,703
with async appender		80,111
Logback 1.2.3		42,894
with async appender		72,881
java.util.logging		55,892



Bibliografía

- https://java.com/es/download/help/log_files.html
- <https://tinylog.org/v2/>
- <https://logging.apache.org/log4j/2.x/>
- <http://logback.qos.ch/>