

Log4J

Framework open source
hébergé par la fondation Apache
<https://logging.apache.org/log4j/2.x/>



XH

1

Les frameworks

- Choix stratégique pour un projet
- LOG4J est utilisé par JBoss, WildFly, Hibernate, ...
- Un comité pour gérer la destinée de Log4J
 - the Apache Logging Services Project Management Committee

2



Les frameworks (aparté)

- Quand elles recrutent, les entreprises parlent « framework »
 - Symphony pour PHP
 - Spring pour Java
 - Log4J pour Java
- LOG4J à mettre sur votre CV

3



Log4J v2

- Pour plus d'exemples,
 - https://koor.fr/Java/Tutorial/java_logging_log4j2.wp
- Pour plus d'exemples, consultez le site officiel:
 - <https://logging.apache.org/log4j/2.x/>
- Ainsi que, la page wikipédia anglaise:
 - <https://en.wikipedia.org/wiki/Log4j>
- Log4J v1.2 n'évolue plus
 - On August 5, 2015 the Apache Logging Services Project Management Committee announced that Log4j 1 had reached end of life and that users of Log4j 1 are recommended to upgrade to Apache Log4j 2.
- Log4J v2 basé sur Java Logging API
 - Java Logging API ([JSR 47](#),...)
 - Module [java.logging](#) Since:9
 - Package [java.util.logging](#)

XH

4



Alternatives en Java

- Java Logging
- Jlog
- Protomatter
- SLF4J
- LogBack

XH

5



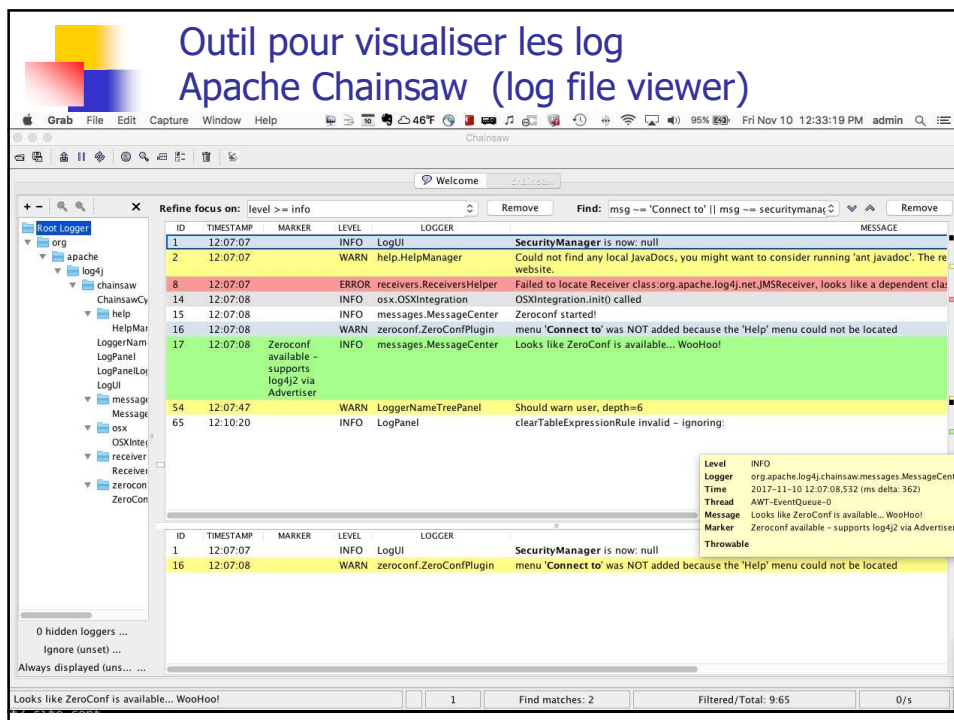
Un journal pour votre applicatif ?

- Applicatif tournant en tâche de fond
- Les journaux : fichier éditable avec l'extension .log
- Horodatage des log, un degré de gravité

```
2003-57-29 05:57:31,271 WARN org.progx - exemple4.java:16 - Starting application
2003-57-29 05:57:31,302 INFO org.progx - exemple4.java:9 - Application is alive.
2003-57-29 05:57:31,305 WARN org.progx - exemple4.java:18 - Exiting application
2003-00-29 06:00:23,291 WARN org.progx - exemple4.java:16 - Starting application
2003-00-29 06:00:23,323 INFO org.progx - exemple4.java:9 - Application is alive.
2003-00-29 06:00:23,328 WARN org.progx - exemple4.java:18 - Exiting application
2003-00-29 06:00:33,469 WARN org.progx - exemple4.java:16 - Starting application
2003-00-29 06:00:33,501 INFO org.progx - exemple4.java:9 - Application is alive.
2003-00-29 06:00:33,505 WARN org.progx - exemple4.java:18 - Exiting application
2003-00-29 06:00:38,326 WARN org.progx - exemple4.java:16 - Starting application
2003-00-29 06:00:38,357 INFO org.progx - exemple4.java:9 - Application is alive.
2003-00-29 06:00:38,362 WARN org.progx - exemple4.java:18 - Exiting application
```

XH

6



Ecrire un log en java

```
//Au début de chaque classe, créez un attribut "logger"
...
Logger logger = Logger.getLogger("org.progx.BackupDavis");

...
//Ensuite, vous pouvez utiliser les méthodes écrivant le
journal

logger.info("Démarrage");
...
logger.log(Level.INFO, "Démarrage");
```

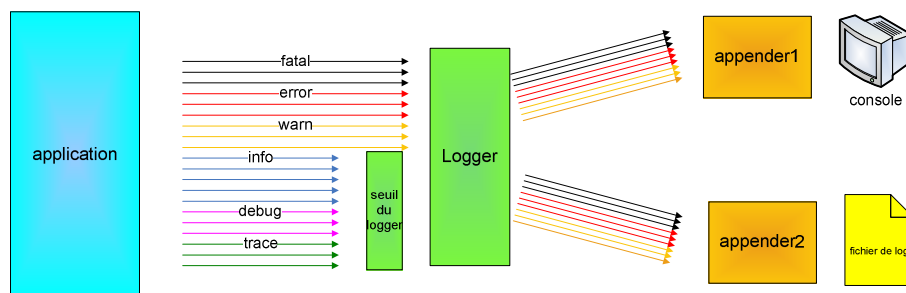
XH

8



Schéma des LogEvent

- Ici le seuil (level) est réglé à WARN
 - On laisse passer WARN, ERROR, FATAL
 - Écrit dans le fichier de configuration



XH

9



Quelques concepts (non exhaustif)

- Source: https://www.tutorialspoint.com/log4j/log4j_architecture.htm
- **Logger Object**
 - The top-level layer is the Logger which provides the Logger object. The Logger object is responsible for capturing logging information and they are stored in a namespace hierarchy.
- **Appender Object**
 - This is a lower-level layer which provides Appender objects. The Appender object is responsible for publishing logging information to various preferred destinations such as a database, file, console, UNIX Syslog, etc.
- **Layout Object**
 - The layout layer provides objects which are used to format logging information in different styles. It provides support to appender objects before publishing logging information.
 - Layout objects play an important role in publishing logging information in a way that is human-readable and reusable.

XH

10



Concepts (suite)

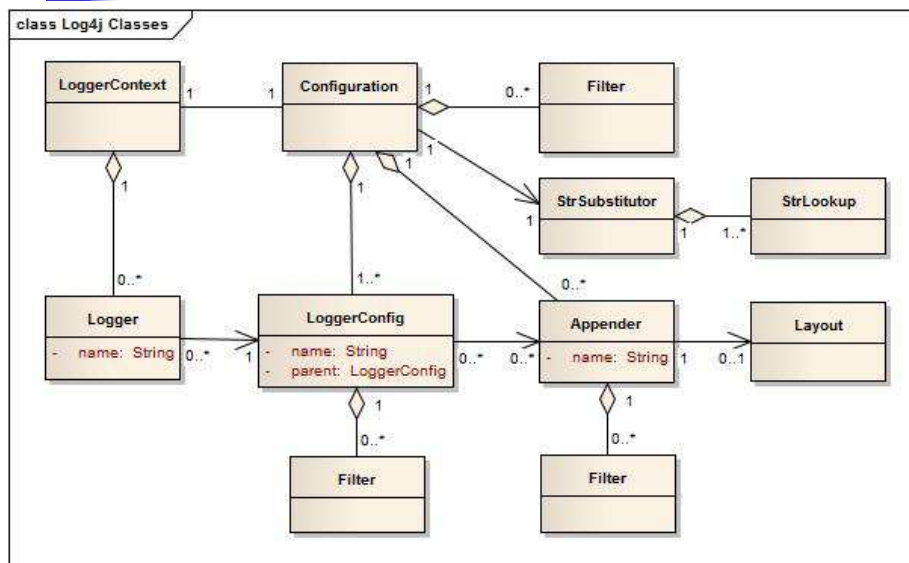
- Source: https://www.tutorialspoint.com/log4j/log4j_architecture.htm
- Level Object
 - The Level object defines the granularity and priority of any logging information. There are seven levels of logging defined within the API: OFF, DEBUG, INFO, ERROR, WARN, FATAL, and ALL.
- ...

XH

11



Diagramme de classes avec les concepts Log4J v1.2



XH

12



Exemples de fichiers de configuration XML

- Attention, le fichier de configuration doit être dans le classpath c.a.d. sous src !
 - (ou bien dans le source-folder resource)
- Au format **XML**, JSON, YAML, properties
- Voir les exemples sur le site officiel rubrique configuration:
 - <https://logging.apache.org/log4j/2.x/manual/configuration.html>
- Ou la page wikipédia
 - <https://en.wikipedia.org/wiki/Log4j>

XH

13



wikipédia Log4J v2 (page anglaise)

- Voir slide suivant

WIKIPEDIA
The Free Encyclopedia

Log4j

From Wikipedia, the free encyclopedia

Apache **Log4j** is a Java-based logging utility. It was originally written by Ceki Gülcü and is part of the **Apache Logging Services** project of the Apache Software Foundation. Log4j is one of several Java logging frameworks.

Gülcü has since started the SLF4J and Logback^[3] projects, with the intention of offering a successor to Log4j.

The Apache Log4j team has created a successor to Log4j 1 with version number 2.^[4] Log4j 2 was developed with a focus on

Apache Log4j	
Developer(s)	Apache Software Foundation
Initial release	January 8, 2001; 18 years ago ^[1]
Stable release	2.13.0 / December 11, 2019; 6 days ago ^[2]
Repository	Log4j Depository ^[2]

XH

14