

 eBook Gratuit

# APPRENEZ

---

## log4j

eBook gratuit non affilié créé à partir des  
**contributeurs de Stack Overflow.**

#log4j

# Table des matières

À propos.....	1
<b>Chapitre 1: Démarrer avec log4j.....</b>	<b>2</b>
Remarques.....	2
Versions.....	2
Exemples.....	2
Installation et configuration.....	2
Installation.....	2
Configuration.....	3
Maven.....	3
springboot avec maven.....	3
lierre.....	4
gradle.....	4
<b>Chapitre 2: configuration.....</b>	<b>5</b>
Exemples.....	5
Fichier de propriétés Log4j.....	5
Résoudre le problème d'exécution avec la configuration de log4j.....	5
<b>Chapitre 3: enregistreurs log4j.....</b>	<b>7</b>
Introduction.....	7
Exemples.....	7
Log4cxx.....	7
Log4cplus.....	7
Log4cpp.....	7
<b>Chapitre 4: log4j journalise la personnalisation.....</b>	<b>9</b>
Exemples.....	9
Fichier de configuration.....	9
Configuration.....	9
Découverte de la configuration.....	9
Configuration XML.....	9
Exemple XML.....	9
<b>Crédits.....</b>	<b>11</b>

---

# À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [log4j](#)

It is an unofficial and free log4j ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official log4j.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

# Chapitre 1: Démarrer avec log4j

## Remarques

Cette section fournit une vue d'ensemble de ce qu'est log4j et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets dans log4j, et établir un lien avec les sujets connexes. La documentation de log4j étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

## Cycle de vie Log4j

Log4j 1.x est en fin de vie à compter du 5 août 2015. [1] [2]. [Apache Log4j 2](#) est le successeur de Log4j 1.x.

[1] <https://logging.apache.org/log4j/1.2/>

[2] [https://blogs.apache.org/foundation/entry/apache\\_logging\\_services\\_project\\_announces](https://blogs.apache.org/foundation/entry/apache_logging_services_project_announces)

## Versions

Version	Remarquer	Date de sortie
2.8	dernière version	2017-01-21
2.7		2016-10-02
2.6.2		2016-07-09
2.4	Log4j 2.4 et versions ultérieures nécessite Java 7	2015-09-20
2.3.6	dernière version qui supporte java 6	2015-05-15
2.0	première version stable de la branche 2.x. Brise la compatibilité api. Utilisez le pont: log4j-1.2-api.jar	2014-07-01
1.2.17	EOF log4j branche 1.x	2015-08-05

## Exemples

### Installation et configuration

## Installation

L'installation de Log4j2 est aussi simple que de mettre log4j2 jar dans classpath de l'application. Bien que vous souhaitiez [personnaliser les journaux de sortie via un fichier de configuration supplémentaire](#)

## Configuration

### Maven

Pour ajouter log4j au projet dans maven, ajoutez sa dépendance: Dans pom.xml, ajoutez la dépendance suivante:

```
<dependencies>
  <dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-api</artifactId>
    <version>${log4j2.version}</version>
  </dependency>
  <dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-core</artifactId>
    <version>${log4j2.version}</version>
  </dependency>
</dependencies>
```

### springboot avec maven

Spring-boot est un framework couramment utilisé pour les applications Web. Il prend en charge la configuration automatique pour de nombreuses fonctionnalités, y compris la façade de journalisation comme log4j2. Pour ajouter log4j2 à votre projet spring-boot, assurez-vous d'exclure la façade de journalisation par défaut: commons-logging. Log4j sera utilisé quand il ne fait que consigner la façade sur classpath.

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
    <!-- exclude spring-boot java commons logging in favour of log4j2 -->
    <exclusions>
      <exclusion>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-logging</artifactId>
      </exclusion>
    </exclusions>
  </dependency>
  <!-- add log4j2 to spring-boot: -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-log4j2</artifactId>
  </dependency>
```

Notez qu'il n'y a pas de version dans l'extrait ci-dessus. C'est parce que le projet hérite de la version du parent. Assurez-vous également d'hériter de spring-boot-starter-parent, en ajoutant:

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>1.3.3.RELEASE</version>
</parent>
```

## lierre

Dans ivy.xml, ajoutez la dépendance suivante:

```
<dependencies>
  <dependency org="org.apache.logging.log4j" name="log4j-api" rev="${log4j2.version}" />
  <dependency org="org.apache.logging.log4j" name="log4j-core" rev="${log4j2.version}" />
</dependencies>
```

## gradle

Dans votre fichier .gradle:

```
dependencies {
  compile group: 'org.apache.logging.log4j', name: 'log4j-api', version: '2.6.2'
  compile group: 'org.apache.logging.log4j', name: 'log4j-core', version: '2.6.2'
}
```

Lire Démarrer avec log4j en ligne: <https://riptutorial.com/fr/log4j/topic/5845/demarrer-avec-log4j>

# Chapitre 2: configuration

## Exemples

### Fichier de propriétés Log4j

Vous trouverez ci-dessous un fichier de configuration pour log4j. Log4j2 peut utiliser la même syntaxe, mais il existe différentes classes d'appenders:

```
log4j.rootLogger=INFO, FOO

## ConsoleAppender
log4j.appender.CA=org.apache.log4j.ConsoleAppender
log4j.appender.CA.layout=org.apache.log4j.PatternLayout
log4j.appender.CA.layout.ConversionPattern= %d{hh:mm:ss,SSS} [%t] %-5p %c %x - %m%n

## FileAppender
log4j.appender.FOO=org.apache.log4j.RollingFileAppender
log4j.appender.FOO.File=${catalina.home}/logs/app.log
log4j.appender.FOO.Append=true
log4j.appender.FOO.layout=org.apache.log4j.PatternLayout
log4j.appender.FOO.layout.ConversionPattern= %d{hh:mm:ss,SSS} [%t] %-5p %c %x - %m%n

## attaching appender to specific package:
log4j.logger.com.example.package=INFO, CA
```

Directive `log4j.rootLogger` définit le niveau de journalisation et l'appendeur pour toute classe ne répondant pas aux critères de `logger`. Notez que le `name` appendeur est défini après le mot "appendeur".

### Résoudre le problème d'exécution avec la configuration de log4j

Les utilisateurs peuvent rencontrer le problème suivant:

```
log4j:WARN No appenders could be found for logger (dao.hsqlmanager).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
```

Cela peut être dû en partie au fait que le fichier `log4j.properties` ou `.xml` ne se trouve pas dans le projet lui-même. (Cela peut se produire lorsque vous expédiez un outil, où l'outil / JAR se trouve dans un répertoire et toute la configuration est un autre répertoire).

Vous devrez ensuite spécifier le chemin d'accès au fichier `log4j.properties` ou `.xml`. Dans l'utilitaire de ligne de commande,

```
java -Dlog4j.configuration=file:///path/To/log4j.properties YourProject.jar
```

ou si vous avez un script pour exécuter l'outil, vous pouvez ajouter

```
-Dlog4j.configuration=file:///path/To/log4j.properties
```

à l'endroit où vous faites l'action équivalente de la version en ligne de commande. Ce n'est pas que `log4j.configuration` soit spécifié dans un format d'URL préfixé par `file:///` .

Lire configuration en ligne: <https://riptutorial.com/fr/log4j/topic/7510/configuration>



---

# Chapitre 3: enregistreurs log4j

## Introduction

Avantages et inconvénients des différents enregistreurs pouvant être utilisés pour créer un journal au format log4j à afficher à l'aide du visualiseur Log4View.

Je vais examiner 3 enregistreurs dans cet article, Log4cxx, Log4cplus et Log4cpp.

## Exemples

### Log4cxx

<https://logging.apache.org/log4cxx/>

- en cours d'incubation - il n'y a pas de version officielle
- mise à jour / corrections de bugs une fois au cours des 12 dernières années, dernière version 2008
- L'utilisateur peut sélectionner différents niveaux LogLevels - TRACE, DEBUG, INFO, WARN, ERROR et FATAL
- Loggers hiérarchiques
- il est possible de se connecter de manière asynchrone
- prend en charge plusieurs appenders
- l'utilisateur peut choisir d'activer ou de désactiver l'enregistreur
- le journal peut être envoyé à des cibles de sortie différentes et multiples
- formats de sortie sélectionnés par l'utilisateur
- bien documenté
- est sous licence Apache, une licence open source certifiée par l'Open Source Initiative

### Log4cplus

<https://sourceforge.net/projects/log4cplus/>

- mises à jour / corrections de bugs - la dernière version date de janvier 2016
- L'utilisateur peut sélectionner différents niveaux LogLevels - Loggers hiérarchiques TRACE, DEBUG, INFO, WARN, ERROR et FATAL
- prend en charge les applications multithread mais n'est pas sûr d'être utilisé à partir de gestionnaires de signaux asynchrones
- format de sortie sélectionné par l'utilisateur: SimpleLayout, PatternLayout, TTCCLayout
- prend en charge plusieurs enregistreurs
- pas bien documenté
- est sous licence Apache License V2.0

### Log4cpp

<https://sourceforge.net/projects/log4cpp/>

- les corrections de bogues ont lieu environ une fois par an, la dernière version datant d'avril 2015
- prend en charge les applications multithread • aucune documentation claire n'existe
- est sous licence GNU Lesser General Public License (LGPL) à partir de la version 0.2.1, avant que celles-ci ne soient publiées sous licence GPL.

Lire enregistreurs log4j en ligne: <https://riptutorial.com/fr/log4j/topic/10570/enregistreurs-log4j>

---

# Chapitre 4: log4j journalise la personnalisation

## Exemples

### Fichier de configuration

## Configuration

Le fichier de configuration Log4j peut être dans l'un des formats suivants:

- JSON
- YAML
- propriétés (fichier texte)
- XML

### Découverte de la configuration

1. Log4j inspectera la propriété système `log4j.configurationFile` et, si elle est définie, tentera de charger la configuration.
2. Si aucune propriété système n'est définie, log4j recherchera `log4j2-test.properties` dans le classpath.
3. Si aucun fichier de ce type n'est trouvé, log4j recherchera `log4j2-test.yaml` ou `log4j2-test.yml` dans le classpath.
4. Si aucun fichier de ce type n'est trouvé, log4j recherchera `log4j2-test.json` ou `log4j2-test.jsn` dans le classpath.
5. Si aucun fichier de ce type n'est trouvé, log4j recherche `log4j2-test.xml` dans le classpath.
6. Si un fichier de test ne peut pas être localisé, le fichier log4j recherche le `log4j2.properties` sur le `log4j2.properties` de `log4j2.properties`.
7. Si un fichier de propriétés ne peut pas être localisé, le fichier log4j recherche le `log4j2.yaml` ou `log4j2.yml` sur le `log4j2.yml` de `log4j2.yml`.
8. Si un fichier YAML ne peut pas être localisé, le fichier log4j recherche le `log4j2.json` ou le `log4j2.jsn` sur le `log4j2.jsn` de `log4j2.jsn`.
9. Si un fichier JSON ne peut pas être localisé, log4j essaiera de localiser `log4j2.xml` sur le `log4j2.xml` de `log4j2.xml`.
10. Si aucun fichier de configuration ne peut être localisé, la configuration par défaut sera utilisée. Cela entraînera la sortie de journalisation pour aller à la console.

### Configuration XML

## Exemple XML

La configuration ci-dessous configure deux appenders (sortie de journal). Les premiers journaux

sur la sortie du système standard (console) et les autres journaux à classer. Dans cet exemple, l'emplacement du fichier peut être défini de manière statique dans la configuration ( `file` appender) ou dynamiquement via la fonctionnalité de filtrage maven ( `<Property name="APPENDER">` ). Les fichiers dans le fichier seront emballés par jour. Le format de Conversion Pattern format de ligne de journal est défini comme une variable.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- 'status' sets log level for parsing configuration file itself -->
<Configuration status="INFO">
  <Properties>
    <!-- Sets variable PID, if it's not present in log4j context will take value as below: -->
    <Property name="PID">????</Property>
    <!-- Sets variable 'LOG_PATTERN', defining how log line will look like -->
    <Property name="LOG_PATTERN">%clr{%d{yyyy-MM-dd HH:mm:ss.SSS}}{faint} %clr{%5p}
%clr{%s}{magenta} %clr{%X{usr}}{green} %clr{---}{faint}%clr{[%15.15t]}{faint} %clr{%-
40.40c{1.}:%l}{cyan} %clr{:}{faint} %m%n%wEx</Property>
    <!-- LOG_DIR may be set by maven filtering feature: -->
    <Property name="LOG_DIR">@logging.path@</Property>
    <!-- APPENDER may be set by maven filtering feature, to set 'console' or 'file' appender:
-->
    <Property name="APPENDER">@logging.default.appender@</Property>
  </Properties>
  <Appenders>
    <!-- Sets console output: -->
    <Console name="console" target="SYSTEM_OUT">
      <PatternLayout pattern="${LOG_PATTERN}"/>
    </Console>
    <!-- Sets output to file, and names it 'file' -->
    <RollingRandomAccessFile append="true" fileName="${LOG_DIR}/log4j2.log"
      filePattern = "${LOG_DIR}/log4j2.%d{yyyy-MM-dd}.nr%i.log.gz" name="file">
      <PatternLayout>
        <Pattern>${LOG_PATTERN}</Pattern>
      </PatternLayout>
      <Policies>
        <TimeBasedTriggeringPolicy />
      </Policies>
    </RollingRandomAccessFile>
  </Appenders>
  <Loggers>
    <!-- Sets debug for Class 'com.example.package.Clazz', and attaches to file 'file' -->
    <Logger name="com.example.package.Clazz" level="debug">
      <AppenderRef ref="file"/>
    </Logger>
    <!-- Sets 'info' for package 'org.springframework' -->
    <Logger name="org.springframework" level="info" />
    <Root level="WARN">
      <AppenderRef ref="${APPENDER}"/>
    </Root>
  </Loggers>
</Configuration>
```

Lire log4j journalise la personnalisation en ligne: <https://riptutorial.com/fr/log4j/topic/6848/log4j-journalise-la-personnalisation>

# Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec log4j	<a href="#">Augustin Ghauratto</a> , <a href="#">bn.</a> , <a href="#">Community</a> , <a href="#">Michaël K.</a>
2	configuration	<a href="#">Augustin Ghauratto</a> , <a href="#">Ishnark</a>
3	enregistreurs log4j	<a href="#">Merav Kochavi</a>
4	log4j journalise la personnalisation	<a href="#">Augustin Ghauratto</a> , <a href="#">whrrgarbl</a>