JAVA
FONDAMENTAUX

Présenté par :

Xavier TABUTEAU

Log4j

- Log4j (abréviation de Logging for Java) est une bibliothèque open source développée par la Fondation Apache. Elle permet de :

  Gérer les logs de manière configurable (fichier, console, base de données, etc...).
  Contrôler le niveau de verbosité (DEBUG, INFO, ERROR, etc…).
  Formater les messages de manière personnalisée.
  Séparer le code applicatif de la logique de journalisation.

- La journalisation (ou logging) consiste à enregistrer des informations sur l'exécution d'un programme : erreurs, avertissements, événements importants, etc.

Cela aide à :

  Diagnostiquer les problèmes
  Suivre le comportement d'une application
  Analyser les performances ou l'usage
  Déboguer sans utiliser System.out.println() (mauvaise pratique).

- Log4j repose sur trois composants principaux :

  Logger : l'objet utilisé par le développeur pour écrire les messages de log.
  Appender : définit où les logs seront envoyés (console, fichier, etc...).
  Layout / Pattern : définit comment les logs seront formatés.

Log4j

- Installation de Log4j 2 :

Avec le gestionnaire de dépendance maven, rajouter au pom.xml ce code :

```
<dependencies>
   <dependency>
      <groupId>org.apache.logging.log4j</groupId>
      <artifactId>log4j-api</artifactId>
      <version>2.23.1</version>
   </dependency>
   <dependency>
      <groupId>org.apache.logging.log4j</groupId>
      <artifactId>log4j-core</artifactId>
      <version>2.23.1</version>
   </dependency>
</dependencies>
```

Sans gestionnaire de dépendance il faut télécharger les fichiers « log4j-api-x.y.z.jar » et « log4j-core-x.y.z.jar ». Lien de téléchargement : https://logging.apache.org/log4j/2.x/download.html
On place les fichiers jar dans un sous dossier du projet (« libs » par exemple).
Ensuite il faut intégrer ces fichiers jar au projet. Pour cela, faire un clic droit sur le dossier du projet, propriété. Une fenêtre apparait. Choisir « Java Build Path » dans la partie de gauche. Dans la partie de droite, sélectionner l'onglet « libraries » puis ClassPath (ou ModulePath si on utilise les modules), cliquer sur « add jar », sélectionner les fichiers jar, cliquer sur le bouton « Ouvrir » pour les ajouter. Une fois les deux fichiers dans la liste, cliquer sur le bouton « Apply and Close ».

Log4j

- Configuration de log4j :

Pour utiliser log4j, il faut un fichier de configuration « XML » de log4j que l'on nommera « log4j2.xml » qu'on peut ajouter dans un sous dossier du projet (par exemple : « conf_log4j_xml »). Voici un exemple (celui qui nous allons utilisé) :

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="WARN">
  <Appenders>
    <!-- Affiche les logs dans la console -->
    <Console name="Console" target="SYSTEM_OUT">
      <PatternLayout pattern="%d{HH:mm:ss} [%t] %-5level %logger{36} - %msg%n" />
    </Console>
  </Appenders>

  <Loggers>
    <!-- Logger principal -->
    <Root level="debug">
      <AppenderRef ref="Console"/>
    </Root>
  </Loggers>
</Configuration>
```

Le PatternLayout est le format de l'affichage des logs. Il peut être modifié.
Root level="debug" est le niveau du déclanchement des logs. Par exemple, si le level est info, les debug ne s'affiche pas.

Log4j

Un autre exemple de configuration du fichier « log4j2.xml » :

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="WARN">
    <Appenders>
        <!-- Affiche les logs dans la console -->
        <Console name="ConsoleAppender" target="SYSTEM_OUT">
            <PatternLayout pattern="[%d{HH:mm:ss}] [%-5level] %c{1} - %msg%n"/>
        </Console>
        <!-- Écrit les logs dans un fichier -->
        <File name="FileAppender" fileName="logs/app.log">
            <PatternLayout pattern="[%d{yyyy-MM-dd HH:mm:ss}] [%-5level] %c - %msg%n"/>
        </File>
    </Appenders>
    <Loggers>
        <!-- Logger pour une classe spécifique -->
        <Logger name="com.example" level="debug" additivity="false">
            <AppenderRef ref="ConsoleAppender"/>
            <AppenderRef ref="FileAppender"/>
        </Logger>
        <!-- Logger racine -->
        <Root level="info">
            <AppenderRef ref="ConsoleAppender"/>
        </Root>
    </Loggers>
</Configuration>
```

Log4j

- PatternLayout :

%d{dd/MM/yyyy HH:mm:ss}	Date et heure du log	14:35:22
%level ou %p	Niveau du log	INFO, DEBUG, ERROR
%-5level	Niveau du log aligné sur 5 caractères	INFO_ , DEBUG
%c ou %logger	Nom complet du logger (classe)	com.example.Main
%c{1}	Nom simple de la classe	Main
%t	Nom du thread	main
%msg ou %m	Message de log	“Erreur de connexion”
%n	Nouvelle ligne	(saut de ligne)
%F	Nom du fichier source	Main.java
%L	Numéro de ligne	42
%M	Nom de la méthode	main
%throwable	Stacktrace en cas d’exception	

Le « - » aligne à gauche.

- Bonnes pratiques

Inclure au moins la date et le niveau pour pouvoir suivre les logs facilement.
Ajouter la classe ou la méthode seulement si nécessaire (utile en debug).
Utiliser %n pour garantir des lignes séparées dans les fichiers log.
Eviter les patterns trop longs en production pour améliorer les performances.

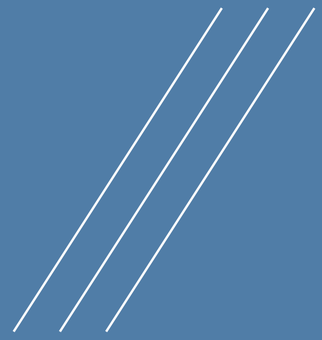
Log4j

• Niveau de priorité des logs :

TRACE	Niveau le plus détaillé, pour suivre le déroulement précis du programme (utile pour le débogage fin). Exemple : Suivi de variables, entrées/sorties de méthodes.
DEBUG	Messages généraux de débogage, pour comprendre le fonctionnement du code. Exemple : Informations utiles pendant le développement.
INFO	Informations générales sur le déroulement normal de l'application. Exemple : Démarrage du serveur, connexion réussie, etc.
WARN	Signale un événement inhabituel mais non bloquant. Exemple : Délais dépassés, tentative de reconnexion, configuration manquante par défaut, etc.
ERROR	Indique une erreur empêchant une action de s'exécuter correctement. Exemple : Exception capturée, ressource non trouvée, etc.
FATAL	Erreur critique qui provoque généralement l'arrêt du programme. Exemple : Erreur système, échec irrécupérable.
OFF	Désactive complètement la journalisation. Exemple : Aucun log.

conf_log4j_xml\log4j2.xml
Log4j.java

JAVA
FONDAMENTAUX



Présenté par
Xavier TABUTEAU