Configuración Log4j

Control de versiones del documento

Versión	Fecha	Autor	Motivo del cambio
1.0	28/05/2015	Antonio Morcillo Martínez	Creación del documento

Configuración *Log4j*

A pesar de que *Liferay* dispone de sus propios mecanismos de *logging*, basados en *Log4j*, consideramos que es especialmente útil que cada uno de los *plugins* que se instalan en la plataforma disponga de su propio fichero de log con unas políticas de *log* propias y que permita localizar rápidamente los mensajes de log propios de cada uno de los *plugins*.
Esto se puede conseguir de forma sencilla

Ficheros de configuración

Para disponer de una configuración independiente para nuestro *plugin* bastará con tener en el *CLASSPATH* del plugin un par de ficheros.

log4j.dtd

El fichero de validación de esquema. Debe incluirse para evitar potenciales problemas con el *classloading* de *Liferay* que hagan que *Log4j* no lo encuentre a la hora de validar el fichero de configuración
Puede obtenerse por ejemplo

```
http://grepcode.com/file/repo1.maven.org/maven2/log4j/log4j/1.2.16/org/apache/log4j/xml/log4j.dtd
```

O en cualquiera de los plugins existentes hasta el momento

log4j.xml

En este fichero especificamos la configuración en sí.

Una configuración tipo podría ser la siguiente

```
<?xml version="1.0"?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

    <appender name="FILE"
class="org.apache.log4j.rolling.RollingFileAppender">
        <rollingPolicy
class="org.apache.log4j.rolling.TimeBasedRollingPolicy">
            <param name="FileNamePattern" value="../../logs/mi-
plugin.%d{yyyy-MM-dd}.log" />
        </rollingPolicy>

        <layout class="org.apache.log4j.PatternLayout">
```

```
            <param name="ConversionPattern" value="%d{ABSOLUTE} %-5p
[%c{1}:%L] %m%n" />
        </layout>
    </appender>

    <root>
        <priority value="${rest-services.log.level}" />
        <!-- appender-ref ref="CONSOLE" /-->
        <appender-ref ref="FILE" />
    </root>
</log4j:configuration>
```

Donde:

- Se especifica un *appender* llamado *FILE* con rotación diaria *class="org.apache.log4j.rolling.RollingFileAppender"*.
- Se especifica la ruta y el formato de los ficheros de log *name="FileNamePattern" value="../../logs/mi-plugin.%d{yyyy-MM-dd}.log"*
- El nivel de log está pametrizado mediante el *token ${rest-services.log.level}* que será sustituido por *Maven* en tiempo de compilación
- Se asigna el appender *FILE* a todo el *plugin*

Para más información

```
http://logging.apache.org/log4j/2.x/
```

Utilizando Log4j en clases Java

Para obtener una referencia a log4j basta con tener en nuestra clase una variable estática como la siguiente

```
    private static Logger LOGGER = Logger.getLogger(MI_CLASE.class);
```

Si el IDE utilizado no fuera capaz de resolver el import necesario; habría que poner la siguiente declaración en la zona de importaciones de la clase donde se vaya a utilizar el logger

```
import org.apache.log4j.Logger;
```

Configuración Maven

Si no se hereda de ningún pom donde ya vaya declarada la dependencia es necesario añadir al pom del plugin la siguiente declaración

```
<dependency>
      <groupId>log4j</groupId>
      <artifactId>log4j</artifactId>
      <version>${log4j.version}</version>
      <scope>provided</scope>
</dependency>
```

Es importante destacar que se debe poner scope provided **sólo** si se usa como plugin de Liferay (hook, ext o portlet) ya que Liferay nos proporciona de serie la librería mediante su cargador de clases