# Логирование класса/пакета

Например, для класса com.liferay.portal.security.ldap.PortalLDAPImporterImpl или пакета com.liferay.portal.security.ldap:

log4j.properties:

Пример 1

log4j.logger.com.liferay.portal.security.ldap.PortalLDAPImporterImpl = OFF, LDAP\_FILE

portal-log4j-ext.xml

Пример 2

<logger name="com.liferay.portal.security.ldap">  
 <priority value="INFO" />  
 <appender-ref ref="LDAP\_FILE" />  
</logger>

LDAP\_FILE - appender

# Перенаправление логов

По умолчанию все логи, у которых нету логера идут в root logger.  
Он имеет вид

Пример 3

<root>  
 <priority value="INFO" />  
 <appender-ref ref="CONSOLE" />  
 <appender-ref ref="TEXT\_FILE" />  
 <appender-ref ref="XML\_FILE" />  
</root>

Можно создавать logger для перехвата логов. См. Пример 2

Чтобы обрабатывать лог используется appender.

Пример 4

<appender class="org.apache.log4j.rolling.RollingFileAppender" name="LDAP\_FILE">  
 <rollingPolicy class="org.apache.log4j.rolling.TimeBasedRollingPolicy">  
 <param name="FileNamePattern" value="@liferay.home@/logs/LDAP@spi.id@.%d{yyyy-MM-dd}.log" />  
 </rollingPolicy>  
 <layout class="org.apache.log4j.EnhancedPatternLayout">  
 <param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss.SSS} %-5p [%t][%c{1}:%L] %m%n" />  
 </layout>  
</appender>

org.apache.log4j.rolling.RollingFileAppender – тип лога с функционалом хранить логи по частям. Есть множество других классов аппендеров с разным функционалом

rollingPolicy отвечает за то где и как будут храниться логи. Как будут удаляться, какие будут иметь имена.

layout отвечает за форматирование лога.