**Log4J Interview Questions**

**What is Log4j and what are its uses?**

Log4J is a reliable, fast and flexible logging framework (APIs) written in Java which is distributed under the Apache Software License. It is used by programmers to output log statements to a variety of targets. These are very useful in case of errors as the root cause of the error can be located.

Log4J is developed by apache to log any errors, output etc.

**Why Log4J?**

Log4j is fully configurable using .properties or .xml.

Log4j enables you to find out easily which are information messages, which are debug messages, which are errors and which are fatal errors.

Log4j records information upto class level. For example, you can output only error messages for one class and for another info. Messages only. Similarly at package level also.

If any error occurs, it will show you the class name also, if configured.

**How to install Log4j?**

Go to google and type apache log4j download.

Download the binary zip file.

**Important note: Now once you unzip this you will see different jar files and you need only log4jcore and log4japi. If you include anymore files in build path you will get classnotfound exception.**

**What is Log4j configuration file?**

Log4j configuration file is an xml file which is used by Log4j framework to find what information like info, error etc to be logged, how to log it and where to log it.

**What if we don’t use the Log4j configuration file?**

If you don’t use the log 4j 2 configuration file, it will use default configuration file and debug and info messages won’t be printed.

**Which tags in Log4j defines where to log the information?**

Appenders tags define the place where to log the information.

**Which tags in Log4j define how to log?**

PatternLayout tag.

**Which tags in Log4j define what information to log?**

Loggers tag.

Loggers tag contains two tags. Logger and Root.

These two tags contain an attribute level.

If level is trace, all the information info, debug, error, fatal will be printed.

If level is error, only error and fatal information will be logged.

**What is the difference between Logger and Root?**

If you have multiple packages and for one package you want only errors and fatal information should be logged and for other you want all the information to be logged, then you can mention that one package where you want only errors and fatal under logger tag with level as error and under root you can mention the level as trace. So that only for that one package errors and fatal information will be logged and for all other packages all other information will be logged.

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<Configuration status=*"WARN"*>

<Appenders>

<Console name=*"Console"* target=*"SYSTEM\_OUT"*>

<PatternLayout pattern=*"%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"*/>

</Console>

</Appenders>

<Loggers>

<logger name=*"loggingpackage2.logtest1"* level=*"error"* additivity=*"false"*>

<AppenderRef ref=*"Console"*/>

</logger>

<Root level=*"trace"*>

<AppenderRef ref=*"Console"*/>

</Root>

</Loggers>

</Configuration>

In the above example we defined loggingpackage2 under logger with level as error so only errors and fatal information will be logged for these.

For all others, since we logged the level as trace, all the information will be logged.

**How to log into a file?**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<Configuration status=*"WARN"*>

<Properties>

<Property name=*"basePath"*>./logs</Property>

</Properties>

<Appenders>

<RollingFile name=*"File"* fileName=*"${basePath}/output.log"* filePattern=*"${basePath}/output-%d{yyyy-MM-dd}.log"*>

<PatternLayout pattern=*"%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"*/>

<SizeBasedTriggeringPolicy size=*"500"* />

</RollingFile>

<Console name=*"Console"* target=*"SYSTEM\_OUT"*>

<PatternLayout pattern=*"%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"*/>

</Console>

</Appenders>

<Loggers>

<logger name=*"loggingpackage2.logtest1"* level=*"error"* additivity=*"false"*>

<AppenderRef ref=*"File"*/>

</logger>

<Root level=*"trace"*>

<AppenderRef ref=*"File"*/>

</Root>

</Loggers>

</Configuration>

You can define the path in parameter <Properties>. You can use this basepath in RollingFile to log whatever we want.

<Properties>

<Property name=*"basePath"*>./logs</Property>

</Properties>

This will first create a folder logs.

<RollingFile name=*"File"* fileName=*"${basePath}/output.log"* filePattern=*"${basePath}/output-%d{yyyy-MM-dd}.log"*>

<PatternLayout pattern=*"%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"*/>

<SizeBasedTriggeringPolicy size=*"500"* />

</RollingFile>

This section under Appenders will create a log file as output.log under basepath. Once it reaches the file size mentioned in <SizeBasedTriggeringPolicy size=*"500"* /> it will create another file with the pattern mentioned in filePattern.

**How to log into different files for different packages?**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<Configuration status=*"WARN"*>

<Properties>

<Property name=*"basePath"*>./logs</Property>

</Properties>

<Appenders>

<RollingFile name=*"package1logs"* fileName=*"${basePath}/package1\_output.log"* filePattern=*"${basePath}/package1\_output-%d{MM-dd-yy-HH-mm-ss}.log"*>

<PatternLayout pattern=*"%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"*/>

<SizeBasedTriggeringPolicy size=*"500"* />

</RollingFile>

<RollingFile name=*"package2logs"* fileName=*"${basePath}/package2\_output.log"* filePattern=*"${basePath}/package2\_output-%d{MM-dd-yy-HH-mm-ss}.log"*>

<PatternLayout pattern=*"%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"*/>

<SizeBasedTriggeringPolicy size=*"500"* />

</RollingFile>

<Console name=*"Console"* target=*"SYSTEM\_OUT"*>

<PatternLayout pattern=*"%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"*/>

</Console>

</Appenders>

<Loggers>

<logger name=*"package1.First"* level=*"error"* additivity=*"false"*>

<AppenderRef ref=*"package1logs"*/>

</logger>

<logger name=*"package2.First"* level=*"trace"* additivity=*"false"*>

<AppenderRef ref=*"package2logs"*/>

</logger>

<Root level=*"trace"*>

<AppenderRef ref=*"Console"*/>

</Root>

</Loggers>

</Configuration>

We have to create different RollingFile tags with different file names.

Also we have to create different logger tags with whatever level of information we need.