The LogRecord class wraps the messages logged to a Logger. You can either log a LogRecord directly to aLogger, or have the Logger wrap what is logged in a LogRecord. That is, all logging methods on the Loggerthat do not take a LogRecord will wrap the parameters passed to the log method in a LogRecord object.

Once a LogRecord is passed to, or produced internally by the Logger, it can be passed to a Filter (if a filter is set), to Handler's, and to Logger's further op the [**Logger hierarchy**](http://tutorials.jenkov.com/java-logging/logger-hierarchy.html).

The LogRecord contains the following getter methods, which can be used in e.g. a Filter to determine if theLogRecord should be logged or not:

getLevel()

getLoggerName()

getMessage()

getMillis()

getParameters()

getResourceBundle()

getResourceBundleName()

getSequenceNumber()

getSourceClassName()

getSourceMethodName()

getThreadID()

getThrown()

Each of these getter methods have a corresponding setter method too, but unless you create LogRecordinstances yourself, you won't need the setter methods. Here is a brief explanation of the getter methods.

The getLevel() method returns the log level the message represented by this LogRecord was logged with.

The getLoggerName() method returns the name of the Logger that logged this LogRecord.

The getMessage() method returns the message logged.

The getMillis() method returns the time in milliseconds this LogRecord was recorded.

The getParameters() method returns the parameters to be inserted into the message of this LogRecord.

The getResourceBundle() method returns the ResourceBundle (if any) used to localize the message of this LogRecord. If no ResourceBundle is used, this method returns null.

The getResourceBundleName() method returns the name of the ResourceBundle (if any) used to localize the message of this LogRecord. If no ResourceBundle is used, null is returned.

The getSequenceNumber() method returns a sequence number generated internally in the LogRecordconstructor when the LogRecord is created.

The getSourceClassName() returns the class name of the class logging the message represented by thisLogRecord. Keep in mind though, that this name may not be entirely correct for various reasons. See the JavaDoc for more details.

The getSourceMethodName() returns the method name of the method logging the message represented by this LogRecord. Keep in mind though, that this name may not be entirely correct for various reasons. See the JavaDoc for more details.

The getThreadID() method returns the ID of the thread logging the message represented by thisLogRecord.

The getThrown() method returns the Throwable that was marked thrown when logging the message represented by this LogRecord. This Throwable is either passed as parameter to a Logger log method, or set directly on a LogRecord which is then passed to a Logger. This Throwable is typically what caused the log message to be logged