

## Logger Object

=====

=> enables logging on the given java class.

=> `Logger logger = Logger.getLogger(current java class related to java.lang.Class object);`

eg: `Logger logger = Logger.getLogger(BankAppProject.class);`

=> use the following method based on the priority level to generate the log message

```
logger.debug("");
logger.info("");
logger.warn("");
logger.error("");
logger.fatal("");
```

=> Allows to specify logger level to retrieve log messages

`logger.setLevel(Level.DEBUG)`[if no logger level is given then the default logger level is DEBUG]

=> Both Appender object and Layout object will be added to logger object directly or indirectly.

=> Instructions/Inputs to logger object can be hardcoded or given by using properties file/xml file.

## Appender Object

=====

=> specifies the destination where to write/record the log messages.

eg:

`FileAppender, RollingFileAppender, DailyRollingFileAppender, JDBCAppender, IMAPAppender, ConsoleAppender`

=> All Appender classes implements from `org.log4j.Appender(I)`.

## Layout Object

=====

=> Given to format the log messages before giving to appender for recording/writing the destination file

eg: `SimpleLayout, HtmlLayout, XmlLayout, PatternLayout.....`

=> All Layout classes extends from `org.log4j.Layout(c)`

## Log4j Architecture

=====

refer diagram

To load the properties file data into java program we need to use the following class

=====

```
static {
    PropertyConfigurator.configure("src\\main\\java\\in\\neuron\\cfgs\\log4j.properties");
}
```

## Using log4j to work with properties file

=====

case1:: Working with HTMLLayout and FileAppender

```
#For HTMLLayout and FileAppender
#specify Logger level to retrieve the log messages
log4j.rootLogger=INFO,R
```

```
#specify appender
log4j.appender.R=org.apache.log4j.FileAppender
```

```
#Specify file name
log4j.appender.R.File=D:\\log.html
```

```
#Disabling append mode on file
log4j.appender.R.append=true
```

```
#specify layout
log4j.appender.R.layout=org.apache.log4j.HTMLLayout
```

```
=====
```

#### Case2:: Working with Multiple Appenders

```
#For Working with multiple Appenders
#specify Logger level to retrieve the log messages
log4j.rootLogger=DEBUG,R,C
```

```
#related ConsoleAppender and SimpleLayout
log4j.appender.C=org.apache.log4j.ConsoleAppender
log4j.appender.C.layout=org.apache.log4j.SimpleLayout
```

```
#related to FileAppender and HTMLLayout
log4j.appender.R=org.apache.log4j.FileAppender
log4j.appender.R.File=D:/log.html
log4j.appender.R.append=true
log4j.appender.R.layout=org.apache.log4j.HTMLLayout
```

```
=====
```

#### Case3:Working with FileAppender and XMLLayout

```
#For XmlLayout and FileAppender
##specify Logger level to retrieve the log messages
log4j.rootLogger=INFO,R
```

```
##specify appender
log4j.appender.R=org.apache.log4j.FileAppender
```

```
##Specify file name
log4j.appender.R.File=D:/info.xml
```

```
##enabling append mode on file
log4j.appender.R.append=true
```

```
##specify layout
log4j.appender.R.layout=org.apache.log4j.xml.XMLLayout
```

```
=====
```

#### Case4: RollingFileAppender and PatternLayout

```
# ForRollingFileAppender and PatternLayout
log4j.rootLogger=DEBUG,R
log4j.appender.R=org.apache.log4j.RollingFileAppender
log4j.appender.R.File=log_info.txt
```

```
log4j.appender.R.MaxFileSize=10KB
log4j.appender.R.MaxBackupIndex=3
log4j.appender.R.append=true
log4j.appender.R.layout=org.apache.log4j.PatternLayout
log4j.appender.R.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %p %10.15c %10M
%-10t %r %L %m%n
```

# For DailyRollingFileAppender and PatternLayout

```
log4j.rootLogger=DEBUG,R
log4j.appender.R=org.apache.log4j.DailyRollingFileAppender
log4j.appender.R.File=log_info1.txt
log4j.appender.R.append=true
log4j.appender.R.datePattern='.'yyyy-MM-dd-HH-mm
log4j.appender.R.layout=org.apache.log4j.PatternLayout
log4j.appender.R.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %p %10.15c %10M
%-10t %r %L %m%n
```

Note: In real timepractise, the popularly used appender is "DailyRollingFileAppender" and layout is "PatternLayout".