**Step 1 : Add dependency**

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.17</version>

</dependency>

**Step 2: Make a properties/yml file by the name Log4j.properties**

#Log4J configuration

log4j.rootLogger=DEBUG,FILE

log4j.appender.FILE=org.apache.log4j.DailyRollingFileAppender

log4j.appender.FILE.Append = true

log4j.appender.FILE.DatePattern = '.'yyy-MM-dd

log4j.appender.FILE.File=/apps/logs/sbi.log

log4j.appender.FILE.layout=org.apache.log4j.PatternLayout

log4j.appender.FILE.layout.ConversionPattern=[%d] %t %c %L %-5p - %m%n

log4j.logger.org.springframework=WARN

loggermethod

[%d] DateTime %t thread %c class %L line %-5p - %m%n message

[2024-04-25 18:23:58,265] http-nio-8084-exec-2 com.excelr.h2dbdemo.controller.MyController 38 INFO - Add Student API Completed

**Step 3 : Make logger object**

**import** org.apache.log4j.Logger;

**private** **static** **final** Logger ***logger***=Logger.*getLogger*(DepartmentController.**class**);

**Step 4 : Use logger object**

***logger.error("Error Message"); //200***

***logger.warn("Warn Message"); //300***

***logger.info("Info Message"); //400***

***logger.debug("Debug Message"); //***

Logger – Captures information

Layout

Appender – publish, where u need to publish –default System.out, database, file, network

Appender Types

* **ConsoleAppender:** Console Appender appends the log events to System.out or System.err using a layout specified by the user. System.out is a default target. It is useful for debugging purposes, but not much beneficial to use in a production environment.
* **FileAppender:** Appends log events to a file. It supports two more appender classes:
* **RollingFileAppender, DailyRollingFileAppender:** Both are the most widely used appenders that provide support to write logs to file.
* **JDBCAppender:** The JDBCAppender is used for sending log events to a database. Each append call adds to an ArrayList buffer. When the buffer is filled, each log event is placed in a SQL statement and executed.
* **SMTPAppender:** It is used to send an email when a specific logging event occurs, typically on errors or fatal errors.
* **SocketAppender:** It is used for remote storage.

To write your information about logging into multiple files, you have to use **org.apache.log4j.RollingFileAppender** class which extends the FileAppender class and inherits all its properties.

DatePattern controls the rollover schedule using one of the following patterns:

|  |  |
| --- | --- |
| **DatePattern** | **Description** |
| '.' yyyy-MM | Roll over at the end of each month and at the beginning of the next month. |
| '.' yyyy-MM-dd | Roll over at midnight each day. This is the default value. |
| '.' yyyy-MM-dd-a | Roll over at midday and midnight of each day. |
| '.' yyyy-MM-dd-HH | Roll over at the top of every hour. |
| '.' yyyy-MM-dd-HH-mm | Roll over every minute. |
| '.' yyyy-ww | Roll over on the first day of each week depending upon the locale. |

**DatePattern:** This indicates when to rollover the file and the naming convention to be followed. By default, rollover is performed at midnight per day.