Logging

What is logging and why logging?

Logging: It is a process of storing application execution details into a console / file is called logging.

Why logging?

To understand the application runtime behaviour.

By using logging, we can find root cause analysis (RCA) of the exceptions.

To implement logging in application we have several frameworks in the market.

1. Log4j
2. Log4j2
3. Logback
4. Logstash

Logging Architecture

1. Logger

Logger is a predefined class and it is providing some pre define methods to write messages.

For each class we need to create one logger object.

Ex:

* Trace
* Debug
* Info (Default level)
* Warn
* Error

1. Layout

It represents log messages pattern

<date> <time> <class-name> <level> <message>

We can write our custom messages also by using logback.xml configuration file.

1. Appender

It is used to write log messages into destination file.

Ex: console appender (default)

File appender

JDBC appender (but log messages we won’t store in databases.)

Understand log levels

1. Trace
2. Debug
3. Info (default level)
4. Warn
5. Error

When log messages are printing by default it prints from info, warn, error messages only because info is default level.

To change log level, go to application properties and write below key value

Logging.level.root=debug

Now logs will print from debug, info, warn, error.

By default spring boot will use console appender, hence log messages will print in console.

To change the appender from console to file, use below key value in properties

Logging.file.name=lb.log

But here there is a problem, daily many people are using our application and those messages are storing in single file due to that file size will increase and finding messages in file is difficult.

To overcome this issue logging with Rolling policy came into picture

Logging with Rolling policy

It is used to create multiple log files to store the messages of our applications.

Rolling policy can be implemented in two ways

1. Size based rolling

In logback.xml file we can give explicitly size to store messages in file.

Ex:

Suppose I have given 1GB in xml file, one file reached limit of 1gb then immediately It will create new file and then starts storing log messages.

1. Time based rolling

Every day new log file should be created.

Understand logback.xml configuration

In logback.xml we configure

1. Rolling file appender
2. Log message pattern
3. Log level

We create logback.xml at src/main/resources folder.