Design document for:

Logger Implementation in Web PLP

Written by

Mukulsingh Jadhav

Version #1

Tuesday, February 27, 2018

PLP Logger Implementation

I am using log4j 2.10.0 for the logging. Log4j 2 architecture is as follows:

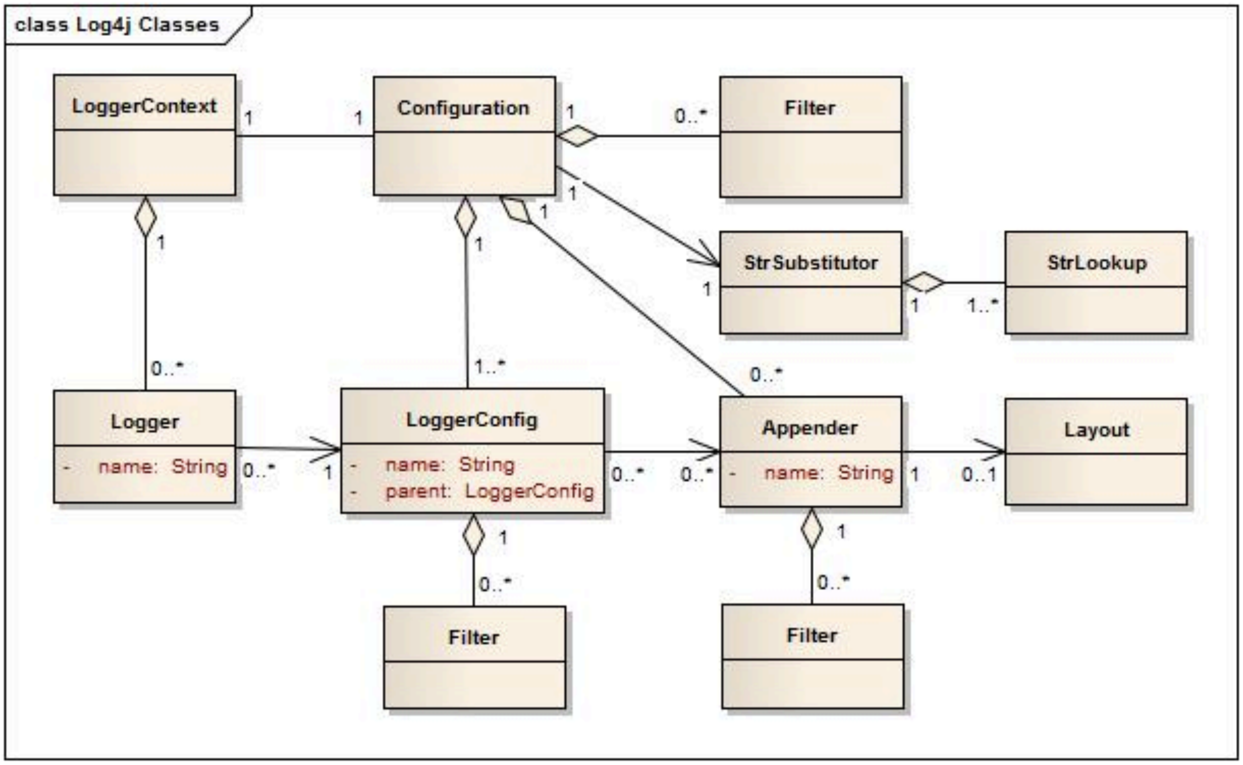

Fig 1. Log4j 2 Architecture [1]

Applications using the Log4j 2 API will request a Logger with a specific name from the LogManager. The LogManager will locate the appropriate LoggerContext and then obtain the Logger from it. If the Logger must be created it will be associated with the LoggerConfig that contains either a) the same name as the Logger, b) the name of a parent package, or c) the root LoggerConfig. LoggerConfig objects are created from Logger declarations in the configuration. The LoggerConfig is associated with the Appenders that actually deliver the LogEvents [2].

Log4j 2 has various features:

1. API separation
2. Improved performance
3. Avoid lock-in
4. Automatic reloading of configurations
5. Custom log levels
6. Garbage free

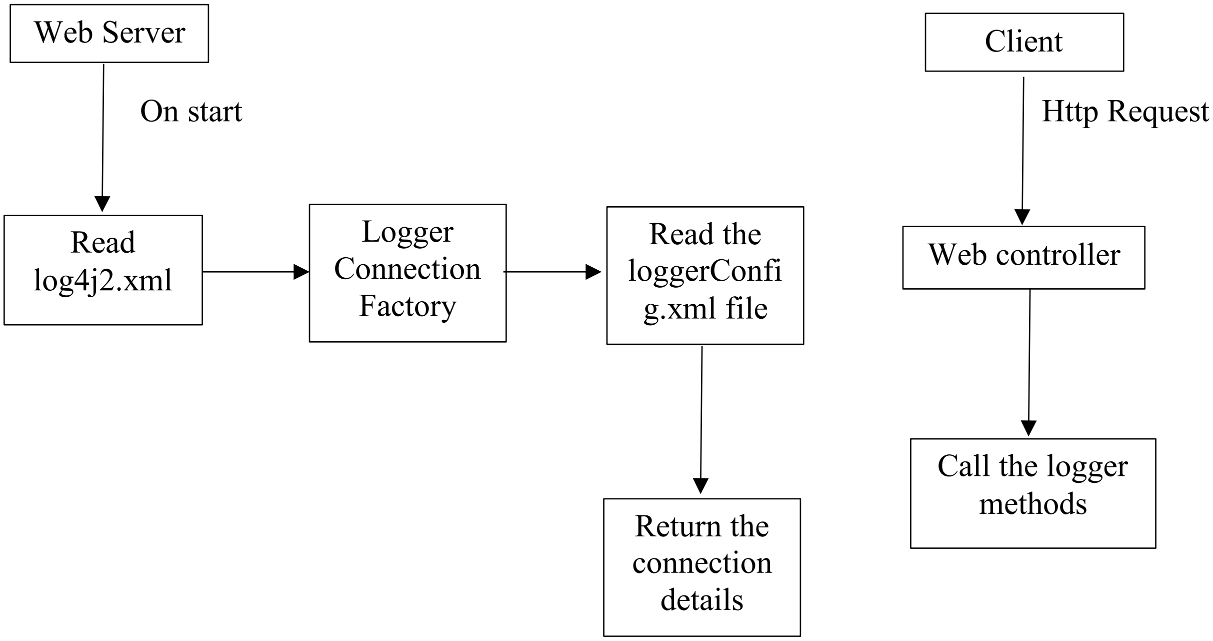

Fig. 2. Design diagram of Logger implementation

As the number of users will be in 100's, all the logs will be written synchronously in the database. This configuration can be changed later on whenever required by modifying the log4j2.xml file.

The system is using JDBC appender to log the messages into the database. If the number of users/requests increases in future, we can maintain the buffer size in which messages will be queued and once the buffer is full, batch insertion will take place.

We have decided to use INFO level for writing information messages and ERROR for the exception messages. Log4j2 has provision to record custom level messages. We have decided to use ASSEMBLE level to write messages for assemble function and SIMULATE level for simulation function.

If we want to log only the error messages in the future, log4j2 provides level attribute for the specific logger. If the level is set to debug, it will record all the messages. If the level is set to INFO, then only INFO level messaged will be recorded. This is true for other level messages as well.

References:

[1] log4j2 architecture - *https://logging.apache.org/log4j/2.0/manual/architecture.html*
[2] log4j2 info - *https://logging.apache.org/log4j/2.0/manual/architecture.html*