



# How to overcome 'Logging FOMO'

In an ideal world, developers wouldn't need to waste precious time adding and removing countless log lines for every single feature in the application. Instead, they would focus on building more value-focused features. In this perfect world, if a problem arose and some data was missing, developers could extract that data effortlessly. Unfortunately, in the world we are in, where logging hasn't materially changed in over 30 years, nothing could be further from the truth.

"Logging FOMO" is the fear of missing out on log lines and the data they comprise. It is the anxiety that something may go south in the future, with no log lines will be there to tell you how and why it happened.

## The necessity of many log lines

When developing a new feature, an engineer is likely to write anywhere between 100 to 1000 lines of code and a good number of those will be log lines. This is due to the 'what ifs', the possible scenarios of everything that could go wrong, each of which necessitates one or more of those extra log lines to solve it. Log lines are often used in order to know if a certain code line has been reached, that some operation has failed, or just to get the value of a key variable.

It is important to note that when things go wrong, and they usually do, the first step to solving the problem is to understand what happened. Observability tools such as log aggregations can only provide us answers based on the data that is already being collected. And so, engineers spend much of their time adding, tweaking, and removing log lines to collect the data points needed. Often, these changes comprise many of the pull requests that are done each day.



***"Logging FOMO" is the fear of missing out on log lines and the data they comprise.*** ”





# ROOKOUT

## The difficulties of logging

The process of getting logging “just right” for your application is no small task. A lot of time and effort are spent doing so, that could be otherwise used to build better software. There are many scenarios in which you’ll find yourself wanting to add more log lines. Even more so, it’s an extremely long process to change your logs, get an extra piece of data to fix a bug, or remove a piece of data you no longer want to extract.

First, you’ll need to re-run tests with the hope that they are stable and deterministic and that no one else wrote a test which depends on the log you’ve just changed. Then, you will need to search for someone with the skill, authority, and the availability to approve your pull request. After that, if you’ve successfully deployed that change to production, you pray that your simple log line won’t throw an exception or otherwise impact the flow in some unexpected edge case.

## The cost of logging

Unfortunately, performance and availability impacts are not the only costs and risks associated with logging. Logs are likely to hold potentially security-sensitive information, such as passwords. The information you are logging could fall under GDPR or HIPAA and may have to be governed and secured accordingly. Even worse, your organization might be paying way too much for log aggregation service, prompting you to stop whatever it is you are doing and work on removing repetitive or big logs to lower costs.

And that might not even be the worst of it. Log lines don’t always accurately reflect the true state of the system. The more verbose of log lines were often written by developers for their own use, and can easily be misunderstood when going back to that code later. Some log lines are even plain wrong, misrepresenting the code around them, especially as systems change and evolve. And after you figure out a bad log line has sent you down a wild goose chase, you have to fix it, spend more time and effort on releases and hotfixes.

## The future of logging

Logging is frustrating. Yet, new technologies are emerging, separating logging from the application’s source code, making it a much more agile endeavor. With the use of techniques such as Non-Breaking Breakpoints and Virtual-Logging, one can instantly add, tweak, or even remove log lines without having to write a single line of code or waiting for the next deployment.

This agility results in up-to-date logging, higher-quality data, better governance, at a lower total cost of ownership. Taking away risks and slow turnaround times, engineers are empowered to get the data as the need for it arises, removing FOMO from the equation.

Spend your mental resources building the perfect product rather than struggling to balance your Observability needs. The future is here.

[TRY FREE](#)