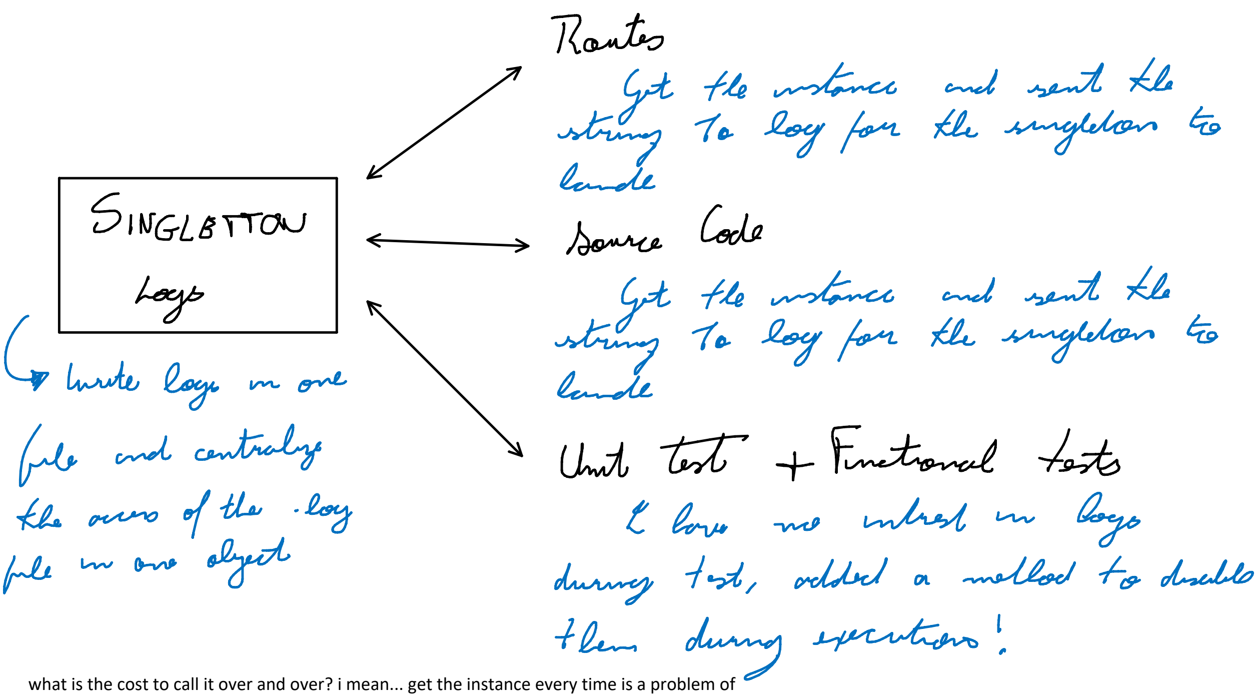
Logger

segunda-feira, 14 de abril de 2025 06:56



Routes: Get the instance and sent the string To log for the singleton to handle

Source Code: Get the instance and sent the string To log for the singleton to handle

Unit test + Functional tests: I have no intrest in logs during test, added a method to disable them during executions!

Write logs in one file and centralize the acess of the log file in one object

what is the cost to call it over and over? i mean... get the instance every time is a problem of performance?

ChatGPT said:

Excellent and very relevant question, Hugo!

💡 Short answer:

Calling Logger::getInstance() repeatedly is very cheap and **not a performance problem** in modern C++.

- static Logger instance; is **only constructed once**, the **first time** the function is called.
- After that, **every call simply returns a reference** to that same instance.
- The compiler **ensures thread-safe initialization** (since C++11), and after that, it's just a function returning a reference → **no memory allocation, no reconstruction, no dynamic lookup**.

But again, the **real bottleneck** in logging is usually:

- Writing to file (disk I/O)
- Flushing the stream

- Implemented in a separate calss because it'll gow and i think it will make it easie to build latter one if changed the source code. If no new change... there is no need to update the libs

So this line:

cpp
CopyEdit
Logger::getInstance().log("Some message");
is **as fast as**:

cpp
CopyEdit
loggerReference.log("Some message");
(assuming loggerReference was obtained from Logger::getInstance() earlier).