📊 Java Logger Library - Design & Implementation Document

🔍 Overview

This document describes the design, architecture, and implementation of a pluggable and extensible **Java Logger Library**. The logger supports routing logs to different sinks such as text files or databases, with configurable log levels, timestamp formatting, and optional advanced features such as log rotation, async writing, and host/tracking ID enrichment.

🌐 Problem Statement

Applications need a configurable, extensible logging framework that:

- Accepts logs with level, namespace, and content
- Automatically enriches logs with timestamps
- Routes logs to appropriate destinations (sinks) like files or databases
- Allows configuration via external property files
- Supports thread-safe, asynchronous logging with log file rotation

☑️ Features

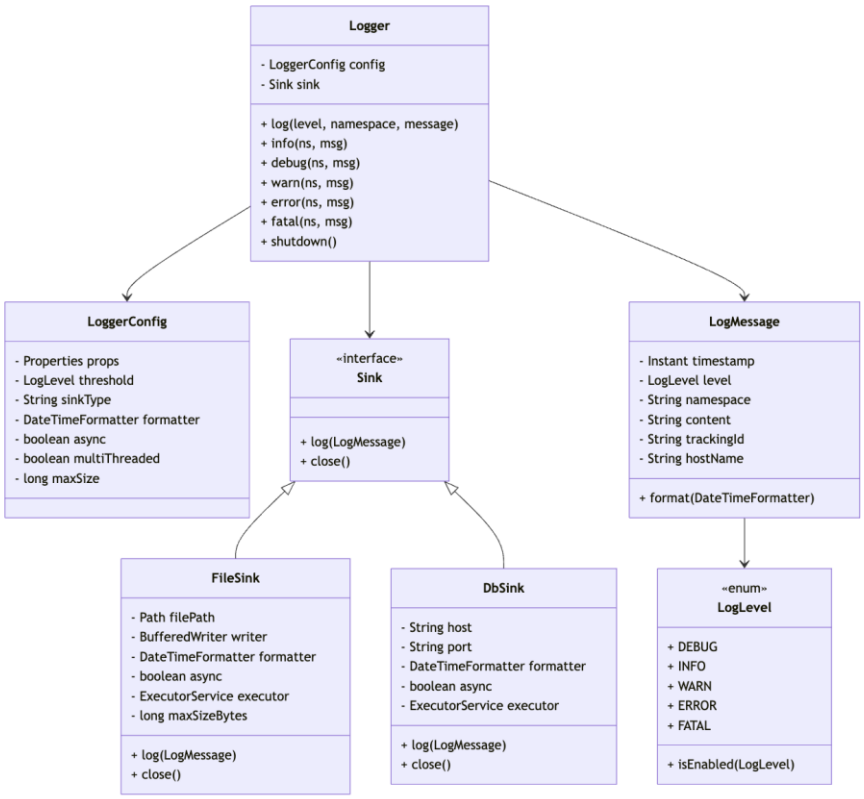
- Log levels: DEBUG, INFO, WARN, ERROR, FATAL
- Timestamp enrichment (format configurable)
- Namespace and component tagging
- Sink routing via configuration
- File sink with autorotation and compression
- DB sink (mocked via stdout)
- Async and multi-threaded writing support
- Extensible sink architecture
- Tracking ID (UUID) and Hostname enrichment

🔧 Architecture

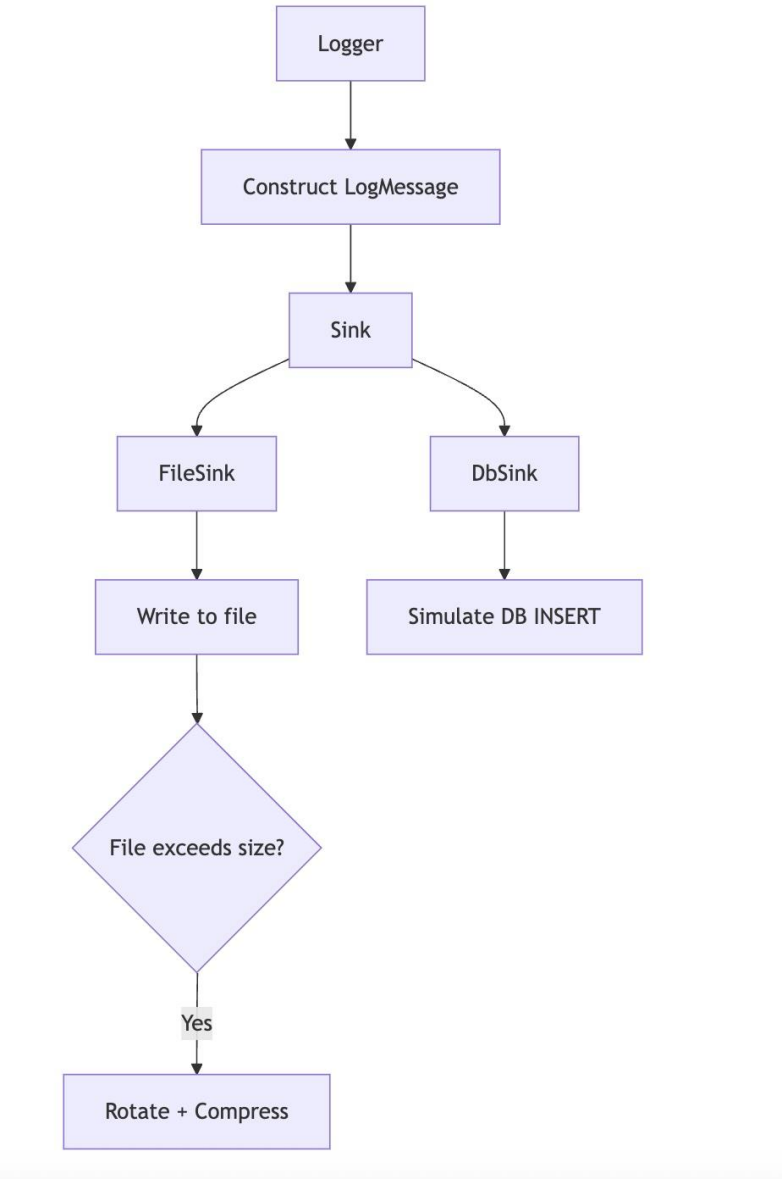
Key Components

- **Logger**: The main interface exposed to the user
- **LogMessage**: Enriched log message object
- **Sink (interface)**: Abstract sink output
- **FileSink**: Writes to local file with rotation and compression
- **DbSink**: Simulates DB write with print statement
- **LoggerConfig**: Parses and holds configuration from .properties
- **LogLevel**: Enum for log level severity comparison

Class Diagram



Flow Chart



🌐 How to Run

CLI

javac -d out src/com/logger/*.java
java -cp out com.logger.Main

IntelliJ

- Mark src as Sources Root
- Right click Main.java → Run

📊 Configuration

File Sink Config

```
ts_format=yyyy-MM-dd HH:mm:ss,SSS
log_level=INFO
sink_type=FILE
file_location=./logs/app.log
thread_model=SINGLE
write_mode=SYNC
max_file_size_bytes=1048576
```

DB Sink Config

```
ts_format=dd:MM:yyyy HH:mm:ss
log_level=ERROR
sink_type=DB
dbhost=127.0.0.1
dbport=5432
thread_model=MULTI
write_mode=ASYNC
```

💼 Sample Output (File Sink)

[2025-05-21 13:59:48,894] [LM-BGL-41512353] [c736e06b-66fb-4926-a8a9-afd330e68b03] [INFO] [App.Main] Logger started
[2025-05-21 13:59:48,989] [LM-BGL-41512353] [a3a2cea7-3da1-4d5e-a6b6-15fd60851e0c] [ERROR] [App.DB] Database error occurred

Conclusion

This logger library is a fully functional, configuration-driven, and extensible framework written in pure Java. It supports all foundational logging behaviors and is flexible enough to be extended with custom sinks or enriched message formats. It meets all problem statement goals and adheres to the best OO design practices.