Introduction to logging in Java

Logging means keeping records of various events and messages in a computer program.

Why do we keep logs? We often need specific information about our program such as user interactions, network status, errors, warnings, debugging information, stack traces in case of exceptions, and so on. It helps us analyze how the program runs, what events and errors occur at runtime – and, knowing why and when certain errors occur, we can fix them.

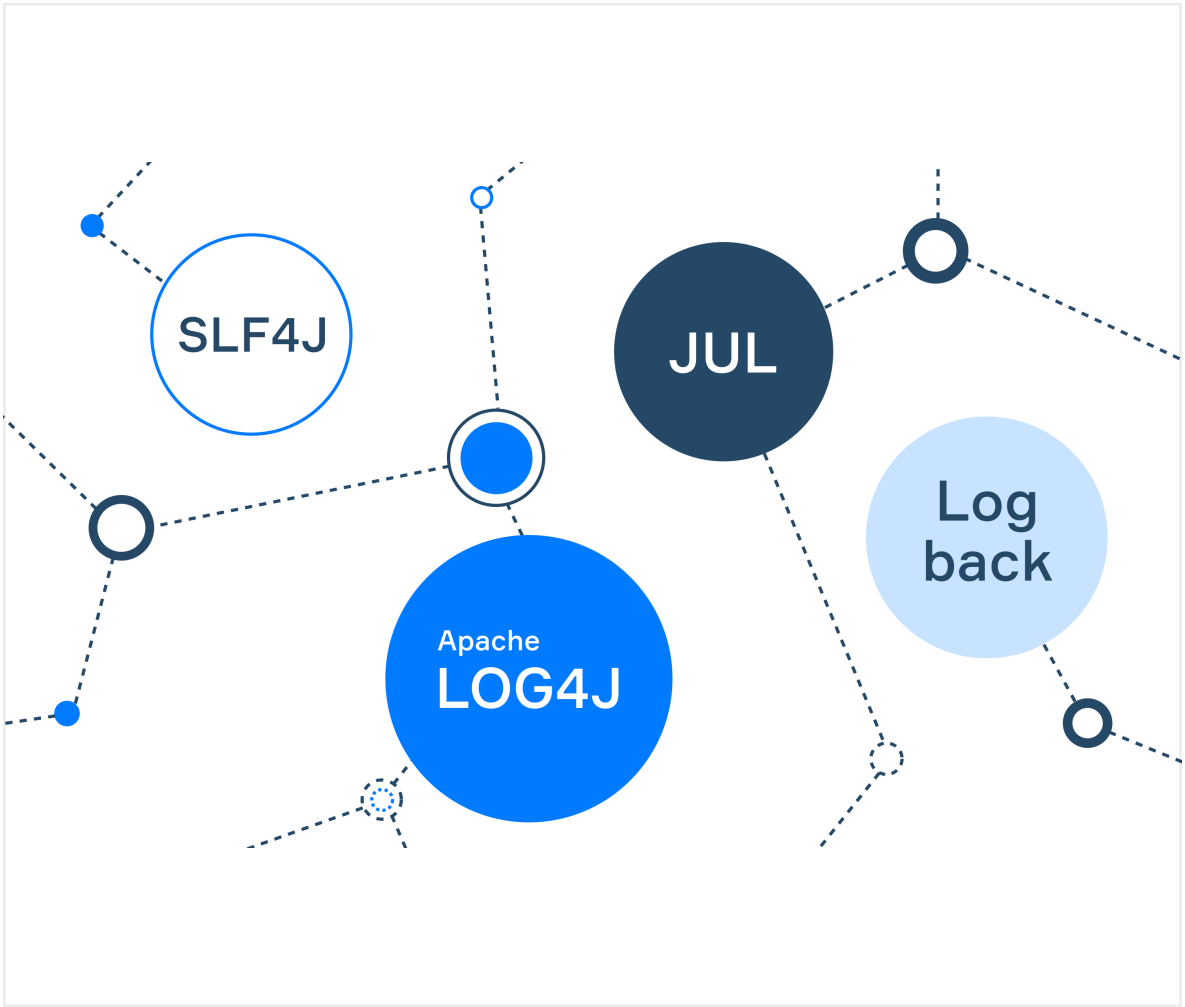
What logging tools can do

The first thing you need to know is that we cannot rely on printing log messages to System.out or System.err.

Our app may run on a remote server or a consumer device, or it may be inaccessible to us for some other reasons, but if we still want to see logs, they must be written to a file or files so we can check them later.

Here are some other things we should take into account when choosing a logging tool:
1. We should be able to control from which components of our app we want to receive log messages.
2. We should be able to filter out logging messages depending on their severity level.
3. Logs should be properly formatted for human readability and machine parsing.
4. Recording logs asynchronously could help us avoid interrupting the main functionality of the app even if it is heavily loaded.

Now let's see what logging libraries are available in Java and what functionalities they provide, and compare their advantages and disadvantages.



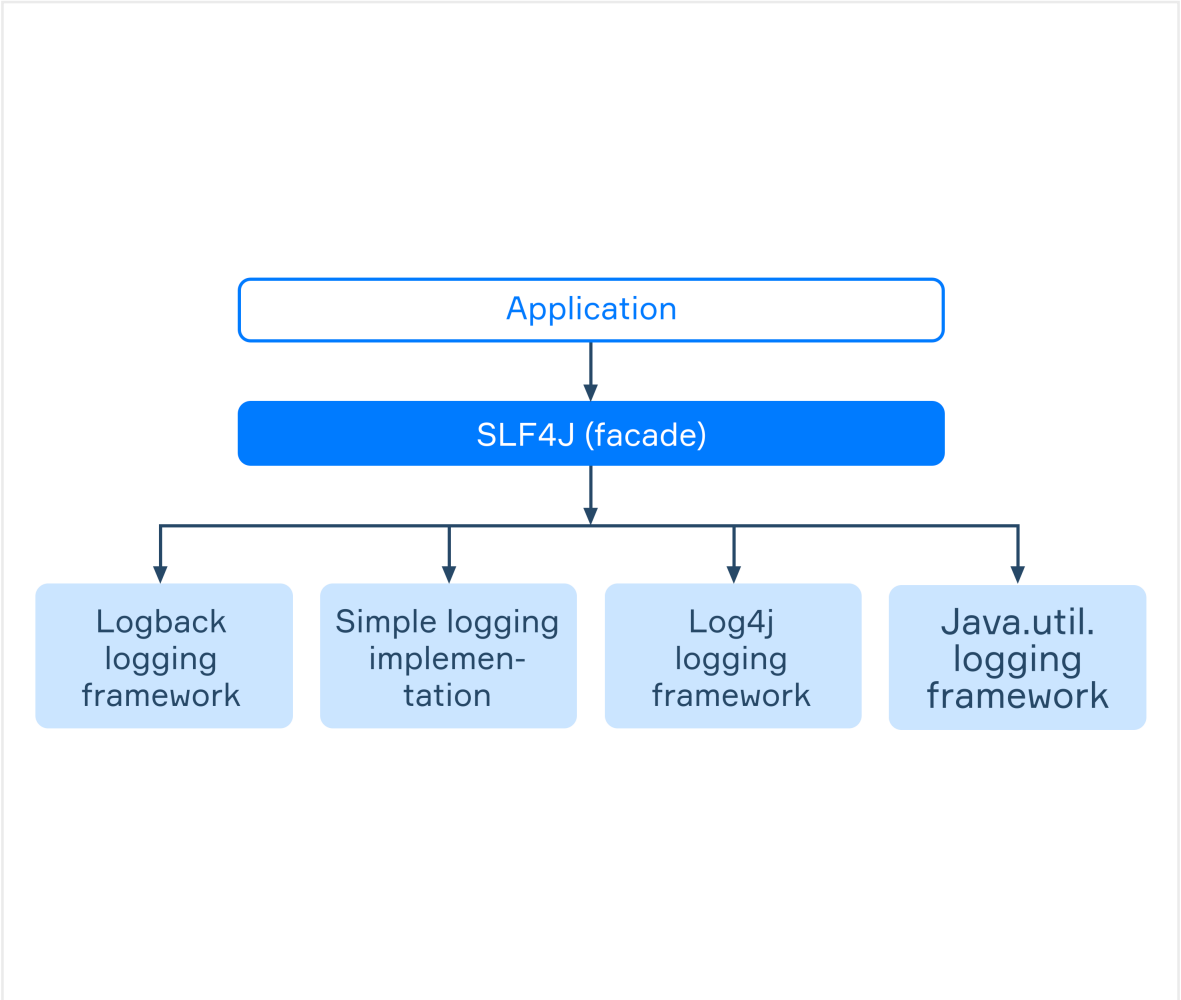
Log4j
The first version of Log4j was released in 1999. It quickly became very popular and remained so for a long time. Now it has been replaced by its successor, **Log4j 2**. It is a high-performance modern logging library that has many advanced features: apart from just strings, it supports logging messages, lambda expressions, filtering based on context, markers, regular expressions and other components of log events. Also, it allows users to create custom log levels.
Log4j 2's API is separate from the implementation, which allows for using it as a **facade** for other logging libraries such as Log4j version 1.2, SLF4J, Apache Commons Logging, and java.util.logging.

Facade here means that a library provides a simple but powerful interface to other logging libraries and frameworks, which allows us to easily use any logger hidden behind the facade's (for example, Log4j 2's) interface and not worry about the logger implementation.

JUL
java.util.logging.Logger, which is commonly referred to as **JUL**, was introduced in Java 1.4. Its most important benefit is that it is part of the Java standard

library, which means that it's always at hand and you don't need to add another dependency to your project. However, JUL is not very performant and less flexible than other popular logging libraries. Nevertheless, it's still a solid choice if you need logging but don't want to bother with third party solutions. Also, the functionality of JUL can be enhanced by using such facades as **SLF4J**.

SLF4J
Simple logging facade for Java or **SLF4J** serves as an abstraction or facade for various logging tools. SLF4J consists of the common part (API) and a few libraries which implement individual plug-in schemes: SLF4J -> Log4j, SLF4J -> java.util.logging, and so on. These libraries can be plugged in by simply putting them in a classpath. These features make SLF4J a log management tool. It is very popular and used in big projects, such as Jetty and Hibernate.



Logback
The Logback project was created as a successor to Log4j. It is an up-to-date and performant logging library that natively implements SLF4J's API to allow switching between Logback and other logging frameworks. Also, it integrates with Servlet containers, such as Tomcat and Jetty, and provides HTTP-access

logging functionality.
Logback was created by the same developer that worked on Log4j. It is based on the same concepts and provides a number of improvements, including (but not limited to) advanced filtering options and automatic reloading of logging configurations. This library is a close competitor to Log4j 2 and is used as a logger on modern high-performance servers.