Log4j

What is log4j? Open source logging framework written in Java and which is reliable, fast and flexible
Why logging? Helps in debugging, maintenance, tracing

Video goals:

- Integrate log4j with basic configuration and output application logs to console
- Use multiple appenders to output logs to console as well as to file
- Overwriting/appending log files
- Setting maximum log file size/deleting old logs
- Output logs to separate files in case of parallel execution. We will generate one log file for each device. The log file name and directory structure will be generated at run time based on the device parameters just like the way we did for screenshots and videos.

Components:
Loggers: capture logs
Appenders: publish logs to various targets like file, console, database, etc.
Layouts: format logs

Thread safe - supports parallel execution
Supports multiple appenders i.e. it can output log to console, file, database all at the same time

Logging levels:
All - all logs,
trace - finer grained informational events e.g. a stack trace when java exception occurs
debug - debug logs e.g. messages you would output while debugging issues (development phase)
info - progress logs e.g. login with user, entering password, etc.
warn - warning messages, execution can continue,
error - error logs, execution can continue e.g. exceptions, assertion failures
fatal - serious events, abort execution e.g. driver initialization failure

ALL < DEBUG < INFO < WARN < ERROR < FATAL < OFF

Logs can be filtered using setLevel(Level.X) method in the code or setting the level in the configuration file.

Only con is that heavy logging might slow down automation.

Steps:
Create maven project
Add log4j dependency
Create log4j2.properties or log4j2.xml file under src/main/resources folder. This file defines all the required configurations like loggers, appenders, layouts, etc. in the form of key value pair.

Two static methods:

- **public static Logger getRootLogger();**
- **public static Logger getLogger(String name);**

1st method: Application instance's root logger and does not have a name
2nd method: named logger object instance. Name can be any string you pass. Usually class name is passed.

Use in code:
static Logger log = Logger.getLogger(ExampleClass.class.getName());

Links:
Configurations: https://logging.apache.org/log4j/2.x/manual/configuration.html
Pattern Layouts: https://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/PatternLayout.html
Write to separate logs: https://logging.apache.org/log4j/2.0/faq.html#separate_log_files