Logging

Log4j

SLF4J provides 5 default logging levels in Spring boot:

**ERROR** - Error logs are serious issues that affect a significant part of the system or some part of your application has failed to operate. Exceptions are considered ERROR level logs. Other examples of error logs are database connection failures and configuration errors. ERROR logs are the most urgent default log level in SLF4J.

**WARN** - Warning logs are used to indicate potential problems that might cause errors and should be monitored in case they fail. Of course, the context of a warning is subjective to the developer and the situation so warning logs might vary from system to system.

**INFO** -INFO is the default logging level that is set by Spring Boot. If no configurations are made, the log level is automatically set to INFO. These types of logs are information that isn't normally needed but is useful in situations like production code debugging or determining when certain data is manipulated.

**DEBUG** - DEBUG logs include more detailed, specific information that isn't needed in normal situations. This is often set as a log level when a developer is trying to deep trace a problem or a bug that is hard to trace.

**TRACE** - TRACE is a more granular version of DEBUG. TRACE logs are exhaustive, imagine logging every single operation the system is doing, from starting a service, initializing new variables, and calling methods.

Steps:

No need to add dependencies in pom.xml for logging (in case of spring boot)

In application.properties:

logging.file.name=ajay.log

in any method, obtain the logger factory to get logger.

Logger logger = LoggerFactory.*getLogger*(Application.**class**);

logger.error("There is a problem");