15 min.

Private logger logger = LoggerFactory.getLogger(ControllerName.class);

This should we write in class outside method , then in any method we can write or log

Like logger.info and logger.warn , debug

Logging levels

It shows importance of log messages

1.Error

2.Warning

3.info

4.Debug

5.Trace

Arrow shows priority

Error : to show the error or exception occurred

Warn : to just show warning

Info : to show just normal info

Debug : to show the debug level info

Trace: to show tracing of application which method executed and not

Here error , warn and info is enabled by default and debug and trace is not

If we enable debug then all the levels above debug will be activate by default , level lower than will not get affected .

To enable:

In application.properties

logging.level.root = TRACE

IF WE enable trace , it will log all the detail log , which was not even a necessary , so recommended to use info.

We can separately provide the logging configuration info

By creating logback-configuration.xml and appender inside it.