LOG FILE:

- ➢ WHAT IS LOG FILE IN SPRING BOOT?

In Spring Boot, a log file is a file where the application writes its logging information. By default, Spring Boot supports logging using Logback, which is the default logging framework.

Logback provides the actual implementation of these logging actions. WHERE **SLF4J** provides the API that you use in your code to perform logging actions.

It provides Logging levels which are used to categorize and control the severity of log messages: LOGING LEVELS ARE:

- **TRACE**: Very detailed logging, used for fine-grained debugging.
- **DEBUG**: Detailed debugging information.
- **INFO**: General operational information.
- **WARN**: Indication of potential problems.
- **ERROR**: Serious issues that need immediate attention.

STEPS TO CREATE CUSTOM LOG FILE:

- ➢ Create a logback-spring.xml file in the src/main/resources directory. This file will contain the configuration for Logback.

```
➢   <configuration>

    <!-- Console appender -->
    <appender name="MYCONSOLE"
class="ch.qos.logback.core.ConsoleAppender">
        <encoder>
            <pattern>%d{yyyy-MM-dd HH:mm:ss} - %msg%n</pattern>
        </encoder>
    </appender>
    <!-- File appender -->
    <appender name="MYFILE"
class="ch.qos.logback.core.rolling.RollingFileAppender">
        <file>logs/app.log</file>
        <rollingPolicy
class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">

<fileNamePattern>logs/app-%d{yyyy-MM-dd}.log</fileNamePattern>
            <maxHistory>30</maxHistory>
        </rollingPolicy>
        <encoder>
            <pattern>%d{yyyy-MM-dd HH:mm:ss} [%thread] %-5level
%logger{36} - %msg%n</pattern>
        </encoder>
    </appender>

    <!-- Log level configuration -->
    <root level="info">
        <appender-ref ref="MYCONSOLE"/>
        <appender-ref ref="MYFILE"/>
    </root>

</configuration>
```

- ➢ START USING LOGGER:

```
➢  package com.app;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class Application {

   private static final Logger logger =
LoggerFactory.getLogger(Application.class);
   public static void main(String[] args) {
      SpringApplication.run(Application.class, args);
      logger.trace("This is a TRACE message.");
      logger.debug("This is a DEBUG message.");
      logger.info("This is an INFO message.");
      logger.warn("This is a WARN message.");
      logger.error("This is an ERROR message.");
   }

}
```

NOTE: INSTEAD OF CREATING IT'S INSTANCE YOU CAN USE @SLF2J ANNOTATION FOR DECLARING IT'S INSTANCE AND INSTEAD OF logger you can use log.info();