

**Log4J**

----------------------------------------------------

\*) Logging : getting messages from application (Production server)

like success, errors, warnings,..etc

\*) Moniter the process, find bugs from realtime environment.

\*) Tools :

1. Log4J 2.x

2. Java Logging

3. Commons Logging

..etc

\*) Log Messages can be stored at different places

Files, Database, Email , Console..etc

==================================================================

3 Components

1. Logger 2. Appender 3. Layout

(which classes) (where to store) (how to print)

\*) Logger : it is a base object in Log4J. It enables/activates

Logging/Moniter of a class/process.

class EmployeeController {

Logger log = LogManager.getLogger(EmployeeController.class);

}

\*) Do not create Logger object for Entity/DTO/POJO classes.

\*) Recomanded Controllers/Services..etc

--- Priority Methods -------------------------

These are pre-defined methods exist in Logger object

which prints messages based on type.

TRACE : To find messages from multiple stages/env/apps.

DEBUG : To print final result of a process/ internal details

INFO : Current Step

WARN : App related messages

ERROR : To give details about exception

FATAL : Environment related issues

ex:Database Connection is closed.

TRACE > DEBUG > INFO > WARN > ERROR > FATAL

=============================================================

Appender:- Location of messages to be stored. (classes)

1. \*\*FileAppender : Write data to Log File (\_\_\_.log)

2. ConsoleAppender : Prints data at console

3. JDBCAppender : Store messages at Database table

4. SMTPAppender : Write messages to Email.

\*) 1 Project can have even multiple appenders.

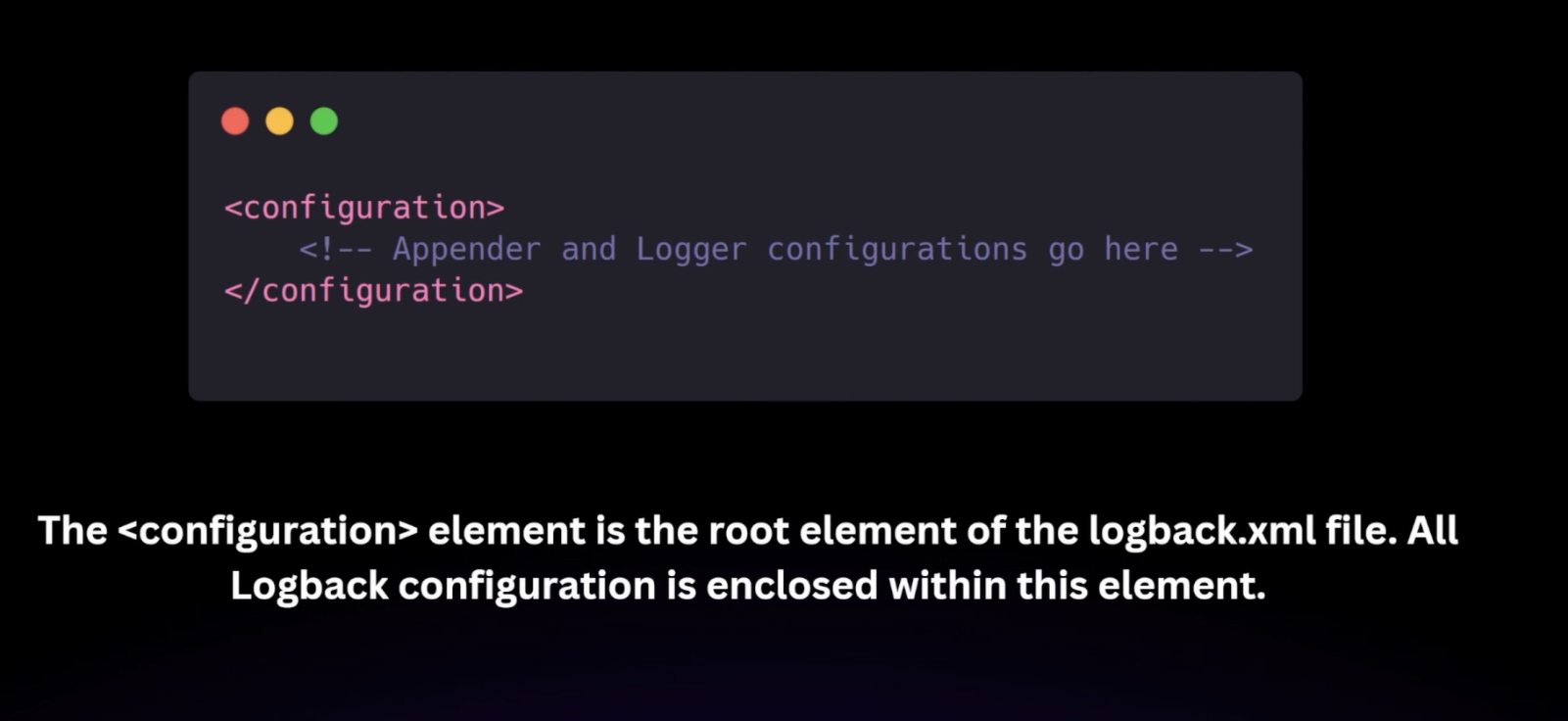
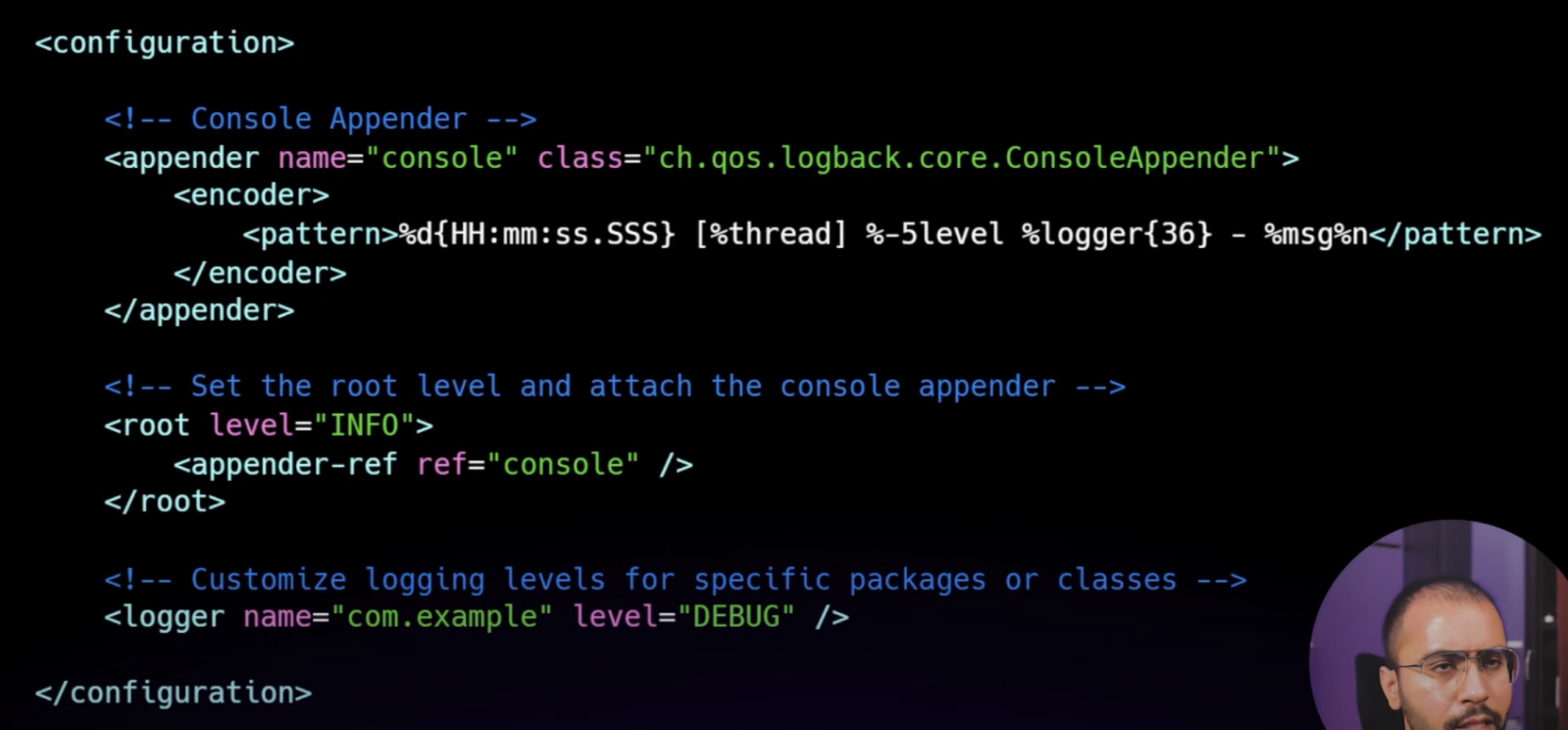
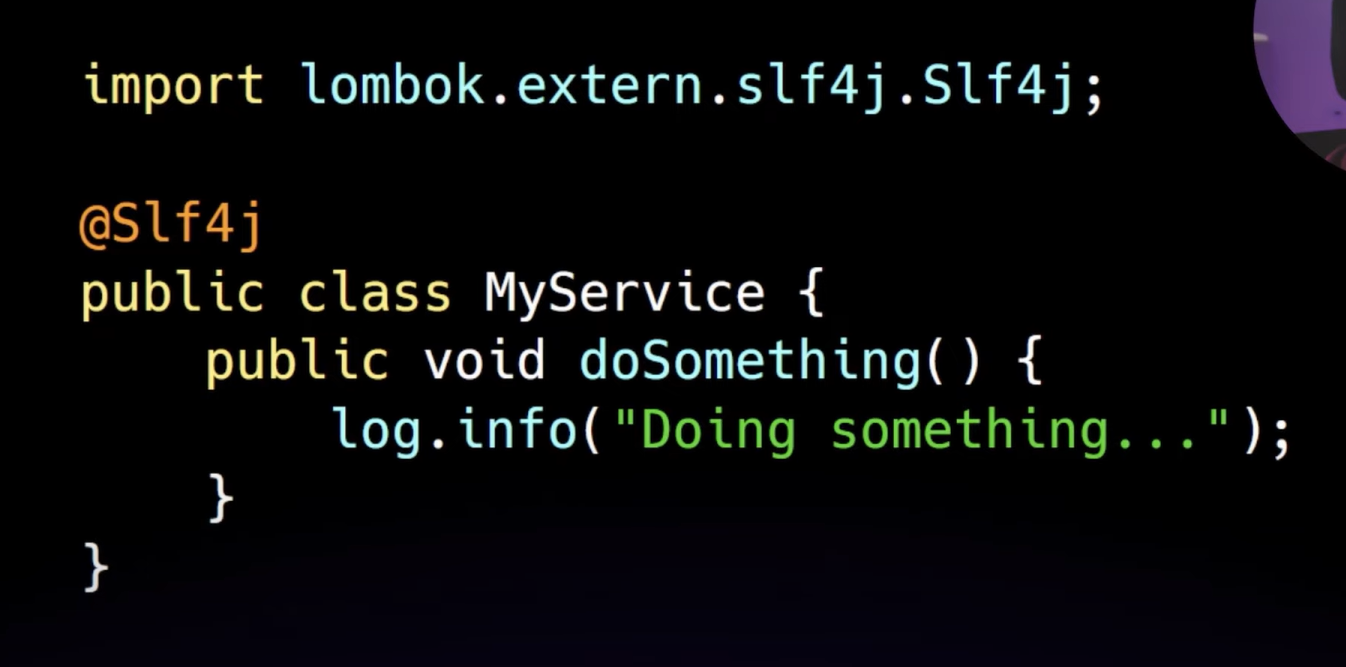
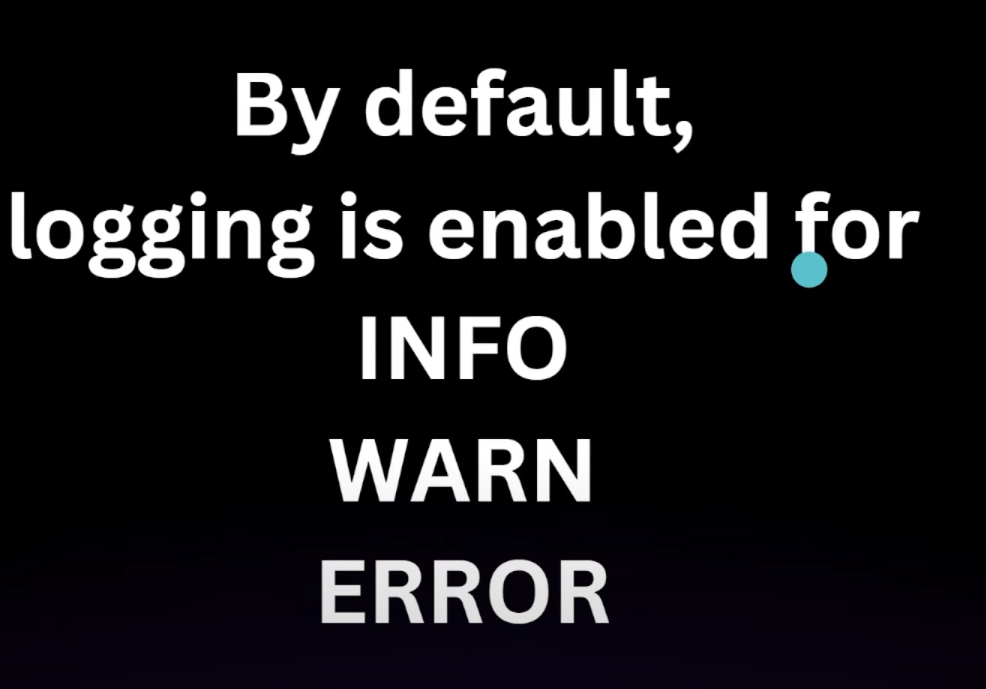
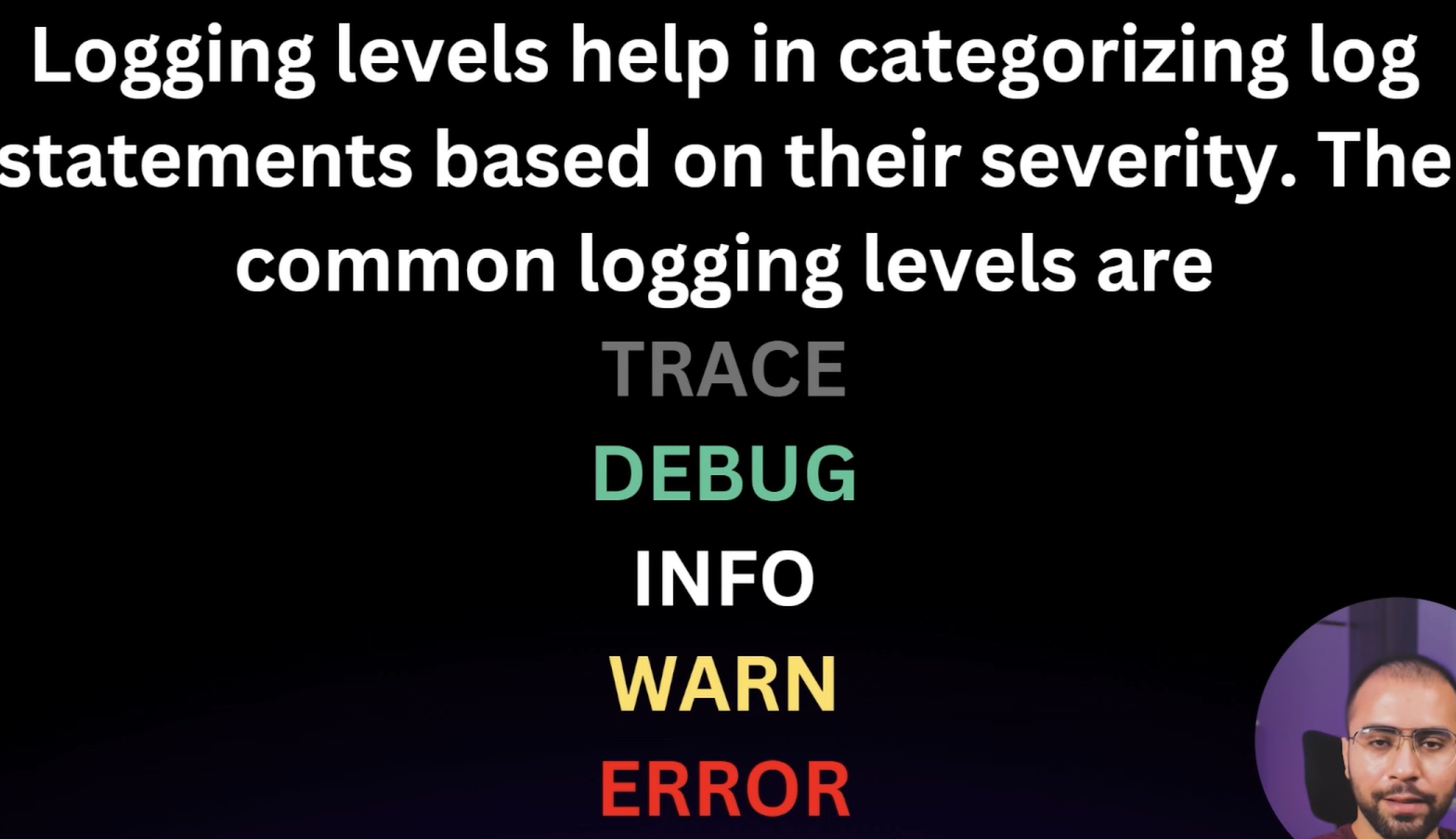
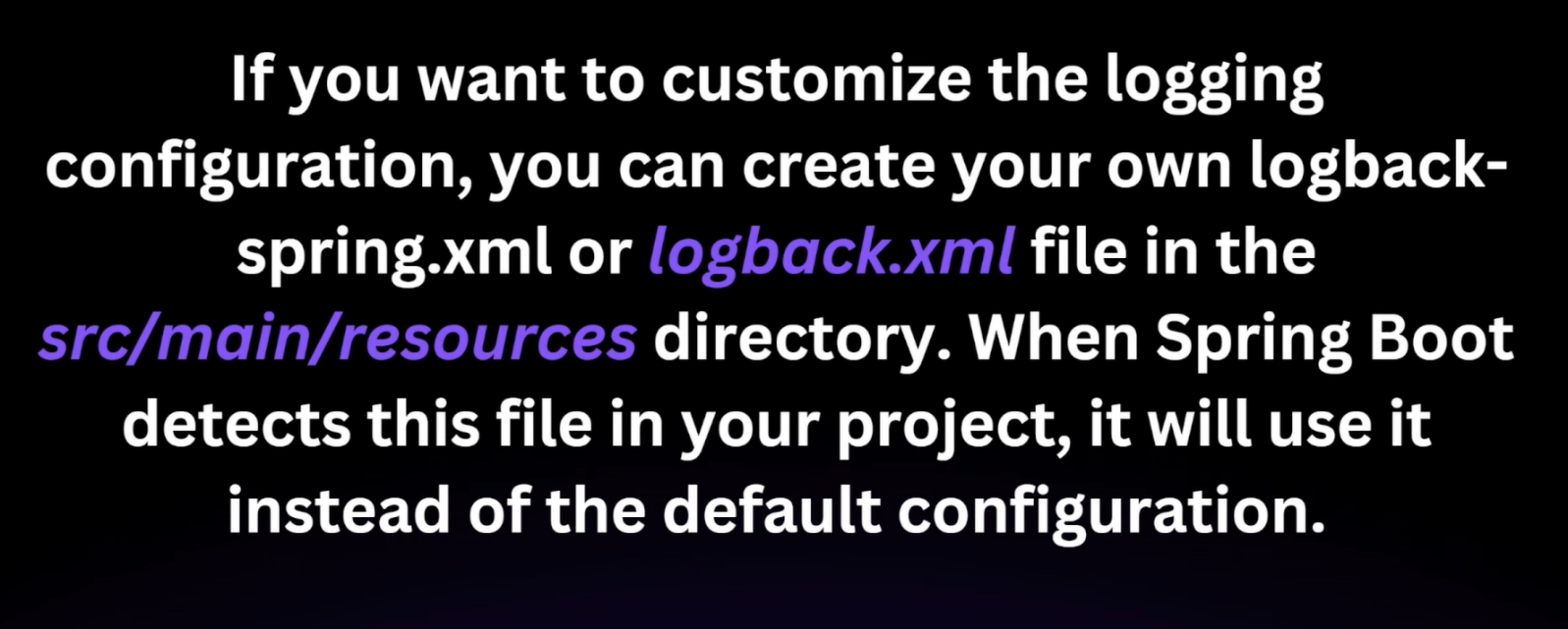
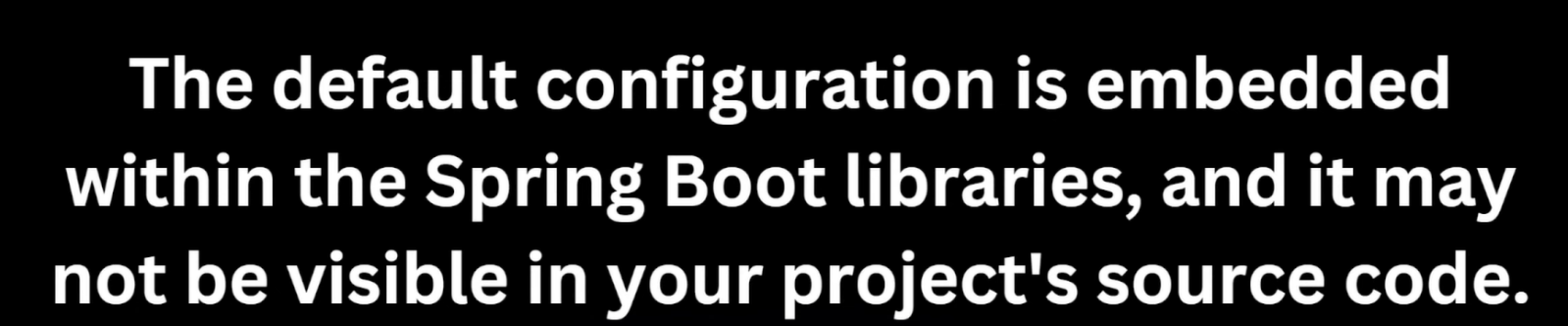
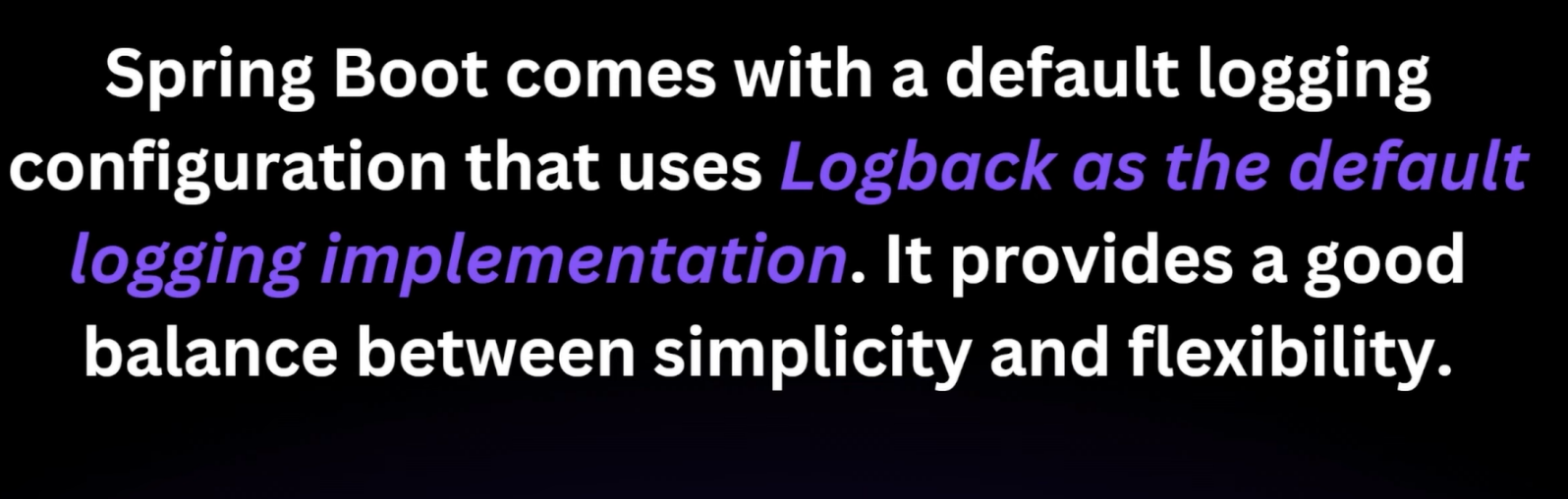
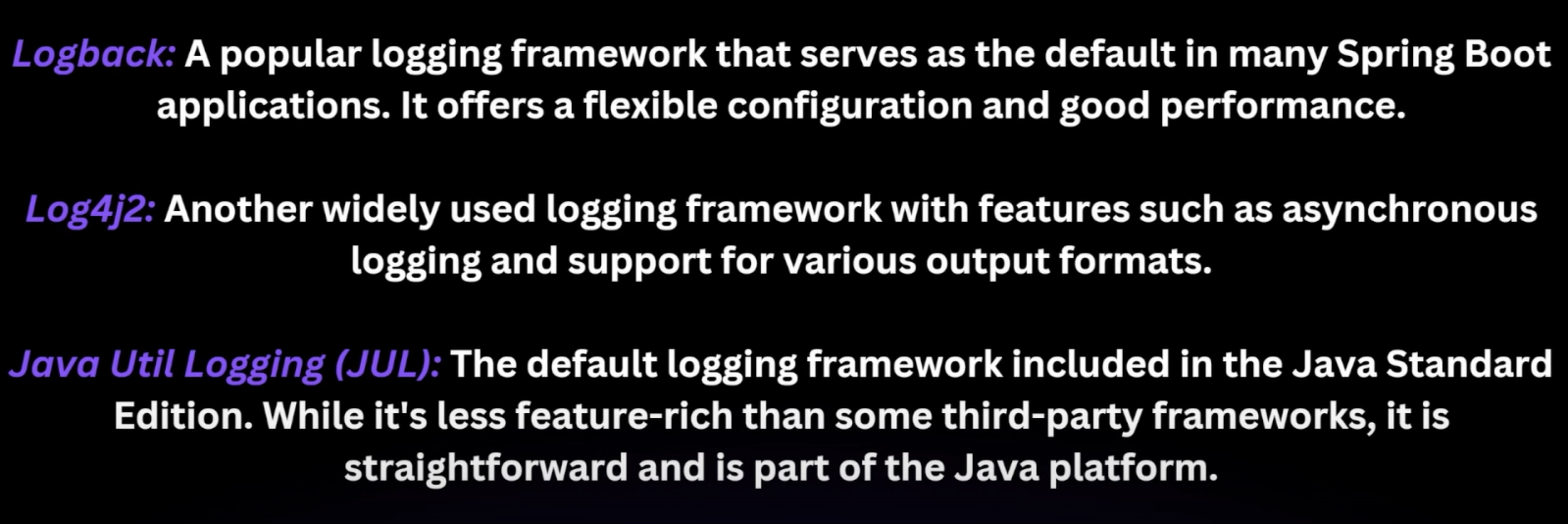
Layout: Message Format/Pattern.

1. Default Layout (Message and NextLine)

2. HTML Layout (\_\_\_.html)

3. XML Layout (\_\_\_.xml)

4.\*\*\* PatternLayout (Your own format)



Logback.xml

<?xml version="1.0" encoding="UTF-8" ?>  
<configuration>  
 <appender name="console" class="ch.qos.logback.core.ConsoleAppender">  
 <encoder>  
 <pattern>  
 %d{yy:MM:dd HH:mm:ss.SSS} %-5level --- [%thread] %logger{36}: %msg %n  
 </pattern>  
 </encoder>  
 </appender>  
 <appender name="file" class="ch.qos.logback.core.rolling.RollingFileAppender">  
 <file>  
 portfolio-log.log  
 </file>  
 <rollingPolicy class="ch.qos.logback.core.rolling.SizeAndTimeBasedRollingPolicy">  
 <fileNamePattern>portfolio-log-%d{yy-MM-dd\_HH-mm}.%i.log</fileNamePattern>  
 <maxFileSize>10MB</maxFileSize>  
 <maxHistory>10</maxHistory>  
 </rollingPolicy>  
 <encoder>  
 <pattern>  
 %d{yy:MM:dd HH:mm:ss.SSS} %-5level --- [%thread] %logger{36}: %msg %n  
 </pattern>  
 </encoder>  
  
 </appender>  
 <root level="INFO">  
 <appender-ref ref="console"/>  
 <appender-ref ref="file"/>  
 </root>  
</configuration>

Output

24:03:17 20:09:18.589 INFO --- [main] c.p.DockerTestiingOnPortFolioApplication: Starting DockerTestiingOnPortFolioApplication using Java 19.0.2 with PID 2964 (D:\Pankaj Sir SpringBoot\Docker-Testiing-on-PortFolio\target\classes started by ASHUTOSH in D:\Pankaj Sir SpringBoot\Docker-Testiing-on-PortFolio)

24:03:17 20:09:18.595 INFO --- [main] c.p.DockerTestiingOnPortFolioApplication: No active profile set, falling back to 1 default profile: "default"