**Logging using SLF4J**

Exercise 1: Logging Error Messages and Warning Levels

Task: Write a Java application that demonstrates logging error messages and warning levels using SLF4J.

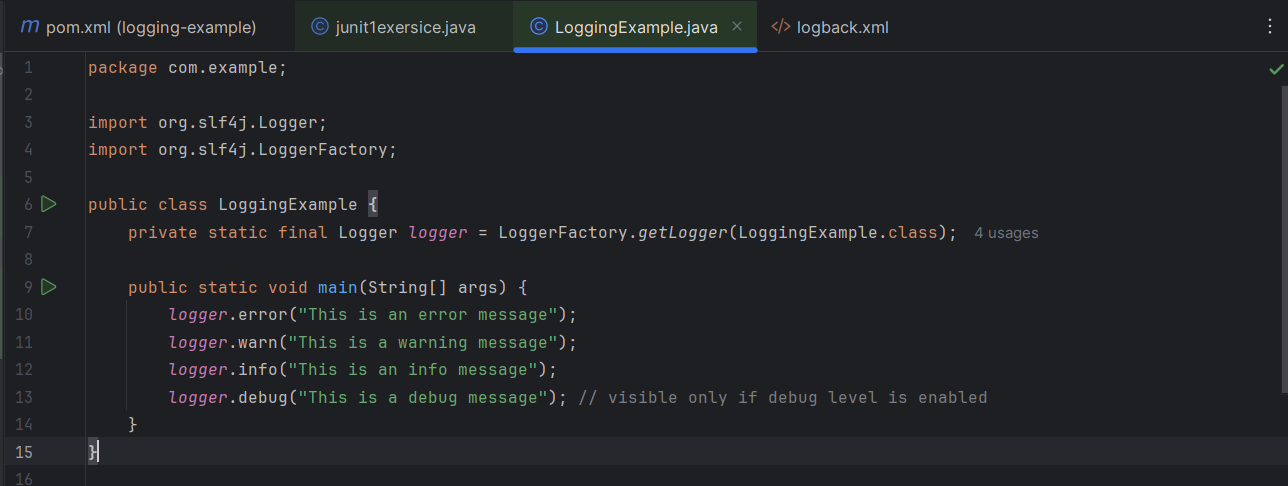
Step-by-Step Solution:

1. Add SLF4J and Logback dependencies to your `pom.xml` file:

org.slf4j slf4j-api 1.7.30 ch.qos.logback logback-classic 1.2.3 2. Create a Java class that uses SLF4J for logging: import org.slf4j.Logger; import org.slf4j.LoggerFactory; public class LoggingExample { private static final Logger logger = LoggerFactory.getLogger(LoggingExample.class); public static void main(String[] args) { logger.error("This is an error message"); logger.warn("This is a warning message"); } }

**SOLUTION:**

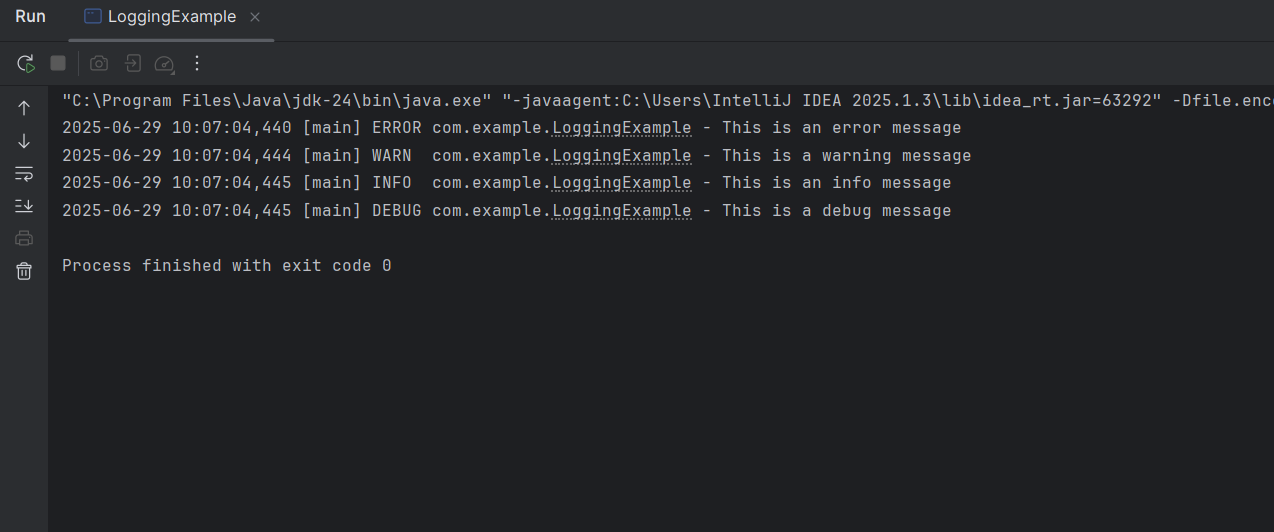
**LoggingExample:**

****

**Logback.xml:**

****

**OUTPUT:**

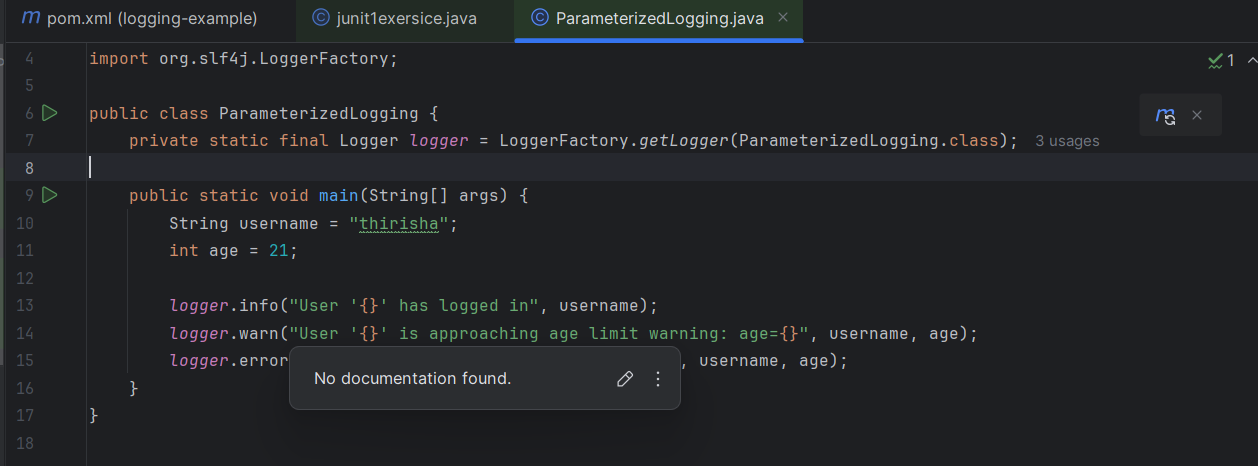
****

**Exercise 2: Parameterized Logging**

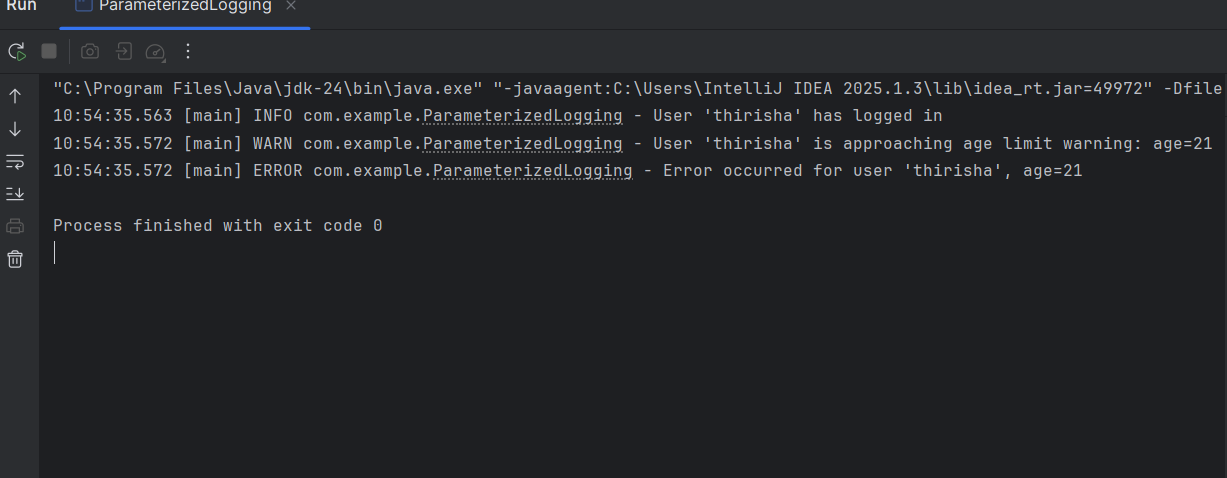
**Task: Write a Java application that demonstrates parameterized logging using SLF4J. Step-by-Step Solution:**

**1. Add SLF4J and Logback dependencies to your `pom.xml` file: org.slf4j slf4j-api 1.7.30 ch.qos.logback logback-classic 1.2.3 2. Create a Java class that uses SLF4J for parameterized logging: Write code for this.**

**SOLUTION:**

****

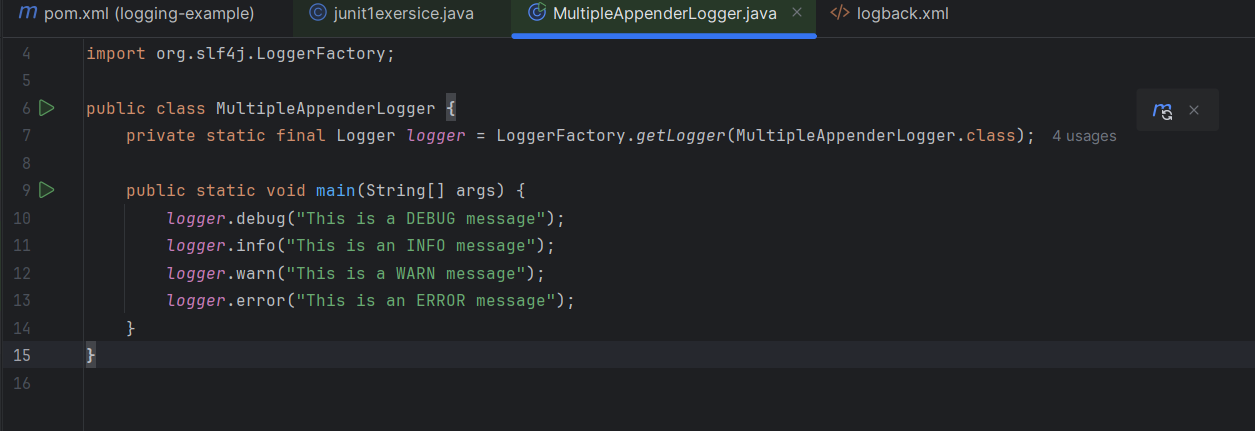
**OUTPUT:**

****

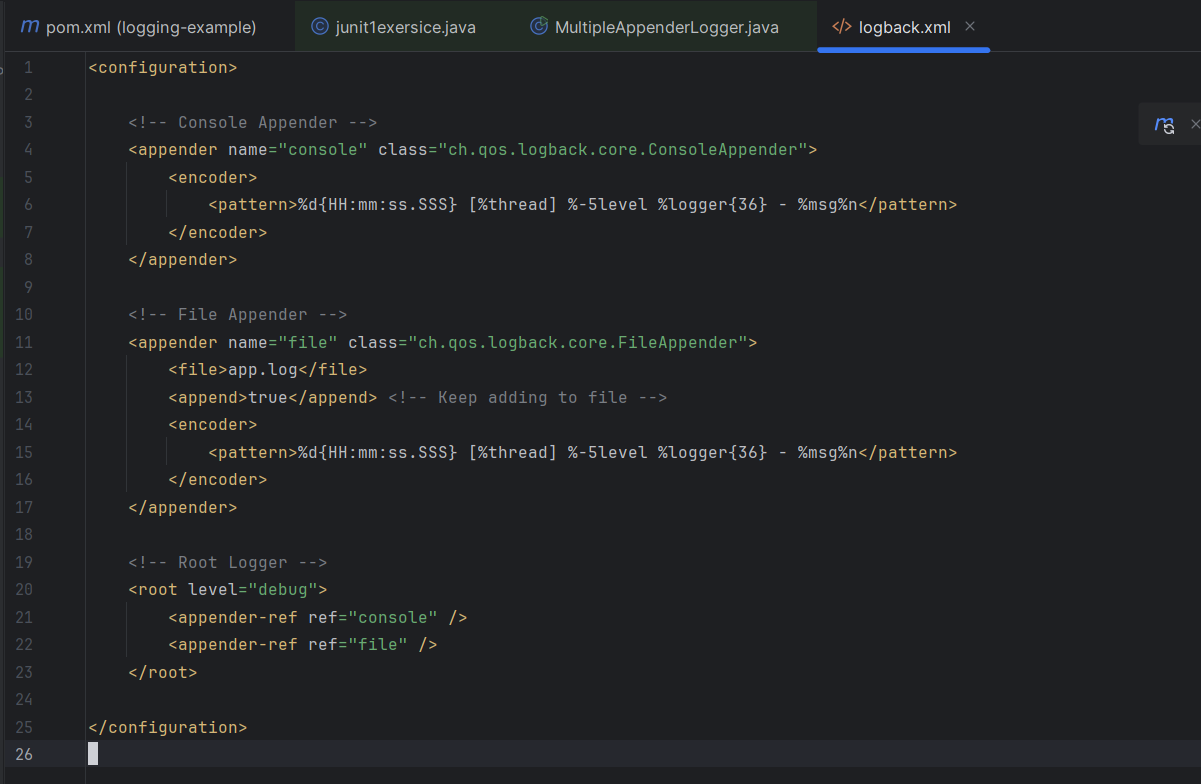
**Exercise 3: Using Different Appenders Task: Write a Java application that demonstrates using different appenders with SLF4J. Step-by-Step Solution: 1. Add SLF4J and Logback dependencies to your `pom.xml` file: org.slf4j slf4j-api 1.7.30 ch.qos.logback logback-classic 1.2.3 2. Create a `logback.xml` configuration file to define different appenders: %d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n app.log %d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n 3. Create a Java class that uses SLF4J for logging: Write code for this**

**SOLUTION:**

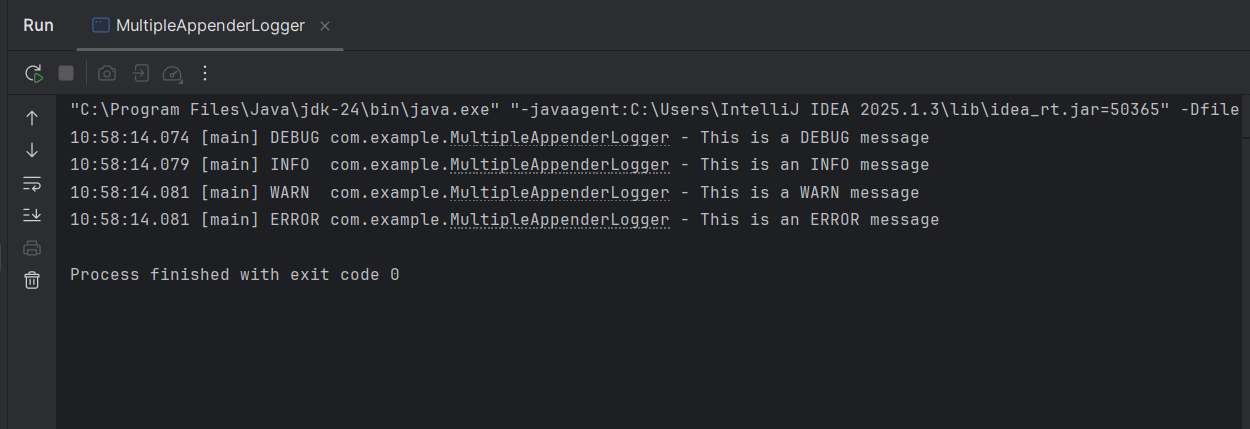
**MultipleAppenderLogger.java:**

****

**Logback.xml:**

****

**OUTPUT:**

****