**Logging using SLF4J**

**Exercise 1: Logging Error Messages and Warning Levels**

### ****Step 1: Add Dependencies to**** pom.xml

<dependencies>

<!-- SLF4J API -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

<!-- Logback Implementation -->

<dependency>

<groupId>ch.qos.logback</groupId>

<artifactId>logback-classic</artifactId>

<version>1.2.3</version>

</dependency></dependencies>

**Step 2: Java Logging Code**

Create a file named LoggingExample.java:

package com.example;

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");

logger.info("This is an info message");

logger.debug("This is a debug message");

logger.trace("This is a trace message");

}

}

**Step 3: Configuration File –** logback.xml

Place this file in src/main/resources:

<configuration>

<appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">

<encoder>

<pattern>%d{yyyy-MM-dd HH:mm:ss} %-5level %logger{36} - %msg%n</pattern>

</encoder>

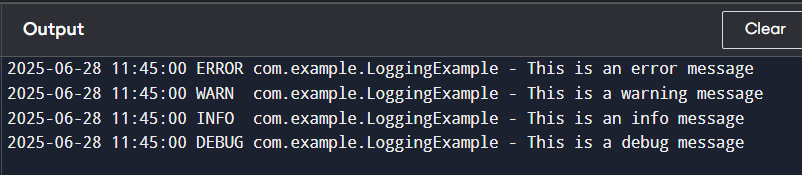
</appender>

<root level="debug">

<appender-ref ref="STDOUT" />

</root></configuration>

**OUTPUT:**



**Exercise 2: Parameterized Logging**

### ****Step 1: Add Dependencies in**** pom.xml

<dependencies>

<!-- SLF4J API -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

<!-- Logback Implementation -->

<dependency>

<groupId>ch.qos.logback</groupId>

<artifactId>logback-classic</artifactId>

<version>1.2.3</version>

</dependency></dependencies>

**Step 2: Java Code for Parameterized Logging**

package com.example;

import org.slf4j.Logger;import org.slf4j.LoggerFactory;

public class ParameterizedLoggingExample {

private static final Logger logger = LoggerFactory.getLogger(ParameterizedLoggingExample.class);

public static void main(String[] args) {

String userName = "Bindhu";

int userAge = 22;

double accountBalance = 1550.75;

logger.info("User {} is {} years old.", userName, userAge);

logger.debug("Account balance for user {} is Rs. {}", userName, accountBalance);

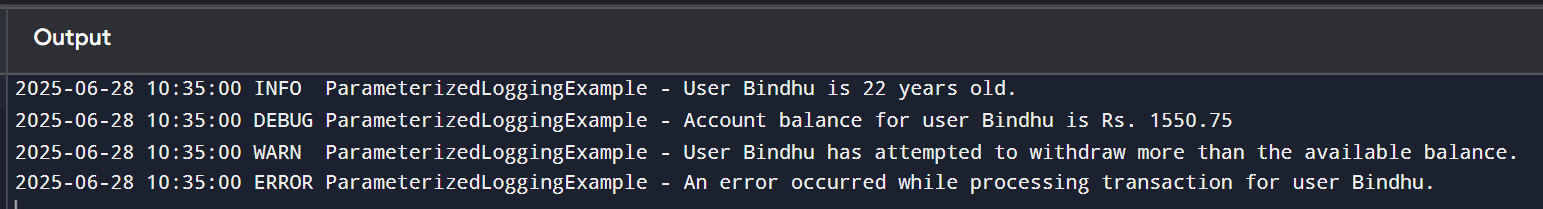
logger.warn("User {} has attempted to withdraw more than the available balance.", userName);

logger.error("An error occurred while processing transaction for user {}.", userName);

}

}

**OUTPUT:**



**Exercise 3: Using Different Appenders**

### ****Step 1: Add Dependencies in**** pom.xml

<dependencies>

<!-- SLF4J API -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

<!-- Logback Implementation -->

<dependency>

<groupId>ch.qos.logback</groupId>

<artifactId>logback-classic</artifactId>

<version>1.2.3</version>

</dependency></dependencies>

**Step 2:** logback.xml **Configuration for Multiple Appenders**

Place this file in src/main/resources/logback.xml:

<configuration>

<!-- Console Appender -->

<appender name="console" class="ch.qos.logback.core.ConsoleAppender">

<encoder>

<pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>

</encoder>

</appender>

<!-- File Appender -->

<appender name="file" class="ch.qos.logback.core.FileAppender">

<file>app.log</file>

<append>true</append>

<encoder>

<pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>

</encoder>

</appender>

<!-- Root Logger with both appenders -->

<root level="debug">

<appender-ref ref="console" />

<appender-ref ref="file" />

</root>

</configuration>

**Step 3: Java Class to Log Messages**

Create MultipleAppendersExample.java in src/main/java/com/example/:

package com.example;

import org.slf4j.Logger;import org.slf4j.LoggerFactory;

public class MultipleAppendersExample {

private static final Logger logger = LoggerFactory.getLogger(MultipleAppendersExample.class);

public static void main(String[] args) {

logger.info("This is an info message logged to both console and file.");

logger.warn("This is a warning message.");

logger.error("This is an error message.");

logger.debug("This is a debug message.");

}

}

**OUTPUT:**

