**Logging using SLF4J**

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

**Code:**

**<!--pom.xml-->**

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example.slf4j</groupId>

<artifactId>slf4j\_logging</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>17</maven.compiler.source>

<maven.compiler.target>17</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

<dependency>

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

<artifactId>logback-classic</artifactId>

<version>1.2.3</version>

</dependency>

</dependencies>

</project>

**//LoggingExample.java**

package com.example.slf4j;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class LoggingExample {

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

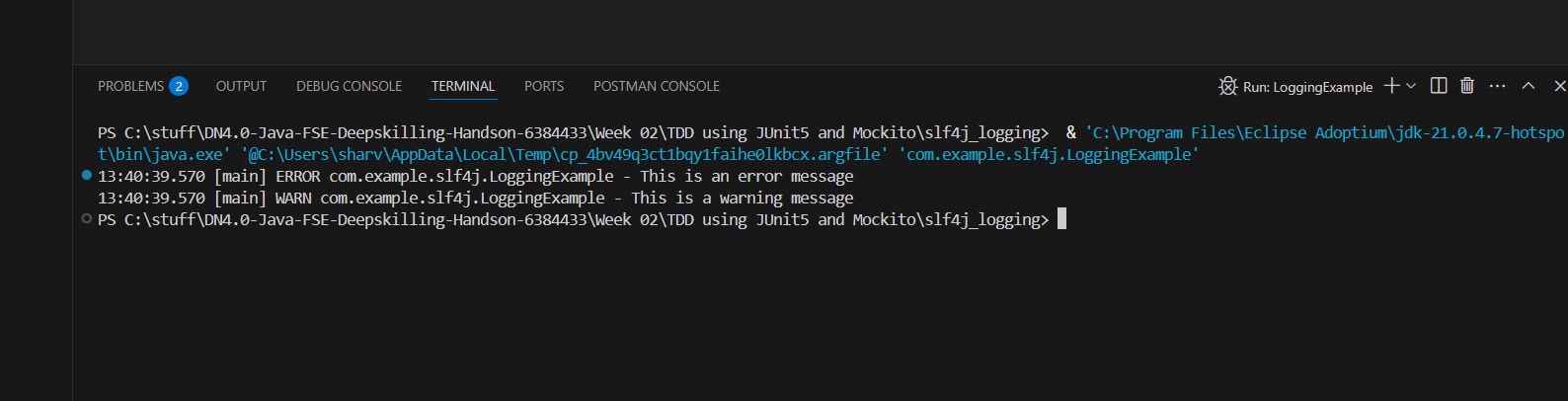
public static void main(String[] args) {

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

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

}

}

**Output:**

**Exercise 2: Parameterized Logging**

**Code:**

**//ParameterizedLoggingExample.java**

package com.example.slf4j;

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 user = "Alice";

int userId = 101;

logger.info("User {} has logged in with ID {}", user, userId);

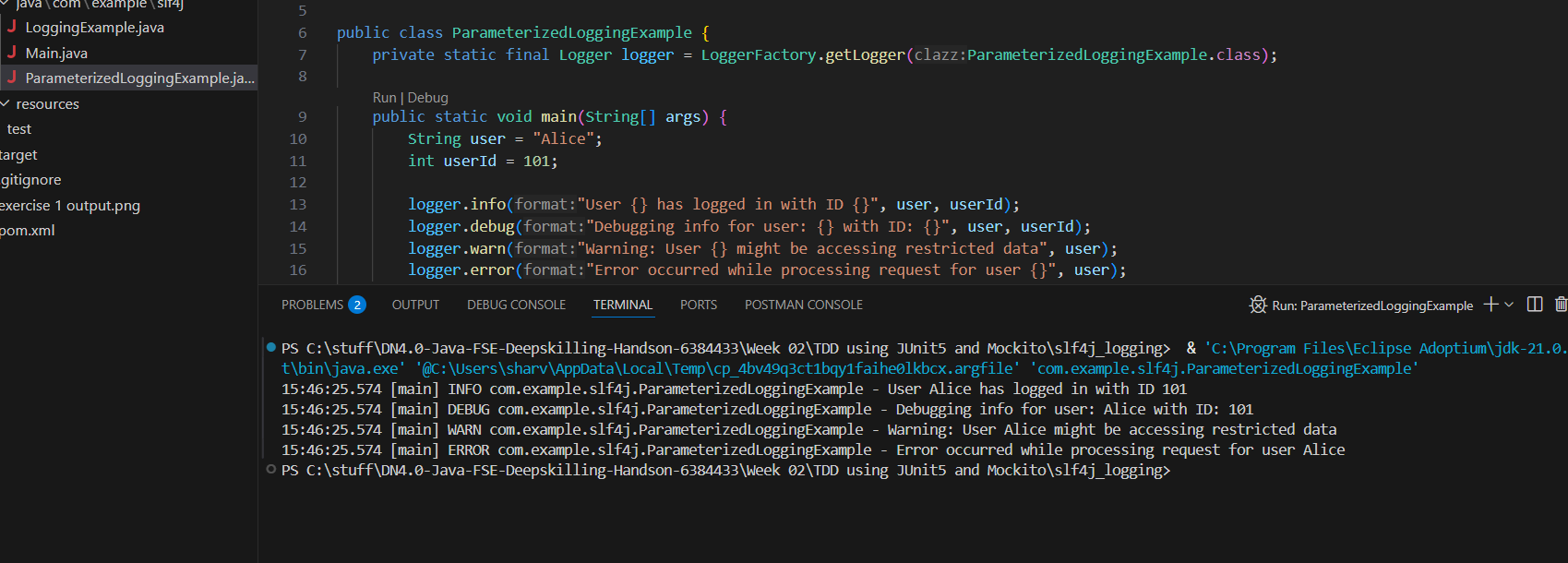
logger.debug("Debugging info for user: {} with ID: {}", user, userId);

logger.warn("Warning: User {} might be accessing restricted data", user);

logger.error("Error occurred while processing request for user {}", user);

}

}

**Output:**

**Exercise 3: Using Different Appenders**

**Code:**

**<!--logback.xml-->**

<configuration>

<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>

<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 level="debug">

<appender-ref ref="console" />

<appender-ref ref="file" />

</root>

</configuration>

**//AppenderLoggingExample.java**

package com.example.slf4j;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class AppenderLoggingExample {

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

public static void main(String[] args) {

logger.info("This message will appear in both console and file.");

logger.debug("Debug message written to both appenders.");

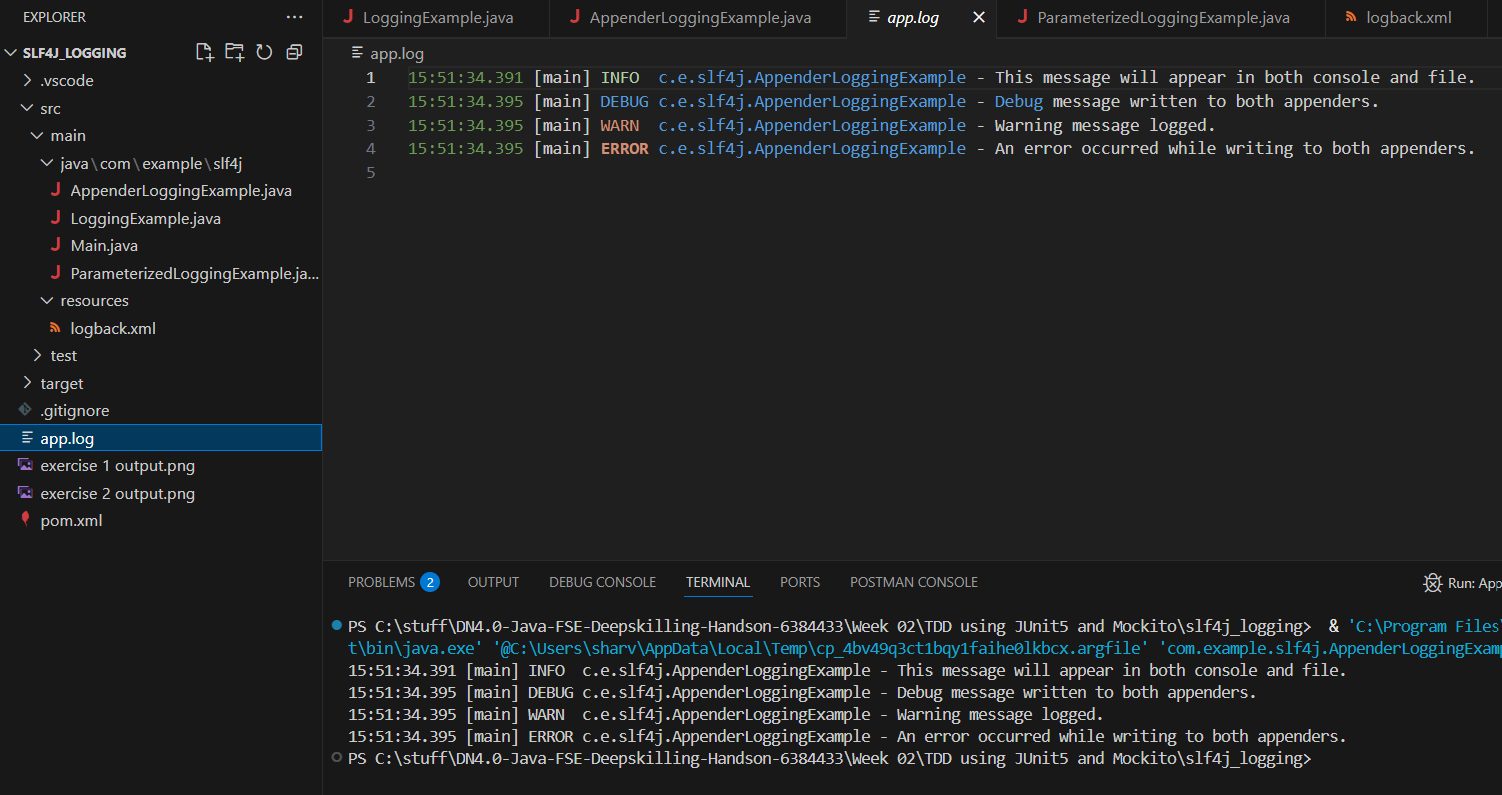
logger.warn("Warning message logged.");

logger.error("An error occurred while writing to both appenders.");

}

}

**Output:**

****