**SL4J LOGGING.**

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

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

**SOLUTION:**

**POM.xml:**

<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</groupId>

<artifactId>logging-demo</artifactId>

<version>1.0-SNAPSHOT</version>

<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 CLASS:**

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");

}

}

**LOGBACK.xml:**

<configuration>

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

<encoder>

<pattern>%d{yyyy-MM-dd HH:mm:ss} [%level] %logger{10} - %msg%n</pattern>

</encoder>

</appender>

<root level="debug">

<appender-ref ref="STDOUT"/>

</root>

</configuration>

**OUTPUT:**

2025-06-27 14:32:10 [ERROR] LoggingExample - This is an error message

2025-06-27 14:32:10 [WARN ] LoggingExample - This is a warning message

**Exercise 2: Parameterized Logging**

Task: Write a Java application that demonstrates parameterized logging using SLF4J.

**SOLUTION:**

POM.xml:

<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>

**PARAMETERIZEDLOGGINGEXAMPLE CLASS:**

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 items = 5;

double price = 19.99;

logger.info("User {} has added {} items to the cart. Total price: ${}", user, items, price);

String operation = "database update";

boolean success = false;

logger.warn("The operation '{}' completed with success status: {}", operation, success);

}

}

**OUTPUT:**

2025-06-27 15:10:45 [INFO ] ParameterizedLoggingExample - User Alice has added 5 items to the cart. Total price: $19.99

2025-06-27 15:10:45 [WARN ] ParameterizedLoggingExample - The operation 'database update' completed with success status: false

**Exercise 3: Using Different Appenders**

Task: Write a Java application that demonstrates using different appenders with SLF4J.

**SOLUTION:**

**POM.xml:**

<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>

**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>

**APPENDEREXAMPLE CLASS:**

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class AppenderExample {

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

public static void main(String[] args) {

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

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

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

}

}

**OUTPUT:**

15:20:10.123 [main] INFO AppenderExample - This is an informational message.

15:20:10.125 [main] WARN AppenderExample - This is a warning.

15:20:10.126 [main] ERROR AppenderExample - This is an error message.