1. **pom.xml dependencies (Maven 4.0)**

<dependencies>

<!-- SLF4J 2 API -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>2.0.13</version>

</dependency>

<!-- Logback implementation compiled for SLF4J 2 -->

<dependency>

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

<artifactId>logback-classic</artifactId>

<version>1.4.14</version>

</dependency>

</dependencies>

**2  src/main/resources/logback.xml**

<configuration>

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

<encoder>

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

</encoder>

</appender>

<root level="INFO">

<appender-ref ref="STDOUT" />

</root>

</configuration>

**3.  src/main/java/com/example/OrderServiceDemo.java**

package com.example;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

/\*\*

\* Demonstrates SLF4J 2.x logging of WARN and ERROR levels.

\*/

public class OrderServiceDemo {

private static final Logger log = LoggerFactory.getLogger(OrderServiceDemo.class);

public static void main(String[] args) {

OrderServiceDemo svc = new OrderServiceDemo();

svc.placeOrder("A‑123", 3); // happy path

svc.placeOrder("B‑456", 0); // triggers warning

svc.placeOrder(null, 5); // triggers error

}

public void placeOrder(String productCode, int quantity) {

if (productCode == null) {

log.error("Order rejected: productCode is null");

return;

}

if (quantity <= 0) {

log.warn("Suspicious quantity {} for product {}", quantity, productCode);

quantity = 1; // reset to minimum

}

// --- pretend to talk to DB / API -----------------------------------

try {

log.info("Placing order: product={} qty={}", productCode, quantity);

// ... imagine DAO.saveOrder(...) here ...

log.debug("Order saved successfully (fake primary key 42)");

} catch (RuntimeException ex) {

log.error("Database failure while placing order {}", productCode, ex);

}

}

}

**4  Run it**

mvn -q exec:java -Dexec.mainClass="com.example.OrderServiceDemo"