**Module 3 : SLF4J Logging**

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

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

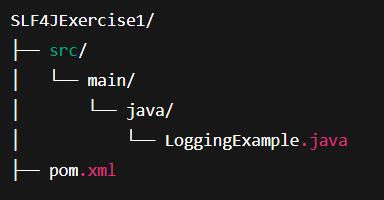
**Solution:**

**Step 1:**

**Description:** This exercise demonstrates how to use the SLF4J logging framework in a Java application to log error messages and warning levels. SLF4J (Simple Logging Facade for Java) provides a simple abstraction of various logging frameworks such as Logback, Log4j, and java.util.logging.

**Step 2:**

**File directory Structure:**

****

**Step 3:**

**Add the following dependencies to ‘pom.xml’ file:**

**<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>**

**Step 4:**

**Java Code [ LoggingExample.java ]:**

**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");**

**}**

**}**

**Step 5:**

**Install Maven and set it up. (Download it and copy the bin path to environment variables)  
Maven** is a powerful build automation and dependency management tool for Java projects. Think of it as the Project Manager for your Java apps.

**It handles:**

* Downloading libraries (like SLF4J, Logback, JUnit)
* Compiling your code
* Running your app
* Packaging it (into .jar/.war)
* Running tests
* Keeping your project organized

**Step 6:**

Use the following commands to compile and run the project in java using maven:

**Code:**

1. mvn compile



1. mvn exec:java

