Logging using SLF4J

Exercise 1: Logging Error Messages and Warning Levels Task:

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

Step-by-Step Solution:

1. Add SLF4J and Logback dependencies to your `pom.xml` file:

<dependency>

<groupId>org.slf4j</groupId>

<artifactld>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

<dependency>

<groupld>ch.qos.logback</groupld>

<artifactld>logback-classic</artifactld>

<version>1.2.3</version>

</dependency>

2. Create a Java class that uses SLF4J for logging:

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");

}

}

SOLUTION:

Step 1: Add SLF4J and Logback to 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>

Step 2: Java Class Using SLF4J Logging

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class AppLogger {

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

public static void main(String[] args) {

log.error("Application encountered a critical error.");

log.warn("This is a warning that might need attention.");

}

}

Expected Output:

ERROR AppLogger - Application encountered a critical error.

WARN AppLogger - This is a warning that might need attention.