Exercise 1: Logging Error Messages and Warning Levels

Code

<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

 <modelVersion>4.0.0</modelVersion>

 <groupId>com.example</groupId>

 <artifactId>junit-demo</artifactId>

 <version>0.0.1-SNAPSHOT</version>

 <dependencies>

   <!-- JUnit 5 -->

   <dependency>

     <groupId>org.junit.jupiter</groupId>

     <artifactId>junit-jupiter</artifactId>

     <version>5.10.0</version>

     <scope>test</scope>

   </dependency>

   <!-- Mockito -->

   <dependency>

     <groupId>org.mockito</groupId>

     <artifactId>mockito-core</artifactId>

     <version>5.12.0</version>

     <scope>test</scope>

   </dependency>

   <dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

<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>

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

package com.example;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class ParameterizedLogging {

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

   public static void main(String[] args) {

       String user = "Sayan";

       int loginAttempts = 3;

***logger***.info("User {} has attempted to log in {} times", user, loginAttempts);

***logger***.debug("Detailed debug log for user: {}", user);

   }

}

Exercise 3: Using Different Appenders

Code

<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>logs/app.log</file>

       <encoder>

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

       </encoder>

   </appender>

   <root level="debug">

       <appender-ref ref="console" />

       <appender-ref ref="file" />

   </root>

</configuration>

package com.example;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class MultipleAppenderExample {

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

   public static void main(String[] args) {

***logger***.info("Logging to both console and file");

***logger***.warn("This is a warning with multiple appenders");

***logger***.error("Error message written to both outputs");

   }

}