

Logging using SLF4J

Exercise 1: Logging Error Messages and Warning Levels

Pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<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-example</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <maven.compiler.source>24</maven.compiler.source>
    <maven.compiler.target>24</maven.compiler.target>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
  <dependencies>
    <!-- SLF4J API -->
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-api</artifactId>
      <version>1.7.30</version>
    </dependency>

    <!-- Logback (backend for SLF4J) -->
    <dependency>
      <groupId>ch.qos.logback</groupId>
```

```
<artifactId>logback-classic</artifactId>
<version>1.2.3</version>
</dependency>
</dependencies>
```

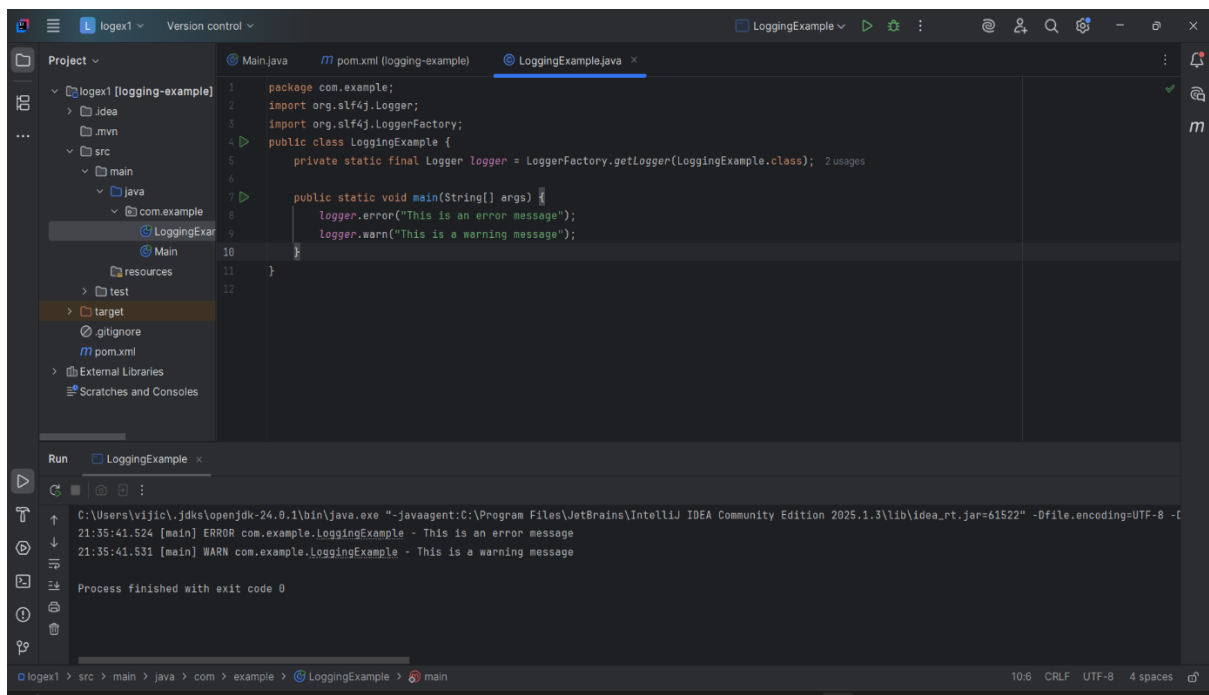
```
</project>
```

LoggingExample.java:

```
package com.example;
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:

The screenshot shows an IDE window with a project named 'logex1'. The 'src/main/java/com/example' directory is expanded, showing 'LoggingExample.java'. The code in the editor matches the provided code block. The 'Run' button is clicked, and the output console shows the following:

```
C:\Users\vijic\jdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.3\lib\idea_rt.jar=61522" -Dfile.encoding=UTF-8 -I
21:35:41.524 [main] ERROR com.example.LoggingExample - This is an error message
21:35:41.531 [main] WARN com.example.LoggingExample - This is a warning message
Process finished with exit code 0
```

Exercise 2: Parameterized Logging

Pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>slf4j-logging</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <maven.compiler.source>24</maven.compiler.source>
    <maven.compiler.target>24</maven.compiler.target>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
  <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>
```

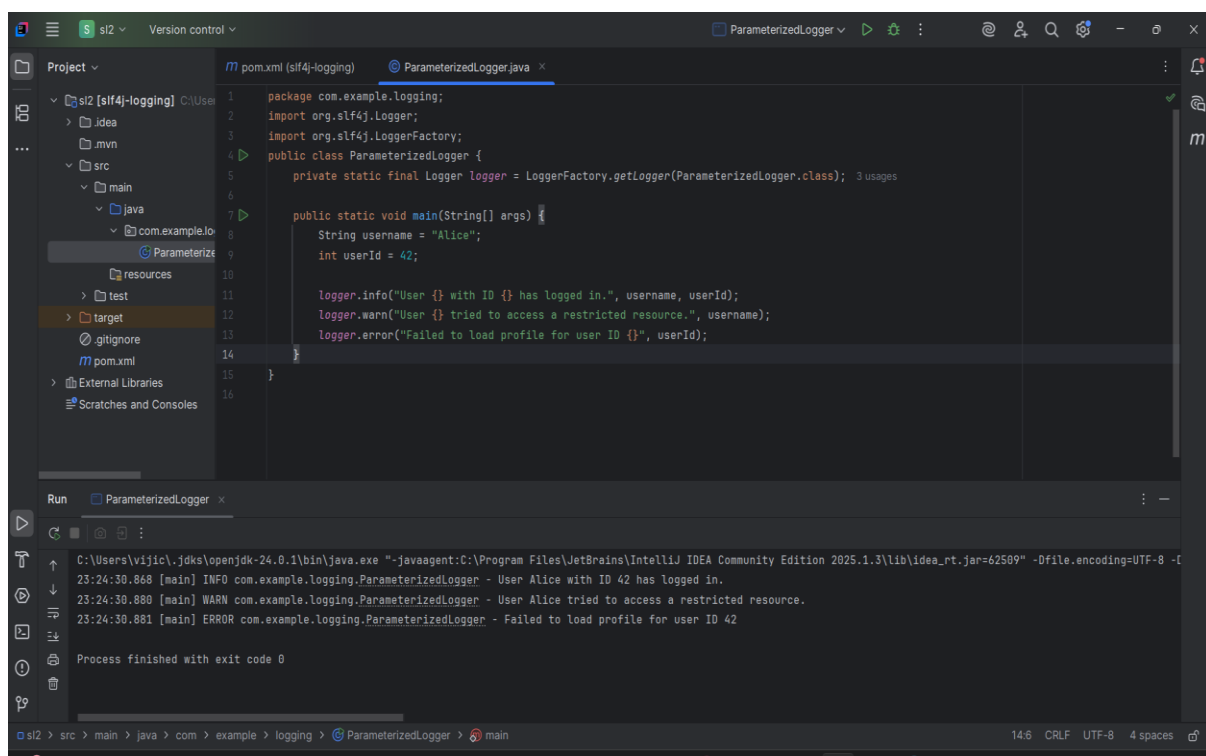
ParameterizedLogger.java:

```
package com.example.logging;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
public class ParameterizedLogger {
    private static final Logger logger =
        LoggerFactory.getLogger(ParameterizedLogger.class);

    public static void main(String[] args) {
        String username = "Alice";
        int userId = 42;

        logger.info("User {} with ID {} has logged in.", username, userId);
        logger.warn("User {} tried to access a restricted resource.", username);
        logger.error("Failed to load profile for user ID {}", userId);
    }
}
```

Output:



The screenshot shows the IntelliJ IDEA interface. The top pane displays the source code for `ParameterizedLogger.java`, which is identical to the code provided in the previous block. The bottom pane shows the output of the program execution. The output consists of three log messages: an INFO message, a WARN message, and an ERROR message, all from the `com.example.logging.ParameterizedLogger` class. The messages are: "User {} with ID {} has logged in.", "User {} tried to access a restricted resource.", and "Failed to load profile for user ID {}". The output also shows the process finished with exit code 0.

```
Run: ParameterizedLogger
C:\Users\vijic\jdk\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.3\lib\idea_rt.jar=62589" -Dfile.encoding=UTF-8 -I
23:24:30.868 [main] INFO com.example.logging.ParameterizedLogger - User Alice with ID 42 has logged in.
23:24:30.880 [main] WARN com.example.logging.ParameterizedLogger - User Alice tried to access a restricted resource.
23:24:30.881 [main] ERROR com.example.logging.ParameterizedLogger - Failed to load profile for user ID 42
Process finished with exit code 0
```

Exercise 3: Using Different Appenders

Pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>LoggingExample</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <maven.compiler.source>24</maven.compiler.source>
    <maven.compiler.target>24</maven.compiler.target>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
  <dependencies>
    <!-- SLF4J API -->
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-api</artifactId>
      <version>1.7.30</version>
    </dependency>
    <!-- Logback Classic (backend for SLF4J) -->
    <dependency>
      <groupId>ch.qos.logback</groupId>
      <artifactId>logback-classic</artifactId>
      <version>1.2.3</version>
    </dependency>
  </dependencies>

</project>
```

LoggerApp.java:

```
package com.example;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
public class LoggerApp {
    private static final Logger logger =
        LoggerFactory.getLogger(LoggerApp.class);

    public static void main(String[] args) {
        logger.info("Application started");
        logger.warn("This is a warning message");
        logger.error("An error occurred");
        logger.debug("Debugging application...");
    }
}
```

<logback.xml>:

configuration>

```
<!-- Console Appender -->
<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>

<!-- File Appender -->
<appender name="file" class="ch.qos.logback.core.FileAppender">
    <file>app.log</file>
    <append>true</append>
    <encoder>
        <pattern>%d{yyyy-MM-dd HH:mm:ss} [%thread] %-5level %logger{36} -
```

```
%msg%n</pattern>
```

```
</encoder>
```

```
</appender>
```

```
<!-- Root Logger -->
```

```
<root level="debug">
```

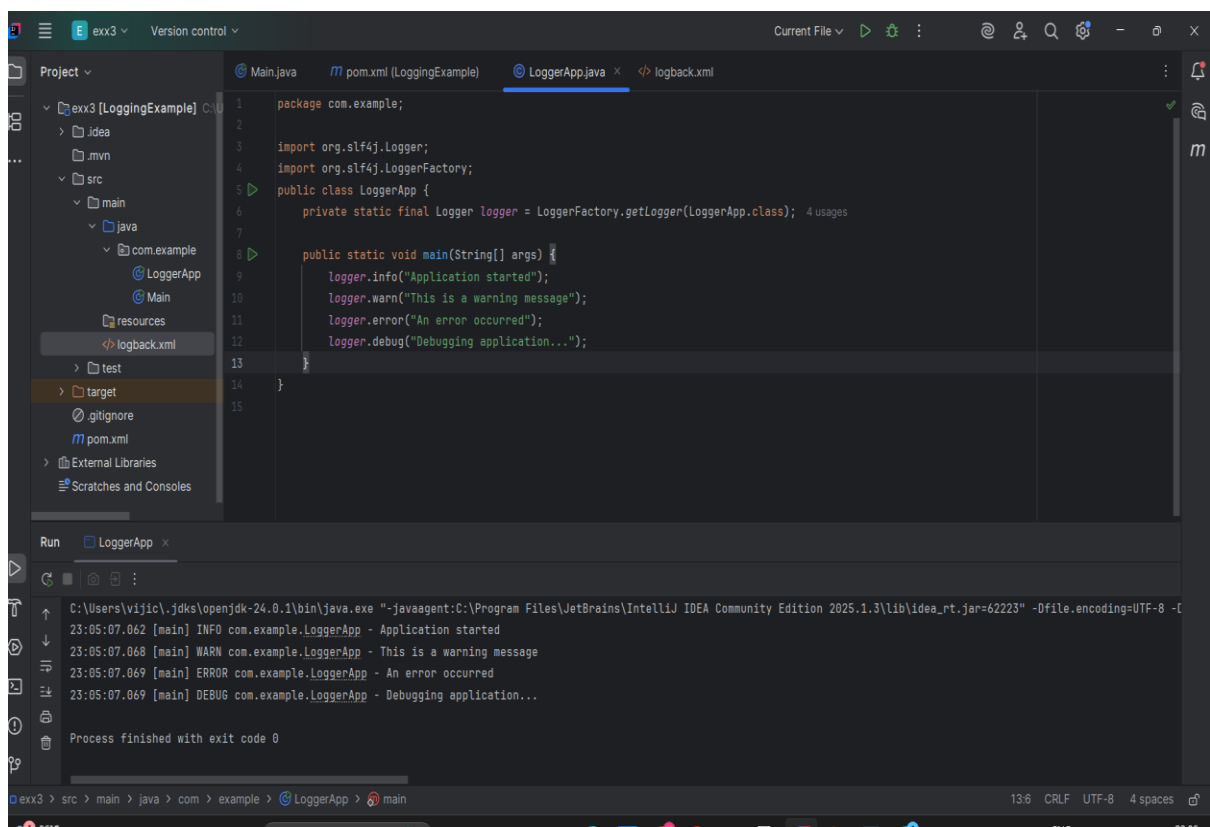
```
  <appender-ref ref="console" />
```

```
  <appender-ref ref="file" />
```

```
</root>
```

```
</configuration>
```

Output:



The screenshot shows an IDE window with the following components:

- Project Explorer:** Displays the project structure for 'exx3 [LoggingExample]'. The 'logback.xml' file is highlighted under the 'resources' folder.
- Editor:** Shows the 'LoggerApp.java' file with the following code:

```
1 package com.example;
2
3 import org.slf4j.Logger;
4 import org.slf4j.LoggerFactory;
5
6 public class LoggerApp {
7     private static final Logger logger = LoggerFactory.getLogger(LoggerApp.class);
8
9     public static void main(String[] args) {
10         logger.info("Application started");
11         logger.warn("This is a warning message");
12         logger.error("An error occurred");
13         logger.debug("Debugging application...");
14     }
15 }
```
- Run Console:** Displays the output of the application run. The command used is:

```
C:\Users\vijic\jdk\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.3\lib\idea_rt.jar=62223" -Dfile.encoding=UTF-8 -I
```

The output shows the following log messages:

```
23:05:07.062 [main] INFO com.example.LoggerApp - Application started
23:05:07.068 [main] WARN com.example.LoggerApp - This is a warning message
23:05:07.069 [main] ERROR com.example.LoggerApp - An error occurred
23:05:07.069 [main] DEBUG com.example.LoggerApp - Debugging application...
```

The console also indicates that the process finished with exit code 0.

