

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>
<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>
```

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");
    }
}
```

Exercise 2: Parameterized Logging

Task: Write a Java application that demonstrates parameterized logging using SLF4J.

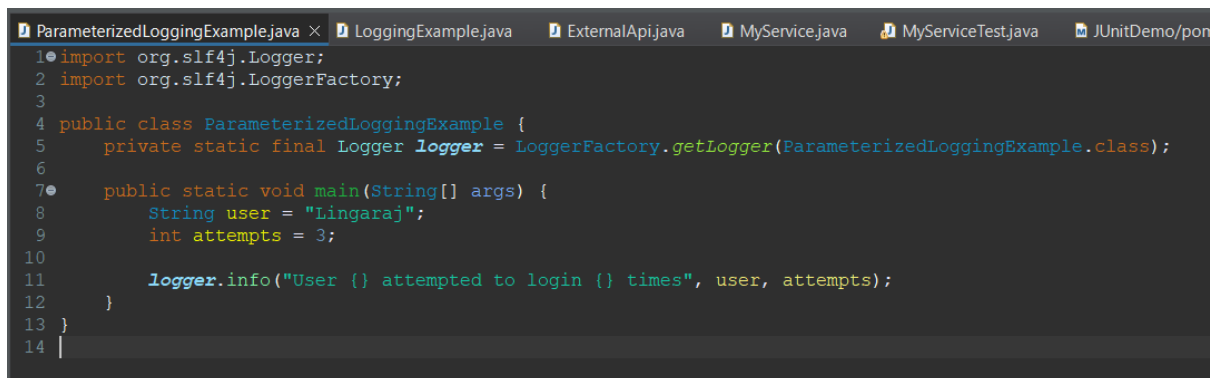
Step-by-Step Solution:

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

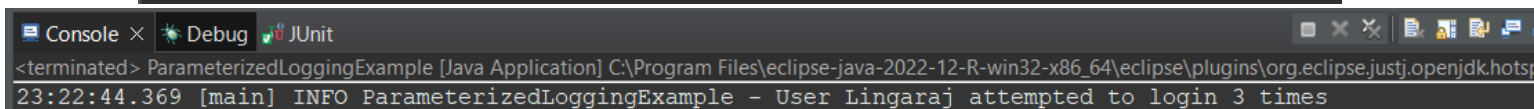
```
<dependency>
<groupId>org.slf4j</groupId>
<artifactId>slf4j-api</artifactId>
<version>1.7.30</version>
</dependency>
<dependency>
<groupId>ch.qos.logback</groupId>
<artifactId>logback-classic</artifactId>
<version>1.2.3</version>
</dependency>
```

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

Write code for this.



```
1 import org.slf4j.Logger;
2 import org.slf4j.LoggerFactory;
3
4 public class ParameterizedLoggingExample {
5     private static final Logger logger = LoggerFactory.getLogger(ParameterizedLoggingExample.class);
6
7     public static void main(String[] args) {
8         String user = "Lingaraj";
9         int attempts = 3;
10
11         logger.info("User {} attempted to login {} times", user, attempts);
12     }
13 }
14
```



```
<terminated> ParameterizedLoggingExample [Java Application] C:\Program Files\eclipse-java-2022-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe
23:22:44.369 [main] INFO ParameterizedLoggingExample - User Lingaraj attempted to login 3 times
```

Exercise 3: Using Different Appenders

Task: Write a Java application that demonstrates using different appenders with SLF4J.

Step-by-Step Solution:

1. Add SLF4J and Logback dependencies to your `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>
```

2. Create a `logback.xml` configuration file to define different appenders:

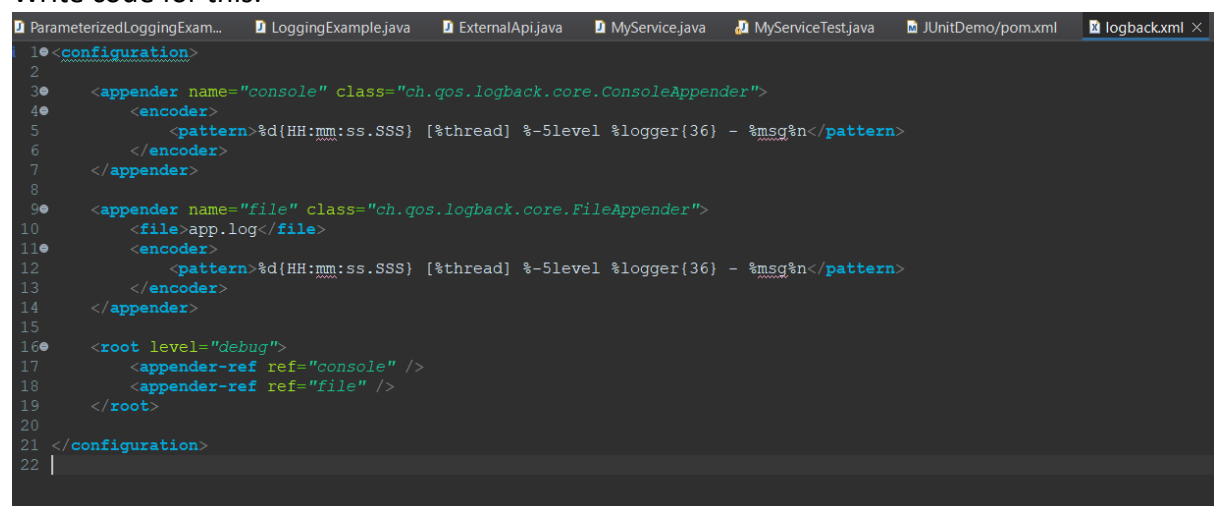
```

<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>app.log</file>
<encoder>
<pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
</encoder>
</appender>
<root level="debug">
<appender-ref ref="console" />
<appender-ref ref="file" />
</root>
</configuration>

```

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

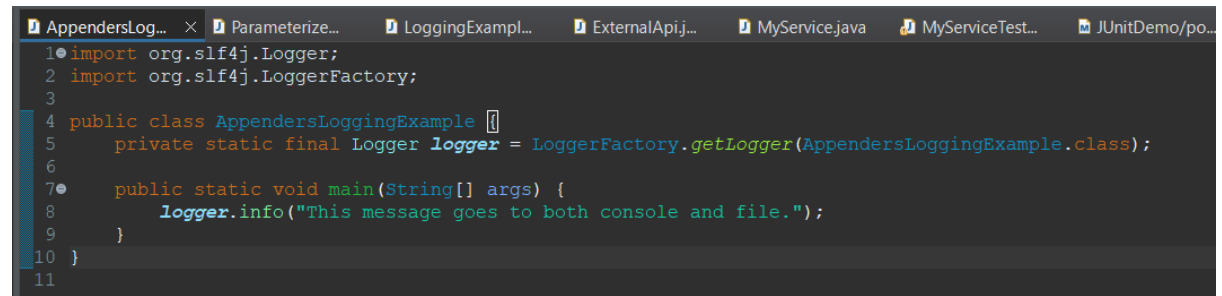
Write code for this.



```

1  <configuration>
2
3  <appender name="console" class="ch.qos.logback.core.ConsoleAppender">
4    <encoder>
5      <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
6    </encoder>
7  </appender>
8
9  <appender name="file" class="ch.qos.logback.core.FileAppender">
10    <file>app.log</file>
11    <encoder>
12      <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
13    </encoder>
14  </appender>
15
16  <root level="debug">
17    <appender-ref ref="console" />
18    <appender-ref ref="file" />
19  </root>
20
21 </configuration>
22

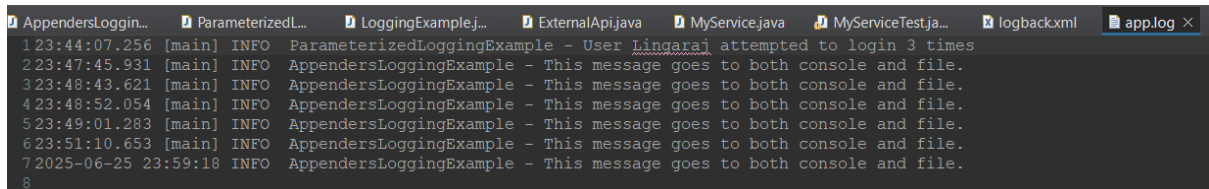
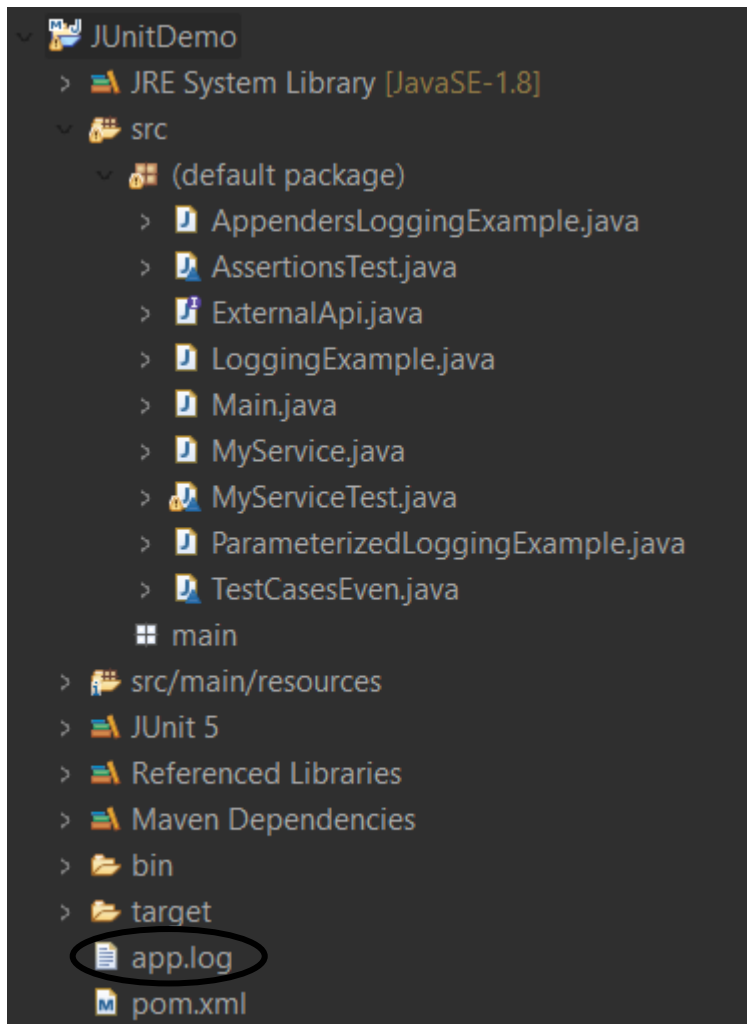
```



```

1  import org.slf4j.Logger;
2  import org.slf4j.LoggerFactory;
3
4  public class AppendersLoggingExample {
5    private static final Logger logger = LoggerFactory.getLogger(AppendersLoggingExample.class);
6
7    public static void main(String[] args) {
8      logger.info("This message goes to both console and file.");
9    }
10 }
11

```



Submitted By:

Name : Lingaraj Nayak

Superset ID : 6387607