

Logger - LogBack

1. First you have to add dependencies to maven.

```
<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/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.jalafoundation.example</groupId>
  <artifactId>logger</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>logger</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-api</artifactId>
      <version>1.7.21</version>
    </dependency>
    <dependency>
      <groupId>ch.qos.logback</groupId>
      <artifactId>logback-core</artifactId>
      <version>1.1.7</version>
    </dependency>
    <dependency>
      <groupId>ch.qos.logback</groupId>
      <artifactId>logback-classic</artifactId>
      <version>1.1.7</version>
    </dependency>
  </dependencies>
</project>
```

2. You have to create a logback.xml file to configure the logger.

```
<configuration>
  <appender name="STD-OUT" class="ch.qos.logback.core.ConsoleAppender">
    <layout class="ch.qos.logback.classic.PatternLayout">
      <Pattern>
        %d{yyyy-MM-dd HH:mm:ss} %-5level - %msg%n
      </Pattern>
    </layout>
  </appender>
  <appender name="FILE-OUT" class="ch.qos.logback.core.FileAppender">
    <file>./logs/example.log</file>
    <encoder class="ch.qos.logback.classic.encoder.PatternLayoutEncoder">
      <Pattern>
        %d{yyyy-MM-dd HH:mm:ss} %-5level - %msg%n
      </Pattern>
    </encoder>
  </appender>
  <appender name="HTML-OUT" class="ch.qos.logback.core.FileAppender">
    <encoder class="ch.qos.logback.core.encoder.LayoutWrappingEncoder">
      <layout class="ch.qos.logback.classic.html.HTMLLayout">
        <pattern>
          %d{yyyy-MM-dd HH:mm:ss} %-5level - %msg%n
        </pattern>
      </layout>
    </encoder>
    <file>./logs/test.html</file>
  </appender>

  <logger name="com.jalafoundation.example" level="info"
    additivity="false">
    <appender-ref ref="HTML-OUT" />
    <appender-ref ref="FILE-OUT" />
    <appender-ref ref="STD-OUT" />
  </logger>

  <root level="info">
    <appender-ref ref="STD-OUT" />
  </root>
</configuration>
```

3. You could create a Log Manager to keep it simple.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class LogManager {
  private static final Logger logger = LoggerFactory.getLogger(LogManager.class);
  private LogManager() {}

  public static void info(String text) {
    logger.info(text);
  }

  public static void warning(String text) {
    logger.warn(text);
  }

  public static void error(String text) {
    logger.error(text);
  }
}
```

4. To use this logger you don't need instantiate or declare anything, with the manager you can log everything.

```
public class App
{
  public static void log()
  {
    LogManager.info("This is a Info");
    LogManager.warning("This is a Warning");
    LogManager.error("This is an Error");
  }
}
```