Mr. RAGHU (Naresh I Technologies, Ameerpet)

SLF4J

The Simple Logging Facade for Java (SLF4J) serves as a simple facade or abstraction for various logging frameworks (e.g. java.util.logging, logback, log4j) allowing the end user to plug in the desired logging framework at *deployment* time.

SLF4J-Log4J

pom.xml

```
<properties>
		<maven.compiler.source>1.8</maven.compiler.source>
		<maven.compiler.target>1.8</maven.compiler.target>
	</properties>
	<dependencies>
		<dependency>
			<groupId>org.slf4j</groupId>
			<artifactId>slf4j-api</artifactId>
			<version>1.7.32</version>
		</dependency>
		<dependency>
			<groupId>org.slf4j</groupId>
			<artifactId>slf4j-log4j12</artifactId>
			<version>1.7.32</version>
		</dependency>
	</dependencies>
```

log4j.properties

```
# rootLogger=PriorityMethod, app names
log4j.rootLogger=TRACE, abc, xyz

# appender
log4j.appender.abc=org.apache.log4j.ConsoleAppender
log4j.appender.abc.Target=System.out
# layout
log4j.appender.abc.layout=org.apache.log4j.PatternLayout
log4j.appender.abc.layout.ConversionPattern=%d{dd/MM/yy} %C [%M] - %p : %m %n

# File Appender
log4j.appender.xyz=org.apache.log4j.FileAppender
log4j.appender.xyz.File=d:/mylogs/myapp.log
log4j.appender.xyz.layout=org.apache.log4j.PatternLayout
log4j.appender.xyz.layout.ConversionPattern=%d{dd/MM/yy} %C [%M] - %p : %m %n
```

Test.java

```
package in.nit;

import java.io.IOException;

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


public class Test {

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

	public static void main(String[] args) throws IOException {
		log.trace("FROM TRACE");
		log.debug("FROM DEBUG");
		log.info("FROM info");
		log.warn("FROM warn");
		log.error("FROM error");
	}
}
```

SLF4J-Logback

pom.xml

```
<properties>
		<maven.compiler.source>1.8</maven.compiler.source>
		<maven.compiler.target>1.8</maven.compiler.target>
	</properties>
	<dependencies>
		<dependency>
			<groupId>org.slf4j</groupId>
			<artifactId>slf4j-api</artifactId>
			<version>1.7.32</version>
		</dependency>
		<dependency>
			<groupId>ch.qos.logback</groupId>
			<artifactId>logback-classic</artifactId>
			<version>1.2.6</version>
		</dependency>
	</dependencies>
```

logback.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
```

```
    <appender name="stdout" class="ch.qos.logback.core.ConsoleAppender">
        <encoder>
          <pattern>%d{yyyy-MM-dd HH:mm:ss:SSS} %5p %t %c{2}:%L - %m%n</pattern>
        </encoder>
    </appender>
    <root level="TRACE">
        <appender-ref ref="stdout"/>
    </root>

    <appender name="FILE" class="ch.qos.logback.core.FileAppender">
      <file>testFile.log</file>
            <append>true</append>
      <immediateFlush>true</immediateFlush>
      <encoder>
            <pattern>%d{yyyy-MM-dd HH:mm:ss:SSS} %5p %t %c{2}:%L -
      %m%n</pattern>
      </encoder>
  </appender>

  <root level="TRACE">
    <appender-ref ref="FILE" />
  </root>
</configuration>
```

Test.java

```
package in.nit;

import java.io.IOException;

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


public class Test {

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

      public static void main(String[] args) throws IOException {
            log.trace("FROM TRACE");
            log.debug("FROM DEBUG");
            log.info("FROM info");
            log.warn("FROM warn");
            log.error("FROM error");
      }
}
```