在分布式系统中，各种无关日志穿行其中，导致我们可能无法直接定位整个操作流程。因此，我们可能需要对一个用户的操作流程进行归类标记，比如使用线程+时间戳，或者用户身份标识等；如此，我们可以从大量日志信息中grep出某个用户的操作流程，或者某个时间的流转记录。

因此，这就有了 Slf4j MDC 方法。

MDC ( Mapped Diagnostic Contexts )

其目的是为了便于我们诊断线上问题而出现的方法工具类。虽然，Slf4j 是用来适配其他的日志具体实现包的，但是针对 MDC功能，目前只有logback 以及 log4j 支持

public class MDC {

//Put a context value as identified by key //into the current thread's context map.

public static void put(String key, String val);

//Get the context identified by the key parameter.

public static String get(String key);

//Remove the context identified by the key parameter.

public static void remove(String key);

//Clear all entries in the MDC.

public static void clear(); }

看看一个MDC使用的简单示例。

public class LogTest {

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

public static void main(String[] args) {

MDC.put("THREAD\_ID", String.valueOf(Thread.currentThread().getId()));

logger.info("纯字符串信息的info级别日志");

} }

然后看看logback的输出模板配置：

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

**<configuration>**

**<property** name="log.base" value="${catalina.base}/logs" **/>**

**<contextListener** class="ch.qos.logback.classic.jul.LevelChangePropagator"**>**

**<resetJUL>**true**</resetJUL>**

**</contextListener>**

**<appender** name="console" class="ch.qos.logback.core.ConsoleAppender"**>**

**<encoder** charset="UTF-8"**>**

**<pattern>**[%d{yyyy-MM-dd HH:mm:ss} %highlight(%-5p) %logger.%M(%F:%L)] %X{THREAD\_ID} %msg%n**</pattern>**

**</encoder>**

**</appender>**

**<root** level="INFO"**>**

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

**</root>**

**</configuration>**

于是，就有了输出：

[2015-04-30 15:34:35 INFO io.github.ketao1989.log4j.LogTest.main(LogTest.java:29)] 1 纯字符串信息的info级别日志

在日志模板logback.xml 中，使用 %X{ }来占位，替换到对应的 MDC 中 key 的值。

refer: [http://www.ithao123.cn/content-8291525.html](%20http:/www.ithao123.cn/content-8291525.html)