

Slf4jReporter

该类同样是继承自 AbstractReporter 抽象类，实现了 Scheduled 接口。

在该类中有一个属性：

```
// 初始大小大致符合 150 个带有默认 scope 设置的指标  
private int previousSize = 16384;
```

在 report 方法中有一个 tryReport 方法用来发送数据。

```
private void tryReport() {  
    // initialize with previous size to avoid repeated resizing of  
backing array  
    // pad the size to allow deviations in the final string, for  
example due to different double value representations  
    StringBuilder builder = new StringBuilder((int) (previousSize *  
1.1));  
  
    builder  
        .append(lineSeparator)  
        .append("===== Starting metrics  
report =====")  
        .append(lineSeparator);  
  
    builder  
        .append(lineSeparator)  
        .append("-- Counters -----")  
        .append(lineSeparator);  
    for (Map.Entry<Counter, String> metric : counters.entrySet()) {  
        builder  
            .append(metric.getValue()).append(":"  
").append(metric.getKey().getCount())  
            .append(lineSeparator);  
    }  
  
    builder  
        .append(lineSeparator)  
        .append("-- Gauges -----")  
        .append(lineSeparator);  
}
```

```

        for (Map.Entry<Gauge<?>, String> metric : gauges.entrySet()) {
            builder
                .append(metric.getValue()).append(":")
        }.append(metric.getKey().getValue())
            .append(lineSeparator);
    }

    builder
        .append(lineSeparator)
        .append("-- Meters -----")
        .append(lineSeparator);
    for (Map.Entry<Meter, String> metric : meters.entrySet()) {
        builder
            .append(metric.getValue()).append(":")
        }.append(metric.getKey().getRate())
            .append(lineSeparator);
    }

    builder
        .append(lineSeparator)
        .append("-- Histograms -----")
        .append(lineSeparator);
    for (Map.Entry<Histogram, String> metric :
histograms.entrySet()) {
        HistogramStatistics stats =
metric.getKey().getStatistics();
        builder
            .append(metric.getValue()).append(":")
        count=").append(stats.size())
            .append(", min=").append(stats.getMin())
            .append(", max=").append(stats.getMax())
            .append(", mean=").append(stats.getMean())
            .append(", stddev=").append(stats.getStdDev())
            .append(", p50=").append(stats.getQuantile(0.50))
            .append(", p75=").append(stats.getQuantile(0.75))
            .append(", p95=").append(stats.getQuantile(0.95))
            .append(", p98=").append(stats.getQuantile(0.98))
            .append(", p99=").append(stats.getQuantile(0.99))
            .append(", p999=").append(stats.getQuantile(0.999))
            .append(lineSeparator);
    }

    builder
        .append(lineSeparator)
        .append("===== Finished metrics
report =====")

```

```
        .append(lineSeparator);
        LOG.info(builder.toString());

        previousSize = builder.length();
    }
}
```

要使用此 reporter，你必须复制 /opt/flink-metrics-slf4j-1.9.0.jar 到 /lib 文件夹中。

配置示例：

```
metrics.reporter.slf4j.class:
org.apache.flink.metrics.slf4j.Slf4jReporter
metrics.reporter.slf4j.interval: 60 SECONDS
```

<https://ci.apache.org/projects/flink/flink-docs-stable/monitoring/metrics.html#slf4j-orgapacheflinkmetricsslf4jslf4jreporter>