

这个项目的 maven 依赖有：

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-metrics-dropwizard</artifactId>
  <version>${project.version}</version>
</dependency>

<dependency>
  <groupId>io.dropwizard.metrics</groupId>
  <artifactId>metrics-core</artifactId>
  <version>${metrics.version}</version>
</dependency>

<dependency>
  <groupId>io.dropwizard.metrics</groupId>
  <artifactId>metrics-graphite</artifactId>
  <version>${metrics.version}</version>
</dependency>
```

GraphiteReporter

该类继承了 ScheduledDropwizardReporter 类，它的属性有一个：

```
// 表示协议
public static final String ARG_PROTOCOL = "protocol ";

// 有两种选择 TCP 和 UDP，默认是 TCP，如果没设置 protocol 的话
private enum Protocol {
    TCP,
    UDP
}
```

重写了 getReporter 方法，方法内部里面先获取 host 和 port：

```
String host = config.getString(ARG_HOST, null);
int port = config.getInteger(ARG_PORT, -1);

//如果 host 和 port 不符合要求的话抛出异常
if (host == null || host.length() == 0 || port < 1) {
    throw new IllegalArgumentException("Invalid host/port
configuration. Host: " + host + " Port: " + port);
}
```

从配置中获取 prefix、rateConversion、durationConversion、protocol 的值：

```
String prefix = config.getString(ARG_PREFIX, null);
String conversionRate = config.getString(ARG_CONVERSION_RATE,
null);
String conversionDuration =
config.getString(ARG_CONVERSION_DURATION, null);
String protocol = config.getString(ARG_PROTOCOL, "TCP");
```

通过 MetricRegistry 构造 GraphiteReporter 对象，并设置属性 prefix、conversionRate、conversionDuration

```
com.codahale.metrics.graphite.GraphiteReporter.Builder builder =
com.codahale.metrics.graphite.GraphiteReporter.forRegistry(registry
);

if (prefix != null) {
    builder.prefixedWith(prefix);
}

if (conversionRate != null) {
    builder.convertRatesTo(TimeUnit.valueOf(conversionRate));
}

if (conversionDuration != null) {
    builder.convertDurationsTo(TimeUnit.valueOf(conversionDuration));
}
```

下面这段代码是设置协议：

```

Protocol prot;
try {
    prot = Protocol.valueOf(protocol);
} catch (IllegalArgumentException iae) {
    log.warn("Invalid protocol configuration: " + protocol + "
Expected: TCP or UDP, defaulting to TCP.");
    prot = Protocol.TCP;
}

log.info("Configured GraphiteReporter with {host:{}, port:{},
protocol:{}}", host, port, prot);
switch(prot) {
    case UDP:
        return builder.build(new GraphiteUDP(host, port));
    case TCP:
    default:
        return builder.build(new Graphite(host, port));
}

```

如果是 UDP 的话，则通过 host 和 port 构造 GraphiteUDP 对象，如果是 TCP 或者默认就是构造 Graphite 对象。

注意

要使用此 reporter，你必须复制 /opt/flink-metrics-graphite-1.9.jar 到 Flink 的 /lib 文件夹中

参数：

- host: Graphite服务器主机
- port: Graphite服务器端口
- protocol: 使用协议（TCP / UDP）

配置示例：

```

metrics.reporter.grph.class:
org.apache.flink.metrics.graphite.GraphiteReporter
metrics.reporter.grph.host: localhost
metrics.reporter.grph.port: 2003
metrics.reporter.grph.protocol: TCP

```

