

```

    2: required binary body,
  }

  enum Status {
    OK,
    FAILED,
    ERROR,
    UNKNOWN
  }

  service ThriftSourceProtocol {
    Status append(1: ThriftFlumeEvent event),
    Status appendBatch(1: list<ThriftFlumeEvent> events),
  }

```

IDL 可以用来以 Thrift 支持的任何语言生成 Thrift 客户端。生成的代码可以用来将数据发送到 Flume 的 Thrift Source。

Thrift Source 的配置非常简单，类似于 Avro Source（见表 3-4）。

41 表3-4 Thrift Source配置

配置参数	默认值	描述
type	-	Thrift Source 的别名是 Thrift。也可以使用 FQCN, <code>org.apache.flume.source.ThriftSource</code> （大小写敏感）
bind	-	绑定的 IP 地址 / 主机名。使用 0.0.0.0 绑定机器所有的接口
port	-	绑定的端口号
threads	-	Source 处理请求使用的最大工作线程的数量

bind 参数指明要绑定的接口的 hostname 或 IP 地址；使用 0.0.0.0 绑定所有的接口。**port** 参数指明使用的端口号——客户端将使用这个端口发送事件给 Source。这些是必须配置参数。

Thrift Source 的 **threads** 参数与 Avro Source 相比，工作方式稍有不同。Flume（自 1.4.0 版本起）构建且包含 0.7.0 版本的 Thrift。这意味着支持客户端（用 Flume SDK 写的程序）以相同的程序写入到 0.92 版本的 HBase（或更早的版本）。如果对该 HBase 版本的支持没有要求，那么建议 Flume 附带的 Thrift 版本替换为更新的版本，尽管由于 Thrift-generated 代码不兼容，Flume 可能也需要对新版本进行重新编译。

当使用低于 0.8.0 版本的 Thrift，Flume 使用 Thrift 的 `TThreadPoolServer`，每个客户端连接使用一个线程，**threads** 参数控制 Source 创建的线程的最大数，从而间接地控制客