

```

        "Request is not in expected format. " +
        "Please refer documentation for expected format.", ex);
    }
    return events;
}

@Override
public void configure(Context context) {
    insertTimestamp = context.getBoolean(CONF_INSERT_TIMESTAMP,
        false);
}
}

```

例 3-3 展示了 XML 处理程序的配置示例。配置指示了处理程序为每个事件插入处理时的时间戳。通常，任意数量的参数都能以这种方式传递给 HTTP Source 处理程序。

HTTPHandler 接口是 flume-ng-core Maven 组件的一部分，可以在你的应用中的 *pom.xml* 的 *dependency* 部分添加例 3-6。

例3-6 在你的应用中包含以下的flume-ng-core组件

```

<dependency>
  <groupId>org.apache.flume</groupId>
  <artifactId>flume-ng-core</artifactId>
  <version>1.5.0</version>
</dependency>

```

使用第 8 章的“部署自定义代码”一节里展示的 *pluins.d* 框架给 Flume Agent 部署自定义的 HTTP 处理程序。

Spooling Directory Source

50

在很多场景中，应用程序生成被写入文件的数据。通常这些文件不是简单的文本文件，或者每一行转换为一个事件没有意义，而是若干行的组共同组成一个事件。一个例子就是堆栈跟踪。这通常很难或不可能修改这些应用，来使用 Flume Client API 直接发送数据给 Flume。在这种情况下，Flume 的 Spooling Directory Source 可以被用来从这些文件中读取数据。

Spooling Directory Source 监视读取事件的目录。尽管新文件可以被实时地添加到该目录，Source 还是期望目录中的文件是不变的。文件一旦被移入到该目录，它不应该被写入。如果你在处理日志文件，这样做的一个好方法是配置日志系统当日志滚动时移动文件。Source 也期望文件名是不重名的。如果这两种情况发生一种，Source 将会抛出异常并终止。这时重启 Source 的唯一方式是重启 Agent 自身。