常。如果传入的数据是不规范的并且不能被转换为 Flume 事件，HTTP Source 期望处理程序抛出 HTTPBadRequestException 异常。这个操作必须是幂等的，并且处理程序每次针对相同的输入必须抛出相同的异常。如果处理程序抛出 HTTPBadRequestException 异常，HTTP Source 将返回 HTTP 错误代码 400 来通知客户端这个请求是不规范的。如果处理程序抛出其他的异常，Source 将返回 HTTP 错误代码 500 来通知客户端 HTTP Source 有内部错误。然后由客户来决定在这种情况下如何重试。如果其中一个 Source 正在写入的管道抛出一个管道异常，Source 将返回错误代码 503，以提示管道暂时满负荷，客户端应该稍后重试。

JAR 文件包含的处理程序(或处理程序的 *.class* 文件)和所有的依赖，应该通过第 8 章的“部署自定义代码”一节中讨论的 plugins.d 机制将它们添加到 Flume 的环境变量中。

如果配置中没有指定处理程序，HTTP Source 将使用与 Flume 绑定的处理程序，它能处理 JSON 格式的事件。每个事件可以包含包装为数组的几个事件，尽管 Source 写入的管道可能有限制的事务能力。处理程序接受 UTF-8、UTF-16 或 UTF-32 编码的 JSON 格式的数据，并且将它转换成一个列表的事件。事件的正文是原始 HTTP 请求序列化的字符集。处理程序接受的格式如下：

46

```
[{
    "headers" : {
       "event1Header1" : "event1Value1",
       "event1header2" : "event1Value2"
    },
    "body" : "This is the body of the first event."

},
{
    "headers" : {
       "event2Header1" : "event2Value1",
       "event2Header2" : "event2Value2"
    },
    "body" : "This is the body of the second event"
}]
```

在例 3-5 里展示的 HTTPSourceXMLHandler 是对 HTTP Source 适用的另一个处理程序的例子。处理程序将 XML 格式的数据转换为 Flume 事件。这类处理程序很简单，接受例 3-4 里展示的 XML 格式的数据。处理程序期望的格式很简单，只有处于 <events> 和 </events> 之间的数据才被处理。每个事件被期望处于 <events> 和 </events> 标签之间。唯一限制每个请求事件的数量的是 Source 写入的管道的事务处理能力。每个事件可以有一或多个节段，位于 <headers> 和 </headers> 标签间。每个 header 被用作 header 名字