

```

@Override
public void configure(Context context) {
    writeHeaders = context.getBoolean(CONFIG_WRITE_HEADERS,
        DEFAULT_WRITE_HEADERS);
}

@Override
public void configure(
    ComponentConfiguration componentConfiguration) {

    @SuppressWarnings("unchecked")
    @Override
    protected void prepareIndexRequest(
        IndexRequestBuilder indexRequestBuilder,
        String indexName, String indexType, Event event)
        throws IOException {
        indexRequestBuilder.setIndex(indexName).setType(indexType);
        if (writeHeaders) {
            Map source = (Map) event.getHeaders();
            source.put(BODY_HEADER,
                new String(event.getBody(), Charsets.UTF_8));
            indexRequestBuilder.setSource((Map<String, Object>) source);
        } else {
            indexRequestBuilder.setSource(event.getBody());
        }
    }
}
}

```

133

例 5-9 展示了配置的序列化器的一个例子。

这里展示的所有序列化器基本类别都是 `flume-ng-elasticsearch-sink` 工件的一部分。为了在你创建的序列化器中包含它，添加以下内容到 `pom.xml` 文件的依赖部分。

```

<dependency>
    <groupId>org.apache.flume.flume-ng-sinks</groupId>
    <artifactId>flume-ng-elasticsearch-sink</artifactId>
    <version>1.5.0</version>
</dependency>

```



Elastic Search Client API

因为 Elastic Search 需要 Flume 使用完全相同 Elastic Search 版本作为集群，Flume 没有打包 Elastic Search 客户端库。用户必须在 Agent 的环境变量中部署库和它们的所有依赖，以使得 Sink 能够写数据。