

ElasticSearchEventSerializer。如果这个参数没有指定值，事件会被序列化为 Kibana 友好型格式。使用 `serializer` 前缀可以传递任何参数给序列化器。例 5-10 展示了 `ElasticSearchIndexRequestBuilderFactory` 接口。

例5-10 ElasticSearchIndexRequestBuilderFactory接口

```
package org.apache.flume.sink.elasticsearch;

public interface ElasticSearchIndexRequestBuilderFactory extends Configurable,
    ConfigurableComponent {
    IndexRequestBuilder createIndexRequest(Client client,
        String indexPrefix, String indexType, Event event) throws IOException;
}
```

为了更简单地实现这个接口，Flume 提供了一个抽象类，该类提供了一个简单的实现，`AbstractElasticSearchIndexRequestBuilderFactory`，能像前面一节介绍的那样每天创建一个新的索引。继承自该类的序列化器将每天自动创建一个新的索引（当然是在该类没有覆盖 `createIndexRequest` 方法来改变行为的情况下）。

为了使用自动创建索引的功能，而且也能以某些自定义的格式插入数据，那么继承自该类的一个类的 FQCN 应该设置为 `serializer` 参数的值。为了定制插入到索引中的数据，需要实现 `prepareIndexRequest` 方法。例 5-11 可以用来将事件插入到 Elastic Search 索引。
如果配置了写报头，那么报头插入它们各自的键和体（这个序列化器假设报头没有 body 键）。如果没有配置写 header，那就只是简单地写入体。

例5-11 IndexRequestBuilderFactory实现的一个例子

```
package usingflume.ch05;

public class HeaderAndBodyIndexRequestBuilderFactory extends
    AbstractElasticSearchIndexRequestBuilderFactory {

    private String CONFIG_WRITE_HEADERS = "writeHeaders";
    // By default, don't write the headers.
    private boolean DEFAULT_WRITE_HEADERS = false;
    private boolean writeHeaders = false;
    private static final String BODY_HEADER = "body";

    public HeaderAndBodyIndexRequestBuilderFactory() {
        this(FastDateFormat.getDateInstance(FastDateFormat.FULL));
    }

    protected HeaderAndBodyIndexRequestBuilderFactory(
        FastDateFormat dateFormat) {
        super(dateFormat);
    }
}
```