

# 17. Ingest Node

作者: [felayman](#)

原文链接: <https://ld246.com/article/1512830107517>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<blockquote>

<p>并非完全翻译,会穿插相关的 Java API 来来辅助进行说明(但基本会覆盖官方所有翻译)</p>

</blockquote>

<ul>

<li>

<p>原文地址: <a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.elastic.co%2Fguide%2Fen%2Felasticsearch%2Freference%2Fcurrent%2Fingest-processors.html" target="\_blank" rel="nofollow ugc">Processors</a> </p>

</li>

<li>

<p>版本:<a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.elastic.co%2Fguide%2Fen%2Felasticsearch%2Freference%2Fcurrent%2Findex.html" target="\_blank" rel="nofollow ugc">6.0.0</a> </p>

</li>

</ul>

<p><strong>说明</strong> </p>

<p>本文是建立在有一些 Elasticsearch 基础和了解相关 Pipeline 概念的人</p>

<h2 id="Ingest-Node">Ingest Node</h2>

<h3 id="简介Ingest-Node">简介 Ingest Node</h3>

<p>Ingest Node(预处理节点)是 ES 用于功能上命名的一种节点类型,可以通过在 elasticsearch.xml 行如下配置来标识出集群中的某个节点是否是 Ingest Node.</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">node.ingest: false
```

```
</span></span></code></pre>
```

<p>上述将 node.ingest 设置成 false,则表明当前节点不是 Ingest Node,不具有预处理能力,当然 Elasticsearch 默认所有节点都是 Ingest Node,即集群中所有的节点都具有预处理能力.</p>

<p><strong>何为预处理呢?</strong> </p>

<p>用过 Logstash 对日志进行处理的用户都知道,一般情况下我们并不会直接将原始的日志进行加载 Elasticsearch 集群,而是对原始日志信息进行(深)加工后保存到 Elasticsearch 集群中.比如 Logstash 持多种解析器比如 json,kv,date 等,比较经典的是 grok.这里我们不会对 Logstash 的解析器进行详细明,只是为了描述一个问题,有些时候我们需要 Logstash 来对加载到 Elasticsearch 中的数据进行处理,个处理,从概念上而言,我们也能称之为"预处理.而这里我们所说的预处理也其实就是类似的概念.</p>

<p>可以这么说,在 Elasticsearch 没有提供 IngestNode 这一概念时,我们想对存储在 Elasticsearch 的数据在存储之前进行加工处理的话,我们只能依赖 Logstash 或自定义插件来完成这一功能,但是在 Elasticsearch 5.x 版本中,官方在内部集成了部分 Logstash 的功能,这就是 Ingest,而具有 Ingest 能力的点称之为 Ingest Node.</p>

<p>请查看 <a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.elastic.co%2Fguide%2Fen%2Flogstash%2Fcurrent%2Ffilter-plugins.html" target="\_blank" rel="nofollow ugc">Filter plugins</a> 来了解更多关于 Logstash 解析器的内容.</p>

<p>如果要脱离 Logstash 来对在 Elasticsearch 写入文档之前对文档进行加工处理,比如为文档某个段设置默认值,重命名某个字段,设置通过脚本来完成更加复杂的加工逻辑,我们则必须要了解两个基本念: Pipeline 和 Processors.</p>

<p>参考: <a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.felayan.com%2Farticles%2F2017%2F11%2F24%2F1511515806564.html" target="\_blank" rel="nofollow ugc">(译) Ingest Node (预处理节点)</a> 或 <a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.elastic.co%2Fguide%2Fen%2Felasticsearch%2Freference%2Fcurrent%2Fingest.html" target="\_blank" rel="nofollow ugc">Ingest Node</a> 来了解更多关于上述两个概念.下文只是单说明两者.</p>

<h3 id="简介Pipeline-Processors">简介 Pipeline、Processors</h3>

<p>管道(Pipeline)是众所周知的一个概念,Elasticsearch 引入这一概念,是为了让那些有过工作经验人来说更直白,更轻松的理解这一概念.本人是主要用 Java 进行开发,这里就以 Pipeline 和 java 中的 Stream 进行类比,两者从功能和概念上很类似,我们经常会对 Stream 中的数据进行处理,比如 map 操作,p ek 操作,reduce 操作,count 操作等,这些操作从行为上说,就是对数据的加工,而 Pipeline 也是如此,Pipeline 也会对通过该 Pipeline 的数据(一般来说是文档)进行加工,比如上面说到的,修改文档的某个字段值

修改文档某个字段的类型等等.而 Elasticsearch 对该加工行为进行抽象包装,并称之为 Processors.Elasticsearch 命名了多种类型的 Processors 来规范对文档的操作,比如 set,append,date,join,json,kv 等等.这些不同类型的 Processors,我们会在后文进行说明

### Pipeline 定义

定义一个 Pipeline 是件很简单的事情,官方给出了参考:

```
PUT _ingest/pipeline/my-pipeline-id
{
  "description": "describe pipeline",
  "processors": [
    {
      "set": {
        "field": "foo",
        "value": "bar"
      }
    }
  ]
}
```

上面的例子,表明通过指定的 URL 请求 "\_ingest/pipeline" 定义了一个 ID 为 "my-pipeline-id" 的 pipeline,其中请求体中的存在两个必须的元素:

- 

- description 描述该 pipeline 是做什么的

- processors 定义了一系列的 processors,这里只是简单的定义了一个赋值操作,即将字段名为 "foo" 的字段值都设置为 "bar"



如果需要了解更多关于 Pipeline 定义的信息,可以参考: [Ingest APIs](https://ld246.com/forward?oto=https%3A%2F%2Fwww.elastic.co%2Fguide%2Fen%2Felasticsearch%2Freference%2Fcurrent%2Fput-pipeline-api.html)

### 简介 Simulate Pipeline API

既然 Elasticsearch 提供了预处理的能力,总不能是黑盒处理吧,为了让开发者更好的了解和使用预理的方式和原理,官方也提供了相关的接口,来让我们对这些预处理操作进行测试,这些接口,官方称之为: Simulate Pipeline API.想要深入了解 pipeline 以及 processors 的使用,我们基本离不开 Simulate Pipeline API 来辅助帮我们完成许多工作.

下面是一个比较简单的 Simulate Pipeline API 的例子:

```
POST _ingest/pipeline/_simulate
{
  "pipeline": "pipeline",
  "processors": [
    // pipeline definition here
  ]
}
```

```
docs</span><span class="highlight-err">"</span><span class="highlight-w"></span><sp
n class="highlight-o">:</span><span class="highlight-w"></span><span class="highlight-
">[</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w"></span><span class="highlight-p">{</span><span class="highlight-w">
</span><span class="highlight-err">"</span><span class="highlight-n">_source</span><sp
an class="highlight-err">"</span><span class="highlight-o">:</span><span class="highli
ht-w"></span><span class="highlight-p">}</span><span class="highlight-cm">/** first do
ument */</span><span class="highlight-p">}</span><span class="highlight-w"></span>
span class="highlight-p">},</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w"></span><span class="highlight-p">{</span><span class="highlight-w">
</span><span class="highlight-err">"</span><span class="highlight-n">_source</span><sp
an class="highlight-err">"</span><span class="highlight-o">:</span><span class="highli
ht-w"></span><span class="highlight-p">}</span><span class="highlight-cm">/** second
document */</span><span class="highlight-p">}</span><span class="highlight-w"></sp
n><span class="highlight-p">},</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w"></span><span class="highlight-c1">{// ...
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-c1"></span><span class="highlight-w"></span><span class="highlight-p">]
/span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w"></span><span class="highlight-p">}</span><span class="highlight-w">
</span></span></span></code></pre>
```

上面的例子中,在请求 URL 中并没有明确指定使用哪个 pipeline,这种情况下需要我们在请求体中定义一个,当然我们也可以指定一个 pipeline 的 ID,如下面例子:

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">POST _ingest/pipeline/my-pipeline-id/_simulate
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs" : [
</span></span><span class="highlight-line"><span class="highlight-cl">  { "_source": {/**
irst document */} },
</span></span><span class="highlight-line"><span class="highlight-cl">  { "_source": {/**
econd document */} },
</span></span><span class="highlight-line"><span class="highlight-cl">  // ...
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>
```

这样一来,在请求体中我们定义的"docs"中的内容就能够使用该 pipeline 和该 pipeline 种的 processors,下面是一个比较具体的例子:

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">POST _ingest/pipeline/_simulate
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "pipeline" :
</span></span><span class="highlight-line"><span class="highlight-cl">  {
</span></span><span class="highlight-line"><span class="highlight-cl">    "description": "
description",
</span></span><span class="highlight-line"><span class="highlight-cl">    "processors": [
</span></span><span class="highlight-line"><span class="highlight-cl">      {
</span></span><span class="highlight-line"><span class="highlight-cl">        "set" : {
</span></span><span class="highlight-line"><span class="highlight-cl">          "field" : "fiel
2",
</span></span><span class="highlight-line"><span class="highlight-cl">          "value" : "_v
```

```

lue"
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">},
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "_index": "inde
",
',
</span></span><span class="highlight-line"><span class="highlight-cl">      "_type": "type"

</span></span><span class="highlight-line"><span class="highlight-cl">      "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "foo": "bar"
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    },
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "_index": "inde
",
',
</span></span><span class="highlight-line"><span class="highlight-cl">      "_type": "type"

</span></span><span class="highlight-line"><span class="highlight-cl">      "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "foo": "rab"
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">>]
</span></span><span class="highlight-line"><span class="highlight-cl">>]
</span></span></code></pre>
<p>在上面具体的例子中,我们并没有在请求 URL 中指定使用哪个 pipeline,因此我们不得不在请求体
即时定义一个 pipeline 和对应的 processors,例子很简单,只是让通过该 pipeline 的的文档中的字段
称为"field2"的字段值为 "_value",文档我们也指定了,就是"docs"中定义的文档,该请求的响应信息如下:
</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "doc": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_index": "
ndex",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_type": "t
pe",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "field2":
_value",
</span></span><span class="highlight-line"><span class="highlight-cl">          "foo": "b
r"
</span></span><span class="highlight-line"><span class="highlight-cl">        },
</span></span><span class="highlight-line"><span class="highlight-cl">        "_ingest": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "timest
mp": "2017-05-04T22:30:03.187Z"
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span></code></pre>

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">    },
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "doc": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_index": "
index",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_type": "t
pe",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "field2":
_value",
</span></span><span class="highlight-line"><span class="highlight-cl">          "foo": "r
b"
</span></span><span class="highlight-line"><span class="highlight-cl">      },
</span></span><span class="highlight-line"><span class="highlight-cl">      "_ingest": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "timest
mp": "2017-05-04T22:30:03.188Z"
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl">> }
</span></span><span class="highlight-line"><span class="highlight-cl">> }
</span></span></code></pre>

```

从具体的响应结果中看到,在文档通过 pipeline 时(或理解为被 pipeline 中的 processors 加工后)新的文档与原有的文档产生了差异,这些差异体现为:

<ol>

<li>文档都新增了 field2 字段,这点我们可以通过对比响应前后的定义在"docs"中文档的\_source 内中看出</li>

<li>额外增加了一些临时(官方称之为瞬态)的字段,比如 timestamp,他们都在\_ingest 节点下,(这些)字都是临时与源文档存在一起,在被 pipeline 中的 processors 加工后后返回给对应的批量操作或索引作之后,这些信息就不会携带返回.</li>

</ol>

<h3 id="访问Pipeline中的内容">访问 Pipeline 中的内容</h3>

如果我们只是定义一个 Pipeline 而不能访问该 Pipeline 的上下文信息,那么设计 Pipeline 就显得是多此一举.与 logstash 不同的时候,logstash 环境与 Elasticsearch 环境是隔离的,两者无法形成上下环境,而 Pipeline 则不同,天然的集成在 Elasticsearch 中,从而很好的与其形成上线文环境,这个环境让 Pipeline 能够更充分的利用起来,比如与 Pipeline 能够形成最直接的上线文关系的是文档信息,由此可见,们可以在 Pipeline(应该是 processors 中)能够直接访问通过 pipeline 的文档信息,如文档的字段,元数据信息.</p>

<p>下面的例子说明了我们如何在 processors 中访问这些上下文信息.</p>

<p><strong>访问源文档字段</strong></p>

```

<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "set": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "field": "my_fiel
"

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">    "value": 582.1

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">  }

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">> }

```

```

</span></span></code></pre>

```

这个例子直接引用了通过该 Pipeline 内的文档字为"my\_field"的字段,并将所有文档该字段的值置为 582.1.</p>

或者我们也可以通过\_source 字段来访问源文档的字段,如下</p>

```

<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">{

```

```

cl">{
</span></span><span class="highlight-line"><span class="highlight-cl"> "set": {
</span></span><span class="highlight-line"><span class="highlight-cl"> "field": "_source
my_field"
</span></span><span class="highlight-line"><span class="highlight-cl"> "value": 582.1
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>
<p><strong>访问文档的元数据字段</strong></p>
<p>每个文档都会有一些元数据字段信息(metadata filed),比如_id, _index,_type 等,我们在 process
rs 中也可以直接访问这些信息的,比如下面的例子:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl"> "set": {
</span></span><span class="highlight-line"><span class="highlight-cl"> "field": " id"
</span></span><span class="highlight-line"><span class="highlight-cl"> "value": "1"
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>
<p>直接访问文档的_id 字段,并设置该值为 1</p>
<p><strong>访问瞬态元字段(ingest metadata field)</strong></p>
<p>这个地方并没有直接翻译官方名称,因为不好理解,我使用了我自己的理解意图去翻译该名称.其实
义是一致的,官方称之为ingest metadata field,翻译过来且为"摄取元数据字段?"(请高人翻译,我实在
法很好的翻译出来),从官方对该词的释义中可以理解出来,这些字段是临时保存的,在这些文档被处理完
之后返回给对应的批量请求和索引请求的时候,这些字段不会一并返回,因此这里称之为"瞬态元字段",
面是一个如何访问这些字段的例子:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl"> "set": {
</span></span><span class="highlight-line"><span class="highlight-cl"> "field": "receive
"
</span></span><span class="highlight-line"><span class="highlight-cl"> "value": "{{_inge
t.timestamp}}"
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>
<p>该例子会在文档中临时加入一个名称为"received"的字段,值为瞬态元字段中的 timestamp,以{{_i
gest.timestamp}}的格式来进行访问,看到{{}}是不是有种很熟悉的感觉,因为只有在涉及上下文的环境
一般才会使用这种表达式.</p>
<h3 id="Processors类型详解">Processors 类型详解</h3>
<ul>
<li>Append Processor 追加处理器</li>
<li>Convert Processor 转换处理器</li>
<li>Date Processor 日期转换器</li>
<li>Date Index Name Processor 日期索引名称处理器</li>
<li>Fail Processor 失败处理器</li>
<li>Foreach Processor 循环处理器</li>
<li>Grok Processor Grok 处理器</li>
<li>Gsub Processor</li>
<li>Join Processor</li>
<li>JSON Processor</li>
<li>KV Processor</li>
<li>Lowercase Processor</li>
<li>Remove Processor</li>

```

- </li>Rename Processor</li>
- </li>Script Processor</li>
- </li>Split Processor</li>
- </li>Sort Processor</li>
- </li>Trim Processor</li>
- </li>Uppercase Processor</li>
- </li>Dot Expander Processor</li>

#### Append Processor

顾名思义,追加处理器.就是当我们加载原始文档到 Elasticsearch 的时候,某些字段的描述信息需要我们定制性的增加一些额外说明.

使用方法如下:

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "append": {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">   "field": "field1",
</span> </span> <span class="highlight-line"> <span class="highlight-cl">   "value": ["item2
, "item3", "item4"]
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> </code> </pre>
```

比如我们需要从第三方新导入一批商品,因为某种原因这批新商品必须要打上"家居"的标签来示该批商品是"家居分类",当然一个商品也可能不止一个分类,这个时候要求我们在导入这批文档数据的时候,需要在原有的标签字段中新增一个"家居"标签,这个时候我们就通过追加处理器来完成.

下面的例子我们通过 Simulate Pipeline API 来帮助完成

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> POST _ingest/pipeline/_simulate
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">   "pipeline": {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     "description":
新增一个家居标签",
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     "processors": [
</span> </span> <span class="highlight-line"> <span class="highlight-cl">       {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">         "append": {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">           "field": "tag"
</span> </span> <span class="highlight-line"> <span class="highlight-cl">           "value": [
</span> </span> <span class="highlight-line"> <span class="highlight-cl">             "家居"
</span> </span> <span class="highlight-line"> <span class="highlight-cl">           ]
</span> </span> <span class="highlight-line"> <span class="highlight-cl">         }
</span> </span> <span class="highlight-line"> <span class="highlight-cl">       }
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     ],
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     "docs": [
</span> </span> <span class="highlight-line"> <span class="highlight-cl">       {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">         "_index": "inde
",
</span> </span> <span class="highlight-line"> <span class="highlight-cl">         "_type": "type"
</span> </span> <span class="highlight-line"> <span class="highlight-cl">         "_id": "id",
</span> </span> <span class="highlight-line"> <span class="highlight-cl">         "_source": {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">           "code": "0000
1",
</span> </span> <span class="highlight-line"> <span class="highlight-cl">         "tag": "衣柜"
</span> </span> </code> </pre>
```





```

</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
ex",
</span></span><span class="highlight-line"><span class="highlight-cl">
",
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
002",
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
"2017-11-24T13:19:55.555Z"
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
ex",
</span></span><span class="highlight-line"><span class="highlight-cl">
",
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
003",
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
"2017-11-24T13:19:55.555Z"
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
]
</span></span><span class="highlight-line"><span class="highlight-cl">}]
</span></span></code></pre>

```

我们可以看到,通过该处理器处理之后,新的文档中的 tag 字段都新增了一个"家居"标签,同时如果文档没有该字段,则会新建该字段.

#### Convert Processor

该类型的 Processor 是用于在写入某些文档之前对该文档的字段类型进行转换,比如文档中有个字符串类型的"价格"字段,我们希望将其转换成 float 类型,则我们可以使用该处理器来完成这项操作,实现下:

```

<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">POST _ingest/pipeline/_simulate
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "pipeline": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "description":

```





```

",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "code": "00
003",
</span></span><span class="highlight-line"><span class="highlight-cl">        "price": 799
8
</span></span><span class="highlight-line"><span class="highlight-cl">      },
</span></span><span class="highlight-line"><span class="highlight-cl">      "_ingest": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "timestamp":
"2017-11-26T10:56:47.664Z"
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl">>
</span></span></code></pre>
<p>可以看到,新的文档,price 字段将不再是字符串类型了,而是 float 类型.</p>
<p>使用 Convert Processor 需要注意的是,目前官方支持转换的类型有如下几个:integer, float, strin
, boolean, and auto. </p>
<p>关于更多该处理器的使用说明,请参考: <a href="https://ld246.com/forward?goto=https%3A
2F%2Fwww.elastic.co%2Fguide%2Fen%2Felasticsearch%2Freference%2Fcurrent%2Fconvert-p
rocessor.html" target="_blank" rel="nofollow ugc">https://www.elastic.co/guide/en/elasticsea
ch/reference/current/convert-processor.html</a></p>
<h4 id="Date-Processor">Date Processor</h4>
<p>日期处理器,顾名思义,就是将原文档中的某个日期字段转换成一个 Elasticsearch 识别的时间戳字
(一般默认为 @timestamp),该时间戳字段将会新增到原文档中.(目前发现该功能比较鸡肋,因为转换后
时间格式只能是 ISO 8601 时间格式的,并不能转换成其他格式,因此很少用)</p>
<p>下面是一个使用 Date Processor 的例子:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">POST _ingest/pipeline/_simulate
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "pipeline": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "description":
转换成指定格式的日期格式字段",
</span></span><span class="highlight-line"><span class="highlight-cl">    "processors": [
</span></span><span class="highlight-line"><span class="highlight-cl">      {
</span></span><span class="highlight-line"><span class="highlight-cl">        "date": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "field": "date"
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">      },
</span></span><span class="highlight-line"><span class="highlight-cl">      "formats": ["
NIX"],
</span></span><span class="highlight-line"><span class="highlight-cl">      "timezone":
Asia/Shanghai"
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl"> },
</span></span><span class="highlight-line"><span class="highlight-cl"> "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">   {
</span></span><span class="highlight-line"><span class="highlight-cl">     "_index": "inde
",
</span></span><span class="highlight-line"><span class="highlight-cl">     "_type": "type"

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
1",
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
',
</span></span><span class="highlight-line"><span class="highlight-cl">
96379"
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>
<p>就是将原有的字符串表示的日期格式进行格式转换(最终转换成的是 ISO 时间格式),结果为:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
ex",
</span></span><span class="highlight-line"><span class="highlight-cl">
',
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
696379",
</span></span><span class="highlight-line"><span class="highlight-cl">
001",
</span></span><span class="highlight-line"><span class="highlight-cl">
p": "2017-11-26T19:39:39.000+08:00",
</span></span><span class="highlight-line"><span class="highlight-cl">
8",
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
"2017-11-26T12:01:16.787Z"
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>
<p>我们可以看到多了一个默认字段 @timestamp,该字段名称可以通过 target_field 字段来指定该段名称.</p>
<p>为什么说只能生成 ISO 格式的时间呢? 这方面的资料官方没有提供,只能自己去源码中看,这里</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">public void execute(IngestDocument ingestDocument) {
</span></span><span class="highlight-line"><span class="highlight-cl">    String value
    ingestDocument.getFieldValue(field, String.class);
</span></span><span class="highlight-line"><span class="highlight-cl">

```

```

</span></span><span class="highlight-line"><span class="highlight-cl"> DateTime da
eTime = null;
</span></span><span class="highlight-line"><span class="highlight-cl"> Exception las
Exception = null;
</span></span><span class="highlight-line"><span class="highlight-cl"> for (Functio
<&lt;String, DateTime&&gt; dateParser : dateParsers) {
</span></span><span class="highlight-line"><span class="highlight-cl"> try {
</span></span><span class="highlight-line"><span class="highlight-cl"> dateTim
= dateParser.apply(value);
</span></span><span class="highlight-line"><span class="highlight-cl"> } catch (Ex
eption e) {
</span></span><span class="highlight-line"><span class="highlight-cl"> //try the
next parser and keep track of the exceptions
</span></span><span class="highlight-line"><span class="highlight-cl"> lastExce
tion = ExceptionsHelper.useOrSuppress(lastException, e);
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> if (dateTime
= null) {
</span></span><span class="highlight-line"><span class="highlight-cl"> throw new
IllegalArgumenteException("unable to parse date [" + value + "]", lastException);
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> ingestDocu
ent.setFieldValue(targetField, ISODateTimeFormat.dateTime().print(dateTime));
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>

```

<p>其中最后一行,在设置字段的时候,发现源码中已经写死为 ISODateTimeFormat.个人感觉官方应该会自定义 pattern.</p>

#### <p>Date Index Name Processor 算得上是一个比较强大的处理了,它的目的是让通过该处理器的文能够分配到符合指定时间格式的索引中,前提是按照官方提供的说明来进行使用:</p> <p>我们先创建一个该类型的 pipeline,如下:</p> ``` <pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">curl -XPUT 'localhost:9200/_ingest/pipeline/monthlyindex?pretty' -H 'Content-Type: appli ation/json' -d' </span></span><span class="highlight-line"><span class="highlight-cl">{ </span></span><span class="highlight-line"><span class="highlight-cl"> "description": " onthly date-time index naming", </span></span><span class="highlight-line"><span class="highlight-cl"> "processors" : [ </span></span><span class="highlight-line"><span class="highlight-cl"> { </span></span><span class="highlight-line"><span class="highlight-cl"> "date_index_n me" : { </span></span><span class="highlight-line"><span class="highlight-cl"> "field" : "dat 1", </span></span><span class="highlight-line"><span class="highlight-cl"> "index_name prefix" : "myindex-", </span></span><span class="highlight-line"><span class="highlight-cl"> "date_roundi g" : "M" </span></span><span class="highlight-line"><span class="highlight-cl"> } </span></span><span class="highlight-line"><span class="highlight-cl"> } </span></span><span class="highlight-line"><span class="highlight-cl"> ] </span></span><span class="highlight-line"><span class="highlight-cl">}</code></pre> ``` 原文链接: [17. Ingest Node](#)

```

</span></span><span class="highlight-line"><span class="highlight-cl">'
</span></span></code></pre>
<p>我们再创建一个文档,如下(该文档会使用该 pipeline)</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">curl -XPUT 'localhost:9200/myindex/type/1?pipeline=monthlyindex&pretty' -H 'Content-Type: application/json' -d'
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "date1" : "2016-
4-25T12:02:01.789Z"
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">'
</span></span></code></pre>
<p>结果如下:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "_index" : "myin
ex-2016-04-01",
</span></span><span class="highlight-line"><span class="highlight-cl">  "_type" : "type",
</span></span><span class="highlight-line"><span class="highlight-cl">  "_id" : "1",
</span></span><span class="highlight-line"><span class="highlight-cl">  "_version" : 1,
</span></span><span class="highlight-line"><span class="highlight-cl">  "result" : "create
",
</span></span><span class="highlight-line"><span class="highlight-cl">  "_shards" : {
</span></span><span class="highlight-line"><span class="highlight-cl">    "total" : 2,
</span></span><span class="highlight-line"><span class="highlight-cl">    "successful" : 1,
</span></span><span class="highlight-line"><span class="highlight-cl">    "failed" : 0
</span></span><span class="highlight-line"><span class="highlight-cl">  },
</span></span><span class="highlight-line"><span class="highlight-cl">  "created" : true
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span></code></pre>
<p>以上请求将不会将此 document (文档) 放入 myindex 索引, 而是放入到 myindex-2016-04-0
索引中.</p>
<p>这就是日期索引名称处理器(Date Index Name Processor)强大的地方,我们写入的文档可以根据
中的某个日期格式的字段来指定该文档将写入哪个索引中,该功能配上 template,能够实现很强大的日
收集功能,比如按月按天来将日志写入 Elasticsearch.</p>
<p>如果了解 Date Index Name Processor 的使用,请参考:<a href="https://ld246.com/forward
goto=https%3A%2F%2Fwww.elastic.co%2Fguide%2Fen%2Felasticsearch%2Freference%2Fcurrent%2Fdate-index-name-processor.html" target="_blank" rel="nofollow ugc">Date Index N
me Processor</a></p>
<h4 id="Fail-Processor">Fail Processor</h4>
<p>该处理器比较简单,就是当文档通过该 pipeline 的时候,一旦出现异常,该 pipeline 指定的错误信
就会返回给请求者.</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">POST _ingest/pipeline/_simulate
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "pipeline": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "description":
monthly date-time index naming",
</span></span><span class="highlight-line"><span class="highlight-cl">  "processors" : [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "fail": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "message": "a
error message"
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span></code></pre>

```



```

</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl"> },
</span></span><span class="highlight-line"><span class="highlight-cl"> "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl"> {
</span></span><span class="highlight-line"><span class="highlight-cl">   "_index": "myi
dex",
</span></span><span class="highlight-line"><span class="highlight-cl">   "_type": "type"

</span></span><span class="highlight-line"><span class="highlight-cl">   "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">   "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">     "id": "1"
</span></span><span class="highlight-line"><span class="highlight-cl">   }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>
<p>结果如下:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl"> "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl"> {
</span></span><span class="highlight-line"><span class="highlight-cl">   "error": {
</span></span><span class="highlight-line"><span class="highlight-cl">     "root_cause":

</span></span><span class="highlight-line"><span class="highlight-cl">     {
</span></span><span class="highlight-line"><span class="highlight-cl">       "type": "ex
ception",
</span></span><span class="highlight-line"><span class="highlight-cl">       "reason": "
ava.lang.IllegalArgumentException: org.elasticsearch.ingest.common.FailProcessorException:
n error message",
</span></span><span class="highlight-line"><span class="highlight-cl">       "header": {
</span></span><span class="highlight-line"><span class="highlight-cl">         "processo
_type": "fail"
</span></span><span class="highlight-line"><span class="highlight-cl">       }
</span></span><span class="highlight-line"><span class="highlight-cl">     }
</span></span><span class="highlight-line"><span class="highlight-cl">   },
</span></span><span class="highlight-line"><span class="highlight-cl">   "type": "exce
ption",
</span></span><span class="highlight-line"><span class="highlight-cl">   "reason": "ja
a.lang.IllegalArgumentException: org.elasticsearch.ingest.common.FailProcessorException: an
r error message",
</span></span><span class="highlight-line"><span class="highlight-cl">   "caused_by":

</span></span><span class="highlight-line"><span class="highlight-cl">   "type": "ille
al_argument_exception",
</span></span><span class="highlight-line"><span class="highlight-cl">   "reason": "o
g.elasticsearch.ingest.common.FailProcessorException: an error message",
</span></span><span class="highlight-line"><span class="highlight-cl">   "caused_by"
{
</span></span><span class="highlight-line"><span class="highlight-cl">   "type": "fai
_processor_exception",
</span></span><span class="highlight-line"><span class="highlight-cl">   "reason": "
n error message"

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">    },
</span></span><span class="highlight-line"><span class="highlight-cl">    "header": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "processor_
type": "fail"
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">    ]
</span></span><span class="highlight-line"><span class="highlight-cl">>
</span></span></code></pre>
<h4 id="Foreach-Processor">Foreach Processor</h4>
<p>一个 Foreach Processor 是用来处理一些数组字段,数组内的每个元素都会使用到一个相同的处理器,比如</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">POST_ingest/pipeline/_simulate
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "pipeline": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "description": "
foreach processor",
</span></span><span class="highlight-line"><span class="highlight-cl">  "processors" : [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "foreach": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "field": "value
",
</span></span><span class="highlight-line"><span class="highlight-cl">      "processor": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "uppercase":
{
</span></span><span class="highlight-line"><span class="highlight-cl">          "field": "_i
gest_value"
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">},
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "_index": "inde
",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_type": "type"
</span></span><span class="highlight-line"><span class="highlight-cl">      "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "id": "1",
</span></span><span class="highlight-line"><span class="highlight-cl">        "values": ["hel
o", "world", "felayman"]
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">>
</span></span></code></pre>

```

上面的例子中,文档中的 values 字段是一个数组类型的字段,其中每个元素都需要使用 Foreach Processor 中的一个共同的 uppercase Processor 来保证该字段中的每个元素都能执行相同的操作

<p>结果如下:</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "docs": [
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "doc": {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "_index": "in
ex",
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "_type": "typ
",
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "_id": "id",
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "_source": {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "id": "1",
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "values": [
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "HELLO",
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "WORLD",
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "FELAYM
N"
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ]
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> },
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "_ingest": {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "_value": nul
,
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "timestamp"
"2017-11-27T08:55:17.496Z"
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ]
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> </code> </pre>
```

<p>关于每个 Processor,深入下去都有许多需要说明的,这里没有深入,只是让大家了解其基本使用。 </p>

<p>详情请参考: <a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.elastic.co%2Fguide%2Fen%2Felasticsearch%2Freference%2Fcurrent%2Fforeach-processor.html" target=\_blank rel="nofollow ugc">Foreach Processore</a> </p>

#### <h4 id="Grok-Processor">Grok Processor</h4>

<p>Grok Processor 可以算的上是一个比较实用的处理器了,会经常使用到日志格式切割上,有用过 lo stash 的用户应该都知道 Grok 的强大.这里并不会涉及到到 Grok 详细的语法知识.这里我们只是略过说明 Elasticsearch 中的ingest node 中的 pipeline 也能提供 logstash 中的 Grok 的功能.</p>

<p>如下的使用例子:</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> POST _ingest/pipeline/_simulate
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "pipeline": {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "description": "
rok processor",
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "processors" : [
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "grok": {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "field": "mess
ge",
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "patterns": [
%{IP:client} %{WORD:method} %{URIPATHPARAM:request} %{NUMBER:bytes} %{NUMBER:du
```

```

ation}]
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl"> },
</span></span><span class="highlight-line"><span class="highlight-cl"> "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl"> {
</span></span><span class="highlight-line"><span class="highlight-cl">   "_index": "inde
",
</span></span><span class="highlight-line"><span class="highlight-cl">   "_type": "type"

</span></span><span class="highlight-line"><span class="highlight-cl">   "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">   "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">     "message": "5
.3.244.1 GET /index.html 15824 0.043"
</span></span><span class="highlight-line"><span class="highlight-cl">   }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl">> }
</span></span></code></pre>
<p>上面例子中的日志可能是 nginx 的日志,格式为:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">55.3.244.1 GET /index.html 15824 0.043
</span></span></code></pre>
<p>对应的 Grok 语法为:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">%{IP:client} %{WORD:method} %{URIPATHPARAM:request} %{NUMBER:bytes} %{NUMBER
duration}
</span></span></code></pre>
<p>因此该文本格式的日志信息在经过 Grok Processor 之后,能够解析成一个标准文档的格式,该文
可以使用 Elasticsearch 提供的检索和聚合功能充分使用到,而原始的文本格式的日志信息则无法做到
一点</p>
<p>返回结果如下:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "doc": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "_index": "in
ex",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_type": "typ
",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "duration":
0.043",
</span></span><span class="highlight-line"><span class="highlight-cl">          "request": "/"
index.html",
</span></span><span class="highlight-line"><span class="highlight-cl">          "method":
"GET",
</span></span><span class="highlight-line"><span class="highlight-cl">          "bytes": "15
24",
</span></span><span class="highlight-line"><span class="highlight-cl">          "client": "55
3.244.1",

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">      "message":
55.3.244.1 GET /index.html 15824 0.043"
</span></span><span class="highlight-line"><span class="highlight-cl">    },
</span></span><span class="highlight-line"><span class="highlight-cl">    "_ingest": {
</span></span><span class="highlight-line"><span class="highlight-cl">      "timestamp":
"2017-11-27T09:07:40.782Z"
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl">> ]
</span></span><span class="highlight-line"><span class="highlight-cl">> }
</span></span></code></pre>
<p>可以看到,对应的切割点都转换成文档中的一个字段.</p>
<p>关于 Grok Processor 的详细介绍,请参考:<a href="https://ld246.com/forward?goto=https%
A%2F%2Fwww.elastic.co%2Fguide%2Fen%2Felasticsearch%2Freference%2Fcurrent%2Fgrok-p
rocessor.html" target="_blank" rel="nofollow ugc">Grok Processor</a></p>
<h4 id="Gsub-Processor">Gsub Processor</h4>
<p>Gsub Processor 能够解决一些字符串中才特有的问题,比如我想把字符串格式的日期格式如"yyyy
MM-dd HH:mm:ss"转换成"yyyy/MM/dd HH:mm:ss"的格式,我们可以借助于 Gsub Processor 来
决,而 Gsub Processor 也正是利用正则来完成这一任务的.</p>
<p>如:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">POST _ingest/pipeline/_simulate
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "pipeline": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "description": "
sub processor",
</span></span><span class="highlight-line"><span class="highlight-cl">  "processors": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "gsub": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "field": "mess
age",
</span></span><span class="highlight-line"><span class="highlight-cl">        "pattern": "-",
</span></span><span class="highlight-line"><span class="highlight-cl">        "replacement
: "/"
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl"> },
</span></span><span class="highlight-line"><span class="highlight-cl"> "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">  {
</span></span><span class="highlight-line"><span class="highlight-cl">    "_index": "inde
",
</span></span><span class="highlight-line"><span class="highlight-cl">    "_type": "type"

</span></span><span class="highlight-line"><span class="highlight-cl">    "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">    "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">      "message": "2
17-11-27 00:00:00"
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl">> }
</span></span></code></pre>

```

<p>返回结果为:</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">{
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "docs": [
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "doc": {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "_index": "in
ex",
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "_type": "typ
",
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "_id": "id",
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "_source": {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "message":
2017/11/27 00:00:00"
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> },
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "_ingest": {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "timestamp":
"2017-11-27T09:15:05.502Z"
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ]
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> </code> </pre>
```

<p>就将 message 字段中的日期格式修改成我们想要的格式了.</p>

<p>更多关于 Gsub Processor 的介绍,请参考 <a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.elastic.co%2Fguide%2Fen%2Felasticsearch%2Freference%2Fcurrent%2Fgsubprocessor.html" target="\_blank" rel="nofollow ugc">Gsub Processor</a> </p>

<h4 id="Join-Processor">Join Processor</h4>

<p>Join Processor 能够将原本一个数组类型的字段值,分解成以指定分隔符分割的一个字符串.</p>

<p>如下例子:</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">POST _ingest/pipeline/_simulate
</span> </span> <span class="highlight-line"> <span class="highlight-cl">{
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "pipeline": {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "description": "j
in processor",
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "processors": [
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "join": {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "field": "me
sage",
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "separator":
",",
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "separator":
",",
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ]
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> },
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "docs": [
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "_index": "inde
x",
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> "_type": "type"
</span> </span> </code> </pre>
```

```

</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
hello,"world","felayman"]
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span></code></pre>
<p>结果为:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">{
</span></span><span class="highlight-line"><span class="highlight-cl"> "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl"> {
</span></span><span class="highlight-line"><span class="highlight-cl"> "doc": {
</span></span><span class="highlight-line"><span class="highlight-cl"> "_index": "in
ex",
</span></span><span class="highlight-line"><span class="highlight-cl"> "_type": "typ
",
</span></span><span class="highlight-line"><span class="highlight-cl"> "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl"> "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl"> "message":
hello、、、、world、、、、felayman"
</span></span><span class="highlight-line"><span class="highlight-cl"> },
</span></span><span class="highlight-line"><span class="highlight-cl"> "_ingest": {
</span></span><span class="highlight-line"><span class="highlight-cl"> "timestamp":
"2017-11-27T09:25:41.260Z"
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>
<p>可以看到,原文档中数组格式的的 message 字段,经过 Join Processor 处理后,变成了字符串类型
字段"hello、、、、world、、、、felayman".</p>
<p>更多关于 Join Processor 的内容,请参考:<a href="https://ld246.com/forward?goto=https%
A%2F%2Fwww.elastic.co%2Fguide%2Fen%2Felasticsearch%2Freference%2Fcurrent%2Fjoin-p
rocessor.html" target="_blank" rel="nofollow ugc">Join Processor</a></p>
<h4 id="JSON-Processor">JSON Processor</h4>
<p>JSON Processor 也是用来处理字符串类型的字段,可以将那些符合 JSON 格式(或被 JSON 串化)
文本,在经过 JSON Processor 加工之后,解析成对应的 JSON 格式,如下例子:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">POST _ingest/pipeline/_simulate
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl"> "pipeline": {
</span></span><span class="highlight-line"><span class="highlight-cl"> "description": "j
in processor",
</span></span><span class="highlight-line"><span class="highlight-cl"> "processors": [
</span></span><span class="highlight-line"><span class="highlight-cl"> {
</span></span><span class="highlight-line"><span class="highlight-cl"> "json": {
</span></span><span class="highlight-line"><span class="highlight-cl"> "field": "me
sage"
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>

```

```

</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl"> },
</span></span><span class="highlight-line"><span class="highlight-cl"> "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl"> {
</span></span><span class="highlight-line"><span class="highlight-cl">   "_index": "inde
",
</span></span><span class="highlight-line"><span class="highlight-cl">   "_type": "type"
}
</span></span><span class="highlight-line"><span class="highlight-cl">   "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">   "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">     "message": "{
foo\": 2000}"
}
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>
<p>结果为:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl"> "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl"> {
</span></span><span class="highlight-line"><span class="highlight-cl">   "doc": {
</span></span><span class="highlight-line"><span class="highlight-cl">     "_index": "in
ex",
</span></span><span class="highlight-line"><span class="highlight-cl">     "_type": "typ
",
</span></span><span class="highlight-line"><span class="highlight-cl">     "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">     "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">       "message": {
</span></span><span class="highlight-line"><span class="highlight-cl">         "foo": 2000
</span></span><span class="highlight-line"><span class="highlight-cl">       }
</span></span><span class="highlight-line"><span class="highlight-cl">     },
</span></span><span class="highlight-line"><span class="highlight-cl">     "_ingest": {
</span></span><span class="highlight-line"><span class="highlight-cl">       "timestamp":
"2017-11-27T09:30:28.546Z"
</span></span><span class="highlight-line"><span class="highlight-cl">   }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>
<p>可以看到,原本是字符串格式的字段 message,在处理之后变成了一个标准的 JSON 格式.</p>
<p>更多关于 JSON Processor 的内容,请参考:<a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.elastic.co%2Fguide%2Fen%2Felasticsearch%2Freference%2Fcurrent%2Fjson-processor.html" target="_blank" rel="nofollow ugc">JSON Processor</a></p>
<h4 id="KV-Processor">KV Processor</h4>
<p>KV Processor 用来 K,V 字符串格式的处理,比如 K=V, K:V,K-&gt;V 等格式的解析.</p>
<p>如下例:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">POST _ingest/pipeline/_simulate
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "pipeline": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "description": "

```



```

v processor",
</span></span><span class="highlight-line"><span class="highlight-cl"> "processors": [
</span></span><span class="highlight-line"><span class="highlight-cl">   {
</span></span><span class="highlight-line"><span class="highlight-cl">     "kv": {
</span></span><span class="highlight-line"><span class="highlight-cl">       "field": "me
sage",
</span></span><span class="highlight-line"><span class="highlight-cl">       "field_split":
"
",
</span></span><span class="highlight-line"><span class="highlight-cl">       "value_split"
":
",
</span></span><span class="highlight-line"><span class="highlight-cl">     }
</span></span><span class="highlight-line"><span class="highlight-cl">   }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl"> },
</span></span><span class="highlight-line"><span class="highlight-cl"> "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">   {
</span></span><span class="highlight-line"><span class="highlight-cl">     "_index": "inde
",
</span></span><span class="highlight-line"><span class="highlight-cl">     "_type": "type"
",
</span></span><span class="highlight-line"><span class="highlight-cl">     "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">     "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">       "message": "ip
127.0.0.1"
</span></span><span class="highlight-line"><span class="highlight-cl">     }
</span></span><span class="highlight-line"><span class="highlight-cl">   }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>
<p>其中字段 message 的原始值为"ip:127.0.0.1",我们想将该格式的内容新增一个独立的字段如 ip:
27.0.0.1 这种格式.</p>
<p>结果为:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"> "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">   {
</span></span><span class="highlight-line"><span class="highlight-cl">     "doc": {
</span></span><span class="highlight-line"><span class="highlight-cl">       "_index": "in
ex",
</span></span><span class="highlight-line"><span class="highlight-cl">       "_type": "typ
",
</span></span><span class="highlight-line"><span class="highlight-cl">       "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">       "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">         "message": "
p:107.0.0.1",
</span></span><span class="highlight-line"><span class="highlight-cl">         "ip": "107.0.
.1"
</span></span><span class="highlight-line"><span class="highlight-cl">       },
</span></span><span class="highlight-line"><span class="highlight-cl">       "_ingest": {
</span></span><span class="highlight-line"><span class="highlight-cl">         "timestamp"
"2017-11-27T09:46:48.715Z"
</span></span><span class="highlight-line"><span class="highlight-cl">     }
</span></span><span class="highlight-line"><span class="highlight-cl">   }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code>

```

```

</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span></code></pre>
<p>更多关于 KV Processor 的内容,请参考:<a href="https://ld246.com/forward?goto=https%3
%2F%2Fwww.elastic.co%2Fguide%2Fen%2Felasticsearch%2Freference%2Fcurrent%2Fkv-proc
ssor.html" target="_blank" rel="nofollow ugc">KV Processor</a></p>
<h4 id="Lowercase-Processor">Lowercase Processor</h4>
<p>Lowercase Processor 也是一个专用于字符串类型的字段处理器,顾名思义,是将字符串都转换成
写格式.</p>
<p>如下:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">POST_ingest/pipeline/_simulate
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "pipeline": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "description": "l
wercase processor",
</span></span><span class="highlight-line"><span class="highlight-cl">    "processors": [
</span></span><span class="highlight-line"><span class="highlight-cl">      {
</span></span><span class="highlight-line"><span class="highlight-cl">        "lowercase": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "field": "me
sage"
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    ]
</span></span><span class="highlight-line"><span class="highlight-cl">  },
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "_index": "inde
",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_type": "type"
</span></span><span class="highlight-line"><span class="highlight-cl">      "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "message": "
ELLO,WORLD,FELAYMAN."
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">}</span></span></code></pre>
<p>结果为:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "doc": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "_index": "in
ex",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_type": "typ
",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "message":
hello,world,felayman."
</span></span></code></pre>

```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> },
</span></span><span class="highlight-line"><span class="highlight-cl"> "ingest": {
</span></span><span class="highlight-line"><span class="highlight-cl"> "timestamp"
"2017-11-27T09:49:58.564Z"
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl">> ]
</span></span></code></pre>
```

<p>可以看到原文档中的 message 字段的值,在使用 Lowercase Processor 之后,都变成大写的了.</p>

<p>更多关于 Lowercase Processor 的内容,请参考:<a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.elastic.co%2Fguide%2Fen%2Felasticsearch%2Freference%2Fcurrent%2Fk-processor.html" target="\_blank" rel="nofollow ugc">Lowercase Processor</a></p>

#### <p>Remove Processor 是用来处理在写入文档之前,删除原文档中的某些字段值的.</p> <p>如下:</p> ``` <pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">POST _ingest/pipeline/_simulate </span></span><span class="highlight-line"><span class="highlight-cl">{ </span></span><span class="highlight-line"><span class="highlight-cl"> "pipeline": { </span></span><span class="highlight-line"><span class="highlight-cl"> "description": " remove processor", </span></span><span class="highlight-line"><span class="highlight-cl"> "processors": [ </span></span><span class="highlight-line"><span class="highlight-cl"> { </span></span><span class="highlight-line"><span class="highlight-cl"> "remove": { </span></span><span class="highlight-line"><span class="highlight-cl"> "field": "extr _field" </span></span><span class="highlight-line"><span class="highlight-cl"> } </span></span><span class="highlight-line"><span class="highlight-cl"> } </span></span><span class="highlight-line"><span class="highlight-cl"> ] </span></span><span class="highlight-line"><span class="highlight-cl"> }, </span></span><span class="highlight-line"><span class="highlight-cl"> "docs": [ </span></span><span class="highlight-line"><span class="highlight-cl"> { </span></span><span class="highlight-line"><span class="highlight-cl"> "_index": "inde ", </span></span><span class="highlight-line"><span class="highlight-cl"> "_type": "type" </span></span><span class="highlight-line"><span class="highlight-cl"> "_id": "id", </span></span><span class="highlight-line"><span class="highlight-cl"> "_source": { </span></span><span class="highlight-line"><span class="highlight-cl"> "extra_field": extra_field", </span></span><span class="highlight-line"><span class="highlight-cl"> "message": " ello, felayman." </span></span><span class="highlight-line"><span class="highlight-cl"> } </span></span><span class="highlight-line"><span class="highlight-cl"> } </span></span><span class="highlight-line"><span class="highlight-cl"> ] </span></span><span class="highlight-line"><span class="highlight-cl">> ] </span></span></code></pre> ``` <p>在经过 Remove Processor 处理后,extra\_field 字段将不存在了.</p> <p>结果为下:</p> ``` <pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">{ ``` 原文链接: 17. Ingest Node

```

</span></span><span class="highlight-line"><span class="highlight-cl"> "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl"> {
</span></span><span class="highlight-line"><span class="highlight-cl">   "doc": {
</span></span><span class="highlight-line"><span class="highlight-cl">     "_index": "in
ex",
</span></span><span class="highlight-line"><span class="highlight-cl">     "_type": "typ
",
</span></span><span class="highlight-line"><span class="highlight-cl">     "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">     "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">       "message": "
hello, felayman."
</span></span><span class="highlight-line"><span class="highlight-cl">     },
</span></span><span class="highlight-line"><span class="highlight-cl">     "_ingest": {
</span></span><span class="highlight-line"><span class="highlight-cl">       "timestamp"
"2017-11-27T09:58:46.038Z"
</span></span><span class="highlight-line"><span class="highlight-cl">     }
</span></span><span class="highlight-line"><span class="highlight-cl">   }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl">> ]
</span></span></code></pre>
<h4 id="Rename-Processor">Rename Processor</h4>
<p>Rename Processor 是用来处理在文档写入 Elasticsearch 之前修改某个文档的字段名称</p>
<p>如下:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl"><span class="highlight-line"><span class="highlight-cl">POST _ingest/pipeline/_simulate
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-line"><span class="highlight-cl">  "pipeline": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "description": "
remove processor",
</span></span><span class="highlight-line"><span class="highlight-cl">    "processors": [
</span></span><span class="highlight-line"><span class="highlight-cl">      {
</span></span><span class="highlight-line"><span class="highlight-cl">        "rename": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "field": "old_
ame",
</span></span><span class="highlight-line"><span class="highlight-cl">          "target_field"
"new_name"
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    ]
</span></span><span class="highlight-line"><span class="highlight-cl">  },
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "_index": "inde
",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_type": "type"
</span></span><span class="highlight-line"><span class="highlight-cl">      "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "old_name": "o
ld_name"
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span></code></pre>

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>
<p>会发现原文档的字段 old_name 被重新命名为 new_name 字段</p>
<p>结果为:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "doc": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "_index": "in
ex",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_type": "typ
",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "new_name"
"old_name"
</span></span><span class="highlight-line"><span class="highlight-cl">        },
</span></span><span class="highlight-line"><span class="highlight-cl">        "_ingest": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "timestamp"
"2017-11-27T10:00:48.694Z"
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">>
</span></span></code></pre>
<h4 id="Script-Processor">Script Processor</h4>
<p>Script Processor 算的上是最强大的处理了,因为能充分利用 Elasticsearch 提供的脚本能力.</p>
<p>这里也不会详细介绍 Elasticsearch 中的脚本如何使用,有关信息,请参考:<a href="https://ld246.
om/forward?goto=https%3A%2F%2Fwww.elastic.co%2Fguide%2Fen%2Felasticsearch%2Frefe
rence%2Fcurrent%2Fmodules-scripting.html" target="_blank" rel="nofollow ugc">Script</a>
</p>
<p>我么看下在 Script Processor 中使用脚本的例子:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">POST _ingest/pipeline/_simulate
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "pipeline": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "description": "
cript processor",
</span></span><span class="highlight-line"><span class="highlight-cl">    "processors": [
</span></span><span class="highlight-line"><span class="highlight-cl">      {
</span></span><span class="highlight-line"><span class="highlight-cl">        "script": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "lang": "pain
ess",
</span></span><span class="highlight-line"><span class="highlight-cl">          "inline": "ctx
viewCount = (ctx.viewCount) *10"
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    ]
</span></span><span class="highlight-line"><span class="highlight-cl">  },
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "_index": "inde

```

```

",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_type": "type"
</span></span><span class="highlight-line"><span class="highlight-cl">      "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "_viewCount":
100
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">>
</span></span></code></pre>
<p>这里我们通过脚本让原文档中的 viewCount 字段的值扩大十倍,结果为:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">      "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "doc": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "_index": "in
ex",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_type": "typ
",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "_viewCount"
1000
</span></span><span class="highlight-line"><span class="highlight-cl">        },
</span></span><span class="highlight-line"><span class="highlight-cl">        "_ingest": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "timestamp"
"2017-11-27T10:07:56.112Z"
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">>
</span></span></code></pre>
<h4 id="Set-Processor">Set Processor</h4>
<p>Set Processor 作用于两种不同情况:</p>
<ol>
<li>指定字段存在时,修改指定字段的值</li>
<li>指定字段不存在时,新增该字段并设置该字段的值</li>
</ol>
<p>例子如下:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">POST _ingest/pipeline/_simulate
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "pipeline": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "description": "
et processor",
</span></span><span class="highlight-line"><span class="highlight-cl">    "processors": [
</span></span><span class="highlight-line"><span class="highlight-cl">      {
</span></span><span class="highlight-line"><span class="highlight-cl">        "set": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "field": "cate
ory",

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">      "value": "家
"
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">},
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "_index": "inde
"
'
</span></span><span class="highlight-line"><span class="highlight-cl">      "_type": "type"

</span></span><span class="highlight-line"><span class="highlight-cl">      "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "id": "0001"
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">>
</span></span></code></pre>
<p>这里我们为每个使用 Set Processor 的文档新增一个分类"家居"</p>
<p>结果为:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "doc": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "_index": "in
ex",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_type": "typ
",
'
</span></span><span class="highlight-line"><span class="highlight-cl">        "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "id": "0001",
</span></span><span class="highlight-line"><span class="highlight-cl">          "category":
家居"
</span></span><span class="highlight-line"><span class="highlight-cl">        },
</span></span><span class="highlight-line"><span class="highlight-cl">        "_ingest": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "timestamp"
"2017-11-27T10:12:05.306Z"
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">>
</span></span></code></pre>
<h4 id="Split-Processor">Split Processor</h4>
<p>Split Processor 用于将一个以指定分隔分开的字符串转换成一个数组类型的字段</p>
<p>如下:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">POST _ingest/pipeline/_simulate
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "pipeline": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "description": "

```

```

plit processor",
</span></span><span class="highlight-line"><span class="highlight-cl"> "processors": [
</span></span><span class="highlight-line"><span class="highlight-cl">   {
</span></span><span class="highlight-line"><span class="highlight-cl">     "split": {
</span></span><span class="highlight-line"><span class="highlight-cl">       "field": "mes
age",
</span></span><span class="highlight-line"><span class="highlight-cl">       "separator":
_"
</span></span><span class="highlight-line"><span class="highlight-cl">     }
</span></span><span class="highlight-line"><span class="highlight-cl">   }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl"> },
</span></span><span class="highlight-line"><span class="highlight-cl"> "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">   {
</span></span><span class="highlight-line"><span class="highlight-cl">     "_index": "inde
",
</span></span><span class="highlight-line"><span class="highlight-cl">     "_type": "type"
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">     "id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">     "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">       "message": "h
ello-world"
</span></span><span class="highlight-line"><span class="highlight-cl">     }
</span></span><span class="highlight-line"><span class="highlight-cl">   }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>
<p>其中原文档的 message 字段值将会有"hello-world"转换为["hello","world"]</p>
<p>结果为: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl"> "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">   {
</span></span><span class="highlight-line"><span class="highlight-cl">     "doc": {
</span></span><span class="highlight-line"><span class="highlight-cl">       "_index": "in
ex",
</span></span><span class="highlight-line"><span class="highlight-cl">       "_type": "typ
",
</span></span><span class="highlight-line"><span class="highlight-cl">       "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">       "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">         "message": [
</span></span><span class="highlight-line"><span class="highlight-cl">           "hell",
</span></span><span class="highlight-line"><span class="highlight-cl">           "world"
</span></span><span class="highlight-line"><span class="highlight-cl">         ]
</span></span><span class="highlight-line"><span class="highlight-cl">       },
</span></span><span class="highlight-line"><span class="highlight-cl">       "_ingest": {
</span></span><span class="highlight-line"><span class="highlight-cl">         "timestamp":
"2017-11-27T10:13:35.982Z"
</span></span><span class="highlight-line"><span class="highlight-cl">       }
</span></span><span class="highlight-line"><span class="highlight-cl">     }
</span></span><span class="highlight-line"><span class="highlight-cl">   }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>

```



#### Sort Processor

Sort Processor 用于处理数组类型的字段,可以将存储在原文档中某个数组类型的字段中的元素照升序或降序来对原元素进行排序

如下:

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">POST_ingest/pipeline/_simulate
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "pipeline": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "description": "
ort processor",
</span></span><span class="highlight-line"><span class="highlight-cl">  "processors": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "sort": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "field": "cate
ory",
</span></span><span class="highlight-line"><span class="highlight-cl">        "order": "asc
",
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">},
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "_index": "inde
",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_type": "type"
",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "category": [2
3,4,1]
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">>}
</span></span></code></pre>
```

我们使用升序来修改原文档 category 字段的值的存储排序

结果为:

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "doc": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "_index": "in
ex",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_type": "typ
",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">        "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "category": [
</span></span><span class="highlight-line"><span class="highlight-cl">            1,
</span></span><span class="highlight-line"><span class="highlight-cl">            2,
</span></span><span class="highlight-line"><span class="highlight-cl">            3,
</span></span><span class="highlight-line"><span class="highlight-cl">            4
</span></span></code></pre>
```

```

</span></span><span class="highlight-line"><span class="highlight-cl">    ]
</span></span><span class="highlight-line"><span class="highlight-cl">    },
</span></span><span class="highlight-line"><span class="highlight-cl">    "_ingest": {
</span></span><span class="highlight-line"><span class="highlight-cl">      "timestamp"
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  }
</span></span><span class="highlight-line"><span class="highlight-cl">  }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">>
</span></span></code></pre>

```

#### Trim Processor

哎,Elasticsearch 开发者真的没谁了,基本上能把 String 类中的方法都拿过来搬一套过来使用.

Trim Processor 是专门用于处理字符串两端的空格问题,如下

```

<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">POST _ingest/pipeline/_simulate
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "pipeline": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "description": "t
im processor",
</span></span><span class="highlight-line"><span class="highlight-cl">    "processors": [
</span></span><span class="highlight-line"><span class="highlight-cl">      {
</span></span><span class="highlight-line"><span class="highlight-cl">        "trim": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "field": "mes
age"
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    ]
</span></span><span class="highlight-line"><span class="highlight-cl">  },
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "_index": "inde
",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_type": "type"
</span></span><span class="highlight-line"><span class="highlight-cl">      "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "message": "
ello, felayman."
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">>
</span></span></code></pre>

```

请注意,是去除字符串两端的空格.

结果为:

```

<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "doc": {
</span></span><span class="highlight-line"><span class="highlight-cl">        "_index": "in
ex",

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">
",
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
hello, felayman."
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
"2017-11-27T09:56:17.946Z"
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>
<h4 id="Uppercase-Processor">Uppercase Processor</h4>
<p>该处理器类似于 Lowercase Processor,将字符串文本统一转换成大写.</p>
<p>如下:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">
cl">POST _ingest/pipeline/_simulate
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "pipeline": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "description": "
ppercase processor",
</span></span><span class="highlight-line"><span class="highlight-cl">    "processors": [
</span></span><span class="highlight-line"><span class="highlight-cl">      {
</span></span><span class="highlight-line"><span class="highlight-cl">        "uppercase": {
</span></span><span class="highlight-line"><span class="highlight-cl">          "field": "mes
age"
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">},
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {
</span></span><span class="highlight-line"><span class="highlight-cl">      "_index": "inde
",
</span></span><span class="highlight-line"><span class="highlight-cl">      "_type": "type"
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span><span class="highlight-line"><span class="highlight-cl">},
</span></span><span class="highlight-line"><span class="highlight-cl">  "message": "h
llo,world,felayman."
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  }
</span></span><span class="highlight-line"><span class="highlight-cl">]
</span></span><span class="highlight-line"><span class="highlight-cl">>
</span></span></code></pre>
<p>结果为:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">
cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl">    {

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
ex",
</span></span><span class="highlight-line"><span class="highlight-cl">
",
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
HELLO,WORLD,FELAYMAN."
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
"2017-11-27T09:53:33.672Z"
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>
<p>更多关于 Uppercase Processor 的内容,请参考:<a href="https://ld246.com/forward?goto=htps%3A%2F%2Fwww.elastic.co%2Fguide%2Fen%2Felasticsearch%2Freference%2Fcurrent%2Fppercase-processor.html" target="_blank" rel="nofollow ugc">Uppercase Processor</a></p>

```

#### Dot Expander Processor

### 自定义 Processor

如果发现官方提供的 Processor 不满足我们的加工逻辑怎么办? 不用担心,官方提供了很好的插机制来帮助我们对其进行扩展,想实现一个自定义的 Processor 需要两个过程:

- 实现 IngestPlugin 接口,并实现 IngestPlugin 中的 getProcessors 方法
- 实现自定义的 Processor

下面我们给一个具体的例子,我们将任何传入的一个字段的值都变成大写(最简化模型,不指定字段,用 ingest/pipeline/ simulate api 模拟该操作)

新建一个项目,名称为 elasticsearch-help,新建 FirstUpperPlugin 类,如下:

```

<pre><code class="language-go highlight-chroma"><span class="highlight-line"><span cla
s="highlight-cl"><span class="highlight-kn">package</span> <span class="highlight-nx">o
g</span><span class="highlight-p">.</span><span class="highlight-nx">elasticsearch</sp
n><span class="highlight-p">.</span><span class="highlight-nx">help</span><span class
"highlight-p">;</span>
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-kn">import</span> <span class="highlight-nx">org</span><span class="highlight-p"
.</span><span class="highlight-nx">elasticsearch</span><span class="highlight-p">.</spa
><span class="highlight-nx">help</span><span class="highlight-p">.</span><span class=
highlight-nx">processor</span><span class="highlight-p">.</span><span class="highlight
nx">FirstUpperProcessor</span><span class="highlight-p">;</span>
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-kn">import</span> <span class="highlight-nx">org</span><span class="highlight-p"
.</span><span class="highlight-nx">elasticsearch</span><span class="highlight-p">.</spa
><span class="highlight-nx">ingest</span><span class="highlight-p">.</span><span clas
="highlight-nx">Processor</span><span class="highlight-p">;</span>
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-kn">import</span> <span class="highlight-nx">org</span><span class="highlight-p"
.</span><span class="highlight-nx">elasticsearch</span><span class="highlight-p">.</spa
><span class="highlight-nx">plugins</span><span class="highlight-p">.</span><span cla

```

```

s="highlight-nx">IngestPlugin</span> <span class="highlight-p">;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-kn">import</span> <span class="highlight-nx">org</span> <span class="highlight-p"
.</span> <span class="highlight-nx">elasticsearch</span> <span class="highlight-p">.</spa
> <span class="highlight-nx">plugins</span> <span class="highlight-p">.</span> <span cla
s="highlight-nx">Plugin</span> <span class="highlight-p">;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-kn">import</span> <span class="highlight-nx">java</span> <span class="highlight-p"
.</span> <span class="highlight-nx">util</span> <span class="highlight-p">.</span> <spa
class="highlight-nx">Collections</span> <span class="highlight-p">;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-kn">import</span> <span class="highlight-nx">java</span> <span class="highlight-p"
.</span> <span class="highlight-nx">util</span> <span class="highlight-p">.</span> <spa
class="highlight-nx">Map</span> <span class="highlight-p">;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-cm">/**
</span> </span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span cla
s="highlight-cm"> * @author lei.fang@shijue.me
</span> </span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span cla
s="highlight-cm"> * @since . 2017-11-26
</span> </span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span cla
s="highlight-cm"> */</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-nx">public</span> <span class="highlight-nx">class</span> <span class="highlight-n
">FirstUpperPlugin</span> <span class="highlight-nx">extends</span> <span class="highl
ght-nx">Plugin</span> <span class="highlight-nx">implements</span> <span class="highl
ght-nx">IngestPlugin</span> <span class="highlight-p">{</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="h
ghlight-err">@</span> <span class="highlight-nx">Override</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="h
ghlight-nx">public</span> <span class="highlight-nx">Map</span> <span class="highlight
p">&lt;</span> <span class="highlight-nx">String</span> <span class="highlight-p">,</sp
n"> <span class="highlight-nx">Processor</span> <span class="highlight-p">.</span> <span
class="highlight-nx">Factory</span> <span class="highlight-p">&gt;</span> <span class="
ighlight-nf">getProcessors</span> <span class="highlight-p">(</span> <span class="highli
ht-nx">Processor</span> <span class="highlight-p">.</span> <span class="highlight-nx">P
arameters</span> <span class="highlight-nx">parameters</span> <span class="highlight-p"
)</span> <span class="highlight-p">{</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class=
highlight-k">return</span> <span class="highlight-nx">Collections</span> <span class="hi
hlight-p">.</span> <span class="highlight-nf">singletonMap</span> <span class="highlight
p">(</span> <span class="highlight-nx">FirstUpperProcessor</span> <span class="highlight
p">.</span> <span class="highlight-nx">TYPE</span> <span class="highlight-p">,</span>
span class="highlight-nx">new</span> <span class="highlight-nx">FirstUpperProcessor</s
an> <span class="highlight-p">.</span> <span class="highlight-nf">Factory</span> <span c
lass="highlight-p">());</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="h
ghlight-p">}</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-p">}</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> </code> </pre>

```

其中我们实现了 `getProcessors` 方法,该方法主要用来提供我们自定义的额 `Processor`,其中 `FirstUpperProcessor` 源码如下:

```
<pre> <code class="language-go highlight-chroma"> <span class="highlight-line"> <span cla
s="highlight-cl"> <span class="highlight-kn">package</span> <span class="highlight-nx">o
g</span> <span class="highlight-p">.</span> <span class="highlight-nx">elasticsearch</sp
n> <span class="highlight-p">.</span> <span class="highlight-nx">help</span> <span class
"highlight-p">.</span> <span class="highlight-nx">processor</span> <span class="highligh
-p">;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-kn">import</span> <span class="highlight-nx">org</span> <span class="highlight-p"
.</span> <span class="highlight-nx">elasticsearch</span> <span class="highlight-p">.</spa
> <span class="highlight-nx">ingest</span> <span class="highlight-p">.</span> <span clas
="highlight-nx">AbstractProcessor</span> <span class="highlight-p">;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-kn">import</span> <span class="highlight-nx">org</span> <span class="highlight-p"
.</span> <span class="highlight-nx">elasticsearch</span> <span class="highlight-p">.</spa
> <span class="highlight-nx">ingest</span> <span class="highlight-p">.</span> <span clas
="highlight-nx">ConfigurationUtils</span> <span class="highlight-p">;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-kn">import</span> <span class="highlight-nx">org</span> <span class="highlight-p"
.</span> <span class="highlight-nx">elasticsearch</span> <span class="highlight-p">.</spa
> <span class="highlight-nx">ingest</span> <span class="highlight-p">.</span> <span clas
="highlight-nx">IngestDocument</span> <span class="highlight-p">;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-kn">import</span> <span class="highlight-nx">org</span> <span class="highlight-p"
.</span> <span class="highlight-nx">elasticsearch</span> <span class="highlight-p">.</spa
> <span class="highlight-nx">ingest</span> <span class="highlight-p">.</span> <span clas
="highlight-nx">Processor</span> <span class="highlight-p">;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-kn">import</span> <span class="highlight-nx">java</span> <span class="highlight-p"
.</span> <span class="highlight-nx">util</span> <span class="highlight-p">.</span> <span
class="highlight-nx">Map</span> <span class="highlight-p">;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-cm">/**
</span> </span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span cla
s="highlight-cm"> * @author lei.fang@shijue.me
</span> </span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span cla
s="highlight-cm"> * @since . 2017-11-26
</span> </span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span cla
s="highlight-cm"> */</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-nx">public</span> <span class="highlight-nx">class</span> <span class="highlight-n
">FirstUpperProcessor</span> <span class="highlight-nx">extends</span> <span class="h
ghlight-nx">AbstractProcessor</span> <span class="highlight-p">{</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="h
ghlight-nx">public</span> <span class="highlight-nx">static</span> <span class="highligh
-nx">final</span> <span class="highlight-nx">String</span> <span class="highlight-nx">T
PE</span> <span class="highlight-p">=</span> <span class="highlight-s">"firstUpper"</s
an> <span class="highlight-p">;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="h
ghlight-nx">private</span> <span class="highlight-nx">final</span> <span class="highligh
```



```

</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class=
highlight-nx">public</span> <span class="highlight-nx">FirstUpperProcessor</span> <spa
class="highlight-nf">create</span><span class="highlight-p">(</span><span class="highl
ght-nx">Map</span><span class="highlight-p">&lt;</span><span class="highlight-nx">Str
ng</span><span class="highlight-p">,</span> <span class="highlight-nx">Processor</spa
><span class="highlight-p">.</span><span class="highlight-nx">Factory</span><span class="highli
ght-p">&gt;</span> <span class="highlight-nx">registry</span><span class="highli
ght-p">,</span> <span class="highlight-nx">String</span> <span class="highlight-nx">p
rocessorTag</span><span class="highlight-p">,</span></span>
</span></span><span class="highlight-line"><span class="highlight-cl">
    <span class="highlight-nx">Map</span><span class="highlight-p">&lt;</span><span class="highli
ght-nx">String</span><span class="highlight-p">,</span> <span class="highli
ght-nx">Object</span><span class="highlight-p">&gt;</span> <span class="highlight-nx">c
nfig</span><span class="highlight-p">)</span> <span class="highlight-nx">throws</span>
<span class="highlight-nx">Exception</span> <span class="highlight-p">{</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span clas
s="highlight-nx">String</span> <span class="highlight-nx">field</span> <span class="highl
ght-p">=</span> <span class="highlight-nx">ConfigurationUtils</span><span class="highli
ght-p">.</span><span class="highlight-nf">readStringProperty</span><span class="highli
ght-p">(</span><span class="highlight-nx">TYPE</span><span class="highlight-p">,</spa
> <span class="highlight-nx">processorTag</span><span class="highlight-p">,</span><span class="highli
ght-nx">config</span><span class="highlight-p">,</span> <span class="highli
ght-s">"field"</span><span class="highlight-p">);</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span clas
s="highlight-k">return</span> <span class="highlight-nx">new</span> <span class="highli
ght-nf">FirstUpperProcessor</span><span class="highlight-p">(</span><span class="highli
ght-nx">processorTag</span><span class="highlight-p">,</span> <span class="highlight-nx
">field</span><span class="highlight-p">);</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class=
highlight-p">}</span></span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class="h
ghlight-p">}</span></span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="high
ight-p">}</span></span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> <span class="high
ight-p">}</span></span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>

```

可以看到,其中核心的方法为 execute,主要是将传入的字段的价值转换成大写,这里只是调用了简单的 toUpperCase()方法.</p>

<p>我们使用 \_ingest/pipeline/\_simulate api 来进行测试:</p>

```

<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">curl -XGET 'http://localhost:9200/_ingest/pipeline/_simulate?pretty' -d '{
</span></span><span class="highlight-line"><span class="highlight-cl">  "pipeline": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "description":
字符串首字母转换成大写",
</span></span><span class="highlight-line"><span class="highlight-cl">    "processors": [
</span></span><span class="highlight-line"><span class="highlight-cl">      {
</span></span><span class="highlight-line"><span class="highlight-cl">        "firstUpper":{
</span></span><span class="highlight-line"><span class="highlight-cl">          "field": "mess
ge"
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">  ]
</span></span></code></pre>

```



```

</span></span><span class="highlight-line"><span class="highlight-cl"> },
</span></span><span class="highlight-line"><span class="highlight-cl"> "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl"> {
</span></span><span class="highlight-line"><span class="highlight-cl">   "_index": "inde
",
</span></span><span class="highlight-line"><span class="highlight-cl">   "_type": "type"

</span></span><span class="highlight-line"><span class="highlight-cl">   "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">   "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">     "message": "h
llo,world."
</span></span><span class="highlight-line"><span class="highlight-cl">   }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl">}'
</span></span></code></pre>
<p>结果为:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">{
</span></span><span class="highlight-line"><span class="highlight-cl"> "docs": [
</span></span><span class="highlight-line"><span class="highlight-cl"> {
</span></span><span class="highlight-line"><span class="highlight-cl">   "doc": {
</span></span><span class="highlight-line"><span class="highlight-cl">     "_index": "in
ex",
</span></span><span class="highlight-line"><span class="highlight-cl">     "_type": "typ
",
</span></span><span class="highlight-line"><span class="highlight-cl">     "_id": "id",
</span></span><span class="highlight-line"><span class="highlight-cl">     "_source": {
</span></span><span class="highlight-line"><span class="highlight-cl">       "message":
Hello,world."
</span></span><span class="highlight-line"><span class="highlight-cl">   },
</span></span><span class="highlight-line"><span class="highlight-cl">   "_ingest": {
</span></span><span class="highlight-line"><span class="highlight-cl">     "timestamp"
"2017-11-27T10:16:16.409Z"
</span></span><span class="highlight-line"><span class="highlight-cl">   }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> ]
</span></span><span class="highlight-line"><span class="highlight-cl">}'
</span></span></code></pre>
<p>可以看到,自定义的 Ingest node 插件成功了.</p>
<p>完结.</p>

```