

ElasticSearch系列文章：数据修改

Elasticsearch提供了近乎实时的数据操作和搜索功能。默认情况下，从你索引/更新/删除你的数据动作开始到它出现在你的搜索结果中，大概会有1秒钟的延迟。这和其它的SQL平台不同，它们的数据在一个事务完成之后就会立即可用。

索引/替换文档

我们先前看到，怎样索引一个文档。现在我们再次调用那个命令：

```
curl -XPUT 'localhost:9200/customer/external/1?pretty' -d '{
  "name": "John Doe"
}'
```

以上的命令将会把这个文档索引到customer索引、external类型中，其ID是1。如果我们对一个不同（或相同）的文档应用以上的命令，Elasticsearch将会用一个新的文档来替换（重新索引）当前ID为1的那个文档。

```
curl -XPUT 'localhost:9200/customer/external/1?pretty' -d '{
  "name": "Jane Doe"
}'
```

以上的命令将ID为1的文档的name字段的值从“John Doe”改成了“Jane Doe”。如果我们使用一个不同的ID，一个新的文档将会被索引，当前已经在索引中的文档则不会受到影响。

```
curl -XPUT 'localhost:9200/customer/external/2?pretty' -d '{
  "name": "Jane Doe"
}'
```

以上的命令，将会为一个ID为2的文档建立索引。

在索引的时候，ID部分是可选的。如果不指定，Elasticsearch将产生一个随机的ID来索引这个文档。Elasticsearch 生成的ID会作为索引API调用的一部分被返回。

下面的例子展示了怎样在没有指定ID的情况下来索引一个文档：

```
curl -XPOST 'localhost:9200/customer/external?pretty' -d '{
  "name": "Jane Doe"
}'
```

注意，在上面的情形中，由于我们没有指定一个ID，我们使用的是POST而不是PUT。

更新文档

除了可以索引、替换文档之外，我们也可以更新一个文档。但要注意，Elasticsearch底层并不支持原地更新。在我们想要做一次更新的时候，Elasticsearch先删除旧文档，然后再索引更新的新文档。

下面的例子展示了怎样将ID为1的文档的name字段改成“Jane Doe”：

```
curl -XPOST 'localhost:9200/customer/external/1/_update?pretty' -d '{
  "doc": { "name": "Jane Doe" }
}'
```

下面的例子展示了怎样将ID为1的文档的name字段改成“Jane Doe”的同时，给它加上age字段：

```
curl -XPOST 'localhost:9200/customer/external/1/_update?pretty' -d '{
  "doc": { "name": "Jane Doe", "age": 20 }
}'
```

更新也可以通过使用简单的脚本来进行。这个例子使用一个脚本将age加5：

```
curl -XPOST 'localhost:9200/customer/external/1/_update?pretty' -d '
```

```
{  
  "script" : "ctx._source.age += 5"  
}
```

在上面的例子中，ctx._source指向当前被更新的文档。

注意，目前的更新操作只能一次应用在一个文档上。将来Elasticsearch将提供同时更新符合指定查询条件的多个文档的功能（类似于SQL的UPDATE-WHERE语句）。

删除文档

删除文档是非常直观的。以下的例子展示了怎样删除ID为2的文档：

```
curl -XDELETE 'localhost:9200/customer/external/2?pretty'
```

也能够一次删除符合某个查询条件的多个文档。以下的例子展示了如何删除名字中包含“John”的所有客户：

```
curl -XDELETE 'localhost:9200/customer/external/_query?pretty' -d '  
{  
  "query": { "match": { "name": "John" } }  
}'
```

注意，以上的URI变成了/_query，以此来表明这是一个“查询删除”API，删除满足请求体中的查询条件的索引。我们仍然使用DELETE动词。

批处理

除了能够对单个的文档进行索引、更新和删除之外，Elasticsearch也提供了操作的批量处理功能，它通过使用_bulk API实现。这个功能之所以重要，是因为它提供了非常高效的机制来尽可能快的完成多个操作，与此同时使用尽可能少的网络往返。

作为一个快速的例子，以下调用在一次bulk操作中索引了两个文档（ID 1 - John Doe and ID 2 - Jane Doe）：

```
curl -XPOST 'localhost:9200/customer/external/_bulk?pretty' -d '
```

```
{"index":{"_id":"1"}}
{"name": "John Doe" }
{"index":{"_id":"2"}}
{"name": "Jane Doe" }
,
```

以下例子在一个bulk操作中，首先更新第一个文档（ID为1），然后删除第二个文档（ID为2）

```
curl -XPOST 'localhost:9200/customer/external/_bulk?pretty' -d '
{"update":{"_id":"1"}}
{"doc": { "name": "John Doe becomes Jane Doe" }}
{"delete":{"_id":"2"}}
,
```

注意上面的delete动作，由于删除动作只需要被删除文档的ID，所以并没有对应的源文档。

bulk API按顺序执行这些动作。如果其中一个动作因为某些原因失败了，它将会继续处理后面的动作。当bulk API返回时，它将提供每个动作的状态（按照同样的顺序），所以你能够看到某个动作成功与否。

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接：[【】（）](#)