

[23种非常实用的ElasticSearch查询例子\(3\)](#)

本系列文章将展示ElasticSearch中23种非常实用的查询使用方法。由于篇幅原因，本系列文章分为六篇，

本文是此系列的第三篇文章。欢迎关注大数据技术博客微信公共账号:iteblog_hadoop。

[《23种非常实用的ElasticSearch查询例子\(1\)》](#)

[《23种非常实用的ElasticSearch查询例子\(2\)》](#)

[《23种非常实用的ElasticSearch查询例子\(3\)》](#)

[《23种非常实用的ElasticSearch查询例子\(4\)》](#)

[《23种非常实用的ElasticSearch查询例子\(5\)》](#)

[《23种非常实用的ElasticSearch查询例子\(6\)》](#)

Match Phrase Query(匹配短语查询)

匹配短语查询要求查询字符串中的terms要么都出现在Document中、要么terms按照输入顺序依次出现在结果中。在默认情况下，查询输入的terms必须在搜索字符串紧挨着出现，否则将查询不到。不过我们可以指定slop参数，来控制输入的terms之间有多少个单词仍然能够搜索到，如下所示：

```
curl -XGET 'https://www.iteblog.com:9200/iteblog_book_index/book/_search' -d '{
  "query": {
    "multi_match": {
      "query": "search engine",
      "fields": [
        "title",
        "summary"
      ],
      "type": "phrase",
      "slop": 3
    }
  },
  "_source": [
    "title",
    "summary",
    "publish_date"
  ]
}'
```

[返回结果]

```
{
  "took": 17,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "failed": 0
  },
  "hits": {
    "total": 2,
    "max_score": 0.22327082,
    "hits": [
      {
        "_index": "iteblog_book_index",
        "_type": "book",
        "_id": "4",
        "_score": 0.22327082,
        "_source": {
          "summary": "Comprehensive guide to implementing a scalable search engine using Apache Solr",
          "title": "Solr in Action",
          "publish_date": "2014-04-05"
        }
      },
      {
        "_index": "iteblog_book_index",
        "_type": "book",
        "_id": "1",
        "_score": 0.16113183,
        "_source": {
          "summary": "A distributed real-time search and analytics engine",
          "title": "Elasticsearch: The Definitive Guide",
          "publish_date": "2015-02-07"
        }
      }
    ]
  }
}
```

从上面的例子可以看出，id为4的document被搜索（summary字段里面精确匹配到了search engine），并且分数比较高；而id为1的document也被搜索到了，虽然其summary中的search和engine单词并不是紧挨着的，但是我们指定了slop属性，所以被搜索到了。如果我们将"slop": 3条件删除，那么id为1的文档将不会被搜索到。



微信扫一扫，加关注
即可及时了解Spark、Hadoop或者Hbase
等相关的文章
欢迎关注微信公共帐号:iteblog_hadoop

过往记忆博客(<http://www.iteblog.com>)
专注于Hadoop、Spark、Flume、Hbase等
技术的博客，欢迎关注。

Hadoop、Hive、Hbase、Flume等交流群：138615359和149892483

如果想及时了
解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

Match Phrase Prefix Query(匹配短语前缀查询)

匹配短语前缀查询可以指定单词的一部分字符前缀即可查询到该单词，和match phrase query一样我们也可以指定slop参数；同时其还支持max_expansions参数限制被匹配到的terms数量来减少资源的使用。使用如下：

```
////////////////////////////////////  
User: 过往记忆  
Date: 2016-08-17  
Time: 23:36  
blog: https://www.iteblog.com  
本文地址：https://www.iteblog.com/archives/1747.html  
过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货  
过往记忆博客微信公共帐号：iteblog_hadoop  
////////////////////////////////////
```

```
curl -XGET 'https://www.iteblog.com:9200/iteblog_book_index/book/_search' -d '  
{  
  "query": {  
    "match_phrase_prefix": {  
      "summary": {  
        "query": "search en",  
        "slop": 3,  
        "max_expansions": 10  
      }  
    }  
  }  
},
```

```
"_source": [  
  "title",  
  "summary",  
  "publish_date"  
]  
'
```

[返回结果]

```
{  
  "took": 9,  
  "timed_out": false,  
  "_shards": {  
    "total": 1,  
    "successful": 1,  
    "failed": 0  
  },  
  "hits": {  
    "total": 2,  
    "max_score": 0.5161346,  
    "hits": [  
      {  
        "_index": "iteblog_book_index",  
        "_type": "book",  
        "_id": "4",  
        "_score": 0.5161346,  
        "_source": {  
          "summary": "Comprehensive guide to implementing a scalable search engine using Apache Solr",  
          "title": "Solr in Action",  
          "publish_date": "2014-04-05"  
        }  
      },  
      {  
        "_index": "iteblog_book_index",  
        "_type": "book",  
        "_id": "1",  
        "_score": 0.37248808,  
        "_source": {  
          "summary": "A distributed real-time search and analytics engine",  
          "title": "Elasticsearch: The Definitive Guide",  
          "publish_date": "2015-02-07"  
        }  
      }  
    ]  
  }  
}
```

```
}
```

需要留意的是，匹配短语前缀查询是有性能消耗的，所有使用之前需要小心。

Query String

query_string查询提供了一种手段可以使用一种简洁的方式运行multi_match queries, bool queries, boosting, fuzzy matching, wildcards, regexp以及range queries的组合查询。在下面的例子中，我们运行了一个模糊搜索(fuzzy search)，搜索关键字是search algorithm，并且作者包含grant ingersoll或者tom morton。并且搜索了所有的字段，其中summary字段的权重为2：

```
////////////////////////////////////
User: 过往记忆
Date: 2016-08-17
Time: 23:36
bolg: https://www.iteblog.com
本文地址：https://www.iteblog.com/archives/1747.html
过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
过往记忆博客微信公共帐号：iteblog_hadoop
////////////////////////////////////
curl -XGET 'https://www.iteblog.com:9200/iteblog_book_index/book/_search' -d '
{
  "query": {
    "query_string": {
      "query": "(saerch~1 algorithm~1) AND (grant ingersoll) OR (tom morton)",
      "fields": ["_all", "summary^2"]
    }
  },
  "_source": [ "title", "summary", "authors" ],
  "highlight": {
    "fields": {
      "summary": {}
    }
  }
}'
```

[返回结果]

```
{
  "took": 25,
  "timed_out": false,
```

```
"_shards": {
  "total": 1,
  "successful": 1,
  "failed": 0
},
"hits": {
  "total": 1,
  "max_score": 0.12186339,
  "hits": [
    {
      "_index": "iteblog_book_index",
      "_type": "book",
      "_id": "2",
      "_score": 0.12186339,
      "_source": {
        "summary": "organize text using approaches such as full-text search, proper name
recognition, clustering, tagging, information extraction, and summarization",
        "authors": [
          "grant ingersoll",
          "thomas morton",
          "drew farris"
        ],
        "title": "Taming Text: How to Find, Organize, and Manipulate It"
      },
      "highlight": {
        "summary": [
          "organize text using approaches such as full-text <em>search</em>, proper na
me recognition, clustering, tagging, information extraction, and summarization"
        ]
      }
    }
  ]
}
```

限于篇幅的原因，本系列文章分为六部分，欢迎关注过往记忆大数据技术博客及时了解大数据相关文章，微信公共账号：iteblog_hadoop。

本博客文章除特别声明，全部都是原创！
转载本文请加上：转载自过往记忆（<https://www.iteblog.com/>）
本文链接：【】（）