





Elasticsearch DSL 语法使用宝典

2022 年 8 月

1、本文档使用提要

- (1) 对搜索引擎有最基础的了解，Elasticsearch 是个什么，它是用来干嘛的，能帮我解决什么问题。这些基础问题不能是当前你看文档时的想法，本文档适用于已经处于对 ES 已经入门的开发人员。
- (2) 本文档内所有案例均使用 Chrome Head 插件操作。
- (3) 本文档内所有案例均使用 ES 7.6.0 以上版本。

2、初始化索引

2.1 写入一点文档-即自建索引

地址前缀：	http://175.6.147.66:9200/	地址后缀：	film/_doc/1
方式：	POST		
{ "title": "流浪地球", "publishDate": "2019-02-01", "year": 2019, "content": "《流浪地球》根据刘慈欣同名小说改编，故事设定在 2075 年，讲述了太阳即将毁灭，已经不适合人类生存，而面对绝境，人类将开启“流浪地球”计划，试图带着地球一起逃离太阳系，寻找人类新家园的故事", "director": "郭帆", "country": "中国", "price": "45 亿" }			
{ "title": "战狼 2", "publishDate": "2017-07-27", "year": 2017, "content": "战狼 2 是吴京执导的动作军事电影，由吴京、弗兰克·格里罗、吴刚、张翰、卢靖姗、淳于珊珊、丁海峰等主演。该片于 2017 年 7 月 27 日在中国内地上映。2020 年 3 月 20 日，《战狼 II》重映。该片讲述了脱下军装的冷锋被卷入了一场非洲国家的叛乱，本来能够安全撤离的他无法忘记军人的职责，重回战场展开救援的故事。该片以 56.8 亿元雄踞国产电影历史最高票房纪录。", "director": "吴京", "country": "中国", "price": "65 亿" }			
{ "title": "独行月球", "publishDate": "2022-08-07", "year": 2022, "content": "独行月球是由张迟昱执导，沈腾、马丽主演的科幻喜剧片，于 2022 年 7 月 29 日在中国大陆上映。该片改编自韩国漫画家赵石创作的同名漫画，讲述了人类为抵御小行星的撞击，拯救地球，在月球部署了月盾计划。陨石提前来袭，全员紧急撤离时，维修工独孤月因为意外，错过了领队马蓝星的撤离通知，一个人落在了月球。不料月盾计划失败，独孤月成为了“宇宙最后的人类”，开始了他在月球上			

破罐子破摔的生活的故事。", "director": "张迟昱", "country": "中国", "price": "28 亿" }
{ "title": "绿皮书", "publishDate": "2018-09-11", "year": 2018, "content": "绿皮书讲述是一个黑人钢琴家为了改变自己民族被歧视的命运,用行动去扭转白人、黑人观念的故事。里面白人歧视黑人的情节很多,比如:不让雪利博士进餐厅、不让雪利博士用厕所、不让雪利博士进他们的小镇、黑夜不让黑人出行.....还有因为这种种加诸在黑人身上的不公平待遇,所以一群黑人看到托尼为雪利博士服务觉得很不可思议。", "director": "彼得·法雷里", "country": "美国", "price": "35 亿" }
{ "title": "长津湖之水门桥", "publishDate": "2022-02-01", "year": 2022, "content": "长津湖之水门桥是由陈凯歌、徐克、林超贤监制,徐克执导,吴京、易烊千玺领衔主演,段奕宏特别主演,张涵予友情主演,朱亚文、李晨、韩东君、杜淳、耿乐主演的历史战争电影。该片以抗美援朝战争第二次战役中的长津湖战役后期的水门桥战役为背景,讲述志愿军第七连战士们在结束了新兴里和下碣隅里的战斗之后,又接到了更艰巨的阻击任务的故事。", "director": "徐克", "country": "中国", "price": "40.66 亿" }

3、删除 DSL

地址前缀：	http://175.6.147.66:9200/	地址后缀：	film/6
方式：	delete		

4、查询 DSL

4.1 分词器

地址前缀：	http://175.6.147.66:9200/	地址后缀：	_analyze
方式：	POST		
{ "analyzer": "standard", "text": "Elastic 资料 2022-08-19 分享会" }			
{ "analyzer": "ik_smart", "text": "Elastic 资料 2022-08-19 分享会" }			
{			

<pre>"analyzer": "ik_max_word", "text": "Elastic 资料 2022-08-19 分享会" }</pre>
/_analyze 代表分词分析。 analyzer 分词类型，这里是默认的 standard 分词器。 text 要分词的内容。
Ik 分词器对中文来说会很准确，但是它包含的大部分是汉语词典的词汇。 特殊化的操作：比如-专有词汇、同义词 要在它的基础上去做扩展。

4.2 查询标准语法

地址前缀：	http://175.6.147.66:9200/	地址后缀：	film/_search
方式：	POST		
<pre>{ "query": { "bool": { "must": [], "must_not": [], "should": [] } }, "from": 0, "size": 10, "sort": [], "aggs": {} }</pre>			
Must：必须 Must_not：必须不 Should：应该、可能、能够 From：开始 Size：本页数量 Sort：排序 Aggs：聚合			

4.3 简单查询

4.3.1 简单查询 - match

地址前缀：	http://175.6.147.66:9200/	地址后缀：	film/_search
方式：	POST		
<pre>{ "query": { "bool": { "must": [{ "match": { "title": "地球" } }], "must_not": [], "should": [] } }, "from": 0, "size": 10, </pre>			

<pre>"sort": [], "aggs": {} }</pre>

4.3.2 简单查询 - term

地址前缀：	http://175.6.147.66:9200/	地址后缀：	film/_search
方式：	POST		
<pre>{ "query": { "bool": { "must": [{ "term": { "title.keyword": "流浪地球" } }], "must_not": [], "should": [] } }, "from": 0, "size": 10, "sort": [], "aggs": {} }</pre>			

4.3.3 简单查询 - rang

地址前缀：	http://175.6.147.66:9200/	地址后缀：	film/_search
方式：	POST		
<pre>{ "query": { "bool": { "must": [{ "range": { "year": { "gte": 2018, "lte": 2022, "relation": "within" } } }], "must_not": [], "should": [] } }, "from": 0, "size": 10, "sort": [], "aggs": {} }</pre>			
注意到 relation 关键字。relation 字段上的范围查询支持一个关系参数，该参数可以是 WITHIN, CONTAINS, INTERSECTS 之一（默认）。 WITHIN：在内部、在含有内 CONTAINS：包含、含有			

4.4 复杂查询

4.4.1 复杂查询 - term + match

我们要查 2022 年上映的电影名称中可能包含地球关键词的 DSL。

地址前缀：	http://175.6.147.66:9200/	地址后缀：	film/_search
方式：	POST		

```
{
  "query": {
    "bool": {
      "must": [
        {
          "term": {
            "year": 2022
          }
        }
      ],
      "must_not": [],
      "should": [{
        "match": {
          "title": "地球"
        }
      }]
    }
  },
  "from": 0,
  "size": 10
}
```

Must : 强势语法, 必须、一定、必定

Should : 应该、可能、能够

我们要查 2022 年上映的电影名称中一定包含地球关键词的 DSL。

地址前缀：	http://175.6.147.66:9200/	地址后缀：	film/_search
方式：	POST		

```
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "title": "地球"
          }
        },
        {
          "term": {
            "year": 2022
          }
        }
      ],
      "must_not": [],
      "should": []
    }
  },
  "from": 0,
  "size": 10
}
```

Must：强势语法，必须、一定、必定 Should：应该、可能、能够

4.4.2 复杂查询 - term + match + sort

我们要查在中国创作上映的电影名称中可能包含地球关键词且做匹配相关性倒序的 DSL。			
地址前缀：	http://175.6.147.66:9200/	地址后缀：	film/_search
方式：	POST		
<pre>{ "query": { "bool": { "must": [{ "term": { "country.keyword": "中国" } }, { "match": { "title": "地球" } }], "must_not": [], "should": [] } }, "from": 0, "size": 10, "sort": [{ "_score": { "order": "desc" } }] }</pre>			
Sort：查询结果排序处理，可以多个			

我们要查在中国创作上映的电影名称中可能包含地球关键词且做上映年份倒序的 DSL。			
地址前缀：	http://175.6.147.66:9200/	地址后缀：	film/_search
方式：	POST		
<pre>{ "query": { "bool": { "must": [{ "term": { "country.keyword": "中国" } }, { "match": { "title": "地球" } }], "must_not": [], "should": [] } } }</pre>			

<pre> }, "from": 0, "size": 10, "sort": [{ "year": { "order": "desc" } }, { "_score": { "order": "desc" } }] } </pre>
Sort：查询结果排序处理，可以多个

4.4.3 复杂查询 - term + match + sort + 高亮

我们要查在中国创作上映的电影名称中可能包含地球关键词且做 上映年份倒序 ，并对关键词高亮的DSL。			
地址前缀：	http://175.6.147.66:9200/	地址后缀：	film/_search
方式：	POST		
<pre> { "query": { "bool": { "must": [{ "term": { "country.keyword": "中国" } }, { "match": { "title": "地球" } }], "must_not": [], "should": [] } }, "from": 0, "size": 10, "sort": [{ "year": { "order": "desc" } }, { "_score": { "order": "desc" } }], "highlight": { "pre_tags": [""], </pre>			

<pre> "post_tags": [""], "require_field_match": true, "fields": { "title": {} } } </pre>
Highlight：高亮属性，返回结果内有高亮的数据集。 pre_tags：高亮的HTML 标签开始 post_tags：高亮的HTML 标签结束 Fields：做match 查询的列名称，可以多个

我们要查在中国创作上映的电影名称中和电影简述中可能包含地球关键词且做 上映年份倒序 ，并对关键词高亮的DSL。			
地址前缀：	http://175.6.147.66:9200/	地址后缀：	film/_search
方式：	POST		
<pre> { "query": { "bool": { "must": [{ "term": { "country.keyword": "中国" } }, { "match": { "title": "地球" } }, { "match": { "content": "地球" } }], "must_not": [], "should": [] } }, "from": 0, "size": 10, "sort": [{ "year": { "order": "desc" } }, { "_score": { "order": "desc" } }], "highlight": { "pre_tags": [""], "post_tags": [""], }, </pre>			

<pre> "require_field_match": true, "fields": { "title": {}, "content": {} } } </pre>
Highlight：高亮属性，返回结果内有高亮的数据集。 pre_tags：高亮的HTML 标签开始 post_tags：高亮的HTML 标签结束 Fields：做match 查询的列名称，可以多个

4.4.4 复杂查询 - term + match + sort + 高亮 + 去除低分

我们要查在中国创作上映的电影名称中和电影简述中可能包含地球关键词且做上映年份倒序，同时去除干扰数据并对关键词高亮的DSL。

地址前缀：	http://175.6.147.66:9200/	地址后缀：	film/_search
方式：	POST		
<pre> { "query": { "bool": { "must": [{ "term": { "country.keyword": "中国" } }, { "match": { "title": "地球" } }, { "match": { "content": "地球" } }], "must_not": [], "should": [] } }, "from": 0, "size": 10, "sort": [{ "year": { "order": "desc" } }, { "_score": { "order": "desc" } }], "highlight": { "pre_tags": [""], </pre>			

<pre>"post_tags": [""], "require_field_match": true, "fields": { "title": {}, "content": {} } }, "min_score": 3 }</pre>
min_score : 最低相关性得分

4.5 复杂使用案例

4.5.1 复杂查询 - term + match + sort + 高亮 + 查询权重 + 去除低分

我们要查在中国创作上映的电影名称中和电影简述中可能包含地球关键词且做上映年份倒序，同时去除干扰数据并对关键词高亮的DSL。需要对关键词做权重分数处理			
地址前缀：	http://175.6.147.66:9200/	地址后缀：	film/_search
方式：	POST		
<pre>{ "explain": true, "query": { "bool": { "must": [{ "term": { "country.keyword": "中国" } }, { "match": { "title": { "query": "地球", "boost": 25 } } }, { "match": { "content": { "query": "地球", "boost": 5 } } }], "must_not": [], "should": [] } }, "from": 0, "size": 10, "sort": [{ "year": { "order": "desc" } }] }</pre>			

<pre> }, { "_score": { "order": "desc" } }], "highlight": { "pre_tags": [""], "post_tags": [""], "require_field_match": true, "fields": { "title": {}, "content": {} } }, "min_score": 3 } </pre>
min_score：最低相关性得分
本次采集有特意提到了权重该如何分配的问题。 1：权重是一个相对业务很重的选项，以真材实价网为例，用户查询任何关键词，我们必定优先命中材料名称，其次才是去命中规格型号，假设两者权重相等，造成的结果就是跟搜索给的结果用户不满意。 2：因为材料名称是基本是中文，而规格型号是数字、英文、中文、特殊字符组成。从分词器可得知，材料名称分词后效果很好，而规格型号分词后的结果是相对繁杂无章。 3：依照惯例、材料名称一般是用户输入的简称或者全称，规格型号只是某一个或者几个，如果两者的权重一致，规格型号会干扰到中文的分词。所以权重一般以业务为核心运转，真材实价网的权重一般分配比为：材料名称 5 份、规格型号 1 份。具体再微调

4.5.2 复杂查询 - 聚合算法-信息价、综合价模式

我们要查：哪些国家在哪些年份有电影。 聚合的根本逻辑：一定要是类似 SQL 里面可以做多层归类分组的字段。			
地址前缀：	http://175.6.147.66:9200/	地址后缀：	film/_search
方式：	POST		
<pre> { "query": { "bool": { "must": [], "must_not": [], "should": [] } }, "from": 0, "size": 10, "aggs": { "country": { "terms": { "field": "country.keyword", "order": { "_key": "asc" } } }, "aggregations": { "year": { "terms": { "field": "year", </pre>			

<pre> "order": { "_key": "asc" } } } }, "sort": [], "highlight": {} } } </pre>
一定记住：聚合查询也是可以跟条件一起走的，那么结果是它也是会返回列表数据。 因此：聚合查询的结果是，列表返回 + 聚合结果返回。

4.5.3 复杂查询 - 聚合算法-购物网站模式

我们要查：哪些国家在哪些年份有电影。 聚合的根本逻辑：一定要是类似 SQL 里面可以做多层归类分组的字段。			
地址前缀：	http://175.6.147.66:9200/	地址后缀：	film/_search
方式：	POST		
<pre> { "query": { "bool": { "must": [], "must_not": [], "should": [] } }, "from": 0, "size": 10, "aggs": { "country": { "terms": { "field": "country.keyword", "order": { "_key": "asc" } } }, "year": { "terms": { "field": "year", "order": { "_key": "asc" } } } }, "sort": [], "highlight": {} } </pre>			
一定记住：聚合查询也是可以跟条件一起走的，那么结果是它也是会返回列表数据。 因此：聚合查询的结果是，列表返回 + 聚合结果返回。			

4.5.4 复杂查询 - term + match + sort + 高亮 + 查询权重 + 去除低分 + 聚合

我们要查在中国创作上映的电影名称中和电影简述中可能包含地球关键词且做上映年份倒序，同时去除

干扰数据并对关键词高亮的DSL。需要对关键词做权重分数处理，结果还需要知道在上述条件下：**中国**在**哪几个年份**有**这些信息**。

地址前缀：	http://175.6.147.66:9200/	地址后缀：	film/_search
方式：	POST		

```
{
  "query": {
    "bool": {
      "must": [
        {
          "term": {
            "country.keyword": "中国"
          }
        },
        {
          "match": {
            "title": {
              "query": "地球",
              "boost": 25
            }
          }
        },
        {
          "match": {
            "content": {
              "query": "地球",
              "boost": 5
            }
          }
        }
      ],
      "must_not": [],
      "should": []
    }
  },
  "from": 0,
  "size": 10,
  "aggs": {
    "country": {
      "terms": {
        "field": "country.keyword",
        "order": {
          "_key": "asc"
        }
      }
    },
    "year": {
      "terms": {
        "field": "year",
        "order": {
          "_key": "asc"
        }
      }
    }
  },
  "sort": [
    {
      "year": {
        "order": "desc"
      }
    },
    {
      "_score": {
        "order": "desc"
      }
    }
  ]
}
```

```
}  
],  
"highlight": {  
  "pre_tags": [  
    "<span style='color:red'>"  
  ],  
  "post_tags": [  
    "</span>"  
  ],  
  "require_field_match": true,  
  "fields": {  
    "title": {},  
    "content": {}  
  }  
},  
"min_score": 3  
}
```