基于词项和基于全文的搜索

课程demo

```
DELETE products
PUT products
{
  "settings": {
    "number_of_shards": 1
  }
}


POST /products/_bulk
{ "index": { "_id": 1 }}
{ "productID" : "XHDK-A-1293-#fJ3","desc":"iPhone" }
{ "index": { "_id": 2 }}
{ "productID" : "KDKE-B-9947-#kL5","desc":"iPad" }
{ "index": { "_id": 3 }}
{ "productID" : "JODL-X-1937-#pV7","desc":"MBP" }

GET /products

POST /products/_search
{
  "query": {
    "term": {
      "desc": {
        //"value": "iPhone"
        "value":"iphone"
      }
    }
  }
}

POST /products/_search
{
  "query": {
    "term": {
      "desc.keyword": {
        //"value": "iPhone"
        //"value":"iphone"
      }
    }
  }
}


POST /products/_search
{
  "query": {
    "term": {
      "productID": {
        "value": "XHDK-A-1293-#fJ3"
      }
    }
  }
}

POST /products/_search
{
  //"explain": true,
  "query": {
    "term": {
      "productID.keyword": {
        "value": "XHDK-A-1293-#fJ3"
      }
    }
  }
}




POST /products/_search
{
  "explain": true,
  "query": {
    "constant_score": {
      "filter": {
        "term": {
          "productID.keyword": "XHDK-A-1293-#fJ3"
        }
      }

    }
  }
}


#设置 position_increment_gap
DELETE groups
PUT groups
{
  "mappings": {
    "properties": {
      "names":{
        "type": "text",
        "position_increment_gap": 0
      }
    }
  }
}

GET groups/_mapping

POST groups/_doc
{
  "names": [ "John Water", "Water Smith"]
}

POST groups/_search
```

```
{
  "query": {
    "match_phrase": {
      "names": {
        "query": "Water Water",
        "slop": 100
      }
    }
  }
}


POST groups/_search
{
  "query": {
    "match_phrase": {
      "names": "Water Smith"
    }
  }
}
```

相关阅读

- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/term-level-queries.html
- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/full-text-queries.html

结构化搜索

课程demo

```

#结构化搜索，精确匹配
DELETE products
POST /products/_bulk
{ "index": { "_id": 1 }}
{ "price" : 10,"avaliable":true,"date":"2018-01-01", "productID" : "XHDK-A-1293-#fJ3" }
{ "index": { "_id": 2 }}
{ "price" : 20,"avaliable":true,"date":"2019-01-01", "productID" : "KDKE-B-9947-#kL5" }
{ "index": { "_id": 3 }}
{ "price" : 30,"avaliable":true, "productID" : "JODL-X-1937-#pV7" }
{ "index": { "_id": 4 }}
{ "price" : 30,"avaliable":false, "productID" : "QQPX-R-3956-#aD8" }

GET products/_mapping



#对布尔值 match 查询，有算分
POST products/_search
{
  "profile": "true",
  "explain": true,
  "query": {
    "term": {
      "avaliable": true
    }
  }
}



#对布尔值，通过constant score 转成 filtering，没有算分
POST products/_search
{
  "profile": "true",
  "explain": true,
  "query": {
    "constant_score": {
      "filter": {
        "term": {
          "avaliable": true
        }
      }
    }
  }
}


#数字类型 Term
POST products/_search
{
  "profile": "true",
  "explain": true,
  "query": {
    "term": {
      "price": 30
    }
  }
}

#数字类型 terms
POST products/_search
{
  "query": {
    "constant_score": {
      "filter": {
        "terms": {
          "price": [
            "20",
            "30"
          ]
        }
      }
    }
  }
}

#数字 Range 查询
GET products/_search
{
    "query" : {
        "constant_score" : {
            "filter" : {
                "range" : {
                    "price" : {
                        "gte" : 20,
                        "lte"  : 30
                    }
                }
            }
        }
    }
}


# 日期 range
POST products/_search
{
    "query" : {
        "constant_score" : {
            "filter" : {
                "range" : {
                    "date" : {
                      "gte" : "now-1y"
                    }
                }
            }
        }
    }
}



#exists查询
POST products/_search
{
  "query": {
    "constant_score": {
      "filter": {
        "exists": {
          "field": "date"
        }
      }
    }
  }
}

#处理多值字段
POST /movies/_bulk
{ "index": { "_id": 1 }}
{ "title" : "Father of the Bridge Part II","year":1995, "genre":"Comedy"}
{ "index": { "_id": 2 }}
{ "title" : "Dave","year":1993,"genre":["Comedy","Romance"] }


#处理多值字段，term 查询是包含，而不是等于
POST movies/_search
{
  "query": {
    "constant_score": {
      "filter": {
        "term": {
          "genre.keyword": "Comedy"
        }
      }
    }
  }
}


#字符类型 terms
POST products/_search
{
  "query": {
    "constant_score": {
      "filter": {
        "terms": {
          "productID.keyword": [
            "QQPX-R-3956-#aD8",
            "JODL-X-1937-#pV7"
          ]
        }
      }
    }
  }
}



POST products/_search
{
  "profile": "true",
  "explain": true,
  "query": {
    "match": {
      "price": 30
    }
  }
}


POST products/_search
{
  "profile": "true",
  "explain": true,
  "query": {
    "term": {
      "date": "2019-01-01"
    }
  }
}

POST products/_search
{
  "profile": "true",
  "explain": true,
  "query": {
    "match": {
      "date": "2019-01-01"
    }
  }
}




POST products/_search
{
  "profile": "true",
  "explain": true,
  "query": {
    "constant_score": {
      "filter": {
        "term": {
          "productID.keyword": "XHDK-A-1293-#fJ3"
        }
      }
    }
  }
}

POST products/_search
{
  "profile": "true",
  "explain": true,
  "query": {
    "term": {
      "productID.keyword": "XHDK-A-1293-#fJ3"
    }
  }
}

#对布尔数值
POST products/_search
{
  "query": {
    "constant_score": {
      "filter": {
        "term": {
          "avaliable": "false"
        }
      }
    }
  }
}

POST products/_search
{
  "query": {
    "term": {
      "avaliable": {
        "value": "false"
      }
    }
  }
}
```

```
POST products/_search
{
  "profile": "true",
  "explain": true,
  "query": {
    "term": {
      "price": {
        "value": "20"
      }
    }
  }
}

POST products/_search
{
  "profile": "true",
  "explain": true,
  "query": {
    "match": {
      "price": "20"
    }
    }
  }
}


POST products/_search
{
  "query": {
    "constant_score": {
      "filter": {
        "bool": {
          "must_not": {
            "exists": {
              "field": "date"
            }
          }
        }
      }
    }
  }
}
```

相关阅读

https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-exists-query.html
https://www.elastic.co/guide/en/elasticsearch/reference/7.1/term-level-queries.html

搜索的相关性算分

课程demo

```
PUT testscore
{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "properties": {
      "content": {
        "type": "text"
      }
    }
  }
}


PUT testscore/_bulk
{ "index": { "_id": 1 }}
{ "content":"we use Elasticsearch to power the search" }
{ "index": { "_id": 2 }}
{ "content":"we like elasticsearch" }
{ "index": { "_id": 3 }}
{ "content":"The scoring of documents is caculated by the scoring formula" }
{ "index": { "_id": 4 }}
{ "content":"you know, for search" }



POST /testscore/_search
{
  //"explain": true,
  "query": {
    "match": {
      "content":"you"
      //"content": "elasticsearch"
      //"content":"the"
      //"content": "the elasticsearch"
    }
  }
}

POST testscore/_search
{
    "query": {
        "boosting" : {
            "positive" : {
                "term" : {
                    "content" : "elasticsearch"
                }
            },
            "negative" : {
                 "term" : {
                     "content" : "like"
                }
            },
            "negative_boost" : 0.2
        }
    }
}


POST tmdb/_search
{
  "_source": ["title","overview"],
  "query": {
    "more_like_this": {
      "fields": [
        "title^10","overview"
      ],
      "like": [{"_id":"14191"}],
      "min_term_freq": 1,
      "max_query_terms": 12
    }
  }
}
```

Query & Filtering 与多字符串多字段查询

课程demo

```
POST /products/_bulk
{ "index": { "_id": 1 }}
{ "price" : 10,"avaliable":true,"date":"2018-01-01", "productID" : "XHDK-A-1293-#fJ3" }
{ "index": { "_id": 2 }}
{ "price" : 20,"avaliable":true,"date":"2019-01-01", "productID" : "KDKE-B-9947-#kL5" }
{ "index": { "_id": 3 }}
{ "price" : 30,"avaliable":true, "productID" : "JODL-X-1937-#pV7" }
{ "index": { "_id": 4 }}
{ "price" : 30,"avaliable":false, "productID" : "QQPX-R-3956-#aD8" }



#基本语法
POST /products/_search
{
  "query": {
    "bool" : {
      "must" : {
        "term" : { "price" : "30" }
      },
      "filter": {
        "term" : { "avaliable" : "true" }
      },
      "must_not" : {
        "range" : {
          "price" : { "lte" : 10 }
        }
      },
      "should" : [
        { "term" : { "productID.keyword" : "JODL-X-1937-#pV7" } },
        { "term" : { "productID.keyword" : "XHDK-A-1293-#fJ3" } }
      ],
      "minimum_should_match" :1
    }
  }
}

#改变数据模型，增加字段。解决数组包含而不是精确匹配的问题
POST /newmovies/_bulk
{ "index": { "_id": 1 }}
{ "title" : "Father of the Bridge Part II","year":1995,
"genre":"Comedy","genre_count":1 }
{ "index": { "_id": 2 }}
{ "title" : "Dave","year":1993,"genre":["Comedy","Romance"],"genre_count":2 }

#must，有算分
POST /newmovies/_search
{
  "query": {
    "bool": {
      "must": [
        {"term": {"genre.keyword": {"value": "Comedy"}}},
        {"term": {"genre_count": {"value": 1}}}

      ]
    }
  }
}

#Filter。不参与算分，结果的score是0
POST /newmovies/_search
{
  "query": {
    "bool": {
      "filter": [
        {"term": {"genre.keyword": {"value": "Comedy"}}},
        {"term": {"genre_count": {"value": 1}}}
        ]

    }
  }
}


#Filtering Context
POST _search
{
  "query": {
    "bool" : {

      "filter": {
        "term" : { "avaliable" : "true" }
      },
      "must_not" : {
        "range" : {
          "price" : { "lte" : 10 }
        }
      }
    }
  }
}


#Query Context
POST /products/_bulk
{ "index": { "_id": 1 }}
{ "price" : 10,"avaliable":true,"date":"2018-01-01", "productID" : "XHDK-A-1293-#fJ3" }
{ "index": { "_id": 2 }}
{ "price" : 20,"avaliable":true,"date":"2019-01-01", "productID" : "KDKE-B-9947-#kL5" }
{ "index": { "_id": 3 }}
{ "price" : 30,"avaliable":true, "productID" : "JODL-X-1937-#pV7" }
{ "index": { "_id": 4 }}
{ "price" : 30,"avaliable":false, "productID" : "QQPX-R-3956-#aD8" }


POST /products/_search
{
  "query": {
    "bool": {
      "should": [
        {
          "term": {
            "productID.keyword": {
              "value": "JODL-X-1937-#pV7"}}
        },
        {"term": {"avaliable": {"value": true}}
        }
      ]
    }
  }
}


#嵌套，实现了 should not 逻辑
POST /products/_search
{
  "query": {
    "bool": {
      "must": {
        "term": {
          "price": "30"
        }
      },
      "should": [
        {
          "bool": {
            "must_not": {
              "term": {
                "avaliable": "false"
              }
            }
          }
        }
      ],
      "minimum_should_match": 1
    }
  }
}


#Controll the Precision
POST _search
{
  "query": {
    "bool" : {
      "must" : {
        "term" : { "price" : "30" }
      },
      "filter": {
        "term" : { "avaliable" : "true" }
      },
      "must_not" : {
        "range" : {
          "price" : { "lte" : 10 }
        }
      },
      "should" : [
        { "term" : { "productID.keyword" : "JODL-X-1937-#pV7" } },
        { "term" : { "productID.keyword" : "XHDK-A-1293-#fJ3" } }
      ],
      "minimum_should_match" :2
    }
  }
}



POST /animals/_search
{
  "query": {
    "bool": {
      "should": [
        { "term": { "text": "brown" }},
        { "term": { "text": "red" }},
        { "term": { "text": "quick"   }},
        { "term": { "text": "dog"   }}
      ]
    }
  }
}

POST /animals/_search
{
  "query": {
    "bool": {
      "should": [
        { "term": { "text": "quick" }},
        { "term": { "text": "dog"   }},
        {
          "bool":{
            "should":[
               { "term": { "text": "brown" }},
                 { "term": { "text": "brown" }},
            ]
          }

        }
      ]
    }
  }
}


DELETE blogs
POST /blogs/_bulk
{ "index": { "_id": 1 }}
{"title":"Apple iPad", "content":"Apple iPad,Apple iPad" }
{ "index": { "_id": 2 }}
{"title":"Apple iPad,Apple iPad", "content":"Apple iPad" }


POST blogs/_search
{
  "query": {
    "bool": {
      "should": [
        {"match": {
          "title": {
            "query": "apple,ipad",
            "boost": 1.1
          }
        }},

        {"match": {
          "content": {
            "query": "apple,ipad",
            "boost":
          }
        }}
      ]
    }
  }
}

DELETE news
POST /news/_bulk
{ "index": { "_id": 1 }}
{ "content":"Apple Mac" }
{ "index": { "_id": 2 }}
{ "content":"Apple iPad" }
{ "index": { "_id": 3 }}
{ "content":"Apple employee like Apple Pie and Apple Juice" }


POST news/_search
{
  "query": {
    "bool": {
      "must": {
        "match":{"content":"apple"}
      }
    }
  }
}

POST news/_search
{
  "query": {
    "bool": {
      "must": {
        "match":{"content":"apple"}
      },
      "must_not": {
        "match":{"content":"pie"}
      }
    }
  }
}

POST news/_search
{
  "query": {
    "boosting": {
      "positive": {
        "match": {
          "content": "apple"
        }
      },
      "negative": {
        "match": {
          "content": "pie"
        }
      },
      "negative_boost": 0.5
    }
  }
}
```

相关阅读

- https://www.elastic.co/guide/en/elasticsearch/reference/current/query-filter-context.html
- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/query-dsl-boosting-query.html

单字符串多字段查询：Dis Max Query

课程demo

```
PUT /blogs/_doc/1
{
    "title": "Quick brown rabbits",
    "body":  "Brown rabbits are commonly seen."
}
```

```
PUT /blogs/_doc/2
{
    "title": "Keeping pets healthy",
    "body":  "My quick brown fox eats rabbits on a regular basis."
}

POST /blogs/_search
{
    "query": {
        "bool": {
            "should": [
                { "match": { "title": "Brown fox" }},
                { "match": { "body":  "Brown fox" }}
            ]
        }
    }
}

POST blogs/_search
{
    "query": {
        "dis_max": {
            "queries": [
                { "match": { "title": "Quick pets" }},
                { "match": { "body":  "Quick pets" }}
            ]
        }
    }
}


POST blogs/_search
{
    "query": {
        "dis_max": {
            "queries": [
                { "match": { "title": "Quick pets" }},
                { "match": { "body":  "Quick pets" }}
            ],
            "tie_breaker": 0.2
        }
    }
}
```

相关阅读

- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/query-dsl-dis-max-query.html

单字符串多字段查询：Multi Match

课程demo

```
POST blogs/_search
{
    "query": {
        "dis_max": {
            "queries": [
                { "match": { "title": "Quick pets" }},
                { "match": { "body":  "Quick pets" }}
            ],
            "tie_breaker": 0.2
        }
    }
}

POST blogs/_search
{
  "query": {
    "multi_match": {
      "type": "best_fields",
      "query": "Quick pets",
      "fields": ["title","body"],
      "tie_breaker": 0.2,
      "minimum_should_match": "20%"
    }
  }
}



POST books/_search
{
    "multi_match": {
        "query":  "Quick brown fox",
        "fields": "*_title"
    }
}


POST books/_search
{
    "multi_match": {
        "query":  "Quick brown fox",
        "fields": [ "*_title", "chapter_title^2" ]
    }
}



DELETE /titles
PUT /titles
{
    "settings": { "number_of_shards": 1 },
    "mappings": {
        "my_type": {
            "properties": {
                "title": {
                    "type":     "string",
                    "analyzer": "english",
                    "fields": {
                        "std":   {
                            "type":     "string",
                            "analyzer": "standard"
                        }
                    }
                }
            }
        }
    }
}

PUT /titles
{
  "mappings": {
    "properties": {
      "title": {
        "type": "text",
        "analyzer": "english"
      }
    }
  }
}

POST titles/_bulk
{ "index": { "_id": 1 }}
{ "title": "My dog barks" }
{ "index": { "_id": 2 }}
{ "title": "I see a lot of barking dogs on the road " }


GET titles/_search
{
  "query": {
    "match": {
      "title": "barking dogs"
    }
  }
}

DELETE /titles
PUT /titles
{
  "mappings": {
    "properties": {
      "title": {
        "type": "text",
        "analyzer": "english",
        "fields": {"std": {"type": "text","analyzer": "standard"}}
      }
    }
  }
}

POST titles/_bulk
{ "index": { "_id": 1 }}
{ "title": "My dog barks" }
{ "index": { "_id": 2 }}
{ "title": "I see a lot of barking dogs on the road " }

GET /titles/_search
```

```
{
   "query": {
        "multi_match": {
            "query":  "barking dogs",
            "type":   "most_fields",
            "fields": [ "title", "title.std" ]
        }
    }
}

GET /titles/_search
{
   "query": {
        "multi_match": {
            "query":  "barking dogs",
            "type":   "most_fields",
            "fields": [ "title^10", "title.std" ]
        }
    }
}
```

相关阅读

- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/query-dsl-dis-max-query.html

多语言及中文分词与检索

课程demo

- 来到杨过曾经生活过的地方，小龙女动情地说：“我也想过过过儿过过的生活。”
- 你也想犯范范玮琪犯过的错吗
- 校长说衣服上除了校徽别别别的
- 这几天天天天气不好
- 我背有点驼，麻麻说“你的背得背背背背佳

```
#stop word

DELETE my_index
PUT /my_index/_doc/1
{ "title": "I'm happy for this fox" }

PUT /my_index/_doc/2
{ "title": "I'm not happy about my fox problem" }


POST my_index/_search
{
  "query": {
    "match": {
      "title": "not happy fox"
    }
  }
}
```

```
#虽然通过使用 english （英语）分析器，使得匹配规则更加宽松，我们也因此提高了召回率，但却降低了
精准匹配文档的能力。为了获得两方面的优势，我们可以使用multifields（多字段）对 title 字段建立
两次索引： 一次使用 `english`（英语）分析器，另一次使用 `standard`（标准）分析器:

DELETE my_index

PUT /my_index
{
  "mappings": {
    "blog": {
      "properties": {
        "title": {
          "type": "string",
          "analyzer": "english"
        }
      }
    }
  }
}

PUT /my_index
{
  "mappings": {
    "blog": {
      "properties": {
        "title": {
          "type": "string",
          "fields": {
            "english": {
              "type":     "string",
              "analyzer": "english"
            }
          }
        }
      }
    }
  }
}


PUT /my_index/blog/1
{ "title": "I'm happy for this fox" }

PUT /my_index/blog/2
{ "title": "I'm not happy about my fox problem" }

GET /_search
{
  "query": {
    "multi_match": {
      "type":     "most_fields",
      "query":    "not happy foxes",
      "fields": [ "title", "title.english" ]
    }
  }
}
```

```
#安装插件
./elasticsearch-plugin install https://github.com/medcl/elasticsearch-analysis-ik/releases/download/v7.1.0/elasticsearch-analysis-ik-7.1.0.zip
#安装插件
bin/elasticsearch install https://github.com/KennFalcon/elasticsearch-analysis-hanlp/releases/download/v7.1.0/elasticsearch-analysis-hanlp-7.1.0.zip




#ik_max_word
#ik_smart
#hanlp: hanlp默认分词
#hanlp_standard: 标准分词
#hanlp_index: 索引分词
#hanlp_nlp: NLP分词
#hanlp_n_short: N-最短路分词
#hanlp_dijkstra: 最短路分词
#hanlp_crf: CRF分词（在hanlp 1.6.6已开始废弃）
#hanlp_speed: 极速词典分词

POST _analyze
{
  "analyzer": "hanlp_standard",
  "text": ["剑桥分析公司多位高管对卧底记者说，他们确保了唐纳德·特朗普在总统大选中获胜"]

}

#Pinyin
PUT /artists/
{
    "settings" : {
        "analysis" : {
            "analyzer" : {
                "user_name_analyzer" : {
                    "tokenizer" : "whitespace",
                    "filter" : "pinyin_first_letter_and_full_pinyin_filter"
                }
            },
            "filter" : {
                "pinyin_first_letter_and_full_pinyin_filter" : {
                    "type" : "pinyin",
                    "keep_first_letter" : true,
                    "keep_full_pinyin" : false,
                    "keep_none_chinese" : true,
                    "keep_original" : false,
                    "limit_first_letter_length" : 16,
                    "lowercase" : true,
                    "trim_whitespace" : true,
                    "keep_none_chinese_in_first_letter" : true
                }
            }
        }
    }
}
```

```
GET /artists/_analyze
{
  "text": ["刘德华 张学友 郭富城 黎明 四大天王"],
  "analyzer": "user_name_analyzer"
}
```

相关资源

- Elasticsearch IK分词插件 https://github.com/medcl/elasticsearch-analysis-ik/releases
- Elasticsearch hanlp 分词插件 https://github.com/KennFalcon/elasticsearch-analysis-hanlp
- 分词算法综述 https://zhuanlan.zhihu.com/p/50444885

一些分词工具，供参考：

- 中科院计算所NLPIR http://ictclas.nlpir.org/nlpir/
- ansj分词器 https://github.com/NLPchina/ansj_seg
- 哈工大的LTP https://github.com/HIT-SCIR/ltp
- 清华大学THULAC https://github.com/thunlp/THULAC
- 斯坦福分词器 https://nlp.stanford.edu/software/segmenter.shtml
- Hanlp分词器 https://github.com/hankcs/HanLP
- 结巴分词 https://github.com/yanyiwu/cppjieba
- KCWS分词器(字嵌入+Bi-LSTM+CRF) https://github.com/koth/kcws
- ZPar https://github.com/frcchang/zpar/releases
- IKAnalyzer https://github.com/wks/ik-analyzer

Space Jam，一次全文搜索的实例

环境要求

- Python 2.7.15
- 可以使用pyenv管理多个python版本（可选）

进入 tmdb-search目录

Run
pip install -r requirements.txt
Run python ./ingest_tmdb_from_file.py

课程demo

```
POST tmdb/_search
{
"_source": ["title","overview"],
 "query": {
   "match_all": {}
 }
}

POST tmdb/_search
{
  "_source": ["title","overview"],
  "query": {
    "multi_match": {
      "query": "basketball with cartoon aliens",
      "fields": ["title","overview"]
    }
  },
  "highlight" : {
        "fields" : {
            "overview" : { "pre_tags" : ["\\033[0;32;40m"], "post_tags" :
["\\033[0m"] },
            "title" : { "pre_tags" : ["\\033[0;32;40m"], "post_tags" :
["\\033[0m"] }

        }
    }
}
```

相关

- Windows 安装 pyenv https://github.com/pyenv-win/pyenv-win
- Mac 安装pyenv https://segmentfault.com/a/1190000017403221
- Linux 安装 pyenv https://blog.csdn.net/GX_1_11_real/article/details/80237064
- Python.org https://www.python.org/

使用 Search Template 和 Index Alias 查询

课程Demo

```
POST _scripts/tmdb
{
  "script": {
    "lang": "mustache",
    "source": {
      "_source": [
        "title","overview"
      ],
      "size": 20,
      "query": {
        "multi_match": {
          "query": "{{q}}",
          "fields": ["title","overview"]
        }
      }
    }
  }
}
DELETE _scripts/tmdb

GET _scripts/tmdb

POST tmdb/_search/template
{
    "id":"tmdb",
    "params": {
        "q": "basketball with cartoon aliens"
    }
```

```
}


PUT movies-2019/_doc/1
{
  "name":"the matrix",
  "rating":5
}

PUT movies-2019/_doc/2
{
  "name":"Speed",
  "rating":3
}

POST _aliases
{
  "actions": [
    {
      "add": {
        "index": "movies-2019",
        "alias": "movies-latest"
      }
    }
  ]
}

POST movies-latest/_search
{
  "query": {
    "match_all": {}
  }
}

POST _aliases
{
  "actions": [
    {
      "add": {
        "index": "movies-2019",
        "alias": "movies-lastest-highrate",
        "filter": {
          "range": {
            "rating": {
              "gte": 4
            }
          }
        }
      }
    }
  ]
}

POST movies-lastest-highrate/_search
{
  "query": {
    "match_all": {}
  }
```

```
}
```

综合排序：Function Score Query 优化算分

课程Demo

```
DELETE blogs
PUT /blogs/_doc/1
{
  "title":   "About popularity",
  "content": "In this post we will talk about...",
  "votes":   0
}

PUT /blogs/_doc/2
{
  "title":   "About popularity",
  "content": "In this post we will talk about...",
  "votes":   100
}

PUT /blogs/_doc/3
{
  "title":   "About popularity",
  "content": "In this post we will talk about...",
  "votes":   1000000
}


POST /blogs/_search
{
  "query": {
    "function_score": {
      "query": {
        "multi_match": {
          "query":    "popularity",
          "fields": [ "title", "content" ]
        }
      },
      "field_value_factor": {
        "field": "votes"
      }
    }
  }
}

POST /blogs/_search
{
  "query": {
    "function_score": {
      "query": {
        "multi_match": {
          "query":    "popularity",
          "fields": [ "title", "content" ]
        }
      },
      "field_value_factor": {
        "field": "votes",
        "modifier": "log1p"
      }
    }
  }
}


POST /blogs/_search
{
  "query": {
    "function_score": {
      "query": {
        "multi_match": {
          "query":    "popularity",
          "fields": [ "title", "content" ]
        }
      },
      "field_value_factor": {
        "field": "votes",
        "modifier": "log1p" ,
        "factor": 0.1
      }
    }
  }
}


POST /blogs/_search
{
  "query": {
    "function_score": {
      "query": {
        "multi_match": {
          "query":    "popularity",
          "fields": [ "title", "content" ]
        }
      },
      "field_value_factor": {
        "field": "votes",
        "modifier": "log1p" ,
        "factor": 0.1
      },
      "boost_mode": "sum",
      "max_boost": 3
    }
  }
}

POST /blogs/_search
{
  "query": {
    "function_score": {
      "random_score": {
        "seed": 911119
      }
    }
  }
}
```

相关阅读

- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/query-dsl-function-score-query.html

Term & Phrase Suggester

课程Demo

```
DELETE articles
PUT articles
{
  "mappings": {
    "properties": {
      "title_completion":{
        "type": "completion"
      }
    }
  }
}

POST articles/_bulk
{ "index" : { } }
{ "title_completion": "lucene is very cool"}
{ "index" : { } }
{ "title_completion": "Elasticsearch builds on top of lucene"}
{ "index" : { } }
{ "title_completion": "Elasticsearch rocks"}
{ "index" : { } }
{ "title_completion": "elastic is the company behind ELK stack"}
{ "index" : { } }
{ "title_completion": "Elk stack rocks"}
{ "index" : {} }


POST articles/_search?pretty
{
  "size": 0,
  "suggest": {
    "article-suggester": {
      "prefix": "elk ",
      "completion": {
        "field": "title_completion"
      }
    }
  }
}

DELETE articles
```

```
POST articles/_bulk
{ "index" : { } }
{ "body": "lucene is very cool"}
{ "index" : { } }
{ "body": "Elasticsearch builds on top of lucene"}
{ "index" : { } }
{ "body": "Elasticsearch rocks"}
{ "index" : { } }
{ "body": "elastic is the company behind ELK stack"}
{ "index" : { } }
{ "body": "Elk stack rocks"}
{ "index" : {} }
{  "body": "elasticsearch is rock solid"}


POST _analyze
{
  "analyzer": "standard",
  "text": ["Elk stack  rocks rock"]
}

POST /articles/_search
{
  "size": 1,
  "query": {
    "match": {
      "body": "lucen rock"
    }
  },
  "suggest": {
    "term-suggestion": {
      "text": "lucen rock",
      "term": {
        "suggest_mode": "missing",
        "field": "body"
      }
    }
  }
}


POST /articles/_search
{

  "suggest": {
    "term-suggestion": {
      "text": "lucen rock",
      "term": {
        "suggest_mode": "popular",
        "field": "body"
      }
    }
  }
}


POST /articles/_search
{
  "suggest": {
    "term-suggestion": {
      "text": "lucen rock",
      "term": {
        "suggest_mode": "always",
        "field": "body",
      }
    }
  }
}


POST /articles/_search
{

  "suggest": {
    "term-suggestion": {
      "text": "lucen hocks",
      "term": {
        "suggest_mode": "always",
        "field": "body",
        "prefix_length":0,
        "sort": "frequency"
      }
    }
  }
}


POST /articles/_search
{
  "suggest": {
    "my-suggestion": {
      "text": "lucne and elasticsear rock hello world ",
      "phrase": {
        "field": "body",
        "max_errors":2,
        "confidence":0,
        "direct_generator":[{
          "field":"body",
          "suggest_mode":"always"
        }],
        "highlight": {
          "pre_tag": "<em>",
          "post_tag": "</em>"
        }
      }
    }
  }
}
```

相关阅读

自动补全与基于上下文的提示

课程demo

```
DELETE articles
PUT articles
{
  "mappings": {
    "properties": {
      "title_completion":{
        "type": "completion"
      }
    }
  }
}

POST articles/_bulk
{ "index" : { } }
{ "title_completion": "lucene is very cool"}
{ "index" : { } }
{ "title_completion": "Elasticsearch builds on top of lucene"}
{ "index" : { } }
{ "title_completion": "Elasticsearch rocks"}
{ "index" : { } }
{ "title_completion": "elastic is the company behind ELK stack"}
{ "index" : { } }
{ "title_completion": "Elk stack rocks"}
{ "index" : {} }


POST articles/_search?pretty
{
  "size": 0,
  "suggest": {
    "article-suggester": {
      "prefix": "elk ",
      "completion": {
        "field": "title_completion"
      }
    }
  }
}


DELETE comments
PUT comments
PUT comments/_mapping
{
  "properties": {
    "comment_autocomplete":{
      "type": "completion",
      "contexts":[{
        "type":"category",
        "name":"comment_category"
      }]
    }
  }
}
```

```
POST comments/_doc
{
  "comment":"I love the star war movies",
  "comment_autocomplete":{
    "input":["star wars"],
    "contexts":{
      "comment_category":"movies"
    }
  }
}

POST comments/_doc
{
  "comment":"Where can I find a Starbucks",
  "comment_autocomplete":{
    "input":["starbucks"],
    "contexts":{
      "comment_category":"coffee"
    }
  }
}


POST comments/_search
{
  "suggest": {
    "MY_SUGGESTION": {
      "prefix": "sta",
      "completion":{
        "field":"comment_autocomplete",
        "contexts":{
          "comment_category":"coffee"
        }
      }
    }
  }
}
```

跨集群搜索

课程demo

```
//启动3个集群

bin/elasticsearch -E node.name=cluster0node -E cluster.name=cluster0 -E
path.data=cluster0_data -E discovery.type=single-node -E http.port=9200 -E
transport.port=9300
bin/elasticsearch -E node.name=cluster1node -E cluster.name=cluster1 -E
path.data=cluster1_data -E discovery.type=single-node -E http.port=9201 -E
transport.port=9301
bin/elasticsearch -E node.name=cluster2node -E cluster.name=cluster2 -E
path.data=cluster2_data -E discovery.type=single-node -E http.port=9202 -E
transport.port=9302


//在每个集群上设置动态的设置
```

```
PUT _cluster/settings
{
  "persistent": {
    "cluster": {
      "remote": {
        "cluster0": {
          "seeds": [
            "127.0.0.1:9300"
          ],
          "transport.ping_schedule": "30s"
        },
        "cluster1": {
          "seeds": [
            "127.0.0.1:9301"
          ],
          "transport.compress": true,
          "skip_unavailable": true
        },
        "cluster2": {
          "seeds": [
            "127.0.0.1:9302"
          ]
        }
      }
    }
  }
}

#cURL
curl -XPUT "http://localhost:9200/_cluster/settings" -H 'Content-Type:
application/json' -d'
{"persistent":{"cluster":{"remote":{"cluster0":{"seeds":
["127.0.0.1:9300"],"transport.ping_schedule":"30s"},"cluster1":{"seeds":
["127.0.0.1:9301"],"transport.compress":true,"skip_unavailable":true},"cluster2":
{"seeds":["127.0.0.1:9302"]}}}}}'

curl -XPUT "http://localhost:9201/_cluster/settings" -H 'Content-Type:
application/json' -d'
{"persistent":{"cluster":{"remote":{"cluster0":{"seeds":
["127.0.0.1:9300"],"transport.ping_schedule":"30s"},"cluster1":{"seeds":
["127.0.0.1:9301"],"transport.compress":true,"skip_unavailable":true},"cluster2":
{"seeds":["127.0.0.1:9302"]}}}}}'

curl -XPUT "http://localhost:9202/_cluster/settings" -H 'Content-Type:
application/json' -d'
{"persistent":{"cluster":{"remote":{"cluster0":{"seeds":
["127.0.0.1:9300"],"transport.ping_schedule":"30s"},"cluster1":{"seeds":
["127.0.0.1:9301"],"transport.compress":true,"skip_unavailable":true},"cluster2":
{"seeds":["127.0.0.1:9302"]}}}}}'


#创建测试数据
curl -XPOST "http://localhost:9200/users/_doc" -H 'Content-Type:
application/json' -d'
{"name":"user1","age":10}'

curl -XPOST "http://localhost:9201/users/_doc" -H 'Content-Type:
application/json' -d'
```

```
{"name":"user2","age":20}'

curl -XPOST "http://localhost:9202/users/_doc" -H 'Content-Type:
application/json' -d'
{"name":"user3","age":30}'


#查询
GET /users,cluster1:users,cluster2:users/_search
{
  "query": {
    "range": {
      "age": {
        "gte": 20,
        "lte": 40
      }
    }
  }
}
```

相关阅读

- 在Kibana中使用Cross data search https://kelonsoftware.com/cross-cluster-search-kibana/

集群分布式模型及选主与脑裂问题

课程Demo

```
bin/elasticsearch -E node.name=node1 -E cluster.name=geektime -E
path.data=node1_data
bin/elasticsearch -E node.name=node2 -E cluster.name=geektime -E
path.data=node2_data
bin/elasticsearch -E node.name=node3 -E cluster.name=geektime -E
path.data=node3_data
```

相关阅读

- https://www.elastic.co/cn/blog/a-new-era-for-cluster-coordination-in-elasticsearch

分片与集群的故障转移

文档分布式存储

分片及其生命周期

剖析分布式查询及相关性评分

课程demo

```
DELETE message
PUT message
{
  "settings": {
    "number_of_shards": 20
  }
}

GET message

POST message/_doc?routing=1
{
  "content":"good"
}

POST message/_doc?routing=2
{
  "content":"good morning"
}

POST message/_doc?routing=3
{
  "content":"good morning everyone"
}

POST message/_search
{
  "explain": true,
  "query": {
    "match_all": {}
  }
}


POST message/_search
{
  "explain": true,
  "query": {
    "term": {
      "content": {
        "value": "good"
      }
    }
  }
}


POST message/_search?search_type=dfs_query_then_fetch
{

  "query": {
    "term": {
      "content": {
        "value": "good"
      }
    }
  }
}
```

排序及Doc Values & Fielddata

课程demo

```
#单字段排序
POST /kibana_sample_data_ecommerce/_search
{
  "size": 5,
  "query": {
    "match_all": {

    }
  },
  "sort": [
    {"order_date": {"order": "desc"}}
  ]
}

#多字段排序
POST /kibana_sample_data_ecommerce/_search
{
  "size": 5,
  "query": {
    "match_all": {

    }
  },
  "sort": [
    {"order_date": {"order": "desc"}},
    {"_doc":{"order": "asc"}},
    {"_score":{ "order": "desc"}}
  ]
}

GET kibana_sample_data_ecommerce/_mapping

#对 text 字段进行排序。默认会报错，需打开fielddata
POST /kibana_sample_data_ecommerce/_search
{
  "size": 5,
  "query": {
    "match_all": {

    }
  },
  "sort": [
    {"customer_full_name": {"order": "desc"}}
  ]
}

#打开 text的 fielddata
PUT kibana_sample_data_ecommerce/_mapping
{
  "properties": {
    "customer_full_name" : {
          "type" : "text",
          "fielddata": true,
          "fields" : {
            "keyword" : {
              "type" : "keyword",
              "ignore_above" : 256
            }
          }
        }
  }
}

#关闭 keyword的 doc values
PUT test_keyword
PUT test_keyword/_mapping
{
  "properties": {
    "user_name":{
      "type": "keyword",
      "doc_values":false
    }
  }
}

DELETE test_keyword

PUT test_text
PUT test_text/_mapping
{
  "properties": {
    "intro":{
      "type": "text",
      "doc_values":true
    }
  }
}

DELETE test_text


DELETE temp_users
PUT temp_users
PUT temp_users/_mapping
{
  "properties": {
    "name":{"type": "text","fielddata": true},
    "desc":{"type": "text","fielddata": true}
  }
}

Post temp_users/_doc
{"name":"Jack","desc":"Jack is a good boy!","age":10}

#打开fielddata 后，查看 docvalue_fields数据
POST  temp_users/_search
{
  "docvalue_fields": [
    "name","desc"
    ]
}

#查看整型字段的docvalues
POST  temp_users/_search
{
  "docvalue_fields": [
    "age"
    ]
}
```

分页与遍历 - From, Size, Search_after & Scroll API

课程Demo

```

POST tmdb/_search
{
  "from": 10000,
  "size": 1,
  "query": {
    "match_all": {

    }
  }
}

#Scroll API
DELETE users

POST users/_doc
{"name":"user1","age":10}

POST users/_doc
{"name":"user2","age":11}


POST users/_doc
{"name":"user2","age":12}

POST users/_doc
{"name":"user2","age":13}

POST users/_count

POST users/_search
{
    "size": 1,
    "query": {
        "match_all": {}
    },
    "sort": [
        {"age": "desc"} ,
        {"_id": "asc"}
    ]
}

POST users/_search
{
    "size": 1,
    "query": {
        "match_all": {}
    },
    "search_after":
        [
          10,
          "ZQ0vYGsBrR8X3IP75QqX"],
    "sort": [
        {"age": "desc"} ,
        {"_id": "asc"}
    ]
}


#Scroll API
DELETE users
POST users/_doc
{"name":"user1","age":10}

POST users/_doc
{"name":"user2","age":20}

POST users/_doc
{"name":"user3","age":30}

POST users/_doc
{"name":"user4","age":40}

POST /users/_search?scroll=5m
{
    "size": 1,
    "query": {
        "match_all" : {
        }
    }
}


POST users/_doc
{"name":"user5","age":50}
POST /_search/scroll
{
    "scroll" : "1m",
    "scroll_id" :
"DXF1ZXJ5QW5kRmV0Y2gBAAAAAAAAAWAWbWdoQXR2d3ZUd2kzSThwVTh4bVE0QQ=="
}
```

处理并发读写操作

课程DEMO

```

DELETE products
PUT products

PUT products/_doc/1
{
  "title":"iphone",
  "count":100
}



GET products/_doc/1

PUT products/_doc/1?if_seq_no=1&if_primary_term=1
{
  "title":"iphone",
  "count":100
}



PUT products/_doc/1?version=30000&version_type=external
{
  "title":"iphone",
  "count":100
}
```

Bucket & Metric Aggregation

demos

```
DELETE /employees
PUT /employees/
{
  "mappings" : {
      "properties" : {
        "age" : {
          "type" : "integer"
        },
        "gender" : {
          "type" : "keyword"
        },
        "job" : {
          "type" : "text",
          "fields" : {
            "keyword" : {
```

```
       "type" : "keyword",
       "ignore_above" : 50
      }
     }
    },
    "name" : {
     "type" : "keyword"
    },
    "salary" : {
     "type" : "integer"
    }
   }
  }
}

PUT /employees/_bulk
{ "index" : {  "_id" : "1" } }
{ "name" : "Emma","age":32,"job":"Product
Manager","gender":"female","salary":35000 }
{ "index" : {  "_id" : "2" } }
{ "name" : "Underwood","age":41,"job":"Dev Manager","gender":"male","salary":
50000}
{ "index" : {  "_id" : "3" } }
{ "name" : "Tran","age":25,"job":"Web Designer","gender":"male","salary":18000 }
{ "index" : {  "_id" : "4" } }
{ "name" : "Rivera","age":26,"job":"Web Designer","gender":"female","salary":
22000}
{ "index" : {  "_id" : "5" } }
{ "name" : "Rose","age":25,"job":"QA","gender":"female","salary":18000 }
{ "index" : {  "_id" : "6" } }
{ "name" : "Lucy","age":31,"job":"QA","gender":"female","salary": 25000}
{ "index" : {  "_id" : "7" } }
{ "name" : "Byrd","age":27,"job":"QA","gender":"male","salary":20000 }
{ "index" : {  "_id" : "8" } }
{ "name" : "Foster","age":27,"job":"Java Programmer","gender":"male","salary":
20000}
{ "index" : {  "_id" : "9" } }
{ "name" : "Gregory","age":32,"job":"Java
Programmer","gender":"male","salary":22000 }
{ "index" : {  "_id" : "10" } }
{ "name" : "Bryant","age":20,"job":"Java Programmer","gender":"male","salary":
9000}
{ "index" : {  "_id" : "11" } }
{ "name" : "Jenny","age":36,"job":"Java
Programmer","gender":"female","salary":38000 }
{ "index" : {  "_id" : "12" } }
{ "name" : "Mcdonald","age":31,"job":"Java Programmer","gender":"male","salary":
32000}
{ "index" : {  "_id" : "13" } }
{ "name" : "Jonthna","age":30,"job":"Java
Programmer","gender":"female","salary":30000 }
{ "index" : {  "_id" : "14" } }
{ "name" : "Marshall","age":32,"job":"Javascript
Programmer","gender":"male","salary": 25000}
{ "index" : {  "_id" : "15" } }
{ "name" : "King","age":33,"job":"Java Programmer","gender":"male","salary":28000
}
{ "index" : {  "_id" : "16" } }
{ "name" : "Mccarthy","age":21,"job":"Javascript
Programmer","gender":"male","salary": 16000}
{ "index" : {  "_id" : "17" } }
{ "name" : "Goodwin","age":25,"job":"Javascript
Programmer","gender":"male","salary": 16000}
{ "index" : {  "_id" : "18" } }
{ "name" : "Catherine","age":29,"job":"Javascript
Programmer","gender":"female","salary": 20000}
{ "index" : {  "_id" : "19" } }
{ "name" : "Boone","age":30,"job":"DBA","gender":"male","salary": 30000}
{ "index" : {  "_id" : "20" } }
{ "name" : "Kathy","age":29,"job":"DBA","gender":"female","salary": 20000}

# Metric 聚合，找到最低的工资
POST employees/_search
{
  "size": 0,
  "aggs": {
    "min_salary": {
      "min": {
        "field":"salary"
      }
    }
  }
}

# Metric 聚合，找到最高的工资
POST employees/_search
{
  "size": 0,
  "aggs": {
    "max_salary": {
      "max": {
        "field":"salary"
      }
    }
  }
}

# 多个 Metric 聚合，找到最低最高和平均工资
POST employees/_search
{
  "size": 0,
  "aggs": {
    "max_salary": {
      "max": {
        "field": "salary"
      }
    },
    "min_salary": {
      "min": {
        "field": "salary"
      }
    },
    "avg_salary": {
      "avg": {
        "field": "salary"
      }
    }
  }
}

# 一个聚合，输出多值
POST employees/_search
{
  "size": 0,
  "aggs": {
    "stats_salary": {
      "stats": {
        "field":"salary"
      }
    }
  }
}




# 对keword 进行聚合
POST employees/_search
{
  "size": 0,
  "aggs": {
    "jobs": {
      "terms": {
        "field":"job.keyword"
      }
    }
  }
}


# 对 Text 字段进行 terms 聚合查询，失败
POST employees/_search
{
  "size": 0,
  "aggs": {
    "jobs": {
      "terms": {
        "field":"job"
      }
    }
  }
}

# 对 Text 字段打开 fielddata，支持terms aggregation
PUT employees/_mapping
{
  "properties" : {
    "job":{
       "type":     "text",
       "fielddata": true
    }
  }
}
```

```
# 对 Text 字段进行 terms 分词。分词后的terms
POST employees/_search
{
  "size": 0,
  "aggs": {
    "jobs": {
      "terms": {
        "field":"job"
      }
    }
  }
}

POST employees/_search
{
  "size": 0,
  "aggs": {
    "jobs": {
      "terms": {
        "field":"job.keyword"
      }
    }
  }
}


# 对job.keyword 和 job 进行 terms 聚合，分桶的总数并不一样
POST employees/_search
{
  "size": 0,
  "aggs": {
    "cardinate": {
      "cardinality": {
        "field": "job"
      }
    }
  }
}


# 对 性别的 keyword 进行聚合
POST employees/_search
{
  "size": 0,
  "aggs": {
    "gender": {
      "terms": {
        "field":"gender"
      }
    }
  }
}


#指定 bucket 的 size
POST employees/_search
{
  "size": 0,
  "aggs": {
    "ages_5": {
      "terms": {
        "field":"age",
        "size":3
      }
    }
  }
}



# 指定size，不同工种中，年纪最大的3个员工的具体信息
POST employees/_search
{
  "size": 0,
  "aggs": {
    "jobs": {
      "terms": {
        "field":"job.keyword"
      },
      "aggs":{
        "old_employee":{
          "top_hits":{
            "size":3,
            "sort":[
              {
                "age":{
                  "order":"desc"
                }
              }
            ]
          }
        }
      }
    }
  }
}



#Salary Ranges 分桶，可以自己定义 key
POST employees/_search
{
  "size": 0,
  "aggs": {
    "salary_range": {
      "range": {
        "field":"salary",
        "ranges":[
          {
            "to":10000
          },
          {
            "from":10000,
            "to":20000
          },
          {
            "key":">20000",
            "from":20000
          }
        ]
      }
    }
  }
}


#Salary Histogram,工资0到10万，以 5000一个区间进行分桶
POST employees/_search
{
  "size": 0,
  "aggs": {
    "salary_histrogram": {
      "histogram": {
        "field":"salary",
        "interval":5000,
        "extended_bounds":{
          "min":0,
          "max":100000

        }
      }
    }
  }
}


# 嵌套聚合1，按照工作类型分桶，并统计工资信息
POST employees/_search
{
  "size": 0,
  "aggs": {
    "Job_salary_stats": {
      "terms": {
        "field": "job.keyword"
      },
      "aggs": {
        "salary": {
          "stats": {
            "field": "salary"
          }
        }
      }
    }
  }
}

# 多次嵌套。根据工作类型分桶，然后按照性别分桶，计算工资的统计信息
POST employees/_search
{
  "size": 0,
  "aggs": {
    "Job_gender_stats": {
      "terms": {
        "field": "job.keyword"
      },
      "aggs": {
        "gender_stats": {
          "terms": {
            "field": "gender"
          },
          "aggs": {
            "salary_stats": {
              "stats": {
                "field": "salary"
              }
            }
          }
        }
      }
    }
  }
}
```

相关阅读

- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/search-aggregations-metrics.html
- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/search-aggregations-bucket.html

Pipeline 聚合分析

课程 demo

```
DELETE employees
PUT /employees/_bulk
{ "index" : {  "_id" : "1" } }
{ "name" : "Emma","age":32,"job":"Product
Manager","gender":"female","salary":35000 }
{ "index" : {  "_id" : "2" } }
{ "name" : "Underwood","age":41,"job":"Dev Manager","gender":"male","salary":
50000}
{ "index" : {  "_id" : "3" } }
{ "name" : "Tran","age":25,"job":"Web Designer","gender":"male","salary":18000 }
{ "index" : {  "_id" : "4" } }
{ "name" : "Rivera","age":26,"job":"Web Designer","gender":"female","salary":
22000}
{ "index" : {  "_id" : "5" } }
{ "name" : "Rose","age":25,"job":"QA","gender":"female","salary":18000 }
{ "index" : {  "_id" : "6" } }
{ "name" : "Lucy","age":31,"job":"QA","gender":"female","salary": 25000}
{ "index" : {  "_id" : "7" } }
{ "name" : "Byrd","age":27,"job":"QA","gender":"male","salary":20000 }
{ "index" : {  "_id" : "8" } }
{ "name" : "Foster","age":27,"job":"Java Programmer","gender":"male","salary":
20000}
{ "index" : {  "_id" : "9" } }
{ "name" : "Gregory","age":32,"job":"Java
Programmer","gender":"male","salary":22000 }
{ "index" : {  "_id" : "10" } }
{ "name" : "Bryant","age":20,"job":"Java Programmer","gender":"male","salary":
9000}
{ "index" : {  "_id" : "11" } }
{ "name" : "Jenny","age":36,"job":"Java
Programmer","gender":"female","salary":38000 }
{ "index" : {  "_id" : "12" } }
{ "name" : "Mcdonald","age":31,"job":"Java Programmer","gender":"male","salary":
32000}
{ "index" : {  "_id" : "13" } }
{ "name" : "Jonthna","age":30,"job":"Java
Programmer","gender":"female","salary":30000 }
{ "index" : {  "_id" : "14" } }
{ "name" : "Marshall","age":32,"job":"Javascript
Programmer","gender":"male","salary": 25000}
{ "index" : {  "_id" : "15" } }
{ "name" : "King","age":33,"job":"Java Programmer","gender":"male","salary":28000
}
{ "index" : {  "_id" : "16" } }
{ "name" : "Mccarthy","age":21,"job":"Javascript
Programmer","gender":"male","salary": 16000}
{ "index" : {  "_id" : "17" } }
{ "name" : "Goodwin","age":25,"job":"Javascript
Programmer","gender":"male","salary": 16000}
{ "index" : {  "_id" : "18" } }
{ "name" : "Catherine","age":29,"job":"Javascript
Programmer","gender":"female","salary": 20000}
{ "index" : {  "_id" : "19" } }
{ "name" : "Boone","age":30,"job":"DBA","gender":"male","salary": 30000}
{ "index" : {  "_id" : "20" } }
{ "name" : "Kathy","age":29,"job":"DBA","gender":"female","salary": 20000}



# 平均工资最低的工作类型
POST employees/_search
{
  "size": 0,
  "aggs": {
    "jobs": {
      "terms": {
        "field": "job.keyword",
        "size": 10
      },
      "aggs": {
        "avg_salary": {
          "avg": {
            "field": "salary"
          }
        }
      }
    },
    "min_salary_by_job":{
      "min_bucket": {
        "buckets_path": "jobs>avg_salary"
      }
    }
  }
}


# 平均工资最高的工作类型
POST employees/_search
{
  "size": 0,
  "aggs": {
    "jobs": {
      "terms": {
        "field": "job.keyword",
        "size": 10
      },
      "aggs": {
        "avg_salary": {
          "avg": {
            "field": "salary"
          }
        }
      }
    },
    "max_salary_by_job":{
      "max_bucket": {
        "buckets_path": "jobs>avg_salary"
      }
    }
  }
}


# 平均工资的平均工资
POST employees/_search
{
  "size": 0,
  "aggs": {
    "jobs": {
      "terms": {
        "field": "job.keyword",
        "size": 10
      },
      "aggs": {
        "avg_salary": {
          "avg": {
            "field": "salary"
          }
        }
      }
    },
    "avg_salary_by_job":{
      "avg_bucket": {
        "buckets_path": "jobs>avg_salary"
      }
    }
  }
}
```

```
# 平均工资的统计分析
POST employees/_search
{
  "size": 0,
  "aggs": {
    "jobs": {
      "terms": {
        "field": "job.keyword",
        "size": 10
      },
      "aggs": {
        "avg_salary": {
          "avg": {
            "field": "salary"
          }
        }
      }
    },
    "stats_salary_by_job":{
      "stats_bucket": {
        "buckets_path": "jobs>avg_salary"
      }
    }
  }
}


# 平均工资的百分位数
POST employees/_search
{
  "size": 0,
  "aggs": {
    "jobs": {
      "terms": {
        "field": "job.keyword",
        "size": 10
      },
      "aggs": {
        "avg_salary": {
          "avg": {
            "field": "salary"
          }
        }
      }
    },
    "percentiles_salary_by_job":{
      "percentiles_bucket": {
        "buckets_path": "jobs>avg_salary"
      }
    }
  }
}



#按照年龄对平均工资求导
POST employees/_search
```

```
{
  "size": 0,
  "aggs": {
    "age": {
      "histogram": {
        "field": "age",
        "min_doc_count": 1,
        "interval": 1
      },
      "aggs": {
        "avg_salary": {
          "avg": {
            "field": "salary"
          }
        },
        "derivative_avg_salary":{
          "derivative": {
            "buckets_path": "avg_salary"
          }
        }
      }
    }
  }
}


#Cumulative_sum
POST employees/_search
{
  "size": 0,
  "aggs": {
    "age": {
      "histogram": {
        "field": "age",
        "min_doc_count": 1,
        "interval": 1
      },
      "aggs": {
        "avg_salary": {
          "avg": {
            "field": "salary"
          }
        },
        "cumulative_salary":{
          "cumulative_sum": {
            "buckets_path": "avg_salary"
          }
        }
      }
    }
  }
}

#Moving Function
POST employees/_search
{
  "size": 0,
  "aggs": {
    "age": {
      "histogram": {
        "field": "age",
        "min_doc_count": 1,
        "interval": 1
      },
      "aggs": {
        "avg_salary": {
          "avg": {
            "field": "salary"
          }
        },
        "moving_avg_salary":{
          "moving_fn": {
            "buckets_path": "avg_salary",
            "window":10,
            "script": "MovingFunctions.min(values)"
          }
        }
      }
    }
  }
}
```

相关阅读

- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/search-aggregations-pipeline.html

作用范围与排序

课程demo

```
DELETE /employees
PUT /employees/
{
  "mappings" : {
      "properties" : {
        "age" : {
          "type" : "integer"
        },
        "gender" : {
          "type" : "keyword"
        },
        "job" : {
          "type" : "text",
          "fields" : {
            "keyword" : {
              "type" : "keyword",
              "ignore_above" : 50
            }
          }
        },
        "name" : {
          "type" : "keyword"
        },
        "salary" : {
          "type" : "integer"
        }
      }
    }
}

PUT /employees/_bulk
{ "index" : {  "_id" : "1" } }
{ "name" : "Emma","age":32,"job":"Product
Manager","gender":"female","salary":35000 }
{ "index" : {  "_id" : "2" } }
{ "name" : "Underwood","age":41,"job":"Dev Manager","gender":"male","salary":
50000}
{ "index" : {  "_id" : "3" } }
{ "name" : "Tran","age":25,"job":"Web Designer","gender":"male","salary":18000 }
{ "index" : {  "_id" : "4" } }
{ "name" : "Rivera","age":26,"job":"Web Designer","gender":"female","salary":
22000}
{ "index" : {  "_id" : "5" } }
{ "name" : "Rose","age":25,"job":"QA","gender":"female","salary":18000 }
{ "index" : {  "_id" : "6" } }
{ "name" : "Lucy","age":31,"job":"QA","gender":"female","salary": 25000}
{ "index" : {  "_id" : "7" } }
{ "name" : "Byrd","age":27,"job":"QA","gender":"male","salary":20000 }
{ "index" : {  "_id" : "8" } }
{ "name" : "Foster","age":27,"job":"Java Programmer","gender":"male","salary":
20000}
{ "index" : {  "_id" : "9" } }
{ "name" : "Gregory","age":32,"job":"Java
Programmer","gender":"male","salary":22000 }
{ "index" : {  "_id" : "10" } }
{ "name" : "Bryant","age":20,"job":"Java Programmer","gender":"male","salary":
9000}
{ "index" : {  "_id" : "11" } }
{ "name" : "Jenny","age":36,"job":"Java
Programmer","gender":"female","salary":38000 }
{ "index" : {  "_id" : "12" } }
{ "name" : "Mcdonald","age":31,"job":"Java Programmer","gender":"male","salary":
32000}
{ "index" : {  "_id" : "13" } }
{ "name" : "Jonthna","age":30,"job":"Java
Programmer","gender":"female","salary":30000 }
{ "index" : {  "_id" : "14" } }
{ "name" : "Marshall","age":32,"job":"Javascript
Programmer","gender":"male","salary": 25000}
{ "index" : {  "_id" : "15" } }
{ "name" : "King","age":33,"job":"Java Programmer","gender":"male","salary":28000
}
{ "index" : {  "_id" : "16" } }
{ "name" : "Mccarthy","age":21,"job":"Javascript
Programmer","gender":"male","salary": 16000}
{ "index" : {  "_id" : "17" } }
{ "name" : "Goodwin","age":25,"job":"Javascript
Programmer","gender":"male","salary": 16000}
{ "index" : {  "_id" : "18" } }
{ "name" : "Catherine","age":29,"job":"Javascript
Programmer","gender":"female","salary": 20000}
{ "index" : {  "_id" : "19" } }
{ "name" : "Boone","age":30,"job":"DBA","gender":"male","salary": 30000}
{ "index" : {  "_id" : "20" } }
{ "name" : "Kathy","age":29,"job":"DBA","gender":"female","salary": 20000}



# Query
POST employees/_search
{
  "size": 0,
  "query": {
    "range": {
      "age": {
        "gte": 20
      }
    }
  },
  "aggs": {
    "jobs": {
      "terms": {
        "field":"job.keyword"

      }
    }
  }
}


#Filter
POST employees/_search
{
  "size": 0,
  "aggs": {
    "older_person": {
      "filter":{
        "range":{
          "age":{
            "from":35
          }
        }
      },
      "aggs":{
         "jobs":{
           "terms": {
        "field":"job.keyword"
      }
      }
    }},
    "all_jobs": {
      "terms": {
        "field":"job.keyword"

      }
    }
  }
}



#Post field. 一条语句，找出所有的job类型。还能找到聚合后符合条件的结果
POST employees/_search
{
  "aggs": {
    "jobs": {
      "terms": {
        "field": "job.keyword"
      }
    }
  },
  "post_filter": {
    "match": {
      "job.keyword": "Dev Manager"
    }
  }
}


#global
POST employees/_search
{
  "size": 0,
  "query": {
    "range": {
      "age": {
        "gte": 40
      }
    }
  },
  "aggs": {
    "jobs": {
      "terms": {
        "field":"job.keyword"

      }
    },

    "all":{
      "global":{},
      "aggs":{
        "salary_avg":{
          "avg":{
            "field":"salary"
          }
        }
      }
    }
  }
}


#排序 order
#count and key
POST employees/_search
```

```
{
  "size": 0,
  "query": {
    "range": {
      "age": {
        "gte": 20
      }
    }
  },
  "aggs": {
    "jobs": {
      "terms": {
        "field":"job.keyword",
        "order":[
          {"_count":"asc"},
          {"_key":"desc"}
          ]

      }
    }
  }
}


#排序 order
#count and key
POST employees/_search
{
  "size": 0,
  "aggs": {
    "jobs": {
      "terms": {
        "field":"job.keyword",
        "order":[  {
            "avg_salary":"desc"
          }]


      },
    "aggs": {
      "avg_salary": {
        "avg": {
          "field":"salary"
        }
      }
    }
    }
  }
}


#排序 order
#count and key
POST employees/_search
{
  "size": 0,
  "aggs": {
    "jobs": {
```

~~~
   "terms": {
    "field":"job.keyword",
    "order":[  {
      "stats_salary.min":"desc"
     }]


   },
  "aggs": {
   "stats_salary": {
    "stats": {
     "field":"salary"
    }
   }
  }
  }
 }
}
```# 聚合分析的原理及精准度问题
## 课程demo
~~~

```
DELETE my_flights
PUT my_flights
{
 "settings": {
  "number_of_shards": 20
 },
 "mappings" : {
   "properties" : {
    "AvgTicketPrice" : {
     "type" : "float"
    },
    "Cancelled" : {
     "type" : "boolean"
    },
    "Carrier" : {
     "type" : "keyword"
    },
    "Dest" : {
     "type" : "keyword"
    },
    "DestAirportID" : {
     "type" : "keyword"
    },
    "DestCityName" : {
     "type" : "keyword"
    },
    "DestCountry" : {
     "type" : "keyword"
    },
    "DestLocation" : {
     "type" : "geo_point"
    },
    "DestRegion" : {
```

```
  "type" : "keyword"
},
"DestWeather" : {
  "type" : "keyword"
},
"DistanceKilometers" : {
  "type" : "float"
},
"DistanceMiles" : {
  "type" : "float"
},
"FlightDelay" : {
  "type" : "boolean"
},
"FlightDelayMin" : {
  "type" : "integer"
},
"FlightDelayType" : {
  "type" : "keyword"
},
"FlightNum" : {
  "type" : "keyword"
},
"FlightTimeHour" : {
  "type" : "keyword"
},
"FlightTimeMin" : {
  "type" : "float"
},
"Origin" : {
  "type" : "keyword"
},
"OriginAirportID" : {
  "type" : "keyword"
},
"OriginCityName" : {
  "type" : "keyword"
},
"OriginCountry" : {
  "type" : "keyword"
},
"OriginLocation" : {
  "type" : "geo_point"
},
"OriginRegion" : {
  "type" : "keyword"
},
"OriginWeather" : {
  "type" : "keyword"
},
"dayOfWeek" : {
  "type" : "integer"
```

```
        },
        "timestamp" : {
          "type" : "date"
        }
      }
    }
}
```

```
POST _reindex
{
  "source": {
    "index": "kibana_sample_data_flights"
  },
  "dest": {
    "index": "my_flights"
  }
}
```

GET kibana_sample_data_flights/*count*
*GET my_flights/*count

```
get kibana_sample_data_flights/_search
```

```
GET kibana_sample_data_flights/_search
{
  "size": 0,
  "aggs": {
    "weather": {
      "terms": {
        "field":"OriginWeather",
        "size":5,
        "show_term_doc_count_error":true
      }
    }
  }
}
```

```
GET my_flights/_search
{
  "size": 0,
  "aggs": {
    "weather": {
      "terms": {
        "field":"OriginWeather",
        "size":1,
        "shard_size":1,
        "show_term_doc_count_error":true
      }
    }
  }
}
```

```
## 相关阅读
# 对象及Nested对象
## 课程demos
```

```
DELETE blog
```

设置blog的 Mapping

```
PUT /blog
{
  "mappings": {
    "properties": {
      "content": {
        "type": "text"
      },
      "time": {
        "type": "date"
      },
      "user": {
        "properties": {
          "city": {
            "type": "text"
          },
          "userid": {
            "type": "long"
          },
          "username": {
            "type": "keyword"
          }
        }
      }
    }
  }
}
```

插入一条 Blog 信息

```
PUT blog/_doc/1
{
  "content":"I like Elasticsearch",
  "time":"2019-01-01T00:00:00",
  "user":{
    "userid":1,
    "username":"Jack",
    "city":"Shanghai"
  }
}
```

查询 Blog 信息

```
POST blog/_search
{
 "query": {
  "bool": {
   "must": [
    {"match": {"content": "Elasticsearch"}},
    {"match": {"user.username": "Jack"}}
   ]
  }
 }
}
```

```
DELETE my_movies
```

电影的Mapping信息

```
PUT my_movies
{
   "mappings" : {
   "properties" : {
    "actors" : {
     "properties" : {
      "first_name" : {
       "type" : "keyword"
      },
      "last_name" : {
       "type" : "keyword"
      }
     }
    },
    "title" : {
     "type" : "text",
     "fields" : {
      "keyword" : {
       "type" : "keyword",
       "ignore_above" : 256
      }
     }
    }
   }
  }
}
```

写入一条电影信息

```
POST my_movies/_doc/1
{
 "title":"Speed",
 "actors":[
  {
   "first_name":"Keanu",
   "last_name":"Reeves"
  },
```

```
  {
    "first_name":"Dennis",
    "last_name":"Hopper"
  }
```

```
 ]
}
```

查询电影信息

```
POST my_movies/_search
{
 "query": {
  "bool": {
   "must": [
    {"match": {"actors.first_name": "Keanu"}},
    {"match": {"actors.last_name": "Hopper"}}
   ]
  }
 }

}
```

DELETE my_movies

创建 Nested 对象 Mapping

```
PUT my_movies
{
   "mappings" : {
   "properties" : {
    "actors" : {
     "type": "nested",
     "properties" : {
      "first_name" : {"type" : "keyword"},
      "last_name" : {"type" : "keyword"}
     }},
    "title" : {
     "type" : "text",
     "fields" : {"keyword":{"type":"keyword","ignore_above":256}}
    }
   }
  }
}
```

```
POST my_movies/_doc/1
{
 "title":"Speed",
 "actors":[
  {
   "first_name":"Keanu",
   "last_name":"Reeves"
  },
```

```
{
  "first_name":"Dennis",
  "last_name":"Hopper"
}
```

```
 ]
}
```

Nested 查询

```
POST my_movies/_search
{
 "query": {
  "bool": {
   "must": [
    {"match": {"title": "Speed"}},
    {
     "nested": {
      "path": "actors",
      "query": {
       "bool": {
        "must": [
         {"match": {
          "actors.first_name": "Keanu"
         }},
```

```
              {"match": {
                "actors.last_name": "Hopper"
              }}
            ]
          }
        }
      }
    }
  ]
}
```

```
 }
}
```

Nested Aggregation

```
POST my_movies/_search
{
 "size": 0,
 "aggs": {
  "actors": {
   "nested": {
    "path": "actors"
   },
   "aggs": {
    "actor_name": {
     "terms": {
      "field": "actors.first_name",
      "size": 10
     }
    }
   }
  }
 }
}
```

普通 aggregation不工作

```
POST my_movies/_search
{
 "size": 0,
 "aggs": {
  "NAME": {
   "terms": {
    "field": "actors.first_name",
    "size": 10
   }
  }
 }
}
```

```
## 相关阅读
- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/query-dsl-nested-query.html
# 文档的父子关系
## 课程demo
```

```
DELETE my_blogs
```

设定 Parent/Child Mapping

```
PUT my_blogs
{
 "settings": {
  "number_of_shards": 2
 },
 "mappings": {
  "properties": {
   "blog_comments_relation": {
    "type": "join",
    "relations": {
     "blog": "comment"
    }
   },
      "content": {
        "type": "text"
      },
      "title": {
        "type": "keyword"
      }
    }
  }
}

#索引父文档
PUT my_blogs/_doc/blog1
{
  "title":"Learning Elasticsearch",
  "content":"learning ELK @ geektime",
  "blog_comments_relation":{
    "name":"blog"
  }
}

#索引父文档
PUT my_blogs/_doc/blog2
{
  "title":"Learning Hadoop",
  "content":"learning Hadoop",
    "blog_comments_relation":{
    "name":"blog"
  }
}

#索引子文档
PUT my_blogs/_doc/comment1?routing=blog1
{
  "comment":"I am learning ELK",
  "username":"Jack",
  "blog_comments_relation":{
    "name":"comment",
    "parent":"blog1"
  }
}

#索引子文档
PUT my_blogs/_doc/comment2?routing=blog2
{
  "comment":"I like Hadoop!!!!!",
  "username":"Jack",
  "blog_comments_relation":{
    "name":"comment",
    "parent":"blog2"
  }
}
```

```
#索引子文档
PUT my_blogs/_doc/comment3?routing=blog2
{
  "comment":"Hello Hadoop",
  "username":"Bob",
  "blog_comments_relation":{
    "name":"comment",
    "parent":"blog2"
  }
}
```

查询所有文档

```
POST my_blogs/_search
{

}
```

```
#根据父文档ID查看
GET my_blogs/_doc/blog2
```

Parent Id 查询

```
POST my_blogs/_search
{
  "query": {
    "parent_id": {
      "type": "comment",
      "id": "blog2"
    }
  }
}
```

Has Child 查询,返回父文档

```
POST my_blogs/_search
{
  "query": {
    "has_child": {
      "type": "comment",
      "query" : {
                "match": {
                    "username" : "Jack"
                }
            }
    }
  }
}
```

Has Parent 查询，返回相关的子文档

```
POST my_blogs/_search
{
  "query": {
    "has_parent": {
      "parent_type": "blog",
      "query" : {
                "match": {
                    "title" : "Learning Hadoop"
                }
            }
    }
  }
}
```

#通过ID ，访问子文档
GET my_blogs/*doc/comment3*
#通过ID和routing ，访问子文档
*GET my_blogs/*doc/comment3?routing=blog2

#更新子文档

```
PUT my_blogs/_doc/comment3?routing=blog2
{
    "comment": "Hello Hadoop??",
    "blog_comments_relation": {
      "name": "comment",
      "parent": "blog2"
    }
}
```

```
## 相关阅读
- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/query-dsl-has-child-query.html
- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/query-dsl-has-parent-query.html
- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/query-dsl-parent-id-query.html
- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/query-dsl-parent-id-query.html
# Update by Query & Reindex
## 课程demo
```

DELETE blogs/

写入文档

```
PUT blogs/_doc/1
{
  "content":"Hadoop is cool",
  "keyword":"hadoop"
}
```

查看 Mapping

```
GET blogs/_mapping
```

修改 Mapping，增加子字段，使用英文分词器

```
PUT blogs/_mapping
{
      "properties" : {
        "content" : {
          "type" : "text",
          "fields" : {
            "english" : {
              "type" : "text",
              "analyzer":"english"
            }
          }
        }
      }
    }
```

写入文档

```
PUT blogs/_doc/2
{
  "content":"Elasticsearch rocks",
    "keyword":"elasticsearch"
}
```

查询新写入文档

```
POST blogs/_search
{
  "query": {
    "match": {
      "content.english": "Elasticsearch"
    }
  }

}
```

查询 Mapping 变更前写入的文档

```
POST blogs/_search
{
  "query": {
    "match": {
      "content.english": "Hadoop"
    }
  }
}
```

Update所有文档

```
POST blogs/_update_by_query
{

}
```

查询之前写入的文档

```
POST blogs/_search
{
  "query": {
    "match": {
      "content.english": "Hadoop"
    }
  }
}
```

查询

```
GET blogs/_mapping
```

```
PUT blogs/_mapping
{
        "properties" : {
        "content" : {
          "type" : "text",
          "fields" : {
            "english" : {
              "type" : "text",
              "analyzer" : "english"
            }
          }
        },
        "keyword" : {
          "type" : "keyword"
        }
      }
}
```

```
DELETE blogs_fix
```

创建新的索引并且设定新的Mapping

```
PUT blogs_fix/
{
  "mappings": {
        "properties" : {
```

```
        "content" : {
          "type" : "text",
          "fields" : {
            "english" : {
              "type" : "text",
              "analyzer" : "english"
            }
          }
        },
        "keyword" : {
          "type" : "keyword"
        }
      }
  }
}
```

Reindx API

```
POST  _reindex
{
  "source": {
    "index": "blogs"
  },
  "dest": {
    "index": "blogs_fix"
  }
}
```

```
GET  blogs_fix/_doc/1
```

测试 Term Aggregation

```
POST blogs_fix/_search
{
  "size": 0,
  "aggs": {
    "blog_keyword": {
      "terms": {
        "field": "keyword",
        "size": 10
      }
    }
  }
}
```

Reindx API，version Type Internal

```
POST  _reindex
{
  "source": {
    "index": "blogs"
  },
  "dest": {
    "index": "blogs_fix",
    "version_type": "internal"
  }
}
```

文档版本号增加

```
GET  blogs_fix/_doc/1
```

Reindx API，version Type Internal

```
POST  _reindex
{
  "source": {
    "index": "blogs"
  },
  "dest": {
    "index": "blogs_fix",
    "version_type": "external"
  }
}
```

Reindx API，version Type Internal

```
POST  _reindex
{
  "source": {
    "index": "blogs"
  },
  "dest": {
    "index": "blogs_fix",
    "version_type": "external"
  },
  "conflicts": "proceed"
}
```

Reindx API，version Type Internal

```
POST  _reindex
{
  "source": {
    "index": "blogs"
  },
  "dest": {
    "index": "blogs_fix",
    "op_type": "create"
  }
}
```

```
GET _tasks?detailed=true&actions=*reindex
```

```
## 相关阅读
- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/docs-reindex.html
- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/docs-update-by-query.html
# Ingest Pipeline & Painless Script
## 课程demo
```

#########Demo for Pipeline###############

DELETE tech_blogs

```
#Blog数据，包含3个字段，tags用逗号间隔
PUT tech_blogs/_doc/1
{
 "title":"Introducing big data......",
 "tags":"hadoop,elasticsearch,spark",
 "content":"You konw, for big data"
}
```

测试split tags

```
POST ingest/pipeline/simulate
{
 "pipeline": {
  "description": "to split blog tags",
  "processors": [
   {
    "split": {
     "field": "tags",
     "separator": ","
    }
   }
  ]
 },
 "docs": [
  {
   "index": "index",
   "id": "id",
   "source": {
    "title": "Introducing big data......",
    "tags": "hadoop,elasticsearch,spark",
    "content": "You konw, for big data"
   }
  },
  {
   "index": "index",
   "id": "idxx",
   "source": {
    "title": "Introducing cloud computering",
    "tags": "openstack,k8s",
    "content": "You konw, for cloud"
   }
  }
 ]
}
```

```
#同时为文档，增加一个字段。blog查看量
POST ingest/pipeline/simulate
{
 "pipeline": {
   "description": "to split blog tags",
   "processors": [
     {
       "split": {
         "field": "tags",
         "separator": ","
       }
     },
```

```
    {
      "set":{
        "field": "views",
        "value": 0
      }
    }
  ]
```

```
 },
 "docs": [
   {
     "index":"index",
     "id":"id",
     "_source":{
       "title":"Introducing big data......",
 "tags":"hadoop,elasticsearch,spark",
 "content":"You konw, for big data"
     }
   },
```

```
  {
    "_index":"index",
    "_id":"idxx",
    "_source":{
      "title":"Introducing cloud computering",
```

```
 "tags":"openstack,k8s",
 "content":"You konw, for cloud"
     }
   }
```

```
  ]
```

```
}
```

为ES添加一个 Pipeline

```
PUT _ingest/pipeline/blog_pipeline
{
  "description": "a blog pipeline",
  "processors": [
    {
      "split": {
        "field": "tags",
        "separator": ","
      }
    },
```

```
    {
      "set":{
        "field": "views",
        "value": 0
      }
    }
  ]
```

```
}
```

```
#查看Pipleline
GET _ingest/pipeline/blog_pipeline
```

```
#测试pipeline
POST ingest/pipeline/blog_pipeline/simulate
{
  "docs": [
    {
      "_source": {
        "title": "Introducing cloud computering",
        "tags": "openstack,k8s",
        "content": "You konw, for cloud"
      }
    }
  ]
}
```

```
#不使用pipeline更新数据
PUT tech_blogs/_doc/1
{
  "title":"Introducing big data......",
  "tags":"hadoop,elasticsearch,spark",
  "content":"You konw, for big data"
}
```

```
#使用pipeline更新数据
PUT tech_blogs/_doc/2?pipeline=blog_pipeline
{
  "title": "Introducing cloud computering",
  "tags": "openstack,k8s",
  "content": "You konw, for cloud"
}
```

```
#查看两条数据，一条被处理，一条未被处理
POST tech_blogs/_search
{}
```

```
#update_by_query 会导致错误
POST tech_blogs/_update_by_query?pipeline=blog_pipeline
{
}
```

```
#增加update_by_query的条件
POST tech_blogs/_update_by_query?pipeline=blog_pipeline
{
    "query": {
        "bool": {
            "must_not": {
                "exists": {
                    "field": "views"
                }
            }
        }
    }
}
```

#########Demo for Painless###############

增加一个 Script Prcessor

```
POST ingest/pipeline/simulate
{
  "pipeline": {
    "description": "to split blog tags",
    "processors": [
      {
        "split": {
          "field": "tags",
          "separator": ","
        }
      },
      {
        "script": {
          "source": """
          if(ctx.containsKey("content")){
            ctx.content_length = ctx.content.length();
          }else{
            ctx.content_length=0;
          }
        """
      }
    },

    {
      "set":{
        "field": "views",
        "value": 0
      }
    }
  ]
```

```
 },
 "docs": [
  {
    "index":"index",
    "id":"id",
    "_source":{
      "title":"Introducing big data......",
 "tags":"hadoop,elasticsearch,spark",
 "content":"You konw, for big data"
    }
  },
```

```
  {
    "_index":"index",
    "_id":"idxx",
    "_source":{
      "title":"Introducing cloud computering",
```

```
 "tags":"openstack,k8s",
 "content":"You konw, for cloud"
    }
  }
```

```
  ]
```

```
}
```

```
DELETE tech_blogs
PUT tech_blogs/_doc/1
{
 "title":"Introducing big data......",
 "tags":"hadoop,elasticsearch,spark",
 "content":"You konw, for big data",
 "views":0
}
```

```
POST tech_blogs/update/1
{
 "script": {
  "source": "ctx.source.views += params.new_views",
  "params": {
      "new_views":100
    }
  }
}
```

查看views计数

```
POST tech_blogs/_search
{

}
```

```
#保存脚本在 Cluster State
POST scripts/update_views
{
  "script":{
    "lang": "painless",
    "source": "ctx.source.views += params.new_views"
  }
}
```

```
POST tech_blogs/_update/1
{
  "script": {
    "id": "update_views",
    "params": {
      "new_views":1000
    }
  }
}
```

```
GET tech_blogs/_search
{
  "script_fields": {
    "rnd_views": {
      "script": {
        "lang": "painless",
        "source": """
          java.util.Random rnd = new Random();
          doc['views'].value+rnd.nextInt(1000);
        """
      }
    }
  },
  "query": {
    "match_all": {}
  }
}
```

```
## 相关阅读
- https://www.elastic.co/cn/blog/should-i-use-logstash-or-elasticsearch-ingest-nodes
- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/ingest-apis.html
- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/ingest-processors.html
- https://www.elastic.co/guide/en/elasticsearch/painless/7.1/painless-lang-spec.html
- https://www.elastic.co/guide/en/elasticsearch/painless/7.1/painless-api-reference.html
# Elasticsearch 数据建模实例
##课程demo
```

Data Modeling Example

Index 一本书的信息

```
PUT books/_doc/1
{
  "title":"Mastering ElasticSearch 5.0",
  "description":"Master the searching, indexing, and aggregation features in ElasticSearch Improve users’ search experience with Elasticsearch’s functionalities and develop your own Elasticsearch plugins",
  "author":"Bharvi Dixit",
  "public_date":"2017",
  "cover_url":"https://images-na.ssl-images-amazon.com/images/I/51OeaMFxcML.jpg"
}



#查询自动创建的Mapping
GET books/_mapping

DELETE books

#优化字段类型
PUT books
{
      "mappings" : {
      "properties" : {
        "author" : {"type" : "keyword"},
        "cover_url" : {"type" : "keyword","index": false},
        "description" : {"type" : "text"},
        "public_date" : {"type" : "date"},
        "title" : {
          "type" : "text",
          "fields" : {
            "keyword" : {
              "type" : "keyword",
              "ignore_above" : 100
            }
          }
        }
      }
    }
}
#Cover URL index 设置成false，无法对该字段进行搜索
POST books/_search
{
  "query": {
    "term": {
      "cover_url": {
        "value": "https://images-na.ssl-images-amazon.com/images/I/51OeaMFxcML.jpg"
      }
    }
  }
}
#Cover URL index 设置成false，依然支持聚合分析
POST books/_search
{
  "aggs": {
    "cover": {
      "terms": {
        "field": "cover_url",
        "size": 10
      }
    }
  }
}
DELETE books
#新增 Content字段。数据量很大。选择将Source 关闭
PUT books
{
      "mappings" : {
      "_source": {"enabled": false},
      "properties" : {
        "author" : {"type" : "keyword","store": true},
        "cover_url" : {"type" : "keyword","index": false,"store": true},
        "description" : {"type" : "text","store": true},
         "content" : {"type" : "text","store": true},
        "public_date" : {"type" : "date","store": true},
        "title" : {
          "type" : "text",
          "fields" : {
            "keyword" : {
              "type" : "keyword",
              "ignore_above" : 100
            }
          },
          "store": true
        }
      }
  }
}
```

Index 一本书的信息,包含Content

```
PUT books/_doc/1
{
  "title":"Mastering ElasticSearch 5.0",
  "description":"Master the searching, indexing, and aggregation features in ElasticSearch Improve users’ search experience with Elasticsearch’s functionalities and develop your own Elasticsearch plugins",
  "content":"The content of the book......Indexing data, aggregation, searching.    something else. something in the way............",
  "author":"Bharvi Dixit",
  "public_date":"2017",
  "cover_url":"https://images-na.ssl-images-amazon.com/images/I/51OeaMFxcML.jpg"
}
```

```
#查询结果中，Source不包含数据
POST books/_search
{}
```

```
#搜索，通过store 字段显示数据，同时高亮显示 conent的内容
POST books/_search
{
  "stored_fields": ["title","author","public_date"],
  "query": {
    "match": {
      "content": "searching"
    }
  },

  "highlight": {
    "fields": {
      "content":{}
    }
  }

}
```

```
##相关阅读
# Elasticsearch 数据建模最佳实践

# 课程demo
```

Cookie Service

```
##索引数据，dynamic mapping 会不断加入新增字段
PUT cookie_service/_doc/1
{
 "url":"www.google.com",
 "cookies":{
   "username":"tom",
  "age":32
 }
}

PUT cookie_service/_doc/2
{
 "url":"www.amazon.com",
 "cookies":{
  "login":"2019-01-01",
  "email":"xyz@abc.com"
 }
}

DELETE cookie_service
#使用 Nested 对象，增加key/value
PUT cookie_service
{
 "mappings": {
  "properties": {
   "cookies": {
    "type": "nested",
    "properties": {
     "name": {
      "type": "keyword"
     },
     "dateValue": {
      "type": "date"
     },
     "keywordValue": {
      "type": "keyword"
     },
     "IntValue": {
      "type": "integer"
     }
    }
   },
   "url": {
    "type": "text",
    "fields": {
     "keyword": {
      "type": "keyword",
      "ignore_above": 256
     }
    }
   }
  }
 }
}

##写入数据，使用key和合适类型的value字段
PUT cookie_service/_doc/1
{
 "url":"www.google.com",
 "cookies":[
    {
      "name":"username",
      "keywordValue":"tom"
    },
    {
       "name":"age",
      "intValue":32
```

```
    }
```

```
   ]
 }
PUT cookie_service/_doc/2
{
 "url":"www.amazon.com",
 "cookies":[
    {
      "name":"login",
      "dateValue":"2019-01-01"
    },
    {
       "name":"email",
      "IntValue":32
```

```
    }
```

```
   ]
 }
```

Nested 查询，通过bool查询进行过滤

```
POST cookie_service/_search
{
  "query": {
    "nested": {
      "path": "cookies",
      "query": {
        "bool": {
          "filter": [
            {
            "term": {
              "cookies.name": "age"
            }},
            {
              "range":{
                "cookies.intValue":{
                  "gte":30
                }
              }
            }
          ]
        }
      }
    }
  }
}
```

在Mapping中加入元信息，便于管理

```
PUT softwares/
{
  "mappings": {
    "_meta": {
      "software_version_mapping": "1.0"
    }
  }
}
```

```
GET softwares/mapping
PUT softwares/doc/1
{
  "software_version":"7.1.0"
}
```

```
DELETE softwares
```

优化,使用inner object

```
PUT softwares/
{
  "mappings": {
    "_meta": {
      "software_version_mapping": "1.1"
    },
    "properties": {
      "version": {
        "properties": {
          "display_name": {
            "type": "keyword"
          },
          "hot_fix": {
            "type": "byte"
          },
          "marjor": {
            "type": "byte"
          },
          "minor": {
            "type": "byte"
          }
        }
      }
    }
  }
}
```

```
#通过 Inner Object 写入多个文档
PUT softwares/_doc/1
{
  "version":{
  "display_name":"7.1.0",
  "marjor":7,
  "minor":1,
  "hot_fix":0
  }

}
```

```
PUT softwares/_doc/2
{
  "version":{
  "display_name":"7.2.0",
  "marjor":7,
  "minor":2,
  "hot_fix":0
  }
}
```

```
PUT softwares/_doc/3
{
  "version":{
  "display_name":"7.2.1",
  "marjor":7,
  "minor":2,
  "hot_fix":1
  }
}
```

通过 bool 查询，

```
POST softwares/_search
{
  "query": {
    "bool": {
      "filter": [
        {
          "match":{
            "version.marjor":7
          }
        },
        {
          "match":{
            "version.minor":2
          }
        }
    ]
  }
```

```
 }
}
```

Not Null 解决聚合的问题

```
DELETE ratings
PUT ratings
{
  "mappings": {
      "properties": {
        "rating": {
          "type": "float",
          "null_value": 1.0
        }
      }
    }
}
```

```
PUT ratings/doc/1
{
 "rating":5
}
PUT ratings/doc/2
{
 "rating":null
}
```

```
POST ratings/search
POST ratings/search
{
  "size": 0,
  "aggs": {
    "avg": {
      "avg": {
        "field": "rating"
      }
    }
  }
}
```

```
POST ratings/_search
{
  "query": {
    "term": {
      "rating": {
        "value": 1
      }
    }
```

```
 }
}
```

```
## 相关阅读
- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/general-recommendations.html
- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/tune-for-disk-usage.html
- https://www.elastic.co/guide/en/elasticsearch/reference/7.1/tune-for-search-speed.html
# 第二部分总结与测验
##课程demo
```

```
DELETE test
PUT test/_doc/1
{
 "content":"Hello World"
}
```

```
POST test/_search
{
 "profile": "true",
 "query": {
  "match": {
   "content": "Hello World"
  }
 }
}
```

```
POST test/_search
{
 "profile": "true",
 "query": {
  "match": {
   "content": "hello world"
  }
 }
}
```

```
POST test/_search
{
 "profile": "true",
 "query": {
  "match": {
   "content.keyword": "Hello World"
  }
 }
}
```

```
POST test/_search
{
 "profile": "true",
 "query": {
    "match": {
      "content.keyword": "hello world"
    }
  }
}

POST test/_search
{
  "profile": "true",
  "query": {
    "term": {
      "content": "Hello World"
    }
  }
}

POST test/_search
{
  "profile": "true",
  "query": {
    "term": {
      "content": "hello world"
    }
  }
}

POST test/_search
{
  "profile": "true",
  "query": {
    "term": {
      "content.keyword": "Hello World"
    }
  }
}
```