qq_42383787
码龄3年 暂无认证

76	25万+	9万+	5万+	
原创	周排名	总排名	访问	等级

1290	7	20	9	65
积分	粉丝	获赞	评论	收藏

私信 关注

搜博主文章

热门文章

分类专栏

最新评论

ES QueryBuilder学习
feather。: 让初次接触者 能够快速理解基础语法

ES 搜索11 (查询语句提升权重)
qq_42383787: ....这个只是翻译了官方的原例...

ES 搜索11 (查询语句提升权重)
小老犇: 同样是程序员，为何你能如此秀 ..

最新文章

2020年 12篇 2019年 68篇
2018年 11篇

目录

ES QueryBuilder学习

原创 qq_42383787 2019-04-24 09:10:40 6245 收藏 24 版权
分类专栏： ES 文章标签： QueryBuilder

```
    public void test() throws IOException {
        # 创建一个Search 对象
        SearchRequest searchRequest = new SearchRequest();

        # 创建一个Builder 对象 对条件进行封装
        SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();

        # 查询条件: 字段名为text  内容含有19的数据
        searchSourceBuilder.query(QueryBuilders.matchQuery("text", "19"));
        # 查询条件: 在上面条件的基础上 加上字段 jiage 内容含有329的数据
        searchSourceBuilder.query(QueryBuilders.matchQuery("jiage", "329"));

        # 从搜索结果中取第0条开始的10条数据，数据量最多不要超过10000 会报错，有解决方案百度
        searchSourceBuilder.from(0);
        searchSourceBuilder.size(10);

        # 排序根据字段id 按照正序排列
        searchSourceBuilder.sort(new FieldSortBuilder("id").order(SortOrder.ASC));

        searchSourceBuilder.fetchSource(false);

        # 参数，用于是否需要过滤
        String[] includeFields = new String[]{"id", "dizhi","text","jiage"};
        # 第1个参数是 需要显示的字段，第2个参数是需要过滤的字段
        searchSourceBuilder.fetchSource(includeFields, null);

        # 进行构建
        searchRequest.source(searchSourceBuilder);
        searchRequest.scroll(TimeValue.timeValueMinutes(1L));


    SearchResponse searchResponse = client.search(searchRequest, RequestOptions.DEF
         # 获取数据 数据可以是多种类型的
        SearchHits hits = searchResponse.getHits();
        SearchHit[] hits1 = hits.getHits();

        List list = new ArrayList();
        for (SearchHit hit : hits1) {
            list.add(hit.getSourceAsString());
        }
        System.out.println(list);
    }
}


        SearchHit[] searchHits = hits.getHits();
        for (SearchHit hit : searchHits) {
        // 结果的Index
            String index = hit.getIndex();
        // 结果的type
            String type = hit.getType();
        // 结果的ID
            String id = hit.getId();
        // 结果的评分
            float score = hit.getScore();
        // 查询的结果 JSON字符串形式
            String sourceAsString = hit.getSourceAsString();
        // 查询的结果 Map的形式
            Map<String, Object> sourceAsMap = hit.getSourceAsMap();
        // Document的title
            String documentTitle = (String) sourceAsMap.get("title");
        // 结果中的某个List
            List<Object> users = (List<Object>) sourceAsMap.get("user");
        // 结果中的某个Map
             Map<String, Object> innerObject = (Map<String, Object>)sourceAsMap.get("inn
        }
```

QueryBuilder 常用来配合 Search 查询来使用

boolQuery() 布尔查询，可以用来组合多个查询条件
fuzzyQuery() 相似度查询
matchAllQuery() 查询所有数据
regexpQuery() 正则表达式查询
termQuery() 词条查询
wildcardQuery() 模糊查询
等等.....

1.matchQuery 匹配查询:

matchQuery可以简单理解为mysql中的like，但是我不知道我这么理解对不对，因为在elasticsearch中使用matchQuery查询时，他会对查询的field进行分词，打个比方，我们搜索"联想笔记本电脑"，他可能会将他拆分为：“联想”，“电脑”，“联想电脑”，那么如果一个filed中包括 联想 两个字就可以被搜出来。当然我们进行查询的这个field的mapping必须是text类型。（如果是中文分词的话，还需要配置中文分词器），他的查询语句和上边基本相似

```
public void test() throws IOException {
        SearchRequest searchRequest = new SearchRequest();
        SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();

        # matchQuery 匹配查询
        searchSourceBuilder.query(QueryBuilders.matchQuery("text", "19"));


    # multiMatchQuery(Object text, String... fieldNames) 多个字段匹配某一个值
    # 搜索name中或interest中包含有music的文档 (必须与music一致)
    public void test() throws IOException {
        SearchRequest searchRequest = new SearchRequest();
        SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();


     searchSourceBuilder.query(QueryBuilders.multiMatchQuery("music", "name", "inter
}17


    # wildcardQuery()模糊查询，?匹配单个字符，*匹配多个字符
    # 搜索名字中含有jack文档 （name中只要包含jack即可）
    public void test() throws IOException {
        SearchRequest searchRequest = new SearchRequest();
        SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();

        searchSourceBuilder.query(QueryBuilders.wildcardQuery("name","*jack?*"));
}
```

2.matchAllQuery 查询所用

查询指定index和type中的所用记录，相当于sql：select * from sales

```
    public void test() throws IOException {
        SearchRequest searchRequest = new SearchRequest();
        SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();
        searchSourceBuilder.query(QueryBuilders.matchAllQuery());
```

3.termQuery等值搜索

我们在数据库中进行查询的时候，sql：select sales from tvs where brand = ‘小米’，那么在elasticsearch中的javaapi怎么写呢？这里我们用到一个termQuery，他相当于sql语句中的“=”，使用这个搜索一般是对索引中keyword的mapping进行等值搜索　　term query 属于过滤器查询，可以处理数字（numbers）、布尔值（Booleans）、日期（dates）以及文本（text）。

```
    public void test() throws IOException {
        SearchRequest searchRequest = new SearchRequest();
        SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();
        searchSourceBuilder.query(QueryBuilders.termQuery("name", "张三"));
}


    # terms 是多值判断

    public void test() throws IOException {
        SearchRequest searchRequest = new SearchRequest();
        SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();

     searchSourceBuilder.query(QueryBuilders.termsQuery("name", "张三", "李四", "王五"
}
```

4.matchPhraseQuery短语搜索

理解：你会发现，使用“小别克老”没有查询出任何结果，而使用“小别克听”则查询出了我们需要的结果，这便matchPhraseQuery和matchQuery等的区别，在使用matchQuery等时，即使你传入的是“小别克老”，在执行查询时，“小别克老”会被分词器分词，例如paoding解析成“小别/别克/老”，而使用matchPhraseQuery时，“小别克老”并不会被分词器分词，而是直接以一个短语的形式查询，而如果你在创建索引所使用的field的value中没有这么一个短语（顺序无差，且连接在一起），那么将查询不出任何结果。

```
    public void test() throws IOException {
        SearchRequest searchRequest = new SearchRequest();
        SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();
        searchSourceBuilder.query(QueryBuilders.matchPhraseQuery("name", "张三"));
}
```

5.prefixQuery前缀搜索

如我我们需要查询的title中有“大话西游电影”，“大话西游小说”，使用prefixQuery查询“大话西游”，那么那两条数据就会出来

```
    public void test() throws IOException {
        SearchRequest searchRequest = new SearchRequest();
        SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();
        searchSourceBuilder.query(QueryBuilders.prefixQuery("title", "大话西游"));
}
```

6.disMaxQuery

disMaxQuery适用于多个field的进行搜索，我们在多个field搜索时候，可能会遇到多个field匹配到了更多的词会在前面，而一个field匹配了更多的词就会排名靠后。disMax就是解决这个问题，dismax使搜索到的结果，应该是某一个field中匹配到了尽可能多的关键词，被排在前面；而不是尽可能多的field匹配到了少数的关键词，排在了前面

```
    public void test() throws IOException {
        SearchRequest searchRequest = new SearchRequest();
        SearchSourceBuilder searchSourceBuilder = newSearchSourceBuilder();

searchSourceBuilder.query(QueryBuilders.disMaxQuery().add(QueryBuilders.matchQuery("nam
}
```

7.boolQuery 组合查询条件

boolQuery用来将搜索的条件进行组合，即将多个组合条件组合在一起，常用的几种组合方式有must、should、mustNot，我们拿下面对应的sql语句举例子

```

# sql:select * from sales where brand = '小米' and color='红色'，通过bool将两个查询条件组合，
# must相当于sql中的= 必须匹配的意思 3
BoolQueryBuilder boolQuery = QueryBuilders.boolQuery(); 4
       boolQuery.must(QueryBuilders.termQuery("brand", "小米")) 5
               .must(QueryBuilders.termQuery("color", "红色")); 6

7
# sql:select * from sales where brand = '小米' or color='红色'; 使用should相当于sql语句中的c
8 BoolQueryBuilder boolQuery2 = QueryBuilders.boolQuery(); 9
       boolQuery2.should(QueryBuilders.termQuery("brand", "小米"))10
               .should(QueryBuilders.termQuery("color", "红色"));11

12
# sql:select * from sales where brand = '小米' and color != '红色' mustNot相当于!= 必须不匹
13 BoolQueryBuilder boolQuery3 = QueryBuilders.boolQuery();14
       boolQuery2.must(QueryBuilders.termQuery("brand", "小米"))15
               .mustNot(QueryBuilders.termQuery("color", "红色"));16
17 # sql:select * from sales where (brand = '小米' or color = '红色') and brand != '长虹'
18 BoolQueryBuilder boolQuery4 = QueryBuilders.boolQuery();
19        BoolQueryBuilder boolQuery5 = QueryBuilders.boolQuery();
20        boolQuery5.should(QueryBuilders.termQuery("brand", "小米"))
21                .should(QueryBuilders.termQuery("color", "红色"));
22        boolQuery4.must(boolQuery5)
23                .mustNot(QueryBuilders.termQuery("brand", "长虹"));
24
25
26
27
28 //设置查询条件
29
30        BoolQueryBuilder boolQuery = QueryBuilders.boolQuery();
31        if (StringUtils.isNotEmpty(text)) {
32            //查询条件：字段名为text，内容含有text的数据
33            boolQuery.must(QueryBuilders.matchQuery("text", text));
34        }
35        if (StringUtils.isNotEmpty(keywords)) {
36            //查询条件：字段名为keywords，内容含有keywords的数据
37            boolQuery.must(QueryBuilders.matchQuery("keywords", keywords));
38        }
39        if (StringUtils.isNotEmpty(topic)) {
40            //查询条件：字段名为topic，内容含有topic的数据
41            boolQuery.must(QueryBuilders.matchQuery("topic", topic));
42        }
```

8.rangeQuery属于过滤器查询

是范围查询，有时候，范围查询比精确值查询更有用，比如我想知道价格在20到40之间的商品；

range query可以处理数字（numbers）、日期（dates）以及字符串，不过字符串还是不要用范围查询的好，效率会很低；

对数字取范围没啥好说的， 就大于、大于等于、小于、小于等于四个符号加数字就可以；

```
# 闭区间查询
QueryBuilder qb1 = QueryBuilders.rangeQuery("${fieldName}").from(${fieldValue1}).to($

# 开区间查询
QueryBuilder qb1 = QueryBuilders.rangeQuery("${fieldName}").from(${fieldValue1}, fals

# 大于
QueryBuilder qb1 = QueryBuilders.rangeQuery("${fieldName}").gt(${fieldValue});
```

```
QueryBuilder qb1 = QueryBuilders.rangeQuery("${fieldName}").gte(${fieldValue});

# 小于
QueryBuilder qb1 = QueryBuilders.rangeQuery("${fieldName}").lt(${fieldValue});

# 小于等于
QueryBuilder qb1 = QueryBuilders.rangeQuery("${fieldName}").lte(${fieldValue});

# 多条件案例
QueryBuilder qb1 = QueryBuilders.moreLikeThisQuery(new String[]{"${fieldName1}"}, new
QueryBuilder qb2 = QueryBuilders.rangeQuery("${fieldName2}").gt("${fieldValue2}");
QueryBuilder qb3 = QueryBuilders.boolQuery().must(qb1).must(qb2);
```

官网介绍：https://www.elastic.co/guide/en/elasticsearch/client/java-rest/current/java-rest-high-query-builders.html

QueryBuilder 11-18

类似pl/sql的软件， 实现可视化sql语句构造操作

优质评论可以帮助作者获得更高权重 评论

feather。： 让初次接触者 能够快速理解基础语法 27 天前 回复

黑伊人8023： 大哥，官网英语看起大脑壳，请求翻译一遍 10 月前 回复

qq_42383787 博主 回复： 你可以用浏览器自带的翻译嘛，如果实在是看不懂的就把关键字复制到百度去搜索一下，一般能懂个大致。 10 月前 回复 1

我都不知道取什么名字： 大哥，组合查询是这样的

```
//设置查询条件

        BoolQueryBuilder boolQuery = QueryBuilders.boolQuery();
        if (StringUtils.isNotEmpty(text)) {
            //查询条件：字段名为text，内容含有text的数据
            boolQuery.must(QueryBuilders.matchQuery("text", text));
        }
        if (StringUtils.isNotEmpty(keywords)) {
            //查询条件：字段名为keywords，内容含有keywords的数据
            boolQuery.must(QueryBuilders.matchQuery("keywords", keywords));
        }
        if (StringUtils.isNotEmpty(topic)) {
            //查询条件：字段名为topic，内容含有topic的数据
            boolQuery.must(QueryBuilders.matchQuery("topic", topic));
        }
```

2 年前 回复

qq_42383787 博主 回复： 大哥，举一反三就好...这个只是个参考不是得把所有情况都写出来... 2 年前 回复

相关推荐

学习GreenDAO的QueryBuilder的使用_flybamboo的博客 5-17
每次forCurrentThread()被调用,在查询使用builder被构建时参数被设置初始化。 Raw queries 以防QueryBuilder不提供你所…

使用jQuery QueryBuilder构建查询_芸灵fly的博客-CSDN... 6-1
最近在给老项目做各种新功能,其中一项是给数据添加各种用户自定义的查询,但是用户不会写SQL,我们得用图形化界面才能…

QueryBuilder For SqlServer1.0(可视化查询工具) 02-23
QueryBuilder For SqlServer1.0(可视化查询工具)支持拖拽,通过连线处理表之间的关系,可视化添加数据查询条件,支持Excel…

VueQueryBuilder用于构建具有嵌套条件的复杂查询UI组件 08-09
Vue Query Builder 用于构建具有嵌套条件的复杂查询UI组件

Java JPA QueryBuilder (自己写的)_没事new一下的博客 5-24
publicQueryBuilderequals(Object obj, String column){ QueryParam queryParam =newQueryParam(obj, column, EQUALS)…

java操作QueryBuilders常见用法_子之乐鱼之乐的博客-CS... 5-30
package com.elasticsearch; import org.elasticsearch.action.ActionListener; import org.elasticsearch.action.search.Search…

Es QueryBuilder 简单查询 qq_36189144的博客 3546
1.matchQuery(String name,Object text) matchQuery("filedname","value")匹配单个字段，匹配字段名为filedname,值为valu…

QueryBuilder简单查询 S-H_A-N 3万+
1.matchAllQuery() matchAllQuery()方法用来匹配全部文档 public class QueryTest { public static void main(String[] args) { //…

关于elasticsearch的几个QueryBuilder_房东的狗 6-13
关于elasticsearch的几个QueryBuilder TermQueryBuilder 对词条进行完全匹配 计算机=计算机WildcardQueryBuilder 对词…

SelectQueryBuilder的用法_初仔仔的博客 5-30
同样的查询能够使用SelectQueryBuilder类建立。 SelectQueryBuilder query =newSelectQueryBuilder(); query.SelectFromT…

QueryBuilder构造ES查询条件 精准匹配、模糊搜索、in、范围查询 and or xiaozm1223的博客 2万+
package com.xzm.es; import java.net.InetSocketAddress; import java.util.ArrayList; import java.util.Iterator; import java.util.…

记录一次es商品门店查询，关键字SearchSourceBuilder C18298182575的博客 6155
SearchSourceBuilder使用说明 参考：https://www.cnblogs.com/reycg-blog/p/9946821.html @Override public List<String> …

Elasticsearch java api 常用查询方法QueryBuilder构造... 6-8
QueryBuilder qb1 = QueryBuilders.termQuery("${fieldName}","${fieldValue}"); 1 批量 QueryBuilder qb1 = QueryBuilders.te…

ElasticSearch的快速入门 smilefyou的博客 26
简介 概括 elasticsearch是一个基于Lucene的高扩展的分布式搜索服务器，支持开箱即用,隐藏了Lucene的复杂性，对外提…

CSDN开发者助手，常用网站自动整合，多种工具一键调用
CSDN开发者助手由CSDN官方开发，集成一键呼出搜索、万能快捷工具、个性化新标签页和官方免广告四大功能。帮助您…

python中不能使用索引运算的是_MongoDB指南---11、使用复合索引、$操作符… weixin_39996908的博客 81
1、使用复合索引在多个键上建立的索引就是复合索引，在上面的小节中，已经使用过复合索引。复合索引比单键索引要复…