



链滴

Elastic Search RUL Search

作者: [YongSiv](#)

原文链接: <https://ld246.com/article/1606353674710>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



URL Search

Url Query常用参数:

1.简单查询

- q指定查询语句，语法为Query String Syntax
- df q中不指定字段时默认查询的字段，如果不指定，ES会查询所有字段
- sort 指定排序字段
- timeout指定超时时间，默认不超时
- from, size用于分页

example:

```
PUT test_search_index
{
  "settings": {
    "index": {
      "number_of_shards": "1"
    }
  }
}
```

```
PUT test_search_index/doc/_bulk
{"index":{"_id":"1"}}
{"username":"alfred way","job":"java engineer","age":18,"birth":"1990-01-02","isMarried":false}
{"index":{"_id":"2"}}
{"username":"alfred","job":"java senior engineer and java specialist", "age":28,"birth":"1980-05
```

```
07","isMarried":true}
{"index":{"_id":"3"}}
{"username":"lee","job":"java and ruby engineer","age":22,"birth":"1985-08-07","isMarried":false}
{"index":{"_id":"4"}}
{"username":"alfred junior way","job":"ruby engineer","age":23,"birth":"1989-08-07","isMarried":false}
```

GET /my_index/_search?q=alfred&df=user&sort=age:asc&from=4&size=10&timeout=1s

2.过滤查询

-term与phrase

alfred way 等价于alfred OR way, OR要大写

"alfred way" 短语查询

-泛查询

alfred等价于在所有字段中匹配term

-指定字段

name:alfred

检索所有字段包含alfred

GET test_search_index/_search?q=alfred

```
{
  "profile":true
}
```

检索username包含alfred

GET test_search_index/_search?q=username:alfred

检索username包含alfred或者所有字段中包含way

GET test_search_index/_search?q=username:alfred way

检索username包含alfred或者way

GET test_search_index/_search?q=username:(alfred way)

检索username包含alfred way词汇

GET test_search_index/_search?q=username:"alfred way"

检索username包含alfred, 并且age大于26

GET test_search_index/_search?q=username:alfred age:>26

检索birth大于1980且小于1990

GET test_search_index/_search?q=birth:(>1980 AND <1990)

3.分组查询

Group分组设定,使用括号指定匹配规则

(quick OR brown) AND for
status:(active OR pending)title:(full text search)

4.逻辑运算符

-布尔操作符

- AND (&&), OR(||), NOT(!)

name:(tom NOT lee)

注意大写, 不能小写

- + - 分别对应must和must_not

name:(tom +lee -alfred), + -符号要紧跟字符串

name:((lee && !alfred) || (tom && lee && !alfred))

+在url中会被解析空格, 要使用encode后的结果, 即%2B

检索username包含alfred或者所有字段中包含way
GET test_search_index/_search?q=username:alfred AND way

检索username包含alfred和username包含way
GET test_search_index/_search?q=username:(alfred AND way)

检索username包含alfred或者username包含way
GET test_search_index/_search?q=username:(alfred way)
GET test_search_index/_search?q=username:(alfred OR way)

检索username包含alfred且不包含way
GET test_search_index/_search?q=username:(alfred NOT way)
username包含alfred且必须包含way, +号必须写经过encode后的值
GET test_search_index/_search?q=username:(alfred AND %2Bway)

5.数值和日期范围查询

-区间写法, 闭区间用[], 开区间用{}

age:[1 TO 10] 意为 $1 \leq \text{age} \leq 10$

age:[1 TO 10} 意为 $1 \leq \text{age} < 10$

age:[1 TO] 意为 $\text{age} \geq 1$

age:[* TO 10] 意为 ≤ 10

-算数写法

age: ≥ 1

age:(≥ 1 AND ≤ 10)

age:(≥ 10 + ≤ 10)

检索 $1 \leq \text{age} \leq 23$
GET test_search_index/_search?q=age:[1 TO 23]

```
# 检索age<=20
GET test_search_index/_search?q=age:[* TO 20]

# 检索1<=age<28
GET test_search_index/_search?q=age:([1 TO 28])

# 检索 20<=age<=28
GET test_search_index/_search?q=age:(>=20 AND <=28)
GET test_search_index/_search?q=age:(%2B>=20 %2B<=28)
```

6.通配符查询

- ?代表1个字符, *代表0个或多个字符

```
name:t?m
name:tom*
name:t*m
```

通配符匹配执行效率低, 且占用较多内存, 不建议使用
如无特殊需求, 不要将?/*放在最前面

```
GET test_search_index/_search?q=username:/[a]?.!*/
{
  "profile":true
}
```

7.正则和占位匹配

-正则匹配

```
name:/[mb]oat/
```

-模糊匹配

```
name:roam~1
```

匹配与roam差1个character的词, 如foam roams等

-近似度匹配(proximity search)

```
"fox quick"~5
```

以term为单位进行差异比较, 比如"quick for" "quick brown fox"都会匹配

```
GET test_search_index/_search?q=username:alfed~
```

```
GET test_search_index/_search?q=job:"java engineer"~2
```