



Một số câu query hay sử dụng trong Elasticsearch

🕒 Thứ Năm, 16/04/2020 🧑 Trâm Ho

Xin chào mọi người, hôm trước mình có giới thiệu đến các bạn 1 package **Scout Elasticsearch Driver In Laravel** nên hôm nay mình xin giới thiệu đến mọi người một số câu query hay dùng khi mình thực hiện tính năng search trong các dự án nha.

MỘT SỐ CÂU QUERY

Trước khi bắt đầu các bạn nhớ cài elasticsearch, import data và đánh index cho nó nha 🤗🤗🤗

Giả sử mình có 1 VD minh họa sau nha. Trong Model Customer mình có khai báo mapping như sau:

```
2      * Mapping array
3      *
4      * @var array
5      *
6      protected $mapping = [
7          'properties' => [
8              'name' => [
9                  'type' => 'text',
10                 ],
11                 'description' => [
12                     'type' => 'text',
13                 ],
14             ],
15         ];
16     ];
```

1. MATCH QUERY

Match query là một query cơ bản và nổi bật của elasticsearch, nó trả về kết quả nếu bất kỳ document nào chứa query truyền lên.
VD: Mình có query sau:

```
1 GET customers/_search
2 {
3     "query": {
4         "match": {
5             "name": "Ngoc Nguyen"
6         }
7     }
8 }
9
```

=> Kết quả tìm thấy sẽ chứa tất cả các document có chứa từ **Ngoc** hoặc **Nguyễn** đều được trả về.

2. MATCH-PHRASE QUERY

Cũng giống như match query nhưng match_phrase query là search chính xác. Nó chỉ trả về các document nào chứa chính xác giống query truyền lên.
VD:

```
1 GET customers/_search
2 {
3     "query": {
4         "match_phrase": {
5             "name": "Nguyen Thi Bich Ngoc"
6         }
7     }
8 }
9
```

=> Kết quả sẽ chỉ trả về những document nào có text **Nguyen Thi Bich Ngoc** thôi ạ. Đúng theo thứ tự và text truyền lên.

3. MULTI-MATCH QUERY

Về bản chất multi-match query nó giống như search theo match query và match-phrase query, nhưng nó search đi trên nhiều field khác nhau.
VD:

```
1 GET customers/_search
2 {
3     "query": {
4         "multi_match": {
5             "query": "Bich Ngoc",
6             "fields": ["name", "description"],
7             "type": "phrase"
8         }
9     }
10 }
11
```

Trong đó:

query: là query bạn truyền lên

fields: là các field mà bạn muốn search trong đó. Ở đây mình muốn tìm kiếm các document có text **Bich Ngoc** trong field name và description Customer

type: là kiểu mà bạn muốn search. Mặc định khi không truyền lên trường này thì sẽ hiểu đang search theo kiểu match. Có rất nhiều kiểu type chúng ta có thể sử dụng, mn xem thêm ở [đây](#) nha.

Trong ví dụ trên mình dùng search chính xác (match-phrase query). Nó sẽ tìm kiếm xem trong 2 field name và description xem có document nào có text **Bich Ngoc** sẽ trả ra kết quả. À ở đây search theo kiểu OR, 1 trong các field có query text đều trả ra kết quả ạ.

4. TERM QUERY

Loại query này thường được dùng cho việc search các field có kiểu dữ liệu dạng số, ngày tháng. Nếu muốn search text chúng ta nên sử dụng match query.

🗨️ WARNING:

Avoid using the term query for text fields.

By default, Elasticsearch changes the values of text fields as part of analysis. This can make finding exact matches for text field values difficult.

To search text field values, use the match query instead.

Term query là kiểu search chính xác. VD dưới đây sẽ query lấy ra customer có id =1.

```
1 GET customers/_search
2 {
3     "query": {
4         "term": {
5             "customer_id": 1
6         }
7     }
8 }
9
```

5. TERMS QUERY

Dùng để search với nhiều query một lúc. Kết quả truy vấn lấy theo kiểu OR
VD:

```
1 GET customers/_search
2 {
3     "query": {
4         "terms": {
5             "customer_id": [1,2]
6         }
7     }
8 }
9
```

=> Kết quả sẽ trả về customer có id =1 hoặc id =2.

6. RANGE QUERY

Là loại query trả về các document trong 1 khoảng điều kiện query. Range query thường được sử dụng search khoảng time, khoảng tuổi,....
VD:

```
1 GET customers/_search
2 {
3     "query": {
4         "range": {
5             "birthday": {
6                 "gte": "1997/01/01",
7                 "lte": "1997/12/31"
8             }
9         }
10     }
11 }
12
```

Một số thông số:

gt: Tùy chọn lớn hơn

gte: Tùy chọn lớn hơn hoặc bằng

lt: Tùy chọn nhỏ hơn

lte: Tùy chọn nhỏ hơn hoặc bằng

Như trong ví dụ trên, kết quả lấy ra các record có ngày sinh nhật từ 1997/01/01 đến ngày 1997/12/31.

7. BOOL QUERY

Là loại query cho phép kết hợp nhiều query khác nhau cùng một lúc.
VD:

```
1 GET customers/_search
2 {
3     "query": {
4         "bool": {
5             "must": [
6                 {
7                     "match_phrase": {
8                         "name": "Nguyen Ngoc"
9                     }
10                },
11                {
12                    "term": {
13                        "customer_id": 1
14                    }
15                }
16            ]
17        }
18    }
19 }
```

Mình hay sử dụng must và should:

must: giống như điều kiện AND. Tức là kết quả được lấy ra phải thỏa mãn tất cả các query truyền lên.

should: search theo điều kiện OR. Chỉ cần thỏa mãn một trong các điều kiện query truyền lên là đc.

KẾT LUẬN

Trên đây là một số loại query phổ biến mình hay sử dụng trong elasticsearch. Ngoài ra mn có thể xem thêm các query khác ở [đây](#) nha

Chia sẻ bài viết ngay



Nguồn bài viết : [Viblo](#)

Bạn có hứng thú đối việc không?

PHP Developer - Saritasa 🔧 PHP Laravel Symfony	\$ 800 USD - 1,500 USD	📍 Quận 1, TP Hồ Chí Minh	📅 Ngày đăng: 27-12-2021
Senior PHP Developer - AS White 🔧 PHP Laravel AWS	\$ 1,300 USD - 3,000 USD	📍 Quận 1, TP Hồ Chí Minh	📅 Ngày đăng: 27-12-2021
PHP Developers - Chainos Solution 🔧 PHP MySQL Laravel	\$ Thỏa thuận	📍 Cầu Giấy, Hà Nội	📅 Ngày đăng: 24-12-2021
PHP Lead - NFQ Asia (8Bit Rockstars) 🔧 PHP Laravel MS SQL	\$ Lên đến 4,000 USD	📍 Quận 1, TP Hồ Chí Minh	📅 Ngày đăng: 23-12-2021
Backend Developer - Vina Payroll Outsourcing (VPO) 🔧 PHP Laravel API	\$ Thỏa thuận	📍 Quận 1, TP Hồ Chí Minh	📅 Ngày đăng: 22-12-2021

0 Comments

Sort by Oldest

Add a comment...

Facebook Comments Plugin

