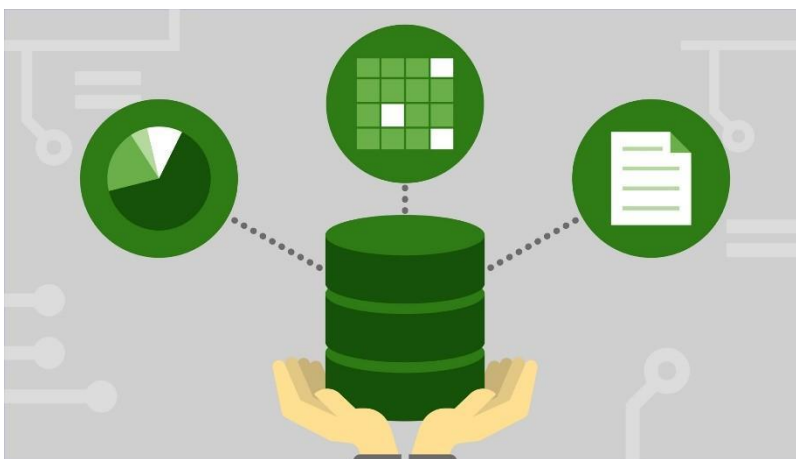


به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پایگاه داده

دستور کار شماره 7

حمیدرضا خدادادی

810197499

دی ماه ۱۴۰۰

گزارش دستور کار انجام شده

در ابتدا الاستیک سرچ و رابط کاربری کیبانا را نصب می کنیم.

در اولین گام، یک داده را با دستور put وارد می کنیم. داده ها به صورت json هستند.

الستیک بررسی می کند که اگر از قبل دیتابیس برای ذخیره داده ها وجود نداشته باشد آن را تولید می کند و داده را اضافه می کند. پس نیازی به ایجاد دستی دیتابیس نیست.

در دستور put خود ایندکس، گروه و شناسه متناظر با داده را وارد می کنیم.

```
1 PUT /movies/movie/1
2 {
3   "title": "The Godfather",
4   "director": "Francis Ford Coppola",
5   "year": 1972
6 }
```

```
1 #! Elasticsearch built-in security features are not enabled. Without authentication,
2 your cluster could be accessible to anyone. See https://www.elastic.co/guide/en
3 /elasticsearch/reference/7.16/security-minimal-setup.html to enable security.
4 #! [types removal] Specifying types in document index requests is deprecated, use the
5 typeless endpoints instead ({index}/{_doc}/{id}, {index}/{_doc}, or {index}/{_create}
6 /{id}).
7 {
8   "_index": "movies",
9   "_type": "movie",
10  "_id": "1",
11  "_version": 1,
12  "result": "created",
13  "_shards": {
14    "total": 2,
15    "successful": 1,
16    "failed": 0
17  },
18  "_seq_no": 0,
19  "_primary_term": 1
20 }
```

حال همان دستور بالا را دوباره اجرا می کنیم. مشاهده می کنیم که ورژن این داده یکی افزوده شده است و result آن از created به updated تغییر پیدا کرده است.

```
1 PUT /movies/movie/1
2 {
3   "title": "The Godfather",
4   "director": "Francis Ford Coppola",
5   "year": 1972
6 }
```

```
1 #! Elasticsearch built-in security features are not enabled. Without authentication,
2 your cluster could be accessible to anyone. See https://www.elastic.co/guide/en
3 /elasticsearch/reference/7.16/security-minimal-setup.html to enable security.
4 #! [types removal] Specifying types in document index requests is deprecated, use the
5 typeless endpoints instead ({index}/{_doc}/{id}, {index}/{_doc}, or {index}/{_create}
6 /{id}).
7 {
8   "_index": "movies",
9   "_type": "movie",
10  "_id": "1",
11  "_version": 2,
12  "result": "updated",
13  "_shards": {
14    "total": 2,
15    "successful": 1,
16    "failed": 0
17  },
18  "_seq_no": 1,
19  "_primary_term": 1
20 }
```

از دو دستور put و post برای ایجاد و به روزرسانی رکورد ها می توان استفاده کرد. بهتر است از put برای ایجاد و از post برای به روزرسانی رکورد استفاده شود.

The screenshot shows the Elasticsearch console interface. The top navigation bar includes 'Console', 'Search Profiler', 'Grok Debugger', 'Painless Lab', and a 'BETA' badge. A 'Click to send request' button is visible. The 'History' tab is active, showing a list of requests. The first request is a POST to 'movies/_doc' with the following JSON body:

```
1 POST /movies/_doc
2 {
3   "title": "The Godfather",
4   "director": "Francis Ford Coppola",
5   "year": 1972
6 }
```

The response is shown on the right, indicating the document was successfully created with ID 'wk526X0B1y1SM8gXx3L3'.

```
1 #! Elasticsearch built-in security features are not enabled. Without authentication,
  your cluster could be accessible to anyone. See https://www.elastic.co/guide/en
  /elasticsearch/reference/7.16/security-minimal-setup.html to enable security.
2 {
3   "_index": "movies",
4   "_type": "_doc",
5   "_id": "wk526X0B1y1SM8gXx3L3",
6   "_version": 1,
7   "result": "created",
8   "_shards": {
9     "total": 2,
10    "successful": 1,
11    "failed": 0
12  },
13   "_seq_no": 5,
14   "_primary_term": 1
15 }
```

برای ادامه کار، همه داده های خود را وارد دیتابیس می کنیم. از دستور put برای این کار استفاده می کنیم.

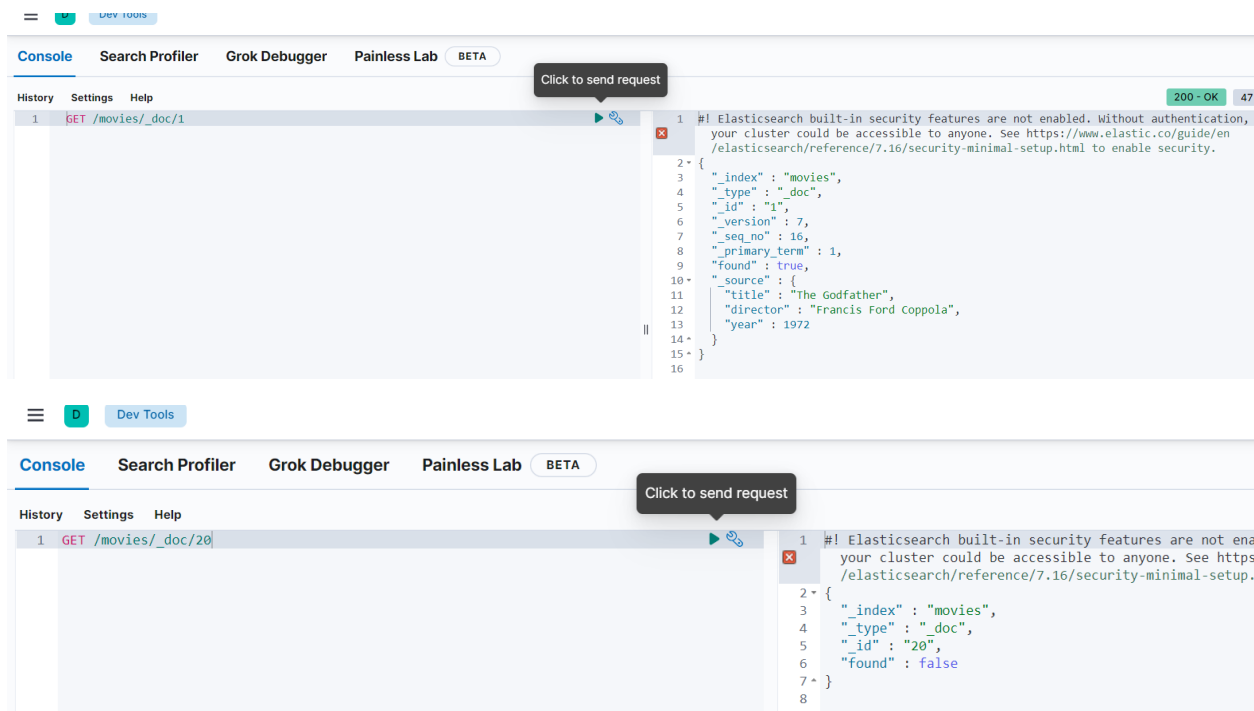
The screenshot shows the Elasticsearch console with multiple PUT requests in the history. The requests are as follows:

- PUT movies/_doc/3: {"title": "To Kill a Mockingbird", "director": "Robert Mulligan", "year": 1962, "genres": ["Crime", "Drama", "Mystery"]}
- PUT movies/_doc/4: {"title": "Apocalypse Now", "director": "Francis Ford Coppola", "year": 1979, "genres": ["Drama", "War"]}
- PUT movies/_doc/5: {"title": "Kill Bill: Vol. 1", "director": "Quentin Tarantino", "year": 2003, "genres": ["Action", "Crime", "Thriller"]}
- PUT movies/_doc/6: {"title": "The Assassination of Jesse James by the Coward Robert Ford", "director": "Andrew Dominik", "year": 2007, "genres": ["Biography", "Crime", "Drama"]}

The response for the last request is shown on the right, indicating the document was successfully created with ID 'wk526X0B1y1SM8gXx3L3'.

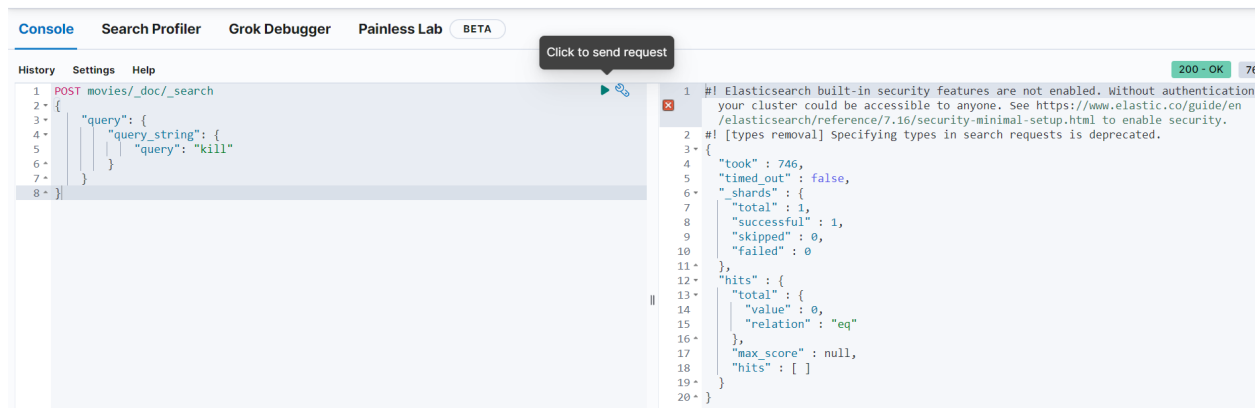
```
1 #! Elasticsearch built-in security features are not enabled. Without authentication,
  your cluster could be accessible to anyone. See https://www.elastic.co/guide/en
  /elasticsearch/reference/7.16/security-minimal-setup.html to enable security.
2 {
3   "_index": "movies",
4   "_type": "_doc",
5   "_id": "6",
6   "_version": 1,
7   "result": "created",
8   "_shards": {
9     "total": 2,
10    "successful": 1,
11    "failed": 0
12  },
13   "_seq_no": 15,
14   "_primary_term": 1
15 }
```

می خواهیم یک رکورد را جست و جو و بازیابی کنیم. برای این کار از دستور get استفاده می کنیم. در تصویر زیر ما اطلاعات یک فیلم با آیدی یک را پیدا کرده ایم. در خروجی ابتدا، ایندکس، گروه، شناسه و فیلد وضعیت پیدا شدن رکورد را خواهیم دید. نتیجه جست و جو درون فیلد source نشان داده خواهد. اگر رکورد پیدا شود، فیلد found برابر true و در غیر این صورت برابر false خواهد شد. در دو عکس زیر خروجی را به ازای هر دو حالت مشاهده می کنید.

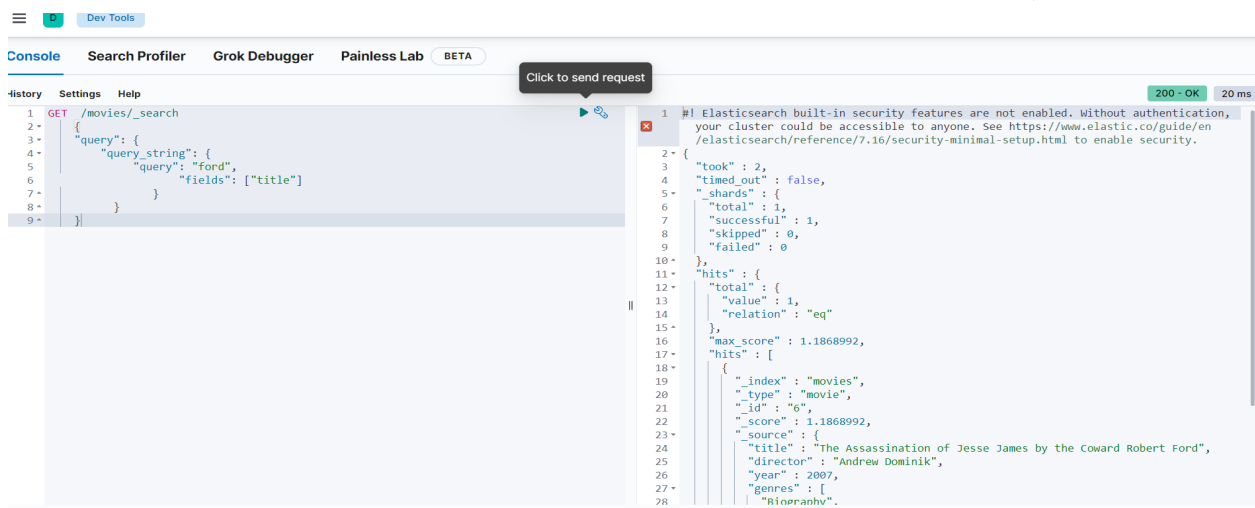


در ادامه سعی می کنیم query زدن در الاستیک را بررسی کنیم. کوئری ها باید به زبان DSL باشند. این زبان، زبان توصیف پرس و جو های الاستیک سرچ است. در شکل زیر کوئری را به صورت استرینگ نوشته ایم. در این کوئری، هر فیلمی که حداقل در یکی از فیلد های آن استرینگ "kill" آمده باشد را پیدا کرده ایم. در خروجی، فیلد took زمان لازم برای جست و جو، hits آرایه کل جواب های پیدا شده و total تعداد کل جواب های پیدا شده است. الاستیک به هر رکوردی که پیدا می کند یک امتیاز اختصاص می دهد و بر اساس امتیاز ها آن را مرتب کرده و نشان می دهد.

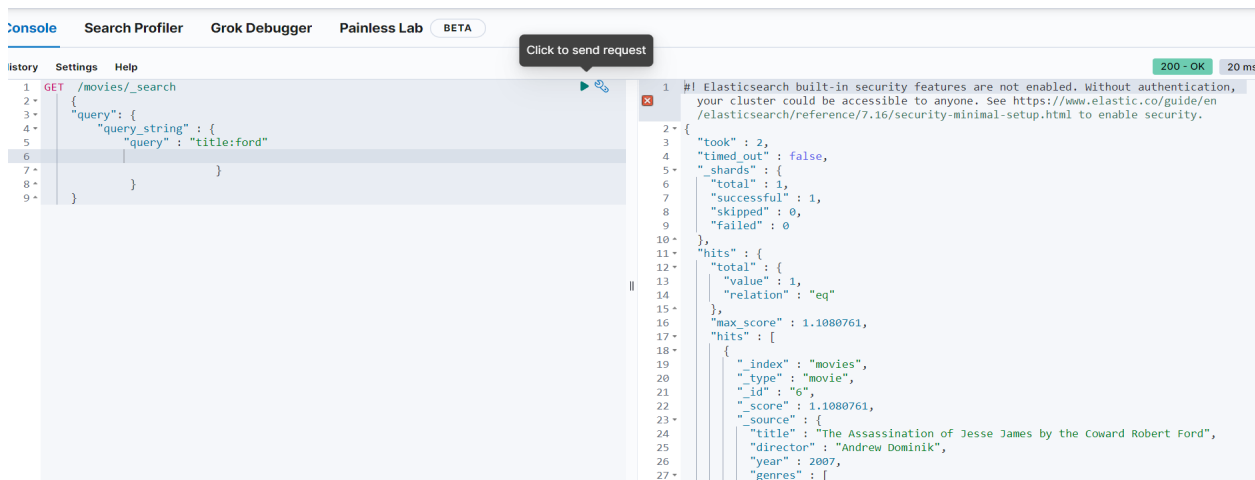
1. در انتهای کوئری های سرچ از `_search` استفاده می کنیم.
2. وقتی بخواهیم در تمامی گروه های یک ایندکس جست و جو کنیم، نام گروه را حذف می کنیم.
3. اگر بخواهیم در تمامی ایندکس ها (همه دیتابیس ها) جست و جو کنیم، نام ایندکس را حذف کرده و `_all` می گذاریم.
4. اگر این کوئری ها را چند بار اجرا کنیم، خروجی آن ها کش می شود و زود تر به پاسخ می رسیم.



در کوئری زیر، جست و جو را محدود تر کرده ایم و در یک فیلد خاص عنوان فیلم ها، استرینگ مورد نظر خود را پیدا کرده ایم.



یک نحوه نوشتن همان کوئری بالا بدین شکل است.



می‌توانیم در لیست فیلد های مورد جست و جو، چند فیلد بنویسیم. بدین شکل:

```
1 GET /movies/_search
2 {
3   "query": {
4     "query_string": {
5       "query": "ford",
6       "fields": ["title", "director"]
7     }
8   }
9 }
```

```
1 #! Elasticsearch built-in security features are not enabled. Without authentication, your cluster could be accessible to anyone. See https://www.elastic.co/guide/en/elasticsearch/reference/7.16/security-minimal-setup.html to enable security.
2 {
3   "took": 4,
4   "timed_out": false,
5   "_shards": {
6     "total": 1,
7     "successful": 1,
8     "skipped": 0,
9     "failed": 0
10  },
11  "hits": {
12    "total": {
13      "value": 8,
14      "relation": "eq"
15    },
16    "max_score": 1.1080761,
17    "hits": [
18      {
19        "_index": "movies",
20        "_type": "movie",
21        "_id": "6",
22        "_score": 1.1080761,
23        "_source": {
24          "title": "The Assassination of Jesse James by the Coward Robert Ford",
25          "director": "Andrew Dominik",
26          "year": 2007,
27          "genres": [
28            "Western"
29          ]
30        }
31      }
32    ]
33  }
34 }
```

در کوئری زیر تمامی فیلم هایی در عنوان آن ها استرینگ "kill" یا "ford" به کار رفته است را پیدا می کنیم. برای این کار از OR استفاده کرده ایم.

```
1 GET /movies/_search
2 {
3   "query": {
4     "query_string": {
5       "query": "title:ford OR kill"
6     }
7   }
8 }
```

```
1 #! Elasticsearch built-in security features are not enabled. Without authentication, your cluster could be accessible to anyone. See https://www.elastic.co/guide/en/elasticsearch/reference/7.16/security-minimal-setup.html to enable security.
2 {
3   "took": 5,
4   "timed_out": false,
5   "_shards": {
6     "total": 1,
7     "successful": 1,
8     "skipped": 0,
9     "failed": 0
10  },
11  "hits": {
12    "total": {
13      "value": 3,
14      "relation": "eq"
15    },
16    "max_score": 1.4193115,
17    "hits": [
18      {
19        "_index": "movies",
20        "_type": "movie",
21        "_id": "3",
22        "_score": 1.4193115,
23        "_source": {
24          "title": "To Kill a Mockingbird",
25          "director": "Robert Mulligan",
26          "year": 1962,
27          "genres": [
28            "Drama"
29          ]
30        }
31      }
32    ]
33  }
34 }
```

در کوئری زیر تمامی فیلم هایی در عنوان آن ها استرینگ "kill" به کار رفته است و کارگردان آن ها Tarantino است را پیدا می کنیم. برای این کار از AND استفاده کرده ایم.

```
1 GET /movies/_search
2 {
3   "query": {
4     "query_string": {
5       "query": "(title:kill) AND (director:Tarantino)"
6     }
7   }
8 }
```

```
1 #! Elasticsearch built-in security features are not enabled. Without authentication, your cluster could be accessible to anyone. See https://www.elastic.co/guide/en/elasticsearch/reference/7.16/security-minimal-setup.html to enable security.
2 {
3   "took": 3,
4   "timed_out": false,
5   "_shards": {
6     "total": 1,
7     "successful": 1,
8     "skipped": 0,
9     "failed": 0
10  },
11  "hits": {
12    "total": {
13      "value": 1,
14      "relation": "eq"
15    },
16    "max_score": 3.7265875,
17    "hits": [
18      {
19        "_index": "movies",
20        "_type": "movie",
21        "_id": "5",
22        "_score": 3.7265875,
23        "_source": {
24          "title": "Kill Bill: Vol. 1",
25          "director": "Quentin Tarantino",
26          "year": 2003,
27          "genres": [
28            "Action"
29          ]
30        }
31      }
32    ]
33  }
34 }
```

در کوئری زیر تمامی فیلم هایی ژانر درام و در سال 1962 را پیدا می کنیم. برای این کار از AND استفاده کرده ایم.

The screenshot shows the Elasticsearch DevTools interface. The left pane contains a REST client with a GET request to `/movies/_search`. The request body is a JSON object with a query string: `"query": "(genres:Drama) AND (year:1962)"`. The right pane shows the response, which includes a warning about security features and a successful search result for the movie "Lawrence of Arabia" by David Lean, released in 1962.

```
1 GET /movies/_search
2 {
3   "query": {
4     "query_string": {
5       "query": "(genres:Drama) AND (year:1962 )"
6     }
7   }
8 }
```

```
1 #! Elasticsearch built-in security features are not enabled. Wit
2 your cluster could be accessible to anyone. See https://www.el
3 /elasticsearch/reference/7.16/security-minimal-setup.html to e
4
5 {
6   "took": 1,
7   "timed_out": false,
8   "_shards": {
9     "total": 1,
10    "successful": 1,
11    "skipped": 0,
12    "failed": 0
13  },
14  "hits": {
15    "total": {
16      "value": 2,
17      "relation": "eq"
18    },
19    "max_score": 1.2795144,
20    "hits": [
21      {
22        "_index": "movies",
23        "_type": "movie",
24        "_id": "2",
25        "_score": 1.2795144,
26        "_source": {
27          "title": "Lawrence of Arabia",
28          "director": "David Lean",
29          "year": 1962,
30          "genres": [
31            "Adventure".
32          ]
33        }
34      }
35    ]
36  }
37 }
```

در کوئری زیر تمامی فیلم هایی ژانر درام و تا سال 2005 را پیدا می کنیم. برای این کار از AND استفاده کرده ایم.

The screenshot shows the Elasticsearch DevTools interface. The left pane contains a REST client with a GET request to `/movies/_search`. The request body is a JSON object with a query string: `"query": "(genres:Drama) AND (year:[* TO 2005])"`. The right pane shows the response, which includes a warning about security features and a successful search result for the movie "Apocalypse Now" by Francis Ford Coppola, released in 1979.

```
1 GET /movies/_search
2 {
3   "query": {
4     "query_string": {
5       "query": "(genres:Drama) AND (year:[* TO 2005] )"
6     }
7   }
8 }
```

```
1 #! Elasticsearch built-in security features are not enabled. Withou
2 your cluster could be accessible to anyone. See https://www.elast
3 /elasticsearch/reference/7.16/security-minimal-setup.html to enab
4
5 {
6   "took": 2,
7   "timed_out": false,
8   "_shards": {
9     "total": 1,
10    "successful": 1,
11    "skipped": 0,
12    "failed": 0
13  },
14  "hits": {
15    "total": {
16      "value": 3,
17      "relation": "eq"
18    },
19    "max_score": 1.3257576,
20    "hits": [
21      {
22        "_index": "movies",
23        "_type": "movie",
24        "_id": "4",
25        "_score": 1.3257576,
26        "_source": {
27          "title": "Apocalypse Now",
28          "director": "Francis Ford Coppola",
29          "year": 1979,
30          "genres": [
31            "Drama".
32          ]
33        }
34      }
35    ]
36  }
37 }
```

در کوئری زیر تمامی فیلم های درام یا فیلم هایی در عنوان آن ها استرینگ "kill" به کار رفته است را پیدا می کنیم. برای این کار از OR استفاده کرده ایم.

```
1 GET /movies/_search
2 {
3   "query": {
4     "query_string": {
5       "query": "(genres:Drama) OR (title:kill)"
6     }
7   }
8 }

1 #! Elasticsearch built-in security features are not enabled
2 , your cluster could be accessible to anyone. See https://
3 /elasticsearch/reference/7.16/security-minimal-setup.htm
4 {
5   "took": 1,
6   "timed_out": false,
7   "_shards": {
8     "total": 1,
9     "successful": 1,
10    "skipped": 0,
11    "failed": 0
12  },
13  "hits": {
14    "total": {
15      "value": 5,
16      "relation": "eq"
17    },
18    "max_score": 1.698826,
19    "hits": [
20      {
21        "_index": "movies",
22        "_type": "movie",
23        "_id": "3",
24        "_score": 1.698826,
25        "_source": {
26          "title": "To Kill a Mockingbird",
27          "director": "Robert Mulligan",
28          "year": 1962,
29          "genres": [
```

مثال بالا را تغییر می دهیم. به فیلم هایی در عنوان آن ها استرینگ "kill" به کار رفته است، با استفاده از عملگر \wedge یا Boost امتیاز بیشتری می دهیم. این کار در نحوه نمایش رکورد ها تاثیر خواهد گذاشت.

رکوردی که هر دو شرط دو طرف OR را داشته باشد، امتیاز بیشتری نسبت به سایر رکورد ها خواهد داشت.

```
1 GET /movies/_search
2 {
3   "query": {
4     "query_string": {
5       "query": "(genres:Drama) OR (title:kill^2)"
6     }
7   }
8 }

1 #! Elasticsearch built-in security features are not
2 , your cluster could be accessible to anyone. See
3 /elasticsearch/reference/7.16/security-minimal-se
4 {
5   "took": 1,
6   "timed_out": false,
7   "_shards": {
8     "total": 1,
9     "successful": 1,
10    "skipped": 0,
11    "failed": 0
12  },
13  "hits": {
14    "total": {
15      "value": 5,
16      "relation": "eq"
17    },
18    "max_score": 3.1181374,
19    "hits": [
20      {
21        "_index": "movies",
22        "_type": "movie",
23        "_id": "3",
24        "_score": 3.1181374,
25        "_source": {
26          "title": "To Kill a Mockingbird",
27          "director": "Robert Mulligan",
28          "year": 1962,
29          "genres": [
```

در ادامه از درخت جست و جو برای کوئری های سرچ خود استفاده می کنیم.
در این کوئری ها از عبارت match استفاده می کنیم. که برابری یک فیلد و یک مقدار را بررسی می کند.

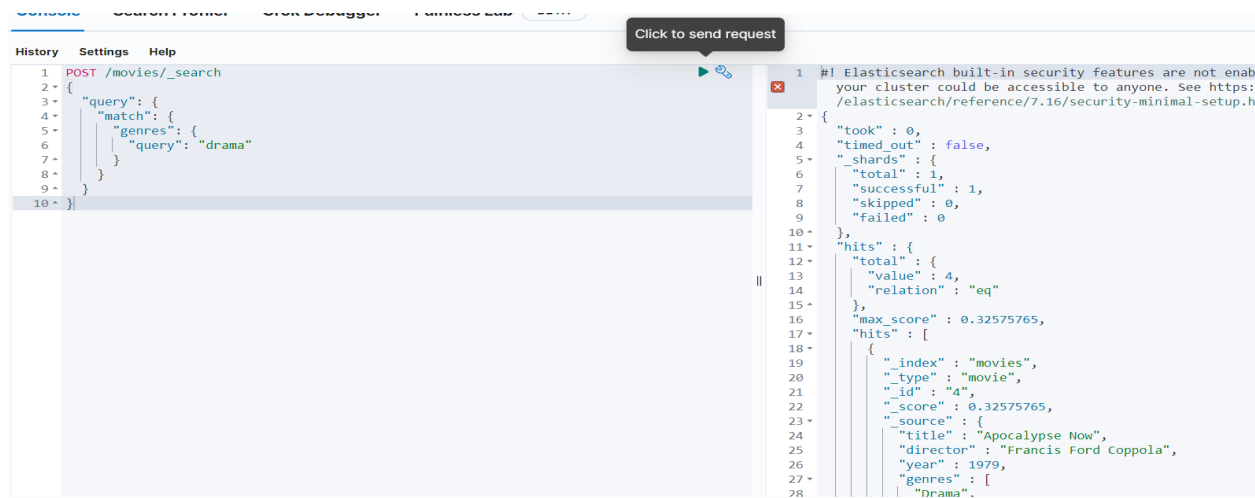
در تصویر زیر به صورت جست و جوی درختی، تمامی فیلم های ژانر درام را پیدا کرده ایم.



```
1 GET /movies/_search
2 {
3   "query": {
4     "match": {
5       "genres": "drama"
6     }
7   }
8 }

1 #! Elasticsearch built-in security features are not enabled.
2 your cluster could be accessible to anyone. See https://www.elasticsearch/reference/7.16/security-minimal-setup.html
3 {
4   "took": 0,
5   "timed_out": false,
6   "_shards": {
7     "total": 1,
8     "successful": 1,
9     "skipped": 0,
10    "failed": 0
11  },
12  "hits": {
13    "total": {
14      "value": 4,
15      "relation": "eq"
16    },
17    "max_score": 0.32575765,
18    "hits": [
19      {
20        "_index": "movies",
21        "_type": "movie",
22        "_id": "4",
23        "_score": 0.32575765,
24        "source": {
25          "title": "Apocalypse Now",
26          "director": "Francis Ford Coppola",
27          "year": 1979,
28          "genres": [
29            "Drama"
30          ]
31        }
32      }
33    ]
34  }
35 }
```

در کوئری زیر، از دستور POST استفاده کرده ایم. این حالت، حالت کلی تری است و می توانیم تنظیماتی را روی هر فیلدی اعمال کنیم.



```
1 POST /movies/_search
2 {
3   "query": {
4     "match": {
5       "genres": {
6         "query": "drama"
7       }
8     }
9   }
10 }
```

```
1 #! Elasticsearch built-in security features are not enabled.
2 your cluster could be accessible to anyone. See https://www.elasticsearch/reference/7.16/security-minimal-setup.html
3 {
4   "took": 0,
5   "timed_out": false,
6   "_shards": {
7     "total": 1,
8     "successful": 1,
9     "skipped": 0,
10    "failed": 0
11  },
12  "hits": {
13    "total": {
14      "value": 4,
15      "relation": "eq"
16    },
17    "max_score": 0.32575765,
18    "hits": [
19      {
20        "_index": "movies",
21        "_type": "movie",
22        "_id": "4",
23        "_score": 0.32575765,
24        "source": {
25          "title": "Apocalypse Now",
26          "director": "Francis Ford Coppola",
27          "year": 1979,
28          "genres": [
29            "Drama"
30          ]
31        }
32      }
33    ]
34  }
35 }
```

می توانیم در کوئری خود یک عملگر اعمال کنیم. بدین صورت عمل می کند که لغات داخل کوئری را جدا می کند. ایست واژه ها را حذف کرده و کلمات جدا شده باقی مانده با هم OR می شوند؛ یا سایر عملگرها بین کلمات اعمال می شوند.

The screenshot shows the Elasticsearch Dev Tools interface. The left pane contains a REST client request:

```
1 POST movies/_search
2 {
3   "query": {
4     "match": {
5       "title": {
6         "query": "Kill Bill",
7         "operator": "or"
8       }
9     }
10  }
11 }
```

The right pane shows the JSON response:

```
1 #! Elasticsearch built-in security features are not
2 your cluster could be accessible to anyone. See h
3 /elasticsearch/reference/7.16/security-minimal-se
4
5 {
6   "took" : 0,
7   "timed_out" : false,
8   "_shards" : {
9     "total" : 1,
10    "successful" : 1,
11    "skipped" : 0,
12    "failed" : 0
13  },
14  "hits" : {
15    "total" : {
16      "value" : 2,
17      "relation" : "eq"
18    },
19    "max_score" : 3.3008273,
20    "hits" : [
21      {
22        "_index" : "movies",
23        "_type" : "movie",
24        "_id" : "5",
25        "_score" : 3.3008273,
26        "_source" : {
27          "title" : "Kill Bill: Vol. 1",
28          "director" : "Quentin Tarantino",
29          "year" : 2003,
30          "genres" : [
31            "Action".
32          ]
33        }
34      }
35    ]
36  }
37 }
```

می توانیم از عبارت multi-match استفاده کنیم. با این کار روی دو یا چند فیلد جست و جوی خود را اعمال می کنیم. در این مثال از Boost هم برای فیلد عنوان استفاده کرده ایم.

The screenshot shows the Elasticsearch Dev Tools interface. The left pane contains a REST client request:

```
1 POST /movies/_search
2 {
3   "query": {
4     "multi_match": {
5       "query": "ford",
6       "fields": [
7         "title^3",
8         "director"
9       ]
10     }
11   }
12 }
```

The right pane shows the JSON response:

```
1 #! Elasticsearch built-in security features are not enabled. Without authentication
2 , your cluster could be accessible to anyone. See https://www.elastic.co/guide/en
3 /elasticsearch/reference/7.16/security-minimal-setup.html to enable security.
4
5 {
6   "took" : 0,
7   "timed_out" : false,
8   "_shards" : {
9     "total" : 1,
10    "successful" : 1,
11    "skipped" : 0,
12    "failed" : 0
13  },
14  "hits" : {
15    "total" : {
16      "value" : 8,
17      "relation" : "eq"
18    },
19    "max_score" : 3.3242283,
20    "hits" : [
21      {
22        "_index" : "movies",
23        "_type" : "movie",
24        "_id" : "6",
25        "_score" : 3.3242283,
26        "_source" : {
27          "title" : "The Assassination of Jesse James by the Coward Robert Ford",
28          "director" : "Andrew Dominik",
29          "year" : 2007,
30          "genres" : [
31            "Biography"
32          ]
33        }
34      }
35    ]
36  }
37 }
```

در مثال زیر، چند عبارت match را با AND و OR ترکیب کرده ایم و از عبارت bool برای جست و جوی استفاده کرده ایم.

در عبارت منطقی و برای ساخت کوئری سرچ های پیشرفته، عبارت must را با AND، عبارت should را با OR و عبارت must_not را با NOT معادل فرض کنید.

و اینکه در مقابل یک bool، هر کدام از عبارات must و should و must_not تنها یک بار اجازه دارند تا حضور داشته باشند.

```
Console Search Promter Crok Debugger Farness Lab BETA
Click to send request

History Settings Help
1 POST /movies/_search
2 {
3   "query": {
4     "bool": {
5       "must": [
6         {
7           "match": {
8             "genres": "drama"
9           }
10        },
11        {
12          "match": {
13            "title": {
14              "query": "kill",
15              "boost": 3
16            }
17          }
18        }
19      ]
20    }
21  }
22 }
```

```
1 #! Elasticsearch built-in security features are not enabled. W
your cluster could be accessible to anyone. See https://www.e
/elasticsearch/reference/7.16/security-minimal-setup.html to
2 {
3   "took": 0,
4   "timed_out": false,
5   "shards": {
6     "total": 1,
7     "successful": 1,
8     "skipped": 0,
9     "failed": 0
10  },
11  "hits": {
12    "total": {
13      "value": 1,
14      "relation": "eq"
15    },
16    "max_score": 4.537449,
17    "hits": [
18      {
19        "_index": "movies",
20        "_type": "movie",
21        "_id": "3",
22        "_score": 4.537449,
23        "_source": {
24          "title": "To Kill a Mockingbird",
25          "director": "Robert Mulligan",
26          "year": 1962,
27          "genres": [
28            "crime"
29          ]
30        }
31      }
32    ]
33  }
34 }
```

مثال زیر، فیلم هایی که در عنوان آن ها کلمه "kill" یا "ford" به کار رفته و ژانر آن ها اسرارآمیز نیست را پیدا کرده ایم.

مشاهده می کنیم چون که در مقابل یک bool، هر کدام از عبارات must و should و must_not تنها یک بار اجازه دارند تا حضور داشته باشند، از bool های تو در تو استفاده کرده ایم.

```
1 GET /movies/_search
2 {
3   "query": {
4     "bool": {
5       "must": {
6         "bool": {
7           "should": [
8             {
9               "match": {
10                "title": "ford"
11              }
12            },
13            {
14              "match": {
15                "title": "kill"
16              }
17            }
18          ]
19        }
20      },
21      "must_not": {
22        "match": {
23          "genres": "Mystery"
24        }
25      }
26    }
27  }
28 }
```

```
1 #! Elasticsearch built-in security features are not enabled.
your cluster could be accessible to anyone. See https://www
/elasticsearch/reference/7.16/security-minimal-setup.html t
2 {
3   "took": 1,
4   "timed_out": false,
5   "shards": {
6     "total": 1,
7     "successful": 1,
8     "skipped": 0,
9     "failed": 0
10  },
11  "hits": {
12    "total": {
13      "value": 2,
14      "relation": "eq"
15    },
16    "max_score": 1.4193115,
17    "hits": [
18      {
19        "_index": "movies",
20        "_type": "movie",
21        "_id": "5",
22        "_score": 1.4193115,
23        "_source": {
24          "title": "Kill Bill: Vol. 1",
25          "director": "Quentin Tarantino",
26          "year": 2003,
27          "genres": [
28            "crime"
29          ]
30        }
31      }
32    ]
33  }
34 }
```

برای کوئری سرچ های غیر متنی از عبارت term استفاده می کنیم.

این نوع کوئری ها در کوئری سرچ های متنی هم بر حسب نیاز کاربرد خواهند داشت.

```
ry Settings Help
GET /movies/_search
{
  "query": {
    "term": {
      "year": {
        "value": "2003",
        "boost": 1.0
      }
    }
  }
}

1 #! Elasticsearch built-in security features are not enabled in your cluster.
2 {
3   "took": 0,
4   "timed_out": false,
5   "_shards": {
6     "total": 1,
7     "successful": 1,
8     "skipped": 0,
9     "failed": 0
10  },
11  "hits": {
12    "total": {
13      "value": 1,
14      "relation": "eq"
15    },
16    "max_score": 1.0,
17    "hits": [
18      {
19        "_index": "movies",
20        "_type": "movie",
21        "_id": "5",
22        "_score": 1.0,
23        "_source": {
24          "title": "Kill Bill: Vol. 1",
25          "director": "Quentin Tarantino",
26          "year": 2003,
27          "genres": [
```

در مثال زیر می خواهیم از فیلتر کردن رکورد ها استفاده کنیم تا جست و جو محدود تر و با سرعت بیشتری صورت پذیرد. برای این کار از عبارت filter استفاده می کنیم. در مثال زیر فیلم های بعد از سال 1960 و ژانر درام را فیلتر کرده و سپس فیلم هایی که در عنوان آن ها کلمه "kill" به کار رفته است را پیدا می کنیم.

```
Console Search Profiler Grok Debugger Painless Lab BETA
Click to send request
History Settings Help
1 GET /movies/_search
2 {
3   "query": {
4     "bool": {
5       "must": [
6         {
7           "match": {
8             "title": "kill"
9           }
10        },
11      ],
12      "filter": {
13        "bool": {
14          "must": [
15            {
16              "range": {
17                "year": {
18                  "gte": 1960
19                }
20              },
21            },
22            {
23              "term": {
24                "genres": {
25                  "value": "drama"
26                }
27              }
28            }
29          ]
30        }
31      }
32    }
33  }
34}
```

```
1 #! Elasticsearch built-in security features are not enabled in your cluster.
2 {
3   "took": 0,
4   "timed_out": false,
5   "_shards": {
6     "total": 1,
7     "successful": 1,
8     "skipped": 0,
9     "failed": 0
10  },
11  "hits": {
12    "total": {
13      "value": 1,
14      "relation": "eq"
15    },
16    "max_score": 1.4193115,
17    "hits": [
18      {
19        "_index": "movies",
20        "_type": "movie",
21        "_id": "3",
22        "_score": 1.4193115,
23        "_source": {
24          "title": "To Kill a Mockingbird",
25          "director": "Robert Mulligan",
26          "year": 1962,
27          "genres": [
28            "crime"
29          ]
30        }
31      }
32    ]
33  }
34}
```

نحوه دیگر فیلتر کردن را در مثال زیر بررسی می کنیم. در این جا از عبارت filter استفاده نکرده ایم. این مثال خلاصه شده مثال قبل است.

```
History Settings Help
1 GET /movies/_search
2 {
3   "query": {
4     "bool": {
5       "must": [
6         {
7           "match": {
8             "title": "kill"
9           }
10        },
11        {
12          "range": {
13            "year": {
14              "gte": 1960
15            }
16          }
17        },
18        {
19          "term": {
20            "genres": {
21              "value": "drama"
22            }
23          }
24        }
25      ]
26    }
27  }
28 }
```

Click to send request

```
1 #! Elasticsearch built-in security features are not
2 your cluster could be accessible to anyone. See http
3 /elasticsearch/reference/7.16/security-minimal-
4
5 {
6   "took" : 0,
7   "timed_out" : false,
8   "_shards" : {
9     "total" : 1,
10    "successful" : 1,
11    "skipped" : 0,
12    "failed" : 0
13  },
14  "hits" : {
15    "total" : {
16      "value" : 1,
17      "relation" : "eq"
18    },
19    "max_score" : 2.6988258,
20    "hits" : [
21      {
22        "_index" : "movies",
23        "_type" : "movie",
24        "_id" : "3",
25        "_score" : 2.6988258,
26        "_source" : {
27          "title" : "To Kill a Mockingbird",
28          "director" : "Robert Mulligan",
29          "year" : 1962,
30          "genres" : [
31            "crime".
32          ]
33        }
34      }
35    ]
36  }
37 }
```

دو مثال زیر تغییر یافته مثال بالا هستند. بدین صورت که در مثال اول، پس از اعمال فیلتر از match_all استفاده کرده ایم تا تمام رکورد های فیلتر شده برگردانده شوند. و در مثال دوم، بخش جست و جو را حذف کرده ایم تا تنها رکورد ها فیلتر شوند.

```
Console Search Profiler Grok Debugger Painless Lab BETA
History Settings Help
1 GET /movies/_search
2 {
3   "query": {
4     "bool": {
5       "must": [
6         {
7           "match_all": {}
8         },
9         {
10          "range": {
11            "year": {
12              "gte": 1960
13            }
14          }
15        },
16        {
17          "term": {
18            "genres": {
19              "value": "drama"
20            }
21          }
22        }
23      ]
24    }
25  }
26 }
```

Click to send request

```
1 #! Elasticsearch built-in security features are not en
2 your cluster could be accessible to anyone. See http
3 /elasticsearch/reference/7.16/security-minimal-setup
4
5 {
6   "took" : 0,
7   "timed_out" : false,
8   "_shards" : {
9     "total" : 1,
10    "successful" : 1,
11    "skipped" : 0,
12    "failed" : 0
13  },
14  "hits" : {
15    "total" : {
16      "value" : 4,
17      "relation" : "eq"
18    },
19    "max_score" : 2.3257577,
20    "hits" : [
21      {
22        "_index" : "movies",
23        "_type" : "movie",
24        "_id" : "4",
25        "_score" : 2.3257577,
26        "_source" : {
27          "title" : "Apocalypse Now",
28          "director" : "Francis Ford Coppola",
29          "year" : 1979,
30          "genres" : [
31            "Drama".
32          ]
33        }
34      }
35    ]
36  }
37 }
```

```
Console Search Profiler Grok Debugger Painless Lab BETA
Click to send request

History Settings Help
1 GET /movies/_search
2 {
3   "query": {
4     "bool": {
5       "must": [
6         {
7           "range": {
8             "year": {
9               "gte": 2000
10            }
11          },
12        },
13        {
14          "term": {
15            "genres": {
16              "value": "drama"
17            }
18          }
19        }
20      ]
21    }
22  }
23 }
```

```
1 #! Elasticsearch built-in security features are not enabled. Without authentication,
2 your cluster could be accessible to anyone. See https://www.elastic.co/guide/en
3 /elasticsearch/reference/7.16/security-minimal-setup.html to enable security.
4 {
5   "took": 0,
6   "timed_out": false,
7   "_shards": {
8     "total": 1,
9     "successful": 1,
10    "skipped": 0,
11    "failed": 0
12  },
13  "hits": {
14    "total": {
15      "value": 1,
16      "relation": "eq"
17    },
18    "max_score": 1.2795144,
19    "hits": [
20      {
21        "_index": "movies",
22        "_type": "movie",
23        "_id": "6",
24        "_score": 1.2795144,
25        "_source": {
26          "title": "The Assassination of Jesse James by the Coward Robert Ford",
27          "director": "Andrew Dominik",
28          "year": 2007,
29          "genres": [

```

اگر در کلمات مورد جست و جو تا حدی غلط های املایی را هم نادیده بگیریم می توانیم از عبارت fuzziness استفاده کنیم. این عمل باعث می شود به صورت حدودی و تقریبی جست و جو به درستی کار کند. در مثال زیر که کلمه پدرخوانده دارای غلط نگارشی است ولی جست و جو توانسته آن را پیدا کند.

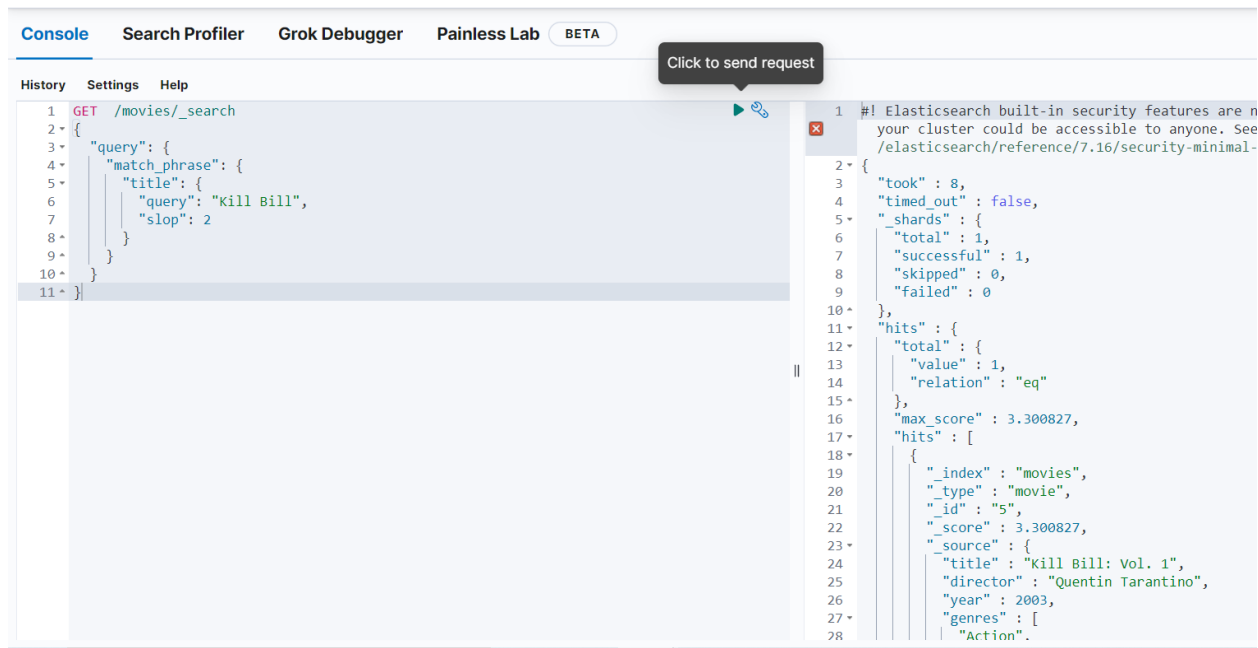
```
Console Search Profiler Grok Debugger Painless Lab BETA
Click to send request

History Settings Help
1 GET /movies/_search
2 {
3   "query": {
4     "match": {
5       "title": {
6         "query": "godfater",
7         "fuzziness": "AUTO"
8       }
9     }
10  }
11 }
```

```
1 #! Elasticsearch built-in security features are not enabled. Without authentication,
2 your cluster could be accessible to anyone. See https://www.elastic.co/guide/en
3 /elasticsearch/reference/7.16/security-minimal-setup.html to enable security.
4 {
5   "took": 19,
6   "timed_out": false,
7   "_shards": {
8     "total": 1,
9     "successful": 1,
10    "skipped": 0,
11    "failed": 0
12  },
13  "hits": {
14    "total": {
15      "value": 6,
16      "relation": "eq"
17    },
18    "max_score": 0.6325867,
19    "hits": [
20      {
21        "_index": "movies",
22        "_type": "movie",
23        "_id": "wk526X0B1Y1SMsGxx3L3",
24        "_score": 0.6325867,
25        "_source": {
26          "title": "The Godfather",
27          "director": "Francis Ford Coppola",
28          "year": 1972
29        }
30      }
31    ]
32  }
33 }
```

اگر نیاز داشته باشیم تا فیلد های مورد جست و جوی ما شامل عبارتی باشد که به صورت عینی باید آمده باشد و کلمات آن عبارت به صورت پشت سرهم و بی فاصله باشند، از مثال زیر می توانیم استفاده کنیم.

که در آن از `match_phrase` استفاده شده و با `slop` فضای خالی بین کلمات تا کلمات بعدی را تنظیم کرده ایم.

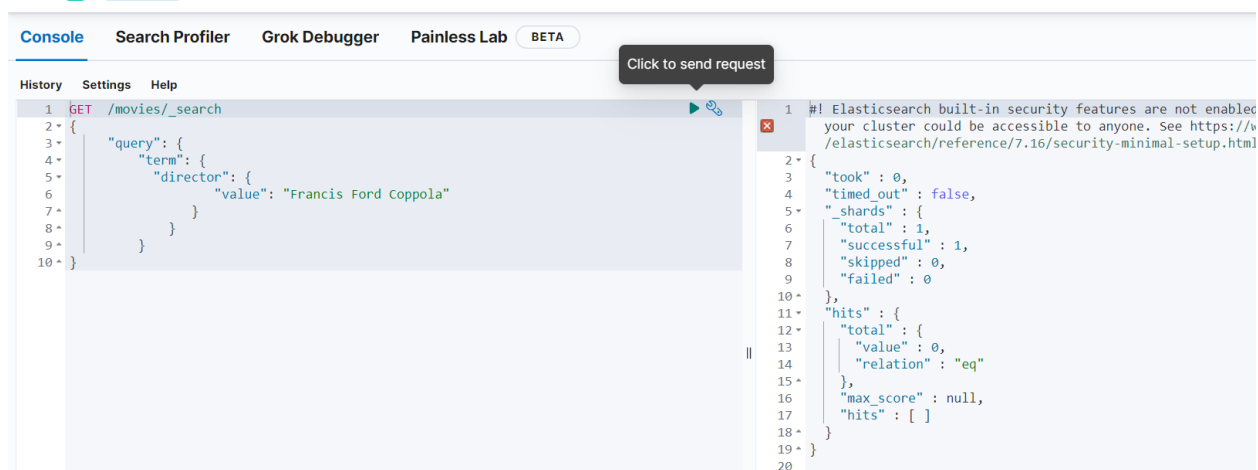


The screenshot shows the Elasticsearch console interface. On the left, a GET request is sent to `/movies/_search` with a `match_phrase` query for the title "Kill Bill" with a `slop` of 2. On the right, the response is displayed, showing a single hit for the movie "Kill Bill: Vol. 1" by Quentin Tarantino, with a score of 3.300827.

```
1 GET /movies/_search
2 {
3   "query": {
4     "match_phrase": {
5       "title": {
6         "query": "Kill Bill",
7         "slop": 2
8       }
9     }
10  }
11 }
```

```
1 #! Elasticsearch built-in security features are not enabled
2 your cluster could be accessible to anyone. See https://www.elastic.co/guide/en/elasticsearch/reference/7.16/security-minimal-setup.html
3 {
4   "took": 8,
5   "timed_out": false,
6   "_shards": {
7     "total": 1,
8     "successful": 1,
9     "skipped": 0,
10    "failed": 0
11  },
12  "hits": {
13    "total": {
14      "value": 1,
15      "relation": "eq"
16    },
17    "max_score": 3.300827,
18    "hits": [
19      {
20        "_index": "movies",
21        "_type": "movie",
22        "_id": "5",
23        "_score": 3.300827,
24        "_source": {
25          "title": "Kill Bill: Vol. 1",
26          "director": "Quentin Tarantino",
27          "year": 2003,
28          "genres": [
29            "Action"
30          ]
31        }
32      }
33    ]
34  }
35 }
```

یک مشکل رایج در کوئری سرچ های الستیک اینگونه است که اگر طبق مثال زیر فیلم ها را با کارگردان مورد نظر خود سرچ کنیم، با اینکه جواب در دیتابیس موجود است ولی آن را پیدا نمی کنیم. چون الستیک متن ها را آنالیز و پردازش کرده و آن ها را به کلمات می شکند. برای حل این مشکل باید کل متن را ایندکس کرده و نوع آن را keyword قرار داد.

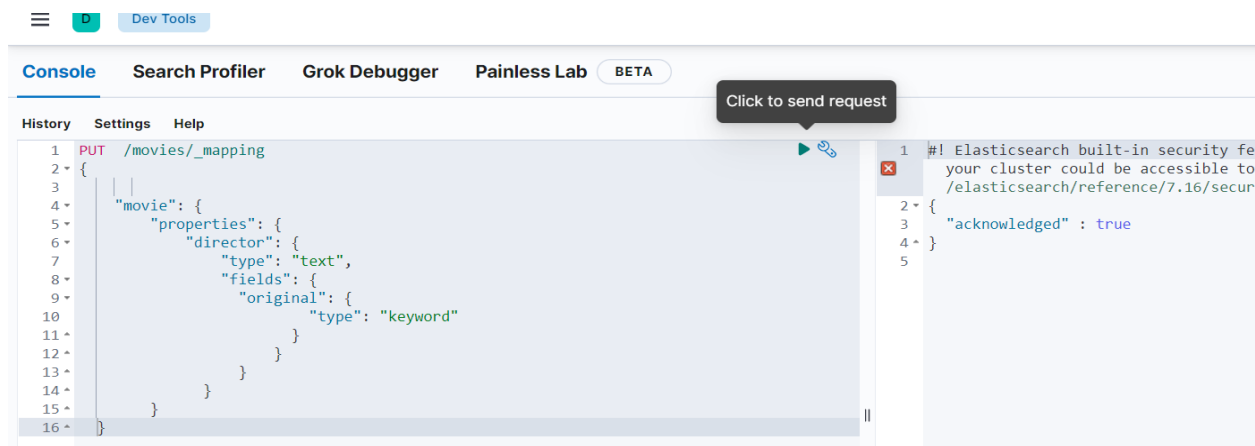


The screenshot shows the Elasticsearch console interface. On the left, a GET request is sent to `/movies/_search` with a `term` query for the director "Francis Ford Coppola". On the right, the response is displayed, showing no hits found.

```
1 GET /movies/_search
2 {
3   "query": {
4     "term": {
5       "director": {
6         "value": "Francis Ford Coppola"
7       }
8     }
9   }
10 }
```

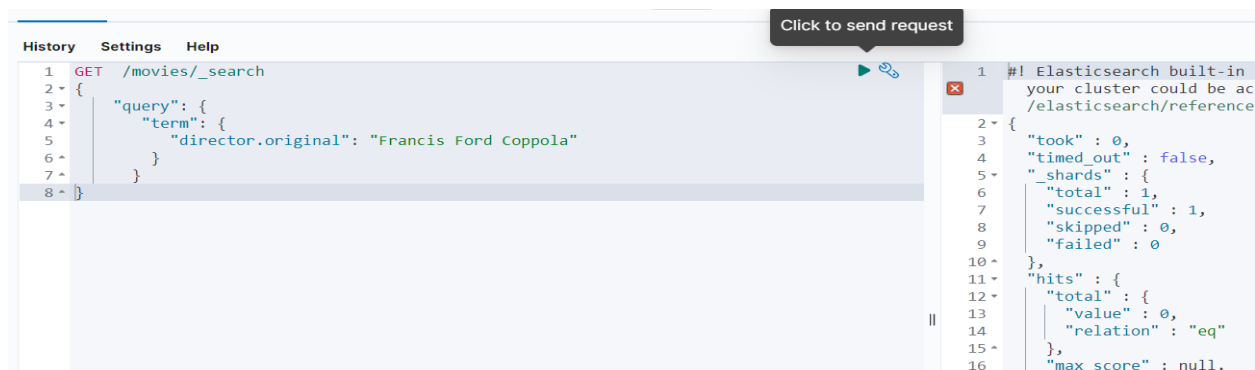
```
1 #! Elasticsearch built-in security features are not enabled
2 your cluster could be accessible to anyone. See https://www.elastic.co/guide/en/elasticsearch/reference/7.16/security-minimal-setup.html
3 {
4   "took": 0,
5   "timed_out": false,
6   "_shards": {
7     "total": 1,
8     "successful": 1,
9     "skipped": 0,
10    "failed": 0
11  },
12  "hits": {
13    "total": {
14      "value": 0,
15      "relation": "eq"
16    },
17    "max_score": null,
18    "hits": [ ]
19  }
20 }
```

راه حل الستیک برای حل این مشکل استفاده از فیلد های چند مقداره است. تا علاوه بر مقدار اصلی آن ها، روی نوع دیگری از ایندکس ها آنالیز و پردازش شود. راه حل دیگر استفاده از field.keyword است که برای همه فیلد ها در دسترس است.



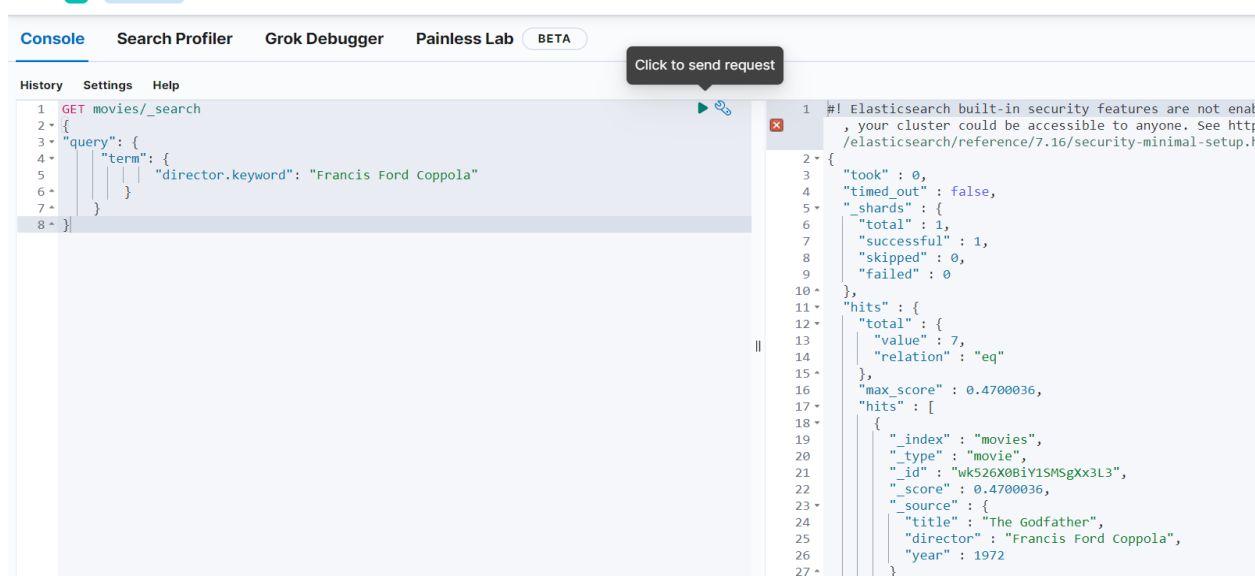
```
1 PUT /movies/_mapping
2 {
3   {
4     "movie": {
5       "properties": {
6         "director": {
7           "type": "text",
8           "fields": {
9             "original": {
10              "type": "keyword"
11            }
12          }
13        }
14      }
15    }
16  }
```

```
1 #! Elasticsearch built-in security fe
2 your cluster could be accessible to
3 /elasticsearch/reference/7.16/secur
4 {
5   "acknowledged" : true
6 }
```



```
1 GET /movies/_search
2 {
3   "query": {
4     "term": {
5       "director.original": "Francis Ford Coppola"
6     }
7   }
8 }
```

```
1 #! Elasticsearch built-in
2 your cluster could be ac
3 /elasticsearch/reference
4 {
5   "took" : 0,
6   "timed_out" : false,
7   "_shards" : {
8     "total" : 1,
9     "successful" : 1,
10    "skipped" : 0,
11    "failed" : 0
12  },
13  "hits" : {
14    "total" : {
15      "value" : 0,
16      "relation" : "eq"
17    },
18    "max_score" : null,
19  }
```



```
1 GET /movies/_search
2 {
3   "query": {
4     "term": {
5       "director.keyword": "Francis Ford Coppola"
6     }
7   }
8 }
```

```
1 #! Elasticsearch built-in security features are not enab
2 , your cluster could be accessible to anyone. See http
3 /elasticsearch/reference/7.16/security-minimal-setup.f
4 {
5   "took" : 0,
6   "timed_out" : false,
7   "_shards" : {
8     "total" : 1,
9     "successful" : 1,
10    "skipped" : 0,
11    "failed" : 0
12  },
13  "hits" : {
14    "total" : {
15      "value" : 7,
16      "relation" : "eq"
17    },
18    "max_score" : 0.4700036,
19    "hits" : [
20      {
21        "_index" : "movies",
22        "_type" : "movie",
23        "_id" : "wkS26X08iY1SMsGxx3L3",
24        "_score" : 0.4700036,
25        "_source" : {
26          "title" : "The Godfather",
27          "director" : "Francis Ford Coppola",
28          "year" : 1972
29        }
30      }
31    ]
32  }
```


در ادامه index pattern و داشبورد مورد نیاز خود را ایجاد کردیم. و یک بار به صورت ابر کلمات و بار دیگر به صورت pie ویژوال می کنیم.

- Tag Cloud:



●

