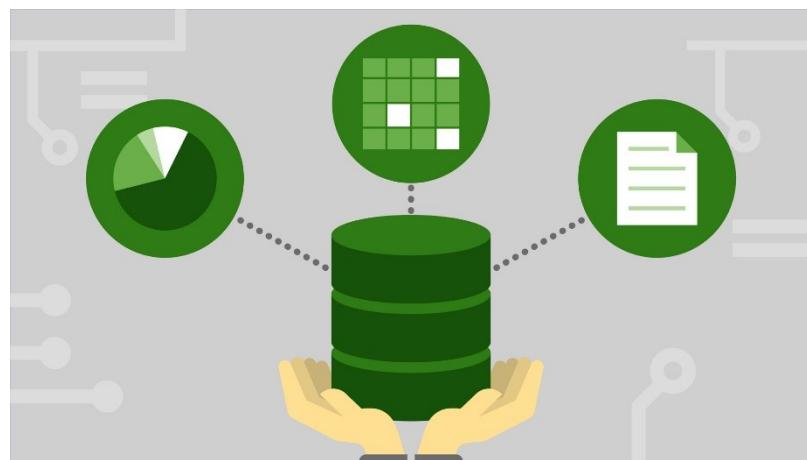


به نام خدا



دانشگاه تهران  
پردیس دانشکده‌های فنی  
دانشکده برق و کامپیوتر



## آزمایشگاه پایگاه داده

ستور کار شماره ۷

آتیه آرمین

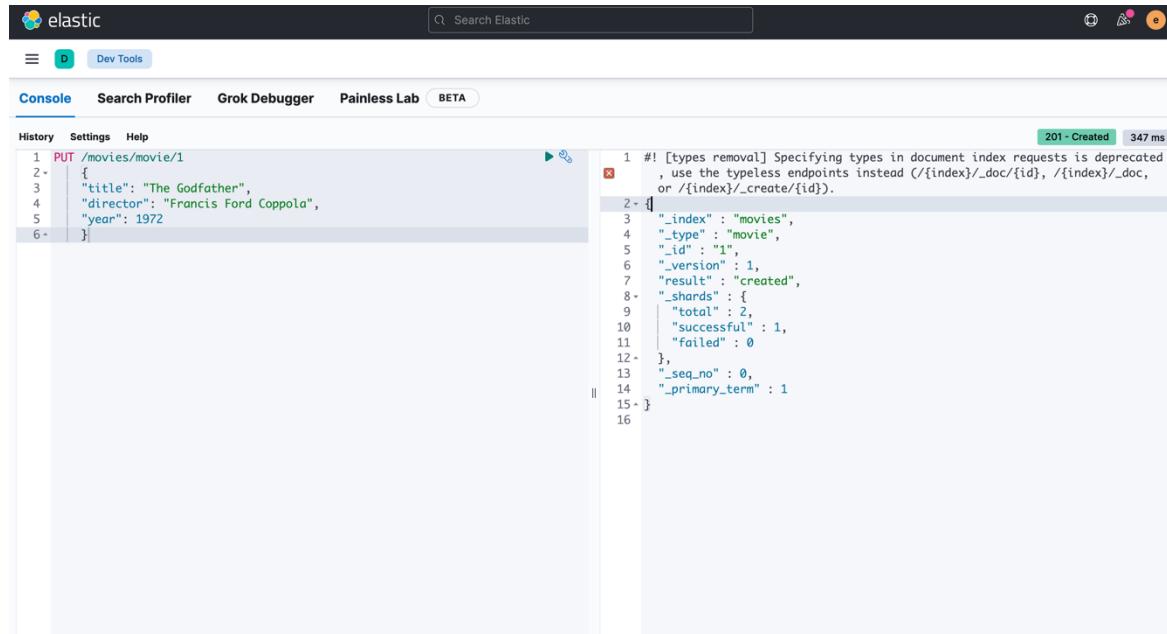
۸۱۰۱۹۷۶۴۸

مهرماه ۱۴۰۰

## گزارش دستورکار انجام شده

### بخش اول

تکه کد داده شده در واقع برای اضافه کردن یک رکورد است. در خروجی نیز index آن و id آن مشخص است. در دومین اجرا version و نوع دستور تغییر کرد. در این حالت اگر فیلد جدیدی به داده ها اضافه کنیم و یا یکی از فیلد ها را تغییر دهیم و دستور را اجرا کنیم، یک نسخه جدید از داده ها ایجاد و ذخیره می شود.



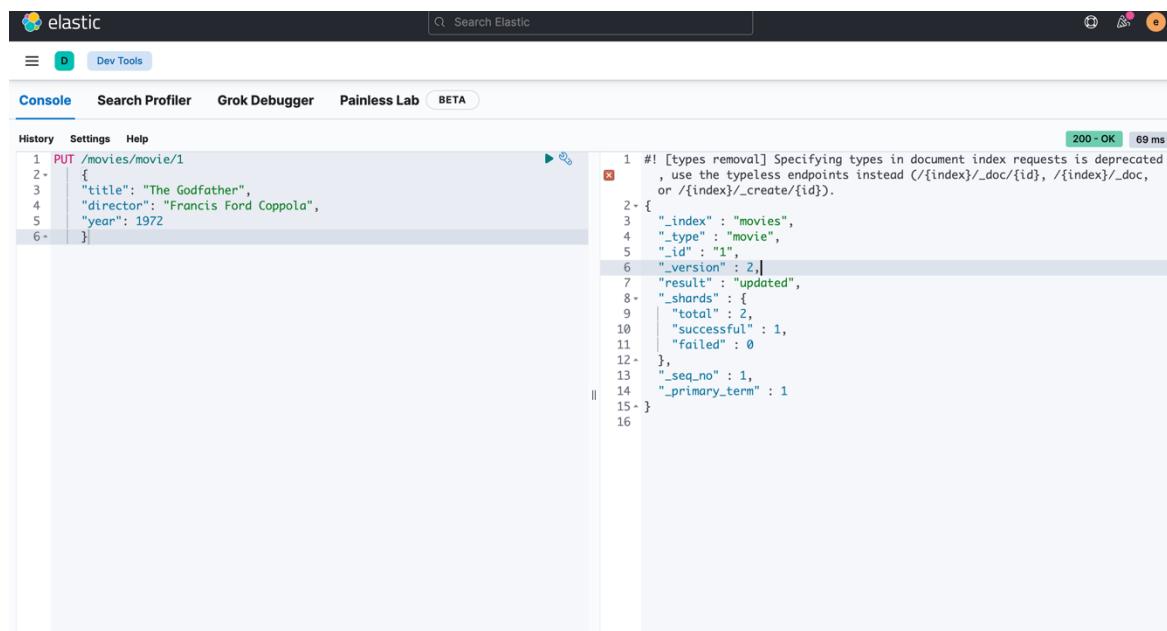
```

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

201 - Created 347 ms

```

1 #! [types removal] Specifying types in document index requests is deprecated
   , use the typeless endpoints instead (/index/_doc/{id}, /{index}/_doc,
   or /{index}/_create/{id}).
2 {
3   "_index" : "movies",
4   "_type" : "movie",
5   "_id" : "1",
6   "_version" : 1,
7   "result" : "created",
8   "_shards" : {
9     "total" : 2,
10    "successful" : 1,
11    "failed" : 0
12  },
13  "_seq_no" : 0,
14  "_primary_term" : 1
15 }
16
  
```



```

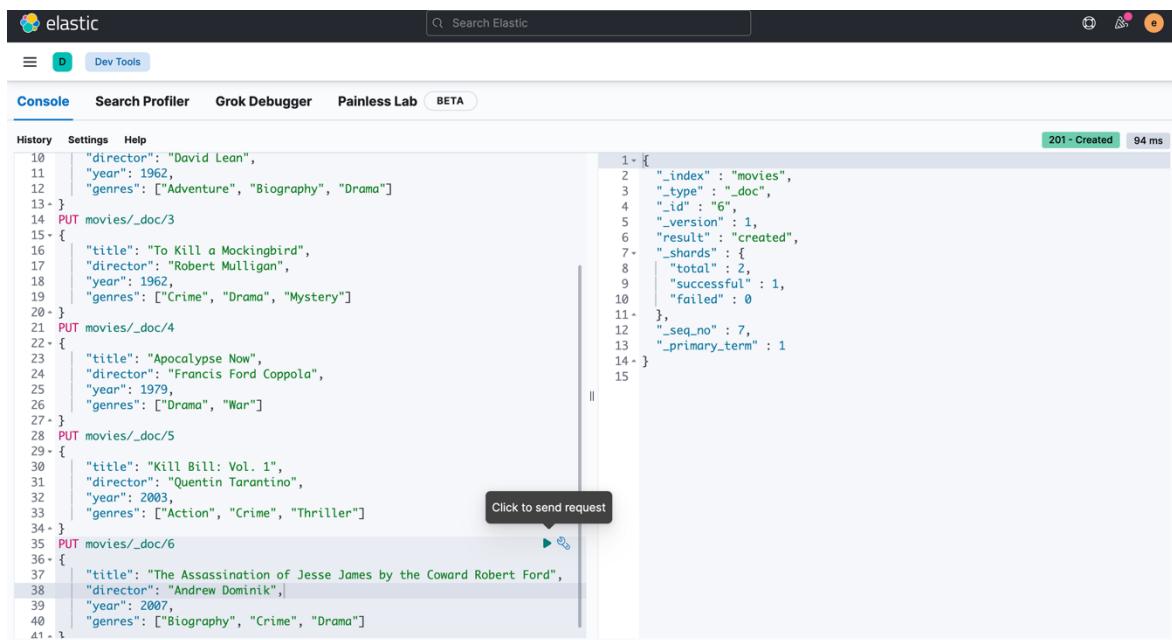
PUT /movies/movie/1
{
  "title": "The Godfather",
  "director": "Francis Ford Coppola",
  "year": 1972
}
  
```

200 - OK 69 ms

```

1 #! [types removal] Specifying types in document index requests is deprecated
   , use the typeless endpoints instead (/index/_doc/{id}, /{index}/_doc,
   or /{index}/_create/{id}).
2 {
3   "_index" : "movies",
4   "_type" : "movie",
5   "_id" : "1",
6   "_version" : 2,
7   "result" : "updated",
8   "_shards" : {
9     "total" : 2,
10    "successful" : 1,
11    "failed" : 0
12  },
13  "_seq_no" : 1,
14  "_primary_term" : 1
15 }
16
  
```

دستورات زیر رکورد های مختلف را اضافه میکنند. چون id و شناسه رکورد ها مشخص است از `put` استفاده شده است.

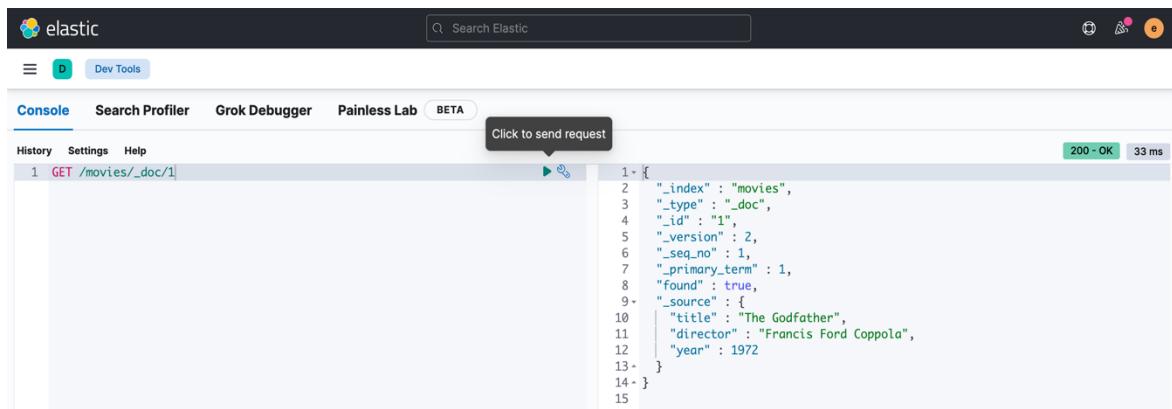


```

10 "director": "David Lean",
11 "year": 1962,
12 "genres": ["Adventure", "Biography", "Drama"]
13 }
14 PUT movies/_doc/3
15 {
16   "title": "To Kill a Mockingbird",
17   "director": "Robert Mulligan",
18   "year": 1962,
19   "genres": ["Crime", "Drama", "Mystery"]
20 }
21 PUT movies/_doc/4
22 {
23   "title": "Apocalypse Now",
24   "director": "Francis Ford Coppola",
25   "year": 1979,
26   "genres": ["Drama", "War"]
27 }
28 PUT movies/_doc/5
29 {
30   "title": "Kill Bill: Vol. 1",
31   "director": "Quentin Tarantino",
32   "year": 2003,
33   "genres": ["Action", "Crime", "Thriller"]
34 }
35 PUT movies/_doc/6
36 {
37   "title": "The Assassination of Jesse James by the Coward Robert Ford",
38   "director": "Andrew Dominik",
39   "year": 2007,
40   "genres": ["Biography", "Crime", "Drama"]
41 }
```

Click to send request

برای جست و جوی اطلاعات یک فیلم خاص با شناسه مشخص از دستور زیر استفاده میکنیم که برای پیدا کردن یک فیلم با شناسه ۱ است. همانطور که مشخص است `found` برابر با `true` است که نشان دهنده وجود این شناسه است.



```

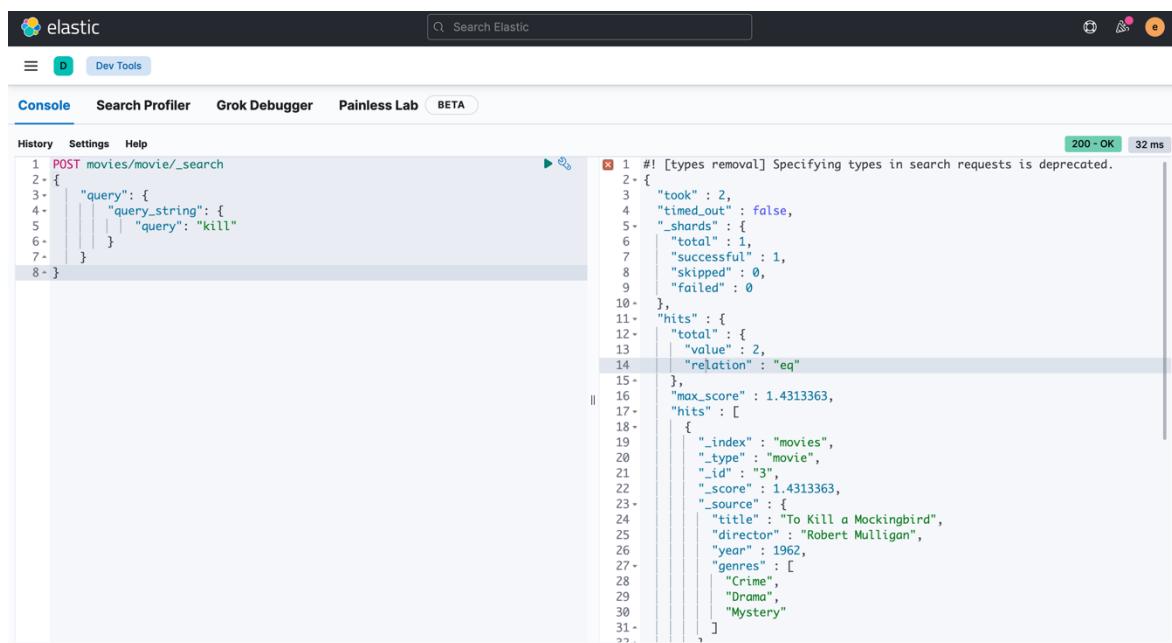
1 GET /movies/_doc/1
```

Click to send request

```

1 {
2   "_index": "movies",
3   "_type": "_doc",
4   "_id": "1",
5   "_version": 2,
6   "_seq_no": 1,
7   "_primary_term": 1,
8   "found": true,
9   "_source": {
10     "title": "The Godfather",
11     "director": "Francis Ford Coppola",
12     "year": 1972
13   }
14 }
15 }
```

برای جست و جو های متى از `query_string` استفاده میشود. کافی است کلمه یا عبارت مورد نظر خود را وارد کرده و فیلتر ها و شرط ها را هم در `query` مشخص کنیم. دستور زیر هر فیلمی که در هر فیلدی از آن کلمه `kill` به کار رفته باشد را پیدا میکند.



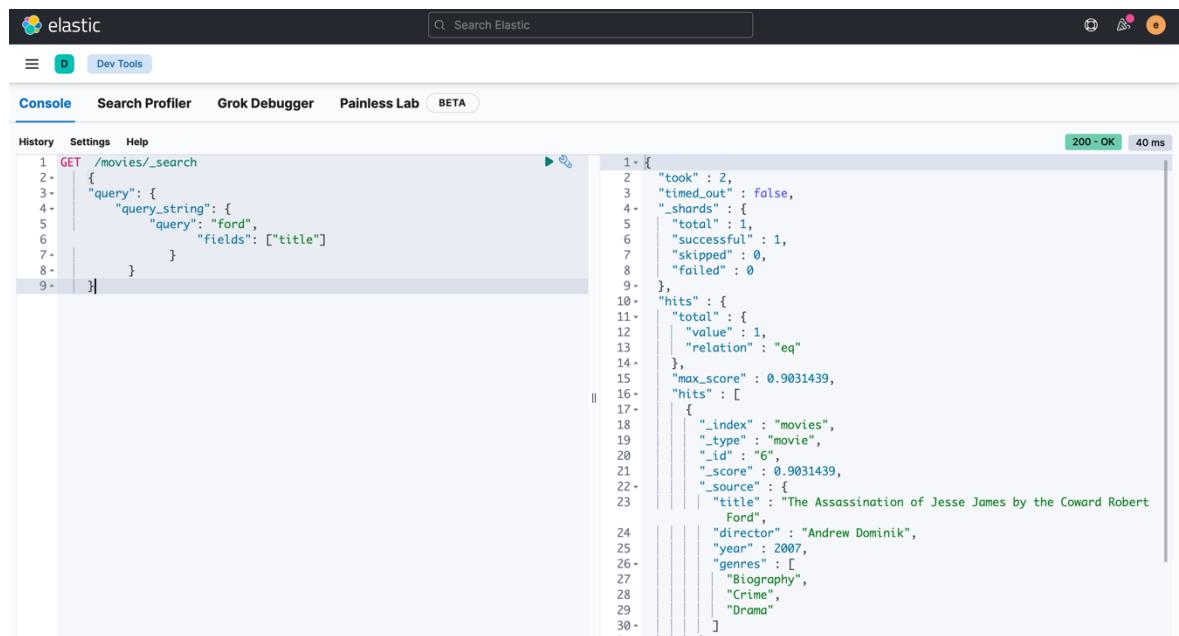
```

1 POST movies/movie/_search
2 {
3   "query": {
4     "query_string": {
5       "query": "kill"
6     }
7   }
8 }
```

```

1 #! [types removal] Specifying types in search requests is deprecated.
2 {
3   "took": 2,
4   "timed_out": false,
5   "_shards": {
6     "total": 1,
7     "successful": 1,
8     "skipped": 0,
9     "failed": 0
10 },
11 },
12 "hits": {
13   "total": {
14     "value": 2,
15     "relation": "eq"
16   },
17   "max_score": 1.4313363,
18   "hits": [
19     {
20       "_index": "movies",
21       "_type": "movie",
22       "_id": "3",
23       "_score": 1.4313363,
24       "_source": {
25         "title": "To Kill a Mockingbird",
26         "director": "Robert Mulligan",
27         "year": 1962,
28         "genres": [
29           "Crime",
30           "Drama",
31           "Mystery"
32         ]
33       }
34     }
35   ]
36 }
```

در دستور زیر، جست و جوی خود را به یک فیلد خاص یعنی title محدود کرده و به دنبال فیلم هایی هستیم که کلمه ford در آن ها به کار برده شده است.



```

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

```

1 {
2   "took": 2,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10 },
11 "hits": {
12   "total": {
13     "value": 1,
14     "relation": "eq"
15   },
16   "max_score": 0.9031439,
17   "hits": [
18     {
19       "_index": "movies",
20       "_type": "movie",
21       "_id": "6",
22       "_score": 0.9031439,
23       "_source": {
24         "title": "The Assassination of Jesse James by the Coward Robert Ford",
25         "director": "Andrew Dominik",
26         "year": 2007,
27         "genres": [
28           "Biography",
29           "Crime",
30           "Drama"
31         ]
32       }
33     }
34   ]
35 }
```

در ادامه بقیه جستو جو های نام برده را بررسی میکنیم.

تمام فیلم‌هایی که در نام آنها `kill` یا `ford` به کار رفته است:

```

1 GET /movies/_search
2 {
3   "query": {"query_string": {"query": "title:ford OR kill"}}
4 }
5
1+ {
2   "took": 9,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 3,
13      "relation": "eq"
14    },
15    "max_score": 1.4313363,
16    "hits": [
17      {
18        "_index": "movies",
19        "_type": "movie",
20        "_id": "3",
21        "_score": 1.4313363,
22        "_source": {
23          "title": "To Kill a Mockingbird",
24          "director": "Robert Mulligan",
25          "year": 1962,
26          "genres": [
27            "Crime",
28            "Drama",
29            "Mystery"
30          ]
31        }
32      }
33    ]
34  }
35
  
```

تمام فیلم‌هایی که عنوان `kill` داشته و کارگردان آنها `Tarantino` است:

```

1 GET /movies/_search
2 {
3   "query": {"query_string": {"query": "(title:kill) AND (director :Tarantino)"}}
4 }
5
1+ {
2   "took": 10,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 1,
13      "relation": "eq"
14    },
15    "max_score": 3.4809031,
16    "hits": [
17      {
18        "_index": "movies",
19        "_type": "movie",
20        "_id": "5",
21        "_score": 3.4809031,
22        "_source": {
23          "title": "Kill Bill: Vol. 1",
24          "director": "Quentin Tarantino",
25          "year": 2003,
26          "genres": [
27            "Action",
28            "Crime",
29            "Thriller"
30          ]
31        }
32      }
33    ]
34  }
35
  
```

تمام فیلم های ژانر درام در سال ۱۹۶۲ :

```

Click to send request
1 GET /movies/_search
2 {
3   "query": {"query_string": {"query": "(genres:Drama) AND (year:1962 )"}}
4 }
5
1- [
2   "took" : 1,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 1,
6     "successful" : 1,
7     "skipped" : 0,
8     "failed" : 0
9   },
10  "hits" : {
11    "total" : {
12      "value" : 2,
13      "relation" : "eq"
14    },
15    "max_score" : 1.2354951,
16    "hits" : [
17      {
18        "_index" : "movies",
19        "_type" : "movie",
20        "_id" : "2",
21        "_score" : 1.2354951,
22        "_source" : {
23          "title" : "Lawrence of Arabia",
24          "director" : "David Lean",
25          "year" : 1962,
26          "genres" : [
27            "Adventure",
28            "Biography",
29            "Drama"
30          ]
31        }
32      }
33    ]
34  ]
35
36  200 - OK  31 ms
  
```

تمام فیلم های ژانر درام تا سال ۲۰۰۵ :

```

Click to send request
1 GET /movies/_search
2 {
3   "query": {"query_string": {"query": "(genres:Drama) AND (year:[* TO 2005])"}}
4 }
5
1- [
2   "took" : 1,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 1,
6     "successful" : 1,
7     "skipped" : 0,
8     "failed" : 0
9   },
10  "hits" : {
11    "total" : {
12      "value" : 3,
13      "relation" : "eq"
14    },
15    "max_score" : 1.2741477,
16    "hits" : [
17      {
18        "_index" : "movies",
19        "_type" : "movie",
20        "_id" : "4",
21        "_score" : 1.2741477,
22        "_source" : {
23          "title" : "Apocalypse Now",
24          "director" : "Francis Ford Coppola",
25          "year" : 1979,
26          "genres" : [
27            "Drama",
28            "War"
29          ]
30        }
31      }
32    ]
33  ]
34
35  200 - OK  29 ms
  
```

تمام فیلم های درام یا فیلم هایی که در نام آن ها کلمه kill به کار رفته است:

The screenshot shows the Elastic Dev Tools interface with the following details:

- Header:** elastic, Search Elastic, Dev Tools
- Navigation:** Console (selected), Search Profiler, Grok Debugger, Painless Lab (BETA)
- Request:** History, Settings, Help
- Request Body:** A GET request to /movies/\_search with a query string filter for "Drama" or "Kill".
- Request Buttons:** Click to send request, Run (play icon).
- Response Status:** 200 - OK, 34 ms
- Response Body:** The search results, including the total count (1), hits (1), and the single hit's details (index: movies, type: movie, id: 3, score: 1.6668314, source: { title: "To Kill a Mockingbird", director: "Robert Mulligan", year: 1962, genres: ["Crime", "Drama", "Mystery"] }).

در مثل زیر با عملگر `kill` اولویت بیشتری داده شده است.

The screenshot shows the Elasticsearch Dev Tools interface. In the top navigation bar, there are links for Console, Search Profiler, Grok Debugger, Painless Lab (BETA), History, Settings, Help, and a Dev Tools button. A search bar at the top right contains the placeholder "Search Elastic". The main area has tabs for "Console" (selected), "Search Profiler", "Grok Debugger", and "Painless Lab". A "Click to send request" button is located above the search results. The search results show a successful response (200 - OK) with a total execution time of 31 ms. The query used was a GET request to /movies/\_search with a query\_string filter for genres:Drama OR title:kill^2. The response includes a summary of shards (1 total, 1 successful, 0 skipped, 0 failed) and a list of hits, each containing an index, type, id, score, source (title: To Kill a Mockingbird, director: Robert Mulligan, year: 1962, genres: Crime, Drama, Mystery), and \_index/\_type/\_id.

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

Click to send request

```
1 {
2   "took": 3,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 5,
13      "relation": "eq"
14    },
15    "max_score": 3.0981677,
16    "hits": [
17      {
18        "_index": "movies",
19        "_type": "movie",
20        "_id": "3",
21        "_score": 3.0981677,
22        "_source": {
23          "title": "To Kill a Mockingbird",
24          "director": "Robert Mulligan",
25          "year": 1962,
26          "genres": [
27            "Crime",
28            "Drama",
29            "Mystery"
30          ]
31        }
32      }
33    ]
34  }
35}
```

در این قسمت به ساخت درخت جست و جو به صورت مرحله به مرحله می‌پردازیم.  
دستور زیر تمامی فیلم‌ها با ژانر درام را بر می‌گرداند. در این دستور از `match` استفاده شده است که برابری فیلد ژانر با مقدار درام را بررسی می‌کند.

```

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

```

```

1- [
2   "took" : 1,
3   "timed_out" : false,
4-   "_shards" : {
5     "total" : 1,
6     "successful" : 1,
7     "skipped" : 0,
8     "failed" : 0
9   },
10-   "hits" : {
11     "total" : {
12       "value" : 4,
13       "relation" : "eq"
14     },
15     "max_score" : 0.27414778,
16     "hits" : [
17       {
18         "_index" : "movies",
19         "_type" : "movie",
20         "_id" : "4",
21         "_score" : 0.27414778,
22         "_source" : {
23           "title" : "Apocalypse Now",
24           "director" : "Francis Ford Coppola",
25           "year" : 1979,
26           "genres" : [
27             "Drama",
28             "War"
29           ]
30         }
31       },
32     ]
33   }
34 }

```

در دستور زیر، فاصله بین `kill` و `bill` در نظر گرفته نشده و `operator` ای که مشخص شده است که `or` است را در نظر می‌گیرد و درواقع فیلم هایی را نشان مدهد که در `title` آن‌ها یا کلمه `kill` و یا کلمه `bill` وجود داشته باشد.

```

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

```

```

1- [
2   "took" : 1,
3   "timed_out" : false,
4-   "_shards" : {
5     "total" : 1,
6     "successful" : 1,
7     "skipped" : 0,
8     "failed" : 0
9   },
10-   "hits" : {
11     "total" : {
12       "value" : 2,
13       "relation" : "eq"
14     },
15     "max_score" : 3.3900957,
16     "hits" : [
17       {
18         "_index" : "movies",
19         "_type" : "movie",
20         "_id" : "5",
21         "_score" : 3.3900957,
22         "_source" : {
23           "title" : "Kill Bill: Vol. 1",
24           "director" : "Quentin Tarantino",
25           "year" : 2003,
26           "genres" : [
27             "Action",
28             "Crime",
29             "Thriller"
30           ]
31         }
32       }
33     ]
34   }
35 }

```

در دستور زیر با استفاده از **multi-match** همزمان جست و جو بر دو فیلد **title** و **director** انجام می‌شود که اولویت عنوان ۳ برابر اولویت کارگردان است.

```

POST /movies/_search
{
  "query": {
    "multi_match": {
      "query": "ford",
      "fields": [
        "title^3",
        "director"
      ]
    }
  }
}

```

Click to send request

200 - OK 39 ms

```

{
  "took": 3,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 5,
      "relation": "eq"
    },
    "max_score": 2.7094321,
    "hits": [
      {
        "_index": "movies",
        "_type": "movie",
        "_id": "6",
        "_score": 2.7094321,
        "_source": {
          "title": "The Assassination of Jesse James by the Coward Robert Ford",
          "director": "Andrew Dominik",
          "year": 2007,
          "genres": [
            "Biography",
            "Crime",
            "Drama"
          ]
        }
      }
    ]
  }
}

```

دستور بالا با **and** کردن دو **match** قابل انجام است که در زیر با استفاده از درخواست **Boolean** (که هم معنی **and** است) درخواست بالا پیاده‌سازی شده است. همچنین **boost** نشان دهنده همان عملگر **^** است که اولویت میدهد.

```

POST /movies/_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "genres": "drama"
          }
        },
        {
          "match": {
            "title": {
              "query": "kill",
              "boost": 3
            }
          }
        }
      ]
    }
  }
}

```

Click to send request

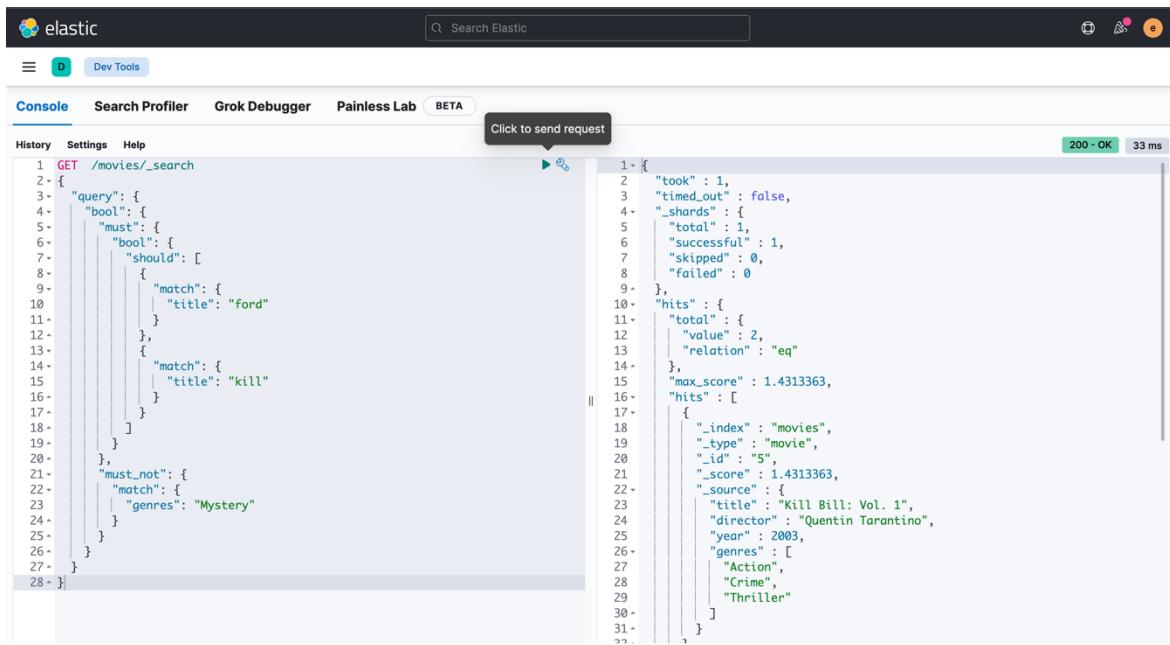
200 - OK 35 ms

```

{
  "took": 2,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 1,
      "relation": "eq"
    },
    "max_score": 4.5295043,
    "hits": [
      {
        "_index": "movies",
        "_type": "movie",
        "_id": "3",
        "_score": 4.5295043,
        "_source": {
          "title": "To Kill a Mockingbird",
          "director": "Robert Mulligan",
          "year": 1962,
          "genres": [
            "Crime",
            "Drama",
            "Mystery"
          ]
        }
      }
    ]
  }
}

```

دستور زیر فیلم هایی را نشان میدهد که در عنوان آن ها کلمه kill یا (must) به کار رفته و (should) ژانر آن ها اسرار آمیز نیست (must\_not).



The screenshot shows the Elasticsearch Dev Tools interface with the 'Console' tab selected. A search request is being typed into the text area:

```

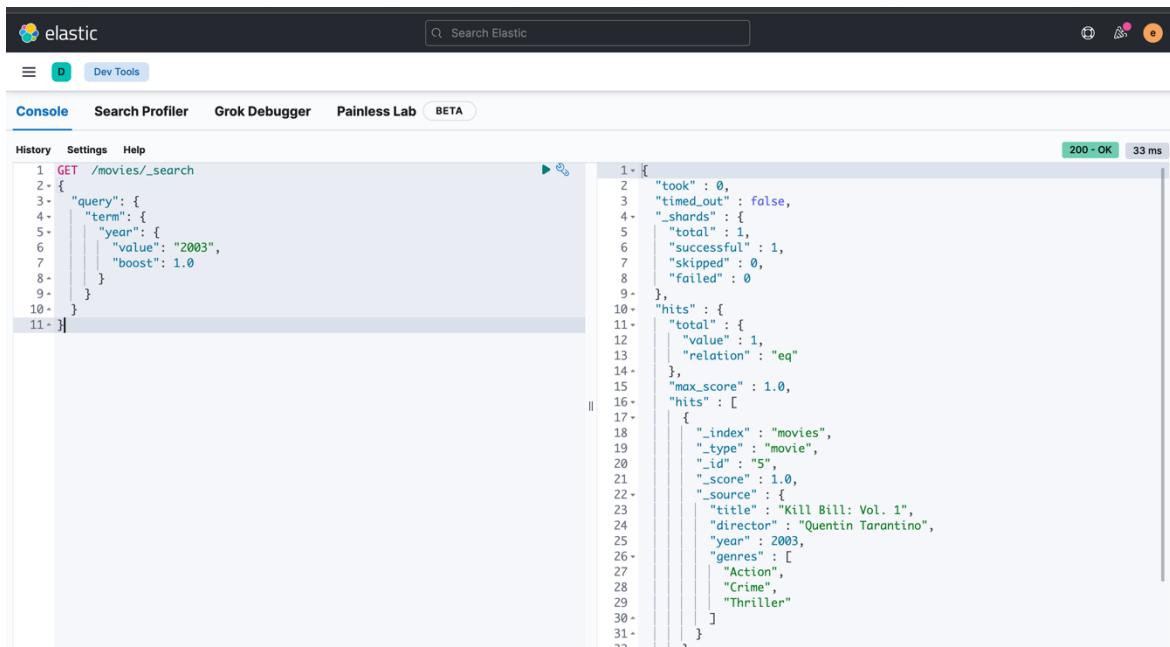
1 GET /movies/_search
2 {
3   "query": {
4     "bool": {
5       "must": [
6         "bool": {
7           "should": [
8             {
9               "match": {
10                  "title": "ford"
11                }
12              },
13              {
14                "match": {
15              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

After sending the request, the results are displayed on the right side:

```

1 {
2   "took": 1,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 2,
13      "relation": "eq"
14    },
15    "max_score": 1.4313363,
16    "hits": [
17      {
18        "_index": "movies",
19        "_type": "movie",
20        "_id": "5",
21        "_score": 1.4313363,
22        "_source": {
23          "title": "Kill Bill: Vol. 1",
24          "director": "Quentin Tarantino",
25          "year": 2003,
26          "genres": [
27            "Action",
28            "Crime",
29            "Thriller"
30          ]
31        }
32      }
33    ]
34  }
35 }
```

حال به جست و جو های غیر متنی با term میپردازیم. مثلا دستور زیر فیلم هایی را نشان میدهد که مقدار سال آن ها برابر با ۲۰۰۳ باشد.



The screenshot shows the Elasticsearch Dev Tools interface with the 'Console' tab selected. A search request is being typed into the text area:

```

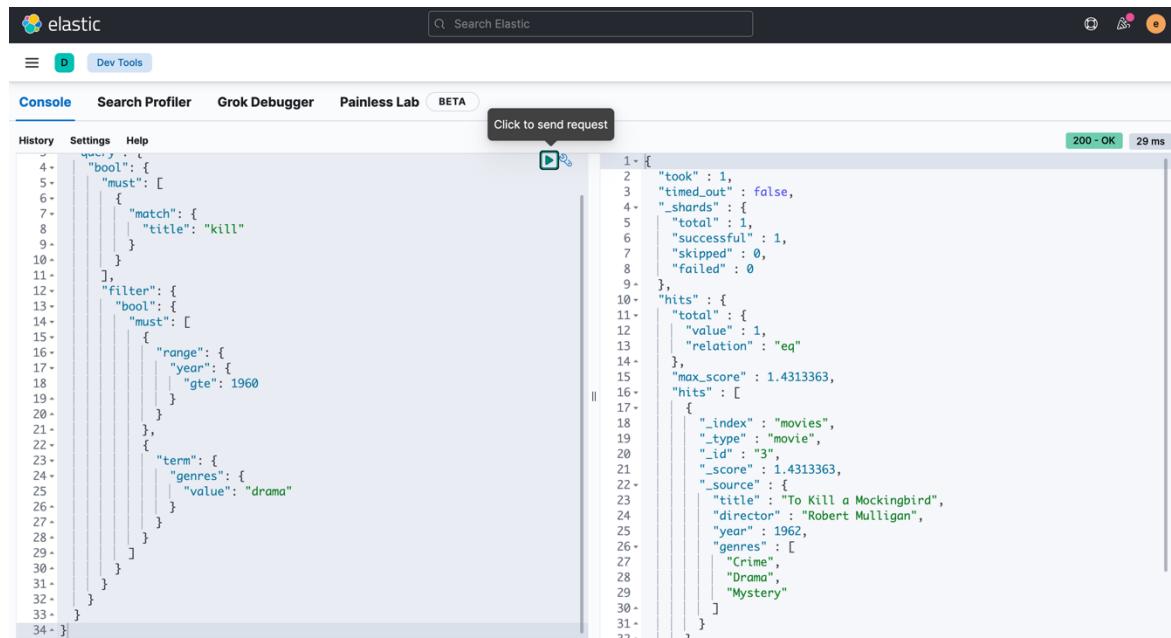
1 GET /movies/_search
2 {
3   "query": {
4     "term": {
5       "year": {
6         "value": "2003",
7         "boost": 1.0
8       }
9     }
10  }
11 }
```

After sending the request, the results are displayed on the right side:

```

1 {
2   "took": 0,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 1,
13      "relation": "eq"
14    },
15    "max_score": 1.0,
16    "hits": [
17      {
18        "_index": "movies",
19        "_type": "movie",
20        "_id": "5",
21        "_score": 1.0,
22        "_source": {
23          "title": "Kill Bill: Vol. 1",
24          "director": "Quentin Tarantino",
25          "year": 2003,
26          "genres": [
27            "Action",
28            "Crime",
29            "Thriller"
30          ]
31        }
32      }
33    ]
34  }
35 }
```

حال به عمل filtering میپردازیم. دستور زیر ابتدا تمام فیلم های تولید شده بعد از سال ۱۹۶۰ که ژانر درام دارند را فیلتر کرده و سپس به دنبال فیلم هایی که در عنوانشان کلمه kill دارند میگردد.



The screenshot shows the Elasticsearch Dev Tools interface. In the left panel, the code for a search request is displayed:

```

1 GET /movies/_search
2 {
3   "query": {
4     "bool": {
5       "must": [
6         {
7           "match": {
8             "title": "kill"
9           }
10      ],
11      "filter": {
12        "bool": {
13          "must": [
14            {
15              "range": {
16                "year": {
17                  "gte": 1960
18                }
19              },
20            {
21              "term": {
22                "genres": {
23                  "value": "drama"
24                }
25              }
26            }
27          ]
28        }
29      }
30    }
31  }
32}
33
34 }

```

In the right panel, the search results are shown:

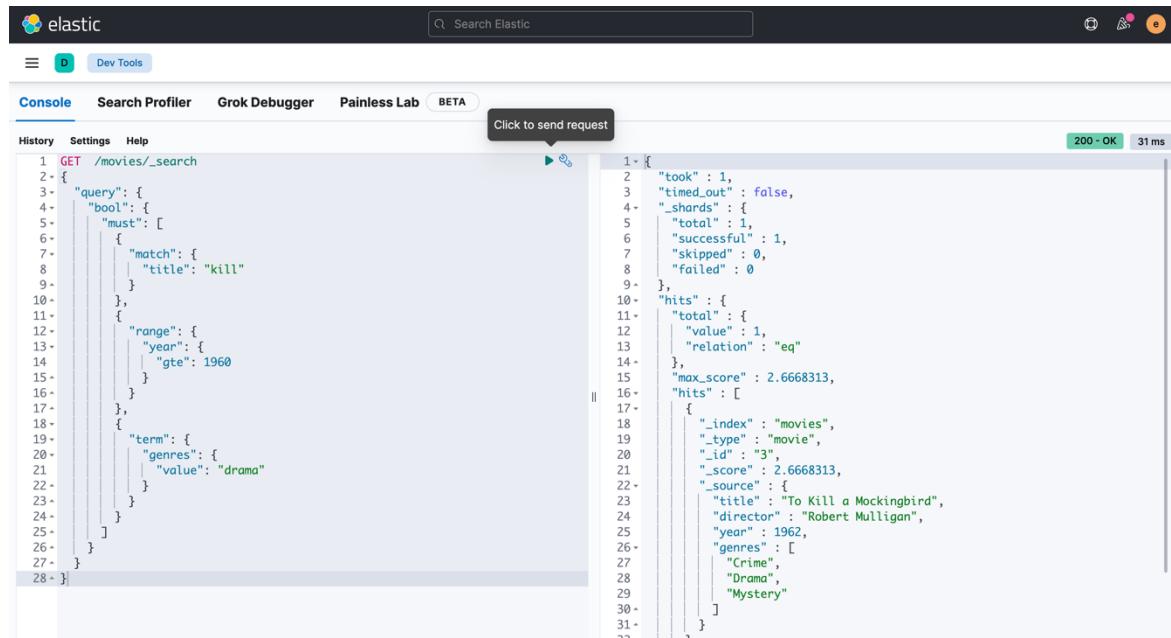
```

1 {
2   "took": 1,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 1,
13      "relation": "eq"
14    },
15    "max_score": 1.4313363,
16    "hits": [
17      {
18        "_index": "movies",
19        "_type": "movie",
20        "_id": "3",
21        "_score": 1.4313363,
22        "_source": {
23          "title": "To Kill a Mockingbird",
24          "director": "Robert Mulligan",
25          "year": 1962,
26          "genres": [
27            "Crime",
28            "Drama",
29            "Mystery"
30          ]
31        }
32      }
33    ]
34  }

```

The results show one movie document with the title "To Kill a Mockingbird", director "Robert Mulligan", year 1962, and genres Crime, Drama, and Mystery. The score is 1.4313363.

میتوان دستور بالا را مختصر تر نوشت. با توجه به آنکه در الگوی سرج درواقع داده های غیر متنی همان فیلتر های ما خواهند بود میتوان دستور بالا را با دو شرط متنی و غیر متنی به صورت زیر پیاده سازی کرد.



The screenshot shows the Elasticsearch Dev Tools interface. In the left panel, the code for a search request is displayed:

```

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          "term": {
19            "genres": {
20              "value": "drama"
21            }
22          }
23        }
24      ]
25    }
26  }
27}
28

```

In the right panel, the search results are shown:

```

1 {
2   "took": 1,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 1,
13      "relation": "eq"
14    },
15    "max_score": 2.6668313,
16    "hits": [
17      {
18        "_index": "movies",
19        "_type": "movie",
20        "_id": "3",
21        "_score": 2.6668313,
22        "_source": {
23          "title": "To Kill a Mockingbird",
24          "director": "Robert Mulligan",
25          "year": 1962,
26          "genres": [
27            "Crime",
28            "Drama",
29            "Mystery"
30          ]
31        }
32      }
33    ]
34  }

```

The results show one movie document with the title "To Kill a Mockingbird", director "Robert Mulligan", year 1962, and genres Crime, Drama, and Mystery. The score is 2.6668313.

در دستور بالا اگر بخواهیم تنها فیلم هایی که پس از سال ۱۹۶۰ ساخته شده و ژانر درام دارند را بدست بیاوریم از ۲ راه حل میتوانیم استفاده کنیم. راه حل اول در عکس اول است که بخش متغیر را برابر با {} : match\_all ناتمام داده هایی که فیلتر شده اند را برگرداند. راه حل دوم در عکس دوم است که کلا بخش پرس و جو را حذف کرده و داده ها تنها فیلتر شوند.

```

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        "term": {
17          "genres": {
18            "value": "drama"
19          }
20        }
21      }
22    ]
23  }
24 }
25 }
```

Click to send request

```

1 {
2   "took": 2,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 4,
13      "relation": "eq"
14    },
15    "max_score": 2.2741477,
16    "hits": [
17      {
18        "_index": "movies",
19        "_type": "movie",
20        "_id": "4",
21        "_score": 2.2741477,
22        "_source": {
23          "title": "Apocalypse Now",
24          "director": "Francis Ford Coppola",
25          "year": 1979,
26          "genres": [
27            "Drama",
28            "War"
29          ]
30        }
31      },
32      ...
33    ]
34  }
35 }
```

200 - OK 36 ms

```

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 }
```

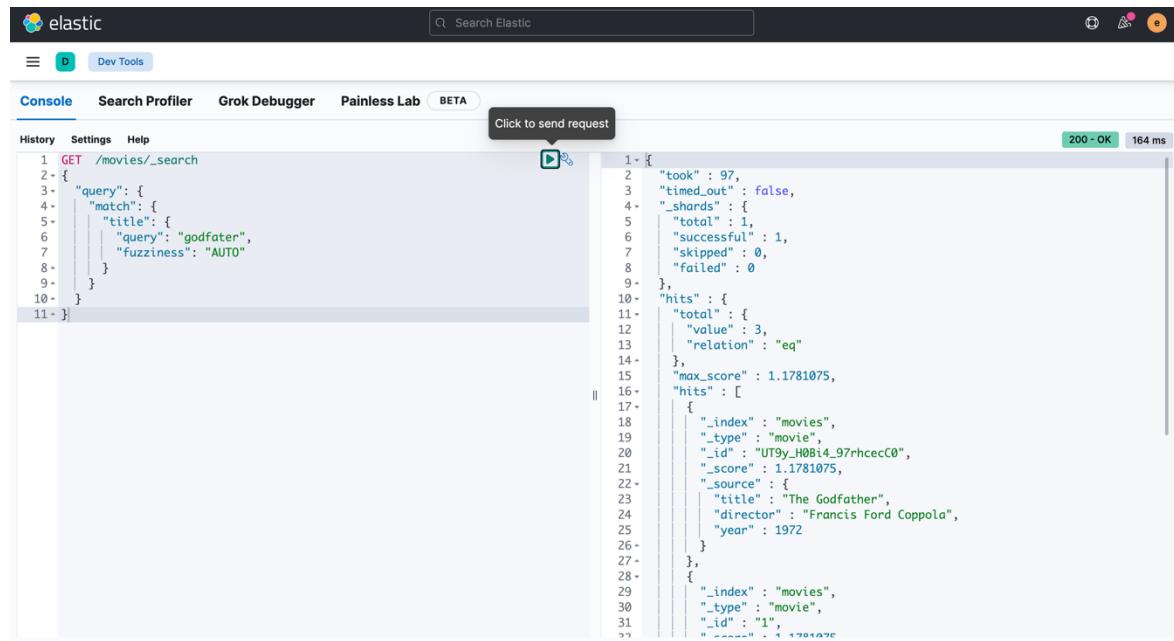
Click to send request

```

1 {
2   "took": 1,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 1,
13      "relation": "eq"
14    },
15    "max_score": 1.2354951,
16    "hits": [
17      {
18        "_index": "movies",
19        "_type": "movie",
20        "_id": "6",
21        "_score": 1.2354951,
22        "_source": {
23          "title": "The Assassination of Jesse James by the Coward Robert Ford",
24          "director": "Andrew Dominik",
25          "year": 2007,
26          "genres": [
27            "Biography",
28            "Crime",
29            "Drama"
30          ]
31        }
32      }
33    ]
34  }
35 }
```

200 - OK 29 ms

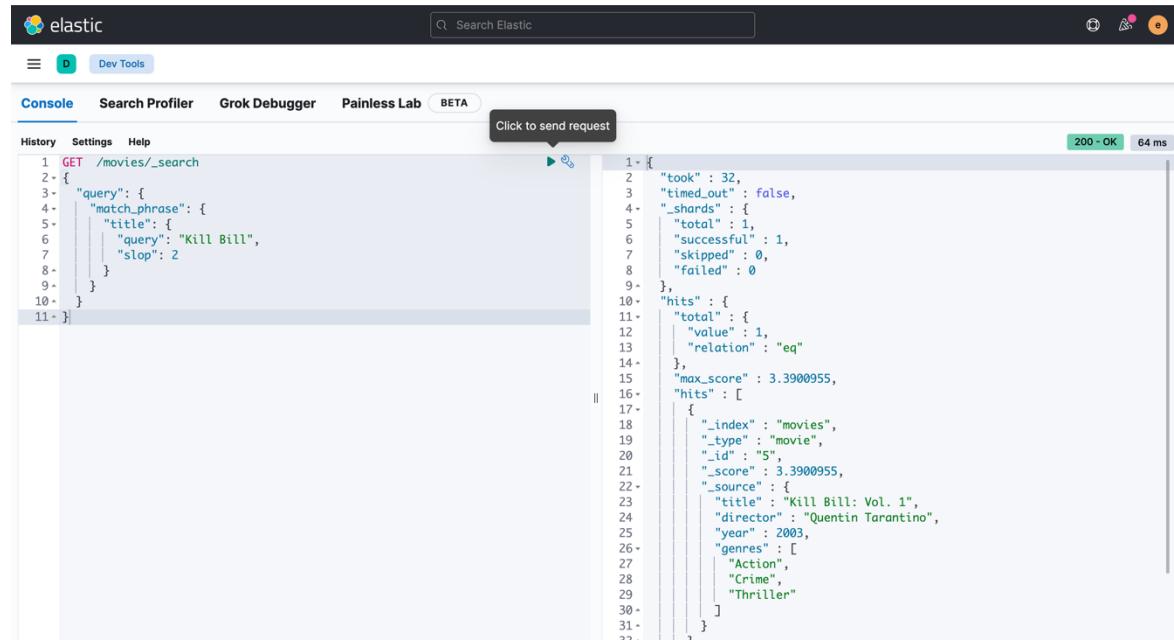
در دستور زیر با استفاده از **fuzziness** میتوانیم بخواهیم تا کلماتی که احتمالاً غلط املایی دارند را هم در نظر بگیرد. که در اینجا فیلم پدرخوانده که به اشتباه **godfater** نوشته شده است را نشان میدهد.



```
elastic Dev Tools
Console Search Profiler Grok Debugger Painless Lab BETA
History Settings Help
Click to send request
200 - OK 164 ms
1 GET /movies/_search
2 {
3   "query": {
4     "match": {
5       "title": {
6         "query": "godfater",
7         "fuzziness": "AUTO"
8       }
9     }
10   }
11 }
```

```
1 {
2   "took": 97,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10   "hits": {
11     "total": {
12       "value": 3,
13       "relation": "eq"
14     },
15     "max_score": 1.1781075,
16     "hits": [
17       {
18         "_index": "movies",
19         "_type": "movie",
20         "_id": "UT9y_H0Bt4_97rhcec0",
21         "_score": 1.1781075,
22         "_source": {
23           "title": "The Godfather",
24           "director": "Francis Ford Coppola",
25           "year": 1972
26         }
27       },
28       {
29         "_index": "movies",
30         "_type": "movie",
31         "_id": "1",
32         "_score": 1.1781075
33       }
34     ]
35   }
36 }
```

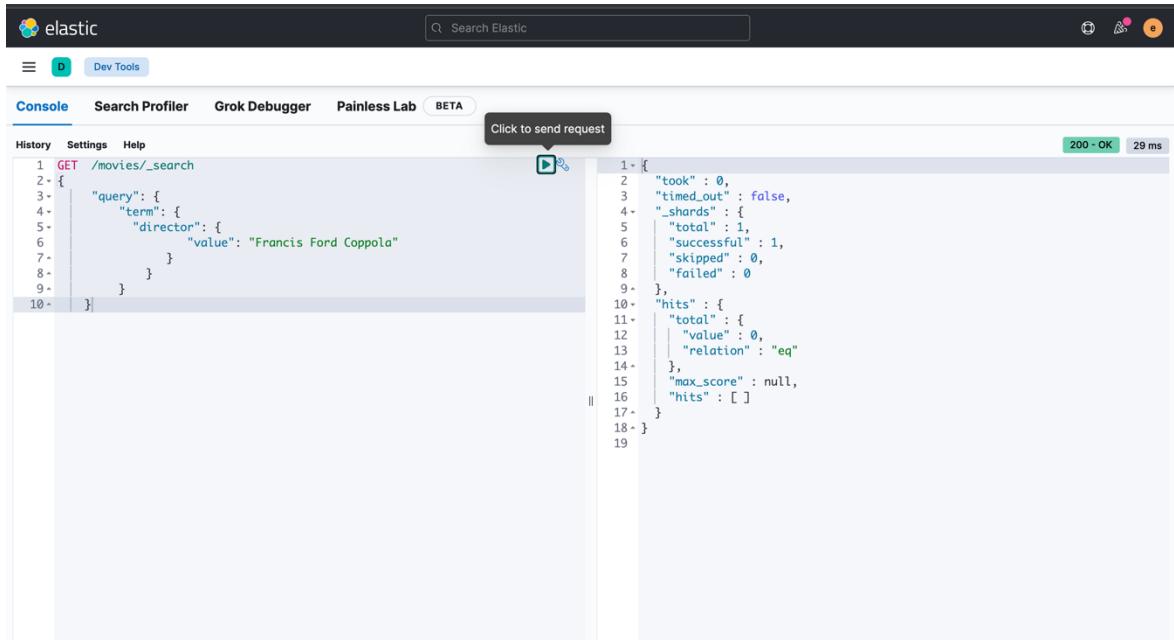
دستور زیر در واقع بیان می‌کند که به دنبال عبارت **kill bill** هستیم و باید دقیقاً کلمات ذکر شده پشت سر هم ظاهر شده باشند و نهایتاً ۳ فضای خالی بین هر کلمه تا کلمه بعدی را قبول خواهیم کرد.



```
elastic Dev Tools
Console Search Profiler Grok Debugger Painless Lab BETA
History Settings Help
Click to send request
200 - OK 64 ms
1 GET /movies/_search
2 {
3   "query": {
4     "match_phrase": {
5       "title": {
6         "query": "Kill Bill",
7         "slop": 2
8       }
9     }
10   }
11 }
```

```
1 {
2   "took": 32,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10   "hits": {
11     "total": {
12       "value": 1,
13       "relation": "eq"
14     },
15     "max_score": 3.3900955,
16     "hits": [
17       {
18         "_index": "movies",
19         "_type": "movie",
20         "_id": "5",
21         "_score": 3.3900955,
22         "_source": {
23           "title": "Kill Bill: Vol. 1",
24           "director": "Quentin Tarantino",
25           "year": 2003,
26           "genres": [
27             "Action",
28             "Crime",
29             "Thriller"
30           ]
31         }
32       }
33     ]
34   }
35 }
```

اگر به دنبال جستجوی کارگردانی با نامی مانند francis ford coppola باشیم و تنها با term و value مقدار کارگردان را برایر با این نام بگذاریم به نتیجه‌های نخواهیم رسید زیرا السنتیک، متن ها را آنالیز و پردازش میکند و آن ها به کلمات تشکیل دهنده‌شان تبدیل میکند. برای حل این موضوع میتوان از keyword استفاده کرد.



```

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

```

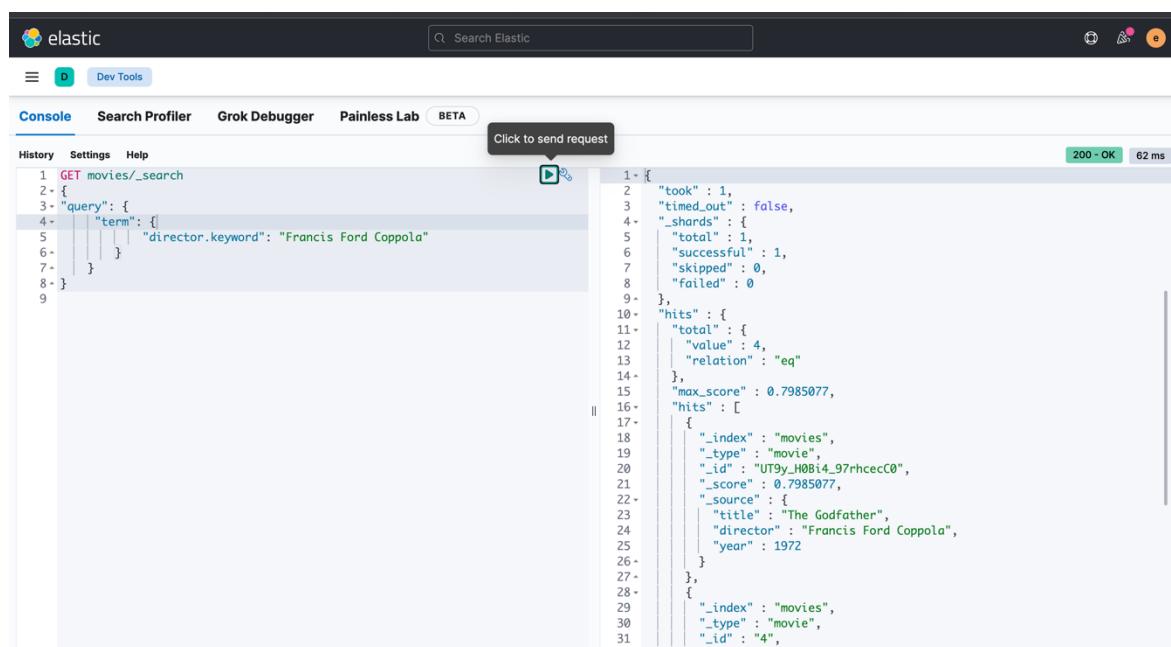
Click to send request

```

1+ {
2   "took" : 0,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 1,
6     "successful" : 1,
7     "skipped" : 0,
8     "failed" : 0
9   },
10+ "hits" : {
11   "total" : {
12     "value" : 0,
13     "relation" : "eq"
14   },
15   "max_score" : null,
16   "hits" : []
17 }
18+
19

```

200 OK 29 ms



```

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

```

Click to send request

```

1+ {
2   "took" : 1,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 1,
6     "successful" : 1,
7     "skipped" : 0,
8     "failed" : 0
9   },
10+ "hits" : {
11   "total" : {
12     "value" : 4,
13     "relation" : "eq"
14   },
15   "max_score" : 0.7985077,
16   "hits" : [
17     {
18       "_index" : "movies",
19       "_type" : "movie",
20       "_id" : "UT9y_HOB14_97rhcecC0",
21       "_score" : 0.7985077,
22       "_source" : {
23         "title" : "The Godfather",
24         "director" : "Francis Ford Coppola",
25         "year" : 1972
26       }
27     },
28     {
29       "_index" : "movies",
30       "_type" : "movie",
31       "_id" : "4",
32       "_score" : 0.7985077
33     }
34   ]
35 }
36

```

200 OK 62 ms

**بخش دوم)**

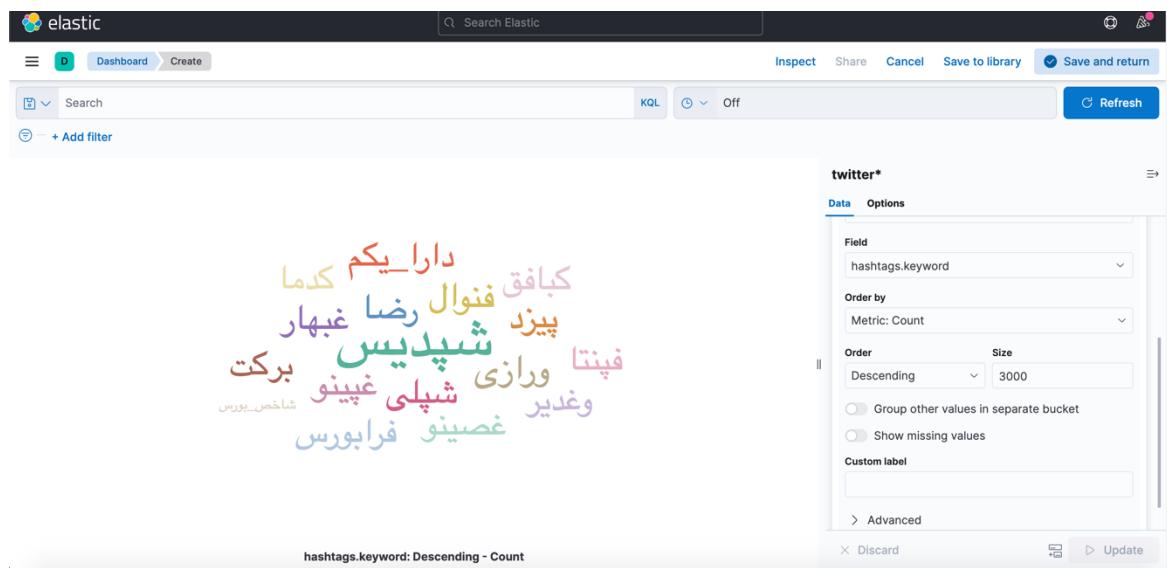
در این بخش ابتدا کد پایتون داده شده را اجرا میکنیم تا حداقل ۵۰۰ tweet در الستیک سرج ذخیره شود.

```
Fetched and inserted 452 tweets so far
Waiting for 60 seconds...
Fetched and inserted 452 tweets so far
Waiting for 60 seconds...
Fetched and inserted 462 tweets so far
Waiting for 60 seconds...
Fetched and inserted 482 tweets so far
Waiting for 60 seconds...
Fetched and inserted 482 tweets so far
Waiting for 60 seconds...
Fetched and inserted 492 tweets so far
Waiting for 60 seconds...
Fetched and inserted 502 tweets so far
Waiting for 60 seconds...
Fetched and inserted 522 tweets so far
Waiting for 60 seconds...
Fetched and inserted 522 tweets so far
Waiting for 60 seconds...
^CTraceback (most recent call last):
  File "untitled.py", line 45, in <module>
    time.sleep(api_sleep)
KeyboardInterrupt
```

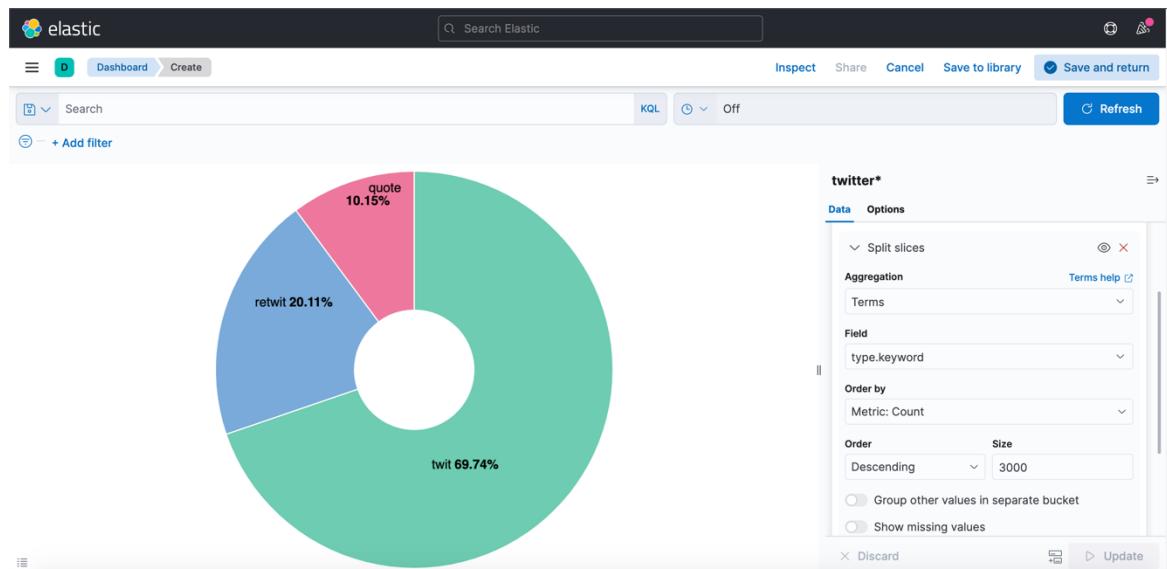
The screenshot shows the Elasticsearch interface with a search bar containing the term 'all'. The results pane on the right displays a single entry under the 'Metrics' section:

- twitter\*** (highlighted in red)
- Data Options**
- Metrics** (underlined in blue) > [Tag\\_size\\_Count](#)
- Buckets** (with a '+ Add' button)

Visualization اول به صورت cloud tag است که پس از آن که آن را برای hashtag ها انتخاب می‌کنیم، سایز هر کلمه بسته به تعداد تکرار آن خواهد بود.



Visualization دوم به صورت pie است که پس از آن که آن را برای type ها انتخاب می‌کنیم، سایز هر slice که مربوط به یک نوع است، تعداد تکرار آن نوع توبیت را نشان می‌دهد.



همین visualization را برای همان hashtag ها هم میتوان به کار برد.

