
IT-309
EXPERIMENT-4 REPORT

Indian Institute of Information Technology, Vadodara

Group-1

Contents

0.1	Elasticsearch And Kibana	2
0.2	Results	2

0.1 ELASTICSEARCH AND KIBANA

- Elasticsearch: It is open source search engine based on the Lucene library and is highly scalable.
- Elasticsearch takes in unstructured data from different locations, stores and indexes it according to user-specified mapping and makes it searchable.
- Kibana: It is an open source data visualization web interface for Elasticsearch
- The data used here is demo data of bank accounts in JSON available at elastic.io

0.2 RESULTS

Results for some of the queries are shown below:

- PUT REQUEST

```

1 PUT /my_playlist/song/6
2 {
3   "title": "1000 years",
4   "artist": "Christina Perri",
5   "album": "Breaking Dawn",
6   "year": 2011
7 }

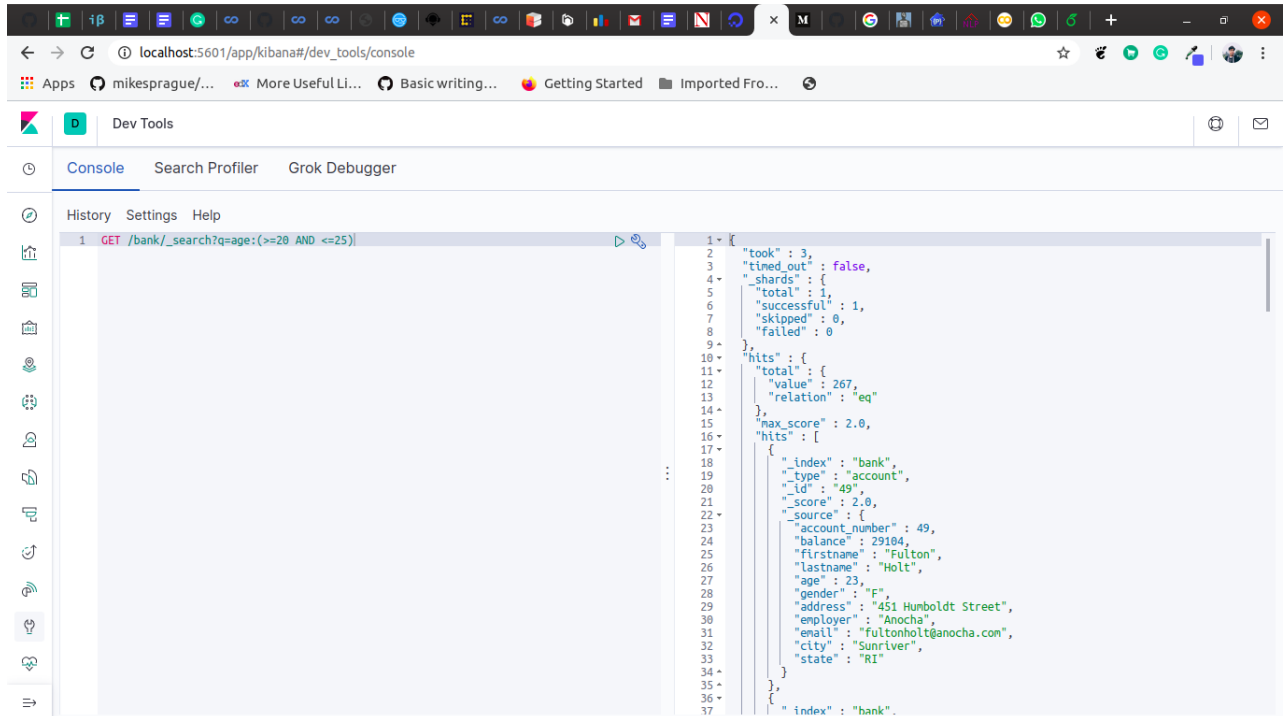
```

```

1 #! Deprecation: [types removal] Specifying types in document index requests is
2 deprecated, use the typeless endpoints instead (/index/_doc/{id}, /index/_doc,
3 or /index/_create/{id}).
4 {
5   "_index": "my_playlist",
6   "_type": "song",
7   "_id": "6",
8   "_version": 8,
9   "result": "updated",
10  "shards": {
11    "total": 2,
12    "successful": 1,
13    "failed": 0
14  },
15  "_seq_no": 7,
16  "_primary_term": 3
17 }

```

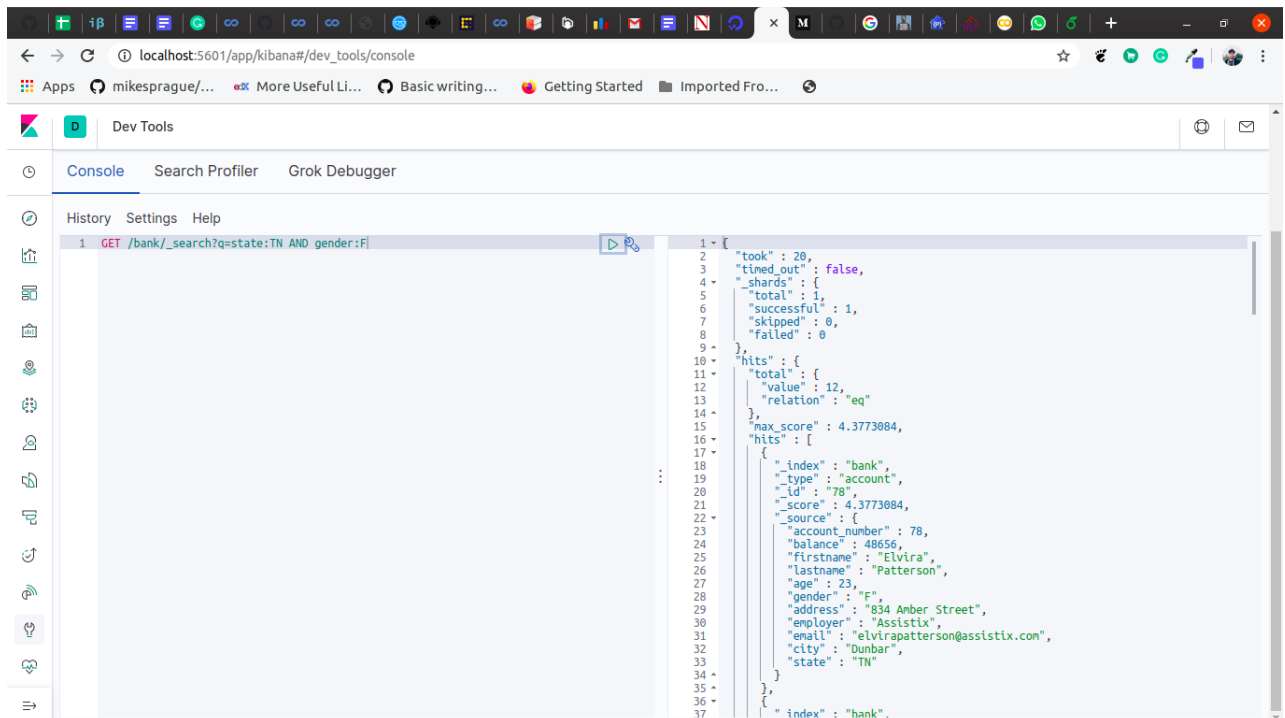
- Query1: "Age>=20 AND <=25"



The screenshot shows the Kibana Dev Tools console with the following details:

- URL:** `localhost:5601/app/kibana#/dev_tools/console`
- Console Tab:** Active, showing the query `GET /bank/_search?q=age:(>=20 AND <=25)`.
- Response:** A JSON object indicating a successful search with 1 hit. The hit details are:
 - `"took": 3`
 - `"timed_out": false`
 - `"_shards": { "total": 1, "successful": 1, "skipped": 0, "failed": 0 }`
 - `"hits": { "total": { "value": 267, "relation": "eq" }, "max_score": 2.0, "hits": [{ "_index": "bank", "_type": "account", "_id": "49", "_score": 2.0, "_source": { "account_number": 49, "balance": 29184, "firstname": "Fulton", "lastname": "Holt", "age": 23, "gender": "F", "address": "451 Humboldt Street", "employer": "Anocha", "email": "fultonholt@anocha.com", "city": "Sunriver", "state": "RI" } }] }`

- Query2: "state: TN AND gender:F"



The screenshot shows the Kibana Dev Tools console with the following details:

- URL:** `localhost:5601/app/kibana#/dev_tools/console`
- Console Tab:** Active, showing the query `GET /bank/_search?q=state:TN AND gender:F`.
- Response:** A JSON object indicating a successful search with 1 hit. The hit details are:
 - `"took": 20`
 - `"timed_out": false`
 - `"_shards": { "total": 1, "successful": 1, "skipped": 0, "failed": 0 }`
 - `"hits": { "total": { "value": 12, "relation": "eq" }, "max_score": 4.3773084, "hits": [{ "_index": "bank", "_type": "account", "_id": "78", "_score": 4.3773084, "_source": { "account_number": 78, "balance": 48656, "firstname": "Elvira", "lastname": "Patterson", "age": 23, "gender": "F", "address": "834 Amber Street", "employer": "Assistix", "email": "elvirapatterson@assistix.com", "city": "Dunbar", "state": "TN" } }] }`

- Query3: "state: UT OR CA"

The screenshot shows the Kibana Dev Tools Console with a GET request to `/bank/_search?q=state:UT OR CA`. The response is a JSON object representing a search result.

```
1 GET /bank/_search?q=state:UT OR CA
2 {
3   "took": 15,
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": 37,
14      "relation": "eq"
15    },
16    "max_score": 4.046554,
17    "hits": [
18      {
19        "_index": "bank",
20        "_type": "account",
21        "_id": "68",
22        "_score": 4.046554,
23        "_source": {
24          "account_number": 68,
25          "balance": 44214,
26          "firstname": "Hall",
27          "lastname": "Key",
28          "age": 25,
29          "gender": "F",
30          "address": "927 Bay Parkway",
31          "employer": "Eventex",
32          "email": "hallkey@eventex.com",
33          "city": "Shawmut",
34          "state": "CA"
35        }
36      },
37      {
38        "_index": "bank".
```

- Example of QueryDSL

The screenshot shows the Kibana Dev Tools Console with a GET request to `/_search` using a complex QueryDSL query. The response is a JSON object representing a search result.

```
1 GET /_search
2 {
3   "query": {
4     "bool": {
5       "must": [
6         { "match": { "address": "Street" } }
7       ],
8       "filter": [
9         { "term": { "gender": "F" } },
10        { "range": { "age": { "gte": 25 } } }
11      ]
12    }
13  }
14 }
15 {
16   "took": 29,
17   "timed_out": false,
18   "_shards": {
19     "total": 6,
20     "successful": 6,
21     "skipped": 0,
22     "failed": 0
23   },
24   "hits": {
25     "total": {
26       "value": 154,
27       "relation": "eq"
28     },
29     "max_score": 0.95395315,
30     "hits": [
31       {
32         "_index": "bank",
33         "_type": "account",
34         "_id": "13",
35         "_score": 0.95395315,
36         "_source": {
37           "account_number": 13,
38           "balance": 32838,
39           "firstname": "Nanette",
40           "lastname": "Bates",
41           "age": 28,
42           "gender": "F",
43           "address": "789 Madison Street",
44           "employer": "Quility",
45           "email": "nanettebates@quility.com",
46           "city": "Nogal",
47           "state": "VA"
48         }
49       },
50       {
51         "_index": "bank".
```