- How Elasticsearch & Kibana work
- Installation & setup
- Index creation, data import, queries

Elasticsearch + Kibana: Full Documentation

1. Introduction to Elasticsearch and Kibana

🔹 What is Elasticsearch?

Elasticsearch is a search and analytics engine that allows you to store, search, and analyze large volumes of data efficiently. It is a NoSQL database that stores data in JSON format and uses REST APIs for querying.

🔹 How does Elasticsearch work?

1. Data is stored in indexes (like tables in SQL databases).
2. Each document is a JSON object (similar to rows in SQL).
3. Queries are executed via REST API (using curl or Kibana Dev Tools).

🔹 What is Kibana?

Kibana is a visual interface for working with Elasticsearch.
With Kibana, you can:

- View and explore data (Discover)
- Build visualizations (Visualize)
- Perform complex queries (Dev Tools)

🔹 2. Installing Elasticsearch and Kibana

2.1 Installing Elasticsearch

Download and extract Elasticsearch:
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.17.7-linux-x86_64.tar.gz
tar -xzf elasticsearch-7.17.7-linux-x86_64.tar.gz
Start Elasticsearch:
cd elasticsearch-7.17.7
./bin/elasticsearch
Verify if Elasticsearch is running:
curl -XGET "http://localhost:9200/?pretty"

2.2 Installing Kibana

Download and extract Kibana:
wget https://artifacts.elastic.co/downloads/kibana/kibana-7.17.7-linux-x86_64.tar.gztar -xzf kibana-7.17.7-linux-x86_64.tar.gz

Start Kibana:
cd kibana-7.17.7-linux-x86_64
./bin/kibana
Open Kibana in a browser:
http://localhost:5601

🔹 3. Working with Elasticsearch: CRUD Operations

3.1 Create an index (movies2)

Create a new index:

curl -XPUT "http://localhost:9200/movies2" -H "Content-Type: application/json" --data-binary @mapping.json

Check if the index is created:

curl -XGET "http://localhost:9200/_cat/indices?v"

Check its structure (mapping):

curl -XGET "http://localhost:9200/movies2/_mapping?pretty"

✅ 3.2 Adding and Deleting Documents

📌 Add a document to movies2:

```
curl -XPOST "http://localhost:9200/movies2/_doc/1" -H "Content-Type: application/json" -d'
{
 "title": "Inception",
 "year": 2010,
 "genres": ["Sci-Fi", "Action"],
 "rating": 8.8
}'
```

Retrieve a document by ID:
curl -XGET "http://localhost:9200/movies2/_doc/1?pretty"
Delete a document:
curl -XDELETE "http://localhost:9200/movies2/_doc/1"

3.3 Deleting and Recreating an Index

Delete the movies2 index (if it exists):

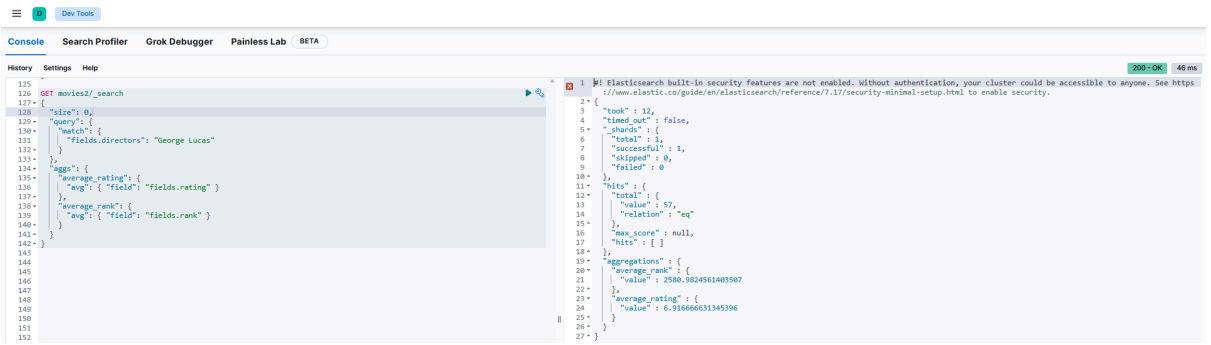
curl -XDELETE "http://localhost:9200/movies2"

Recreate with correct mapping.json: curl -XPUT "http://localhost:9200/movies2" -H "Content-Type: application/json" --data-binary @mapping.json

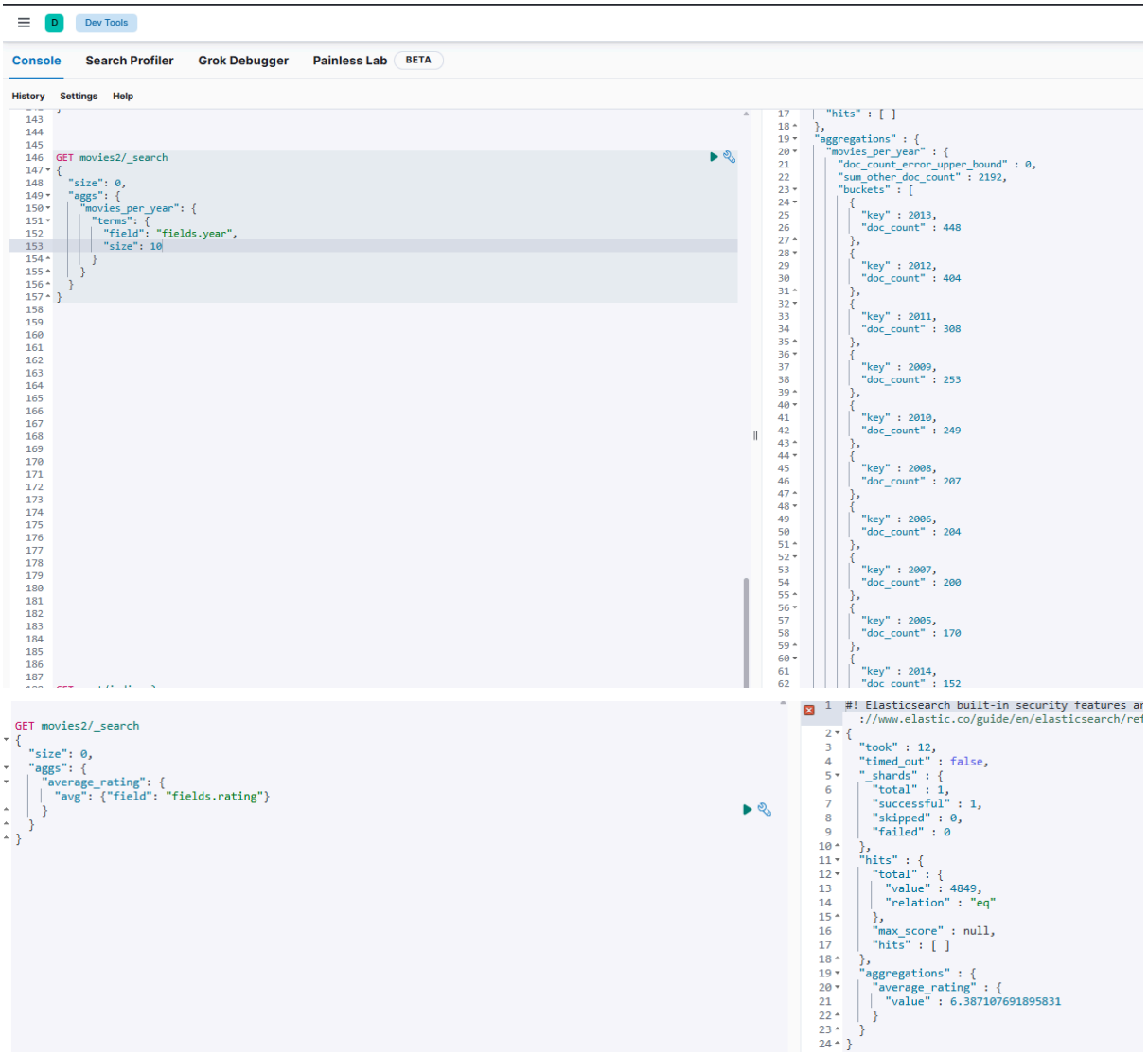
There are some GET functions that I executed in DevTools, but I also used the Elasticsearch library, and I completed everything in all the TPs

```
GET movies2/_search
{
  "size": 0,
  "aggs": {
    "average_rating": {
      "avg": {"field": "fields.rating"}
    }
  }
}
```

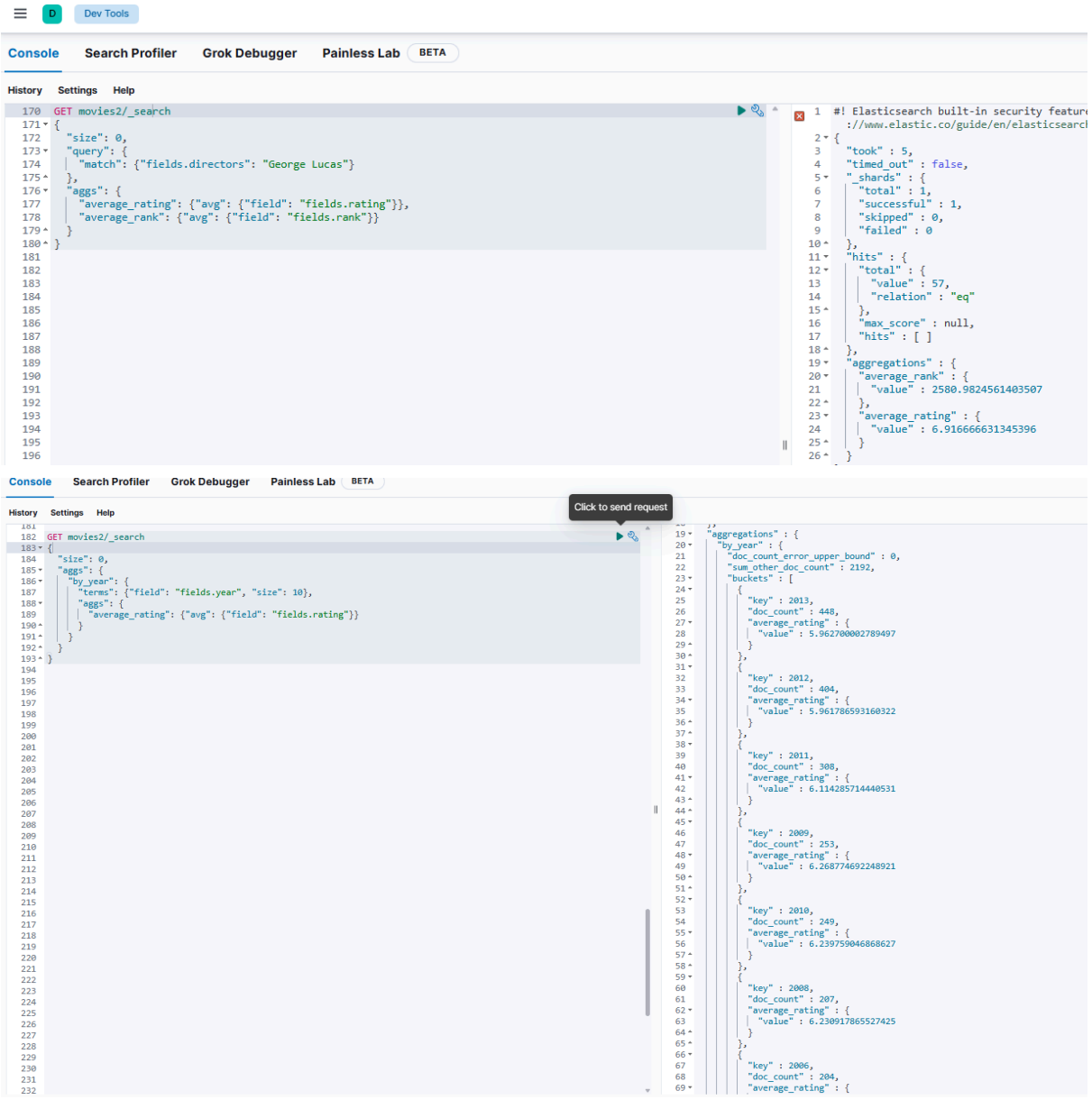
```
{
  "took" : 6,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 4849,
      "relation" : "eq"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "average_rating" : {
      "value" : 6.387107691895831
    }
  }
}
```




```
#! 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/7.17/security-minimal-setup.html to enable security.
health status index                           uuid                   pri rep docs.count docs.deleted store.size pri.store.size
green  open   .geoip_databases                 pKVX873ySCuql4ne6yi-Uw   1   0         39            0     36.7mb         36.7mb
yellow open   movies                          90G7Ga48RGuIQYoJO_wokw   1   1       9550            0      9.1mb          9.1mb
green  open   .apm-custom-link                QeXE1XqCRNuw6F3he1C7tw   1   0          0            0       226b           226b
green  open   .kibana_task_manager_7.17.7_001 6kh9Kw52TOG8rbkzCvs86w   1   0         17           61    349.2kb        349.2kb
green  open   .apm-agent-configuration        lourqh-hTS2Djpfs4QcOJA   1   0          0            0       226b           226b
green  open   .async-search                   cVLENfi8T1Gi9XIRel8fWg   1   0          0            0       3.5kb          3.5kb
green  open   .kibana_7.17.7_001              dr42Nm3NQjmETGIw5f_MoQ   1   0        447            2       4.9mb          4.9mb
yellow open   movies2                         RwTmuo73TWuqeAu9sMDNSw   1   1       4849            0       3.4mb          3.4mb
green  open   .tasks                          RR_rjgAlTTWtryMCT_5xlQ   1   0         17            1      71.2kb         71.2kb
```


Dev Tools
Console
Search Profiler
Grok Debugger
Painless Lab
BETA
History
Settings
Help
GET movies2/_search
{
"size": 0,
"aggs": {
"movies_per_year": {
"terms": {
"field": "fields.year",
"size": 10
}
}
}
}
"hits" : [ ]
},
"aggregations" : {
"movies_per_year" : {
"doc_count_error_upper_bound" : 0,
"sum_other_doc_count" : 2192,
"buckets" : [
{
"key" : 2013,
"doc_count" : 448
},
{
"key" : 2012,
"doc_count" : 404
},
{
"key" : 2011,
"doc_count" : 308
},
{
"key" : 2009,
"doc_count" : 253
},
{
"key" : 2010,
"doc_count" : 249
},
{
"key" : 2008,
"doc_count" : 207
},
{
"key" : 2006,
"doc_count" : 204
},
{
"key" : 2007,
"doc_count" : 200
},
{
"key" : 2005,
"doc_count" : 170
},
{
"key" : 2014,
"doc_count" : 152

GET movies2/_search
{
"size": 0,
"aggs": {
"average_rating": {
"avg": {"field": "fields.rating"}
}
}
}
#! Elasticsearch built-in security features ar
://www.elastic.co/guide/en/elasticsearch/re
{
"took" : 12,
"timed_out" : false,
"_shards" : {
"total" : 1,
"successful" : 1,
"skipped" : 0,
"failed" : 0
},
"hits" : {
"total" : {
"value" : 4849,
"relation" : "eq"
},
"max_score" : null,
"hits" : [ ]
},
"aggregations" : {
"average_rating" : {
"value" : 6.387107691895831
}
}
}


Dev Tools
Console
Search Profiler
Grok Debugger
Painless Lab
BETA
History
Settings
Help
GET movies2/_search
{
"size": 0,
"query": {
"match": {"fields.directors": "George Lucas"}
},
"aggs": {
"average_rating": {"avg": {"field": "fields.rating"}},
"average_rank": {"avg": {"field": "fields.rank"}}
}
}
#! Elasticsearch built-in security feature
://www.elastic.co/guide/en/elasticsearch
{
"took" : 5,
"timed_out" : false,
"_shards" : {
"total" : 1,
"successful" : 1,
"skipped" : 0,
"failed" : 0
},
"hits" : {
"total" : {
"value" : 57,
"relation" : "eq"
},
"max_score" : null,
"hits" : [ ]
},
"aggregations" : {
"average_rank" : {
"value" : 2580.9824561403507
},
"average_rating" : {
"value" : 6.916666631345396
}
}

Console
Search Profiler
Grok Debugger
Painless Lab
BETA
History
Settings
Help
Click to send request
GET movies2/_search
{
"size": 0,
"aggs": {
"by_year": {
"terms": {"field": "fields.year", "size": 10},
"aggs": {
"average_rating": {"avg": {"field": "fields.rating"}}
}
}
}
}
"aggregations" : {
"by_year" : {
"doc_count_error_upper_bound" : 0,
"sum_other_doc_count" : 2192,
"buckets" : [
{
"key" : 2013,
"doc_count" : 448,
"average_rating" : {
"value" : 5.962700002789497
}
},
{
"key" : 2012,
"doc_count" : 404,
"average_rating" : {
"value" : 5.961786593160322
}
},
{
"key" : 2011,
"doc_count" : 308,
"average_rating" : {
"value" : 6.114285714440531
}
},
{
"key" : 2009,
"doc_count" : 253,
"average_rating" : {
"value" : 6.268774692248921
}
},
{
"key" : 2010,
"doc_count" : 249,
"average_rating" : {
"value" : 6.239759046868627
}
},
{
"key" : 2008,
"doc_count" : 207,
"average_rating" : {
"value" : 6.230917865527425
}
},
{
"key" : 2006,
"doc_count" : 204,
"average_rating" : {