🔎 Elasticsearch 8.x Cheatsheet 🔍

All the API endpoints and pro-tips you always forgot about in one place!
Built by developers for developers. Hosted on GitHub, contributions welcome.

elasticsearch

Links Queries Indexes Debug Cluster & Plugins Elasticsearch 8.X ▾

Links

First thing, forget about your `curl` calls and **install Kibana please!**

- Download Elasticsearch, official download page;
- Elasticsearch Reference, official documentation;
- Docker repository, quick start any Elastic software;
- Dev.to Elasticsearch, great source of content about Elastic;
- Official forum and StackOverflow for support;

Queries

There are two syntaxes for the basic queries: a simple one on the left, where you can't use any option, and an extended one on the right. Most of the beginner headache with the DSL come from this:

```
GET _search
{
  "query": {
    "match": {
      "FIELD": "TEXT"
    }
  }
}
```

TO

```
GET _search
{
  "query": {
    "match": {
      "FIELD": {
        "query": "TEXT",
        "OPTION": "VALUE"
      }
    }
  }
}
```

Full search example with aggregation, highlight, filter...

```
GET /_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "title": "smith"
          }
        }
      ],
      "must_not": [
        {
          "match_phrase": {
            "title": "granny smith"
          }
        }
      ],
      "filter": [
        {
          "exists": {
            "field": "title"
          }
        }
      ]
    }
  },
  "aggs": {
    "my_agg": {
      "terms": {
        "field": "user",
        "size": 10
      }
    }
  },
  "highlight": {
    "pre_tags": [
      "<em>"
    ],
    "post_tags": [
      "</em>"
    ],
    "fields": {
      "body": {
        "number_of_fragments": 1,
        "fragment_size": 20
      },
      "title": {}
    }
  },
  "size": 20,
  "from": 100,
  "_source": [
    "title",
    "id"
  ],
  "sort": [
    {
      "_id": {
        "order": "desc"
      }
    }
  ]
}
```

Control total hit count

Accept true, false or a fixed number, default to 10000.

```
GET /_search
{
  "track_total_hits": true,
  "query": {}
}
```

Common queries

```
"multi_match": {
  "query": "Elastic",
  "fields": ["user.*", "title^3"],
  "type": "best_fields"
}
```

```
"bool": {
  "must": [],
  "must_not": [],
  "filter": [],
  "should": [],
  "minimum_should_match" : 1
}
```

```
"range": {
  "age": {
    "gte": 10,
    "lte": 20,
    "boost": 2
  }
}
```

QueryString syntax

Search in the default `_all` field:

```
GET /_search?q=pony
```

Complex search with operator and exact phrase search with boost:

```
GET /_search?q=title:(joli OR code) AND author:"Damien Alexandre"^2
```

Search with wildcard and special queries:

```
GET /_search?q=_exists_:title OR title:singl? noneOrAnyChar*cter
```

Search with fuzzyness and range:

```
GET /_search?q=title:elastichurch~3 AND date:[2016-01-01 TO 2018-12-31]
```

Use in Query DSL (not recommended for user search):

```
GET /_search
{
  "query": {
    "query_string": {
      "default_field": "content",
      "query": "elastic AND (title:lucene OR title:solr)"
    }
  }
}
```

Search After - Pagination cursor

Search with a custom sort:

```
GET products/_search
{
    "size": 10,
    "sort": [
        {"date": "asc"},
        {"_id": "desc"}
    ]
}
```

On the next "page", pass the sort values from the last result:

```
GET product/_search
{
    "size": 10,
    "search_after": [1463538857, "654323"],
    "sort": [
        {"date": "asc"},
        {"_id": "desc"}
    ]
}
```

Indexes and mapping

Create an index with settings and mapping

```
PUT /my_index_name
{
  "settings": {
    "number_of_replicas": 1,
    "number_of_shards": 3,
    "analysis": {},
    "refresh_interval": "1s"
  },
  "mappings": {
    "dynamic": false,
    "properties": {
      "title": {
        "type": "text",
        "analyzer": "english"
      }
    }
  }
}
```

Update index settings dynamically

```
PUT /my_index_name/_settings
{
  "index": {
    "refresh_interval": "-1",
    "number_of_replicas": 0
  }
}
```

Update an index by adding a field to a type

```
PUT /my_index_name/_mapping
{
  "properties": {
    "tag": {
      "type": "keyword"
    }
  }
}
```

Get the mapping and the settings

```
GET /my_index_name
```

```
GET /my_index_name/_mapping
```

```
GET /my_index_name/_settings
```

Create a document (auto-generated ID)

```
POST /my_index_name/_doc
{
  "title": "Elastic is funny",
  "tag": [
    "lucene"
  ]
}
```

Create or update a document

```
PUT /my_index_name/_doc/12abc
{
  "title": "Elastic is funny",
  "tag": [
    "lucene"
  ]
}
```

Delete a document

```
DELETE /my_index_name/_doc/12abc
```

Open and close indexes to save memory and CPU

```
POST /my_index_name/_close
```

```
POST /my_index_name/_open
```

Remove and create aliases

```
POST /_aliases
{
  "actions": [
    {
      "remove": {
        "index": "my_index_name",
        "alias": "foo"
      }
    },
    {
      "add": {
        "index": "my_index_name",
        "alias": "bar",
        "filter" : { "term" : { "user" : "damien" } }
      }
    }
  ]
}
```

List aliases

```
GET /_aliases
```

```
GET /my_index_name/_alias/*
```

```
GET /*/_alias/*
```

```
GET /*/_alias/foo
```

Full custom analyzer declaration

```
PUT /english_example
{
  "settings": {
    "analysis": {
      "filter": {
        "english_stop": {
          "type":       "stop",
          "stopwords":  "_english_"
        },
        "english_stemmer": {
          "type":       "stemmer",
          "language":   "english"
        }
      },
      "analyzer": {
        "my_english": {
          "char_filter":  ["html_strip"],
          "tokenizer":  "standard",
          "filter": [
            "lowercase",
            "english_stop",
            "english_stemmer"
          ]
        }
      }
    }
  }
}
```

Indices monitoring and information

```
GET /my_index_name/_stats
```

```
GET /my_index_name/_segments
```

```
GET /my_index_name/_recovery?pretty&human
```

Indices status and management

```
POST /my_index_name/_cache/clear
```

```
POST /my_index_name/_refresh
```

```
POST /my_index_name/_flush
```

```
POST /my_index_name/_forcemerge
```

Reindex API

Simple Reindex Operation

```
POST /_reindex
{
  "source": {
    "index": "test-index"
  },
  "dest": {
    "index": "test-index-new"
  }
}
```

Selective Reindex Operation

```
POST /_reindex
{
  "source": {
    "index": "test-index",
    "query": {
      "match": {
        "gender": "female"
      }
    }
  },
  "dest": {
    "index": "test-index-new",
    "type": "female"
  }
}
```

Debug and development

Queries

Get a detailed view of what a query do:

```
GET /blog/_validate/query?explain=true
{
  "query": {
    "match": {
      "title": "Smith"
    }
  }
}
```

Get an explanation about a document matching or not:

```
GET /blog/1/_explain
{
  "query": {
    "match": {
      "title": "Smith"
    }
  }
}
```

Analysis

Test how a content is tokenized in a field:

```
GET /blog/_analyze
{
  "field": "title",
  "text": "powerful"
}
```

Test analyzer token output by analyzer:

```
GET /blog/_analyze
{
  "analyzer": "english",
  "text": "powerful"
}
```

Slowlog

Lower the slowlog threshold to see all the search queries in the logs:

```
PUT /blog/_settings
{
  "index.search.slowlog.threshold.query.trace": "0s",
  "index.search.slowlog.level": "trace"
}
```

Go back to the default configuration:

```
PUT /blog/_settings
{
  "index.search.slowlog.threshold.query.trace": "500ms",
  "index.search.slowlog.level": "info"
}
```

Cluster management and plugins

Running with Docker

Elasticsearch:

```
docker network create elastic
docker run --rm -it --name "elastically_es" -e "discovery.type=single-node" --
```

Have a look at the output for password and enrollment token!

Kibana:

```
docker run --rm -it --name "elastically_kibana" --net elastic -p 5601:5601 doc
```

Cluster and node information

```
GET /_cluster/health?pretty

GET /_cluster/health?wait_for_status=yellow&timeout=50s

GET /_cluster/state

GET /_cluster/stats?human&pretty

GET /_cluster/pending_tasks

GET /_nodes

GET /_nodes/stats

GET /_nodes/nodeId1,nodeId2/stats
```

Get the full reference of **all** the settings:

```
GET /_cluster/settings?include_defaults=true&flat_settings=true
```

Moving shards manually

Ask the index my_index_name shard 0 of node1 to go to node2:

```
POST /_cluster/reroute
{
  "commands": [
    {
      "move": {
        "index": "my_index_name",
        "shard": 0,
        "from_node": "node1",
        "to_node": "node2"
      }
    },
    {
      "allocate": {
        "index": "my_index_name",
        "shard": 1,
        "node": "node3"
      }
    }
  ]
}
```

Updating settings

Disable shard allocation, useful before a rolling restart:

```
PUT /_cluster/settings
{
    "transient" : {
        "cluster.routing.allocation.enable" : "none"
    }
}
```

```
PUT /_cluster/settings
{
    "transient" : {
        "cluster.routing.allocation.enable" : "all"
    }
}
```

Snapshots and Restore

```
PUT /_snapshot/my_backup
{
  "type": "fs",
  "settings": {
    "location": "my_backup_location"
  }
}

PUT /_snapshot/my_backup/snapshot_a
{
  "indices": "index_1,index_2",
  "ignore_unavailable": "true",
  "include_global_state": false
}

POST /_snapshot/my_backup/snapshot_a/_restore
{
  "indices": "index_1,index_2",
  "ignore_unavailable": "true",
  "include_global_state": false,
  "rename_pattern": "index_(.+)",
  "rename_replacement": "restored_index_$1"
}
```

Most useful plugins

> Site plugins are no longer supported, look at Kibana applications or other standalone app like Cerebro for basic management.

Analysis ICU
Adding useful tokenizer and token filters from the Unicode ICU library.

```
bin/elasticsearch-plugin install analysis-icu
```

AWS Cloud
Allow discovery and storage in Amazon cloud (EC2 and S3).

```
bin/elasticsearch-plugin install discovery-ec2
bin/elasticsearch-plugin install repository-s3
```

Azure Cloud
Allow discovery and storage in Microsoft Azure cloud.

```
bin/elasticsearch-plugin install discovery-azure-classic
bin/elasticsearch-plugin install repository-azure
```

Plugins management

```
bin/elasticsearch-plugin install file:///path/to/plugin

bin/elasticsearch-plugin list
```

```
bin/elasticsearch-plugin remove [pluginname]
```

Other information

Where to find the plugin binary?

RPM: `/usr/share/elasticsearch/bin`

Debian: `/usr/share/elasticsearch/bin`

What are the default ports?

Kibana: http://localhost:5601/.

Elasticsearch: http://localhost:9200/.

How to set the correct HEAP SIZE value?

The best value for a single purpose Elasticsearch server is **about 50% of available RAM but under 32g**.
Assuming Ubuntu / Debian server, you can change those files:

/etc/security/limits.conf

```
elasticsearch - nofile 65535
elasticsearch - memlock unlimited
```

/etc/default/elasticsearch (on CentOS/RH: /etc/sysconfig/elasticsearch)

```
ES_HEAP_SIZE=20g
MAX_OPEN_FILES=65535
MAX_LOCKED_MEMORY=unlimited
```

Useful settings to change in elasticsearch.yml

```
cluster.name: jolicluster
node.name: ${HOSTNAME} # by default

network.host: [_local_, _site_]
plugin.mandatory: analysis-icu
node.data: true
node.master: true
node.ingest: true
bootstrap.memory_lock: true
action.auto_create_index: +aaa*,-bbb*,+ccc*,-*
```

```
discovery.seed_hosts:
  - 192.168.1.10:9300
  - 192.168.1.11
  - seeds.mydomain.com

# Needed for first cluster boot
cluster.initial_master_nodes:
  - 10.0.10.101
  - 10.0.10.102:9300
  - 10.0.10.102:9301
  - master-node-name

# Disable X-Pack features, choose wisely
xpack.ccr.enabled: false
xpack.data_frame.enabled: false
xpack.enrich.enabled: false
xpack.flattened.enabled: false
xpack.graph.enabled: false
xpack.ilm.enabled: false
xpack.logstash.enabled: false
xpack.ml.enabled: false
xpack.monitoring.enabled: false
xpack.rollup.enabled: false
xpack.security.enabled: false
xpack.slm.enabled: false
xpack.sql.enabled: false
xpack.transform.enabled: false
xpack.vectors.enabled: false
xpack.watcher.enabled: false
```