

# Beginner's Tutorial on Using Elasticsearch for GIS Applications

## What is ElasticSearch ?

Elasticsearch is a **distributed search and analytics engine** designed for fast querying and analysis of **structured and unstructured** data. It is **document-oriented, storing data in JSON format**. With its powerful RESTful API, Elasticsearch can be easily integrated into websites and applications for **real-time search**.

## Tutorial data :

Data : Latest EarthquakeEruption Dataset (Up to 2023)

Data source : [Kaggle](#) → contains csv and json files

Meta data :

- Key Features :

Column Names	Description
mag	The <b>magnitude</b> of the earthquake or eruption.
place	The <b>geographical location</b> where the event occurred.
time	The <b>date</b> and time of the event.
url	<b>URL providing additional information</b> about the event.
magType	<b>Types of Magnitude:</b>
longitude	<b>longitude</b> of the earthquake location
latitude	<b>latitude</b> of the earthquake location
depth	The <b>depth</b> at which the event occurred below the Earth's surface.

## 1- Create an index :

In this tutorial, we will use [Elasticsearch online](#). After creating your user account, open a **new project**. Then, in **Index Management**, click on '**Create an Index**,' name the index '**earthquake**,' and upload the data. I am using a CSV file.

In the following steps, we will run our queries in the Dev Tools console.

## 2- Search

- Search for places where the magnitude equal to 6 :

```
GET earthquake/_search
```

```
{
  "_source": ["mag", "place"],
  "query": {"match": {"mag": 6}}
}
```

- Result :



The screenshot shows the Elasticsearch Dev Tools console. On the left, the 'Shell' tab is active, displaying a GET request to the 'earthquake/\_search' endpoint. The request body is a JSON object with '\_source' set to ['mag', 'place'] and a 'match' query for 'mag': 6. On the right, the response is shown in a light blue theme. It indicates 1 hit with a total score of 1. The hit details show the index as 'earthquake', the ID as 'ZR9F\_5QB4rzfVPD13X33', and the source as {'mag': 6, 'place': 'Izu Islands, Japan region'}. The status bar at the bottom right shows '200 - OK' and '264 ms'.

```
1 GET earthquake/_search
2 {
3   "_source": ["mag", "place"],
4   "query": {"match": {"mag": 6}}
5 }
6
7
8
9
10
11
12
13
```

```
10 "hits": {
11   "total": {
12     "value": 4,
13     "relation": "eq"
14   },
15   "max_score": 1,
16   "hits": [
17     {
18       "_index": "earthquake",
19       "_id": "ZR9F_5QB4rzfVPD13X33",
20       "_score": 1,
21       "_source": {
22         "mag": 6,
23         "place": "Izu Islands, Japan region"
24       }
25     },
26     {
27       "_index": "earthquake",
```

Clear this input

Clear this output

200 - OK 264 ms

- Search for earthquakes that happened in **Japan**

```
GET earthquake/_search {"_source": ["mag", "place"], "query": { "match_phrase": { "place": "Japan" } } }
```

- Result :

```

1 |
2 GET earthquake/_search
3 {
4   "_source": [
5     "mag",
6     "place"
7   ],
8   "query": {
9     "match_phrase": {
10      "place": "Japan"
11    }
12  }
13 }

```

```

16 "hits": [
17   {
18     "_index": "earthquake",
19     "_id": "DR9F_5QB4rzfVPD1vGro",
20     "_score": 5.1924267,
21     "_source": {
22       "mag": 4.4,
23       "place": "Hokkaido, Japan region"
24     }
25   },
26   {
27     "_index": "earthquake",
28     "_id": "cR9F_5QB4rzfVPD1vG_s",
29     "_score": 5.1924267,
30     "_source": {
31       "mag": 5,
32       "place": "Hokkaido, Japan region"
33     }
34   }
35 ]

```

Clear this input | Clear this output | 200 - OK | 629 ms

- Search for earthquakes that happened in **Morocco**

GET earthquake/\_search

```
{ "_source": [ "mag", "place", "depth"], "query": { "match_phrase": { "place": "Morocco" } } }
```

- Result

```

1 |
2 GET earthquake/_search
3 {
4   "_source": [ "mag", "place", "depth"],
5   "query": {
6     "match_phrase": {
7       "place": "Morocco"
8     }
9   }
10 }

```

```

8 "failed": 0
9 },
10 "hits": {
11   "total": {
12     "value": 1,
13     "relation": "eq"
14   },
15   "max_score": 7.5761967,
16   "hits": [
17     {
18       "_index": "earthquake",
19       "_id": "1R9F_5QB4rzfVPD13YH6",
20       "_score": 7.5761967,
21       "_source": {
22         "mag": 4,
23         "depth": 10,
24         "place": "59 km WSW of Oukaïmedene, Morocco"
25       }
26     }
27   ]
28 }

```

Clear this input | Clear this output | 200 - OK | 764 ms

- Search for earthquakes that happened in **Turkey** with a **magnitude type of mb**

GET earthquake/\_search

```
{ "_source": [ "mag", "place", "magType"], "query": { "bool": { "should": [ { "match_phrase": { "place": "Turkey" } }, { "match": { "magType": "mb" } } ] } } }
```

- Result :



The screenshot shows a REST client interface with a 'Shell' tab. The request is a GET to 'earthquake/\_search' with a JSON body. The response is a JSON array of two earthquake records. The first record has a magnitude of 4.4 and is located 25 km SW of Göksun, Turkey. The second record has a magnitude of 4.5 and is located 18 km SSE of Erciş, Turkey. Both are of type 'mb'.

```
1 GET earthquake/_search
2 {
3   "_source": [ "mag", "place", "magType"],
4   "query": {
5     "bool": { "should": [ { "match_phrase": {
6       "place": "Turkey" } }, { "match": {
7         "magType": "mb" } } ]
8   }
9 }
```

```
29 {
30   "_id": "QB9F_5QB4rzfVPD1vGZp",
31   "_score": 9.57058,
32   "_source": {
33     "mag": 4.4,
34     "place": "25 km SW of Göksun, Turkey",
35     "magType": "mb"
36   },
37   {
38     "_index": "earthquake",
39     "_id": "Uh9F_5QB4rzfVPD1vHdt",
40     "_score": 9.57058,
41     "_source": {
42       "mag": 4.5,
43       "place": "18 km SSE of Erciş, Turkey",
44       "magType": "mb"
45     }
46   },
47 }
```

Clear this input Clear this output 200 - OK 572 ms

- Search for earthquakes that happened in **Turkey** with a magnitude **greater than 4.5**

GET earthquake/\_search

```
{ "_source": [ "mag", "place", "magType"], "query": { "bool": { "must": [ { "match_phrase": { "place": "Turkey" } }, { "range": { "mag": { "gt": 4.5 } } } ] } } }
```

- Result



The screenshot shows a REST client interface with a 'Shell' tab. The request is a GET to 'earthquake/\_search' with a JSON body. The response is a JSON object with a 'hits' array containing one earthquake record. The record has a magnitude of 4.7 and is located 18 km NNE of Bolu, Turkey. It is of type 'mb'.

```
1 GET earthquake/_search
2 {
3   "_source": [ "mag", "place", "magType"],
4   "query": {
5     "bool": {
6       "must": [
7         { "match_phrase": { "place":
8           "Turkey" } },
9         { "range": { "mag": { "gt": 4.5 } } }
10      ]
11    }
12  }
13 }
```

```
12 {
13   "value": 1,
14   "relation": "eq"
15 },
16 "max_score": 7.324204,
17 "hits": [
18   {
19     "_index": "earthquake",
20     "_id": "0h9F_5QB4rzfVPD1vG7r",
21     "_score": 7.324204,
22     "_source": {
23       "mag": 4.7,
24       "place": "18 km NNE of Bolu, Turkey",
25       "magType": "mb"
26     }
27   }
28 ]
29 }
```

Clear this input Clear this output 200 - OK 629 ms

### 3- Aggregation

- Count earthquakes for **every magnitude value**

GET earthquake/\_search

```
{ "_source": ["mag", "place", "magType"], "size": 0, "aggs": { "nb_per_mag": { "terms": { "field": "mag", "size": 100, "order": { "_key": "desc" } } } } }
```

- Result

The screenshot shows a REST client interface with a 'Shell' tab. The request is a GET to 'earthquake/\_search' with a JSON body. The response is a JSON object containing aggregation results for the 'mag' field.

```
1 GET earthquake/_search
2 {
3   "_source": ["mag", "place", "magType"],
4   "size": 0,
5   "aggs": {
6     "nb_per_mag": {
7       "terms": {
8         "field": "mag",
9         "size": 100,
10        "order": {
11          "_key": "desc"
12        }
13      }
14    }
15  }
16 }
17
18
```

```
18 "aggregations": {
19   "nb_per_mag": {
20     "doc_count_error_upper_bound": 0,
21     "sum_other_doc_count": 8469,
22     "buckets": [
23       {
24         "key": 6.9,
25         "doc_count": 1
26       },
27       {
28         "key": 6.7,
29         "doc_count": 2
30       },
31       {
32         "key": 6.4,
33         "doc_count": 2
34       },
35     ]
36   }
37 }
```

Clear this input | Clear this output | 200 - OK | 1030 ms

- Count the number of earthquakes for each **magnitude type**.

GET earthquake/\_search

```
{ "_source": ["mag", "place", "magType"], "size": 0, "aggs": { "nb_per_magType": { "terms": { "field": "magType" } } } }
```

- Result

```

1 GET earthquake/_search
2 {
3   "_source": ["mag", "place", "magType"],
4   "size": 0,
5   "aggs": {
6     "nb_per_magType": {
7       "terms": {
8         "field": "magType"
9       }
10    }
11  }
12 }
13
14
15
22 "buckets": [
23   {
24     "key": "m1",
25     "doc_count": 5775
26   },
27   {
28     "key": "md",
29     "doc_count": 2601
30   },
31   {
32     "key": "mb",
33     "doc_count": 825
34   },
35   {
36     "key": "m1(texnet)",
37     "doc_count": 289
38   },
39 ]

```

Clear this input | Clear this output | 200 - OK | 644 ms

#### 4- Double aggregation

- The **earthquake is more destructive** if the **depth is shallower** and the **magnitude higher**, so let's search for these earthquakes with double aggregation.

```

GET earthquake/_search

{"_source": ["mag", "place", "magType", "depth"], "size": 0, "aggs": { "group_mag": { "terms": { "field": "mag", "order": { "_key": "desc" } } }, "aggs": { "depth_group": { "terms": { "field": "depth", "order": { "_key": "asc" } } } } } } }

```

- Result

```

1 GET earthquake/_search
2 {
3   "_source": ["mag", "place", "magType", "depth"],
4   "size": 0,
5   "aggs": {
6     "group_mag": {
7       "terms": {
8         "field": "mag",
9         "order": { "_key": "desc" }
10      }
11    },
12    "aggs": {
13      "depth_group": {
14        "terms": {
15          "field": "depth",
16          "order": { "_key": "asc" }
17        }
18      }
19    }
20  }
21 }
22
23
24 {
25   "key": 6.9,
26   "doc_count": 1,
27   "depth_group": {
28     "doc_count_error_upper_bound": 0,
29     "sum_other_doc_count": 0,
30     "buckets": [
31       {
32         "key": 53.944,
33         "doc_count": 1
34       }
35     ]
36   },
37   "key": 6.7,
38   "doc_count": 2,
39   "depth_group": {
40     "doc_count_error_upper_bound": 0,
41     "sum_other_doc_count": 0,
42     "buckets": [
43       {
44         "key": 53.944,
45         "doc_count": 1
46       },
47       {
48         "key": 53.944,
49         "doc_count": 1
50       }
51     ]
52   }
53 }

```

Clear this input | Clear this output | 200 - OK | 591 ms

## 5- Geospatial queries

- Create a new index :

PUT geo

```
{"mappings": {"properties": { "location": { "type": "geo_point" } }}}
```

Delete an existing index, and create a new one with uploaded data.

Now, let's test a geographic query:

- Search for earthquakes that occurred **within a distance of 50 km** from **central Alaska** [lat: 65.53, long: -144.69].

GET /earthquake\_geo/\_search

```
{"query": { "geo_distance": { "distance": "50km", "location": { "lat": 65.53, "lon": -144.69 } } }}
```

- Result

The screenshot shows a REST client interface with a 'Shell' tab. The request is a GET to `/earthquake_geo/_search` with a JSON body: `{ "query": { "geo_distance": { "distance": "50km", "location": { "lat": 65.53, "lon": -144.69 } } } }`. The response is a JSON object: `{ "mag": 1.8, "depth": 0.6, "@timestamp": "2023-10-17T19:08:31.686Z", "latitude": 65.739, "location": "65.739,-145.6261", "place": "42 km WNW of Central, Alaska", "time": "1697569711686", "magType": "ml", "url": "https://earthquake.usgs.gov/earthquakes/eventpage/ak023dbx7nvk", "longitude": -145.6261, "_index": "earthquake_geo", "_id": "WR9rBZUB4rzfVPD1cMEa", "score": 1 }`. The status is 200 OK and it took 646 ms.

To conclude, if you have a **large volume of data**, don't waste your time with an SQL database—use a **NoSQL database instead**. Thanks to its **horizontal scalability**, it helps optimize search performance. Elasticsearch is a great tool for this purpose, as it's **open-source**, so you don't need to worry about the cost.