

Pokročilé databázové technológie

Zadanie č. 5 – Elasticsearch

ZADANIE:

5. zadanie zamerané na Elastic:

Odovzdanie do 12.12. 23:59. Zadania 1-8 je dokopy za 7,5 boda tak isto ako aj 9 – 11.

Dokopy teda za 15 bodov. Odovzdávate dokument s popisom a queries separátne každú ako json čo posielate a json čo dostanete ako odpoveď.

1. Rozbehajte si 3 inštancie Elasticsearch-u

2. Vytvorte index pre Tweety, ktorý bude mať "optimálny" počet shardov a replík pre 3 nody (aby tam bola distribúcia dotazov vo vyhľadávaní, aj distribúcia uložených dát)

3. Vytvorte mapping pre normalizované dáta z Postgresu - Tweet musí obsahovať údaje rovnaké ako máte už uložené v PostgreSQL. Dbajte na to, aby ste vytvorili polia v správnom dátovom type (polia ktoré má zmysel analyzovať analyzujte správne, tie ktoré nemá, aby neboli zbytočne analyzované (keyword analyzer)) tak aby index nebol zbytočne veľký.

Mapovanie musí byť striktné.

4. Pre index tweets vytvorte 3 vlastné analyzéry (v settings) nasledovne:

1. Analyzér "englando". Tento analyzér bude obsahovať nasledovné:

1.2. filtre: english_possessive_stemmer, lowercase, english_stop, english_stemmer,

1.3. char_filter: html_strip

1.4. tokenizer: štandardný

- ukážku nájdete na stránke elastic.co pre anglický analyzér

2. Analyzér custom_ngram:

2.2. Filtre: lowercase, asciifolding, filter_ngrams (definujte si ho sami na rozmedzie 1-10)

2.3. char_filter: html_strip

2.4. tokenizer: štandardný

3. Analyzér custom_shingles:

3.2. Filtre: lowercase, asciifolding, filter_shingles (definujte si ho sami a dajte token_separator: "")

3.3. char_filter: html_strip

3.4. tokenizer: štandardný

Do mapovania pridajte:

1. každý anglický text (rátajme že každý tweet a description u autora je primárne v angličtine) nech je analyzovaný novým analyzérom "englando"

2. Priradte analyzery

a. author.name nech má aj mapovania pre custom_ngram, a custom_shingles,

b. author.screen_name nech má aj custom_ngram,

c. author.description nech má aj custom_shingles. Toto platí aj pre mentions, ak tam tie záznamy máte.

3. Hashtagy indexujte ako lowercase

5. Vytvorte bulk import pre vaše normalizované Tweety.

6. Importujete dáta do Elasticsearchu prvych 5000 tweetov

7. Experimentujte s nódami, a zistite koľko nódov musí bežať (a ktoré) aby vám Elasticsearch vedel pridávať dokumenty, mazať dokumenty, prezerať dokumenty a vyhľadávať nad nimi?

Dá sa nastaviť Elastic tak, aby mu stačil jeden nód?

8. Upravujte počet retweetov pre vami vybraný tweet pomocou vášho jednoduchého scriptu (v rámci Elasticsearchu) a sledujte ako sa mení `_seq_no` a `_primary_term` pri tom ako zabíjate a spúšťate nódy.

9. Zrušte repliky a importujete všetky tweety

10. Vyhľadajte vo vašich tweetoch spojenie "gates s0ros vaccine micr0chip". V query použite `function_score`, kde jednotlivé medzikroky sú nasledovné:

Query:

1. Must - vyhľadajte vo viacerých poliach (konkrétne: `author.name` (pomocou `shingle`), `content` (cez analyzovaný anglický text), `author.description` (pomocou `shingles`), `author.screen_name` (pomocou `ngram`)) spojenie "gates s0ros vaccine micr0chip", zapojte podporu pre preklepy, operátor je OR.

2.1 tieto polia vo vyhľadávaní boost-nite nasledovne - `author.name * 6`, `content * 8`, `author.description * 6`, `author.screen_name * 10`.

3. Filter - vyfiltrujte len tie, ktoré majú `author.statuses_count > 1000` a tie, ktoré majú hashtag „qanon“

4. Should – boost-nite 10 krat tie, ktoré obsahujú v `mentions.name` (tento objekt je typu `nested`) cez `ngram string "real"`.

5. Nastavte podmienené váhy cez `functions` nasledovne:

5.1. `retweet_count`, ktorý je väčší rovný ako 100 a menší rovný ako 500 na 6,

5.2. `author.followers_count` väčší ako 100 na 3

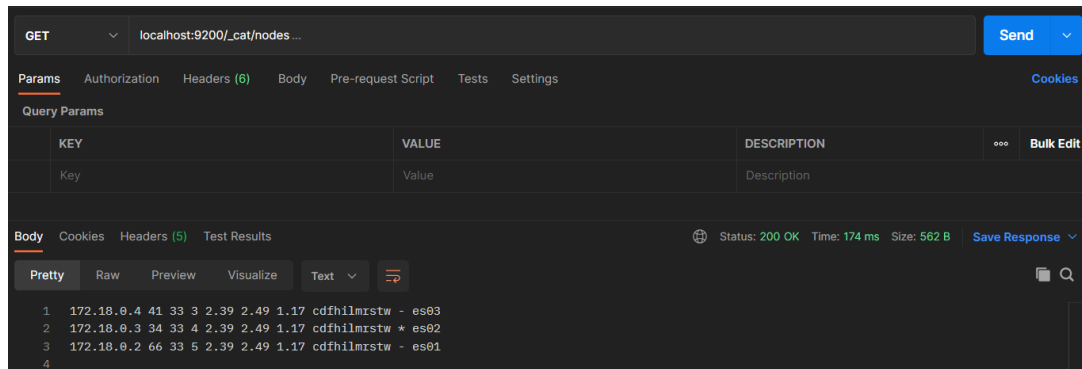
Zobrazte agregácie pre výsledky na konci. Vytvorte `bucket hashtags` podľa hashtagov a spočítajte hodnoty výskytov (na webe by to mohli byť `facets`).

11. Konšpiračné teórie podľa Elasticu. Pracujte zo všetkými tweetami, ktoré máte. Následne pre všetky týždne zistite pomocou vnorených agregácií, koľko `retweet_count` sumárne majú tweety ktoré majú hashtagy z prvého zadania. Teda na základe hashtagov znova rozdeľte tweety do konšpiračných teórií ale pomocou agregácií.

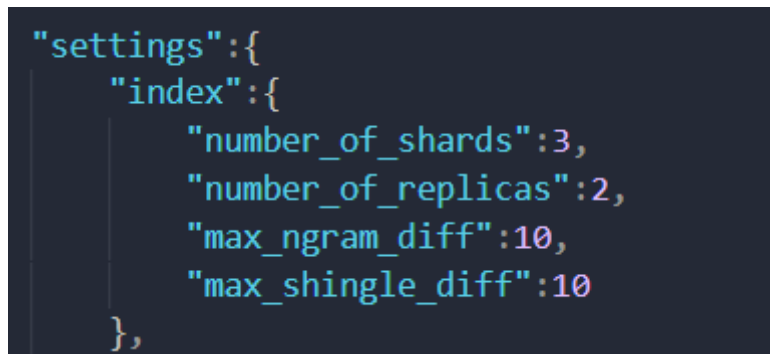
Riešenie:

Úloha 1:

Ako prvé sme si najprv stiahli windows subsystem for linux, cez ktorý sme spúšťali docker a teda aj samotný elastic, ktorý sme najprv stiahli a setupli pomocou inštalačného guidu na oficiálnej stránke elasticu a tak sme si rozbehali 3 inštancie/nody elasticu. Dolu na obrázku môžeme cez postmana vidieť naše inštancie.



Úloha 2:



Úloha 3:

Mapping som vytvoril tak, ako bolo opísané v zadaní úlohy, teda striktné mapovanej, ktoré je priložené v priečinku s riešením, a atribúty by viac menej zodpovedať tým čo sú v postgrese. Pri riešení mám vlastne dva mappingy prvý, ktorý obsahuje parenta(embednutý tweet) a druhý, ktorý parenta nemá, dôvod prečo to takto mám, že pri exportovaní dát z postgresu vzťah medzi parent tweetom a child tweetom dosť času žral a tak som sa od úlohy 9 rozhodol používať mapping bez parenta, keďže bolo samotne naznačené aj na prednáške, že štruktúra toho mappingu závisí najmä od use casu, tak mi to neprišiel ako veľký problém odstrániť parenta keďže v ďalších úlohách sa s ním nepracuje. Rovnako ako prvý tak aj druhý mapping je priložený v súbore.

Úloha 4:

Dané analyzéry sú súčasťou mappingu.

```
"analysis":{
  "analyzer":{
    "englando":{
      "type":"custom",
      "tokenizer":"standard",
      "char_filter":["html_strip"],
      "filter":["english_possessive_stemmer", "lowercase", "english_stop", "english_stemmer"]
    },
    "custom_ngram":{
      "type":"custom",
      "tokenizer":"standard",
      "char_filter":["html_strip"],
      "filter":["lowercase", "asciifolding","filter_ngrams"]
    },
    "custom_shingles":{
      "type":"custom",
      "tokenizer":"standard",
      "char_filter":["html_strip"],
      "filter":["lowercase", "asciifolding","filter_shingles"]
    }
  }
},
```

Úloha 5:

Predtým ako som si bulk importol tweety, tak som si ich najprv cez python sqlalchemy ORM (dumping.py) vytiahol a uložil v bulk json formáte, ktorý môžeme vidieť nižšie.

```
{"index":{"_id":"1204623174912094213", "routing":"1204623174912094213"}}
{"content": "Seven Podcast Recommendations for Discussing China https://t.co/OazSsAyBqH", "location": {"lat": 22.3048612, "lon": 114.169
{"index":{"_id":"1204638276281421824", "routing":"1204638276281421824"}}
{"content": "#China #chinaa50 #chinausa #Animals #travel #TravelChina #ChinaTravel #travelwithkids #travelbloggers #travelphotography #fa
```

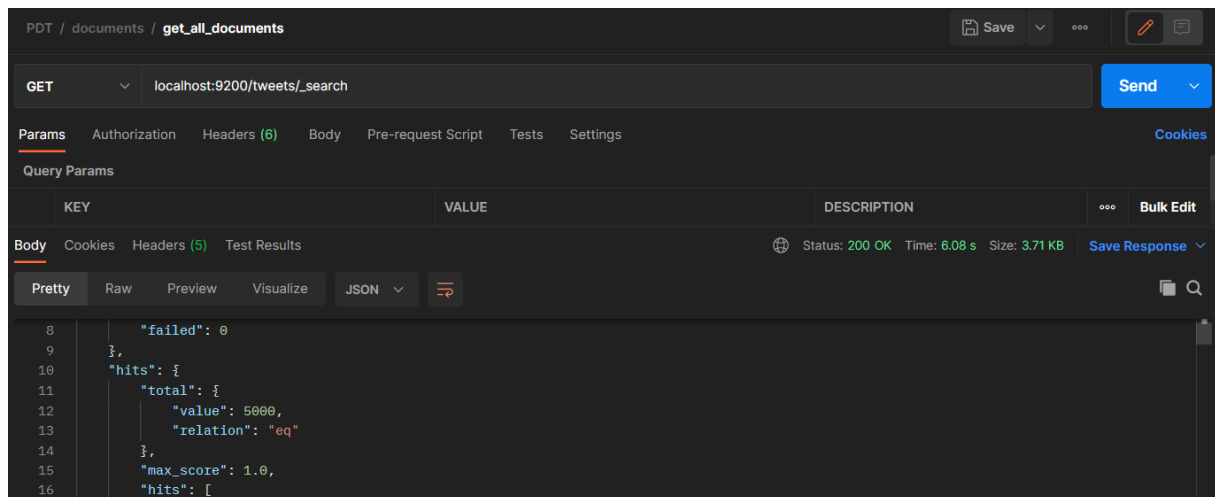
Potom som si spravil bulk script v bashy, ktorý mi dané dáta po 25 000 tweetov pridával do elasticu cez CURL.

Úloha 6:

Prvých 5000 tweetov som importoval, tak že pôvodný script, čo vyberal dáta z postgresu som upravil aby namiesto 25 000 mi dal len 5 000 tweetov a tie som jednoducho cez CURL v cmd bulk importol do elasticu.

CURL príkaz:

```
curl -H "Content-Type: application/x-ndjson" -XPOST localhost:9200/tweets/_bulk --
data-binary "@psql_dump_with_parent2.txt"
```



Úloha 7 :

3 nodes (1 hlavný 2 vedľajšie) :

ip	heap.percent	ram.percent	cpu	load_1m	load_5m	load_15m	node.role	master	name
172.18.0.3	30	33	10	3.71	1.38	0.65	cdfhilmrstw	-	es02
172.18.0.4	38	33	9	3.71	1.38	0.65	cdfhilmrstw	*	es01
172.18.0.2	60	33	9	3.71	1.38	0.65	cdfhilmrstw	-	es03

- Vyhľadávanie funguje
- Prezeranie funguje
- Pridávanie funguje
- Mazanie funguje
- Update (script – zmena retweet_count) funguje

```

    "_seq_no": 10363,
    "_primary_term": 5

```

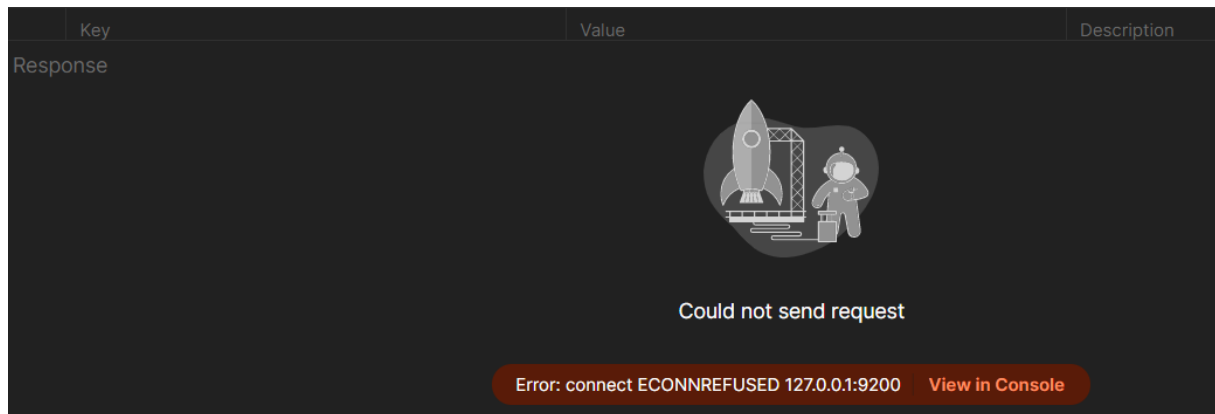
2 nodes (1 hlavný 1 vedľajší)

ip	heap.percent	ram.percent	cpu	load_1m	load_5m	load_15m	node.role	master	name
172.18.0.2	35	25	0	2.62	1.78	0.87	cdfhilmrstw	-	es03
172.18.0.4	73	25	0	2.62	1.78	0.87	cdfhilmrstw	*	es01

- Vyhľadávanie funguje
- Prezeranie funguje
- Pridávanie funguje
- Mazanie funguje
- Update (script – zmena retweet_count) funguje

```
"_seq_no": 10364,  
"_primary_term": 5
```

2 nodes (2 vedľajšie)



Postman uz nefunguje tak skusime cez docker ps.

```
denis@DESKTOP-NA5C1EG:~$ docker ps  
CONTAINER ID   IMAGE                                COMMAND                  CREATED         STATUS  
7aed39b91acf   docker.elastic.co/elasticsearch:7.15.2  "/bin/tini -- /usr/l..." 10 minutes ago Up 10 mi  
nutes          9200/tcp, 9300/tcp                  es03  
bd02eba01ac5   docker.elastic.co/elasticsearch:7.15.2  "/bin/tini -- /usr/l..." 10 minutes ago Up About  
a minute      9200/tcp, 9300/tcp                  es02
```

- Vyhľadavanie nefunguje
- Prezeranie nefunguje
- Pridavanie nefunguje
- Mazanie nefunguje
- Update (script – zmena retweet_count) nefunguje

Teda nefunguje nič, ako v prípade bežného dňa fitkára.

1 node(vedľajší)

- Nefunguje nič

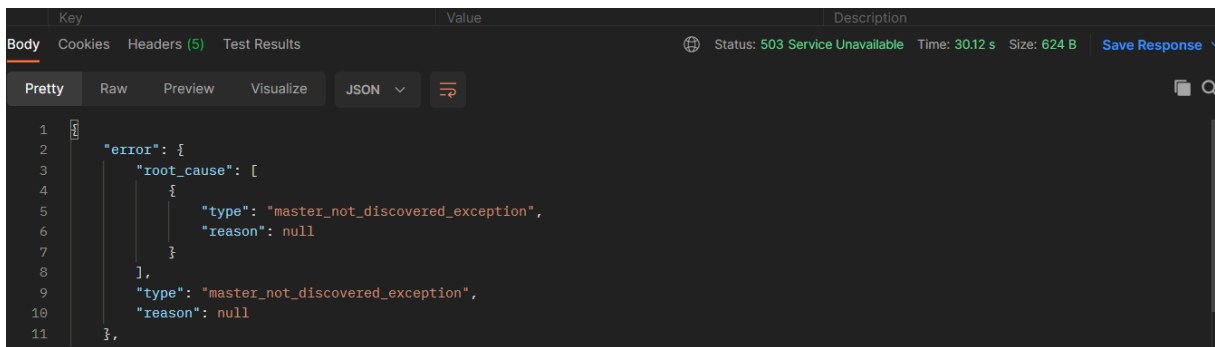
```
CONTAINER ID   IMAGE                                COMMAND                  CREATED         STATUS  
bd02eba01ac5   docker.elastic.co/elasticsearch:7.15.2  "/bin/tini -- /usr/l..." 23 minutes ago Up 14 mi  
nutes          9200/tcp, 9300/tcp                  es02  
denis@DESKTOP-NA5C1EG:~$
```

1 node(hlavný)

```
denis@DESKTOP-NA5C1EG:~$ docker ps  
CONTAINER ID   IMAGE                                COMMAND                  CREATED         STATUS  
fea4fcfd3c3d   docker.elastic.co/elasticsearch:7.15.2  "/bin/tini -- /usr/l..." 43 minutes ago Up 18 mi  
nutes          0.0.0.0:9200->9200/tcp, :::9200->9200/tcp, 9300/tcp  es01
```

- Nedá sa pingnut ale funguje vyhľadavanie
- Prezeranie funguje

- Mazanie nefunguje
- Pridávanie nefunguje
- Update (script – zmena retweet_count) nefunguje



Úloha 8:

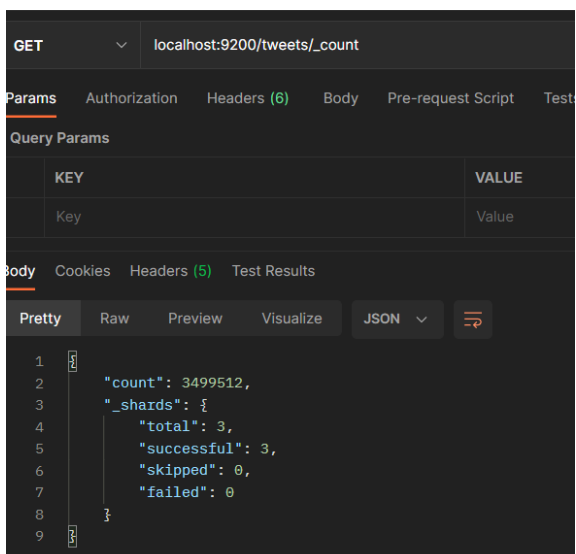
Používal som dva skripty:



_seq_no rástol a _primary_term sa nemenil

Úloha 9:

Pôvodne som chcel importovať aspoň tretinu tweetov teda niečo okolo 10M ale vzhľadom na to, že import ako aj export mi trvali neskutočne dlho a len pri exporte by mi to trvalo okolo 3 dni tak som sa rozhodol exportnúť a importnúť niečo okolo 3,4M tweetov. Cca 3.4M preto, lebo ku koncu mi aj došiel úložný priestor(pôvodne som chcel 3,5). Za to že som neimportol všetky tweety sa ospravedlňujem.



Úloha 10:

1.

Request

```
{
  "query": {
    "bool": {
      "must": {
        "multi_match": {
          "query": "gates s0ros vaccine micr0chip",
          "fields": [
            "author.name.shingle",
            "content",
            "author.description.shingle",
            "author.screen_name"
          ],
          "fuzziness": "AUTO"
        }
      }
    }
  }
}
```

Response

```
Pretty  Raw  Preview  Visualize  JSON  ↕
4      _shards : {
5        "total": 3,
6        "successful": 3,
7        "skipped": 0,
8        "failed": 0
9      },
10     "hits": {
11       "total": {
12         "value": 10000,
13         "relation": "gte"
14       },
15       "max_score": 22.485277,
16       "hits": [
17         {
18           "_index": "tweets",
19           "_type": "_doc",
20           "_id": "1226810263437434880",
21           "_score": 22.485277,
22           "_routing": "1726810263437434880"
23         }
24       ]
25     }
26   }
27 }
```

2.1.

Request

```
{
  "query": {
    "bool": {
      "must": {
        "multi_match": {
          "query": "gates s0ros vaccine micr0chip",
          "fields": [
            "author.name.shingle^6",
            "content^8",
            "author.description.shingle^6",
            "author.screen_name^10"
          ],
          "fuzziness": "AUTO"
        }
      }
    }
  }
}
```

Response

```
    },
    "hits": {
      "total": {
        "value": 10000,
        "relation": "gte"
      },
      "max_score": 150.29823,
      "hits": [
        {
          "_index": "tweets",
          "_type": "_doc",
          "_id": "1220219270342291456",
          "_score": 150.29823,
          "_routing": "1220219248297103360",
          "_source": {
            "content": "RT @SaRaAshcraft: @jturner63 @JenJustjen515 @AutisticByronII @atensnut Fentanyl/Opioids (Sachler), Vaccines (Gates), civil unrest (Soros), \\"_..."
          }
        }
      ]
    }
  ]
}
```

3.

Request

To iste ako predchádzajúca query + filter

```
{
  "filter": [
    {
      "term": {
        "hashtags": "qanon"
      }
    },
    {
      "range": {
        "author.statuses_count": {
          "gt": 1000
        }
      }
    }
  ]
}
```

Response

```
    "hits": {
      "total": {
        "value": 764,
        "relation": "eq"
      },
      "max_score": 140.5072,
      "hits": [
        {
          "_index": "tweets",
          "_type": "_doc",
          "_id": "1221840734271299585",
          "_score": 140.5072,
          "_routing": "1220807218876076033",
          "_source": {
            "content": "RT @RokJohannes: #BillGates #Vaccination #CIA #Soros #DeepState #WWG1WGA #QAnon #MAGA #DrainTheSwamp\n\nBill Gates kept telling us a pandemi..."
          }
        }
      ]
    }
  ]
}
```

4.

Request

Ku query z 2.1 som pridal nasledujúci match a filter

```
{
  "should": {
    "nested": {
      "path": "mentions",
      "query": {
        "match": {
          "mentions.name": {
            "query": "real",
            "boost": 10
          }
        }
      }
    }
  },
  "filter": [
    {
      "range": {
        "author.statuses_count": {
          "gt": 1000
        }
      }
    }
  ]
}
```

Response

```
{
  "took": 8317,
  "timed_out": false,
  "_shards": {
    "total": 3,
    "successful": 3,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 764,
      "relation": "eq"
    },
    "max_score": 147.59425,
    "hits": [
      {
        "_index": "tweets",
        "_type": "_doc",
        "_id": "1221840734271299585",
        "_score": 147.59425,
        "_routing": "1220807218876076033",

```

5.

5.1.

Request

```
44 .....}
45 .....]
46 .....}
47 .....},
48 ✓ ..... "functions": [
49 ✓ ..... [
50 ✓ .....   "filter": [
51 ✓ .....     {
52 ✓ .....       "range": {
53 ✓ .....         "retweet_count": {
54 .....           "gte": 100,
55 .....           "lte": 500
56 .....         }
57 .....       }
58 .....     }
59 .....   ],
60 .....   "weight": 6
```

Response

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▾

```
51 ..... {
52 .....   "range": {
53 .....     "retweet_count": {
54 .....       "gte": 100,
55 .....       "lte": 500
56 .....     }
57 .....   }
58 ..... }
59 ..... ],
```

Body Cookies Headers (5) Test Results 🌐 Status: 200 OK Time: 4.00 s

Pretty Raw Preview Visualize **JSON** ▾

```
13 .....   "relation": "eq"
14 ..... },
15 ..... "max_score": 376.4063,
16 ..... "hits": [
17 .....   {
18 .....     "_index": "tweets",
19 .....     "_type": "_doc",
20 .....     "_id": "1084468832415166467",
21 .....     "_score": 376.4063,
22 .....     "_routing": "1084468832415166467",
23 .....     "source": {
```

5.2.

Request

```
52 ..... "range": {
53 .....   "retweet_count": {
54 .....     "gte": 100,
55 .....     "lte": 500
56 .....   }
57 ..... }
58 ..... }
59 ..... ],
60 ..... "weight": 6
61 ..... },
62 ..... ],
63 ..... "filter": [
64 .....   {
65 .....     "range": {
66 .....       "author.followers_count": {
67 .....         "gt": 100
68 .....       }
69 .....     }
70 .....   }
71 ..... ],
72 ..... "weight": 3
73 ..... ]
```

Response

```
75 ..... }
76 ..... },
77 ..... "aggs": {
78 .....   "bucket_hashtags": {
79 .....     "terms": {
80 .....       "field": "hashtags"
81 .....     }
82 .....   }
83 ..... }
84 ..... }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 5.30 s Size: 4.36 kB Save Response

Pretty Raw Preview Visualize JSON

```
477 ..... "bucket_hashtags": {
478 .....   "doc_count_error_upper_bound": 19,
479 .....   "sum_other_doc_count": 1354,
480 .....   "buckets": [
481 .....     {
482 .....       "key": "qanon",
483 .....       "doc_count": 764
484 .....     },
485 .....     {
486 .....       "key": "coronavirus",
```

Úloha 11:


Request

```
2  ... "aggs": {
3  ...   "agg_week": {
4  ...     "date_histogram": {
5  ...       "field": "happened_at",
6  ...       "calendar_interval": "week"
7  ...     },
8  ...     "aggs": {
9  ...       "Deep State": {
10 ...         "filter": {
11 ...           "terms": {
12 ...             "hashtags": ["DeepstateVirus", "DeepStateVaccine", "DeepStateFauci"]
13 ...           }
14 ...         },
15 ...         "aggs": {
16 ...           "retweet_deep_state": {
17 ...             "sum": {
18 ...               "field": "retweet_count"
19 ...             }
20 ...           }
21 ...         }
22 ...       }
23 ...     }
24 ...   }
25 ... }
```

Response

```
23  ... "Qanon": {
24  ...   "filter": {
25  ...     "terms": {
26  ...       "hashtags": ["QAnon", "MAGA", "WWG1WGA"]
27  ...     }
28  ...   },
29  ...   "aggs": {
30  ...     "retweet_qanon": {
31  ...       "sum": {
32  ...         "field": "retweet_count"
33  ...       }
34  ...     }
35  ...   }
36  ... }
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON 

```
3458  ... "Qanon": {
3459  ...   "doc_count": 0,
3460  ...   "retweet_qanon": {
3461  ...     "value": 0.0
3462  ...   }
3463  ... },
3464  ... "Moon landing is fake": {
3465  ...   "doc_count": 0,
3466  ...   "retweet_moon_landing_fake": 5
```

Výstupy jednotlivých query sa nachádzajú v priečinku, ktorý je opísaný v README.