

WPROWADZENIE DO ELASTICSEARCHA

CZYLI JAK ULEPSZYĆ SWOJĄ
WYSZUKIWARKĘ

O MNIE



Damian Michalik

FullStack Developer w
eSky.pl

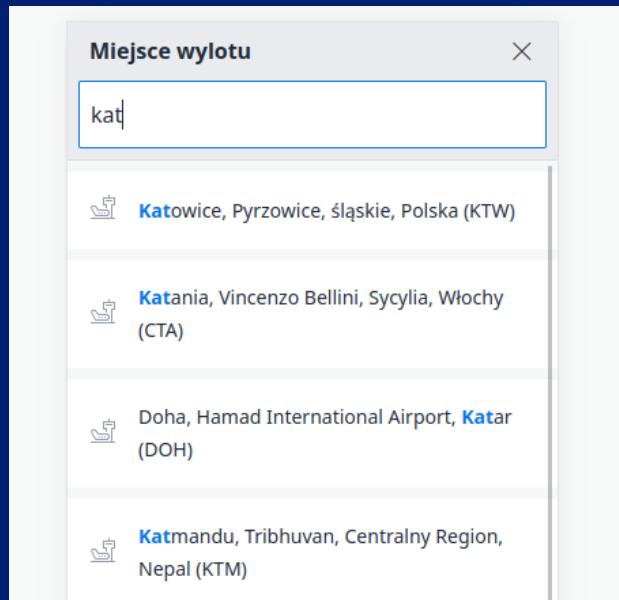
Kontakt

damian.michalik@esky.com

LinkedIn

PRZYCHODZI BIZNES I MÓWI...

Jako użytkownik chciałbym w prosty sposób wyszukać lotnisko wylotu / przylotu



IMPLEMENTACJA BY JUNIOR DEVELOPER

id	code	name
1	KTW	Katowice
2	KRK	Kraków
3	WAW	Warszawa
4	WMI	Modlin

```
SELECT * FROM airports WHERE name LIKE '%INPUT%';
```

TYMCZASEM W SOBOTE...

Oczekiwania



Rzeczywistość

Miejsce wylotu X

Brak wyników
Sprawdź pisownię lub wybierz inne miejsce.



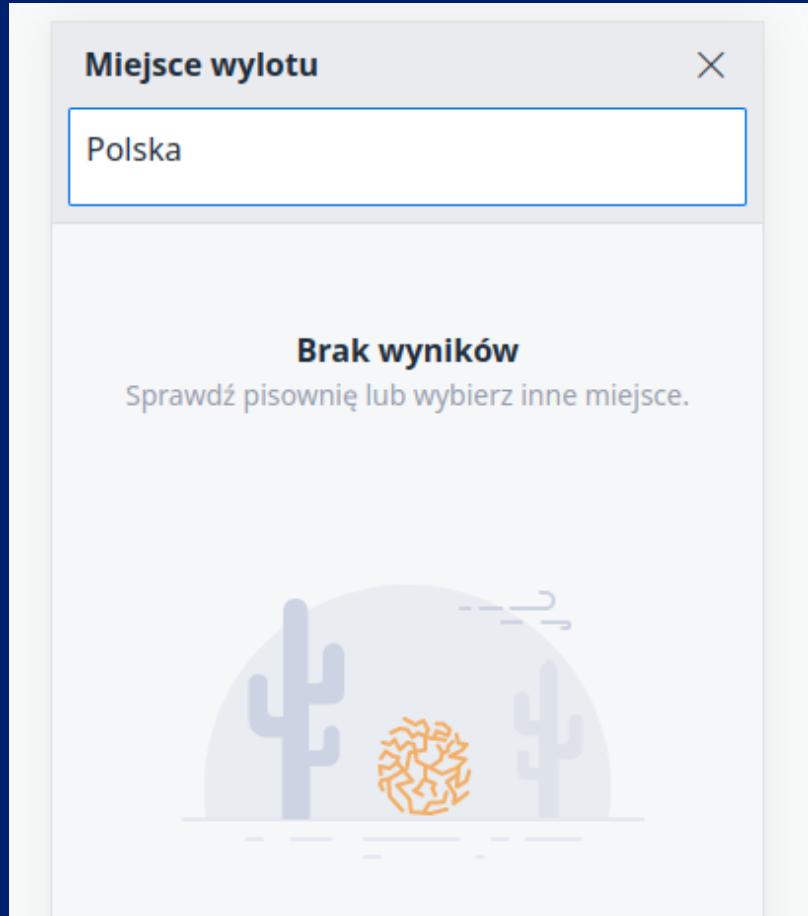
SZYBKI FIX :)

```
ALTER TABLE airports ADD city VARCHAR(255);
```

id	code	name	city
1	KTW	Pyrzowice	Katowice
2	KRK	Balice	Kraków
3	WAW	Okęcie	Warszawa
4	WMI	Modlin	Warszawa

```
SELECT * FROM airports  
WHERE name LIKE '%INPUT%' OR city LIKE '%INPUT%';
```

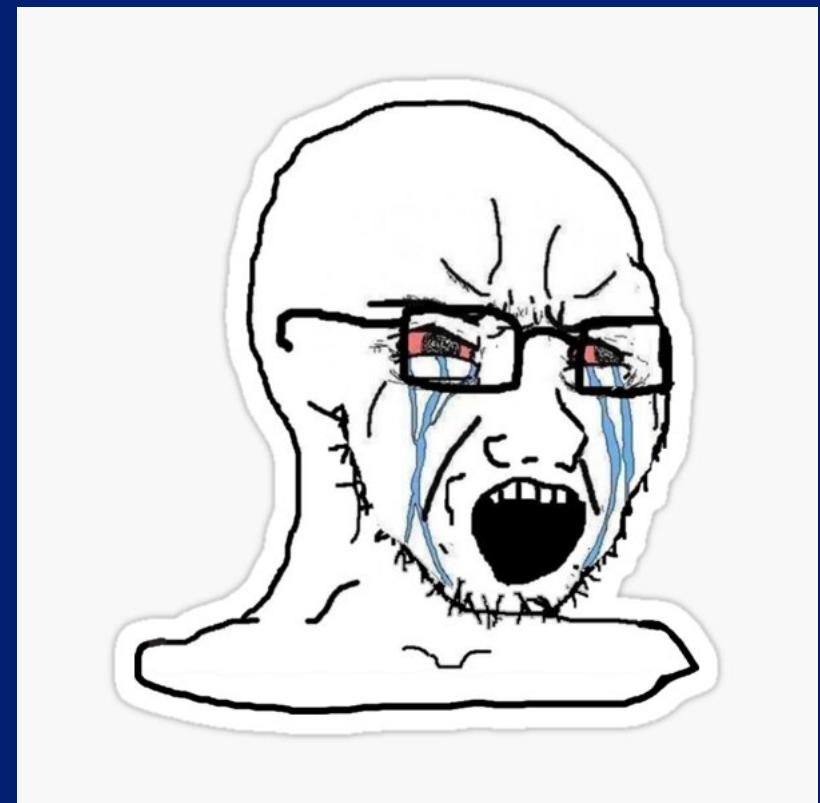
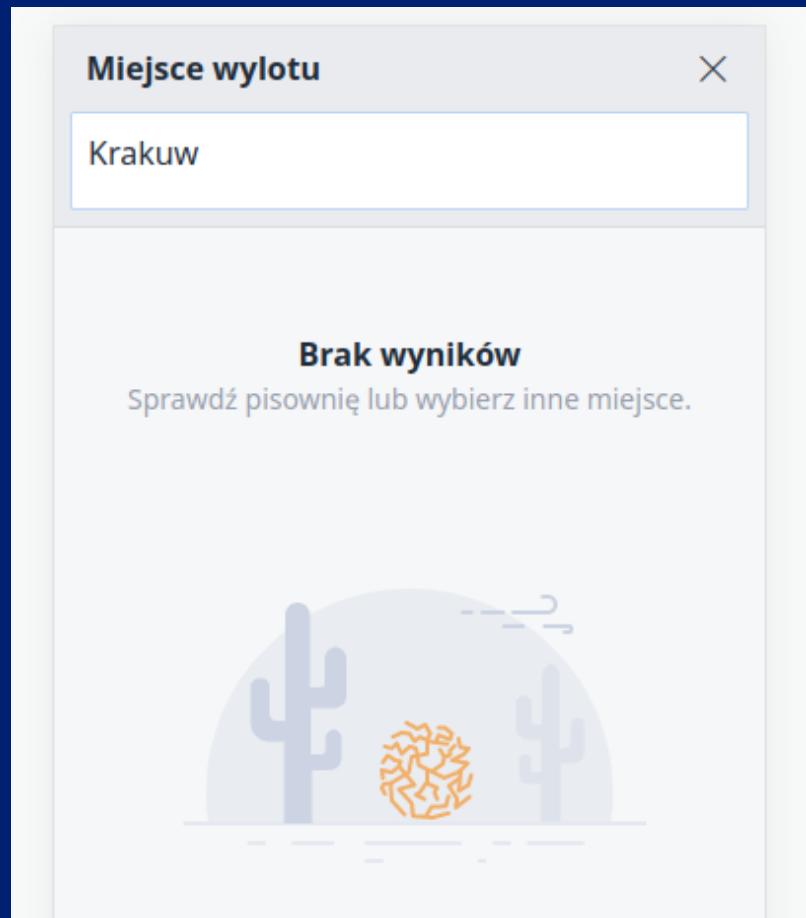
KOLEJNY DZIEŃ W PRACY



```
ALTER TABLE airports  
ADD country VARCHAR(255);
```

```
SELECT * FROM airports  
WHERE name LIKE '%INPUT%'  
OR city LIKE '%INPUT%'  
OR country LIKE '%INPUT%';
```

A CO Z BŁĘDAMI?



JAK PODEJŚĆ DO PROBLEMU W INNY SPOSÓB?

Wyszukiwanie pełnotekstowe

CZYM JEST ELASTICSEARCH?



Elasticsearch jest silnikiem zajmującym się wyszukiwaniem oraz analizą danych.

Do wyszukiwania Elasticsearch wykorzystuje bibliotekę [Apache Lucene](#)

WAŻNE TERMINY

- Dokument - rekord w bazie Elasticsearch, zapisywane w formacie JSON
- Indeks - kolekcja dokumentów o zbliżonej charakterystyce, jest identyfikowany nazwą, po której odwołuje się do niego podczas wszelkich operacji takich jak dodawanie danych, aktualizacja, usuwanie czy wyszukiwanie.

INDEKS ODWRÓCONY

Indeks
...



Dokument 1
warszawa okęcie



Dokument 2
warszawa modlin

	Dokument 1
okęcie	X
warszawa	X

	Dokument 1	Dokument 2
modlin		X
okęcie	X	
warszawa	X	X

ELASTICSEARCH - JAK URUCHOMIĆ LOKALNIE?

Docker Compose

```
version: "3.8"
services:
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:8.6.0
    environment:
      - discovery.type=single-node
      - ES_JAVA_OPTS=-Xms512m -Xmx512m
      - xpack.security.enabled=false
    ports:
      - 9200:9200
```

ELASTICSEARCH - JAK URUCHOMIĆ LOKALNIE?

Weryfikacja czy Elasticsearch działa

```
curl "http://localhost:9200/"
```

```
{
  "name": "3825e7c2175f",
  "cluster_name": "docker-cluster",
  "cluster_uuid": "n4h8NjVkJRmXhmdkI0TrMA",
  "version": {
    "number": "8.6.0",
    "build_flavor": "default",
    "build_type": "docker",
    "build_hash": "f67ef2df40237445caa70e2fef79471cc608d70d",
    "build_date": "2023-01-04T09:35:21.782467981Z",
    "build_snapshot": false,
    "lucene_version": "9.4.2",
    "minimum_wire_compatibility_version": "7.17.0",
    "minimum_index_compatibility_version": "7.0.0"
  },
  "tagline": "You Know, for Search"
}
```

ELASTICSEARCH - KLIENT PHP

Oficjalny klient jest dostępny pod adresem

<https://github.com/elastic/elasticsearch-php>

Instalacja z użyciem Composera

```
composer require elastic/elasticsearch
```

ELASTICSEARCH - KLIENT PHP

Weryfikacja czy możemy się połączyć z
Elasticsearchem

```
use Elastic\Elasticsearch\ClientBuilder;

$client = ClientBuilder::create()
    ->setHosts(['localhost:9200'])
    ->build();

$response = $client->info();

echo $response['version']['number'];
```

TWORZENIE INDEKSÓW

Jeśli wstawiamy dokument do nieistniejącego indeksu, Elasticsearch stworzy taki indeks automatycznie.

Przy takim podejściu Elasticsearch sam podczas indeksowania danych dopasuje typ dla poszczególnych pól.

Jeśli chcemy mieć większą kontrolę na procesem tworzenia indeksu i późniejszej indeksacji danych musimy samemu ustawić odpowiednie parametry.

TWORZENIE INDEKSU Z MAPOWANIEM

```
curl -X PUT "http://localhost:9200/airports_http" \
-H 'Content-Type: application/json' \
-d' {
  "mappings": {
    "properties": {
      "name": { "type": "text" },
      "code": { "type": "keyword" },
      "city": { "type": "text" },
      "country": { "type": "text" }
    }
  }
}'
```

TWORZENIE INDEKSU Z MAPOWANIEM

```
$params = [
    'index' => 'airports_php',
    'body' => [
        'mappings' => [
            'properties' => [
                'code' => ['type' => 'keyword'],
                'name' => ['type' => 'text'],
                'city' => ['type' => 'text'],
                'country' => ['type' => 'text']
            ]
        ]
    ]
];
$client->indices()->create($params);
```

WSTAWIANIE DANYCH

```
curl -X POST "http://localhost:9200/airports_http/_doc/" \
-H "Content-Type: application/json" \
-d '{
  "code": "KTW",
  "name": "Pyrzowice",
  "city": "Katowice",
  "country": "Polska"
}'
```

WSTAWIANIE DANYCH

```
$params = [  
    'index' => 'airports_php',  
    'body' => [  
        'code' => 'KTW',  
        'name' => 'Pyrzowice',  
        'city' => 'Katowice',  
        'country' => 'Polska'  
    ]  
];  
  
$client->index($params);
```

PROCES INDEKSACJI DANYCH

Podczas indeksowania danych Elasticsearch dokonuje analizy danych.

Proces analizy danych odbywa się następująco:

1. Dane są filtrowane
2. Przefiltrowane dane są poddawane procesowi tokenizacji (rozbicia tekstu na tokeny, z reguły jeden token to jedno słowo)
3. Tokeny są poddawane filtracji

PROCES INDEKSACJI DANYCH

W procesie analizy można używać analizatorów wbudowanych w Elasticsearcha lub zdefiniować własny

W przypadku danych tekstowych analizie podlegają tylko pola typu **text**

Pola typu **keyword** nie są poddawane analizie

STANDARD ANALYZER

Domyślny analizator używany jeśli nie został wprost zdefiniowany inny analizator

Składa się z:

- Tokenizera Standard Tokenizer
- Filtru Lower Case Token Filter
- Filtru Stop Token Filter (domyślnie wyłączonego)

ANALIZA DANYCH - PRZYKŁAD

```
curl -X POST "http://localhost:9200/_analyze" \
-H "Content-Type: application/json" \
-d '{
  "analyzer": "standard",
  "text": "Warsaw-Okęcie CHOPIN Airport."
}'
```

WSTAWIANIE WIĘKSZEJ ILOŚCI DANYCH DO INDEKSU

```
{ "index" : { "_index" : "airports_http" } }
{"code": "KRK", "name": "Balice", "city": "Kraków", "country": "Polska"}
{ "index" : { "_index" : "airports_http" } }
{"code": "WMI", "name": "Modlin", "city": "Warszawa", "country": "Polska"}
{ "index" : { "_index" : "airports_http" } }
{"code": "WAW", "name": "Okęcie", "city": "Warszawa", "country": "Polska"}
{ "index" : { "_index" : "airports_http" } }
{"code": "WRO", "name": "Strachowice", "city": "Wrocław", "country": "Polska"}
{ "index" : { "_index" : "airports_http" } }
{"code": "IEG", "name": "Babimost", "city": "Zielona Góra", "country": "Polska"}
```

```
curl -X POST "http://localhost:9200/_bulk" -H 'Content-Type: application/json' \
--data-binary "@airports.json"
```

WSTAWIANIE WIĘKSZEJ ILOŚCI DANYCH DO INDEKSU

```
$airports = [
    ['code' => 'KRK', 'name' => 'Balice', 'city' => 'Kraków', 'country' => 'Polska'],
    ['code' => 'WMI', 'name' => 'Modlin', 'city' => 'Warszawa', 'country' => 'Polska']
];
foreach ($airports as $airport) {
    $params['body'][] = ['index' => ['_index' => 'airports_php']];
    $params['body'][] = [
        'code' => $airport['code'],
        'name' => $airport['name'],
        'city' => $airport['city'],
        'country' => $airport['country']
    ];
}
$responses = $client->bulk($params);
```

ALE MIAŁO BYĆ O
WYSZUKIWANIU

STILL WAITING



MATCH ALL

```
curl "http://localhost:9200/airports_http/_search" \
-H "Content-Type: application/json" \
-d '{
  "query": {
    "match_all": {}
  }
}'
```

```
$searchParams = [
  'index' => 'airports_http',
  'body'  => [
    'query' => [
      'match_all' => (object) []
    ]
  ]
];

$response = $client->search($searchParams);
```

WYNIK

```
{  
  "took": 1,  
  ...  
  "hits": {  
    "total": {  
      "value": 1,  
      ...  
    },  
    "max_score": 1.0,  
    "hits": [  
      {  
        "_index": "airports_http",  
        "_id": "PRBmqYcBV7U37YZbCB8f",  
        "_score": 1.0,  
        "_source": {  
          "code": "KTW"  
          ...  
        }  
      }  
    ]  
  }  
}
```

ZAPYTANIE TERM

```
curl "http://localhost:9200/airports_http/_search" \
-H "Content-Type: application/json" \
-d '{
  "query": {
    "term": {
      "city": "katowice"
    }
  }
}'
```

```
$searchParams = [
  'index' => 'airports_php',
  'body'  => [
    'query' => [
      'term' => [
        'city' => 'katowice'
      ]
    ]
  ]
];

$response = $client->search($searchParams);
```

ZAPYTANIE TERM

```
curl "http://localhost:9200/airports_http/_search" \
-H "Content-Type: application/json" \
-d '{
  "query": {
    "term": {
      "city": "Katowice"
    }
  }
}'
```

```
{
  "hits": {
    "hits": []
  }
}
```

ANALIZA TEKSTU RAZ JESZCZE

Elasticsearch w zależności od rodzaju zapytania dokonuje również analizy szukanej frazy

Jednak w przypadku zapytania **term** analiza nie jest przeprowadzana, w związku z tym np. wielkość liter ma znaczenie

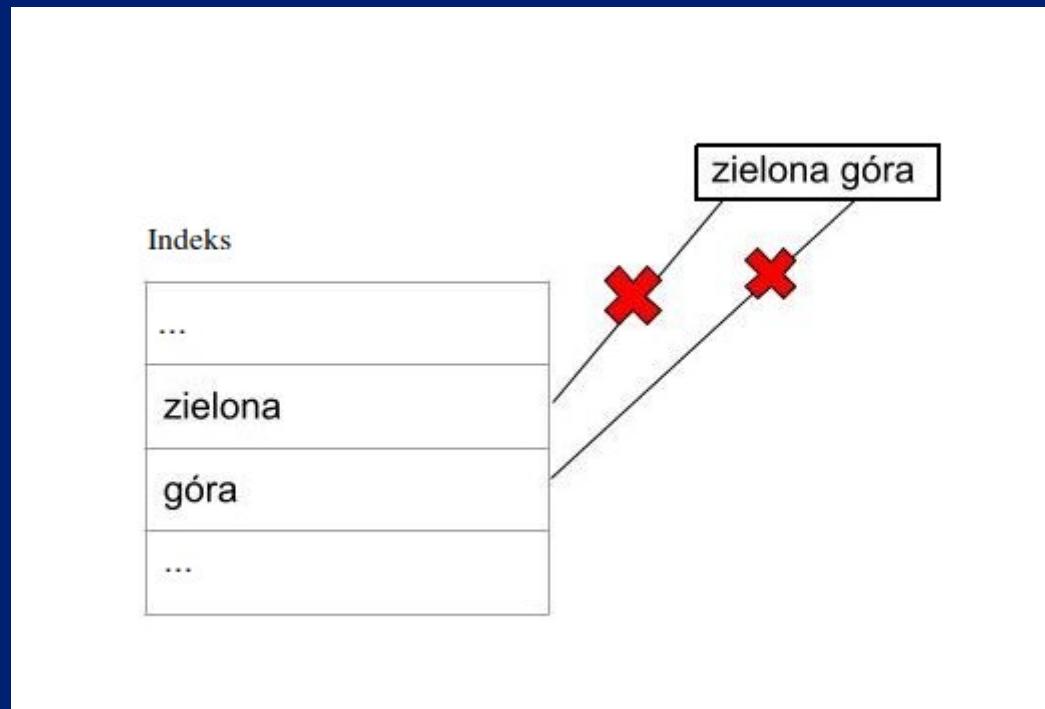
ZAPYTANIA TERM - KOLEJNY PRZYKŁAD

```
curl "http://localhost:9200/airports_http/_search" \
-H "Content-Type: application/json" \
-d '{
  "query": {
    "term": {
      "city": "zielona góra"
    }
  }
}'
```

```
{  
  "hits": {  
    "hits": []  
  }  
}
```

JAK DZIAŁA ZAPYTANIE TERM

Zapytanie typu **term** szuka całej frazy wśród tokenów w indeksie



ZAPYTANIA TERM - KIEDY STOSOWAĆ

Dokumentacja Elasticsearcha odradza stosowania zapytania term dla pól typu text

Użycie zapytania term można rozważyć przy odpytywaniu pól typu keyword

WYSZUKIWANIE PEŁNOTEKSTOWE - MATCH

```
curl "http://localhost:9200/airports_http/_search" \
-H "Content-Type: application/json" \
-d '{
  "query": {
    "match": {
      "city": "katowice"
    }
  }
}'
```

```
$searchParams = [
  'index' => 'airports_http',
  'body'  => [
    'query' => [
      'match' => [
        'city' => 'katowice'
      ]
    ]
  ]
];

$response = $client->search($searchParams);
```

WYSZUKIWANIE PO WIELU POLACH - MULTI MATCH

```
curl "http://localhost:9200/airports_http/_search" \
-H "Content-Type: application/json" \
-d '{
  "query": {
    "multi_match": {
      "fields": ["city", "name", "country"],
      "query": "polska katowice"
    }
  }
}'
```

```
$searchParams = [
  'index' => 'airports_php',
  'body'  => [
    'query' => [
      'multi_match' => [
        'fields' => ['city', 'country', 'name'],
        'query' => 'polska katowice'
      ]
    ]
  ]
];

$response = $client->search($searchParams);
```

WYSZUKIWANIE PO POCZĄTKU FRAZY - MATCH PHRASE PREFIX

```
curl "http://localhost:9200/airports_http/_search" \
-H "Content-Type: application/json" \
-d '{
  "query": {
    "match_phrase_prefix": {
      "city": "kat"
    }
  }
}'
```

```
$searchParams = [
  'index' => 'airports_php',
  'body'  => [
    'query' => [
      'match_phrase_prefix' => [
        'city' => 'kat'
      ]
    ]
  ]
];

$response = $client->search($searchParams);
```

LITERÓWKI

```
curl "http://localhost:9200/airports_http/_search" \
-H "Content-Type: application/json" \
-d '{
  "query": {
    "match": {
      "city": {
        "query": "krakuw",
        "fuzziness": 1
      }
    }
  }
}'
```

ZAPYTANIA ZŁOŻONE

```
curl "http://localhost:9200/airports_http/_search" \
-H "Content-Type: application/json" \
-d '{
  "query": {
    "bool": {
      "should": [
        {
          "match_phrase_prefix": {
            "name": "krakuw"
          }
        },
        {
          "multi_match": {
            "fields": ["name", "city", "country"],
            "query": "krakuw",
            "fuzziness": 1
          }
        }
      ]
    }
  }
}'
```

POMÓŻMY JUNIOROWI ROZWIAZAĆ PROBLEM

```
$searchParams = [
    'index' => 'airports_php',
    'body' => [
        'query' => [
            'bool' => [
                'should' => [
                    ['term' => ['code' => strtoupper($searchPhrase)]],
                    ['match_phrase_prefix' => ['name' => $searchPhrase]],
                    ['match_phrase_prefix' => ['city' => $searchPhrase]],
                    ['match_phrase_prefix' => ['country' => $searchPhrase]],
                    [
                        'multi_match' => [
                            'fields' => ['name', 'city', 'country'],
                            'query' => $searchPhrase,
                            'fuzziness' => 1
                        ]
                    ]
                ]
            ]
        ]
    ]
];
```

PRZYKŁAD Z PRODUKCJI

Elasticsearch w eSky.pl

- Lotniska: ~280 tysięcy
- Miasta: ~5,7 miliona
- Regiony: ~60 tysięcy
- Hotele: ~1,5 miliona

PRZYKŁAD Z PRODUKCJI

Elasticsearch w eSky.pl



NARZĘDZIA UŁATWIAJĄCE PRACĘ Z ELASTICSEARCH

Elasticvue - plugin do Google Chrome / Edge / Firefox

O CZYM NIE POWIEDZIELIŚMY?

- Pozostałe wbudowane analizatory
- Tworzenie własnych analizatorów
- Wyszukiwanie po innych polach niż tekst
(współrzędne geograficzne, zakresy...)
- Ważność wyszukiwania w zależności od pól - boosting
- ...

PYTANIA?



careers.esky.com

SLAJDY I PRZYKŁADY



https://github.com/eskypl/phpers_elasticsearch