

08 - Request body search

Query string bohužel nedokáže pokrýt všechny možnosti vyhledávání, kterými Elasticsearch disponuje. Proto budeme dále queries zapisovat do těla requestu, což se označuje jako [Query DSL](#)

Filtrování dat vs full textové vyhledávání

Obecně je v Elasticsearch rozdíl mezi filtrováním dat a fulltextovým vyhledáváním.

V prvním případě nás zajímá pouze, zda daný výraz odpovídá dotazu, nebo ne. Příkladem takového filtru může být filtrování podle daného číselného rozsahu. Výsledky filtrování je možné efektivně cachovat.

V druhém případě hledáme odpověď na otázku jak moc daný dokument odpovídá vyhledávacímu dotazu.

Abychom rozlišily oba způsoby, budeme používat rozdílné mapování, ale také rozdílné vyhledávací dotazy.

Term-level queries

Nejčastěji používané term-level queries jsou:

- `term` and `terms` — filtrování podle přesné hodnoty:

```
// manufacturer: Microlutions
GET kibana_sample_data_ecommerce/_search
{
  "query": {
    "term": {
      "manufacturer": "Microlutions"
    }
  }
}
```

- `range` — lze využít pro filtrování číselných hodnot a datumu dle rozsahu:

```
// taxful_total_price >= 100 and taxful_total_price < 200
GET kibana_sample_data_ecommerce/_search
{
  "query": {
    "range": {
      "taxful_total_price": {
        "gte": 100,
        "lt": 200
      }
    }
  }
}
```

- `exists` — filtrování dokumentů dle existence daného pole:

```
// customer_phone: *
GET kibana_sample_data_ecommerce/_search
{
  "query": {
    "exists": {
      "field": "customer_phone"
    }
  }
}
```

- `prefix`, `wildcard`, `regexp`, `fuzzy`: vyhledávání s částečnou shodou

Kompletní seznam term-level queries naleznete v [dokumentaci](#).

Compound queries

Compound queries samy o sobě neumožňují vyhledávat nebo filtrovat data, ale umožňují nám spojit více samostatných queries do jedné velké query.

- `bool` — sestává z tří volitelných částí: `must`, `must_not`, `should` (v kombinaci s `minimum_should_max`):

```
GET kibana_sample_data_ecommerce/_search
{
  "query": {
    "bool": {
      "must": [
        { "term": { "manufacturer": "Microlutions" } }
      ],
      "must_not": [
        { "term": { "manufacturer": "Tigress Enterprises" } }
      ],

```

```

    "should": [
      { "term": { "day_of_week": "Saturday" } },
      { "term": { "day_of_week": "Sunday" } }
    ],
    "minimum_should_match" : 1
  }
}

```

- `constant_score` — převede full textový dotaz na filtr
- `function` a `boosting` — umožňují manipulovat se pořadím výsledků

Úkol: Term-level a compound queries

Použijte index `reviews` vytvořený v předchozí kapitole.

1. Vyhledejte dokumenty obsahující e-mailovou adresu (v poli `email`) `emanuel.s@yahoo.com`
2. Vyhledejte dokumenty s hodnocením (`rating`) větším než `2`
3. Vyhledejte dokumenty, kde `email` začíná `tom`
4. Vyhledejte dokumenty obsahující `8 Ball Pool` nebo `Yes Chef!`
5. Vyhledejte dokumenty, kde je `customer_id` vyšší než `13000000`, nebo je `date` novější než `2022-02-20`

Full text queries

Full textové dotazy jsou zpravidla používány v kombinaci s datovým typem `text`. Jejich chování je ovlivněno nastavením textové analýzy, která může být nastavena pro každé pole jinak.

- `match` — základní query pro full textové vyhledávání

```

GET kibana_sample_data_ecommerce/_search
{
  "query": {
    "match": {
      "category": "Shoe"
    }
  }
}

// Query podporuje následující parametry
GET kibana_sample_data_ecommerce/_search
{
  "query": {
    "match": {
      "category": {

```

```

    "query": "Shoe Men",
    "operator": "and",      // <= Shoe AND Men (výchozí je OR)
    "fuzziness": 1         // <= Povolení překlepů
  }
}
}

```

- `multi_match` — rozšiřuje `match` query pro vyhledávání ve více polích

```

GET kibana_sample_data_ecommerce/_search
{
  "query": {
    "multi_match": {
      "query": "Shoe",
      "fields": [
        "products.product_name^3", // <= product name is 3 times more important
        "category"
      ],
      "type": "best_fields"         // <= modifies score calculation
    }
  }
}

```

Výchozí hodnota pro `type` je `best_fields`, která použije pouze nejvyšší skóre z všech dílčích vyhledávání. Další možností je `most_fields`, které kombinuje jednotlivá dílčí skóre.

- `match_phrase` — rozšiřuje `match` query na frázové vyhledávání
- `match_phrase_prefix` — rozšiřuje předchozí query o prefixové vyhledávání v posledním slově
- `query_string` — umožňuje zapsat Lucene query v rámci JSON requestu; dále je k dispozici `simple_query_string`, který je připraven pro zveřejnění koncovým uživatelům
- ...a spousta dalších, viz [dokumentace](#)

Request body search v Kibaně

Query v JSON formátu lze využít i v Kibaně pro filtrování dat. V Discover klikněte na `+ Add filter` a poté na `Edit as Query DSL`. Zde můžete vložit libovolnou query (která by jinak byla pod klíčem `query` v GET requestu).

Discover

Search

+ Add filter

Edit filter

Field: Select a field first

Operator: Waiting

Cancel Save

taxful_total_price category

\$74	Women's Accessories, Women's Shoes
\$105	Women's Accessories, Men's Shoes
\$57.07	Women's Clothing

Discover

Search

+ Add filter

Edit filter

Elasticsearch Query DSL

```

1 {
2   "query": {
3     "match": {
4       "category": "Shoes"
5     }
6   }
7 }

```

Cancel Save

Úkol: Full text search

Znovu vyhledejte v indexu `reviews`:

1. Vyhledejte všechny dokumenty vytvořené s e-mailem z domény `gmail.com` (použijte `match` query)
2. Vyhledejte full textově `fun ball game` v polích `title` a `comments`
3. Vyhledejte `ball game` v poli `title`; zároveň také vyfiltrujte na dokumenty s `rating` vyšším než `3`