

Vyhľadávanie informácií (Information Retrieval)

Zadanie I.

Richard Mocák

xmocak

(21. OKT. 2018)

Dokumentácia

Doména zadania

Zo začiatkov som rozmýšľal dolovať informácie o hudobných koncertoch, čo sa po úvodných konzultáciach zmenilo na vyhľadávanie informácií o hudobných skupinách, konkrétne informácie o hudbníkoch a ich diskografia.

Pre vyhľadávanie som sa zamerlal na 3 stránky, na ktorých sa nachádzali dobre členené informácie o albumoch, poskytovali pokročilejšie rozhranie pre hlbšie prehľadávanie dát a nachádzali sa tam aj informácie o autoroch. Nakoniec som tieto informácie získal zo stránky [discogs.com](http://www.discogs.com) (www.discogs.com), ktorá poskytovala najlepšie členené dáta a umožňovala pokročilé prehľadávanie albumov podľa potrebných atribútov (žáner, štýl, rok, ...).

1. Dolovanie a čistenie dát

Dolovanie dát

V prvom kroku som implementoval webový prehľadávač (z angl. web crawler) prostredníctvom knižníc *Selenium* (simulačný webový prehliadač - Google Chrome) a *BeautifulSoup* (parsovanie HTML obsahu a výber informácií) v programovacom jazyku *Python*.

Script k dolovaniu dát sa nachádza v priečinku *web-crawler* a súbore **web-crawler.py**.

Dolovanie dát prebiehalo prehľadávaním do hĺbky. Prehľadávač som spustil na [stránke prehľadávania albumov](https://www.discogs.com/search/?sort=want%2Cdesc&style_exact=Pop+Rock) (https://www.discogs.com/search/?sort=want%2Cdesc&style_exact=Pop+Rock), kde sú albumy zoradené podľa žiadanosti (atribút Most Wanted) a je ich možné filtrovať podľa žánru. Prehľadávač postupne navštevoval stránku albumu, následne stránku autora ak už danú stránku autora neprehľadával skôr. Po prehľadaní všetkých albumov a autorov, prehľadávač prešiel na ďalšiu stránku a pokračoval v dolovaní údajov. Keďže dolovanie prebiehalo prostredníctvom simulačného webového prehľadávača, dolovanie bolo dosť pomalé a vykonávalo sa iba na jednom vlákne a nie paralelne.

```
# Page Level
for page in range(batch_size):

    # Get album ids ...
    albums_ids = []

    # Album LEVEL
    for element_id in albums_ids:

        album.click()
        # Parse album data
        window.location.goBack()

        artist.click()
        # Parse artist data
        artist.location.goBack()

    # Go to next page
    next_page_button.click()
```

Ukážka dát

Pre každý žáner som spúšťal prehľadávač samostatne, keďže dáta na stránke neboli vždy konzistentne štruktúrované a program bol často nežiaduce zastavený pre chybu. Dáta som ukladal do JSON súborov a ukážka jedného záznamu albumu vyzerá takto:

```
// ALBUM
{
  "name": "Jagged Little Pill",
  "author": "/artist/102789-Alanis-Morissette",
  "genres": ["Rock", "Pop"],
  "styles": ["Alternative Rock", "Acoustic", "Pop Rock"],
  "year": "1995",
  "image_url": "https://img.discogs.com/hajkhakjjkha.jpg",
  "stats_have": "15629",
  "stats_want": "4503",
  "stats_rating": "4.18",
  "stats_ratings_count": "1785",
  "songs": [
    {
      "title": "All I Really Want",
      "time": "4:45"
    },
    ...
  ],
  "comments": [
    {
      "name": "Dexter_prog",
      "date": "August 9, 2018",
      "text": "Great pressing with pristine sound ... Long text"
    },
    ...
  ]
}

// ARTIST
{
  "name": "Coldplay",
  "realName": "Alanis Nadine Morissette",
  "profile": "Canadian singer born on June 1, 1974 in Ottawa, Ontario, ... Long text",
  "sites": ["alanis.com", "MySpace", "YouTube", "Twitter", "Facebook", "Wikipedia"],
  "members": ["Cherie Currie", "Jackie Fox", "Joan Jett"],
}
```

Čistenie dát

Keďže dolované dáta sa nachádzali osobitne pre každý žáner a tiež oddelene údaje o albumoch a oddelene o umelcoch, potreboval som dáta zjednotiť, vyčistiť od nadbytočných bielych znakov, previesť na správny formát (najmä čísla), pridať dátum úpravy a pripraviť na import do elasticsearch (prostr. rozhrania bulk).

Script čistenia dát sa nachádza v priečinku *web-crawler* a názov súboru je **clean-data.py**.

2. Vytvorenie indexov a mapovanie v Elasticsearch

Elasticsearch a nástroj Kibana (obe verzie 6.4.2) som spustil lokálne prostredníctvom platformy **docker**. Elasticsearch bežal na adrese *localhost:9200*.

Vytvorenie indexu, mapovanie, analyzátory

Vytvoril som index **albums**, ktorý obsahoval dokumenty typu **album**.

Pri vytváraní indexu som vytvoril 2 vlastné analyzátory, ktoré pozostávali kombinácie existujúcich tokenizátorov a filtrov.

1. **full_text_analyzer** - Tento analyzátor najmä využívam ako analyzátor pri vyhľadávaní a nie pri procese indexovania. využíva kombináciu štandardného filtra, prevodu na malé písmená a prevodu znakov do ASCII ekvivalentu.
2. **partial_text_analyzer** - Tento analyzátor využívam najmä pri procese indexovanie, keďže okrem rovnakých procesov ako predchádzajúci analyzátor vytvára indexy aj z prefixov ('hell' => 'h', 'he', 'hel', 'hell'), čo umožňuje rýchlejšie vyhľadávanie prefixov, ktoré využijem pri vyhľadávaní dokumentov pri komplexnejších scénároch.

```
{
  "settings": {
    "number_of_shards": 1,
    "number_of_replicas": 0,
    "analysis": {
      "analyzer": {
        "full_text_analyzer": {
          "type": "custom",
          "tokenizer": "standard",
          "filter": ["standard", "lowercase", "asciifolding"]
        },
        "partial_text_analyzer": {
          "type": "custom",
          "tokenizer": "standard",
          "filter": ["standard", "lowercase", "asciifolding", "my_edge_ngram"]
        }
      },
      "filter": {
        "my_edge_ngram": {
          "type": "edgeNGram",
          "min_gram": 1,
          "max_gram": 15,
          "side": "front"
        }
      }
    }
  }
}
```

Mapovanie údajov z ukážky dát na dokument typu **album** mapuje atribúty rôznych typov. Atribúty ako *name*, *songs.title*, *comments.name*, *comments.text* mapujem na typ **text**, aby som využil analyzátory pre process indexovania a vyhľadávania ako som opísal vyššie. Atribúty ako *genres*, *styles* mapujem na typ **keyword** aby som mohol v scenároch využívať možnosť agregácie albumov do vedier práve podľa týchto kľúčových slov.

Zaujímavý je atribút **artist_name**, ktorý je namapovaný ako kombinácia typov *text* pre využitie vlastných analyzátorov ale aj mapovanie na **artist_name.keyword** typu *keyword* pre agregovanie.

V mapovaní sa ešte nachádzajú atribúty numerického typu (*stats_have*, *stats_rating*, ...) a atribúty typu *date* (*created*, *songs.time*, *comments.date*).

```
{
  "mappings": {
    "album": {
      "properties": {
        "name": {
          "type": "text",
          "analyzer": "partial_text_analyzer",
          "search_analyzer": "full_text_analyzer"
        },
        "author": { "type": "keyword" },
        "genres": { "type": "keyword" },
        "styles": { "type": "keyword" },
        "year": {
          "type": "integer"
        },
        "imageUrl": { "type": "keyword" },
        "stats_have": { "type": "integer" },
        "stats_want": { "type": "integer" },
        "stats_rating": { "type": "double" },
        "stats_ratings_count": { "type": "integer" },
        "songs": {
          "type": "nested",
          "properties": {
            "title": {
              "type": "text",
              "analyzer": "partial_text_analyzer",
              "search_analyzer": "full_text_analyzer"
            },
            "time": {
```

```

        "type": "date",
        "format": "m:ss|(m:ss)"
      }
    },
    "comments": {
      "type": "nested",
      "properties": {
        "name": {
          "type": "text",
          "analyzer": "partial_text_analyzer",
          "search_analyzer": "full_text_analyzer"
        },
        "date": {
          "type": "date",
          "format": "MMMM d, yyyy"
        },
        "text": {
          "type": "text",
          "analyzer": "partial_text_analyzer",
          "search_analyzer": "full_text_analyzer"
        }
      }
    },
    "created": {
      "type": "date",
      "format": "yyyy-MM-dd"
    },
    "artist_name": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      },
      "analyzer": "partial_text_analyzer",
      "search_analyzer": "full_text_analyzer"
    },
    "artist_key": {"type": "keyword"},
    "artist_real_name": {
      "type": "text",
      "analyzer": "partial_text_analyzer",
      "search_analyzer": "full_text_analyzer"
    },
    "artist_profile": {
      "type": "text",
      "analyzer": "partial_text_analyzer",
      "search_analyzer": "full_text_analyzer"
    },
    "artist_sites": {"type": "keyword"},
    "artist_members": {
      "type": "text",
      "analyzer": "partial_text_analyzer",
      "search_analyzer": "full_text_analyzer"
    }
  }
}

```

Import dát - bulk rozhranie

Pre import dát do Elasticsearch som využil bulk rozhranie, ktorým som odoslal finálny JSON súbor **albums-02.json**, ktorý sa nachádza v priečinku *web-crawler/EXPORT*.

3. Scenáre použitia a webová aplikácia (Angular)

Rozhodol som sa pre možnosť implementovať 3 netriviálne scenáre vyhľadávania dokumentov prostredníctvom Elasticsearch v aplikácii s používateľským rozhraním pre modifikáciu týchto scenárov. Na základe mojich zručností a skúseností som sa rozhodol pre webovú aplikáciu a aplikačný rámec **Angular 6**. Používateľské prostredie som vytvoril aj s použitím knižnice **@angular/material**, ktorá poskytuje už hotové komponenty. Pre komunikáciu s Elasticsearch som využil oficiálnu Javascriptovú knižnicu **elasticsearch.js**. Zdrojové súbory sa nachádzajú v priečinku *web-app*.


Teraz sa budem trochu detailnejšie venovať každému z 3 scenárov.

Vyhľadávanie albumov podľa mena albumu, mena autora a názvu piesne

Mjusik Search Album, Artist, Song title

AlbumsFacetsArtists

Albums | Results:Items per page: 2061 - 80 of 11039<>



Wounded Rhymes
Lykke Li

Songs


1. Youth Knows No Pain	3:01
2. I Follow Rivers	3:42
3. Love Out Of Lust	4:44
4. Unrequited Love	3:11
5. Get Some	3:23
6. Rich Kids Blues	3:03
7. Sadness Is A Blessing	4:01
8. I Know Places	6:02
9. Jerome	4:20
10. Silent My Song	5:26

Genres:

ElectronicRock

Styles:

Pop RockIndie Rock



The Lost Boys (Original Motion Picture Soundtrack)
Unknown artist

Songs

1. Good Times	3:52
2. Lost In The Shadows (Lost Boys)	6:36
3. Don't Let The Sun Go Down On Me	6:11
4. Laying Down The Law	4:27
5. People Are Strange	3:39
6. Cry Little Sister (Theme From The Lost Boys)	4:47
7. Power Play	3:59
8. I Still Believe	4:52
9. Beauty Has Her Way	3:58
10. To The Shock Of Miss Louise	1:24

Genres:

ElectronicRock

Stage & Screen

Styles:

Alternative RockSoundtrackPop RockSynth-pop



Ocean Avenue
Yellowcard

Songs

1. Way Away	3:22
2. Breathing	3:37
3. Ocean Avenue	3:18
4. Empty Apartment	3:36
5. Life Of A Salesman	3:18
6. Only One	4:17
7. Miles Apart	3:32
8. Twenty Three	3:27

Genres:

Rock

Styles:

Pop RockPop Punk



Play Deep
The Outfield

Songs

1. Say It Isn't So	3:49
2. Your Love	3:44
3. I Don't Need Her	3:56
4. Everytime You Cry	4:29
5. 61 Seconds	4:14
6. Mystery Man	4:06
7. All The Love	3:33
8. Talk To Me	3:36

Genres:

Rock

Styles:

Pop Rock

Component SRC: *src/app/pages/album-search*

Používateľ vie vyhľadávať albumy podľa albumu, umelca alebo titulu pesničky z albumu. Ak používateľ nezadá žiadne vstupný text, vyhľadávajú sa všetky albumy. Vyhľadávanie je stránkované po 20 albumoch na stránku. Vo výsledkoch sa nájde slová farebne vyznačujú (z angl. highlighting). Vo výsledkoch sa zobrazuje obrázok, názov, autor, pesničky, žánre a štýly albumu.

HTTP request takéhoto scenáru vyzerá nasledovne

```
POST http://localhost:9200/albums/album/_search
{
  "from": 0,
  "size": 20,
  "query": {
    "multi_match": {
      "query": "mylo",
      "fields": [
        "name^6",
        "artist_name^3",
        "song.title"
      ]
    }
  },
  "highlight": {
    "fields": {
      "name": {},
      "artist_name": {},
      "songs.title": {}
    }
  }
}
```

Parametre *from* a *size* sa využívajú na stránkovanie.








query je typu *multi_match*, ktorý prehľadáva viacero atribútov dokumentu a skóre môže byť následne posilnené podľa atribútu (**boosted**) ako na príklade kde je výskyt v atribúte *name* 6-násobne relevantnejší ako výskyt v atribúte *song.title*. Keďže sme implementovali analyzátor **partial_text_analyzer**, vráti toto vyhľadvanie aj výskyt v prefixoch slov daných atribútov.

V závere určím, v ktorých atribútoch chcem farebne vyznačovať výskyt prostredníctvom parametra **highlighting**.

Vyhádavanie a filtrovanie albumov podľa agregácií a názvu albumu

Mjusik Search Album Albums Facets Artists

Facets | Results (1429):

	Woman To Woman Joe Cocker	Genres: Rock, Funk / Soul	Styles: Rhythm & Blues, Pop Rock, Funk	Year: 1972
	What A Fool Believes The Doobie Brothers	Genres: Electronic, Rock, Funk / Soul, Pop	Styles: Pop Rock, Disco	Year: 1978
	Release Pet Shop Boys	Genres: Electronic, Rock, Pop	Styles: Pop Rock, Synth- pop	Year: 2002
	Be Yourself Tonight Eurythmics	Genres: Electronic, Rock, Pop	Styles: Synth-pop, Pop Rock	Year: 1985
	Original Soundtrack Zum Film "Christiane F. - Wir Kinder Vom Bahnhof Zoo" David Bowie	Genres: Electronic, Rock, Stage & Screen	Styles: Soundtrack, Pop Rock, Experimental	Year: 1981
	The Singles (The First Ten Years) ABBA	Genres: Electronic, Rock	Styles: Pop Rock, Disco	Year: 1982
	Reaching For The Sky	Genres:	Styles:	Year:

Genres:

- ☒ Rock (675)
- ☐ Electronic (656)
- ☒ Jazz (716)
- ☐ Hip Hop (0)
- ☐ Funk / Soul (631)
- ☒ Folk, World, & Country (279)
- ☐ Pop (226)
- ☐ Latin (69)

Styles:

- ☐ Pop Rock (358)
- ☐ Vinyl (0)
- ☒ Synth-pop (342)
- ☒ Disco (254)
- ☒ Jazz-Funk (502)
- ☒ Funk (321)
- ☒ Experimental (275)
- ☒ House (0)

Decades:

- ☒ * - 1970 (51)
- ☒ 1970 - 1980 (641)
- ☒ 1980 - 1990 (402)

Items per page: 20 1 - 20 of 1429 < >

Component SRC: *src/app/pages/facets*

V druhom scenry umožňuje prostredníctvom vstupného okna vyhľadávať albumy podľa názvu albumu. Používateľ môže filtrovať albumy podľa žánru, štýlova a príslušnej dekády prostredníctvom kategorizovaného formuláru. Ak používateľ nezadal vstup, filtrujú sa všetky dokumenty podľa filtrov z kategorizovaného formuláru. Vyhľadávanie je tiež stránkované.

V tomto scenári sa na elasticsearch dopytojem dvoma HTTP requestami.

1. *Inicializácia agregácií* - prostredníctvom agregácie typu **significant_terms** získam 8 najzaujímavejších hodnôt vyskytujúcich sa v dokumentoch. Túto agregáciu využijem pre atribúty **genres** a **styles**. Formulár dekád, získam prostredníctvom agregácie **range** na atribúte **year**.

```
POST http://localhost:9200/albums/album/_search
{
  "aggregations": {
    "genres_terms": {
      "significant_terms": {
        "field": "genres",
        "size": 8
      }
    },
    "styles_terms": {
      "significant_terms": {
        "field": "styles",
        "size": 8
      }
    },
    "decade_ranges": {
      "range": {
        "field": "year",
        "ranges": [
          { "to": 1970 },
          { "from": 1970, "to": 1980 },
          { "from": 1980, "to": 1990 },
          { "from": 1990, "to": 2000 },
          { "from": 2000, "to": 2010 },
          { "from": 2010 }
        ]
      }
    }
  }
}
```

2. *Vyhľadávanie a filtrovanie albumov* - V tomto requeste sa tiež využívajú parametre *from* a *size* na stránkovanie výsledkov. **query** je typu **bool**, ktorý vyhľadáva indexy atribútu name (podobne ako minule, záhrňa to aj prefixy). Výsledky sú následne filtrované podľa zaškrtnutých prvkov agregácií žanrov, štýlov a dekád. Využitie sú na to filtre **terms** a **range**. V tele requestu sa nachádzajú agregácie ako pri incilaizácii aby sme vedeli aktualizovať počet dokumentov na agregáciu v používateľskom rozhraní.

```
POST http://localhost:9200/albums/album/_search
{
  "from": 0,
  "size": 20,
  "query": {
    "bool": {
      "must": [{"match": { "name": "aa"}}],
      "filter": [
        { "terms": { "genres": ["Rock", "Hip Hop"] }},
        { "terms": { "styles": ["Pop Rock", "Jazz-Funk"] }},
        {
          "bool": {
            "should": [
              { "range": { "year": { "gte": 1990, "lte": 2000 }}},
              { "range": { "year": { "gte": 2010 }}}
            ]
          }
        }
      ]
    }
  },
  "aggregations": {
    // ... SAME AS INITIALIZATION (STEP 1.)
  }
}
```

Vyhľadavanie, filtrovanie a triedenie albumov podľa autora

Mjusik Search Album

Albums Facets Artists

Select artist ...

Coldplay

Bob


Bob Dylan

Bob James

Bobbi Humphrey


Bobby Hutcherson

Bobby Beausoleil




A Rush Of Blood To The Head
Coldplay
Genres: Parlophone, Parlophone

Rating: 4.32
Want: 1368




Viva La Vida Or Death And All His Friends
Coldplay
Genres: Parlophone, Parlophone

Rating: 4.22
Want: 840




Viva La Vida Or Death And All His Friends
Coldplay
Genres: Capitol Records

Rating: 4.22
Want: 629




Talk
Coldplay
Genres: Rock

Rating: 4.21
Want: 1087



Clocks
Coldplay
Genres: Rock

Rating: 4.17
Want:



Mylo Xyloto
Coldplay
Genres: Parlophone, Parlophone

Rating: 4.07
Want:

Component SRC: `src/app/pages/artist-search`

V tomto scenári môže používateľ vyhľadávať albumy podľa názvu albumu, ktorý zadá do vstupného okna. Môže filtrovať albumy podľa autora, ktorého si vyberie zo Autocomplete Select Boxu, ktorý filtruje mená autorov podľa zadaného vstupu. Používateľ môže stránkovať výsledky. Používateľ môže prepínať medzi usporiadaním výsledkov podľa **relevantnosti vyhľadávania** (`_score`) alebo podľa hodnotenia a žiadanosť albumu zostupne (atribúty `stats%rating`, `stats_want`).

V tomto scenári sa na elasticsearch zasielali 2 typy requestov:

1. **Autocomplete autorov** - Tento request vyhľadával také dokumenty, kde atribút **artist_name** (full text) prefix slov obsahoval hľadaný výraz. Zároveň ale odfiltroval preč všetky dokumenty, ktorých atribút **artist_name.keyword** (keyword) neobsahuje už autora, ktorý bol vybratý v autocomplete. Zároveň definujeme aby nevracal žiaden výsledky (**size** je rovné nula), ale iba **agregácie** 10 autorov (atribút `artist_name.keyword`) s najväčším počtom vyhľadaných dokumentov. Z týchto agregácií sa vytvorí zoznam možností v autocomplete select boxe.

```
POST http://localhost:9200/albums/album/_search
{
  "size": 0,
  "query": {
    "bool": {
      "must": [{ "match": { "artist_name": "Bob" } }],
      "filter": {
        "bool": {
          "must_not": { "terms": { "artist_name.keyword": [ "Coldplay" ] } }
        }
      }
    }
  },
  "aggregations": {
    "artists": {
      "terms": { "field": "artist_name.keyword", "size": 10 }
    }
  }
}
```


2. *Filtrovanie, vyhľadavanie a zoradenie albumov* - V tomto requeste prebieha rovnako ako pri predchádzajúcich scenároch stránkovanie prostredníctvom atribútov *from* a *size*. Ak používateľ zvolí možnosť **zoradiť** podľa hodnotenia a žiadanosti, tak sa pridá parameter *sort* pre atribúty *stats_rating* a *stats_want* zoradené zostupne. V *query* sa vyhľadávajú albumy, ktorých atribút *name* obsahuje hľadaný výraz (zase aj prefixy, ako pri predch. scenároch). Výsledky sa filtrujú podľa zvolených autorov zo autocomplete vďaka atribútu *artist_name.keyword*. Ak používateľ nevybral, žiadneho autora, parameter *filter*, sa v requeste nenachádza.

```
POST http://localhost:9200/albums/album/_search
{
  "from": 0,
  "size": 20,
  "sort": [
    { "stats_rating": { "order": "desc" } },
    { "stats_wants": { "order": "desc" } }
  ],
  "query": {
    "bool": { "must": [{ "match": { "name": "The" } } ],
      "filter": {
        "bool": {
          "must": { "terms": { "artist_name.keyword": [ "Coldplay", "Bob Dylan" ] } }
        }
      }
    }
  }
}
```

Zhodnotenie

Tento projekt bol pre mňa veľmi prínosný. Príprava crawlera a dolovanie dát ma veľmi zaujali a určite som rad, že som v tejto oblasti nabral nové znalosti. Najviac ma oslovila práca s Elasticsearch a najmä rýchla odozva vyhľadávania. Okrem toho ma silno zaujala jednoduchosť písania HTTP requestov do elasticsearchu a možnosti indexovania atribútov pre využitie v rôznych scenároch. Verím a dúfam, že ešte budem niekedy v budúcnosti môcť pracovať s elasticsearchom. Snažil som sa do tohto projektu skutočne vložiť a spraviť to najlepšie ako viem. Pomohli mi aj konzultácie, keď som niekedy uviazol na mŕtvom bode a nevedel sa posunúť ďalej, alebo som niečo robil zle.