elasticsearch

Примеры использования

Антон Шелыганов, 220 Вольт

О себе:

- Fullstack PHP Разработчик, Опыт в разработке ~6 лет.
- Работаю в компании 220В,
- До этого Yumasoft, Prometey Telecomunications
- ???
- ???

О чем доклад

- Обзор функционала Elastic Search
- Примеры реализации наиболее востребованных функций поиска
- Описание некоторых приемов интеграции с elastic.
- Примеры доступны на github: https://github.com/antonshell/elastic_examples

Что такое Elastic Search?

- Свободная поисковая система, основанная на Lucene, самая популярная в своей категории
- Разрабатывается компанией Elastic вместе со связанными проектами.

  Logstash - системой сбора данных и анализа журналов.

  Kibana - платформой аналитики и визуализации;
  Все вместе - стек ELK.

Внутреннее устройство

- Написан на Java
- Используется собтвенное NoSql хранилище
- Использует библиотеку поиска Lucene

  https://www.quora.com/What-is-an-intuitive-description-of-how-Lucene-works
- Elastic search изнутри:
  https://www.elastic.co/blog/found-elasticsearch-from-the-bottom-up

Преимущества / Недостатки

- Продвинутые возможности поиска
- Расширяемость, плагины и т.д.
- Независимость, отдельный компонент
- Nosql, произвольная структура данных.
- Производительность - время отклика 10 - 100 мс
- Масштабируемость, Кластеризация

Преимущества / Недостатки

+ Использует Lucene
+ Полнотекстовый поиск
+ Документоориетированная БД, произвольная схема данных
+ Встроенное REST API
+ Расширяемость, плагины и т.д.
+ Производительность, масштабируемость, кластеризация
- Только JSON, Достаточно сложные query
- Иногда непредсказуемые результаты
- Не очень удобен именно для хранения данных
- Безопасность по-умолчанию

Альтернативы

- Sphinx
- Solr, Lucene
- Amazon CloudSearch
- Database search, Fulltext etc.

Sphinx

- Полнотестовый поиск. Субъективно сложнее в настройке.
- Написан на C++
- Строит индексы по имеющейся БД
- Высокая скорость индексации
- Менее требователен к ресурсам

Solr, Lucene

- Lucene по сути, основа Elastic Search. Свободная библиотека для высокопроизводительного полнотекстового поиска https://ru.wikipedia.org/wiki/Lucene
- Solr - предшественник. Платформа полнотекстового поиска с открытым исходным кодом. Основные идеи похожи https://ru.wikipedia.org/wiki/Apache_Solr

Amazon CloudSearch

- Управляемый сервис в облаке AWS, позволяющий легко и экономично настраивать и масштабировать поисковые решения для веб-сайтов
- Amazon CloudSearch поддерживает 34 языка и популярные функции поиска, например подсветку совпадений, автозаполнение и геопространственный поиск. https://aws.amazon.com/ru/cloudsearch/

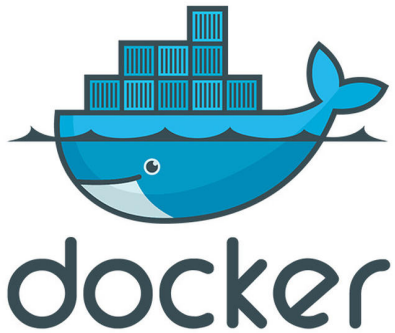
Database search, Fulltext

- Поиск лайками либо полнотестовый поиск на уровне СУБД
- Впринципе можно сделать все, что угодно, ну почти.
- Вопрос производительности.
- И сложности реализации
- Документация https://dev.mysql.com/doc/refman/5.7/en/fulltext-search.html

Установка Elastic Search

- Нативная установка

- В Docker контейнере
  - *docker run -p 9200:9200 -p 9300:9300 -e "discovery.type=single-node" docker.elastic.co/elasticsearch/elasticsearch:6.1.0*
  - *docker run -p 9200:9200 -p 9300:9300 -e "discovery.type=single-node" elasticsearch:5.6.5*



Представление данных

- Индекс - структура хранения данных. Предназначен для организации обработки данных. А также для организации шардинга(разделения на ноды)

  https://www.elastic.co/blog/what-is-an-elasticsearch-index
  https://www.elastic.co/blog/index-vs-type
  https://stackoverflow.com/questions/15025876/what-is-an-index-in-elasticsearch

- Маппинг - описание схемы документов

  https://toster.ru/q/218450
  https://logz.io/blog/elasticsearch-mapping/

- Сравнение: индекс ~ база данных, тип/маппинг ~ таблица, документ ~ запись

Взаимодействие с elastic - Rest API

- Большинство операций можно выполнить через api.
- CRUD данных
- Создание индексов/маппингов
- Добавление синонимов и т.д.
- Диагностика, информация о системе
- Собственно, поиск

Создание индекса / маппинга

- Индекс - общие настройки поиска
- Маппинг - настройки типов данных

```
curl -X PUT \
  http://127.0.0.1:9200/post \
  -H 'Content-Type: application/json' \
  -H 'Postman-Token: 9bb0b988-c4d7-4aa6-b560-10de111e48c7' \
  -H 'cache-control: no-cache' \
  -d '{
  "settings": {
    "index": {
      "similarity": {
        "default": {
          "type": "BM25"
        }
      }
    },
    "analysis": {
      "filter": {
        "my_synonym_filter": {
          "type": "synonym",
          "synonyms": [
            "законный, легальный",
            "автомобиль, машина"
          ]
        },
        "ru_stop": {
          "type": "stop",
          "language": "_russian_"
        },
        "ru_stemmer": {
```

```
curl -X PUT \
  http://127.0.0.1:9200/post/_mapping/type \
  -H 'Content-Type: application/json' \
  -H 'Postman-Token: 77b6340e-53e4-406d-8eb5-8b40beb0b6aa' \
  -H 'cache-control: no-cache' \
  -d '{
  "properties": {
    "id": {
      "type": "integer"
    },
    "name": {
      "type": "text",
      "index": true,
      "search_analyzer": "my_synonyms",
      "analyzer": "my_synonyms",
      "term_vector": "with_positions_offsets_payloads"
    },
    "content": {
      "type": "text",
      "index": true,
      "search_analyzer": "my_synonyms",
      "analyzer": "my_synonyms",
      "term_vector": "with_positions_offsets_payloads"
    },
    "content_suggest": {
      "type": "completion",
      "analyzer": "my_synonyms",
      "search_analyzer": "my_synonyms",
```

Загрузка данных

- Добавление продукта

```
curl -X PUT \
  http://127.0.0.1:9200/post/type/1 \
  -H 'Content-Type: application/json' \
  -H 'Postman-Token: 61c09da4-588f-4a3b-8110-c158bb685e9d' \
  -H 'cache-control: no-cache' \
  -d '{
  "id": 1,
  "name": "Борщевик Сосновского. В МО ввели штрафы за распространение",
  "content": "1 ноября 2018 года Московская Область, без объявления войны (объявленной
      парой лет ранее), ввела финансовые санкции. Против собственников территорий,
      предоставляющих плацдарм для распространения борщевика Сосновского. Ура!Мне,
      правда, интересно, кто будет платить за титаническую плантацию борщевика между
      Шереметьево и ниткой Аэроэкспресса. Плантацию, встречающую гостей и жителей
      Москву сразу по прилёту в белокаменную. Там растут просто миллионы штрафа. Что
      же такое борщевик Сосновского и почему с ним надо бороться. Как это делать. Как
      это делать не нужно. А так же научные и псевдонаучные факты в нескучно
      -популярном изложении."
}'
```

Получение данных

- Получение одной записи

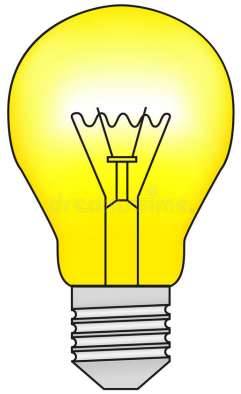
http://127.0.0.1:9200/post/type/1

- Все записи

http://127.0.0.1:9200/post/type/_search

Поиск / Нечеткий поиск

```
curl -X POST \
  http://127.0.0.1:9200/post/type/_search \
  -H 'Content-Type: application/json' \
  -H 'Postman-Token: 2389f71d-f5c8-4b42-bd2b-ee0944a93269' \
  -H 'cache-control: no-cache' \
  -d '{
  "query": {
    "multi_match" : {
      "query": "компидяторы",
      "fuzziness": "AUTO",
      "fields": ["name", "content"]
    }
  },
  "_source": ["id","name","content"]
}'
```



Из интересного

- Запросы к Elastic немного напоминают sql, но их сложнее писать
- Немного напоминает GraphQL, возможно, другие похожие технологии
- Есть эксперементальная поддержка sql для elastic
- Есть инструменты для преобразования sql в elastic query

Фильтры

- Процессор Intel дешевле 400$, с частотой от 3.2 до 3.6
- Аналог sql where
- must, term, match, range, should, bool

```
"query": {
  "bool": {
    "must": [
      {
        "match": {
          "vendor": "Intel"
        }
      },
      {
        "range" : {
          "price" : {
              "lte" : 400
          }
        }
      },
      {
        "bool": {
          "should": [
            {
              "term": {
                "frequency_base": "3.6"
              }
            },
            {
              "term": {
                "frequency_base": "3.2"
              }
            }
```

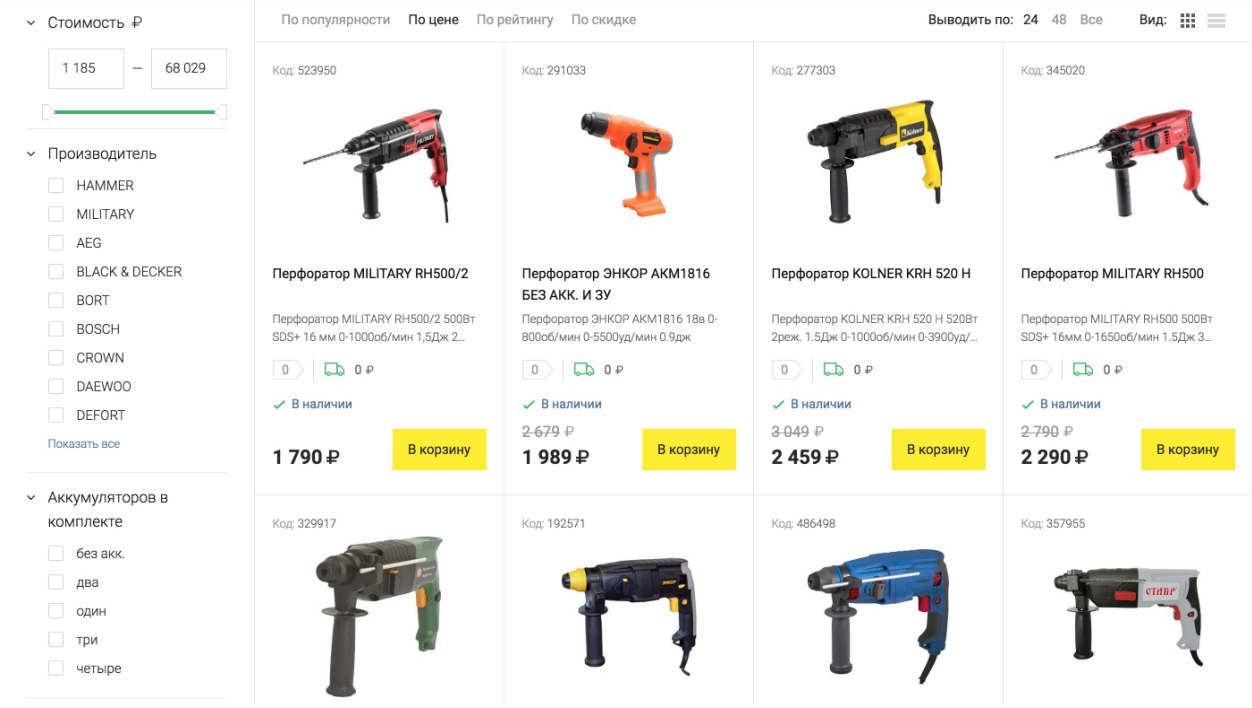
Агрегация - min/max, distinct

```
    "aggregations": {
        "max_price": {
            "value": 929.99
        },
        "min_price": {
            "value": 239.99
        },
        "distinct_cpu": {
            "doc_count_error_upper_bound": 0,
            "sum_other_doc_count": 0,
            "buckets": [
                {
                    "key": "Coffee Lake",
                    "doc_count": 3
                },
                {
                    "key": "Coffee Lake-R",
                    "doc_count": 3
                },
                {
                    "key": "Skylake",
                    "doc_count": 1
                }
            ]
        }
    }
}
```

```
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "vendor": "Intel"
          }
        }
      ]
    }
  },
  "size": 0,
  "aggs": {
    "distinct_cpu": {
      "terms": {
        "field": "cpu.keyword"
      }
    },
    "min_price": {
      "min": {
        "field": "price"
      }
    },
    "max_price": {
      "max": {
        "field": "price"
      }
    }
```

Кейс с фильтрами / агрегацией

- Страница каталога интернет-магазина
- Список товаров - с помощью фильтров
- Значения фильтров - с помощью агрегации



Синонимы

- Равнозначные слова. законный -> легальный, автомобиль -> машина и т.д.
- Задаются при создании индекса

```
curl -X POST \
  http://127.0.0.1:9200/post/type/_search \
  -H 'Content-Type: application/json' \
  -H 'Postman-Token: d3b00837-bcda-424e-bd24-1d77bb2a5320' \
  -H 'cache-control: no-cache' \
  -d '{
  "from" : 0, "size" : 20,
  "query": {
    "multi_match" : {
      "query": "легальный",
      "fields": ["name", "content"]
    }
  }
}'
```


```
"_source": {
    "id": 6,
    "name": "Несертифицированный GPS-трекер из Китая. Законно ли в России?",
    "content": "Иностранные интернет-магазины завалены разнообразными устройствами, оснащёнными
        управлять устройством посредством SMS и мобильных приложений. И, конечно же, большинство
        обыватель, услышав слова «несертифицированный» и «GPS» в одном предложении, скажет: «Это
        попытался разобраться и сконсолидировать в этой статье.",
    "name_suggest": {
        "input": [
            "Несертифицированный GPS-трекер из Китая. Законно ли в России?",
            "GPS-трекер из Китая. Законно ли в России?",
            "из Китая. Законно ли в России?",
            "Китая. Законно ли в России?",
            "Законно ли в России?",
            "ли в России?",
            "в России?",
            "России?"
        ]
    }
}
```

Обновление настроек индекса

- Нужно для обновления синонимов
- Elastic не позволяет редактировать настройки индекса во время работы
- Elastic не позволяет переименовывать индекс
- Можно остановить индекс, обновить настройки, снова запустить индекс.
- Получаем кратковременный простой elastic.

Подсветка поиска

- Подсветка поисковых запросов в результатах поиска

```
curl -X POST \
  http://127.0.0.1:9200/post/type/_search \
  -H 'Content-Type: application/json' \
  -H 'Postman-Token: 14271fef-1aa5-49cd-9f17-f653d04ad000' \
  -H 'cache-control: no-cache' \
  -d '{
    "query" : {
        "match": { "content": "Sony" }
    },
    "highlight" : {
        "fields" : {
            "content" : {}
        }
    },
    "_source": ["id","name","content"]
}'
```

```
"hits": [
    {
        "_index": "post",
        "_type": "type",
        "_id": "17",
        "_score": 1.6480465,
        "_source": {
            "name": "Японская корпорация Sony",
            "id": 17,
            "content": "Японская корпорация Sony представила новый смартфон под назва
        },
        "highlight": {
            "content": [
                "Японская корпорация <em>Sony</em> представила новый смартфон под наз
            ]
        }
    }
]
```

Морфология

- Поиск с учетом русской морфологии. Законными -> Законно

```
curl -X POST \
  http://127.0.0.1:9200/post/type/_search \
  -H 'Content-Type: application/json' \
  -H 'Postman-Token: 1b85d5d6-a72d-4cda-86ff-da5b2374f72e' \
  -H 'cache-control: no-cache' \
  -d '{
  "from" : 0, "size" : 20,
  "query": {
    "multi_match" : {
      "query": "Законными",
      "fields": ["name", "content"]
    }
  }
}'
```


```
"_source": {
    "id": 6,
    "name": "Несертифицированный GPS-трекер из Китая. Законно ли в России?",
    "content": "Иностранные интернет-магазины завалены разнообразными устройствами, оснащё
        управлять устройством посредством SMS и мобильных приложений. И, конечно же, больш
        обыватель, услышав слова «несертифицированный» и «GPS» в одном предложении, скажет
        попытался разобраться и сконсолидировать в этой статье.",
    "name_suggest": {
        "input": [
            "Несертифицированный GPS-трекер из Китая. Законно ли в России?",
            "GPS-трекер из Китая. Законно ли в России?",
            "из Китая. Законно ли в России?",
            "Китая. Законно ли в России?",
            "Законно ли в России?",
            "ли в России?",
            "в России?",
            "России?"
        ]
    }
}
```

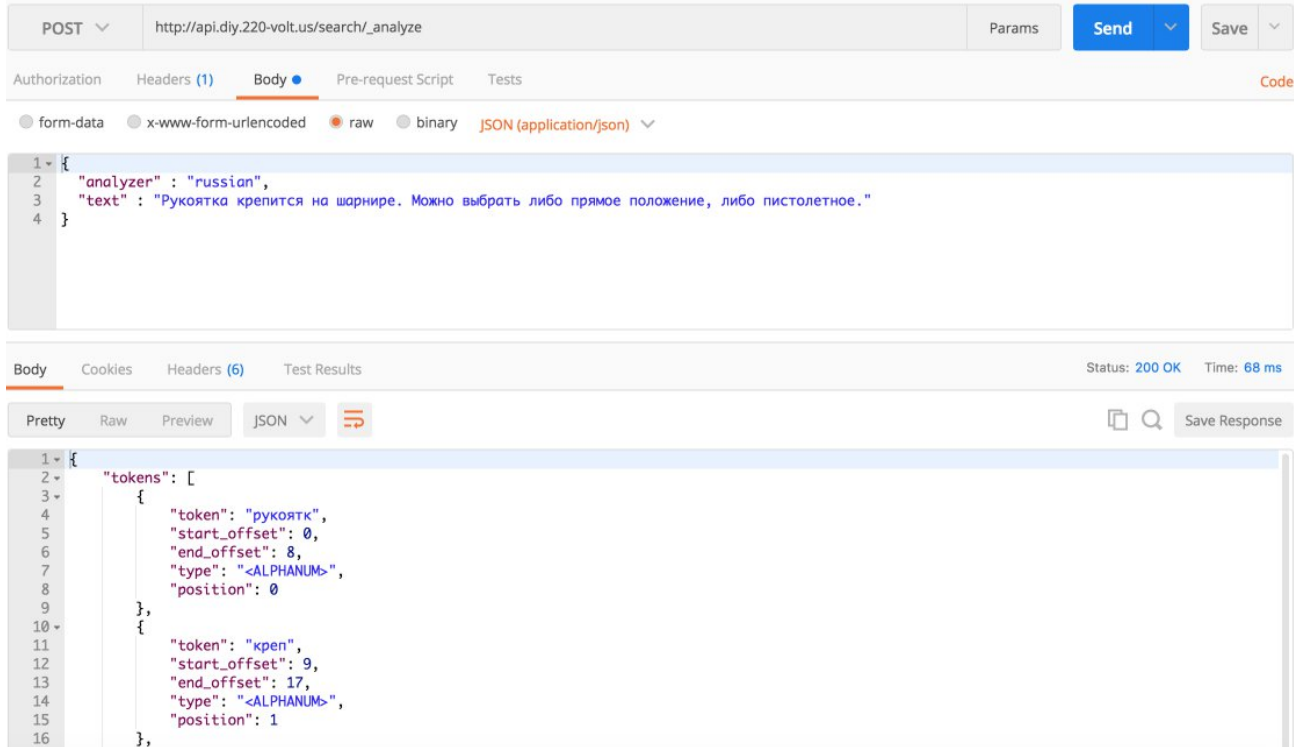
Стоп-слова

- Поиск по предлогам, местоимениям и т.д. не возвращает результатов. Прим: этой, этим, тот, туда, более и т.д.

```
curl -X POST \
  http://127.0.0.1:9200/post/type/_search \
  -H 'Content-Type: application/json' \
  -H 'Postman-Token: d16f7a20-d0bc-4d0b-b16a-94c7749b4d01' \
  -H 'cache-control: no-cache' \
  -d '{
  "from" : 0, "size" : 20,
  "query": {
    "multi_match" : {
      "query": "этой",
      "fields": ["name", "content"]
    }
  }
}'
```

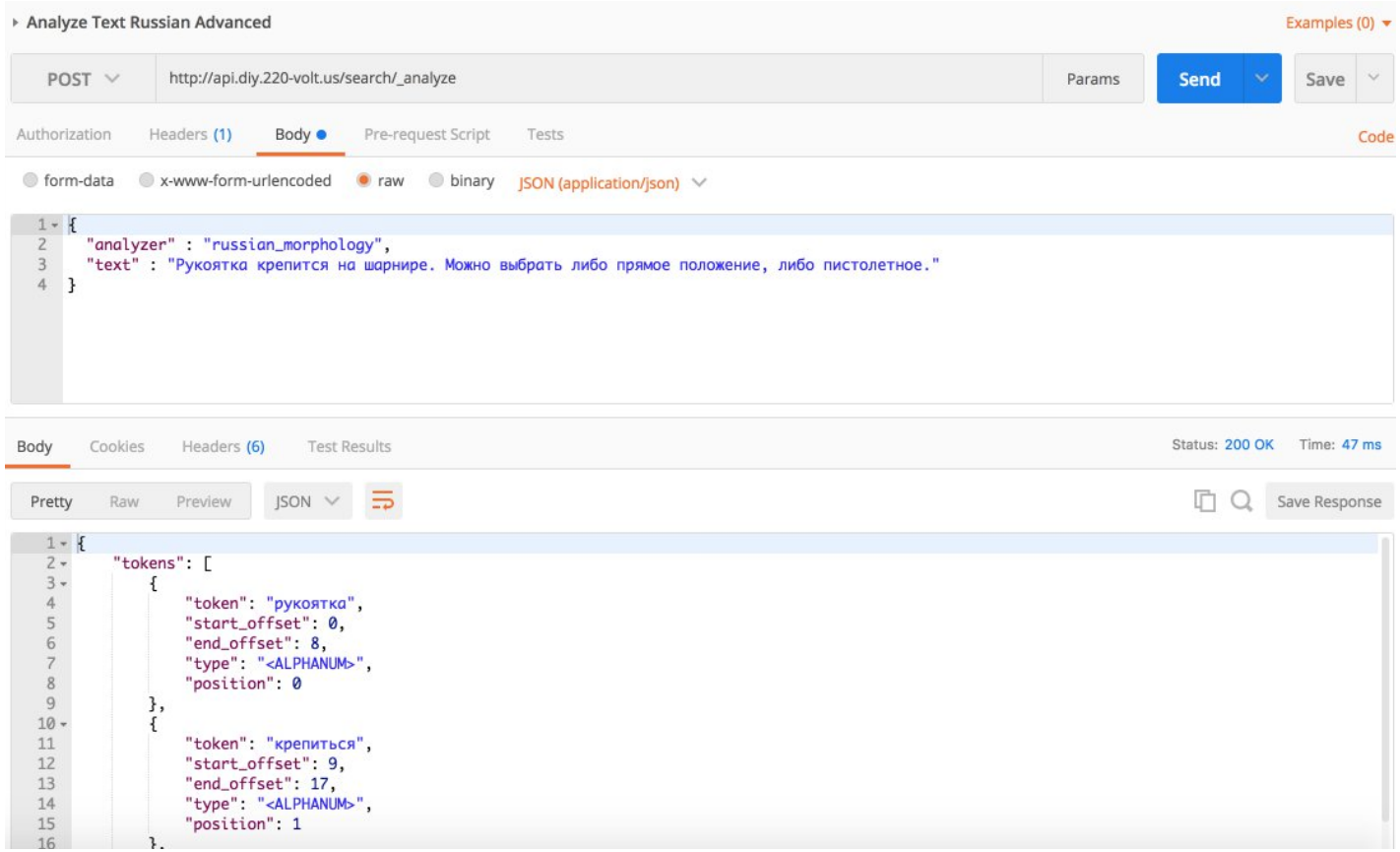
Стандартный анализатор(5.6)

- Работает из коробки
- Реализует базовую морфологию
- Проверка работы:
- Актуально для версии <= 5.6



Плагин с улучшеной морфологией(5.6)

https://github.com/AKuznetsov/russianmorphology
https://github.com/imotov/elasticsearch-analysis-morphology



Поисковые подсказки

- Поиск по фрагментам слов. Прим: авт -> автомобиль
- Работает по отдельному полю name_suggest

```
curl -X POST \
  http://127.0.0.1:9200/post/_suggest \
  -H 'Content-Type: application/json' \
  -H 'Postman-Token: 862e5b7d-5210-4846-82e9-ae80b4a8
  -H 'cache-control: no-cache' \
  -d '{
    "product_suggest":{
        "text":"авт",
        "completion": {
            "field" : "name_suggest"
        }
    }
}'
```


```
        "_source": {
            "id": 14,
            "name": "Просто авто",
            "content": "Просто авто",
            "name_suggest": {
                "input": [
                    "Просто авто",
                    "авто"
                ]
            }
        }
    },
    {
        "text": "автомобили",
        "_index": "post",
        "_type": "type",
        "_id": "15",
        "_score": 1,
        "_source": {
            "id": 15,
            "name": "Японские автомобили",
            "content": "Японские автомобили вновь заняли в США первые места
            "name_suggest": {
                "input": [
                    "Японские автомобили",
                    "автомобили"
```

Поиск по содержимому документов

- Позволяет загружать в elastic документы PDF, DOC(X), XLS(X), PPT(X) и др.
- Извлекает текстовое содержимое, позволяет искать по нему, используя синонимы подсветку запросов и т.д.
- Принимает файлы в формате base64
- Хранит base64 содержимое

Загрузка документа

- Создать pipeline - обработчик документов
- Создать индекс / маппинг
- Преобразовать документ в base64 - функция base64_encode
- Создать запись в elastic, передать строку base64 и другие поля, если нужно
- Удобнее это делать скриптом автоматически

elastic

Интеграция Symfony с Elastic

Symfony

Интеграция

- Загружать все данные.
- Обновлять данные в elastic при изменении
- Использовать очередь
- Хранить настройки индекса / маппинга
- Собирать данные из разных таблиц
- Бесшовное обновление - если нужно

Выгрузка в elastic

- Создать объект индекса/маппинга, содержащий основные настройки
- Определить типы данных для импортируемых полей
- Реализовать Dataprovider - получение данных для выгрузки в elastic
- Консольные команды для выгрузки

Очередь

- Можно сразу отправлять запрос к Elastic, при изменении данных
- Желательно делать это асинхронно, используя очередь
- Можно класть запросы в mysql, потом выбирать оттуда и отправлять.
- Можно использовать очередь на базе Redis.
- Или любую другую очередь
- Классы обработчики, обращаются к Elastic

Синонимы

- Отвертка - Шуруповерт, Стол - Верстак, Углошлифовальная машина - болгарка, и т.д.
- Хранить и редактировать синонимы в базе данных приложениея
- Загружать обновления синнимов в Elastic
- Реализуем метод updateSynonyms
- Индекс на некоторое время закрывается, к кратковременный простой сервиса
- Подробно описано в статье

  http://antonshell.me/post/elastic-search-synonyms

Спасибо за внимание!

- Telegram: https://t.me/antonshell
- Блог: http://antonshell.me/
- Github: https://github.com/antonshell
- Skype: anton.shelyganov.yumasoft

- Примеры из доклада: https://github.com/antonshell/elastic_examples