Добавляем поисковую строку в ваше приложение с Elasticsearch

Маленький google в вашем приложение

Шестаков Алексей

Инженер-Программист

СКБ Контур

Что будет в этом докладе

- Внутренний поиск
- История и Введение в Elasticsearch
- Примеры использования
- Теория
- Как сделать поиск умнее
- Куда двигаться дальше

Чего не будет в этом докладе

- Эксплуатация ES
- Использование ES для хранения логов
- ML (aka Машинное обучение) в поиске.

Встроенный поиск

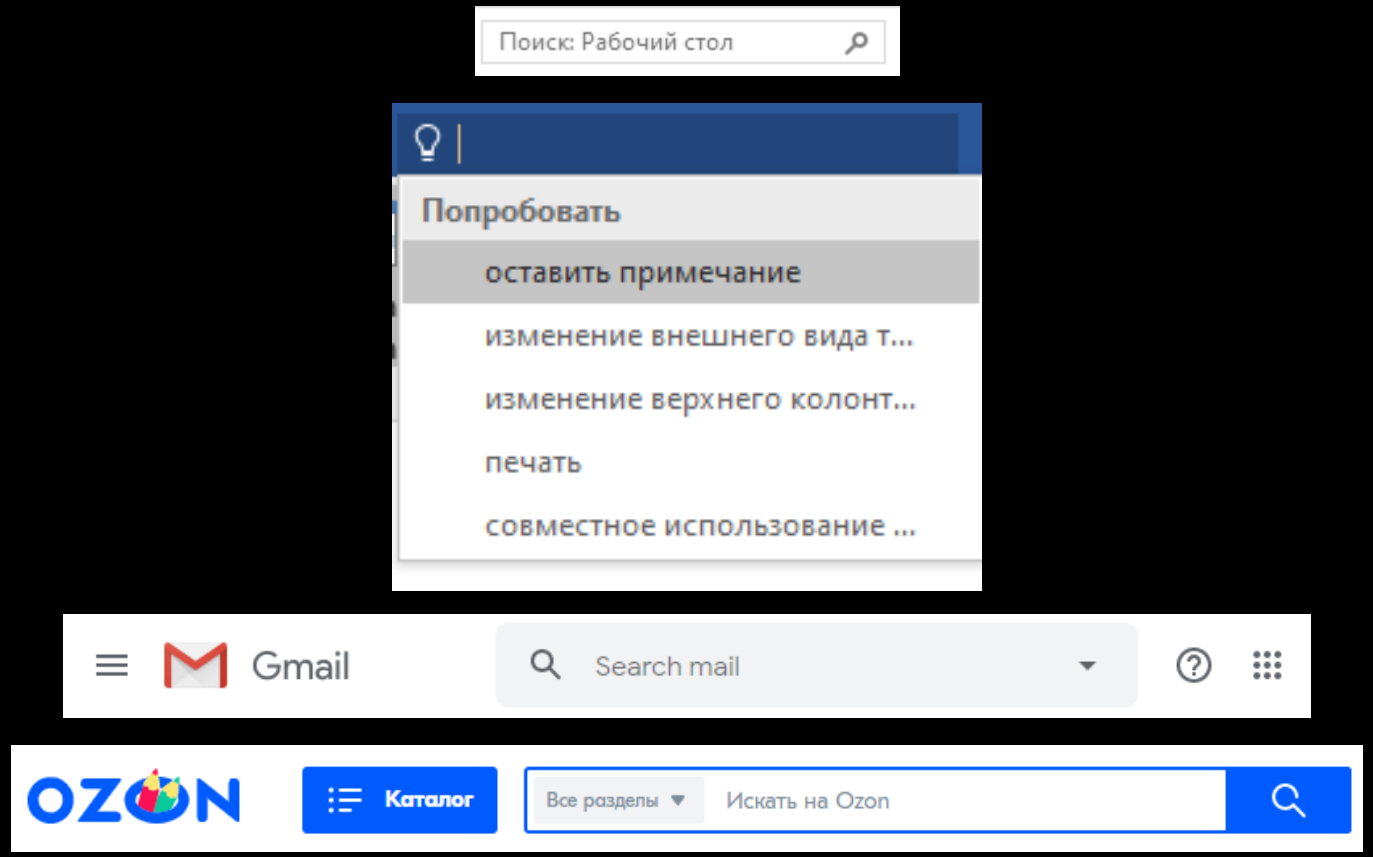
Зачем в вашем приложении встроенный поиск?

Встроенный поиск и почему это круто

- Единая точка навигации
- Быстрая навигация
- Сохраняет массу времени пользователям
- Можно сделать персонифицированным.

Примеры из жизни

- Windows
- MS Office
- GMail
- Ozon
- И ещё много-много примеров....



Нужен ли он вашим пользователям?

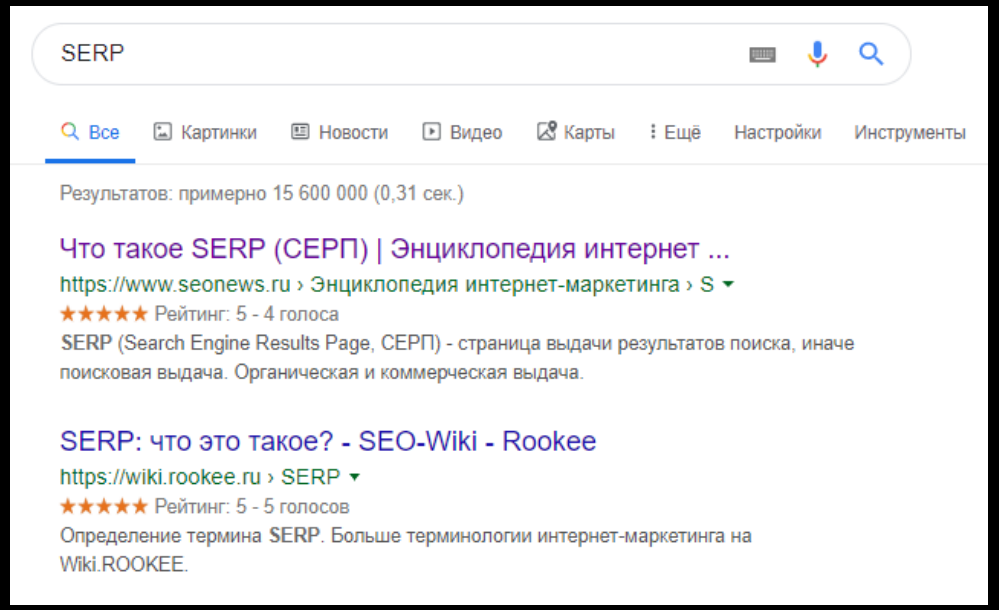
- Много контента
- Много типов контента
- Много фичей
- Сложная навигация меню

Подсказки vs Поисковая страница (SERP)

- Основные виды поиска
  - Поисковая страница aka SERP
  - Поисковые подсказки

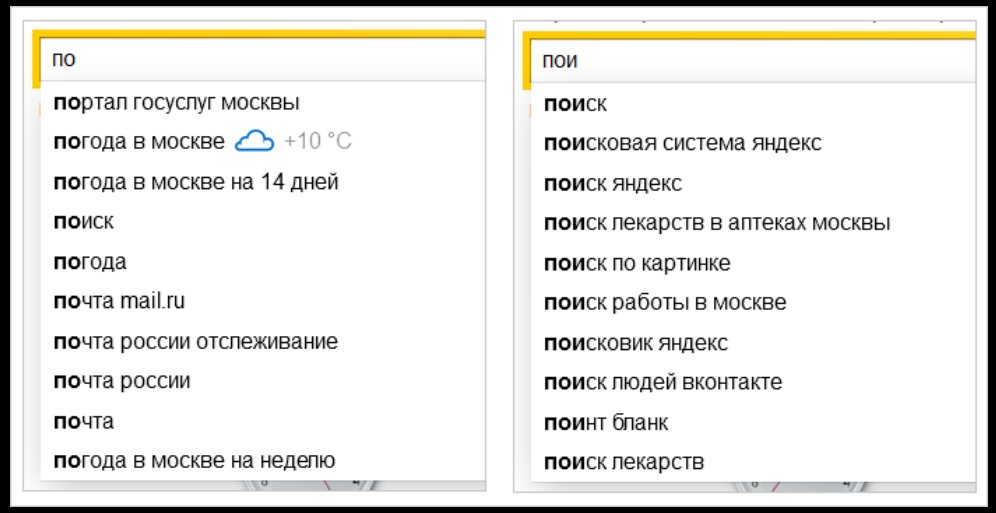
SERP

- Под результаты отдана целая страницы
- Доступны фильтры по типу и другие
- Фрагмент документа – снипет
- Это поиск по **целым словам**



Поисковые подсказки

- Появляется по мере ввода
- Результат ввода – redirect на нужную страницу
- Умный фильтр
- Может отвечать на запрос сразу
- Это **префиксный** поиск



Elastic Search - Введение

Движки полнотекстового поиска

- Специализированное ПО для построения поиска
- Позволяют гибко настраивать ранжирование
- Наборы фильтров и др.

Почему реляционные БД не подходят ?

- Можно возразить что ряд БД имеет инструменты для полнотекстового поиска. (TODO: упростить)
- Примеры MSSQL и PostgreSQL
- Это дополнительная фича
- Не стандартный SQL (TODO: пояснить)

Elasticsearch

- Реплицированный
- Распределенный
- Горизонтально масштабируемый
- Самодостаточный
- Быстрый
- …. движок поиска

Конкуренты Elasticsearch

- Sphinx
  - SQL-подобный язык запросов
- Solr
  - REST
  - Менее популярное решение решение чем Elasticsearch

Почему Я рассказываю про ES?

- На текущий момент самый фичастый движок поиска
- Наиболее просто развертываемый
- Позволяет строить сложные поисковые запросы вне зависимости от их типа

Почему Я рассказываю про ES?

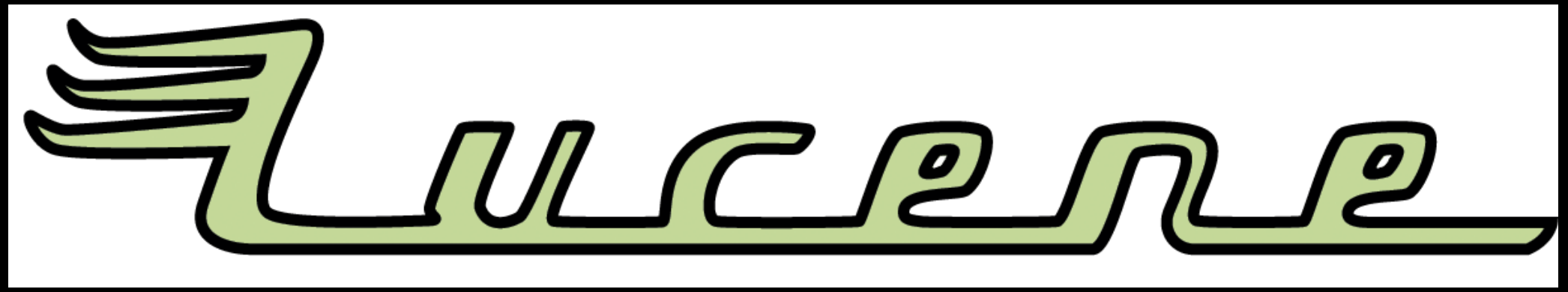
- Можно использовать в качестве основной базы данных
- Полностью самодостаточная. В отличие от Solr не требует развернутого ZooKeeper
- Opensource

Elasticsearch – История

Логотип Elasticsearch

История Elasticsearch

- Opensource
- Автор Shay Banon
- Начал проект Compass в 2004 г.,
- В 2010 г. как результат большого переписывания появился Elasticsearch
- В качестве основы используется Lucene



Логотип Lucene

История Lucene

- Opensource
- Автор Douglass Cutting
- Начал проект в 1997 г.
- В 1999 г. выложил на SourceForge.net
- В 2001 г. вошел в состав проектов фонда Apache

Почему Elasticsearch > Lucene

- Lucene – Java библиотека
- Elasticsearch REST сервер
- Elasticsearch добавляет поверх Lucene
  - Масштабируемость
  - Реплицируемость
  - Аналитические инструменты

Elasticsearch - 🚀

Начнем с простого примера

Индексируем

POST localhost:9200/information/person/1

```
{
 “name” : “Paul”,
 “lastname” : “Smith”,
 “job_description” : “Business Analyst”
}
```

Поиск

GET localhost:9200/_search?q=Pau

GET localhost:9200/_search?q=job_description

Поиск

```
curl -X GET "localhost:9200/information/_search?pretty" -H 'Content-Type: application/json' -d'
{
    "query" : {
        "term" : { "user" : "kimchy" }
    }
}
```

Ищем по префиксу

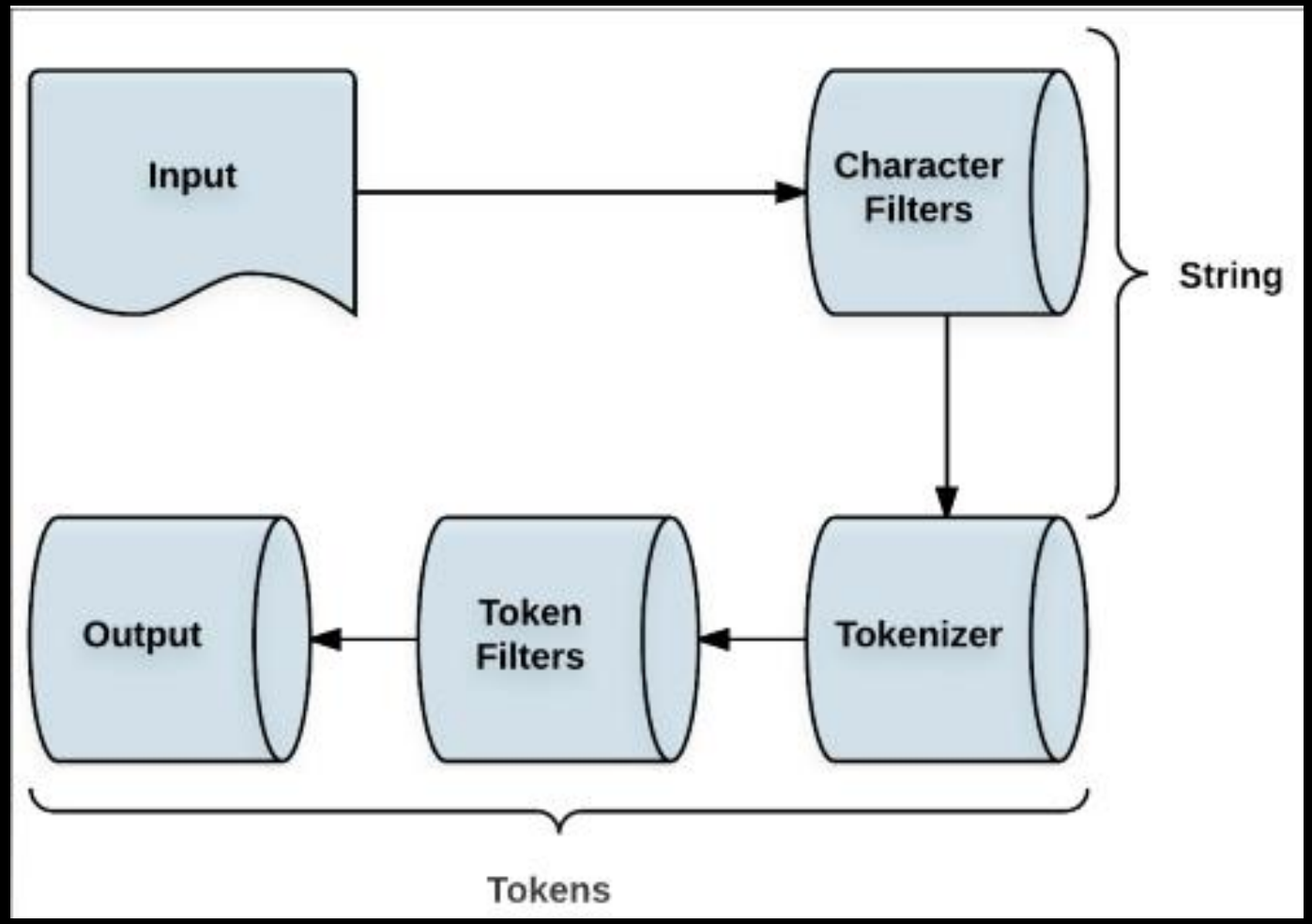
```
curl -X GET "localhost:9200/_search?pretty" -H 'Content-Type: application/json' -d'
{
    "query": {
        "prefix": {
            "user": {
                "value": "ki"
            }
        }
    }
}
```

Теория чтобы искать умнее

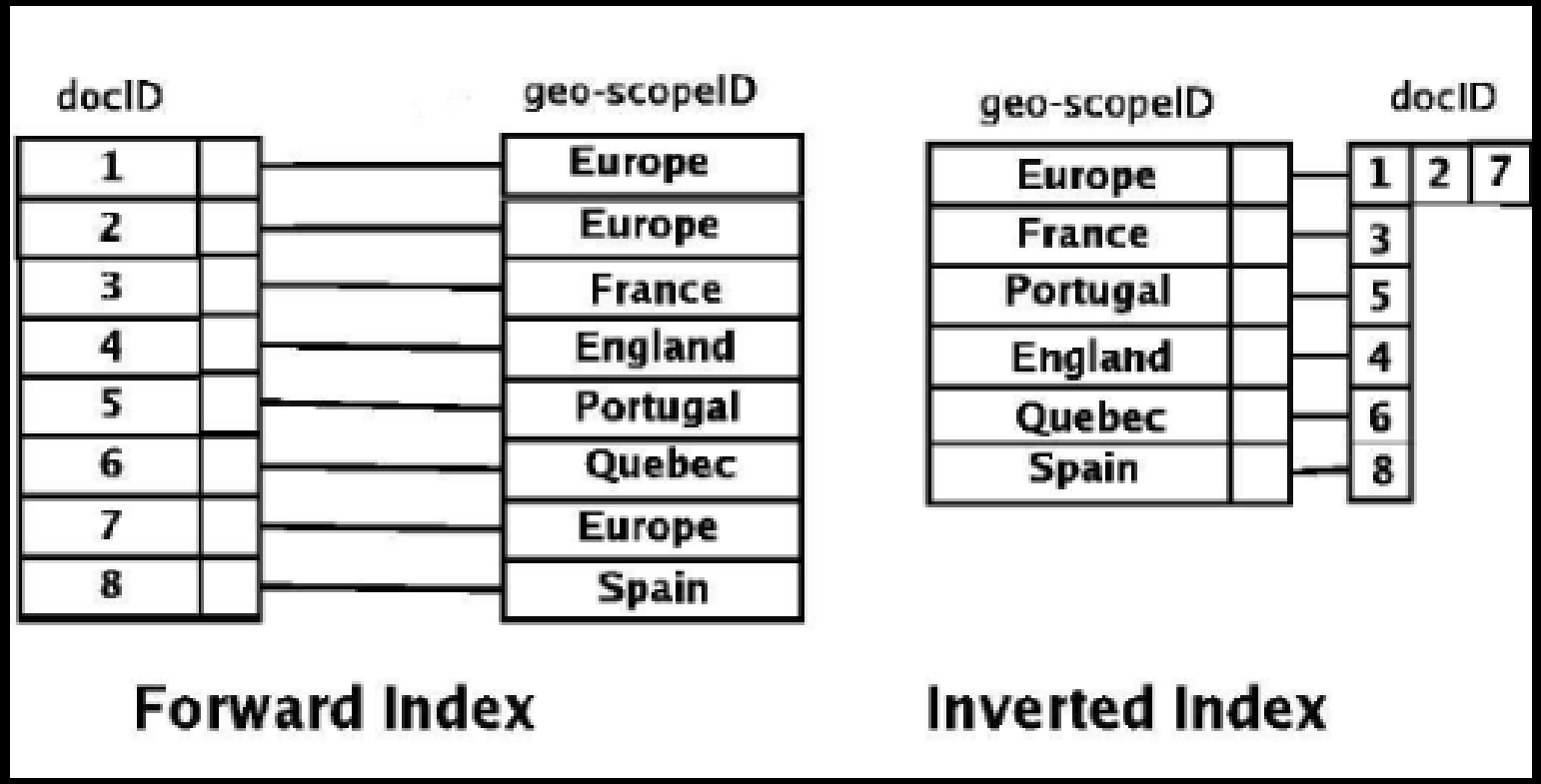
Elasticsearch – не черная коробка

- Elasticsearch использует классическую теорию поиска
- Для того что бы качественно настроить поиск нужно понимать процесс обработки текста

Процесс индексации



Обратный индекс



Токенизация

- Для построения обратного индекса нужно выделить границы слов
- Не для всех языков просто (Hint: Иероглифическое письмо)

Токенизация

- Управляя токенизацей важно понять что мы хотим считать одним «словом»
- Кастомные настройки могут понадобиться для
  - Emailов
  - #хештегов
  - Номера телефонов

Нормализация

- Приводим схожие слова к одному значению
- Хотим ли мы считать слова в разном падеже одним и тем же словом?
- Мы теряем немного информации, но и будем находить больше

Нормализация

- Можно обрезать окончания слов, это **Стеммизация**
- Работает лучше, в английском языке
- Создает некоторые проблемы с мусором

- Можно привести к словарной форме, это **Лемизация**

- Из коробки ES > 6.0 умеет и то, и то

Фильтрация

- Не все слова («токены») полезны в поиске

- HTML разметка
- Markdown

- ES Умеет резать HTML если настроить фильтр.

Ранжирование

- Не все результаты одинаково интересны пользователю
- В каком порядке расположить элементы в поисковой выдаче?
- Сигналы релевантности
  - Совпадение слов в запросе и тексте
  - Свежесть
  - Число просмотров
- Как смешать сигналы, чтобы получить ранг?

Делаем поиск умнее

Настраиваем маппинги

- Маппинг – аналог схемы в ES
- Что нам важно
- Тип поля
- Анализатор

Настраиваем маппинги

- Сущности в ES – иммутабельные
  - Настраивать можно только не индексированные поля
  - Лучше настраивать при создании индексов

Настраиваем маппинги

```
curl -X PUT "localhost:9200/my-index?pretty" -H 'Content-Type: application/json' -d'
{
  "mappings": {
    "properties": {
      "age":    { "type": "integer“ },
      "email":  { "type": "keyword“ , "analyzer": "std_english"  },
      "name":   { "type": "text"  }
    }
  }
}
```

Настраиваем маппинги

- ES Автоматически настраивает поля и для текстовых полей выбирается анализатор “standard”

- Посмотреть настройки маппингов можно так:

```
curl -X GET "localhost:9200/my-index/_mapping?pretty"
```

Настраиваем анализаторы

- Pipeline обработки конкретного поля
- Создаем свой если стандартных мало

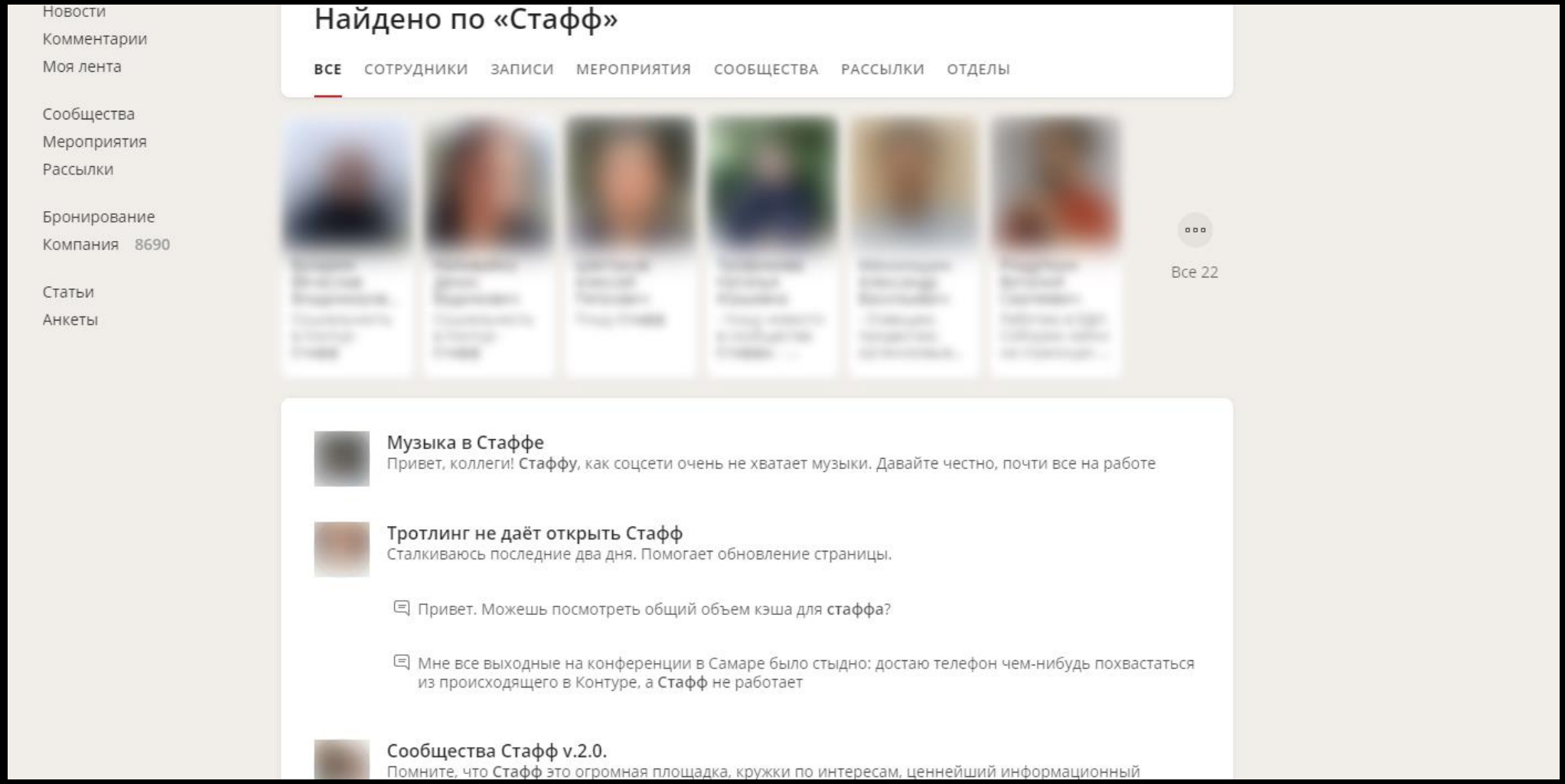
Настраиваем анализаторы

```
PUT my_index {
  "settings": { "analysis": {
      "analyzer": {
        "my_custom_analyzer": { "type": "custom", "tokenizer": "standard",
          "char_filter": ["html_strip" ],
          "filter": [
            "lowercase",
            "asciifolding"
          ] } } } } }
```

Пример использования – Контур.Стафф

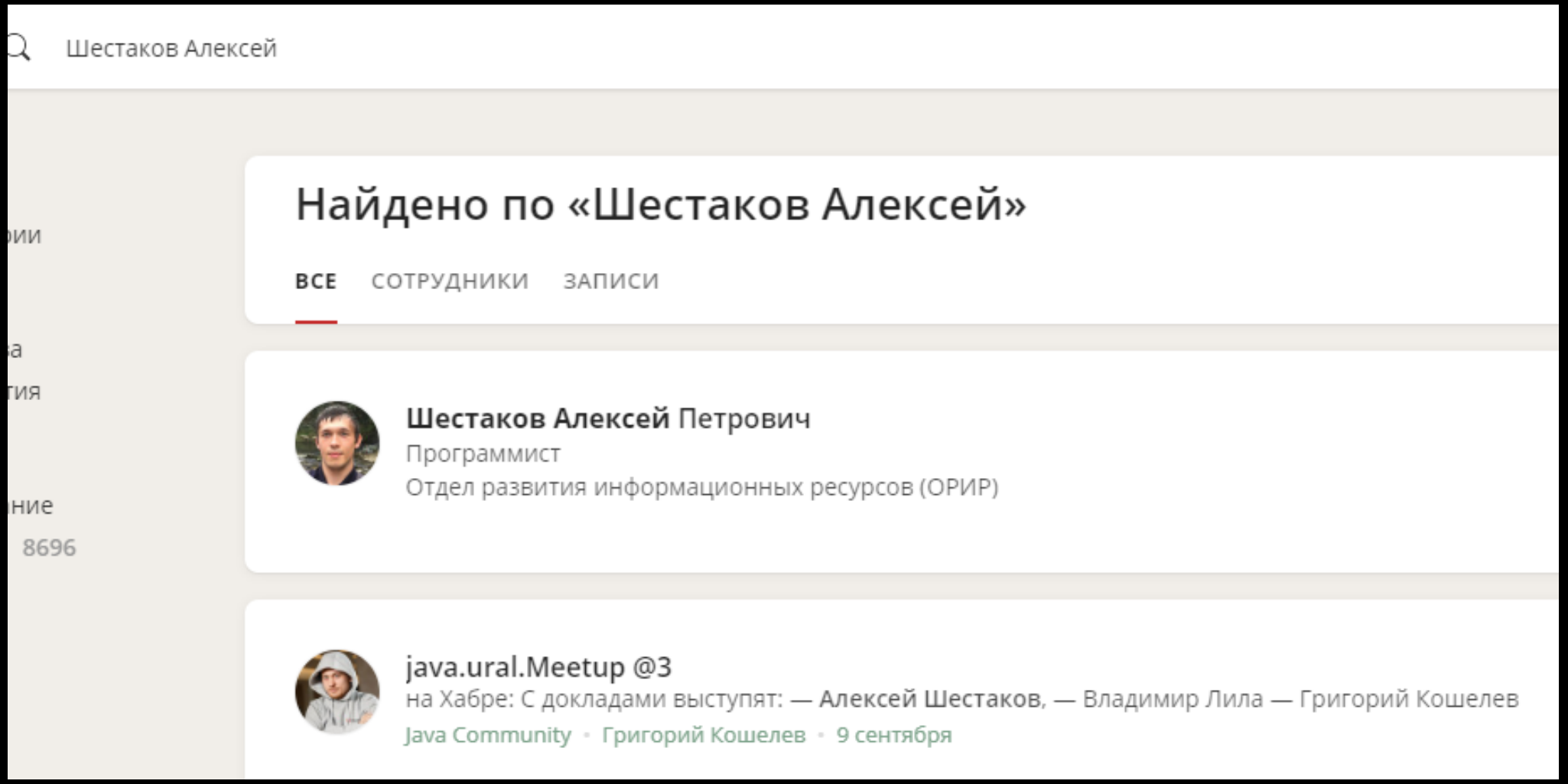
Контур.Стафф

- Использует ES для Поиска
- Справочник сотрудников
- Социальная сеть
- За неделю посещаемость 99%.
- Много разного контента
- ~ 4k обращений к SERP за день



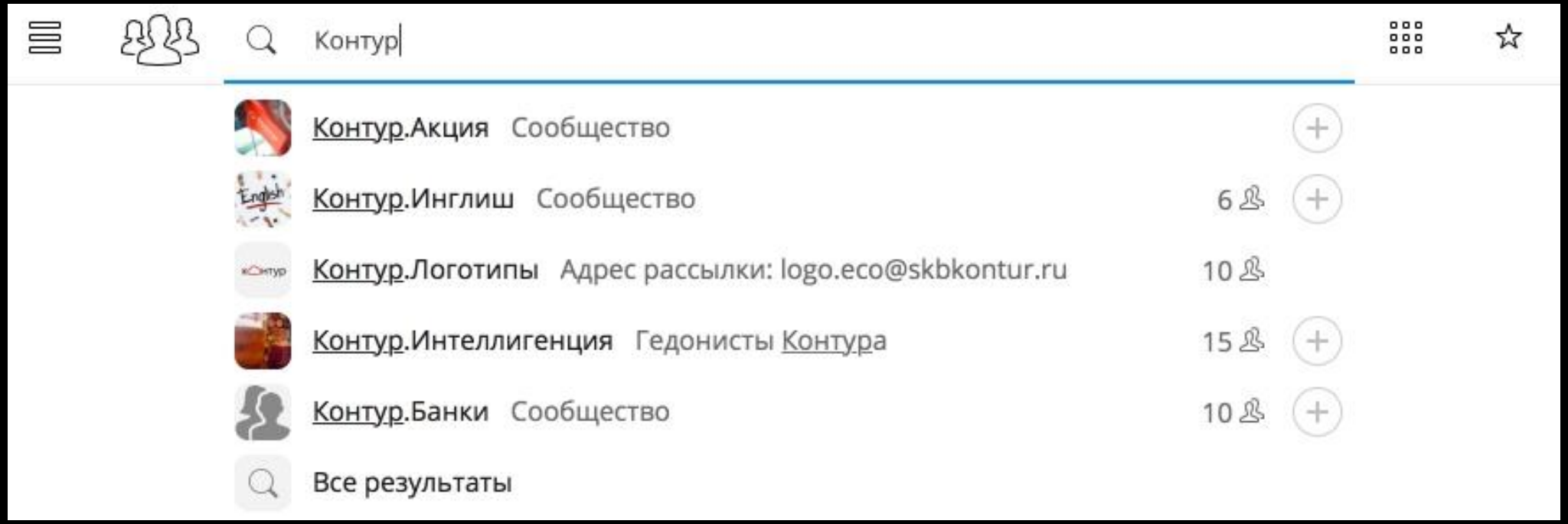
Поиск в Стафф

Это и есть SERP Стафф

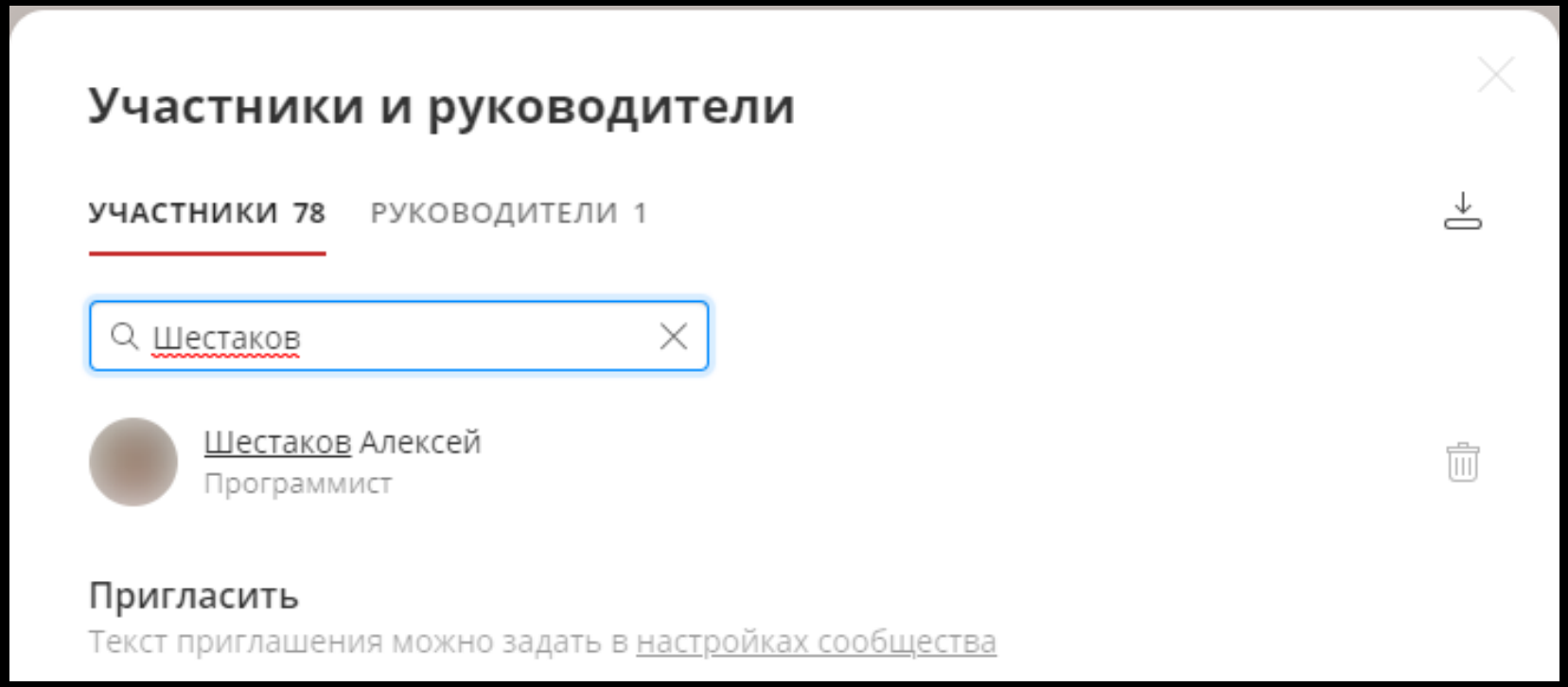


Поиск в Стафф

Это и есть SERP Стафф

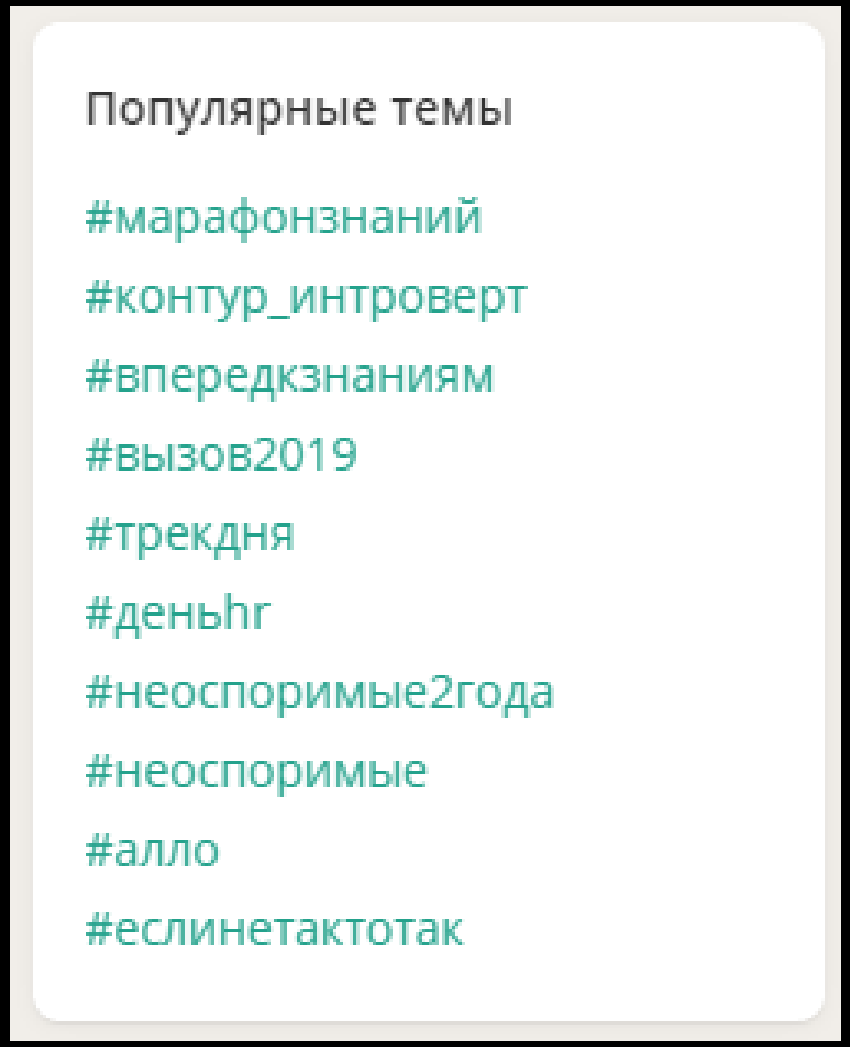


Поисковая подсказка в Стафф



Это тоже подсказки

Аналитика


Популярные темы
#марафонзнаний
#контур_интроверт
#впередкзнаниям
#вызов2019
#трекдня
#деньhr
#неоспоримые2года
#неоспоримые
#алло
#еслинетактотак

Контур.Стафф

- Использует продвинутые фишки ES
  - Скрипты ранжирования для управления результатом
  - Вложенные объекты
  - Релевантностью полей в запросе
- Настроены фильтры управления видимости
- Агрегирование

Что дальше

Управление релевантностью

- Собирать обратную связь.
- В ручную или автоматически (aka ML) настраивать ранжирование

- Для этого:
  - Скрипты ранжирования
  - Специальные плагины

Продвинутые фишки Elasticsearch

- Вложенные документы
- Скрипты ранжирования
- Агрегирование - Подсчет статистик, группировка данных, аналитические возможности ES

- TODO: заменить на примеры

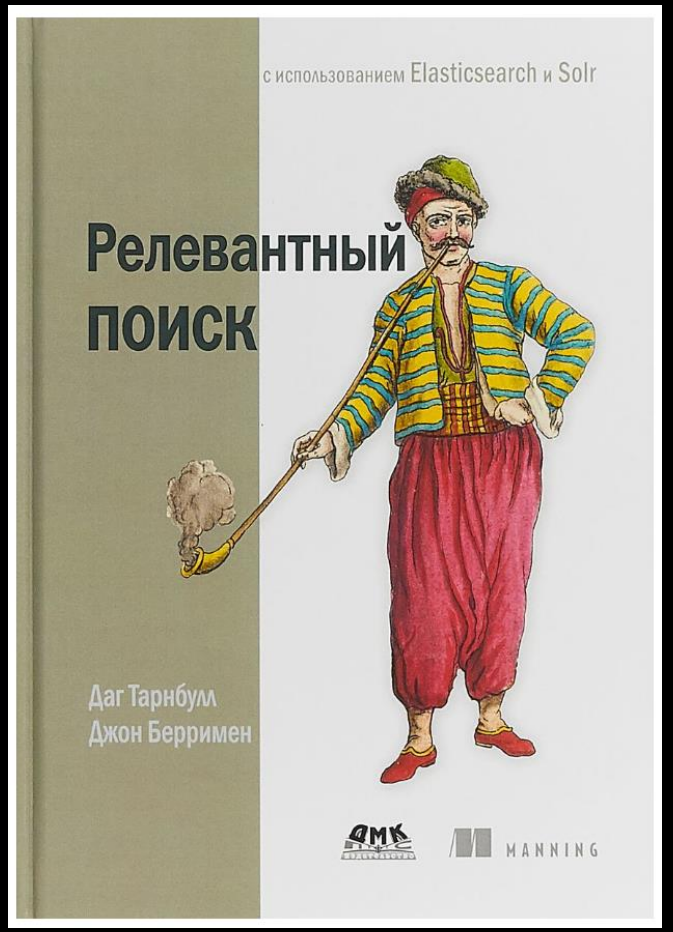
Литература

- Хорошее введение в теорию информационного поиска
- Идеально если вы хотите глубже понимать ES либо написать свой аналог
- В новой редакции нет с 2008 г.
  немного устарела в ML части.

https://www.ozon.ru/context/detail/id/5497130/

Литература

- Конкретные рецепты
- Практика использования ES
- Ориентация на ручное управление ранжированием без ML



https://www.ozon.ru/context/detail/id/144631193/

Вопросы?

Алексей Шестаков

Инженер-Программист

shestakovap@skbkontur.ru

kontur.ru