Добавляем поисковую строку в ваше приложение с Elasticsearch

Маленький google в вашем приложение

Шестаков Алексей

Инженер-Программист

СКБ Контур

Кто я

- Backend
- Внутренняя разработка
- Пишу Контур.Стафф – внутреннюю социальную сеть контура

Что будет в этом докладе

- История и Введение в Elasticsearch
- Примеры использования
- Теория
- Как сделать поиск умнее
- Как мы сделали наш поиск на Elasticsearch

Чего не будет в этом докладе

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

Контур.Стафф – Соцсеть Контура

- Пользователи ~ 8000
- Посты
- Сообщества ~ 10k
- Мероприятия

Нам нужен был поиск

- Разграничивать доступ
- Находить сотрудников
- Находить информацию о сотрудниках
- Находить контент

Почему у нас появился Elasticsearch

У нас тогда было и не подошло:

- SQL
- Mongo
- Lucene

Что такое Elasticsearch?

- Opensource
- Широко используемый
- Масштабируемый
- Быстрый
- Кастомизируемый
- REST сервер(ы)

История Elasticsearch

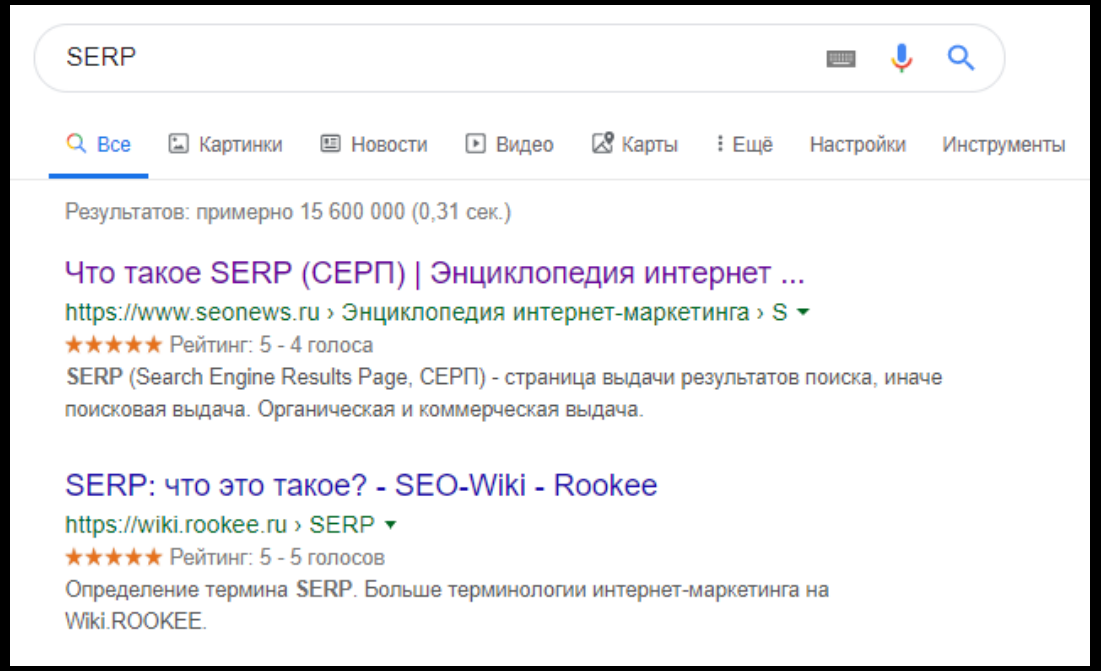
- Opensource
- Автор Shay Banon
- Начал проект Compass в 2004 г.,
- В 2010 г. как результат большого переписывания появился Elasticsearch
- В качестве основы используется **Lucene**

Почему Elasticsearch > Lucene

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

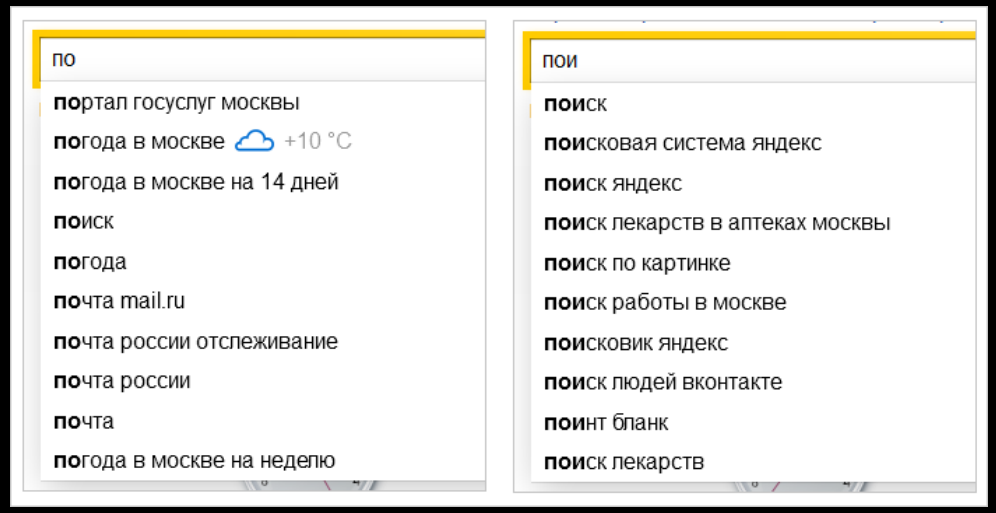
SERP – Search Engine Result Page

- Страница с результатом поиска
- Снипет – часть документа, важная в контексте запроса



Поисковые подсказки

- Search-as-you-type
- Помогает сформировать поисковый запрос
- Переход в SERP или на сразу на результат



Elasticsearch – не черная коробка

- Elasticsearch использует классическую теорию поиска

- Для того что бы качественно настроить поиск нужно понимать процесс обработки текста

Модель – Bag of Words

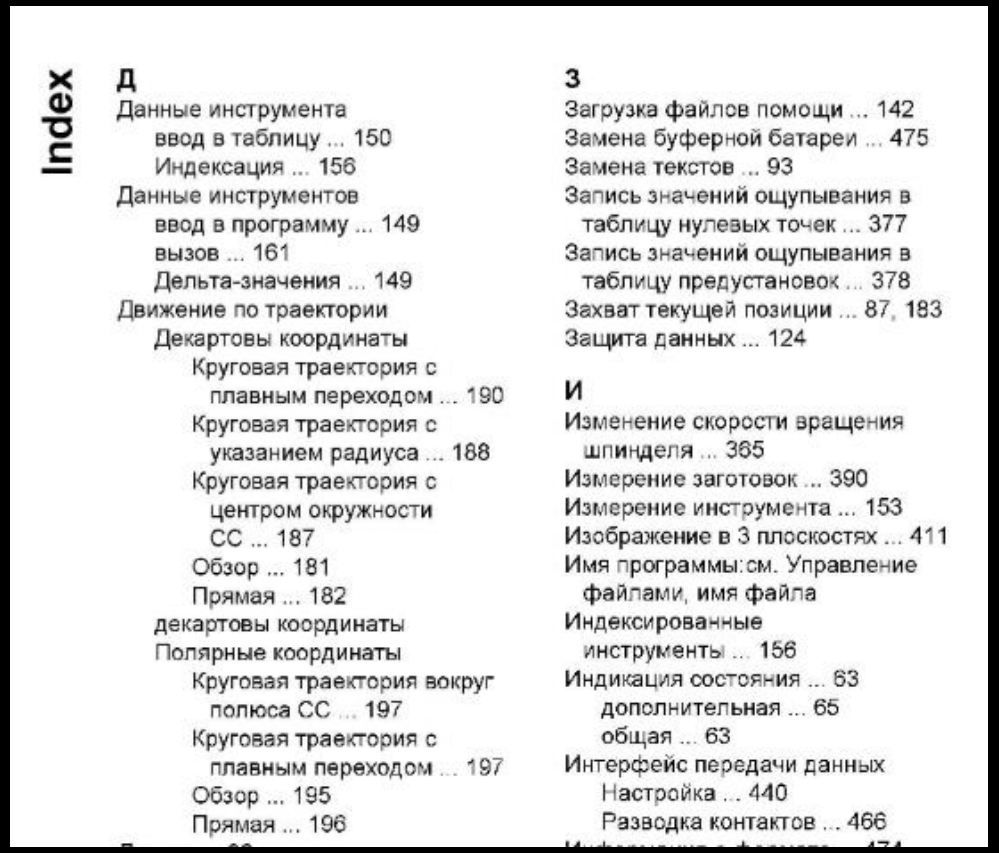
- Документом называется индексируемое текстовое поле
- Каждый документ режется на «слова» или «токены»
- Порядок не учитывается

Что такое «Токен»?

- «Токен» – минимальная единица поиска
- Обычно слово, но не всегда
- Поиск меньших частей «токена» возможен, но более затратен

Поисковый индекс

- ES Строит индекс почти как в книгах
- Запись в индексе:
- Связка токена – и номера документов, где слово встречается
- Еще называется «Инвертированным»



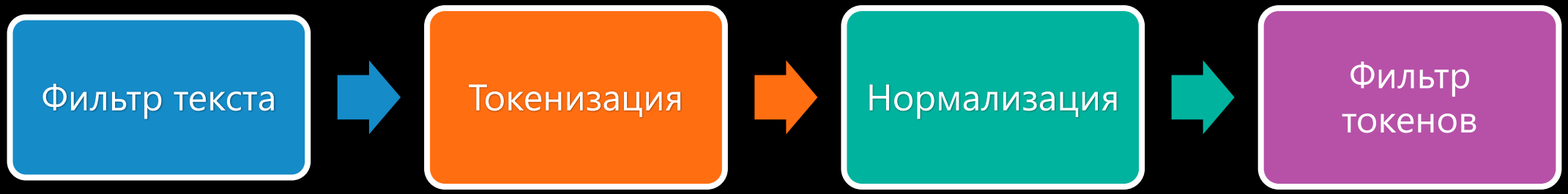
Index

Инвертированный индекс

Слово	Номера документов где слово встречается
все	2,4,7
всегда	2,7
делать	2
дело	3,3

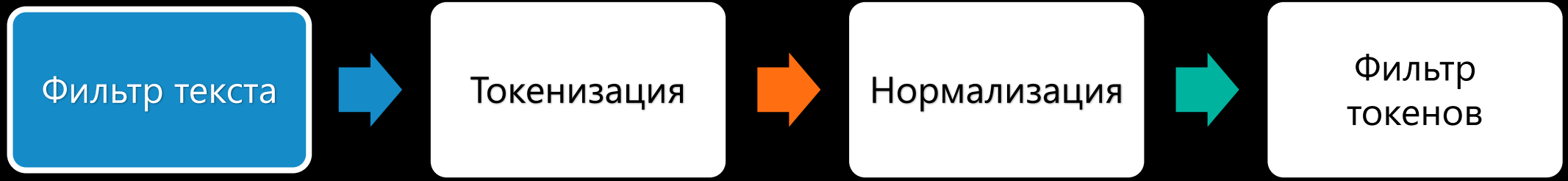
Процесс индексации

- Процесс построения индекса
- Управляя процессом индексации можно
  - Убрать лишнее из выдачи
  - Научить поиск находить больше



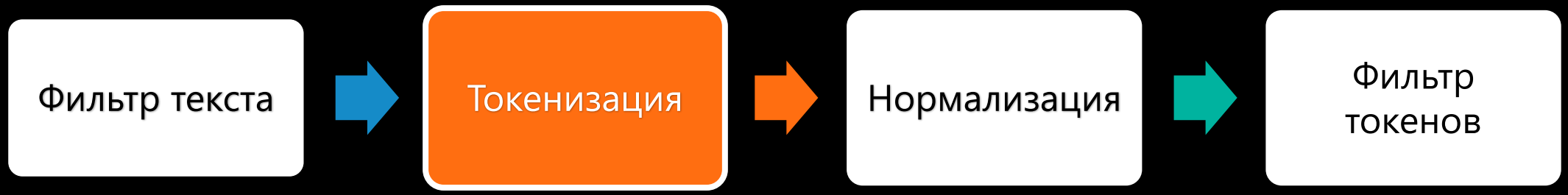
Фильтр текста

- Убираем из текста все что не должно искаться
- В терминах ES – «char_filter»
- Пример – «html_strip» - убирает HTML разметку



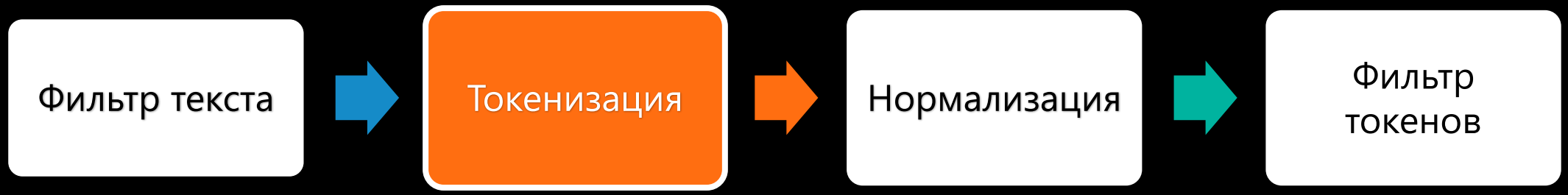
Токенизация

- Управляя токенизацей важно понять что мы хотим считать одним «словом»



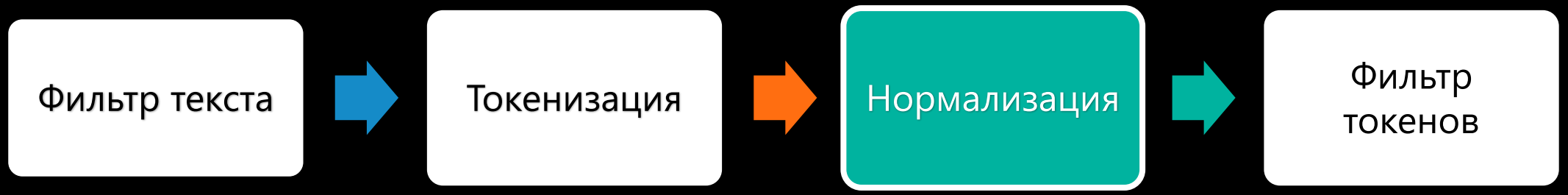
Токенизация

- Мобильные номера
  - + 7 (933) 34344322 –> +7(933)34344322, (933)34344322, 34344322
- От выбора токенов зависит будет ли находиться тот или иной вариант



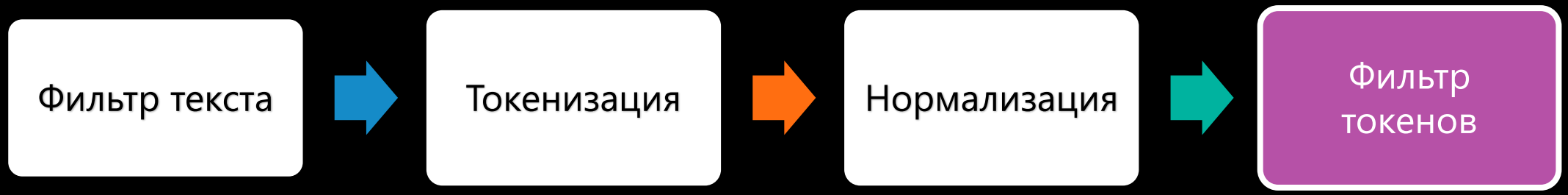
Нормализация

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



Фильтр токенов

- Не все токены нам интересны
  - Пример: предлоги, местоимения – «Стоп-слова»



Elasticsearch – Индексируем

```
POST localhost:9200/information/person/1
{
 “user” : “Paul”,
 “lastname” : “Smith”,
 “job_description” : “Business Analyst”,

“age” : 33
}
```

Elasticsearch – Поиск

curl -X GET "localhost:9200/*information*/_search?pretty" -H 'Content-Type: application/json' -d'

```
{
    "query" : {
        "term" : { "user" : "kimchy" }
    }
}
```

Еще варианты:

GET localhost:9200/_search?q=kimchy

GET localhost:9200/_search?q=user:kimchy

Elasticsearch – Поиск по префиксу

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

Настраиваем анализаторы

- Управляет процессом индексации конкретного поля
- Можно создать свой если стандартный не подходит

Настраиваем анализаторы

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

Настраиваем маппинги

- Маппинг – аналог схемы в ES
- Тип поля
- Анализатор

Настраиваем маппинги

- ES Может автоматически создавать маппинги
- Сущности в ES – иммутабельные
- Настраивать нужно при создании индексов

Настраиваем маппинги

```
curl -X PUT "localhost:9200/information?pretty" -H 'Content-Type: application/json' -d'
{
  "mappings": {
    "properties": {
      "age":    { "type": "integer“ },
      "job_description":  { "type": " text“ , "analyzer": " job_description_analyzer" },
      “user":   { "type": "keyword"  }
    } } }
```

Настраиваем маппинги

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

- Посмотреть настройки маппингов можно так:

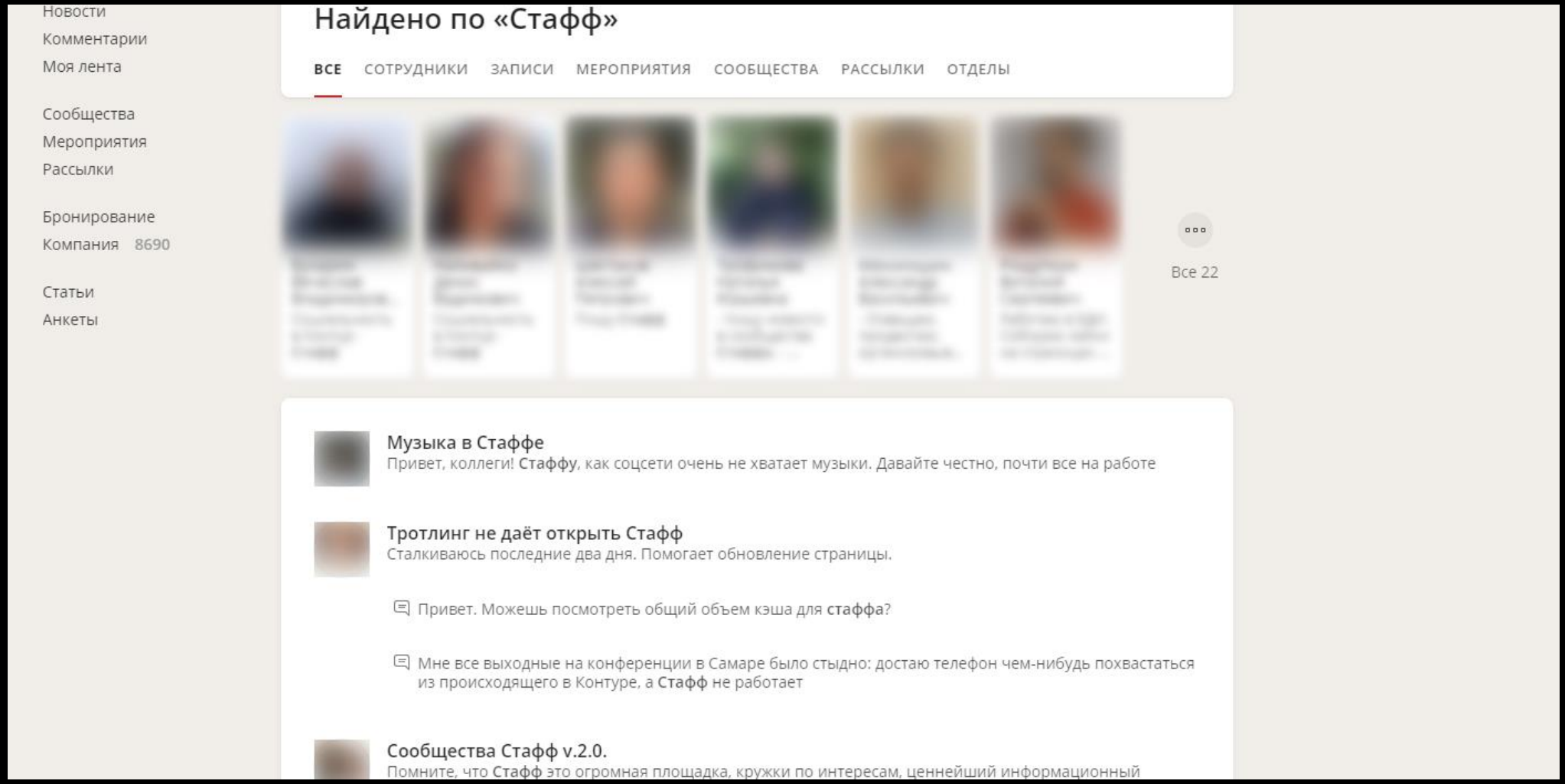
```
curl -X GET "localhost:9200/my-index/_mapping?pretty"
```

Строим правильно запрос поиска

```
curl -X GET "localhost:9200/_search?pretty" -H 'Content-Type: application/json' -d'
{
  "query": {
    "multi_match" : {
      "query" : “Что ищем",
      “name" : [ “name^2", "job_description" ]
    }
  }
}
```

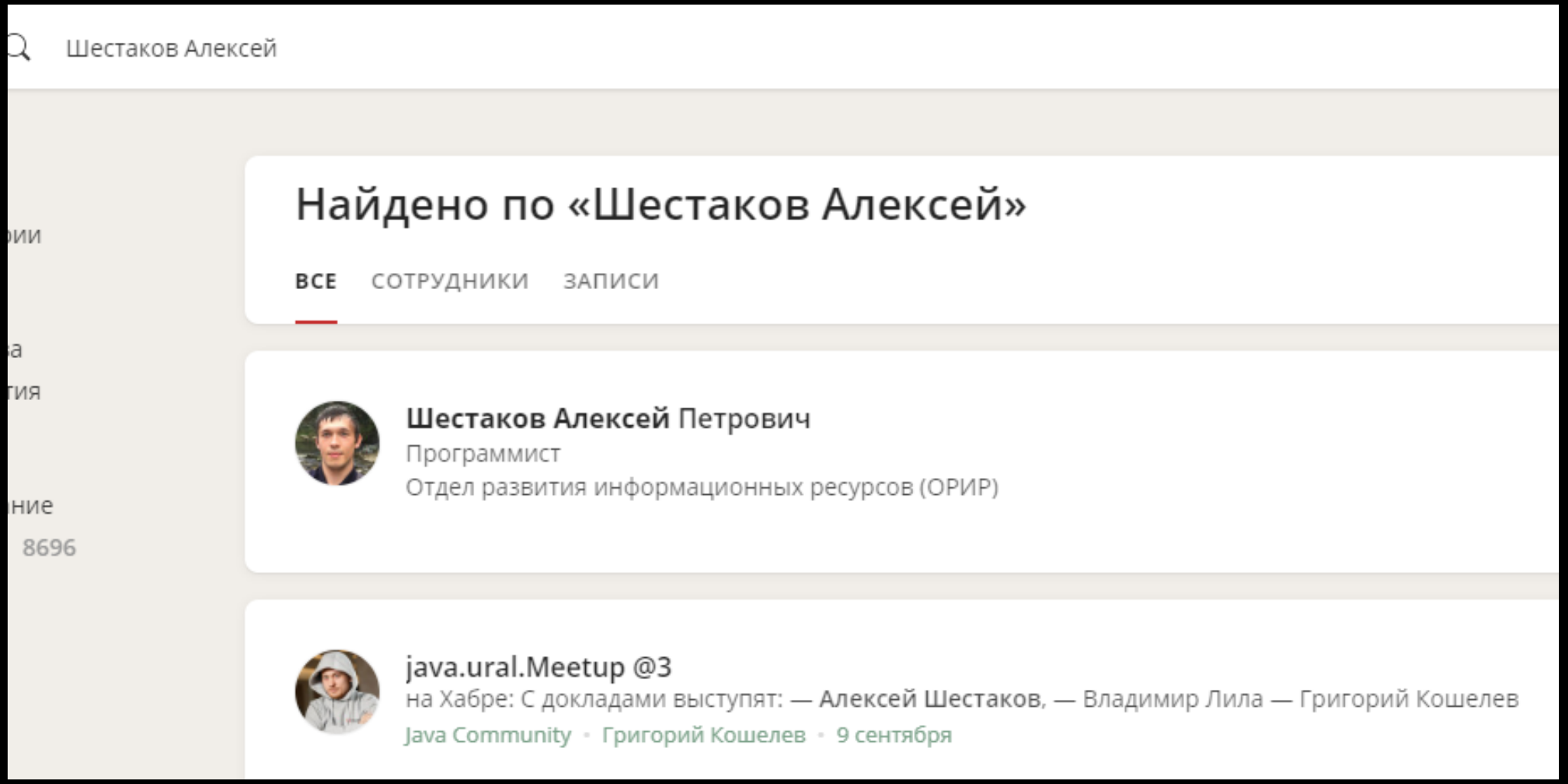
Контур.Стафф

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



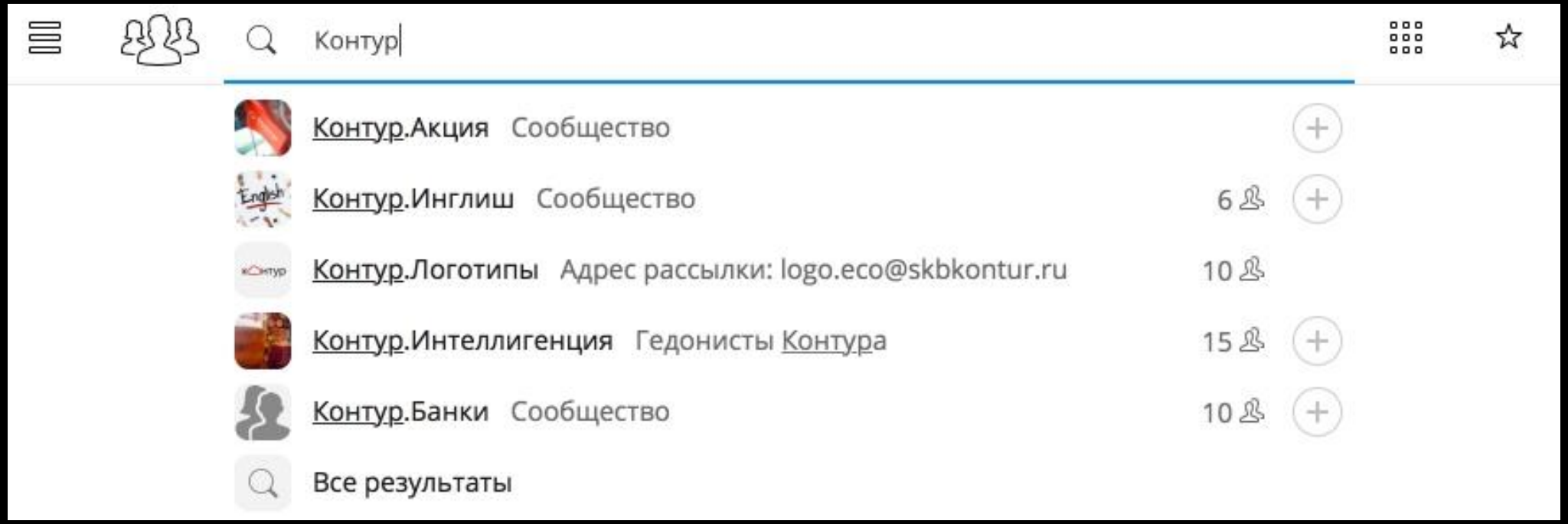
Поиск в Стафф

Это и есть SERP Стафф

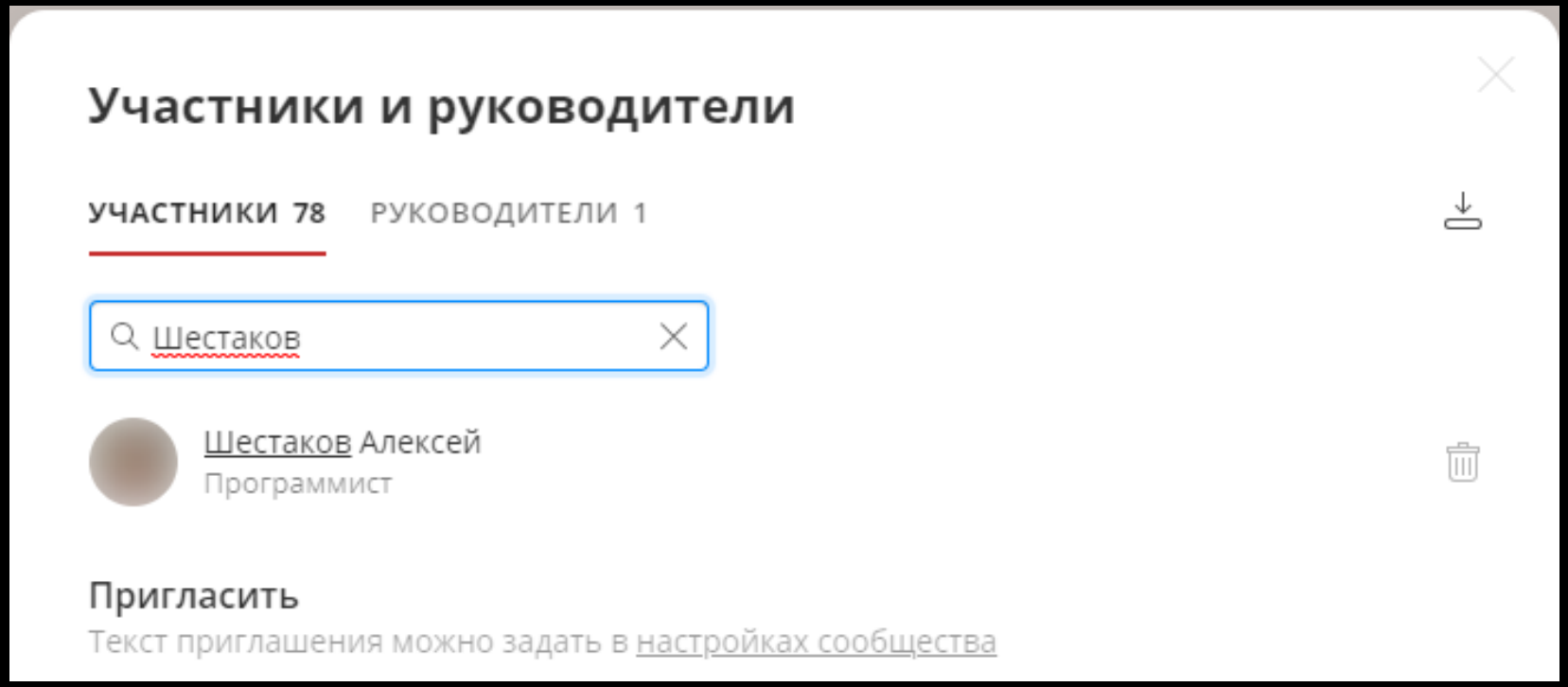


Поиск в Стафф

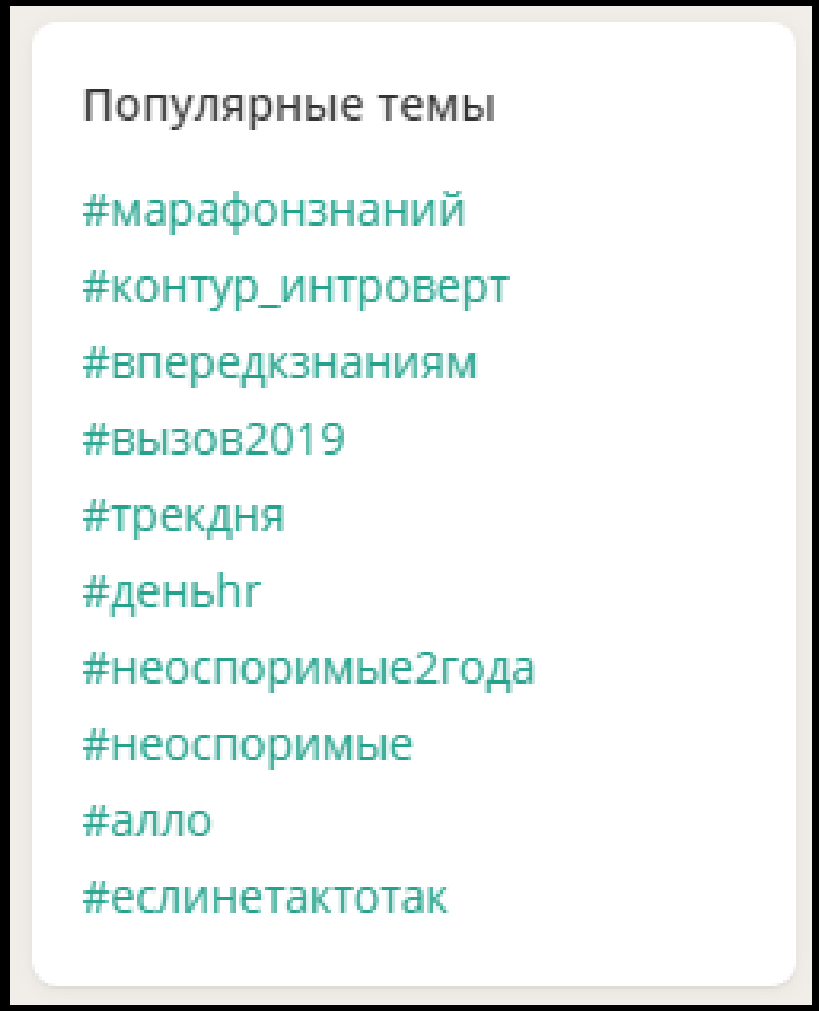
Это и есть SERP Стафф



Поисковая подсказка в Стафф



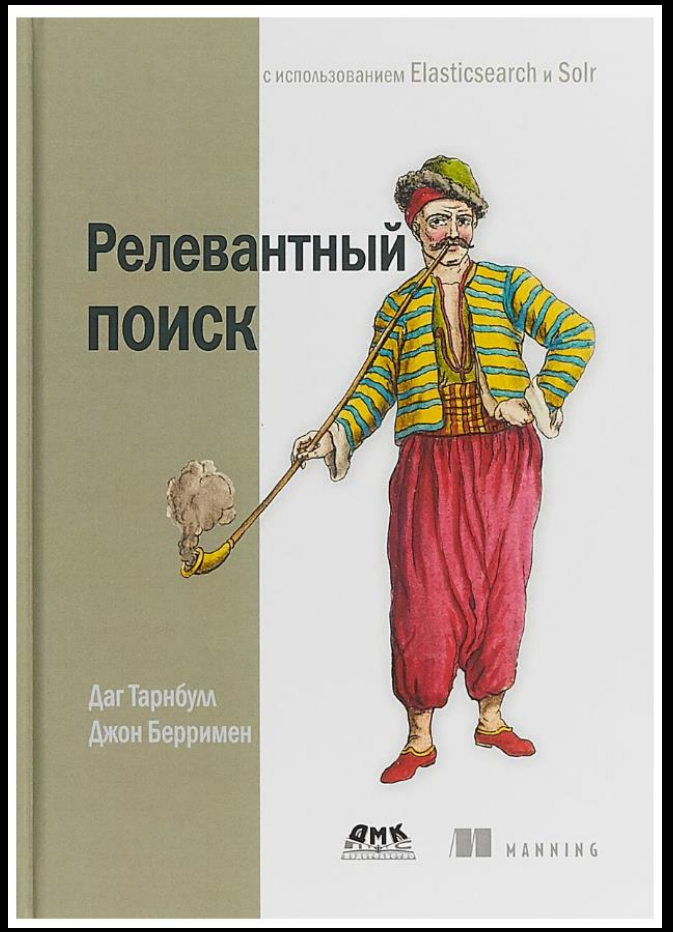
Это тоже подсказки



Популярные теги за последнее время

Литература

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



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

Литература

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

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

Вопросы?

Алексей Шестаков

Инженер-Программист

shestakovap@skbkontur.ru

kontur.ru