

Документация

- Проектная документация
 - Видение проекта
 - Функциональные требования
- Архитектура
- Принцип расчета индекса
- Дизайн-макет
- Документация API
 - Документация сервиса
 - Документация бека
- Описание ролевого доступа

Проектная документация

Проектная документация

Видение проекта

https://docs.google.com/document/d/1MiBwh1aj6nsGZyaqJgGu5FnaC_Xvub0CWARW_Endo8w/edit?u

Проектная документация

Функциональные требования

<https://docs.google.com/document/d/1XaGeG-V8wc3K0vuZUWEoluA0S7fye5aCzVjlofLX6Fs/edit?usp=sharing>

Архитектура

<https://gitlab.rarus.ru/dr-1C-retail/index-spp-doc>

Принцип расчета индекса

1. Индекс - отношение число посетителей к коммерческой площади объектов за N-периодичность, где N может принимать значения: день, неделя, месяц, квартал и год. Индекс всегда разбивается на N-периодичность.

Правила выбора периода каждой N-периодичности:

- день - календарный день
- неделя - календарная неделя (с понедельника по воскресенье)
- месяц - календарный месяц (с первого дня месяца по последний день месяца)
- квартал - календарный квартал (с первого дня квартала по последний день квартала)
- год - календарный год (с первого дня года по последний день года)

Общая формула расчета индекса:

$$\left(\sum_{m=1}^N d_m\right) \cdot \frac{1}{S}$$
 , где d_m - посещаемость объекта за m-день по счету периодичности, N - количество календарных дней в периодичности, S - площадь объекта. Например:

Объект: ТЦ "Солнышко"

Площадь: 136000 м².

Посещаемость:

День	Посещаемость
04.09.2023	460 792
05.09.2023	349 621
06.09.2023	347 337
07.09.2023	339 815
08.09.2023	340 794
09.09.2023	341 773
10.09.2023	350 024
11.09.2023	395 949
12.09.2023	354 614
13.09.2023	404 033
14.09.2023	363 209
15.09.2023	369 375

16.09.2023	351 758
17.09.2023	365 543

Период построения: 04.09.2023 - 17.09.2023

Периодичность: Неделя

Индекс за 04.09-10.09:

$(460792+349621+347337+339815+340794+341773+350024)/136000 = \mathbf{18,60}$

Индекс за 11.09-17.09:

$(395949+354614+404033+363209+369375+351758+365543)/136000 = \mathbf{19,15}$

2. Шаблоны - сохраненные фильтры по атрибутам объекта для формирования индекса. У шаблона может быть указан период фиксации в виде даты начала и даты окончания (либо должна быть функция фиксации индекса за период по шаблону).

Пример шаблона:

```
{
  "template": {
    "address": [
      {
        "fias_ids": [
          "123123",
          "123123",
          "123213"
        ],
        "representation": "Россия; Рязань; Глазнь"
      }
    ],
    "area": [
      {
        "value": 45000,
        "operator": "gte"
      },
      {
        "value": 90000,
        "operator": "lte"
      }
    ],
    "is_mall": true,
  }
}
```

```

      "props": [
        {
          "id": "uid",
          "prop_name": "1706532045",
          "values": [{
            "id": 2,
            "value": "Одежда"
          }],
          "operator": "eq",
          "group": false
        },
        {
          "id": "uid2",
          "operator": "or",
          "group": true,
          "children": [
            {
              "id": "uid3",
              "prop_name": "1706532045",
              "values": [{
                "id": 4,
                "value": "Женская одежда"
              }],
              "operator": "neq",
              "group": false
            },
            {
              "id": "uid4",
              "prop_name": "1706105749",
              "values": [
                {
                  "id": 42,
                  "value": 1
                },
                {
                  "id": 45,
                  "value": 3
                }
              ],
              "operator": "in",

```



```

        "group": false
    },
    [
        ]
    }
]
}
}

```

eq — равно

neq — не равно

lt — меньше

lte — меньше или равно

gt — больше

gte — больше или равно

in - находит объекты со свойством, значение которого равно одному из значений из списка

in - находит объекты со свойством, значение которого не равно ни одному из значений из списка

is_mall принимает значения - true, false, none. none отменяет применение фильтра

3. Фиксация индекса

По существующим шаблонам должен быть предусмотрен механизм фиксации индекса работающий следующим образом: строится индекс по указанным фильтрам шаблона и сохраняется за указанный период фиксации шаблона.

Фиксировать значение индекса рекомендуется в разрезе дня.

Значения индекса для фиксации строятся на основе данных посещаемости по объектам с учетом основных принципов расчета и формулы расчета индекса.

В случае если за определенный период нет данных для расчета индекса, то такой индекс заполняется в разрезе дня со значением равным 0 (нулю).

4. Расчет индекса

Пример запроса индекса доступен [по ссылке](#).

Основные принципы расчета индекса:

- Данные по объектам берутся только за период работы с объектом (атрибуты объекта: дата начала работы, дата окончания работы с объектом).
- Данные по объекту не берутся в случае если по объекту отсутствуют данные за полную N-периодичность, в ином случае данные по объекту участвуют в расчете индекса.
-

В случае построение индекса по шаблону (запрос по "template" из примера запроса):

- Период построение отчета попадает в период фиксации по шаблону:
 - Данные по индексу строятся из фиксированного индекса по шаблону
- Период отчета не попадает в период фиксации по шаблону:
 - Данные по индексу строятся на основании данных посещаемости объектов
- Период отчета по индексу частично попадает в период фиксации по шаблону:
 - За период который попадает в период фиксации данные строятся из фиксированного индекса по шаблону
 - За период который не попадает в период фиксации данные на основании данных посещаемости объектов
- В случае построение индекса по объектом (запрос по "objects" из примера запроса):
 - Данные строятся на основании посещаемости объектов

[Пример расчета индекса в google-таблице](#)

Дизайн-макет

<https://drive.google.com/file/d/1bhgVd5DeUeBM8EOZE8Clovrc1kcXMtPH/view?usp=sharing>

Документация API

Документация API

Документация сервиса

[`https://indshop-pg-02.corp.rarus-cloud.ru/swagger/index.html#/`](https://indshop-pg-02.corp.rarus-cloud.ru/swagger/index.html#/)

Документация API

Документация бека

<https://indshop-pg-03.corp.rarus-cloud.ru/docs#>

Описание ролевого доступа

1. Ролевая система должна обеспечивать уровень доступа на уровне сущности. С правами "Видимость", "Редактор"
 1. Например. недоступна видимость и редактирование шаблонов. Доступно редактирование объектов. Доступно видимость клиентов, не доступно редактирование.
- 2.

Предложение по ролевой системе на бекенде

TODO...

Ролевая система на бекенде реализуется на базе JWT токена

```
{  
  ...  
  
  "user_id": USER_ID,  
  "organization_id": ORGANIZATION_ID,  
  "roles": LIST_OF_ROLES,  
  "type": TOKEN_TYPE,  
  "rights": RIGHTS_MAP,  
  
  ...  
}
```

- USER_ID - id пользователя на бекенде (у суперпользователя равен 0)
- ORGANIZATION_ID - id клиента(организации) (у суперпользователя отсутствует или равен None)
- LIST_OF_ROLES - список (со строками) ролей назначенных пользователю на беке
- TOKEN_TYPE - тип токена (access или refresh)

- RIGHTS_MAP - агрегированный словарь с правами и их значениями

Варианты ролевого доступа (возможно комбинирование)

- доступ к роуту по значения права из RIGHTS_MAP
 - на роут устанавливается связка **right_name: minimum_value**
 - right_name - названия права доступа
 - minimum_value - минимальное значение права для доступа
 - view - 1
 - edit - 2
 - admin - 3
 - при запросе на роут сверяются значение RIGHTS_MAP
 - наличие установленного на роут **right_name** в токене
 - значение **minimum_value** в токене
- доступ по значению ORGANIZATION_ID. внутри роута производится проверка соответствия принадлежности сущности и пользователя к одной организации (для суперпользователя не производится)
- доступ по TOKEN_TYPE
 - только для **POST: /auth/refresh**
 - производится проверка значения TOKEN_TYPE из токена

Кейс (URL)	Право	Минимальное значение права	Сверка ORGANIZATION_ID
Запрос данных из dadata. POST: /dadata/suggest/address	dadata	view	
Запрос данных пресета запроса индекса (для отрисовки отчета беком). GET: /index/query/preset/{preset_id}	index_query_preset	view	
Запрос индекса в json POST: /index/query	index_query	view	
Экспорт индекса в файл POST: /index/query/{format}	index_query	view	
Запрос списка шаблонов запроса индекса GET: /reports	report	view	✓
Запрос информации по конкретному шаблону запроса индекса GET: /reports/{report_id}	report	view	✓
Создание нового шаблона запроса индекса POST: /reports	report	edit	✓

Изменение конкретного шаблона запроса индекса PATCH: /reports/{report_id}	report	edit	✓
Удаление конкретного шаблона запроса индекса DELETE: /reports/{report_id}	report	edit	✓
Запрос списка шаблонов рассылок GET: /campaigns	campaign (report?)	view	✓
Запрос информации по конкретному шаблону рассылки GET: /campaigns/{campaign_id}	campaign (report?)	view	✓
Создание нового шаблона рассылки POST: /campaigns	campaign (report?)	edit	✓
Изменение конкретного шаблона рассылки PATCH: /campaigns/{campaign_id}	campaign (report?)	edit	✓
Удаление конкретного шаблона рассылки DELETE: /campaigns/{campaign_id}	campaign (report?)	edit	✓
Запрос списка ролей GET: /roles	role	view	
Запрос списка прав GET: /roles/rights	role	view	
Создание новой роли POST: /roles	role	edit	
Изменение конкретной роли PATCH: /roles/{role_id}	role	edit	
Удаление конкретной роли DELETE: /roles/{role_id}	role	edit	
Запрос списка пользователей GET: /users	user	view	✓

Запрос информации по конкретному пользователю GET: /users/{user_id}	user	view	✓
Создание нового пользователя POST: /users	user	edit	✓
Изменение конкретного пользователя PATCH: /users/{user_id}	user	edit	✓
Изменение пароля конкретного пользователя PATCH: /users/{user_id}/password * при изменении собственного пароля - без проверки ролей.	user, None*	edit None*	✓
Удаление конкретного пользователя DELETE: /users/{user_id}	user	edit	✓