

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	✓