Взаимодействие с backend на api-platform

многие запросы проходят и под анонимусом, но чтобы послать запрос под юзером нужно добавить в заголовок

**Authorization:** Bearer *%access\_token%*

также, в заголовке **Accept** должно содержаться *application/json*

чуть менее, чем всегда ответ приходит в формате json

при редактировании объектов требуется слать json в теле запроса.(но не при авторизации, запрос токенов по-другому работает.)

запрос на список:

**GET** /api/objects

запрос на отдельный объект

**GET** /api/objects/*{id}*

запрос на создание объекта

**POST** /api/objects

запрос на изменение объекта

**PUT** /api/objects/*{id}*

запрос на удаление объекта

**DELETE** /api/objects/*{id}*

при запросе списка объектов в url-параметры добавляем условия поиска,

при поиске по простым параметрам указываем само значение, при поиске по свойствам-объектам указываем id

поиск по объектам(не считая массив обектов), числам и строкам работает нормально

насчёт дат не уверен

также есть зарезервированные параметры

&start=0&limit=25 - используются для пагинации

&orderby=size &orderbydesc=date - сортировка

также могут быть дополнительные возможности поиска, характерные для конкретных методов

при запросе списка ответ приходит в виде

{

count: int - общее количество объектов по данному критерию поиска(используется для пагинации)

items: array<object> (сами объекты)

}

OAuth2

при первом заходе необходимо делать запрос

POST /api/clients

Form Data

"name": "string",

"allowedGrantTypes": [

"password", "refresh\_token"

]

Response

"id": id,

"name": "string",

"randomId": "key",

"secret": "key",

"allowedGrantTypes": [

"password",

"refresh\_token"

]

в ответ приходят параметры нового клиента.

запоминаем, что нам пришло, и в каждый запрос токена подставляем следующие параметры:

**clientId** = client.id + '\_' + client.random\_id;

**clientSecret** = client.secret;

Получение токена

POST /oauth/v2/token

**Content-Type:**application/x-www-form-urlencoded

Form Data

**client\_id:** *%cliend\_id%*

**grant\_type:** password

**password:** %password%

**username:** %username%

**refresh\_token:** *%refresh\_token%*

**client\_secret:** *%client\_secret%*

Обновление токена

POST /oauth/v2/token

**Content-Type:**application/x-www-form-urlencoded

Form Data

**client\_id:** *%cliend\_id%*

**grant\_type:** refresh\_token

**refresh\_token:** *%refresh\_token%*

**client\_secret:** *%client\_secret%*

Пагинация

На запрос **/api/objects** с параметрами

**limit** - количество элементов на странице

**page** - номер страницы

пример:

api/objects?page=0&limit=1

Сортировка и фильтрация

Сортировка происходит по полям в песочнице (/api) начинающиеся с **order**

поле сортировки в квадратных скобках с значением asc или desc

пример:

api/objects?order[name]=asc

фильтрация происходит по названию свойства фильтрации

пример:

api/objects?name=joan

*возможны кастомные фильтры,работу которых лучше уточнить у backend разработчика.*