

Документация, содержащая информацию,
необходимую для эксплуатации экземпляра
программного обеспечения

Информационная система "Omni-API"

Оглавление	
Модель данных	3
Получение данных из хранилища объектов	4
Язык запросов Omni LHS Brackets	5
Операторы фильтрации	7
Операции фильтрации	7
Критерии отбора операций фильтрации	8
Параметры управления загрузкой объектов (пагинации)	8
Параметры группировки	8
Работа с API при помощи встроенного SwaggerUI	9
Авторизация	10
Выполнение запроса	11

Модель данных

Модель данных представляет собой граф, в котором узлами являются сущности, а ребрами связи между сущностями. Сущности содержат атрибуты. Атрибут представляет свойство объекта, какую-либо его характеристику.

Атрибуты могут быть двух видов: **скалярные** и **ссылочные**.

Скалярные атрибуты могут иметь следующие типы данных:

логический: **bool**

целочисленные: **byte, sbyte, short, ushort, int, uint, long, ulong**

десятичный: **decimal**

с плавающей точкой: **float, double**

текстовый: **string**

дата: **date**

дата и время: **datetime**

глобальный идентификатор: **guid**

Ссылочные атрибуты – это ссылка на объект сущности. Условно делятся на три вида:

ссылка на запись справочника – **Dict(КодСправочника)**

ссылка на объект сущности – **Object(КодСущности)**

ссылка на идентификатор сущности¹ – **Link(КодСущности)**

Объект – экземпляр сущности, описание конкретного объекта предметной области. Объект представляет собой набор значений атрибутов сущности.

¹ данный тип введен для устранения зацикливания загрузки объектов по связям графа.

Получение данных из хранилища объектов

Сервис Omni-API возвращает данные в виде иерархической структуры, которая соответствует части графа, полученной путем загрузки по связям от корневого узла (сущности). Получение данных выполняется по связям модели, если есть ссылка на следующую сущность, то происходит загрузка соответствующих объектов и так до тех пор, пока все объекты по связям не загрузятся.

Язык запросов Omni LHS Brackets

Для выборки требуемых данных на основе LHS Brackets разработан специальный язык запросов.

Выборка определяется: корневой сущностью, набором фильтров, параметром сортировки, параметрами группировки, параметрами пагинации.

Фильтры применяются только для поиска объектов корневой сущности, фильтрация вложенных сущностей не предусмотрена.

Загрузка выполняется от объектов корневой сущности к следующим в порядке вложенности объектам.

При загрузке объектов вложенных сущностей используются значения (идентификаторы) предыдущих уровней для поиска объектов вложенных сущностей.

Общий вид запроса:

<операторы_фильтрации_запроса>

&orderBy=<список_выражений_сортировки>

&groupBy=<список_атрибутов_группировки>

&groupOffset=<смещение_в_группе>

&groupLimit=<лимит_в_группе>

&offset=<смещение>

&limit=<лимит>

Общий вид запроса с использованием параметра query²

query=<операторы_фильтрации_запроса>

&orderBy=<список_выражений_сортировки>

&groupBy=<список_атрибутов_группировки>

&groupOffset=<смещение_в_группе>

&groupLimit=<лимит_в_группе>

&offset=<смещение>

&limit=<лимит>

² параметр query введен для удобства запуска запросов из SwaggerUI, в штатном режиме предполагается передача каждого оператора фильтрации как отдельного параметра.

Операторы фильтрации

Оператор фильтрации предназначен для ограничения выборки данных путем указания допустимых значений атрибута корневой сущности и атрибутов вложенных сущностей соответственно графу модели данных.

Пример оператора фильтрации для атрибута корневой сущности:

```
serialNumber=1234567890
```

Пример оператора фильтрации для атрибута дочерней сущности:

```
meterLink.guid=3fa85f64-5717-4562-b3fc-2c963f66afa6
```

Значения оператора фильтрации указываются справа от знака «=», это может быть единственное значение

```
serialNumber=1234567890
```

либо список, перечисленный через разделитель «,» или «|»

```
serialNumber[in]=1234567890,1234567891,1234567892
```

```
serialNumber[in]=1234567890|1234567891|1234567892
```

Операции фильтрации

Список допустимых операций:

eq – равно;

ne – не равно;

gt – больше;

gte – больше или равно;

lt – меньше;

lte – меньше или равно;

in – принадлежит множеству;

nin – не принадлежит множеству;

empty – пустое значение;

exist – не пустое значение.

Если операция не указана, то применяется операция равенства (eq).

Операция фильтрации указывается в квадратных скобках после перечисления атрибутов:

```
serialNumber[in]=1234567890,987654321,123459876
```

```
meterLink.guid[eq]=3fa85f64-5717-4562-b3fc-2c963f66afa6
```

Операции **empty** и **exist** указываются без значений:

```
serialNumber[exist]
```

Критерии отбора операций фильтрации

Для атрибутов-массивов в дополнение к операциям фильтрации существует возможность указания критерия отбора³:

all – операция фильтрации выполняется для всех элементов значения атрибута;

any – операция фильтрации выполняется хотя бы для одного элемента значения атрибута.

Критерий отбора указывается через дефис перед операцией фильтрации, по умолчанию используется **any**.

Пример, в массиве `arrayValues` есть значения, которые больше 123.45:

`arrayValues[any-gt]=123.45` , то же самое

`arrayValues[gt]=123.45`

Пример, все значения массива `arrayValues` больше 123.45:

`arrayValues[all-gt]=123.45`

Критерии отбора могут быть применены не только для последнего атрибута в операторе фильтрации, но и для предшествующих атрибутов, этом случае в квадратных скобках указывается только критерий отбора, код операции фильтрации для предшествующих атрибутов указывать нельзя.

Данный запрос для сущности `UsagePoint` вернет только те Топ, у которых **все** ТУ имеют значение атрибута `connectISUE` равное «ПУ интеллектуальный подключен к ИСУЭ»:

`arrayAccountPointLink[all].connectISUE=ПУ интеллектуальный подключен к ИСУЭ`

Параметры управления загрузкой объектов (пагинации)

Для указания части выборки, которую нужно загрузить, используются параметры пагинации:

offset – отступ от начала выборки;

limit – количество загружаемых объектов.

Параметры группировки

Параметры группировки позволяют разделить множество объектов выборки на группы и применить к каждой группе групповые параметры управления загрузкой.

Параметры группировки:

groupBy – список атрибутов, по которым следует выполнять группировку;

groupOffset – отступ от начала в группе;

groupLimit – количество загружаемых объектов в группе.

³ допустимо использовать для всех атрибутов, но на результат выполнения запроса с атрибутами не являющимися массивами критерий отбора не повлияет.

Работа с API при помощи встроенного SwaggerUI

SwaggerUI предоставляет собой фреймворк с открытым исходным кодом, который читает сформированную в соответствии с метаданными Omni-API спецификацию OpenAPI и создает веб-страницу с интерактивной документацией.

Swagger
Select a definition: Omni-API v1

Omni-API 20.16.8 OAS3
<http://localhost:27032/swagger/1/omniapi.json>

Authorize

Dictionary

GET /api/dictionaries Справочники

AccessList890

GET /api/access-lists/search-operations Поиск операций в системном журнале

POST /api/access-lists/get-objects-batch Пакетное извлечение объектов по идентификаторам

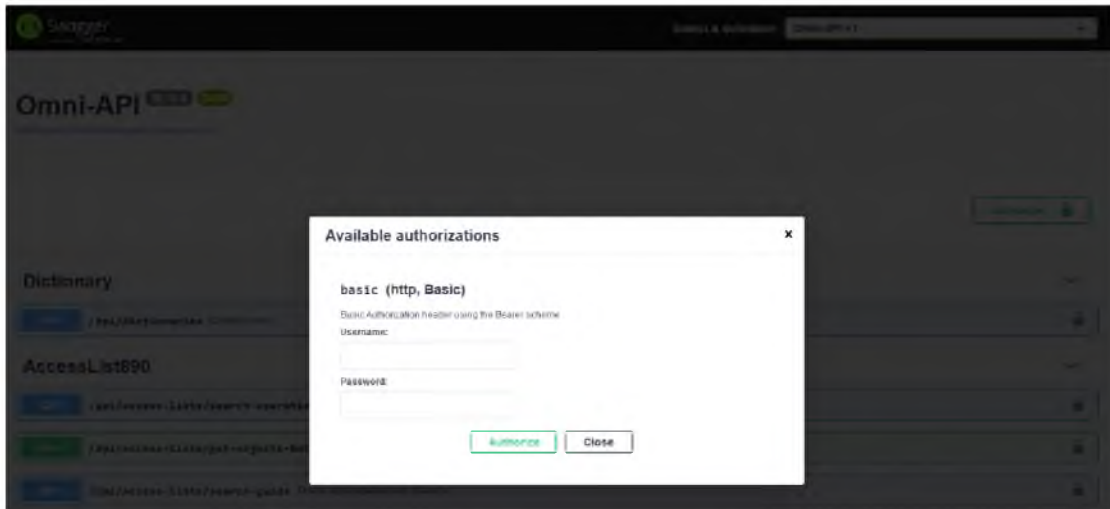
GET /api/access-lists/search-guids Поиск идентификаторов объектов

GET /api/access-lists/search Поиск объектов

POST /api/access-lists/search Расширенный поиск объектов

Авторизация

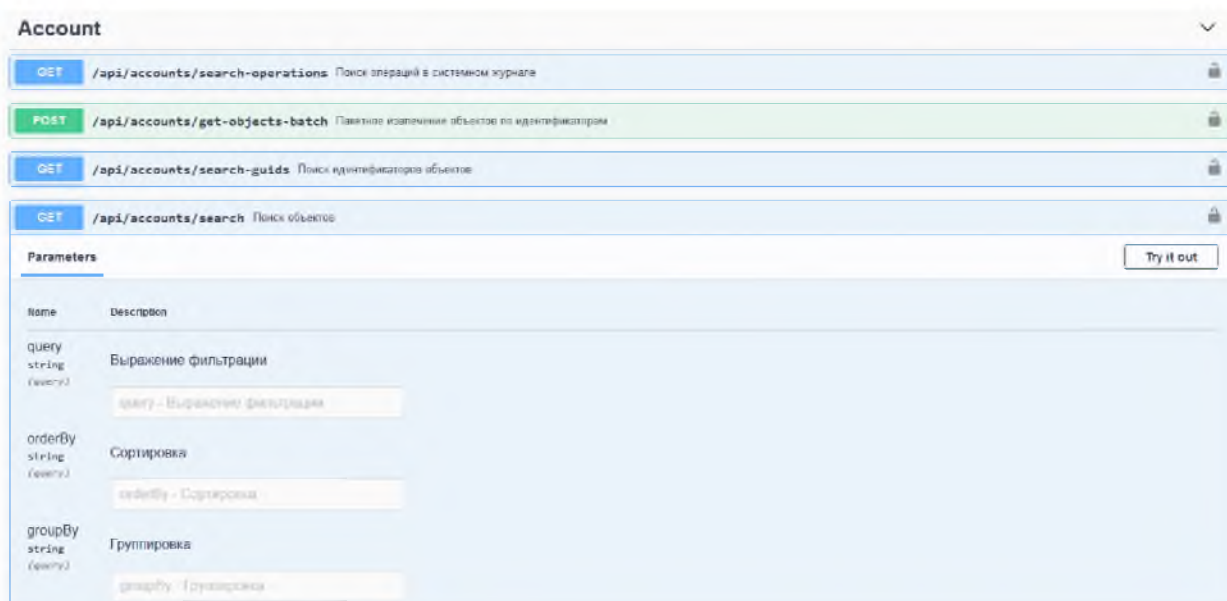
Настройки basic-авторизации заполняются в форме, которая отображается при помощи кнопки **Authorize**



После ввода имени пользователя и пароля необходимо нажать кнопку **Authorize**, после чего введенные данные будут применяться ко всем запросам, выполняемым из SwaggerUI.

Выполнение запроса

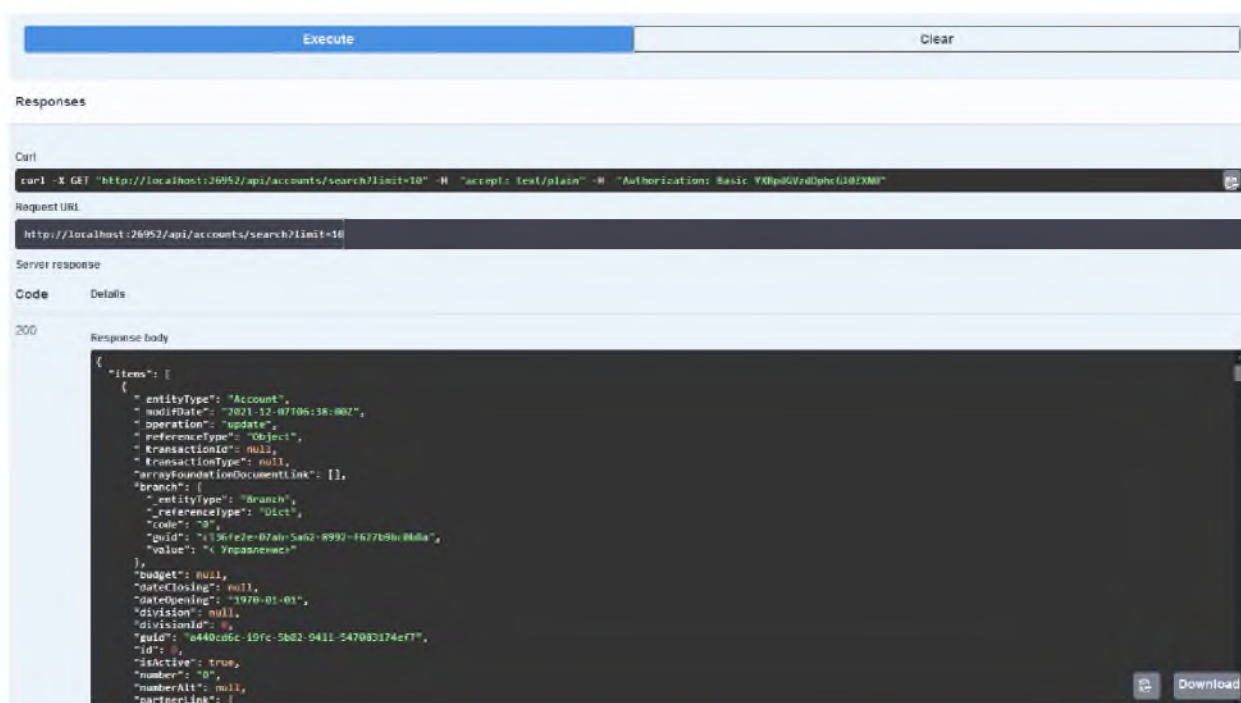
Для выполнения запроса необходимо в разделе нужной сущности развернуть соответствующий пункт конечной точки и нажать кнопку **Try it out**



The screenshot shows the Swagger UI for the 'Account' entity. The 'GET /api/accounts/search' endpoint is selected. The 'Parameters' section is expanded, showing input fields for 'query', 'orderBy', and 'groupBy'. The 'Try it out' button is visible in the top right corner of the parameters section.

После этого поля формы запроса станут доступными для редактирования, в них нужно ввести значения запроса.

После того как поля запроса заполнены нужно нажать кнопку **Execute**, запрос будет отправлен на выполнение. После выполнения запроса и получения ответа, результат отобразится в разделе **Responses**.



The screenshot shows the Swagger UI with the 'Execute' button highlighted in blue. The 'Responses' section is expanded, showing the response body for the 'GET /api/accounts/search' endpoint. The response is a JSON object containing an array of account details.

```
curl -X GET "http://localhost:26952/api/accounts/search?limit=10" -H "accept: text/plain" -H "Authorization: Basic YWp0Y2VudjphcG10ZXMi"
Request URL
http://localhost:26952/api/accounts/search?limit=10
Server response
Code Details
200
Response body
{
  "items": [
    {
      "entityType": "Account",
      "modifDate": "2021-12-07T06:38:00Z",
      "operation": "update",
      "referenceType": "Object",
      "transactionId": null,
      "transactionType": null,
      "arrayFoundationDocumentLink": [],
      "branch": [
        {
          "entityType": "Branch",
          "referenceType": "Dict",
          "code": "0",
          "guid": "1136f7e0-07ab-5a62-8992-1677b9b044a",
          "value": "Учреждение"
        }
      ],
      "budget": null,
      "dateClosing": null,
      "dateOpening": "1970-01-01",
      "division": null,
      "divisionId": 0,
      "guid": "a440c6c-19fc-5b02-9411-547083174e77",
      "id": 0,
      "isActive": true,
      "number": "0",
      "numberAlt": null,
      "partnerLink": []
    }
  ]
}
```