



Айтида iMark ОПИСАНИЕ ОТКРЫТОГО API

POST /passwordauth – Авторизация по паролю.

Описание. **POST** запрос.

Запрос авторизации по паролю или коду авторизации пользователя. Для **POST** варианта запроса, в теле передается **md5** от пароля или кода авторизации.

Примеры для **POST** запроса.

```
{"password":"0x202CB962AC59075B964B07152D234B70"}
```

или

```
{"authcode":"0x202CB962AC59075B964B07152D234B70"}
```

Ответ сервера.

В случае обнаружения пользователя с указанным паролем или кодом авторизации возвращается ответ вида

```
{"expired":1708973625,"id":"000001","name":"Администратор","role":"administrator","signature":"a8706393e7304b8abd9e200ecccc35b"}
```

expired	– срок действия токена до указанного времени в секундах с 01.01.1970
id	– код пользователя
name	– имя пользователя
role	– роль пользователя (administrator, merchant, cashier, pos)
signature	– уникальная последовательность.

Base64 от полученного текста необходимо использовать для последующей авторизации запросов.

Authorization: Bearer token

Authorization: Bearer eyJleHBpcmVkljoxNzA4... fQ==

В случае отсутствия пользователя с указанным паролем / кодом авторизации будет возвращена ошибка **401 Unauthorized**

```
{"error":"invalid_password","message":""}
```

GET /passwordauth – Авторизация по паролю.

Описание. **GET** запрос.

Запрос авторизации по паролю или коду авторизации пользователя. Для этого варианта **md5** от пароля или кода авторизации в формате **base64** передается в заголовке Authorization: Direct

Пример **GET** запроса

```
Authorization: Direct eyJleHBpcmVkljoxNzA4...ifQ==
```

Ответ сервера.

В случае обнаружения пользователя с указанным паролем или кодом авторизации возвращается ответ вида

```
{"expired":1708973625,"id":"000001","name":"Администратор","role":"administrator","signature":"a8706393e7304b8abd9e200ecccc35b"}
```

expired – срок действия токена до указанного времени в секундах с 01.01.1970

Id – код пользователя

name – имя пользователя

role – роль пользователя (administrator, merchant, cashier, pos)

signature – уникальная последовательность.

Base64 от полученного текста необходимо использовать для последующей авторизации запросов.

```
Authorization: Bearer token
```

```
Authorization: Bearer eyJleHBpcmVkljoxNzA4... fQ==
```

В случае отсутствия пользователя с указанным паролем / кодом авторизации будет возвращена ошибка **401 Unauthorized**

```
{"error":"invalid_password","message":""}
```

GET /token – Авторизация по коду пользователя и паролю.

Описание. **GET** запрос.

Запрос авторизации по идентификатору пользователя и паролю.

```
{"id": "<id>", "password": "<hash>"}
```

id – код пользователя
password – md5 от строки <id>:<password>

Указанный текст в формате json преобразуется в кодировку base64 и передается в заголовке Authorization: Direct

Пример.

```
Authorization: Direct eyJleHBpcmVkljoxNzA4...ifQ==
```

Ответ сервера.

В случае обнаружения пользователя с указанными кодом и паролем возвращается ответ вида

```
{"expired":1708973625,"id":"000001","name":"Администратор","role":"administrator","signature":"a8706393e7304b8abd9e200eccc35b"}
```

expired – срок действия токена до указанного в секундах времени
id – код пользователя
name – имя пользователя
role – роль пользователя (administrator, merchant, cashier, pos)
signature – уникальная последовательность.

Base64 от полученного текста необходимо использовать для последующей авторизации запросов.

```
Authorization: Bearer token
```

```
Authorization: Bearer eyJleHBpcmVkljoxNzA4... fQ==
```

В случае отсутствия пользователя с указанным паролем / кодом авторизации будет возвращена ошибка **401 Unauthorized**

```
{"error":"invalid_password","message":""}
```

GET /token – Продление авторизации.

Описание. **GET** запрос.

Продление срока действия токена авторизации.

В заголовке `Authorization: Bearer` передается полученный ранее токен. Срок действия этого токена продлевается ещё на один час.

Пример.

`Authorization: Bearer eyJleHBpcmVkljoxNzA4...ifQ==`

Ответ сервера.

В случае, если токен действует и его срок ещё не истек, то действие такого токена продлевается на один час. В ответе передается обновленная структура json с новым сроком действия.

```
{"expired":1708977225,"id":"000001","name":"Администратор","role":"administrator","signature":"a8706393e7304b8abd9e200ecccc35b"}
```

expired	– срок действия токена
id	– код пользователя
name	– имя пользователя
role	– роль пользователя (administrator, merchant, cashier, pos)
signature	– уникальная последовательность.

Base64 от полученного текста необходимо использовать для последующей авторизации запросов.

`Authorization: Bearer token`

`Authorization: Bearer eyJleHBpcmVkljoxNzA4... fQ==`

В случае отсутствия переданного токена, либо если срок его действия уже истек будет возвращена ошибка **401 Unauthorized**

```
{"error":"invalid_password","message":""}
```

PUT / truemark – Установка токена и адреса сервера ГИС МТ.

Описание. **PUT** запрос.

Запрос предназначен для установки токена и адреса сервера для доступа к ГИС МТ. Запрос передается в виде json структуры вида

```
{"X-RegistrationKey":"ТокенПолученныйВГИСМТ","host":"АдресСервераГИСМТ"}
```

Обязательным заголовком запроса является заголовок авторизации на сервере imark

Authorization: Bearer token

Выполнение запроса доступно для пользователей с ролью Администратор. Передаваемые значения записываются в справочник организаций в строку, отмеченную как организация по умолчанию. Если такой организации нет, а в справочнике есть всего одна организация, то данные записываются в неё.

В случае, если организаций несколько и ни одна не отмечена как организация по умолчанию, или в справочнике нет ни одной организации, то возвращается сообщение об ошибке 404 Not found.

```
{"error":"organisation_not_found","message":"Не найдена организация, для записи токена."}
```

В запросе может быть не передан любой из параметров, в том числе и оба параметра могут быть пустые, либо отсутствовать. В этом случае, у найденной организации будет заполнен только не пустой присутствующий в запросе параметр.

Ответ сервера.

В случае успешного обновления справочника организаций будет возвращены обновленные , либо текущие параметры в виде

```
{"X-RegistrationKey":"Обновленное или текущее значение Токена ГИС МТ","host":"АдресСервераГИСМТ"}
```

В случае возникновения каких-либо ошибок чтения или записи данных в базу данных будет возвращен результат 500 Internal server error и расшифровка возникшей проблемы

```
{"error":"organisations_read_failed","message":"Error executing query... "}
```

GET /codes/offline – Получение списка кодов маркировки.

Описание. **GET** запрос.

Запрос позволяет получить список кодов маркировки, находящихся в базе данных imark.

Формат строки запроса:

```
/codes/offline?start="2023.03.05 12:00:00"&end="2023.03.05 14:00:00"&inn=1234567890&kpp=123456789
```

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для всех ролей пользователей кроме роли **pos**.

Authorization: Bearer token

Параметры запроса.

start - строка. Не используется.

end - строка. Не используется

inn - строка. ИНН организации. Формат: только цифры, не более 12 символов.

Kpp - строка. КПП организации. Формат: 9 цифр.

fiasId - строка. Идентификатор ФИАС. Формат: UUID.

Ответ на запрос.

В случае успеха возвращается ZIP-файл с выгрузкой в формате csv.

В случае возникновения каких-либо ошибок чтения или записи данных в базу данных будет возвращен результат 500 Internal server error и расшифровка возникшей проблемы

```
{"error":"organisations_read_failed","message":"Error executing query... "}
```

POST /products – Загрузка справочника товаров.

Описание. **POST** запрос.

Запрос позволяет загрузить в базу данных imark справочник товаров. Товары, существующие в справочнике товаров imark, но отсутствующие в теле запроса, будут удалены из справочника.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для пользователей **Администратор** и **Товаровед**.

```
{
  "products":
  [
    {
      "id": "000001",
      "name": "Тпки домашние",
      "markgroup": 5,
      "barcodes": ["4601234567890", "4600123456789"]
    },
    {
      "id": "000006",
      "name": "Тапки дикие",
      "markgroup": 5,
      "barcodes": []
    }
  ]
}
```

Описание полей.

products

- список объектов товаров;

id

- строка идентификатор (код) товара;

name

- строка наименование товара;

barcodes

- список строк, список штриховых кодов товара;

markgroup

- число целое, группа маркировки товара.


```

        "available_per_package": "2",
        "total_per_package": "3",
        "mark_status": "1",
        "comment": "Сигареты не GS-1",
        "item_type": "12"
    }
}
}

```

Описание полей ответа на **POST** запрос для получения списка марок с заданными параметрами

count	- число целое, количество выводимых марок;
data	- массив объектов марок и транзакций;
numbers	- строка, марка в формате base64;
transactions	- объект транзакции, информация о последней транзакции;
state	- строка, состояние марки в imark: <ul style="list-style-type: none"> • lock - блокируется / заблокирована; • unlock - разблокируется / доступна;
stamp	- дата и время кассового документа;
document	- строка, номер кассового документа;
pos	- строка, идентификатор кассы;
shift	- число целое, номер смены;
user	- строка. идентификатор пользователя;
note	- строка, текст комментария;
available_per_package	- число целое, доступное количество товара в упаковке;
total_per_package	- число целое, общее количество товара в упаковке;
mark_status	- число целое, статус марки в imark: <ul style="list-style-type: none"> • 0 — доступна; • 1 — доступна частично; • 2 — заблокирована.
comment	- строка, текст комментария;
item_type	- число целое, тип номенклатуры маркированной продукции.

POST /create_stamp – Создание марки.

Описание. **POST** запрос для создания записи о марке в базе данных imark. Используется для заполнения полей у марки, имеющейся в БД, если какие-то из полей не заполнены.

```
{
  "numbers": ["AAA", "BBB", "CCC"]
}
```

numbers — список строк, массив кодов маркировки в формате Base64.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для всех ролей пользователей кроме роли **pos**.

В запросе передаются только марки (и только существующие в БД). Если обнаруживается, что у переданной марки имеются пустые поля в БД, эти пропуски заполняются стандартными значениями:

- mark_status — available;
- item_type — other;

Если список транзакций пуст, создается стандартная транзакция:

- state — unlock;
- action — commit;
- status — 0 (доступна).

Ответ на запрос.

```
{
  "numbers": ["BBB"]
}
```

Массив содержит КМ, которые отсутствуют в БД.

GET unique_product_stamp/<код идентификации> – Получение информации о транзакциях марки.

Описание. **GET** запрос для получения информации о транзакциях указанной марки

Запрос передается с пустым телом, марка в формате base64 добавляется в URL.

Например:

GET http://localhost:8000/unique_product_stamp/NzA3MTc2ODM0NzM1MzdVb3VaMTF

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для всех ролей пользователей кроме роли **pos**.

Ответ на запрос возвращается в виде json с найденной маркой и массивом ее транзакций.

```
{
  "count": 1,
  "data":
  [
    {
      "number": "00000046200068EoxL'2&AQCy0Qf+",
      "transactions":
      [
        {
          "state": "unlock",
          "action": "commit",
          "stamp": "2023-04-20T12:01:38",
          "document": "",
          "pos": "",
          "shift": "",
          "user": "",
          "note": ""
        },
        {
          "state": "lock",
          "action": "begin",
          "stamp": "2023-07-18T19:10:19",
          "document": "",
          "pos": "",
          "shift": "",
          "user": "",
          "note": "update unique product stamps"
        },
        {
          "state": "lock",
          "action": "commit",
          "stamp": "2023-07-18T19:10:19",
          "document": "",
          "pos": "",
          "shift": "",
          "user": "",
          "note": "update unique product stamps"
        }
      ],
      "available_per_package": "2",
      "total_per_package": "3",
      "mark_status": "1",
      "comment": "Сигареты не GS1 --gs",
      "item_type": "12"
    }
  ]
}
```

Описание полей ответа на запрос.

count	- число, количество возвращаемых элементов массива data. Если марка найдена в базе данных, то равно — 1. Иначе 0.
data	- список объектов. Данные о транзакциях и состоянии запрошенного КМ
number	- строка КМ, для которого запрашиваются транзакции, в формате base64
transactions	- список объектов. Транзакции переданной в запросе марки
state	- число целое. строка Состояние марки в FMU: <ul style="list-style-type: none">• lock — блокируется / заблокирована;• unlock — разблокируется / доступна.
action	- строка, действие по изменению состояния акцизной марки в FMU: <ul style="list-style-type: none">• begin — начать;• commit — завершить;• rollback — отменить;• horse.
pos	- строка, идентификатор кассы
shift	- число целое, номер смены
document	- строка, номер чека
user	- строка, идентификатор пользователя

POST /unique_product_stamp – Добавление списка марок.

Описание. **POST** запрос для добавления марки или марок в базу данных imark.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для ролей пользователей Администратор и Товаровед.

Тело запроса должно содержать список кодов маркировки и транзакцию, которая будет добавлена ко всем указанным КМ.

```
{
  "numbers": ["AAA", "BBB", ...],
  "transaction":
  {
    "state": "unlock",
    "action": "horse",
    "pos": "",
    "shift": "",
    "document": "",
    "user": "",
  },
  "available_per_packages": ["2", "", ...],
  "total_per_packages": ["3", "", ...],
  "mark_statuses": ["1", "0", ...],
  "comments": ["some text", "more text"],
  "item_types": ["5", "7", ...]
}
```

Указанная транзакция будет добавлена ко всем переданным маркам. Поэтому, если для марки передается статус «2» (заблокирована), то значение поля state программными средствами подменяется на lock+commit, чтобы марка записалась в БД со статусом «заблокирована».

Описание полей запроса.

numbers - массив строк, коды маркировки в формате base64;
transaction - массив объектов, данные о транзакции
state - строка, состояние марки в imark:
action - строка, выполняемое действие по изменению состояния акцизной марки:
pos - строка, идентификатор кассы
shift - число целое, номер смены
document - строка, номер чека;
user - строка. идентификатор пользователя.
available_per_packages - число целое, общее количество товаров в упаковке
total_per_packages - число целое, доступное количество товаров в упаковке
mark_statuses - массив целых чисел, статусы КМ в imark:
• 0 — доступна;
• 1 — доступна частично;
• 2 — заблокирована
comment - строка, текст комментария
item_types - массив целых чисел, типов маркировки

Если “status” — 2, то в запросе значение “state” автоматически подменяется с “unlock” на “lock+commit”.

Ответ на запрос.

В случае успеха возвращается пустой массив, в случае ошибки — текст с ошибкой.

PUT /unique_product_stamp – Изменение данных о КМ.

Описание. **PUT** запрос для изменения данных о КМ в базе данных imark.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для ролей пользователей Администратор и Товаровед.

Тело запроса должно содержать код маркировки и транзакцию, которая будет добавлена, для изменения состояния КМ.

```
{
  "numbers": "MDEwNDY1NzA2MzQ3MzUzNzlx3BQMDJlIzI5IDkzcURkZiMyOSAxNzlxMDYyOQ==",
  "transaction":
  {
    "state": "lock",
    "action": "begin",
    "pos": "",
    "shift": "",
    "document": "",
    "user": "",
    "note": "comment text"
  }
}
```

Описание полей запроса.

numbers - строка, КМ в формате base64;
transaction - объект, данные о транзакции;
state - строка, состояние марки:
• lock - блокируется / заблокирована;
• unlock - разблокируется / доступна;
action - строка, действие по изменению состояния КМ:
• begin - начать;
• commit - завершить;
• rollback - отменить;
• horse;
pos - строка, идентификатор кассы;
shift - число целое, номер смены;
document - строка, номер чека;
user - строка, идентификатор пользователя;
note - строка, текст комментария.

Ответ на запрос.

Возвращается пустой json. Если КМ не найден в базе данных imark, то он возвращается в теле ответа. Если в запросе не указан КМ, то возвращается ошибка 400.

DELETE /unique_product_stamp/<код идентификации> – Удаление данных КМ.

Описание. **DELETE** запрос для удаления данных о КМ из базы данных imark.

Тело запроса пустое. Марка в формате base64 добавляется в URL.

Например:

```
DELETE http://localhost:8000/unique_product_stamp/NzA3MTc2ODM0NzM1MzdVb3BaMTF
```

Ответ на запрос.

В случае успеха, т.е. КМ удалён из базы данных imark, возвращается пустой json.

GET /organisations – Получение списка организаций.

Описание. **GET** запрос для получения информации о списке организаций, зарегистрированных в базе данных imark.

Запрос передается с пустым телом.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для всех ролей пользователей.

Ответ на запрос возвращается в виде json со списком кодов организаций.

```
{
  "count": 1,
  "readonly": false,
  "data": [ "000001", "000002", "000003" ]
}
```

Описание полей ответа на запрос.

count	- число, количество возвращаемых элементов массива data.
readonly	- логическое, Истина, если авторизованному пользователю доступно изменение, иначе Ложь.
data	- список кодов организаций

GET /organisation/<код организации> – Получение информации о организации.

Описание. **GET** запрос для получения информации о конкретной организации в базе данных imark.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для всех ролей пользователей.

Ответ на запрос возвращается в виде json с информацией об организации.

```
{
  "readonly": false,
  "data":
  {
    "code": "000001",
    "name": "Организация",
    "token": "ГИС МТ токен",
    "host": "127.0.0.1",
    "bydef": true,
    "modified": "2024-02-29T12:00:00"
  }
}
```

Описание полей ответа на запрос.

readonly	- логическое, Истина, если авторизованному пользователю доступно изменение, иначе Ложь.
data	- объект, с описанием организации.
code	- строка, код организации;
name	- строка, наименование организации;
token	- строка, токен, для отправки / получения данных в ГИС МТ;
host	- строка, адрес сервиса ГИС МТ, для отправки запросов;
bydef	- логическое, если Истина, то организация "по умолчанию";
modified	- дата и время последнего изменения записи.

PUT /organization – Запись информации об организации.

Описание. **PUT** запрос для записи информации о конкретной организации в базу данных imark.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для пользователей с ролью администратора или товароведа.

Тело запроса передается в виде объекта json с информацией об организации.

```
{
  "code": "000001",
  "name": "Организация",
  "token": "ГИС МТ токен",
  "host": "127.0.0.1",
  "bydef": true,
}
```

Описание полей запроса.

code	- строка, код организации; если передано пустое значение, то будет добавлена новая организация;
name	- строка, наименование организации;
token	- строка, токен, для отправки / получения данных в ГИС МТ;
host	- строка, адрес сервиса ГИС МТ, для отправки запросов;
bydef	- логическое, если Истина, то организация "по умолчанию";

В ответ на запрос возвращает записанная в базу данных imark информация об организации в виде:

```
{
  "readonly": false,
  "data":
  {
    "code": "000001",
    "name": "Организация",
    "token": "ГИС МТ токен",
    "host": "127.0.0.1",
    "bydef": true,
    "modified": "2024-02-29T12:00:00"
  }
}
```

Описание полей ответа на запрос.

readonly	- логическое, Истина, если авторизованному пользователю доступно изменение, иначе Ложь.
data	- объект, с описанием организации.
code	- строка, код организации; если была добавлена новая организация, то её новый код;
name	- строка, наименование организации;
token	- строка, токен, для отправки / получения данных в ГИС МТ;
host	- строка, адрес сервиса ГИС МТ, для отправки запросов;
bydef	- логическое, если Истина, то организация "по умолчанию";
modified	- дата и время последнего изменения записи.

DELETE /organisation/<код организации> – Удаление организации.

Описание. **DELETE** запрос для удаления записи об о конкретной организации из базы данных imark.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для пользователей с ролью администратора.

Ответ на запрос возвращается в виде json с информацией о операции.

```
{
  "operationResult": "success"
}
```

GET /users – Получение списка пользователей.

Описание. **GET** запрос для получения информации о списке пользователей, зарегистрированных в базе данных imark.

Запрос передается с пустым телом.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для всех пользователей с ролью администратора.

Ответ на запрос возвращается в виде json со списком кодов пользователей.

```
{
  "count": 1,
  "readonly": false,
  "data": [ "000001", "000002", "000003" ]
}
```

Описание полей ответа на запрос.

count	- число, количество возвращаемых элементов массива data.
readonly	- логическое, Истина, если авторизованному пользователю доступно изменение, иначе Ложь.
data	- список кодов пользователей

GET /user/<код пользователя> – Получение информации о пользователе.

Описание. **GET** запрос для получения информации о конкретном пользователе в базе данных imark.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для всех ролей пользователей. Для администраторов возвращается информация о запрошенном пользователе. Для остальных ролей информация об самом авторизованном пользователе. Изменение данных доступно для администраторов и товароведов.

Ответ на запрос возвращается в виде json с информацией о пользователе.

```
{
  "readonly": false,
  "data":
  {
    "code": "000001",
    "name": "Пользователь 1",
    "authcode": "Код авторизации",
    "role": 1,
    "modified": "2024-02-29T12:00:00"
  }
}
```

Описание полей ответа на запрос.

readonly	- логическое, Истина, если авторизованному пользователю доступно изменение, иначе Ложь.
data	- объект, с описанием пользователя.
code	- строка, код пользователя;
name	- строка, имя пользователя;
role	- число, роль пользователя; <ul style="list-style-type: none">▪ Администратор - 0,▪ Товаровед - 1,▪ Кассир - 2,▪ Касса - 3
authcode	- строка, код авторизации пользователя;
modified	- дата и время последнего изменения записи.

PUT /user – Запись информации о пользователе.

Описание. **PUT** запрос для записи информации о конкретном пользователе в базу данных imark.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для пользователей с ролью администратора или товароведа.

Тело запроса передается в виде объекта json с информацией о пользователе.

```
{
  "code": "000001",
  "name": "Пользователь 1",
  "authcode": "Код авторизации",
  "password": "Пароль",
  "role": 1,
}
```

Описание полей запроса.

code	- строка, код пользователя; если передано пустое значение, то будет добавлен новый пользователь;
name	- строка, имя пользователя;
authcode	- строка, код авторизации;
password	- строка, пароль пользователя;
role	- число, роль пользователя;
	▪ Администратор - 0,
	▪ Товаровед - 1,
	▪ Кассир - 2,
	▪ Касса - 3

В ответ на запрос возвращается записанная в базу данных imark информация о пользователе в виде:

```
{
  "readonly": false,
  "data": {
    "code": "000001",
    "name": "Пользователь 1",
    "authcode": "Код авторизации",
    "role": 1,
    "modified": "2024-02-29T12:00:00"
  }
}
```

Описание полей ответа на запрос.

readonly	- логическое, Истина, если авторизованному пользователю доступно изменение, иначе Ложь.
data	- объект, с описанием организации.
code	- строка, код организации;
name	- строка, наименование организации;
role	- число, роль пользователя;
	▪ Администратор - 0,
	▪ Товаровед - 1,
	▪ Кассир - 2,
	▪ Касса - 3
authcode	- строка, код авторизации пользователя;
modified	- дата и время последнего изменения записи.

DELETE /user/<код пользователя> – Удаление пользователя.

Описание. **DELETE** запрос для удаления записи об о конкретном пользователе из базы данных imark.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для пользователей с ролью администратора.

Ответ на запрос возвращается в виде json с информацией о операции.

```
{
  "operationResult": "success"
}
```


GET /mthosts – Получение списка CDN площадок

Описание. **GET** запрос для получения информации о списке адресов доступа к сервисам ГИС МТ, зарегистрированных в базе данных imark.

Запрос передается с пустым телом.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для всех пользователей с ролью администратора.

Ответ на запрос возвращается в виде json со списком кодов адресов.

```
{
  "count": 1,
  "readonly": false,
  "data": [ "000001", "000002", "000003" ]
}
```

Описание полей ответа на запрос.

count	- число, количество возвращаемых элементов массива data.
readonly	- логическое, Истина, если авторизованному пользователю доступно изменение, иначе Ложь.
data	- список кодов записей с адресами доступа к ГИС МТ.

GET /mthost/<код адреса> – Получение информации о CDN площадке.

Описание. **GET** запрос для получения информации о конкретном адресе ГИС МИ в базе данных imark.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для всех ролей пользователей.

Ответ на запрос возвращается в виде json с информацией об адресе.

```
{
  "readonly": false,
  "data":
  {
    "code": "000001",
    "host": "Адрес и порт доступа",
    "note": "Комментарий ",
    "duration": 1,
    "bydef": true,
    "modified": "2024-02-29T12:00:00"
  }
}
```

Описание полей ответа на запрос.

readonly	- логическое, Истина, если авторизованному пользователю доступно изменение, иначе Ложь.
data	- объект, с описанием адреса доступа.
code	- строка, код записи в базе данных imark;
host	- строка, адрес и порт для доступа к сервисам ГИС МТ;
duration	- число, время последнего ответа сервиса ГИС МТ по этому адресу в миллисекундах;
byref	- выбран ли этот адрес, как адрес по умолчанию для выполнения проверки документов
modified	- дата и время последнего изменения записи.

PUT /mthost – Запись информации о площадке.

Описание. **PUT** запрос для записи информации о конкретном адресе доступа к сервисам ГИС МТ в базу данных imark.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для пользователей с ролью администратора.

Тело запроса передается в виде объекта json с информацией о адресе ГИС МТ.

```
{
  "code": "000001",
  "host": "Адрес и порт доступа",
  "note": "Комментарий ",
  "bydef": true,
}
```

Описание полей запроса.

code	- строка, код адреса; если передано пустое значение, то будет добавлен новый адрес;
host	- строка, адрес сервиса ГИС МТ;
note	- строка, комментарий;
bydef	- логическое, нужно ли начать использовать адрес, как адрес по умолчанию для проверки чеков.

В ответ на запрос возвращается записанная в базу данных imark информация о пользователе в виде:

```
{
  "readonly": false,
  "data":
  {
    "code": "000001",
    "host": "Адрес и порт доступа",
    "note": "Комментарий ",
    "duration": 1,
    "bydef": true,
    "modified": "2024-02-29T12:00:00"
  }
}
```

Описание полей ответа на запрос.

readonly	- логическое, Истина, если авторизованному пользователю доступно изменение, иначе Ложь.
data	- объект, с описанием адреса доступа.
code	- строка, код записи в базе данных imark;
host	- строка, адрес и порт для доступа к сервисам ГИС МТ;
duration	- число, время последнего ответа сервиса ГИС МТ по этому адресу в миллисекундах;
byref	- выбран ли этот адрес, как адрес по умолчанию для выполнения проверки документов
modified	- дата и время последнего изменения записи.

DELETE /mthost/<код адреса> – Удаление информации о площадке.

Описание. **DELETE** запрос для удаления записи об о конкретном адресе службы ГСИ МТ из базы данных imark.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для пользователей с ролью администратора.

Ответ на запрос возвращается в виде json с информацией о операции.

```
{  
  "operationResult": "success"  
}
```

GET /settings – Получение настроек.

Описание. **GET** запрос для получения информации о настройках службы imark.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для всех ролей пользователей.

Ответ на запрос возвращается в виде json с информацией о настройках.

```
{
  "readonly": false,
  "data":
  {
    "dbtype": "Тип SQL сервера",
    "version": "Номер Версии",
    "driver": "Имя драйвера SQL сервера",
    "host": "адрес SQL сервера",
    "port": "Порт доступа",
    "user": "Имя Пользователя",
    "database": "Имя базы",
    "backuppath": "z:\imark\backups\backup[N].bak",
    "backuplocalpath": "d:\imark\backups ",
    "backupnumber": 1,
    "backupcount": 5,
    "backupdate": "2024-02-01T12:00:00",
    "backupinterval": "Интервал",
    "logfile": "logfilename"
    "imarkdll": " imarkdll "
    "cdninterval": "Интервал Опроса Серверов ГИС МТ"
  }
}
```

Описание полей ответа на запрос.

readonly	- логическое, Истина, если авторизованному пользователю доступно изменение, иначе Ложь.
version	- число, номер версии формата параметров;
data	- объект, с описанием хранилища базы данных.
dbtype	- строка, тип SQL сервера; допустимые значения <ul style="list-style-type: none">• SQLite - использовать встроенный вариант SQLite DB;• MSSQL - использовать MS SQL сервер
host	- строка, адрес для доступа к серверу базы данных; для SQLite не используется;
port	- число, порт для доступа к серверу базы данных; для SQLite не используется;
user	- имя пользователя для доступа к серверу базы данных; для SQLite не используется;
database	- имя базы данных на сервере; для SQLite полное имя файла с базой данных.
backuppath	- строка, полный путь к папке и имени файла для сохранения резервной копии базы данных; В имени файла и / или папки необходимо указать место для вставки номера резервной копии в виде [N]. Если указан сетевой путь, то служба imark должна иметь права на запись файлов по указанному пути.
backuplocalpath	- строка, полный путь к папке, в которой будет создан файл резервной копии, а затем скопирован и файл, указанный в поле backuppath
backupnumber	- число целое: последний номер сохраненной резервной копии;
backupcount	- число, общее количество сохраняемых резервных копий; по достижении данного значения, нумерация копий будет начата с 1;
backupdate	- строка; дата и время последней созданной резервной копии
backupinterval	- число целое; интервал в минутах между созданием резервных копий.
imarkdll	- строка, имя файла DLL вычислителя
cdninterval	- число целое, интервал опроса серверов ГИС МТ в секундах

Примечание. Резервная копия будет создана в файле, указанном в параметре backuplocalpath (по умолчанию – рабочий каталог imark), а затем файл будет перенесен в постоянное место, указанное в backuppath

PUT /settings – Изменение настроек.

Описание. **PUT** запрос для записи настроек imark.

Обязательным заголовком запроса является заголовок авторизации на сервере imark. Запрос доступен для пользователей с ролью администратора.

Тело запроса передается в виде объекта json с описанием настроек

```
{
  "version":НомерВерсии,
  "dbtype":"Тип SQL сервера",
  "driver":"Имя драйвера SQL сервера",
  "host":"адрес SQL сервера",
  "port": Порт доступа,
  "user":"ИмяПользователя",
  "password":"ПарольПользователя",
  "database":"Имя базы",
  "backuppath":"z:\imark\backups\backup[N].bak",
  "backuplocalpath":"d:\imark\backups ",
  "backupnumber":1,
  "backupcount":5,
  "backupdate":"2024-02-01T12:00:00",
  "backupinterval": Интервал,
  "logfilename":"logfilename"
  "imarkdll":"imark.dll",
  ":cdninterval":ИнтервалОпросаГисМТ
}
```

Описание полей запроса.

version	- число, номер версии формата параметров; в случае, если версия отличается от версии imark параметры не будут загружены, а будет возвращена ошибка 400.
dbtype	- строка, тип SQL сервера; допустимые значения <ul style="list-style-type: none">• SQLLITE - использовать встроенный вариант SQLITE DB;• MSSQL - использовать MS SQL сервер
host	- строка, адрес для доступа к серверу базы данных; для SQLITE не используется;
port	- число, порт для доступа к серверу базы данных; для SQLITE не используется;
user	- имя пользователя для доступа к серверу базы данных; для SQLITE не используется;
password	- пароль пользователя для доступа к серверу базы данных; для SQLITE не используется;
database	- имя базы данных на сервере; для SQLLITE полное имя файла с базой данных.
backuppath	- строка, полный путь к папке и имени файла для сохранения резервной копии базы данных; В имени файла и / или папки необходимо указать место для вставки номера резервной копии в виде [N]. Если указана сетевая папка, то служба imark должна иметь права на запись файлов по указанному пути.
backuplocalpath	- строка, полный путь к папке, в которой будет создан файл резервной копии, а затем скопирован и файл, указанный в поле backuppath
backupnumber	- число целое: последний номер сохраненной резервной копии;
backupcount	- число, общее количество сохраняемых резервных копий; по достижении данного значения, нумерация копий будет начата с 1;
backupdate	- строка; дата и время последней созданной резервной копии
backupinterval	- число целое; интервал в минутах между созданием резервных копий.
logfilename	- строка, имя файла журнала ошибок и сообщений
imarkdll	- строка, имя файла DLL вычислителя
cdninterval	- число целое, интервал опроса серверов ГИС МТ в секундах

Примечание. Резервная копия будет создана в файле, указанном в параметре backuplocalpath (по умолчанию – рабочий каталог imark), а затем файл будет перенесен в постоянное место, указанное в backuppath

Ответ на запрос возвращается в виде json с информацией о операции.

```
{
  "operationResult":"success"
}
```