Руководство по эксплуатации

DaData.ru

Содержание

1	Начало работы.....	3
1.1	Что умеет «Дадата».....	3
1.2	Как получить API-ключ.....	3
1.3	Как сменить пароль.....	3
2	Адреса.....	4
2.1	API стандартизации адресов.....	4
2.2	API подсказок по адресам.....	4
2.3	Геокодирование (координаты по адресу).....	5
2.4	Обратное геокодирование (адрес по координатам).....	6
2.5	Город по IP-адресу.....	8
2.6	Адрес по коду КЛАДР или ФИАС.....	10
2.7	Кадастровый номер.....	10
2.8	API: отделения Почты России.....	11
2.9	Идентификатор города в СДЭК, Voxberry и DPD.....	13
2.10	API: страны.....	14
2.11	Запись справочника по идентификатору (findByld).....	15
2.12	Поля справочника.....	15
2.13	Объект адреса.....	16
3	Компании.....	20
3.1	Организация по ИНН или ОГРН.....	20
3.2	API подсказок по организациям.....	21
3.3	Поиск аффилированных компаний.....	22
3.4	Объект компании.....	24
4	Банки.....	31
4.1	Банк по БИК, SWIFT, ИНН или регистрационному номеру.....	31
4.2	API подсказок по банкам.....	31
4.3	Объект банка.....	32
5	ФИО и телефоны.....	34
5.1	API стандартизации ФИО.....	34
5.2	API подсказок по ФИО.....	35
5.3	API стандартизации телефонов.....	36
6	Паспорта.....	39
6.1	API стандартизации паспортов.....	39

6.2	API: кем выдан паспорт.....	39
7	Email.....	42
7.1	API стандартизации email.....	42
8	Коды ответа на запрос.....	44

1 Начало работы

1.1 Что умеет «Дадата»

Подсказки

Помогают человеку быстро ввести корректные данные. Подсказывают ФИО, email, почтовые адреса, реквизиты компаний и банков по ходу ввода. Подключаются к любым сторонним справочникам. Работают через виджет и API.

Обратное геокодирование

Находит ближайшие адреса (дома, улицы, города) по географическим координатам. Только для России.

Адрес по ФИАС-коду

Возвращает детальную информацию об адресе по коду КЛАДР или ФИАС.

Город по IP-адресу

Определяет город по IP-адресу в России. Поддерживает как IPv4, так и IPv6 адреса. Возвращает детальную информацию о городе, в том числе почтовый индекс.

Организация по ИНН

Определяет реквизиты компании или индивидуального предпринимателя по ИНН или ОГРН.

Стандартизация

Автоматически исправляет ошибки и обогащает адреса, ФИО и телефоны. Проверяет email и паспорта. Работает с файлами и через API.

Поиск дублей

Находит похожих и одинаковых клиентов, адреса, телефоны или компании. Объединяет и строит единую карточку клиента.

Сравнение списков

Находит пересечение разных аудиторий для маркетологов. Автоматически сравнивает списки и показывает результат в удобной excel-таблице.

1.2 Как получить API-ключ

1. Зарегистрируйтесь.
2. Подтвердите емейл. После этого API-ключ автоматически сгенерируется.
3. Скопируйте готовый ключ в личном кабинете.

1.3 Как сменить пароль

1. Выйти из «Дадаты»
2. Нажать на «Войти»
3. Нажать на «Забыли пароль»
4. Указать email и нажать на «Получить инструкции»
5. Перейти по ссылке в письме.
6. Указать новый пароль и нажать на «Поменять пароль»

2 Адреса

2.1 API стандартизации адресов

- ✓ Разбивает адрес по отдельным полям (регион, город, улица, дом, квартира).
- ✓ Рассчитывает корректный индекс по данным Почты России.
- ✓ Определяет координаты.
- ✓ Показывает округ и район города, ближайшее метро, площадь и стоимость квартиры.
- ✓ Достает коды КЛАДР, ФИАС, ОКАТО, ОКТМО и ИФНС.

Только для России.

Пример запроса

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Token ${API_KEY}" \
-H "X-Secret: ${SECRET_KEY}" \
-d '[ "мск сухонска 117-89" ]' \
https://cleaner.dadata.ru/api/v1/clean/address
```

Укажите при вызове:

- Content-Type: application/json.
- \${API_KEY} — API-ключ.
- \${SECRET_KEY} — секретный ключ.

Тело запроса передавайте в кодировке UTF-8.

2.2 API подсказок по адресам

Ищет адреса по любой части адреса от региона до дома («тверская нижний 12» → «Нижегородская обл, г Нижний Новгород, ул Тверская, д 12»). Также ищет по почтовому индексу («105568» → «г Москва, ул Магнитогорская»).

Что умеет и не умеет

- ✓ Работает по всем странам мира (по России до квартиры, по Белоруссии до дома, по остальным странам — до города). Ищет и показывает результаты как на русском языке («Самара, пр-кт Металлургов»), так и на английском («Russia, gorod Samara, prospekt Metallurgov»).
- ✓ Находит актуальные адреса по историческим названиям (Свердловск → Екатеринбург) и синонимам (Питер → Санкт-Петербург).
- ✓ Ищет по частичному совпадению («москва болот» → «г Москва, Болотная наб»), но только в последнем слове запроса («мос болот» не найдет).
- ✓ Исправляет опечатки («самара авиационная») и запросы в неправильной раскладке («vjcrdf» → «москва»).
- ✓ Раскладывает выбранный адрес на гранулярные части (от региона до квартиры).
- ✓ Поддерживает гранулярные подсказки по отдельным частям адреса (регионы, города, улицы, дома).
- ✓ Подсказывает адреса в конкретных регионах, районах, городах и населенных пунктах. Понимает названия («Петергоф»), коды КЛАДР («780000080000») и ФИАС («8f238984-812b-4bb1-850b-49749fb5c56d»).

✓ Учитывает, где вы находитесь (в связке с методом город по IP-адресу).

Не умеет:

- ✗ Автоматически (без участия человека) обработать адреса из базы или файла.
- ✗ Транслитерировать (moskva suhonskaja 11 → 127642, г Москва, ул Сухонская, д 11).

Подсказки не подходят для автоматической обработки адресов. Они предлагают варианты, но не гарантируют, что угадали правильно. Поэтому окончательное решение всегда должен принимать человек.

Для автоматической обработки и транслитерации используйте «API Стандартизации».

Пример запроса

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -d '{ "query": "москва хабап" }' \
  https://suggestions.dadata.ru/suggestions/api/4_1/rs/suggest/address
```

Укажите при вызове:

- Content-Type: application/json или application/xml.
- Accept — аналогично.
- \${API_KEY} — API-ключ.

Тело запроса передавайте в кодировке UTF-8.

2.3 Геокодирование (координаты по адресу)

Определяет координаты адреса (дома, улицы, города). Только для России.

Пример запроса

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -H "X-Secret: ${SECRET_KEY}" \
  -d '[ "москва сухонская 11" ]' \
  https://cleaner.dadata.ru/api/v1/clean/address
```

Укажите при вызове:

- Content-Type: application/json.
- \${API_KEY} — API-ключ.
- \${SECRET_KEY} — секретный ключ.

Тело запроса передавайте в кодировке UTF-8.

Что в ответе

Адрес с координатами:

```
[
  {
    "source": "москва сухонская 11",
    "result": "г Москва, ул Сухонская, д 11",
    "postal_code": "127642",
    "country": "Россия",
    "region": "Москва",
    "city_area": "Северо-восточный",
    "city_district": "Северное Медведково",
    "street": "Сухонская",
    "house": "11",
    "geo_lat": "55.8782557",
    "geo_lon": "37.65372",
    "qc_geo": 0
  }
]
```

Координаты возвращаются в полях `geo_lat` и `geo_lon`. Поле `qc_geo` показывает, насколько точно удалось определить координаты:

Код <code>qc_geo</code>	Описание
0	Точные координаты дома
1	Ближайший дом
2	Улица
3	Населенный пункт
4	Город
5	Координаты не определены

Метод возвращает больше полей, чем показано в примере. Описание всех полей см. в «API Стандартизации».

Покрывтие координат

«Дадата» берет координаты домов и улиц из OpenStreetMap.

Покрывтие по домам:

- Москва — 97%,
- Санкт-Петербург — 91%,
- другие города-миллионники — 69%,
- остальная Россия — 47%.

По улицам:

- Москва — 95%,
- Санкт-Петербург — 94%,
- другие города-миллионники — 81%,
- остальная Россия — 70%.

2.4 Обратное геокодирование (адрес по координатам)

Находит ближайшие адреса (дома, улицы, города) по географическим координатам. Только для России.

Пример запроса

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -d '{ "lat": 55.878, "lon": 37.653 }' \
  https://suggestions.dadata.ru/suggestions/api/4_1/rs/geolocate/address
```

Параметры запроса

Название	Тип	Обяз. ?	По умол ч.	Описание
lat	number	✓		Географическая широта
lon	number	✓		Географическая долгота
count	number		10	Количество результатов (максимум — 20)
radius_meters	number		100	Радиус поиска в метрах (максимум – 1000)
language	string		ru	На каком языке вернуть результат (ru / en)

Пример запроса с ограничением по радиусу в 50 м:

```
{ "lat": 55.601983, "lon": 37.359486, "radius_meters": 50 }
```

Ответ

```
{
  "suggestions": [
    {
      "value": "г Москва, ул Сухонская, д 11",
      "unrestricted_value": "127642, г Москва, ул Сухонская, д 11",
      "data": {...}
    },
    {
      "value": "г Москва, ул Сухонская, д 11А",
      "unrestricted_value": "127642, г Москва, ул Сухонская, д 11А",
      "data": {...}
    },
    {
      "value": "г Москва, ул Сухонская, д 13",
      "unrestricted_value": "127642, г Москва, ул Сухонская, д 13",
      "data": {...}
    },
    {
      "value": "г Москва, ул Сухонская, д 9",
      "unrestricted_value": "127642, г Москва, ул Сухонская, д 9",
      "data": {...}
    }
  ]
}
```

Адреса в ответе идут в порядке удаления от заданных координат.

У адреса обычный набор полей, но некоторые из них не заполняются:

```
city_area
city_district*
flat_*
timezone
beltway_hit
beltway_distance
metro
```

2.5 Город по IP-адресу

- ✓ Определяет город по IP-адресу в России. Использует IP-адрес клиента, значение из параметра ip или заголовок X-Forwarded-For.
- ✓ Поддерживает как IPv4, так и IPv6 адреса.
- ✓ Возвращает детальную информацию о городе, в том числе почтовый индекс.

Пример запроса

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -d '{"ip": "46.226.227.20"}' \
  https://suggestions.dadata.ru/suggestions/api/4_1/rs/iplocate/address
```

Параметры запроса

Название	Тип	Обяз. ?	По умолч.	Описание
ip	string	✓		IP-адрес
language	string		ru	На каком языке вернуть результат (ru / en)

Ответ

```

{
  "location": {
    "value": "г Краснодар",
    "unrestricted_value": "350000, Краснодарский край, г Краснодар",
    "data": {
      "postal_code": "350000",
      "country": "Россия",
      "country_iso_code": "RU",
      "federal_district": "Южный",
      "region_fias_id": "d00e1013-16bd-4c09-b3d5-3cb09fc54bd8",
      "region_kladr_id": "23000000000000",
      "region_iso_code": "RU-KDA",
      "region_with_type": "Краснодарский край",
      "region_type": "край",
      "region_type_full": "край",
      "region": "Краснодарский",
      "area_fias_id": null,
      "area_kladr_id": null,
      "area_with_type": null,
      "area_type": null,
      "area_type_full": null,
      "area": null,
      "city_fias_id": "7dfa745e-aa19-4688-b121-b655c11e482f",
      "city_kladr_id": "2300000100000",
      "city_with_type": "г Краснодар",
      "city_type": "г",
      "city_type_full": "город",
      "city": "Краснодар",
      ...
    }
  }
}

```

Если город не удалось определить, ответ будет таким:

```

{
  "location": null
}

```

Точность определения города

Определение города по IP-адресу — не слишком надёжный способ. Точность составляет 60–80%. Если «Дадата» неправильно определила город или не определила его вовсе — сообщите об этом через специальную форму. Мы исправляем ошибки ежемесячно.

Как не исчерпать лимит запросов

Если просто подключить API «город по IP» на сервере, можно очень быстро выйти за бесплатный лимит в 10 000 запросов в сутки. Причины у этого две:

1. Когда человек ходит по страницам сайта, каждая страница заново пытается определить город.
2. Большое количество запросов генерируют поисковые боты (Яндекс, Гугл, Бинг).

Чтобы решить обе проблемы, мы рекомендуем:

1. Запоминать результат, который вернула «Дадата» — и не делать повторных вызовов на других страницах.
2. Вызывать API из браузера, а не с сервера — это защитит от ботов.

2.6 Адрес по коду КЛАДР или ФИАС

Находит адрес по идентификатору:

- КЛАДР-код (kladr_id), только для России;
- ФИАС-код (fias_id), только для России;
- Идентификатор OpenStreetMap (fias_id), только для Белоруссии;
- Идентификатор GeoNames (geoname_id), для всех остальных стран.

Пример запроса

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-H "Authorization: Token ${API_KEY}" \
-d '{ "query": "9120b43f-2fae-4838-a144-85e43c2bfb29" }' \
https://suggestions.dadata.ru/suggestions/api/4_1/rs/findById/address
```

По КЛАДР-коду ищет до улицы, по ФИАС-коду — до квартиры.

ФИАС-коды домов и квартир иногда меняются, а метод ищет только по актуальным кодам. Поэтому рекомендуем помимо ФИАС-кода сохранять адрес одной строкой — иначе не получится восстановить адрес, когда код изменится.

Параметры запроса

Название	Тип	Обяз. ?	По умолч.	Описание
query	string	✓		Текст запроса
count	number		10	Количество результатов (максимум — 20)
language	string		ru	На каком языке вернуть результат (ru / en)

2.7 Кадастровый номер

Находит кадастровый номер дома или квартиры по ФИАС-коду.

Если знаете ФИАС-код адреса — определите его кадастровый номер с помощью этого метода. Если не знаете ФИАС-код адреса — определите через API-метод разбора адреса из строки, а затем получите по коду кадастровый номер.

«Дадата» получает кадастровые номера из ФИАС. По России в целом они заполнены слабо — для 18% квартир и 10% домов. Но для Москвы ситуация лучше: кадастровые номера указаны у 94% квартир.

Пример запроса

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-H "Authorization: Token ${API_KEY}" \
-H "X-Secret: ${SECRET_KEY}" \
-d '{ "query": "f26b876b-6857-4951-b060-ec6559f04a9a" }' \
https://suggestions.dadata.ru/suggestions/api/4_1/rs/findById/cadastre
```

Параметры запроса

Название	Тип	Обяз. ?	По умолч.	Описание
query	string	✓		Текст запроса
count	number		10	Количество результатов (максимум — 20)

Укажите при вызове:

- Content-Type: application/json.
- \${API_KEY} — API-ключ.
- \${SECRET_KEY} — секретный ключ.

Тело запроса передавайте в кодировке UTF-8

Ответ

```
{
  "suggestions": [
    {
      "value": "f26b876b-6857-4951-b060-ec6559f04a9a",
      "unrestricted_value": "f26b876b-6857-4951-b060-ec6559f04a9a",
      "data": {
        "fias_level": "9",
        "house_cadnum": null,
        "flat_cadnum": "77:02:0004008:4143"
      }
    }
  ]
}
```

Название	Описание
value	ФИАС-код
unrestricted_value	= value
data.fias_level	Уровень адреса по ФИАС: 8 — дом 9 — квартира
data.house_cadnum	Кадастровый номер дома
data.flat_cadnum	Кадастровый номер квартиры

2.8 API: отделения Почты России

Если интернет-магазин доставляет покупки почтой по России, хорошо бы подсказать человеку, где и когда он может забрать посылку. В этом поможет справочник почтовых отделений — в нём есть точный адрес, координаты и часы работы отделения, а ещё отметка, если оно временно закрыто.

«Дадата» помогает найти точное отделение по индексу или ближайшее — по координатам.

Полнотекстовый поиск (suggest)

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -d '{"query": "дежнева 2а"}' \
  https://suggestions.dadata.ru/suggestions/api/4_1/rs/suggest/postal_unit
```

Поиск работает по полям:

- postal_code
- address_str

Запись справочника по идентификатору (findById)

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -d '{"query": "127642"}' \
  https://suggestions.dadata.ru/suggestions/api/4_1/rs/findById/postal_unit
```

Выборка по идентификатору работает по полям: `postal_code`

Поиск ближайшего по координатам (geolocate)

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -d '{"lat": 55.878, "lon": 37.653, "radius_meters": 1000 }' \
  https://suggestions.dadata.ru/suggestions/api/4_1/rs/geolocate/postal_unit
```

Фильтрация

Фильтрация работает по полям `is_closed`, `type_code` и `address_kladr_id`.

Фильтрация по `address_kladr_id` позволяет выбрать отделения из указанного города:

```
{
  "query": "Молодогвардейская",
  "filters": [
    {
      "address_kladr_id": "6300000100000"
    }
  ]
}
```

Фильтрация по `is_closed` позволяет отсеять закрытые отделения:

```
{
  "query": "105",
  "filters": [
    {
      "is_closed": false
    }
  ]
}
```

Поля справочника

Название	Описание
<code>value</code>	Значение одной строкой (как показывается в списке подсказок)
<code>unrestricted_value</code>	Адрес отделения одной строкой
<code>data.postal_code</code>	Почтовый индекс
<code>data.is_closed</code>	<code>true</code> , если отделение закрыто, иначе <code>false</code>
<code>data.type_code</code>	Тип отделения
<code>data.address_str</code>	Адрес одной строкой
<code>data.address_kladr_id</code>	КЛАДР-код населённого пункта
<code>data.address_qc</code>	Код проверки адреса
<code>data.geo_lat</code>	Широта

data.geo_lon	Долгота
data.schedule_mon	Режим работы в понедельник
data.schedule_tue	Режим работы во вторник
data.schedule_wed	Режим работы в среду
data.schedule_thu	Режим работы в четверг
data.schedule_fri	Режим работы в пятницу
data.schedule_sat	Режим работы в субботу
data.schedule_sun	Режим работы в воскресенье

2.9 Идентификатор города в СДЭК, Voxberry и DPD

Службы доставки часто используют собственные идентификаторы городов, и требуют от магазина указывать их в заказе.

Метод `findById/delivery` определяет идентификатор города в службе доставке на основании КЛАДР-кода города.

Пока поддерживаются службы доставки СДЭК, Voxberry и DPD.

Пример запроса

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-H "Authorization: Token ${API_KEY}" \
-d '{ "query": "3100400100000" }' \
https://suggestions.dadata.ru/suggestions/api/4_1/rs/findById/delivery
```

Ответ

```
{
  "suggestions": [
    {
      "value": "3100400100000",
      "unrestricted_value": "fe7eea4a-875a-4235-aa61-81c2a37a0440",
      "data": {
        "kladr_id": "3100400100000",
        "fias_id": "fe7eea4a-875a-4235-aa61-81c2a37a0440",
        "boxberry_id": "01929",
        "cdek_id": "344",
        "dpd_id": "196006461"
      }
    }
  ]
}
```

Если для указанного КЛАДР-кода ничего не найдено, ответ будет таким:

```
{
  "suggestions": []
}
```

Поля в ответе

Название	Описание
data.kladr_id	КЛАДР-код города
data.fias_id	ФИАС-код города

data.boxberry_id	Идентификатор города по справочнику Voxberry
data.cdek_id	Идентификатор города по справочнику СДЭК
data.dpd_id	Идентификатор города по справочнику DPD

2.10 API: страны

Справочник стран мира по стандарту ISO 3166.

Полнотекстовый поиск (suggest)

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -d '{"query": "та"}' \
  https://suggestions.dadata.ru/suggestions/api/4_1/rs/suggest/country
```

Поиск работает по полям:

- code
- alfa2
- alfa3
- name_short
- name

2.11 Запись справочника по идентификатору (findById)

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -d '{"query": "ТН"}' \
  https://suggestions.dadata.ru/suggestions/api/4_1/rs/findById/country
```

Выборка по идентификатору работает по полям:

- code
- alfa2
- alfa3

2.12 Поля справочника

Название	Описание
value	Значение одной строкой (как показывается в списке подсказок)
data.code	Цифровой код страны
data.alfa2	Буквенный код альфа-2
data.alfa3	Буквенный код альфа-3
data.name_short	Краткое наименование страны
data.name	Полное официальное наименование страны

2.13 Объект адреса

Название	Длина	Описание
source	250	Исходный адрес одной строкой
result	500	Стандартизованный адрес одной строкой
postal_code	6	Индекс
country	120	Страна
country_iso_code	2	ISO-код страны
federal_district	20	Федеральный округ
region_fias_id	36	ФИАС-код региона
region_kladr_id	19	КЛАДР-код региона
region_iso_code	6	ISO-код региона
region_with_type	131	Регион с типом
region_type	10	Тип региона (сокращенный)
region_type_full	50	Тип региона
region	120	Регион
area_fias_id	36	ФИАС-код района
area_kladr_id	19	КЛАДР-код района
area_with_type	131	Район в регионе с типом
area_type	10	Тип района в регионе (сокращенный)
area_type_full	50	Тип района в регионе
area	120	Район в регионе
city_fias_id	36	ФИАС-код города
city_kladr_id	19	КЛАДР-код города
city_with_type	131	Город с типом
city_type	10	Тип города (сокращенный)
city_type_full	50	Тип города
city	120	Город
city_area	120	Административный округ (только для Москвы)
city_district_fias_id	36	ФИАС-код района города (заполняется, только если район есть в ФИАС)
city_district_kladr_id	19	КЛАДР-код района города (не заполняется)
city_district_with_type	131	Район города с типом
city_district_type	10	Тип района города (сокращенный)
city_district_type_full	50	Тип района города
city_district	120	Район города
settlement_fias_id	36	ФИАС-код населенного пункта

settlement_kladr_id	19	КЛАДР-код населенного пункта
settlement_with_type	131	Населенный пункт с типом
settlement_type	10	Тип населенного пункта (сокращенный)
settlement_type_full	50	Тип населенного пункта
settlement	120	Населенный пункт
street_fias_id	36	ФИАС-код улицы
street_kladr_id	19	КЛАДР-код улицы
street_with_type	131	Улица с типом
street_type	10	Тип улицы (сокращенный)
street_type_full	50	Тип улицы
street	120	Улица
house_fias_id	36	ФИАС-код дома
house_kladr_id	19	КЛАДР-код дома
house_type	10	Тип дома (сокращенный)
house_type_full	50	Тип дома
house	50	Дом
block_type	10	Тип корпуса/строения (сокращенный)
block_type_full	50	Тип корпуса/строения
block	50	Корпус/строение
entrance	10	Подъезд
floor	10	Этаж
flat_fias_id	36	ФИАС-код квартиры
flat_type	10	Тип квартиры (сокращенный)
flat_type_full	50	Тип квартиры
flat	50	Квартира
flat_area	50	Площадь квартиры
square_meter_price	50	Рыночная стоимость м ²
flat_price	50	Рыночная стоимость квартиры
postal_box	50	Абонентский ящик
fias_id	36	ФИАС-код адреса (идентификатор ФИАС) ROOM.ROOMGUID – если квартира найдена в ФИАС HOUSE.HOUSEGUID – если дом найден в ФИАС ADDROBJ.AOGUID – в противном случае
fias_code		Иерархический код адреса в ФИАС (СС+PPP+GGG+PPP+CCCC+UUUU+DDDD)
fias_level	2	Уровень детализации, до которого адрес найден в ФИАС 0 – страна 1 – регион 3 – район 4 – город 5 – район города

		6 – населенный пункт 7 – улица 8 – дом 9 – квартира 65 – планировочная структура 90 – доп. территория 91 – улица в доп. территории -1 – иностранный или пустой
fias_actuality_state		Признак актуальности адреса в ФИАС 0 – актуальный 1-50 – переименован 51 – переподчинен 99 – удален
kladr_id	19	КЛАДР-код адреса
capital_marker	1	Признак центра района или региона 1 – центр района (Московская обл, Одинцовский р-н, г Одинцово) 2 – центр региона (Новосибирская обл, г Новосибирск) 3 – центр района и региона (Томская обл, г Томск) 4 – центральный район региона (Тюменская обл, Тюменский р-н) 0 – ничего из перечисленного (Московская обл, г Балашиха)
okato	11	Код ОКАТО
oktmo	11	Код ОКТМО
tax_office	4	Код ИФНС для физических лиц
tax_office_legal	4	Код ИФНС для организаций
timezone	50	Часовой пояс города для России, часовой пояс страны — для иностранных адресов. Если у страны несколько поясов, вернёт минимальный и максимальный через слеш: UTC+5/UTC+6
geo_lat	12	Координаты: широта
geo_lon	12	Координаты: долгота
beltway_hit	8	Внутри кольцевой? IN_MKAD – внутри МКАД (Москва) OUT_MKAD – за МКАД (Москва и область) IN_KAD – внутри КАД (Санкт-Петербург) OUT_KAD – за КАД (Санкт-Петербург и область) пусто – в остальных случаях
beltway_distance	3	Расстояние от кольцевой в км. Заполнено, только если beltway_hit = OUT_MKAD или OUT_KAD, иначе пустое
qc_geo	5	Код точности координат
qc_complete	5	Код пригодности к рассылке
qc_house	5	Признак наличия дома в ФИАС
qc	5	Код проверки адреса

unparsed_parts	250	Нераспознанная часть адреса. Для адреса «Москва, Митинская улица, 40, вход с торца» вернет «ВХОД, С, ТОРЦА»
metro		Список ближайших станций метро (до трёх штук)

Координаты есть у 97% домов в Москве, 91% в Санкт-Петербурге, 69% в других городах-миллиониках и 47% по остальной России.

Площадь и стоимость есть у 70% квартир в России.

3 Компании

3.1 Организация по ИНН или ОГРН

Находит компанию или ИП по ИНН или ОГРН. Возвращает все доступные сведения о компании, в отличие от метода `suggest`, который возвращает только базовые поля.

Пример запроса

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -d '{"query": "7707083893"}' \
  https://suggestions.dadata.ru/suggestions/api/4_1/rs/findById/party
```

Параметры запроса

Название	Тип	Обяз.? .	По умолчанию .	Описание
query	string	✓		ИНН или ОГРН
count	number		10	Количество результатов (максимум — 300)
kpp	string			КПП для поиска по филиалам, см. пример
branch_type	string			Головная организация (MAIN) или филиал (BRANCH), см. пример
type	string			Юрлицо (LEGAL) или индивидуальный предприниматель (INDIVIDUAL), см. пример

Головные организации и филиалы

Если у компании есть филиалы, запрос вернёт несколько объектов. Если нужна только головная организация, укажите дополнительный параметр `branch_type`:

```
{
  "query": "7707083893",
  "branch_type": "MAIN"
}
```

Чтобы найти конкретный филиал, укажите дополнительный параметр `kpp`:

```
{
  "query": "7707083893",
  "kpp": "540602001"
}
```

К сожалению, для 50% компаний налоговая служба пока не сообщает КПП филиалов. Такие филиалы с помощью данного метода найти не получится.

Юрлица и индивидуальные предприниматели

Чтобы искать только среди юрлиц, либо только среди индивидуальных предпринимателей, укажите параметр `type`:

```

# Юрлица
{
  "query": "7707083893",
  "type": "LEGAL"
}

# Индивидуальные предприниматели
{
  "query": "784806113663",
  "type": "INDIVIDUAL"
}

```

3.2 API подсказок по организациям

Ищет компании и индивидуальных предпринимателей:

- по ИНН, ОГРН и КПП;
- названию (полному и краткому);
- ФИО (для индивидуальных предпринимателей);
- ФИО руководителя компании;
- адресу до улицы.

Возможности:

- ✓ Ищет по комбинации ИНН, названия и адреса в одном запросе («7736050003 Газ» → «ПАО Газпром», «вавилова сбер» → «ПАО Сбербанк»).
- ✓ Находит конкретный филиал, если указать в запросе КПП («сбербанк 540602001» → «Сибирский банк ПАО Сбербанк»).
- ✓ Понимает слитное и раздельное написание («альфабанк» = «Альфа-Банк»).
- ✓ Ищет по частичному совпадению в ИНН / ОГРН («77094209» → «ООО Акварель») и названиях («росне» → «ПАО «НК «Роснефть»).
- ✓ Подсказывает только организации или только ИП, или и тех и других. Умеет искать только в действующих или ликвидированных компаниях. Может ограничить подсказки кодом ОКВЭД или конкретным регионом России.
- ✓ Учитывает, где вы находитесь (в связке с методом город по IP-адресу).
- ✓ Возвращает основные реквизиты компании из ЕГРЮЛ: краткое и полное название, ОПФ, адрес, ОГРН, ИНН, КПП, ОКВЭД, статус организации, ФИО и должность руководителя.
- ✓ В связке с методом организация по ИНН возвращает вагон дополнительной информации: количество сотрудников, все коды ОКВЭД, сведения о налоговой, ПФР и ФСС, документы и лицензии, учредители и руководители, финансовые показатели, реестр малого и среднего бизнеса.
- ✗ Для 50% компаний налоговая служба пока не сообщает КПП филиалов. Такие филиалы можно найти по ИНН, городу и улице филиала. Например, «7724261610 москва мясницкая» → «Филиал ФГУП "Почта России" (г Москва)».

Пример запроса

```

curl -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -d '{"query": "сбербанк"}' \
  https://suggestions.dadata.ru/suggestions/api/4_1/rs/suggest/party

```

Укажите при вызове:

- Content-Type: application/json или application/xml.
- Accept — аналогично.
- \${API_KEY} — API-ключ.

Тело запроса передавайте в кодировке UTF-8.

Параметры запроса

Название	Тип	Обяз .?	По умол ч.	Описание
query	string	✓		Текст запроса
count	number		10	Количество результатов (максимум — 20)
status	array[string]			Ограничение по статусу организации
type	string			Ограничение по типу организации
locations	array[object]			Ограничение по региону или городу
locations_boost	array[object]			Приоритет города при ранжировании

3.3 Поиск аффилированных компаний

Часто учредитель или директор не ограничивается участием в одном юрлице, а фигурирует в нескольких. Порой — в десятках разных компаний. Знать об этом полезно как для оценки надёжности контрагента, так и для более эффективного маркетинга и продаж. Чтобы облегчить поиск аффилированных компаний, «Дадата» находит организации по ИНН учредителей и руководителей.

Ищет по ИНН физлиц и юрлиц.

Пример запроса

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -d '{"query": "7736207543"}' \
  https://suggestions.dadata.ru/suggestions/api/4_1/rs/findAffiliated/party
```

Параметры запроса

Название	Тип	Обяз .?	По умолч.	Описание
query	string	✓		Текст запроса
count	number		10	Количество результатов (максимум — 300)
scope	array[string]		FOUNDER S, MANAGER S	Где искать — среди учредителей (FOUNDERS), руководителей (MANAGERS), или и тех, и других (по умолчанию).

Ограничение области поиска

По умолчанию метод ищет как среди учредителей, так и среди руководителей. Если требуется искать только среди учредителей, укажите `scope = FOUNDERS`:

```
{
  "query": "7736207543",
  "scope": ["FOUNDERS"]
}
```

Аналогично — если требуется искать только среди руководителей:

```
{
  "query": "7736207543",
  "scope": ["MANAGERS"]
}
```

Ответ

Объект с массивом организаций в поле `suggestions`:

```
{
  "suggestions": [
    {
      "value": "ООО \"ДЗЕН.ПЛАТФОРМА\"",
      "unrestricted_value": "ООО \"ДЗЕН.ПЛАТФОРМА\"",
      "data": { ... }
    },
    {
      "value": "ООО \"ЕДАДИЛ\"",
      "unrestricted_value": "ООО \"ЕДАДИЛ\"",
      "data": { ... }
    },
    {
      "value": "ООО \"ЗНАНИЕ\"",
      "unrestricted_value": "ООО \"ЗНАНИЕ\"",
      "data": { ... }
    },
    {
      "value": "ООО \"НАПА\"",
      "unrestricted_value": "ООО \"НАПА\"",
      "data": { ... }
    },
    {
      "value": "ООО \"ЯНДЕКС.ДРАЙВ\"",
      "unrestricted_value": "ООО \"ЯНДЕКС.ДРАЙВ\"",
      "data": { ... }
    }
  ]
}
```

3.4 Объект компании

Название	Описание
Базовые поля, заполняются для всех тарифов	
<code>value</code>	Наименование компании
<code>unrestricted_value</code>	= <code>value</code>
<code>data.inn</code>	ИНН
<code>data.kpp</code>	КПП

data.ogrn	ОГРН
data.ogrn_date	Дата выдачи ОГРН
data.hid	Внутренний идентификатор в Дадате
data.type	Тип организации LEGAL – юридическое лицо INDIVIDUAL – индивидуальный предприниматель
data.name	Наименование
└ full_with_opf	— полное наименование
└ short_with_opf	— краткое наименование
└ latin	— не заполняется
└ full	— полное наименование без ОПФ △ будет удалено в 2021, используйте full_with_opf
└ short	— краткое наименование без ОПФ △ будет удалено в 2021, используйте short_with_opf
data.fio	ФИО индивидуального предпринимателя
└ surname	— фамилия
└ name	— имя
└ patronymic	— отчество
└ gender	не заполняется
└ source	не заполняется
└ qc	не заполняется
	Коды статистики (только для действующих головных организаций и ИП):
data.okato	Код ОКАТО
data.oktmo	Код ОКТМО
data.okpo	Код ОКПО
data.okogu	Код ОКОГУ
data.okfs	Код ОКФС
data.okved	Код ОКВЭД
data.okved_type	Версия справочника ОКВЭД (2001 или 2014)
data.opf	Организационно-правовая форма
└ code	— код ОКОПФ
└ full	— полное название ОПФ
└ short	— краткое название ОПФ

L type	— версия справочника ОКОПФ (99, 2012 или 2014)
data.management	Руководитель
L name	— ФИО руководителя
L post	— должность руководителя
data.branch_count	Количество филиалов
data.branch_type	Тип подразделения MAIN — головная организация BRANCH — филиал
data.address	Адрес
L value	— адрес одной строкой: <ul style="list-style-type: none"> • адрес организации для юридических лиц; • город проживания для индивидуальных предпринимателей. стандартизован, поэтому может отличаться от записанного в ЕГРЮЛ.
L unrestricted_value	— адрес одной строкой (полный, с индексом) стандартизован, поэтому может отличаться от записанного в ЕГРЮЛ.
L data	— гранулярный адрес
L data.source	— адрес одной строкой как в ЕГРЮЛ
L data.qc	— код проверки адреса 0 — адрес распознан уверенно 1 или 3 — требуется ручная проверка
data.state	Состояние
L actuality_date	— дата последних изменений
L registration_date	— дата регистрации
L liquidation_date	— дата ликвидации
L status	— статус организации ACTIVE — действующая LIQUIDATING — ликвидируется LIQUIDATED — ликвидирована BANKRUPT — банкротство REORGANIZING — в процессе присоединения к другому юрлицу, с последующей ликвидацией
L code	— детальный статус (только на «Максимальном» тарифе)

Дополнительные поля, заполняются в зависимости от тарифа Заполняются при выборе конкретной подсказки. До этого они пустые.	
Тарифы «Расширенный» и «Максимальный»	
data.employee_count	Среднесписочная численность работников
data.finance.tax_system	Система налогообложения ENVD – единый налог на вмененный доход (ЕНВД) ESHN – единый сельскохозяйственный налог (ЕСХН) SRP – система налогообложения при выполнении соглашений о разделе продукции (СРП) USN – упрощенная система налогообложения (УСН) ENVD_ESHN – ЕНВД + ЕСХН USN_ENVD – УСН + ЕНВД
data.okved	Код ОКВЭД
data.okveds[]	Коды ОКВЭД дополнительных видов деятельности
L main	— основной или нет (true / false)
L type	— версия справочника ОКВЭД (2001 или 2014)
L code	— код по справочнику
L name	— наименование по справочнику
data.authorities	Сведения о налоговой, ПФР и ФСС
data.authorities.fts_registration	ИФНС регистрации
L type	— код гос. органа
L code	— код отделения
L name	— наименование отделения
L address	— адрес отделения одной строкой
data.authorities.fts_report	ИФНС отчетности, структура аналогична fts_registration
data.authorities.pf	Отделение Пенсионного фонда, структура аналогична fts_registration
data.authorities.sif	Отделение Фонда соц. страхования, структура аналогична fts_registration
data.citizenship	Гражданство ИП
L code.numeric	— числовой код страны по ОКСМ
L code.alpha_3	— трехбуквенный код страны по ОКСМ
L name.full	— полное наименование страны
L name.short	— краткое наименование страны

Только «Максимальный»	
data.founders[]	Учредители компании
└ ogrn	— ОГРН учредителя (для юрлиц)
└ inn	— ИНН учредителя
└ name	— наименование учредителя (для юрлиц)
└ fio	— ФИО учредителя (для физлиц)
└ hid	— внутренний идентификатор
└ type	— тип учредителя (LEGAL / PHYSICAL)
data.founders[].share	Доля учредителя
└ type	— тип значения (PERCENT / DECIMAL / FRACTION)
└ value	— значение (для type = PERCENT и type = DECIMAL)
└ numerator	— числитель дроби (для type = FRACTION)
└ denominator	— знаменатель дроби (для type = FRACTION)
data.management.disqualified	Наличие дисквалифицированных лиц в руководстве компании. Возвращает true при наличии, иначе null.
data.managers[]	Руководители компании
└ ogrn	— ОГРН руководителя (для юрлиц)
└ inn	— ИНН руководителя
└ name	— наименование руководителя (для юрлиц)
└ fio	— ФИО руководителя (для физлиц)
└ post	— должность руководителя (для физлиц)
└ hid	— внутренний идентификатор
└ type	— тип руководителя EMPLOYEE — сотрудник FOREIGNER — иностранный гражданин LEGAL — юрлицо
data.predecessors[]	Правопредшественники, только для юрлиц
└ ogrn	— ОГРН предшественника
└ inn	— ИНН предшественника
└ name	— наименование предшественника
data.successors[]	Правопреемники, только для юрлиц
└ ogrn	— ОГРН преемника
└ inn	— ИНН преемника
└ name	— наименование преемника
data.capital	Уставной капитал компании
└ type	— тип капитала

L value	— размер капитала
data.finance	Финансовые показатели за год частично заполнены у 70% действующих компаний
L year	— год бух. отчетности
L income	— доходы по бух. отчетности
L expense	— расходы по бух. отчетности
L debt	— недоимки по налогам
L penalty	— налоговые штрафы
data.documents	Документы
data.documents.fts_registration	Свидетельство о регистрации в налоговой
L type	— тип документа (= FTS_REGISTRATION)
L series	— серия документа
L number	— номер документа
L issue_date	— дата выдачи
L issue_authority	— код подразделения
data.documents.pf_registration	Свидетельство о регистрации в Пенсионном фонде, структура аналогична fts_registration
data.documents.sif_registration	Свидетельство о регистрации в Фонде соц. страхования, структура аналогична fts_registration
data.documents.smb	Запись в реестре малого и среднего предпринимательства
L type	— тип документа (= SMB)
L category	— категория предприятия (MICRO, SMALL или MEDIUM)
L issue_date	— дата регистрации в реестре
data.licenses[]	Лицензии
L series	— серия документа
L number	— номер документа
L issue_date	— дата выдачи
L issue_authority	— название выдавшего органа
L suspend_date	— дата приостановки
L suspend_authority	— название приостановившего органа
L valid_from	— дата начала действия
L valid_to	— дата окончания действия
L activities[]	— перечень лицензируемых видов деятельности
L addresses[]	— перечень адресов, по которым действует лицензия
data.phones[]	Телефоны, заполнены у 30% действующих компаний

└ data.source	— телефон одной строкой как в ЕГРЮЛ
└ data.type	— тип телефона (мобильный, стационарный, ...)
└ data.country_code	— код страны
└ data.city_code	— код города / DEF-код
└ data.number	— локальный номер телефона
└ data.provider	— оператор связи
└ data.region	— регион
└ data.city	— город (только для стационарных телефонов)
└ data.timezone	— часовой пояс
└ data.contact	— контактное лицо
data.emails[]	Адреса эл. почты, заполнены у 40% действующих компаний
└ data.source	— email одной строкой как в ЕГРЮЛ
└ data.local	— локальная часть адреса (то, что до «собачки»)
└ data.domain	— домен (то, что после «собачки»)
Не заполняются	
data.source	не заполняется
data.qc	не заполняется

4 Банки

4.1 Банк по БИК, SWIFT, ИНН или регистрационному номеру

Находит банк по любому из идентификаторов:

- БИК,
- SWIFT,
- ИНН,
- ИНН + КПП (для филиалов),
- регистрационному номеру, присвоенному Банком России.

Ищет только по точному совпадению, для частичного совпадения используйте метод suggest.

Пример запроса

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -d '{"query": "044525225"}' \
  https://suggestions.dadata.ru/suggestions/api/4_1/rs/findById/bank
```

4.2 API подсказок по банкам

✓ Ищет кредитные организации:

- по БИК,
- SWIFT,
- ИНН,
- ИНН + КПП,
- названию,
- адресу до улицы.

✓ Фильтрует по типу: банки, НКО и филиалы.

✓ Фильтрует по региону или городу.

✓ Умеет искать как действующие банки, так и банки на ликвидации.

✓ Учитывает, где вы находитесь (геолокация до города).

Пример вызова

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -d '{"query": "сбербанк"}' \
  https://suggestions.dadata.ru/suggestions/api/4_1/rs/suggest/bank
```

4.3 Объект банка

Название	Описание
value	Наименование банка
unrestricted_value	= value
data.bic	Банковский идентификационный код (БИК) ЦБ РФ
data.swift	Банковский идентификационный код в системе SWIFT
data.inn	ИНН
data.kpp	КПП
data.registration_number	Регистрационный номер в ЦБ РФ
data.correspondent_account	Корреспондентский счет в ЦБ РФ
data.treasury_accounts	Казначейские счета (для УФК)
data.name	Наименование
L payment	— платежное наименование
L full	— не заполняется
L short	— краткое наименование
data.payment_city	Город для платежного поручения (поля справочника Тпр + Нпр).
data.opf	Тип кредитной организации
L type	— код типа: BANK — банк BANK_BRANCH — филиал банка NKO — небанковская кредитная организация (НКО) NKO_BRANCH — филиал НКО RKC — расчетно-кассовый центр CBR — управление ЦБ РФ (март 2021) TREASURY — управление Казначейства (март 2021) OTHER — другой
L full	— не заполняется
L short	— не заполняется
data.cbr	Управление ЦБ РФ, к которому относится банк
L name.payment	— наименование управления ЦБ РФ
L bic	— БИК управления ЦБ РФ
L address.value	— адрес управления ЦБ РФ

data.address	Адрес регистрации
└ value	— адрес одной строкой стандартизован, поэтому может отличаться от записанного в справочнике БИК.
└ unrestricted_value	— адрес одной строкой (полный, с индексом) стандартизован, поэтому может отличаться от записанного в справочнике БИК.
└ data	— гранулярный адрес
└ data.source	— адрес одной строкой как в справочнике БИК
└ data.qc	— код проверки адреса 0 — адрес распознан уверенно 1 или 3 — требуется ручная проверка
data.state	Состояние
└ actuality_date	— дата актуальности сведений
└ registration_date	— дата регистрации
└ liquidation_date	— дата ликвидации
└ status	— статус организации ACTIVE — действующая LIQUIDATING — ликвидируется LIQUIDATED — ликвидирована
data.okpo	Не заполняется
data.phone	Не заполняется
data.rkc	Не заполняется

5 ФИО и телефоны

5.1 API стандартизации ФИО

- ✓ Исправляет опечатки и транслитерирует.
- ✓ Проставляет пол.
- ✓ Склоняет по падежам (кого? кому? кем?).

Пример вызова

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -H "X-Secret: ${SECRET_KEY}" \
  -d '[ "Срегей владимерович иванов" ]' \
  https://cleaner.dadata.ru/api/v1/clean/name
```

Укажите при вызове:

- Content-Type: application/json.
- \${API_KEY} — API-ключ.
- \${SECRET_KEY} — секретный ключ.

Тело запроса передавайте в кодировке UTF-8.

Ответ

Название	Длина	Описание
source	100	Исходное ФИО одной строкой
result	150	Стандартизованное ФИО одной строкой
result_genitive	150	ФИО в родительном падеже (кого?)
result_dative	150	ФИО в дательном падеже (кому?)
result_ablative	150	ФИО в творительном падеже (кем?)
surname	50	Фамилия
name	50	Имя
patronymic	50	Отчество
gender	10	Пол М — мужской Ж — женский НД — не удалось однозначно определить
qs	5	Код проверки

Код проверки qs

Требуется ли вручную проверить распознанное значение:

Код qs	Описание	Нужно проверить вручную?
0	Исходное значение распознано уверенно	нет
2	Исходное значение пустое или заведомо «мусорное»	нет

1	Исходное значение распознано с допущениями или не распознано	да
---	--	----

5.2 API подсказок по ФИО

- ✓ Подсказывает ФИО одной строкой или отдельно фамилию, имя, отчество.
- ✓ Исправляет клавиатурную раскладку («фунју» → «Антон»).
- ✓ Определяет пол.

Не умеет:

- ✗ Автоматически (без участия человека) обработать ФИО из базы или файла.
- ✗ Транслитерировать (Julia Somova → Юлия Сомова).
- ✗ Склонять по падежам (кого? кому? кем?).

Подсказки не подходят для автоматической обработки ФИО. Они предлагают варианты, но не гарантируют, что угадали правильно. Поэтому окончательное решение всегда должен принимать человек.

Для автоматической обработки, транслитерации и склонения по падежам используйте «API Стандартизации».

Пример запроса

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -d '{"query": "Викт"}' \
  https://suggestions.dadata.ru/suggestions/api/4_1/rs/suggest/fio
```

Укажите при вызове:

- Content-Type: application/json **или** application/xml.
- Accept — аналогично.
- \${API_KEY} — API-ключ.

Тело запроса передавайте в кодировке UTF-8.

Параметры запроса

Название	Тип	Обяз. ?	По умол ч.	Описание
query	string	✓		Текст запроса
count	number		10	Количество результатов (максимум — 20)
gender	string			Пол (UNKNOWN / MALE / FEMALE)
parts	array [string]			Подсказки по части ФИО

Ответ

Название	Описание
value	ФИО одной строкой
unrestricted_value	= value
data.surname	Фамилия

data.name	Имя
data.patronymic	Отчество
data.gender	Пол MALE — мужской FEMALE — женский UNKNOWN — не удалось однозначно определить
data.qc	Код проверки 0 — все части ФИО известны 1 — в ФИО есть неизвестная часть
Не используется	
data.source	

5.3 API стандартизации телефонов

- ✓ Проверяет телефон.
- ✓ Проставляет актуальный код города / DEF-код.
- ✓ Восстанавливает оператора. Учитывает переносы номера между операторами.
- ✓ Определяет страну, регион, город и часовой пояс.

Пример запроса

```
curl -X POST \
-H "Content-Type: application/json" \
-H "Authorization: Token ${API_KEY}" \
-H "X-Secret: ${SECRET_KEY}" \
-d '[ "раб 846)231.60.14 *139" ]' \
https://cleaner.dadata.ru/api/v1/clean/phone
```

Укажите при вызове:

- Content-Type: application/json.
- \${API_KEY} — API-ключ.
- \${SECRET_KEY} — секретный ключ.

Тело запроса передавайте в кодировке UTF-8.

Ответ

Название	Длина	Описание
source	100	Исходный телефон одной строкой
type	50	Тип телефона
phone	50	Стандартизованный телефон одной строкой
country_code	5	Код страны
city_code	5	Код города / DEF-код
number	10	Локальный номер телефона

extension	10	Добавочный номер
provider	100	Оператор связи (только для России)
country	50	Страна
region	100	Регион (только для России)
city	100	Город (только для стационарных телефонов)
timezone	50	Часовой пояс города для России, часовой пояс страны — для иностранных телефонов. Если у страны несколько поясов, вернёт минимальный и максимальный через слеш: UTC+5/UTC+6
qc_conflict	5	Признак конфликта телефона с адресом
qc	5	Код проверки

Тип телефона

Тип	Пример телефона
Мобильный	+7 911 243-45-68
Стационарный	+7 495 456-55-77
Прямой мобильный	+7 495 243-45-68
Колл-центр	8 800 222-12-22
Неизвестный	+7 333 111112

Код проверки qc

Нужно ли вручную проверить распознанный телефон:

Код qc	Описание	Нужно проверить вручную?
0	Российский телефон, распознан уверенно	Нет
7	Иностранный телефон, распознан уверенно	Нет
2	Телефон пустой или заведомо «мусорный»	Нет
1	Телефон распознан с допущениями или не распознан	Да
3	Обнаружено несколько телефонов, распознан первый	Да

Признак конфликта телефона с адресом qc_conflict

Указал ли клиент телефон, соответствующий его адресу. Удобно для проверки уровня риска:

Код qc_conflict	Описание
0	Телефон соответствует адресу
2	Города адреса и телефона отличаются
3	Регионы адреса и телефона отличаются

Заполняется только для российских адресов. Для иностранных всегда равен 0.

6 Паспорта

6.1 API стандартизации паспортов

- ✓ Проверяет формат серии и номера.
- ✓ Проверяет паспорт по справочнику недействительных паспортов МВД.

Пример вызова

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -H "X-Secret: ${SECRET_KEY}" \
  -d '[ "4509 235857" ]' \
  https://cleaner.dadata.ru/api/v1/clean/passport
```

Укажите при вызове:

- Content-Type: application/json.
- \${API_KEY} — API-ключ.
- \${SECRET_KEY} — секретный ключ.

Тело запроса передавайте в кодировке UTF-8.

Ответ

Название	Длина	Описание
source	100	Исходная серия и номер одной строкой
series	20	Серия
number	20	Номер
qc	5	Код проверки

Код проверки qc

Действует паспорт или нет, по данным Федеральной миграционной службы:

Код qc	Описание	Нужно проверить вручную?
0	Действующий паспорт	нет
2	Исходное значение пустое	нет
1	Неправильный формат серии или номера	да
10	Недействительный паспорт	да

6.2 API: кем выдан паспорт

Самое утомительное поле при вводе паспорта — «Кем выдан». Писать какое-нибудь «Отделом внутренних дел Медведевского района республики Марий Эл» муторно. Поэтому мы сделали подсказки по полю «Паспорт выдан...».

Полнотекстовый поиск (suggest)

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -d '{ "query": "772 053" }' \
  https://suggestions.dadata.ru/suggestions/api/4_1/rs/suggest/fms_unit
```

Поиск работает по полям:

- code
- name

Фильтрация

Фильтрация работает по полям `region_code` (первые 2 цифры КЛАДР-кода региона) и `type` (вид подразделения):

```
{
  "query": "иванов",
  "filters": [
    {
      "region_code": "37"
    }
  ]
}
```

```
{
  "query": "бел",
  "filters": [
    {
      "region_code": "02",
      "type": "2"
    }
  ]
}
```

Поля справочника

Название	Описание
value	Значение одной строкой (как показывается в списке подсказок)
unrestricted_value	= value
data.code	Код подразделения
data.name	Название подразделения в творительном падеже («кем выдан?»)
data.region_code	Код региона (2 цифры)
data.type	Вид подразделения (1 цифра) 0 – подразделение ФМС 1 – ГУВД или МВД региона 2 – УВД или ОВД района или города 3 – отделение полиции

7 Email

7.1 API стандартизации email

- ✓ Проверяет формат адреса.
- ✓ Исправляет распространённые опечатки.
- ✓ Проверяет, не «одноразовый» ли адрес.
- ✓ Классифицирует адреса на личные (@mail.ru, @yandex.ru), корпоративные (@myshop.ru) и «ролевые» (info@, support@).

Пример запроса

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Authorization: Token ${API_KEY}" \
  -H "X-Secret: ${SECRET_KEY}" \
  -d '[ "serega@yandex/ru" ]' \
  https://cleaner.dadata.ru/api/v1/clean/email
```

Укажите при вызове:

- Content-Type: application/json.
- \${API_KEY} — API-ключ.
- \${SECRET_KEY} — секретный ключ.

Тело запроса передавайте в кодировке UTF-8.

Ответ

Название	Длина	Описание
source	100	Исходный email
email	100	Стандартизованный email
local	100	Локальная часть адреса (то, что до «собачки»)
domain	100	Домен (то, что после «собачки»)
type		Тип адреса PERSONAL — личный (@mail.ru, @yandex.ru) CORPORATE — корпоративный (@myshop.ru) ROLE — «ролевой» (info@, support@) DISPOSABLE — одноразовый (@temp-mail.ru)
qs	5	Код проверки

Код проверки qs

Подходит ли email для маркетинговой рассылки:

Код qs	Описание	Нужно проверить вручную?
0	Корректное значение Соответствует общепринятым правилам, реальное существование адреса не проверяется	нет
2	Пустое или заведомо «мусорное» значение	нет

3	«Одноразовый» адрес Домены 10minutemail.com, getairmail.com, temp-mail.ru и аналогичные	нет
1	Некорректное значение Не соответствует общепринятым правилам	да
4	Исправлены опечатки	да

8 Коды ответа на запрос

HTTP-код ответа	Описание
200	Запрос успешно обработан
400	Некорректный запрос
401	В запросе отсутствует API-ключ или секретный ключ Или в запросе указан несуществующий ключ
403	Не подтверждена почта Или недостаточно средств для обработки запроса, пополните баланс
405	Запрос сделан с методом, отличным от POST
429	Слишком много запросов в секунду или новых соединений в минуту
5xx	Произошла внутренняя ошибка сервиса