KULISA DATA
АВТОМОБИЛЬНЫЕ ДАННЫЕ ДЛЯ БИЗНЕСА

REST API · JSON · UTF-8

Kulisa Data API

Описание методов и руководство по интеграции
базы автозапчастей и запчастей для спецтехники

Для кого: интернет-магазины и дистрибьюторы запчастей, сервисы подбора, маркетплейсы, 1С/CRM-интеграторы.
Что даёт: карточки по номеру, кроссы и аналоги, применимость по технике, подбор по VIN, цены рынка, проценка и переоценка списком, выгрузки в 1С/YML/Excel, мониторинг цен.



30,3 млн
номеров деталей (карточек)



500 млн
кроссов и аналогов



1,23 млрд
связей применимости «деталь — модель»



25,3 млн
фотографий деталей



5,3 млн
актуальных цен рынка



5 678
брендов производителей · 1378 марок техники

СОДЕРЖАНИЕ

Оглавление

1. Что такое Kulisa Data API 3
2. Быстрый старт 3
3. Методы API 6
3.1. Аккаунт 6
3.2. Деталь по номеру 7
3.3. Списки и цены 17
3.4. Файлы и фиды 22
3.5. Подбор по автомобилю и технике 25
3.6. Мониторинг цен и вебхуки 33
4. Сценарии для магазина спецтехники 37
5. Тарифы и доступ 39
6. Совместная работа с данными 39
7. Контакты 40

ВСЕ МЕТОДЫ

GET `/api/v1/me`	Тариф и остаток лимита	0
GET `/api/v1/lookup`	Карточка детали по номеру	1
GET `/api/v1/brands`	Бренды по номеру	1
GET `/api/v1/analogs`	Аналоги и кроссы	1
GET `/api/v1/applicability`	Применимость: на какую технику ставится	1
GET `/api/v1/offers`	Предложения поставщиков	1
GET `/api/v1/search`	Поиск по названию	1
GET `/api/v1/card`	Полная карточка одним вызовом	2
POST `/api/v1/bulk`	Проценка списком	по числу позиций
POST `/api/v1/reprice`	Переоценка: ваша цена против рынка	по числу позиций
GET · POST `/api/v1/export`	Выгрузка в файл	по числу позиций
GET `/api/v1/feed.yml`	YML-фид для сайта и маркетплейсов	по числу позиций
GET `/api/v1/vin`	Подбор по VIN	1
GET `/api/v1/bycar`	Детали по марке, модели, году	1
GET `/api/v1/makes`	Справочник марок	0
GET `/api/v1/models`	Модели марки	0
POST `/api/v1/watchlist`	Поставить позиции на мониторинг	0 (проверки — по позициям)
GET `/api/v1/watchlist`	Список позиций на мониторинге	0
DELETE `/api/v1/watchlist/:id`	Снять позицию с мониторинга	0
POST `/api/v1/webhook/test`	Тестовое событие на вебхук	0

РАЗДЕЛ 1

Что такое Kulisa Data API

Kulisa Data — собственная база автозапчастей и запчастей для коммерческой и специальной техники. API отдаёт её по HTTPS в формате JSON: по номеру детали, по списку, по VIN или по марке и модели.

ЧТО МОЖНО ПОЛУЧИТЬ

- **Карточку детали** по номеру: бренд, наименование, характеристики, фото.
- **Кроссы и аналоги** — до 200 взаимозаменяемых номеров других брендов.
- **Применимость** — марки, модели, двигатели и годы техники.
- **Подбор по VIN** и по марке/модели/году.
- **Цены рынка** — минимальная цена и число предложений, свежесть цены, фильтр по региону.
- **Проценку и переоценку** прайса списком до 500 / 5 000 позиций.
- **Выгрузки** в Excel, CSV, JSON, 1С (CommerceML), YML, МойСклад; мониторинг цен с вебхуками.

ОБЪЁМ БАЗЫ НА 24.11.2026

Номеров деталей (карточек)	30 318 056
Кроссов и аналогов	500 218 657
Связей применимости	1 231 888 224
Фотографий	25 387 657
Брендов производителей	5 678
Марок техники в справочнике	1 378
Актуальных цен рынка	5 368 085
Обновлений цен в сутки	≈ 2 530 000

Цифры из живой статистики базы; база пополняется ежедневно.

РАЗДЕЛ 2

Быстрый старт

1 Регистрация

Зарегистрируйтесь на `kulisabase.ru/register` — сразу включается демо-доступ: 14 дней, 1 000 запросов, все методы.

2 Ключ API

Ключ (токен) — в личном кабинете: «Настройки → API-токен» и раздел «Документация». Там же — перевыпуск ключа и песочница.

3 Первый запрос

Передайте ключ в заголовке `Authorization` и вызовите `/lookup`.

4 Тариф

Для рабочей нагрузки выберите тариф (раздел 5) — лимит увеличится сразу после оплаты.

Базовые параметры

Базовый URL	`https://kulisabase.ru/api/v1`
Протокол	HTTPS, REST. Методы `GET`, `POST`, `DELETE`.
Авторизация	Заголовок `Authorization: Bearer YOUR_API_KEY`. Для ссылок на файлы (1С, МойСклад, Google Таблицы, фиды) ключ можно передать параметром `?token=YOUR_API_KEY`.
Формат	Ответы — JSON в UTF-8 (`application/json; charset=utf-8`), файлы — `xlsx`, `csv` (UTF-8 с BOM, разделитель «;»), `xml`. Тело POST — JSON с `Content-Type: application/json`.

Признак успеха	Каждый JSON-ответ содержит `ok: true | false`; при ошибке — поле `error` с текстом на русском.
Номера деталей	Любое написание: пробелы, точки, дефисы и слэши игнорируются (`W 811/80` = `W81180`). Бренды сводятся по семействам (MANN = MANN-FILTER = Mann+Hummel).
Время ответа	По нашим замерам 24.11.2026: одиночные запросы — 0,1–3 с, списки и подбор по технике — 5–10 с. Рекомендуемый тайм-аут клиента — 60 с для одиночных и 120–180 с для списочных методов.

Первый запрос

CURL

```curl
curl "https://kulisabase.ru/api/v1/lookup?q=LF777&brand=FLEETGUARD" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

PYTHON

```python
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.get(API + "/lookup", params={"q": "LF777", "brand": "FLEETGUARD"}, headers=H, timeout=60)
data = r.json()
if data["ok"]:
    for it in data["items"]:
        print(it["brand"], it["number"], it["name"])
else:
    print("Ошибка:", r.status_code, data["error"])
```

Лимиты

Лимит — это количество единиц запросов на период тарифа. Каждый метод списывает единицы до выполнения запроса:

МЕТОДЫ	СТОИМОСТЬ
`/me`, `/makes`, `/models`, `/watchlist`, `/webhook/test`	0 — бесплатно
`/lookup`, `/brands`, `/analogs`, `/applicability`, `/offers`, `/search`, `/vin`, `/bycar`	1
`/card` (4 метода в одном)	2
`/bulk`, `/reprice`, `/export`, `/feed.yml`, проверки мониторинга	1 за каждую позицию списка

Остаток видно в методе `/me` и в кабинете. Если на запрос не хватает лимита (например, на список целиком), возвращается `429` с указанием остатка, лимит при этом не списывается. Отдельного ограничения частоты запросов нет; для больших объёмов используйте списочные методы (`/bulk`, `/reprice`) вместо тысяч одиночных вызовов.

Ошибки и коды ответа

КОД	КОГДА	ПРИМЕР ТЕЛА ОТВЕТА
`200`	Успех	`{"ok": true, …}`
`400`	Нет обязательного параметра, пустой список, VIN не 17 символов	`{"ok": false, "error": "VIN/PIN должен содержать 17 символов"}`

КОД	КОГДА	ПРИМЕР ТЕЛА ОТВЕТА
`401`	Ключ не передан или неверный	`{"ok": false, "error": "неверный или отсутствующий токен", "hint": "Authorization: Bearer <token> или ?token="}`
`403`	Аккаунт заблокирован или тариф без API	`{"ok": false, "error": "аккаунт заблокирован"}`
`429`	Не хватает лимита на запрос	`{"ok": false, "error": "не хватает лимита (осталось 12). Повысьте тариф…"}`
`502`	Источник данных временно недоступен — повторите запрос через несколько секунд	`{"ok": false, "error": "источник данных временно недоступен"}`

РАЗДЕЛ 3

Методы API

Все методы — с реальными ответами живого API, полученными 24.11.2026. Длинные массивы в примерах сокращены (отмечено комментарием «… ещё N»), ключ заменён на `YOUR_API_KEY`.

3.1 Аккаунт

GET `/api/v1/me` СТОИМОСТЬ: **0**

Тариф и остаток лимита

Возвращает текущий тариф, месячный лимит запросов, сколько израсходовано и сколько осталось. Лимит не тратит — удобно для мониторинга в вашей системе.

ПАРАМЕТРЫ

Параметров нет.

CURL

```
curl "https://kulisabase.ru/api/v1/me" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.get(API + "/me", headers=H, timeout=60)
data = r.json()
print(data["tariff_name"], "осталось:", data["req_left"])
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 96 мс · значения аккаунта обезличены


```
{
  "ok": true,
  "email": "you@company.ru",
  "company": "Ваша компания",
  "tariff": "start",
  "tariff_name": "Старт",
  "req_limit": 50000,
  "req_used": 1234,
  "req_left": 48766,
  "markup": 25,
  "webhook_url": null
}
```


ПОЛЯ ОТВЕТА

`tariff / tariff_name`	код и название тарифа
`req_limit`	лимит запросов на период
`req_used / req_left`	израсходовано / осталось

`markup`	наценка из настроек кабинета, % (используется в выгрузках и переоценке)
`webhook_url`	адрес вашего вебхука или null

Ошибки: `401` нет или неверный ключ

3.2 Деталь по номеру

GET `/api/v1/lookup` СТОИМОСТЬ: **1**

Карточка детали по номеру

Главный метод: по номеру детали возвращает карточки всех брендов с этим номером — наименование, характеристики, число кроссов и связок применимости, минимальную цену рынка и фото. Номер можно передавать в любом написании: `320/04133`, `320-04133` и `32004133` найдут одно и то же. Если указать `brand`, выдача сузится до одного производителя (написания бренда сводятся: MANN, MANN-FILTER, Mann+Hummel).

ПАРАМЕТРЫ

ПАРАМЕТР	ТИП	ОБЯЗ.	ОПИСАНИЕ
`q`	строка	да	номер детали, до 40 символов
`brand`	строка	нет	бренд — сузит выдачу до одного производителя

CURL

```
curl "https://kulisabase.ru/api/v1/lookup?q=0451103316&brand=BOSCH" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.get(API + "/lookup", params={"q": "0451103316", "brand": "BOSCH"}, headers=H, timeout=60)
data = r.json()
for it in data["items"]:
    print(it["brand"], it["number"], it["name"], it["market"])
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 2566 мс


```
{
  "ok": true,
  "q": "0451103316",
  "number_fix": "0451103316",
  "items": [
    {
      "id": "5958",
      "brand": "Bosch",
      "number": "0451103316",
      "number_fix": "0451103316",
      "name": "Фильтр масляный Accent/ Elantra/ Ceed/ Cerato",
```

```
      "country": "Германия",
      "attributes": {
        "D1": "82 mm",
        "D2": "62 mm",
        "Ø": "84 mm",
        "Height": "72 mm",
        "Connecting thread": "M 20 x 1,5",
        "Opening Pressure Bypass Valve": "1 bar"
        // … ещё 9 полей
      },
      "quality": 3,
      "appl": 1817,
      "crosses": 3574,
      "market": {
        "min": 150,
        "offers": 309,
        "at": "2026-11-24T14:31:12.408901+01:00"
      },
      "photos": [
        "https://kulisabase.ru/img/b4df922fd06494751788d5b57c64de57e905f88d.jpg",
        "https://kulisabase.ru/img/64d240d85128c349330375cb5e5a21cdddc38211.jpg"
        // … ещё 9 элементов
      ]
    }
  ],
  "pending": false,
  "queued": false
}
```


ПОЛЯ ОТВЕТА

`items[]`	карточки всех брендов, у которых есть деталь с таким номером (до 30)
`brand, number, number_fix`	бренд, номер как у производителя и нормализованный номер (без пробелов, точек, дефисов, слэшей)
`name, country`	наименование детали и страна бренда
`brand_raw, brand_note`	появляются, если в источнике вместо бренда стояла страна или обезличенное слово («Китай», «Россия», PRC, OEM, «Запчасть»): страна переносится в `country`, бренд — реальный, если он однозначно есть в названии, иначе «не указан»; исходное значение — в `brand_raw`
`attributes`	технические характеристики (размеры, резьба, давление и т. п.)
`quality`	полнота карточки: 3 — полная (характеристики, кроссы, применимость), 1 — только номер и название
`crosses, appl`	сколько у детали кроссов (аналогов) и уникальных связок применимости
`market`	`min` — минимальная цена рынка, ₽; `offers` — число предложений; `at` — когда цена обновлена
`photos`	ссылки на фото (https), можно показывать у себя на сайте
`queued, pending`	`queued: true` — карточка неполная и поставлена в очередь на дообогащение; `pending: true` — по номеру пока ничего нет, но он уже в очереди. Повторите запрос позже

Ошибки: `400` нет обязательного параметра · `401` нет или неверный ключ · `429` не хватает лимита · `502` источник данных временно недоступен

GET `/api/v1/brands` стоимость: **1**

Бренды по номеру

Какие производители выпускают деталь с таким номером. Помогает уточнить бренд перед запросом карточки, если номер «общий» (например, `LF777` есть у Fleetguard и у десятков производителей аналогов). До 50 строк.

ПАРАМЕТРЫ

ПАРАМЕТР	ТИП	ОБЯЗ.	ОПИСАНИЕ
`q`	строка	да	номер детали

CURL

```
curl "https://kulisabase.ru/api/v1/brands?q=LF777" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.get(API + "/brands", params={"q": "LF777"}, headers=H, timeout=60)
data = r.json()
print([row["brand"] for row in data["rows"]])
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 91 мс


```
{
  "ok": true,
  "rows": [
    {
      "brand": "AOSS",
      "number": "LF777",
      "name": "Фильтр масляный LF777 3889311 11996228 AOSS",
      "quality": 3
    },
    {
      "brand": "Filmant",
      "number": "LF777",
      "name": "Фильтр масляный Cummins Caterpillar",
      "quality": 3
    },
    {
      "brand": "Fleetguard",
      "number": "LF777",
      "name": "Фильтр масл. Fleetguard",
      "quality": 3
    },
    {
      "brand": "Foton",
      "number": "LF777",
      "name": "Фильтр масляный (PRC) Fleetguard LF777, шт",
      "quality": 3
    }
    // … ещё 28 элементов
```

```
  ]
}
```

Ошибки: `400` нет обязательного параметра · `401` нет или неверный ключ · `429` не хватает лимита · `502` источник данных временно недоступен

GET /api/v1/analogs

СТОИМОСТЬ: **1**

Аналоги и кроссы

Взаимозаменяемые детали других брендов (до 200): бренд, номер, наименование, минимальная цена рынка, число предложений и число фото. Сортировка — сначала позиции с ценой и предложениями.

ПАРАМЕТРЫ

ПАРАМЕТР	ТИП	ОБЯЗ.	ОПИСАНИЕ
`q`	строка	да	номер детали
`brand`	строка	нет	бренд исходной детали

CURL

```
curl "https://kulisabase.ru/api/v1/analogs?q=LF777&brand=FLEETGUARD" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.get(API + "/analogs", params={"q": "LF777", "brand": "FLEETGUARD"}, headers=H, timeout=60)
data = r.json()
for a in data["rows"][:10]:
    print(a["brand"], a["number"], a["min_price"], a["offers"])
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 1110 мс


```
{
  "ok": true,
  "rows": [
    {
      "brand": "Mann+Hummel",
      "number": "WP111023",
      "name": "Фильтр масляный BY PASS FH12/16 FS10 H=260.0мм     Mann+Hummel",
      "min_price": "936",
      "offers": 415,
      "imgs": "2"
    },
    {
      "brand": "Hengst",
      "number": "H200WN01",
      "name": "Фильтр масляный VOLVO",
      "min_price": "1006",
      "offers": 319,
      "imgs": "1"
    },
    {
```

```
      "brand": "Knecht/Mahle",
      "number": "OC282",
      "name": "Фильтр масляный",
      "min_price": "1591",
      "offers": 189,
      "imgs": "1"
    }
    // … ещё 197 элементов
  ]
}
```

ПОЛЯ ОТВЕТА

`rows[].brand, number, name`	бренд, номер и наименование аналога
`min_price, offers`	минимальная цена рынка, ₽, и число предложений (null — цен нет)
`imgs`	сколько фото есть у аналога

Ошибки: `400` нет обязательного параметра · `401` нет или неверный ключ · `429` не хватает лимита · `502` источник данных временно недоступен

GET /api/v1/applicability

СТОИМОСТЬ: 1

Применимость: на какую технику ставится

Марки, модели, двигатели и годы выпуска техники, на которую ставится деталь (до 500 строк). Каждая связка «марка · модель · двигатель · годы» приходит одной строкой, даже если она есть в нескольких источниках. Строки с пометкой `inferred: true` выведены через кроссы (известна только марка) — их стоит подтверждать по VIN или каталогу.

ПАРАМЕТРЫ

ПАРАМЕТР	ТИП	ОБЯЗ.	ОПИСАНИЕ
`q`	строка	да	номер детали
`brand`	строка	нет	бренд детали

CURL

```
curl "https://kulisabase.ru/api/v1/applicability?q=LF777&brand=FLEETGUARD" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.get(API + "/applicability", params={"q": "LF777", "brand": "FLEETGUARD"}, headers=H,
timeout=60)
data = r.json()
for a in data["rows"]:
    print(a["make"], a["model"], a["engine"], a["years"])
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 138 мс


```
{
  "ok": true,
  "rows": [
    {
      "make": "ERF",
      "model": "ECT",
      "engine": "10.35 ST D 2066 LF 03 350 л.с. Дизель",
      "years": "2002 - наст",
      "source": "autopiter"
    },
    {
      "make": "FORD",
      "model": "FORD L-SERIES AT9513",
      "engine": "AT9513 (N14) 422 л.с. дизель",
      "years": "01.1994 - 05.1997",
      "source": "tecdoc2026"
    },
    {
      "make": "FORD",
      "model": "FORD L-SERIES AT9513",
      "engine": "AT9513 (N14) 447 л.с. дизель",
      "years": "01.1994 - 05.1997",
      "source": "tecdoc2026"
    }
    // … ещё 76 элементов
  ],
  "inferred_only": false
}
```


ПОЛЯ ОТВЕТА

`rows[].make, model`	марка и модель
`engine`	двигатель: код, мощность, тип топлива
`years`	годы выпуска
`source`	источник связки
`inferred, confidence, note`	только у выведенных по кроссам строк
`inferred_only`	true — все строки выведены по кроссам, точной применимости нет

Ошибки: `400` нет обязательного параметра · `401` нет или неверный ключ · `429` не хватает лимита · `502` источник данных временно недоступен

GET `/api/v1/offers` стоимость: 1

Предложения поставщиков

Детальные предложения поставщиков по номеру: наличие, срок поставки в часах, склад, закупочная и розничная цена, признак аналога (до 300). Это сохранённый срез: дату показывает `fetched_at`, признак `fresh` — срез моложе 24 часов. В каждом ответе есть и цена рынка: `market` — минимальная цена, медиана, число предложений и источников, дата последнего обновления (`at`, `age_days`); `market_by_brand` — то же по брендам. Если детальных предложений по номеру нет, `source: "market"` и в `note` сказано, что показана последняя цена рынка.

ПАРАМЕТРЫ

ПАРАМЕТР	ТИП	ОБЯЗ.	ОПИСАНИЕ
`q`	строка	да	номер детали
`brand`	строка	нет	бренд детали

CURL

```
curl "https://kulisabase.ru/api/v1/offers?q=LF777" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.get(API + "/offers", params={"q": "LF777"}, headers=H, timeout=60)
data = r.json()
for o in data["rows"][:5]:
    print(o["supplier"], o["availability"], o["deadline_h"], o["price_out"])
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 2154 мс · по номеру нет детальных предложений — цена рынка


```
{
  "ok": true,
  "rows": [],
  "fresh": false,
  "queued": false,
  "collector": "off",
  "fetched_at": null,
  "source": "market",
  "market": {
    "brand": "Fleetguard",
    "min_price": 451,
    "median": 1671,
    "offers": 184,
    "sources": 3,
    "at": "2026-11-24T11:59:30.733Z",
    "age_days": 0,
    "source": "market",
    "note": "последняя цена рынка по нашему срезу (как market в /lookup); детальные предложения
поставщиков сейчас не собираются"
  },
  "market_by_brand": [
    {
      "brand": "Fleetguard",
      "min_price": 451,
      "median": 1671,
      "offers": 184,
      "sources": 3,
      "at": "2026-11-24T11:59:30.733Z",
      "age_days": 0
    },
    {
      "brand": "Cummins",
      "min_price": 1003,
      "median": 1003,
      "offers": 20,
      "sources": 1,
      "at": "2026-11-16T02:39:44.617Z",
      "age_days": 8
```

```
    }
    // … ещё 8 элементов
  ],
  "note": "Детальных предложений по номеру нет — показана последняя цена рынка с датой среза."
}
```

ПОЛЯ ОТВЕТА

`rows[].supplier, warehouse`	поставщик и склад
`availability`	количество в наличии
`deadline_h / deadline_max_h`	срок поставки, часов (мин./макс.)
`probability`	вероятность поставки, %
`price_in / price_out`	закупочная и розничная цена, ₽
`is_analog`	true — предложение по аналогу
`fresh, fetched_at`	свежесть среза предложений и время сбора
`source`	`offers` — есть детальные предложения, `market` — только цена рынка, `none` — данных нет
`market, market_by_brand`	цена рынка: `min_price`, `median`, `offers`, `sources`, `at`, `age_days`

Ошибки: `400` нет обязательного параметра · `401` нет или неверный ключ · `429` не хватает лимита · `502` источник данных временно недоступен

GET /api/v1/search

СТОИМОСТЬ: 1

Поиск по названию

Поиск деталей по словам из наименования (до 40 строк): «фильтр масляный JCB», «колодки тормозные передние». Ищутся все слова одновременно.

ПАРАМЕТРЫ

ПАРАМЕТР	ТИП	ОБЯЗ.	ОПИСАНИЕ
`q`	строка	да	слова из названия, до 80 символов

CURL

```
curl "https://kulisabase.ru/api/v1/search?q=фильтр масляный JCB" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.get(API + "/search", params={"q": "фильтр масляный JCB"}, headers=H, timeout=60)
data = r.json()
for row in data["rows"]:
    print(row["brand"], row["number"], row["name"])
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 1040 мс


```
{
  "ok": true,
  "rows": [
    {
      "id": "3212",
      "brand": "Jcb Landpower",
      "number": "02630795",
      "number_fix": "02630795",
      "name": "Фильтр масляный JCB - 02/630795",
      "quality": 3,
      "imgs": "0",
      "min_price": "1966"
    },
    {
      "id": "3265",
      "brand": "Jcb Landpower",
      "number": "02630225A",
      "number_fix": "02630225A",
      "name": "Фильтр масляный JCB 02/630225A",
      "quality": 3,
      "imgs": "1",
      "min_price": "237"
    },
    {
      "id": "8358092",
      "brand": "Hengst",
      "number": "H19W04",
      "number_fix": "H19W04",
      "name": "Фильтр масляный Hengst H19W04 (W 950/7) JCB; Komatsu; John Deere, 12шт/уп",
      "quality": 3,
      "imgs": "1",
      "min_price": "500"
    }
    // … ещё 37 элементов
  ]
}
```


Ошибки: `400` нет обязательного параметра · `401` нет или неверный ключ · `429` не хватает лимита · `502` источник данных временно недоступен

GET /api/v1/card

СТОИМОСТЬ: **2**

Полная карточка одним вызовом

Объединяет четыре метода: карточки (`/lookup`) + аналоги (`/analogs` , до 200) + применимость (`/applicability` , до 500) + предложения (`/offers`). Удобно для страницы товара или ответа менеджеру. Стоит 2 единицы лимита вместо 4.

ПАРАМЕТРЫ

ПАРАМЕТР	ТИП	ОБЯЗ.	ОПИСАНИЕ
`q`	строка	да	номер детали
`brand`	строка	нет	бренд детали

CURL

```
curl "https://kulisabase.ru/api/v1/card?q=LF777&brand=FLEETGUARD" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.get(API + "/card", params={"q": "LF777", "brand": "FLEETGUARD"}, headers=H, timeout=60)
data = r.json()
print(len(data["analogs"]), "аналогов,", len(data["applicability"]), "строк применимости")
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 1879 мс


```
{
  "ok": true,
  "q": "LF777",
  "items": [
    {
      "id": "16804969",
      "brand": "Fleetguard",
      "number": "LF777",
      "number_fix": "LF777",
      "name": "Фильтр масл. Fleetguard",
      "country": "Канада",
      "attributes": {
        "Ø": "116,08 mm",
        "Height": "249,2 mm",
        "Thread Size": "1 3/8-16 UNS-2B",
        "Outer diameter": "116,08 mm",
        "Высота [мм]": "249,2",
        "Диаметр [мм]": "116,08",
        "Размер резьбы": "1 3/8-16 UNS-2B",
        "Внешний диаметр [мм]": "116,08"
      },
      "quality": 3,
      "appl": 79,
      "crosses": 667,
      "market": {
        "min": 451,
        "offers": 182,
        "at": "2026-11-24T12:59:30.733877+01:00"
      },
      "photos": [
        "https://kulisabase.ru/img/b9064301b525db8028a9469a92dc7d9ec4a1756f.jpg"
        // … ещё 2 элементов
      ]
    }
  ],
  "analogs": [
    {
      "brand": "Mann+Hummel",
      "number": "WP111023",
      "name": "Фильтр масляный BY PASS FH12/16 FS10 H=260.0мм     Mann+Hummel",
      "min_price": "936",
      "offers": 415,
      "imgs": "2"
    },
    {
```

```
      "brand": "Hengst",
      "number": "H200WN01",
      "name": "Фильтр масляный VOLVO",
      "min_price": "1006",
      "offers": 319,
      "imgs": "1"
    }
    // … ещё 198 элементов
  ],
  "applicability": [
    {
      "make": "ERF",
      "model": "ECT",
      "engine": "10.35 ST D 2066 LF 03 350 л.с. Дизель",
      "years": "2002 - наст",
      "source": "autopiter"
    },
    {
      "make": "FORD",
      "model": "FORD L-SERIES AT9513",
      "engine": "AT9513 (N14) 422 л.с. дизель",
      "years": "01.1994 - 05.1997",
      "source": "tecdoc2026"
    }
    // … ещё 77 элементов
  ],
  "offers": []
}
```


ПОЛЯ ОТВЕТА

`items`	как в /lookup
`analogs`	как rows в /analogs
`applicability`	как rows в /applicability
`offers`	как rows в /offers

Ошибки: `400` нет обязательного параметра · `401` нет или неверный ключ · `429` не хватает лимита · `502` источник данных временно недоступен

3.3 Списки и цены

POST `/api/v1/bulk` СТОИМОСТЬ: **по числу позиций**

Проценка списком

До 500 позиций за запрос. По каждой строке — найденные карточки (лучшая первой), число кроссов, применимость, фото, минимальная цена рынка и число предложений, а также расшифровка цены: медиана, источник, свежесть. Бренд можно не указывать. Можно передать `region` — тогда цена считается по складам выбранного региона (см. ниже).

ПАРАМЕТРЫ

ПАРАМЕТР	ТИП	ОБЯЗ.	ОПИСАНИЕ
`items`	массив	да	`[{"brand": "…", "number": "…"}]`, до 500; `brand` необязателен

ПАРАМЕТР	ТИП	ОБЯЗ.	ОПИСАНИЕ
`region`	строка	нет	`all` (вся Россия, по умолчанию) · `my` (регион из кабинета) · `fo:ЮФО` · `r:77,50` (коды субъектов) · `Краснодарский край`
`region_mode`	строка	нет	`only` — только склады региона; `deliv` — плюс доставка в регион
`region_days`	число	нет	для `deliv`: максимальный срок доставки, дней

CURL

```
curl -X POST "https://kulisabase.ru/api/v1/bulk" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"items": [{"brand": "FLEETGUARD", "number": "LF777"}, {"number": "320/04133"}]}'
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.post(API + "/bulk", headers=H, json={
    "items": [
        {
            "brand": "FLEETGUARD",
            "number": "LF777"
        },
        {
            "number": "320/04133"
        }
    ]
}, timeout=120)
data = r.json()
for row in data["results"]:
    c = row["cards"][0] if row["cards"] else {}
    print(row["input"], c.get("brand"), c.get("min_price"), c.get("offers"))
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 6 810 мс


```
{
  "ok": true,
  "count": 4,
  "results": [
    {
      "input": "LF777",
      "brand": "FLEETGUARD",
      "found": 1,
      "brand_match": true,
      "cards": [
        {
          "brand": "Fleetguard",
          "number": "LF777",
          "name": "Фильтр масл. Fleetguard",
          "quality": 3,
          "appl": 79,
          "crosses": 667,
          "imgs": 4,
          "min_price": 451,
          "offers": 182,
          "price_offers_low": true,
          "min_price_raw": "451",
```

```
          "price": {
            "price": 451,
            "median": 451,
            "avg": 451,
            "at": "2026-11-24T11:59:30.733Z",
            "tier": 2,
            "label": "поставщики",
            "used": 1,
            "stale": 0,
            "outliers": 0,
            "excluded": 0,
            "stale_last": null
          }
        }
      ]
    },
    {
      "input": "320/04133",
      "brand": "",
      "found": 8,
      "brand_match": true,
      "cards": [
        {
          "brand": "JCB",
          "number": "320/04133",
          "name": "Масляный фильтр",
          "quality": 1,
          "appl": 0,
          "crosses": 41,
          "imgs": 0,
          "min_price": null,
          "offers": null,
          "price_offers_low": false,
          "min_price_raw": "1500",
          "price": {
            "price": null,
            "median": null,
            "avg": null,
            "at": null,
            "tier": 0,
            "label": "только устаревшие цены",
            "used": 0,
            "stale": 1,
            "outliers": 0,
            "excluded": 0,
            "stale_last": {
              "price": 1500,
              "at": "2026-11-01T04:56:40.035Z",
              "level": 3,
              "title": "интернет-магазин mirjcb"
            }
          }
        }
        // … ещё 7 элементов
      ]
    }
    // … ещё 2 элементов
  ]
}
```


ПОЛЯ ОТВЕТА

`results[].input, brand`	что вы передали
`found`	сколько карточек найдено

`cards[]`	карточки, лучшая первой: brand, number, name, crosses, appl, imgs
`min_price, offers`	минимальная цена рынка и число предложений
`price`	расшифровка цены: `price`, `median`, `label` (источник), `stale` (есть только устаревшие цены), `stale_last` — последняя известная цена и её дата

Ошибки: `400` пустой список · `401` ключ · `429` лимита не хватает на весь список · `502` источник недоступен

POST `/api/v1/reprice`

СТОИМОСТЬ: **по числу позиций**

Переоценка: ваша цена против рынка

Передаёте свои позиции с ценой (и при желании закупкой) — получаете минимальную цену рынка, позицию («дороже рынка» / «в рынке» / «дешевле»), рекомендацию и новую цену по выбранному правилу. До 5 000 позиций.

Правила (`rules.mode`): `market` — минимум рынка +5%; `undercut` — дешевле минимума на `pct` %; `hold` — не дороже рынка + `pct` %; `margin` — от закупки с наценкой `markup` %. Во всех режимах цена не опускается ниже закупки + `min_margin` %.

ПАРАМЕТРЫ

ПАРАМЕТР	ТИП	ОБЯЗ.	ОПИСАНИЕ
`items`	массив	да	`[{"brand", "number", "price", "cost"}]`; `cost` (закупка) необязателен
`rules`	объект	нет	`{"mode": "hold", "pct": 5, "min_margin": 15, "markup": 25, "round": 10}`
`region, region_mode, region_days`		нет	как в `/bulk`

CURL

```
curl -X POST "https://kulisabase.ru/api/v1/reprice" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"items": [{"brand": "FLEETGUARD", "number": "LF777", "price": 1500, "cost": 1100}], "rules":
{"mode": "hold", "pct": 5, "min_margin": 15}}'
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.post(API + "/reprice", headers=H, json={
    "items": [
        {
            "brand": "FLEETGUARD",
            "number": "LF777",
            "price": 1500,
            "cost": 1100
        }
    ],
    "rules": {
        "mode": "hold",
        "pct": 5,
```

```
        "min_margin": 15
    }
}, timeout=120)
data = r.json()
for r in data["results"]:
    print(r["number"], r["price"], "→", r["new_price"], r["rule"])
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 5703 мс


```
{
  "ok": true,
  "total": 2,
  "matched": 2,
  "results": [
    {
      "brand": "FLEETGUARD",
      "number": "LF777",
      "price": 1500,
      "market": 451,
      "offers": 182,
      "via": "точно",
      "pos": "дороже рынка",
      "sugg": 496,
      "rec": "дороже минимума на 233%, снизить до ~496",
      "market_raw": 451,
      "lvl": 2,
      "price_at": "2026-11-24T11:59:30.733Z",
      "used": 1,
      "dropped": 0,
      "wh": "",
      "wh_city": "",
      "wh_days": null,
      "new_price": 474,
      "delta_pct": -68.4,
      "rule": "не дороже рынка +5%"
    }
    // … ещё 1 элементов
  ],
  "rules": {
    "mode": "hold",
    "pct": 5,
    "min_margin": 15,
    "markup": 25,
    "round": 1
  },
  "summary": {
    "total": 2,
    "with_market": 2,
    "to_raise": 0,
    "to_lower": 2
  }
}
```


ПОЛЯ ОТВЕТА

`results[].market, offers`	минимальная цена рынка и число предложений
`pos, rec`	позиция относительно рынка и текстовая рекомендация
`new_price, delta_pct, rule`	новая цена по правилу, изменение в %, какое правило сработало
`summary`	итог: сколько позиций поднять / снизить

Ошибки: `400` нет обязательного параметра · `401` нет или неверный ключ · `429` не хватает лимита · `502` источник данных временно недоступен

3.4 Файлы и фиды

GET · POST `/api/v1/export`

СТОИМОСТЬ: **по числу позиций**

Выгрузка в файл

Готовый файл по списку номеров: `xlsx`, `csv`, `json`, `1c` (CommerceML 2.05 для загрузки в 1С и Битрикс), `yml`, `moysklad` (CSV импорта МойСклад). Ключ можно передать параметром `?token=` — так ссылку можно вставить прямо в 1С, МойСклад или Google Таблицы (`=IMPORTDATA(…&format=csv)`). Если бренд не указан, берётся наиболее вероятный производитель номера — с ценой рынка, собственными кроссами и применимостью (для `W81180` — Mann+Hummel); чтобы задать бренд явно, передавайте позиции через POST с `items: [{"brand", "number"}]`. В CSV, XLSX и JSON есть колонки `СтатусЦены` (есть цена / нет цены / не найдено) и `Примечание`; в МойСклад и 1С позиции без цены не попадают.

ПАРАМЕТРЫ

ПАРАМЕТР	ТИП	ОБЯЗ.	ОПИСАНИЕ
`list`	строка	да	номера через запятую (GET) или `items` (POST), до 500
`format`	строка	нет	`xlsx` (по умолчанию) · `csv` · `json` · `1c` · `yml` · `moysklad`
`markup`	число	нет	наценка, % — добавит колонку розничной цены

CURL

```
curl -o export.csv "https://kulisabase.ru/api/v1/export?format=csv&list=LF777,W81180" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.get(API + "/export", params={"format": "xlsx", "list": "LF777,W81180"}, headers=H,
timeout=60)
open("export.xlsx", "wb").write(r.content)
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 10185 мс · format=json


```
{
  "generated": "2026-11-24T14:14:04.248Z",
  "count": 2,
  "items": [
    {
      "input": "LF777",
      "brand": "Fleetguard",
      "number": "LF777",
      "name": "Фильтр масл. Fleetguard",
      "crosses": 667,
      "appl": 79,
```

```
      "imgs": 4,
      "min_price": 451,
      "offers": 182,
      "found": true,
      "price_status": "есть цена",
      "note": ""
    },
    {
      "input": "W81180",
      "brand": "Mann+Hummel",
      "number": "W81180",
      "name": "Фильтр масляный",
      "crosses": 4909,
      "appl": 1782,
      "imgs": 7,
      "min_price": 270,
      "offers": 26,
      "found": true,
      "price_status": "есть цена",
      "note": ""
    }
  ]
}
```


HTTP 200 format=csv · text/csv, UTF-8 с BOM, разделитель «;»


```
Запрос;Бренд;Артикул;Наименование;Кроссы;Применяемость;Фото;МинЦена;Предложений;СтатусЦены;Примечание

"LF777";"Fleetguard";"LF777";"Фильтр масл. Fleetguard";"667";"79";"4";"451";"182";"есть цена";""

"W81180";"Mann+Hummel";"W81180";"Фильтр масляный";"4909";"1782";"7";"270";"26";"есть цена";""
```


Ошибки: `400` пустой список · `401` · `429` · `502`

GET `/api/v1/feed.yml`

СТОИМОСТЬ: **по числу позиций**

YML-фид для сайта и маркетплейсов

YML-фид (Яндекс Маркет, InSales, Tilda и др.): название, бренд, артикул, цена (с наценкой `markup`), до 3 фото, число кроссов и применимость. Позиции без цены в фид не включаются (нулевая цена не проходит модерацию площадок): их список — комментарием в конце файла и числом в заголовке `X-Kulisa-Skipped-No-Price`. Ссылку с `?token=` можно указать как источник фида — фид собирается заново при каждом обращении.

ПАРАМЕТРЫ

ПАРАМЕТР	ТИП	ОБЯЗ.	ОПИСАНИЕ
`list`	строка	да	номера через запятую
`markup`	число	нет	наценка, %

CURL

```
curl -o feed.yml "https://kulisabase.ru/api/v1/feed.yml?list=W81180,320/04133,LF777&markup=25" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

PYTHON

```
import requests
```

```
API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.get(API + "/feed.yml", params={"list": "W81180", "markup": 25}, headers=H, timeout=60)
open("feed.yml", "w", encoding="utf-8").write(r.text)
```

ПРИМЕР ОТВЕТА

HTTP 200 application/xml · list=W81180,320/04133,LF777 · у 320/04133 нет цены — в фид не включён

```
<?xml version="1.0" encoding="UTF-8"?>
<yml_catalog date="2026-11-24 14:14">
  <shop>
    <name>KulisaBase</name>
    <company>Kulisa Data</company>
    <url>https://kulisabase.ru</url>
    <currencies><currency id="RUB" rate="1"/></currencies>
    <categories><category id="1">Автозапчасти</category></categories>
    <offers>
      <offer id="Mann_Hummel_W81180" available="true">
        <name>Фильтр масляный</name>
        <vendor>Mann+Hummel</vendor>
        <vendorCode>W81180</vendorCode>
        <price>270</price>
        <currencyId>RUB</currencyId>
        <categoryId>1</categoryId>
        <picture>https://kulisabase.ru/img/8f8ef7c5070aa0f857c0db14dca7e488eccdfad6.jpg</picture>
        <picture>https://kulisabase.ru/img/618c3f28d23c33b8232a528179aa7ee288ceb34e.jpg</picture>
        <picture>https://kulisabase.ru/img/ed1a8849c0cdf5177838fbcbb6c35aa3b32f2134.jpg</picture>
        <param name="Кроссов">4909</param>
        <param name="Применяемость">1782</param>
      </offer>
      <offer id="Fleetguard_LF777" available="true">
        <name>Фильтр масл. Fleetguard</name>
        <vendor>Fleetguard</vendor>
        <vendorCode>LF777</vendorCode>
        <price>451</price>
        <currencyId>RUB</currencyId>
        <categoryId>1</categoryId>
        <picture>https://kulisabase.ru/img/b9064301b525db8028a9469a92dc7d9ec4a1756f.jpg</picture>
        <picture>https://kulisabase.ru/img/0b2e50a437533a1d1de26ba3fc36ce1215cb7067.jpg</picture>
        <picture>https://alparta.ru/tdimg/4581/5152fbb0ac2763c6ed65aee52dc6566253a54f36.webp</picture>
        <param name="Кроссов">667</param>
        <param name="Применяемость">79</param>
      </offer>
    </offers>
  </shop>
</yml_catalog>
<!— без цены, в фид не включены (1): JCB 320/04133 —>
```

Ошибки: `400` · `401` · `429` · `502`

3.5 Подбор по автомобилю и технике

GET `/api/v1/vin` стоимость: **1**

Подбор по VIN

Расшифровка VIN (производитель, страна, год) и определение модели и поколения по каталогу базы; в ответе — подходящие модификации и список деталей с категориями. Если по VIN подходят несколько вариантов кузова, `ambiguous: true`, а в `models` — кандидаты: повторите запрос с `model`. Фильтр `part` оставит только нужный тип деталей.

PIN спецтехники. Метод принимает и 17-значные PIN по ISO 10261: JCB, Caterpillar, Komatsu, Hitachi, Doosan/Develon, Volvo CE, Liebherr, John Deere, Hyundai CE. Производитель определяется по префиксу (`pin: true`), модель — по коду в PIN и справочнику моделей (`JCB3CX…` → 3CX, `CAT0320D…` → 320D, `KMTPC200…` → PC200, `VCE0L120…` → L120). Если модель из PIN не читается (у ряда производителей код модели закрытый), приходит честный ответ с маркой, списком моделей и подсказкой `hint.bycar` — запрос `/bycar` с маркой и моделью. Если префикс не распознан, марку можно передать параметром `make`.

ПАРАМЕТРЫ

ПАРАМЕТР	ТИП	ОБЯЗ.	ОПИСАНИЕ
`vin`	строка	да	17 символов: VIN или PIN спецтехники
`make`	строка	нет	марка спецтехники, если префикс PIN не распознан
`model`	строка	нет	уточнение модели — название из `models`
`part`	строка	нет	тип детали, например «колодки», «фильтр масляный»
`limit`	число	нет	сколько деталей вернуть, до 500 (по умолчанию 300)

CURL

```
curl "https://kulisabase.ru/api/v1/vin?vin=Z6FAXXESMAKC24396&limit=5" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.get(API + "/vin", params={"vin": "Z6FAXXESMAKC24396", "part": "фильтр"}, headers=H,
timeout=60)
data = r.json()
print(data["resolved"]["model"], data["resolved"]["generation"])
for p in data["parts"][:10]:
    print(p["category"], p["brand"], p["number"], p["min_price"])
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 6 753 мс

```
{
  "ok": true,
  "vin": "Z6FAXXESMAKC24396",
  "decoded": {
```

```
    "vin": "Z6FAXXESMAKC24396",
    "valid": true,
    "errors": [],
    "wmi": "Z6F",
    "vds": "AXXESM",
    "vis": "AKC24396",
    "serial": "C24396",
    "manufacturer": "Ford Otosan",
    "country": "Турция",
    "make": "FORD",
    "region": "Европа",
    "year": 2019,
    "year_candidates": [
      2019,
      1989
    ],
    "check_digit_applicable": false,
    "check_digit_ok": null
  },
  "vehicle": null,
  "resolved": {
    "make": "FORD",
    "model": "TRANSIT",
    "year": 2019,
    "model_src": "line-code",
    "generation": "TRANSIT V363 Фургон (FCD, FDD)",
    "series": [
      "TRANSIT V363 Фургон (FCD, FDD)",
      "TRANSIT V363 с бортовой платформой/ходовая часть (FED, FFD)"
      // … ещё 1 элементов
    ],
    "confidence": 0.83,
    "exact_engine": false,
    "year_src": "VIN, 11-й знак (Ford)",
    "engine_hint": null
  },
  "note": "Под этот VIN подходят несколько вариантов кузова — при необходимости уточните модель из
списка.",
  "ambiguous": true,
  "mods": [
    {
      "vid": 190407,
      "engine": "2.0 EcoBlue",
      "codes": "BKFA BKFB YMF6 YMFA YMFB YMFS",
      "years": "2016-",
      "kw": 96,
      "ccm": 1995,
      "body": "Фургон"
    }
    // … ещё 53 элементов
  ],
  "models": [
    {
      "model": "TRANSIT V363 с бортовой платформой/ходовая часть (FED, FFD)",
      "make": "FORD",
      "mods": 48,
      "years": "2013-н.в.",
      "fits": true,
      "main": true
    },
    {
      "model": "TRANSIT V363 Фургон (FCD, FDD)",
      "make": "FORD",
      "mods": 47,
```

```
      "years": "2013-н.в.",
      "fits": true,
      "main": true
    }
    // … ещё 44 элементов
  ],
  "warnings": [],
  "parts": [
    {
      "number_fix": "FDLS7924",
      "brand": "MOOG",
      "number": "FD-LS-7924",
      "name": "Тяга / стойка, стабилизатор",
      "type": "Тяга / стойка, стабилизатор",
      "category": "Подвеска колеса",
      "oe": false,
      "mods": 40,
      "models": [
        "TRANSIT V363 Фургон (FCD, FDD)"
      ],
      "min_price": 1332,
      "offers": 13
    },
    {
      "number_fix": "081049002099",
      "brand": "MEKRA",
      "number": "08.1049.002.099",
      "name": "Фонарь указателя поворота",
      "type": "Фонарь указателя поворота",
      "category": "Сигнализация",
      "oe": false,
      "mods": 40,
      "models": [
        "TRANSIT V363 Фургон (FCD, FDD)"
      ],
      "min_price": null,
      "offers": null
    }
    // … ещё 3 элементов
  ],
  "parts_count": 49841,
  "mods_count": 40,
  "apps_scanned": 1198,
  "categories": [
    {
      "name": "Подвеска колеса",
      "cnt": 2
    },
    {
      "name": "Пневматическая система",
      "cnt": 2
    },
    {
      "name": "Сигнализация",
      "cnt": 1
    }
  ]
}
```


HTTP 200 реальный ответ · 386 мс · PIN спецтехники JCB3CX4TE01234567


```
{
  "ok": true,
  "vin": "JCB3CX4TE01234567",
  "pin": true,
```

```
  "decoded": {
    "vin": "JCB3CX4TE01234567",
    "valid": true,
    "pin": true,
    "standard": "ISO 10261 (PIN спецтехники)",
    "wmi": "JCB",
    "model_code": "3CX4T",
    "serial": "234567",
    "manufacturer": "JCB",
    "country": "Великобритания",
    "make": "JCB",
    "year": null,
    "errors": []
  },
  "vehicle": null,
  "resolved": {
    "make": "JCB",
    "model": "3CX-4T",
    "year": null,
    "model_src": "pin-model-code",
    "generation": null,
    "confidence": 0.6,
    "exact_engine": false,
    "year_src": "год по PIN спецтехники не определяется",
    "engine_hint": null
  },
  "note": "PIN спецтехники JCB: модель 3CX-4T прочитана по коду в PIN (3CX4T). Год и комплектация по PIN
не определяются — при сомн…",
  "ambiguous": false,
  "hint": {
    "bycar": "/api/v1/bycar?make=JCB&model=3CX-4T",
    "models": "/api/v1/models?make=JCB"
  },
  "models": [
    {
      "model": "FASTTRAC",
      "make": "JCB",
      "mods": 6,
      "years": null,
      "fits": true,
      "main": false
    },
    {
      "model": "3CX",
      "make": "JCB",
      "mods": 23,
      "years": null,
      "fits": true,
      "main": false
    },
    {
      "model": "ROBOT",
      "make": "JCB",
      "mods": 33,
      "years": null,
      "fits": true,
      "main": false
    }
    // … ещё 37 элементов
  ],
  "parts": [
    {
      "number_fix": "04501700",
      "brand": "PETERS ENNEPETAL",
```

```
      "number": "045.017-00",
      "name": "Стремянка рессоры",
      "mods": 2,
      "models": [
        "3CX-4T"
      ],
      "min_price": null,
      "offers": null
    },
    {
      "number_fix": "12303138",
      "brand": "JCB",
      "number": "12303138",
      "name": "Подушка двигателя н.м. 123/03138",
      "mods": 2,
      "models": [
        "3CX-4T"
      ],
      "min_price": null,
      "offers": null
    }
    // … ещё 3 элементов
  ],
  "parts_count": 5,
  "mods_count": 0,
  "apps_scanned": 5,
  "categories": [
    {
      "name": "Стремянка",
      "cnt": 1
    },
    {
      "name": "Подушка",
      "cnt": 1
    }
    // … ещё 3 элементов
  ]
}
```


ПОЛЯ ОТВЕТА

`decoded`	расшифровка VIN: WMI, производитель, страна, год, контрольная цифра
`resolved`	марка, модель, поколение и уверенность (`confidence` 0…1)
`ambiguous, models`	несколько кандидатов — уточните `model`
`mods`	модификации: двигатель, коды, годы, кВт, объём, кузов
`parts[]`	детали: категория, бренд, номер, `oe` (оригинал), цена рынка
`parts_count, categories`	всего деталей и разбивка по категориям
`pin, hint`	только для PIN спецтехники: признак и подсказка запроса `/bycar`

Ошибки: `400` VIN/PIN не 17 символов · `401` · `429` · `502`

GET `/api/v1/bycar` СТОИМОСТЬ: **1**

Детали по марке, модели, году

Подбор без VIN: детали для марки (обязательно), модели, года и двигателя — до 300 позиций с категориями, ценой рынка и числом предложений. Марка — как в `/makes`.

ПАРАМЕТРЫ

ПАРАМЕТР	ТИП	ОБЯЗ.	ОПИСАНИЕ
`make`	строка	да	марка, как в `/makes`
`model`	строка	нет	модель
`year`	число	нет	год выпуска
`engine`	строка	нет	двигатель
`limit`	число	нет	сколько деталей вернуть

CURL

```
curl "https://kulisabase.ru/api/v1/bycar?make=FORD&model=FOCUS&year=2012&limit=5" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.get(API + "/bycar", params={"make": "JCB", "limit": 50}, headers=H, timeout=60)
data = r.json()
for p in data["parts"]:
    print(p["brand"], p["number"], p["name"], p["models"][:3])
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 8786 мс


```
{
  "ok": true,
  "resolved": {
    "make": "FORD",
    "model": "FOCUS",
    "year": 2012,
    "engine": null
  },
  "parts": [
    {
      "number_fix": "29224",
      "brand": "Febi",
      "number": "29224",
      "name": "Рулевой наконечник",
      "mods": 211,
      "models": [
        "FOCUS II (DA_, HCP, DP)",
        "FOCUS I (DAW, DBW)",
        "FOCUS II SALOON (DB_, FCH, DH)"
        // … ещё 3 элементов
      ],
      "min_price": 1000,
      "offers": 590
    },
    {
      "number_fix": "29223",
      "brand": "Febi",
      "number": "29223",
      "name": "Рулевой наконечник",
      "mods": 211,
```

```
      "models": [
        "FOCUS II (DA_, HCP, DP)",
        "FOCUS I (DAW, DBW)",
        "FOCUS II SALOON (DB_, FCH, DH)"
        // … ещё 3 элементов
      ],
      "min_price": 500,
      "offers": 545
    }
    // … ещё 3 элементов
  ],
  "parts_count": 5,
  "mods_count": 306,
  "apps_scanned": 5,
  "categories": [
    {
      "name": "Рулевой",
      "cnt": 2
    },
    {
      "name": "Рулевая",
      "cnt": 2
    }
    // … ещё 1 элементов
  ]
}
```


Ошибки: `400` нет make · `401` · `429` · `502`

GET /api/v1/makes

СТОИМОСТЬ: **0**

Справочник марок

Все марки с числом связок применимости (сейчас 1378 марок). Лимит не тратит.

ПАРАМЕТРЫ

Параметров нет.

CURL

```
curl "https://kulisabase.ru/api/v1/makes" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.get(API + "/makes", headers=H, timeout=60)
data = r.json()
print(len(data["makes"]), "марок")
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 102 мс


```
{
  "ok": true,
  "makes": [
    {
```

```
      "make": "MERCEDES-BENZ",
      "cnt": 182770013
    },
    {
      "make": "VW",
      "cnt": 105033208
    },
    {
      "make": "MAN",
      "cnt": 88240508
    },
    {
      "make": "AUDI",
      "cnt": 72837459
    },
    {
      "make": "BMW",
      "cnt": 66474478
    }
    // … ещё 1 373 элементов
  ]
}
```


Ошибки: `401`

GET `/api/v1/models`

СТОИМОСТЬ: **0**

Модели марки

Модели выбранной марки с числом связок и модификаций — для всех марок из `/makes`, включая грузовики и спецтехнику (MAN, Scania, DAF, Iveco, КАМАЗ, JCB, Caterpillar, Komatsu, Hitachi, Doosan, Liebherr). Лимит не тратит. Если справочник марки в этот момент пересчитывается, приходит предварительный список по каталогу модификаций с `partial: true`.

ПАРАМЕТРЫ

ПАРАМЕТР	ТИП	ОБЯЗ.	ОПИСАНИЕ
`make`	строка	да	марка, как в `/makes`

CURL

```
curl "https://kulisabase.ru/api/v1/models?make=JCB" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.get(API + "/models", params={"make": "JCB"}, headers=H, timeout=60)
data = r.json()
for m in data["models"][:20]:
    print(m["model"], m["mods"])
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 166 мс · make=JCB


```
{
  "ok": true,
  "make": "JCB",
  "models": [
    {
      "model": "двигатель",
      "cnt": 3003,
      "mods": 360
    },
    {
      "model": "FASTTRAC",
      "cnt": 1202,
      "mods": 6
    },
    {
      "model": "3CX",
      "cnt": 413,
      "mods": 23
    },
    {
      "model": "ROBOT",
      "cnt": 346,
      "mods": 33
    }
    // … ещё 3 439 элементов
  ]
}
```


Ошибки: `401` · `502`

3.6 Мониторинг цен и вебхуки

POST `/api/v1/watchlist` стоимость: **0 (проверки — по позициям)**

Поставить позиции на мониторинг

Добавляет позиции в мониторинг цены. Система периодически проверяет цену рынка; при изменении отправляет событие на ваш вебхук. Сами проверки тратят лимит по числу позиций.

ПАРАМЕТРЫ

ПАРАМЕТР	ТИП	ОБЯЗ.	ОПИСАНИЕ
`items`	массив	да	`[{"brand", "number"}]`

CURL

```
curl -X POST "https://kulisabase.ru/api/v1/watchlist" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"items": [{"brand": "MANN-FILTER", "number": "W81180"}]}'
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}
```

```
r = requests.post(API + "/watchlist", headers=H, json={"items": [{"brand": "MANN-FILTER", "number":
"W81180"}]}, timeout=120)
data = r.json()
print(data)
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 115 мс


```
{
  "ok": true,
  "added": 1
}
```


Ошибки: `400` пустой список · `401`

GET /api/v1/watchlist

СТОИМОСТЬ: 0

Список позиций на мониторинге

Позиции на мониторинге: последняя и предыдущая цена, число предложений, время проверки и изменения.

ПАРАМЕТРЫ

Параметров нет.

CURL

```
curl "https://kulisabase.ru/api/v1/watchlist" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.get(API + "/watchlist", headers=H, timeout=60)
data = r.json()
for w in data["items"]:
    print(w["id"], w["number"], w["last_price"], w["prev_price"])
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 74 мс


```
{
  "ok": true,
  "items": [
    {
      "id": 22,
      "brand": "MANN-FILTER",
      "number": "W81180",
      "last_price": null,
      "prev_price": null,
      "last_offers": null,
      "checked_at": null,
      "changed_at": null
    }
```

```
  ]
}
```

Ошибки: `401`

DELETE /api/v1/watchlist/:id

стоимость: **0**

Снять позицию с мониторинга

Удаляет позицию по `id` из `/watchlist`.

ПАРАМЕТРЫ

ПАРАМЕТР	ТИП	ОБЯЗ.	ОПИСАНИЕ
`id`	число	да	id позиции (в пути)

CURL

```
curl -X DELETE "https://kulisabase.ru/api/v1/watchlist/22" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

PYTHON

```
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.delete(API + "/watchlist/22", headers=H, timeout=60)
data = r.json()
print(data)
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 78 мс

```
{
  "ok": true
}
```

Ошибки: `401`

POST /api/v1/webhook/test

стоимость: **0**

Тестовое событие на вебхук

Отправляет тестовое событие на ваш вебхук. Адрес и секрет задаются в кабинете (Настройки → Вебхук). События приходят POST-запросом с JSON; подпись HMAC-SHA256 вашим секретом — в заголовке `x-kulisa-signature`. Пример ответа ниже — когда вебхук ещё не настроен.

ПАРАМЕТРЫ

Параметров нет.

CURL

```
curl -X POST "https://kulisabase.ru/api/v1/webhook/test" \
  -H "Authorization: Bearer YOUR_API_KEY" \
```

```
  -H "Content-Type: application/json" \
  -d '{}'
```

PYTHON

```python
import requests

API = "https://kulisabase.ru/api/v1"
H = {"Authorization": "Bearer YOUR_API_KEY"}

r = requests.post(API + "/webhook/test", headers=H, json={}, timeout=120)
data = r.json()
print(data)
```

ПРИМЕР ОТВЕТА

HTTP 200 реальный ответ · 68 мс

```
{
  "ok": false,
  "error": "вебхук не настроен (Настройки → Вебхук)"
}
```

Ошибки: `401`

РАЗДЕЛ 4

Сценарии для магазина запчастей спецтехники

Три типовые задачи менеджера и сайта — на реальных номерах и реальных ответах API от 24.11.2026.

A По номеру найти аналоги и применимость

Клиент присылает номер **LF777** и спрашивает, чем заменить и на что ставится.

1. `GET /brands?q=LF777` — номер есть у 32 производителей; выбираем оригинал **Fleetguard**.
2. `GET /card?q=LF777&brand=FLEETGUARD` — одним вызовом (2 единицы лимита): карточка, **200 аналогов** (Caterpillar, Donaldson, Mann+Hummel, Hengst, Blumaq, SDLG и др.) и **79 строк применимости** (ERF, Ford, Freightliner, International, Kenworth, Plaxton).
3. Для витрины берём `items[0].photos` и `attributes`; для менеджера — аналоги с `min_price` и `offers`.

```
import requests
API, H = "https://kulisabase.ru/api/v1", {"Authorization": "Bearer YOUR_API_KEY"}

def analogs_and_fitment(number, brand=None):
    p = {"q": number, **({"brand": brand} if brand else {})}
    d = requests.get(API + "/card", params=p, headers=H, timeout=60).json()
    card = d["items"][0] if d["items"] else None
    analogs = [(a["brand"], a["number"], a["min_price"]) for a in d["analogs"]]
    fitment = {(r["make"], r["model"], r["engine"], r["years"]) for r in d["applicability"]}
    return card, analogs, sorted(fitment)

card, analogs, fitment = analogs_and_fitment("LF777", "FLEETGUARD")
print(card["name"], len(analogs), "аналогов,", len(fitment), "моделей техники")
```

Номера в любом написании находятся одинаково: `320/04133`, `320-04133`, `32004133` (масляный фильтр JCB — 41 кроссов).

B Подбор по VIN и по технике

Для автомобилей, LCV и грузовиков с 17-значным VIN метод `/vin` определяет модель и поколение и отдаёт список деталей. Пример: VIN `Z6FAXXESMAKC24396` → **TRANSIT V363 Фургон (FCD, FDD)**, 2019 г., уверенность 0,83, 40 модификаций, 49 841 деталей по категориям.

Для спецтехники тот же метод принимает PIN (ISO 10261): `JCB3CX4TE01234567` → JCB 3CX, `CAT0320DCKGG01234` → Caterpillar 320D, `KMTPC200C01012345` → Komatsu PC200 — с деталями по модели. Если код модели в PIN закрытый (Hitachi, Doosan и др.), ответ честно называет марку и подсказывает `/bycar`. Короткие заводские серийные номера старой техники — через `/bycar` по марке и модели или подбор от номера детали (сценарий A):

```
# техника без VIN: марка/модель → детали
r = requests.get(API + "/bycar", params={"make": "JCB", "model": "FASTTRAC", "limit": 100},
                 headers=H, timeout=90).json()
for p in r["parts"]:
    print(p["brand"], p["number"], p["name"], p["min_price"])
```

Покрытие применимости по маркам техники и двигателей сейчас (число связок «деталь ↔ модель» в базе, `/makes`):

МАРКА	СВЯЗОК
MAN	88 240 508
VOLVO	66 105 317
SCANIA	46 477 910
IVECO	24 487 908
DAF	13 960 455
JOHN DEERE	456 847
CUMMINS	188 674
KAMAZ	158 980
DEUTZ	127 113
NEW HOLLAND	103 242
PERKINS	40 774

МАРКА	СВЯЗОК
CATERPILLAR	19 035
JCB	3 761
LIEBHERR	2 310
KUBOTA	2 039
SHACMAN	1 854
SANY	1 158
YANMAR	810
XCMG	757
DOOSAN	311
HITACHI	50
KOMATSU	2

Число связок учитывает все модели марки; у Volvo, Iveco и MAN сюда входят и легковые/лёгкие коммерческие модели.

> Грузовики и двигатели (MAN, Scania, DAF, Iveco, Volvo, Cummins, Deutz, Perkins, John Deere) покрыты глубоко. По строительной и складской технике (JCB, Caterpillar, Liebherr, Komatsu, Doosan, Sany, XCMG) связок меньше — здесь ваши каталоги дадут наибольший эффект: для их приёма готов импорт (см. раздел 6).

С Проверить цену рынка

Проценка прайса или склада: отправляем список в `/bulk` (до 500 позиций за запрос) — по каждой строке минимальная цена рынка, число предложений и свежесть цены. Для LF777 Fleetguard: минимум **451 ₽**, **182 предложений**. Если актуальных цен нет, в `price.stale_last` приходит последняя известная цена с датой и источником.

Чтобы сразу получить рекомендацию по своей цене — `/reprice`: передаёте свою цену и закупку, получаете позицию относительно рынка и новую цену по правилу.

```
items = [{"brand": "FLEETGUARD", "number": "LF777", "price": 1500, "cost": 1100},
         {"number": "320/04133", "price": 2100}]
r = requests.post(API + "/reprice", headers=H, timeout=180,
                  json={"items": items, "rules": {"mode": "hold", "pct": 5, "min_margin":
15}}).json()
for x in r["results"]:
    print(x["number"], x["price"], "рынок:", x["market"], "→", x["new_price"], x["rec"])
```

РАЗДЕЛ 5

Тарифы и доступ

Публичные тарифы Kulisa Data (как на kulisabase.ru). Лимит — единицы запросов в месяц, стоимость единиц по методам — в разделе 2.

Демо

0 ₽

1 000 запросов · 14 дней

- все методы API
- песочница в кабинете
- проверка покрытия по вашим номерам

Старт

9 900 ₽/мес

50 000 запросов в месяц

- все методы API
- выгрузки и YML-фиды
- мониторинг цен и вебхуки

Бизнес

24 900 ₽/мес

250 000 запросов в месяц

- всё из «Старт»
- мониторинг до 1 000 артикулов
- помощь с подключением

Максимум

49 900 ₽/мес

1 500 000 запросов в месяц

- всё из «Бизнес»
- для крупных каталогов и сетей
- приоритетная поддержка

Оплата в кабинете картой, через СБП, SberPay или T-Pay; для юрлиц — по счёту (запрос на **info@kulisabase.ru**). Объёмы выше тарифа «Максимум», выделенные выгрузки и условия обмена данными обсуждаются индивидуально.

РАЗДЕЛ 6

Совместная работа с данными

Собственные каталоги по спецтехнике — применимость по моделям и серийным номерам, OE-номера, кроссы — подключаются к базе: после загрузки подбор по вашим маркам в `/applicability`, `/bycar`, `/models` и `/vin` становится точнее. Данные загружаются в отдельный раздел со ссылкой на источник, проверяются и только потом сливаются с общей применимостью; любую загрузку можно откатить целиком.

Какой формат удобен

Файлы **XLSX** (первый лист или указанный), **CSV** (UTF-8 или Windows-1251, разделитель «;», «,» или табуляция) или **JSON** (массив объектов). Одна строка — одна связка «техника ↔ деталь». Названия колонок могут быть русскими или английскими — распознаются автоматически; лишние колонки сохраняются как есть.

КОЛОНКА	ОБЯЗАТЕЛЬНО	ПРИМЕР
Номер детали (Артикул, Part number)	да	`320/04133`
Бренд детали (Производитель)	желательно	`JCB`, `FLEETGUARD` — если пусто, считаем номером производителя техники
Марка техники	да	`JCB`, `CATERPILLAR`, `KOMATSU`
Модель техники	да*	`3CX`, `320D`, `PC200-8`
VIN / PIN / серийный номер	да*	`JCB3CX4TE01234567`

КОЛОНКА	ОБЯЗАТЕЛЬНО	ПРИМЕР
Двигатель, годы выпуска	нет	`QSB6.7`, `2015-2019`
Наименование, узел/группа, количество, OE-номер	нет	`Фильтр масляный`, `Двигатель`, `1`

* Нужна модель или VIN/PIN; лучше — оба. Если в файле есть отдельный список машин «PIN ↔ модель», модель подставится в строки с тем же PIN автоматически. Кроссы — отдельным файлом: бренд · номер · бренд аналога · номер аналога · (необяз.) тип связи OE / аналог / замена.

Привязки к конкретным VIN/PIN машин остаются только у нас и наружу не отдаются; в общую применимость попадают связки «модель ↔ деталь».

РАЗДЕЛ 7

Контакты

KULISA DATA
АВТОМОБИЛЬНЫЕ ДАННЫЕ ДЛЯ БИЗНЕСА

Kulisa Data — база автозапчастей и API

Сайт и документация: **kulisabase.ru**
Регистрация и демо-доступ: **kulisabase.ru/register**
Почта: **info@kulisabase.ru**

Вопросы по подключению, тарифам и обмену данными — на почту; ответ в рабочее время.