



Документация MarkClean API

Версия API: v1

Базовый URL: `https://markclean.online/api`

Формат данных: JSON

Аутентификация: API-ключ (`X-API-Key` / `Authorization: Bearer`) для `/convert` и `/converter/history` ; HTTP-сессия (`cookie session_id`) для учётной, API-key и биллинга



Содержание

1. [Обзор](#)
2. [Получение API-ключа](#)
3. [Аутентификация](#)
4. [Конвертация URL в Markdown](#)
5. [История конвертаций](#)
6. [Опции конвертации](#)
7. [API-ключи](#)
8. [Биллинг: пакеты и подписки](#)
9. [Проверка баланса](#)
10. [Обработка ошибок](#)
11. [Квоты, лимиты и кэширование](#)
12. [Примеры кода](#)
13. [FAQ](#)

Обзор

MarkClean API позволяет программно конвертировать веб-страницы в чистый Markdown, оптимизированный для LLM, RAG-систем и документации.

Ключевые возможности:

-  Конвертация любой HTML-страницы в Markdown
-  Автоматическая очистка от мусора (реклама, навигация, футеры) через Readability
-  Двухуровневое кэширование — повторные запросы не списывают кредиты
-  Гибкие опции вывода (заголовки, frontmatter, оглавление, удаление картинок/ссылок)
-  История всех конвертаций с дедупликацией
-  Система кредитов + ежедневный лимит + подписка
-  Управление API-ключами (создание, чтение, удаление)
-  Пакеты кредитов и подписки

Получение API-ключа

API-ключ можно создать **только** через веб-интерфейс (требуется авторизация по сессии).

Шаги:

1. Войдите в аккаунт на markclean.online
2. Перейдите на страницу **API Keys** (или откройте `/api-keys`)
3. Нажмите **"Сгенерировать ключ"**
4.  **ВАЖНО:** Сохраните ключ немедленно — он будет показан **только один раз!**

Формат ключа

```
mk_live_Xk9mP2vQ7rT4wL8nZ3sY6tB1cD5fG0hJ
|
└─ 32 символа base64url (криптографически случайные, 24 байта)
└─ префикс (8 символов)
```

Общая длина: 40 символов

 **Безопасность:** В базе данных хранится только SHA-256 хеш ключа. Если вы потеряете ключ — восстановить его невозможно, только создать новый.

Аутентификация

API поддерживает два независимых способа авторизации:

1. API-ключ (для API-клиентов)

Все запросы к защищённым эндпоинтам должны содержать API-ключ в одном из заголовков:

Способ 1: X-API-Key (рекомендуется)

```
X-API-Key: mk_live_Xk9mP2vQ7rT4wL8nZ3sY6tB1cD5fG0hJ
```

Способ 2: Authorization: Bearer

```
Authorization: Bearer mk_live_Xk9mP2vQ7rT4wL8nZ3sY6tB1cD5fG0hJ
```

Оба способа равнозначны. При отсутствии или невалидности ключа сервер вернёт `401 Unauthorized` с ошибкой `invalid_api_key`.

2. Сессия (для веб-интерфейса и защищённых эндпоинтов)

Для веб-интерфейса и защищённых эндпоинтов (API-ключи, биллинг, история, профиль) используется HTTP-сессия:

- Cookie: `session_id` (UUID)
- Устанавливается при регистрации/логине
- Живёт 30 дней
- Передаётся автоматически браузером

Конвертация URL в Markdown

Основной эндпоинт API. Доступен **анонимам** и **пользователям с API-ключом**.

Запрос

```
POST /api/convert
Content-Type: application/json
X-API-Key: mk_live_YOUR_KEY_HERE
```

Примечание: Для анонимов API-ключ не нужен, но в этом случае применяется IP-лимит (3 запроса/24 часа).

Тело запроса

```
{
  "url": "https://example.com/article",
  "include_title": true,
  "include_frontmatter": false,
  "include_toc": false,
  "strip_images": false,
  "strip_links": false
}
```

Поле	Тип	Обязательное	По умолчанию	Описание
url	string		—	URL веб-страницы для конвертации (http/https)
include_title	bool		false	Добавить заголовок страницы как H1 (# Title)
include_frontmatter	bool		false	Добавить YAML-шапку с метаданными

Поле	Тип	Обязательное	По умолчанию	Описание
include_toc	bool	✗	false	Добавить оглавление (Table of Contents)
strip_images	bool	✗	false	Удалить все теги
strip_links	bool	✗	false	Удалить <a> теги (текст сохраняется)

Успешный ответ (200 ОК)

```
{
  "title": "Example Article",
  "content": "# Example Article\n\nThis is the converted content...",
  "metadata": {
    "words": 4521,
    "tokens": 1234,
    "bytes": 4521
  },
  "quota_incremented": true
}
```

Поле	Тип	Описание
title	string	Заголовок страницы (может отсутствовать)
content	string	Результат конвертации в Markdown
metadata.words	int	Количество слов в контенте
metadata.tokens	int	Количество токенов (для LLM, cl100k)
metadata.bytes	int	Размер в байтах
quota_incremented	bool	true — списан кредит; false — cache hit или лимит

Пример cURL

```
curl -X POST https://markclean.online/api/convert \
-H "Content-Type: application/json" \
-H "X-API-Key: mk_live_YOUR_KEY_HERE" \
-d '{
  "url": "https://metanit.com/web/html5/2.6.php",
  "include_title": true,
  "include_frontmatter": true,
  "strip_images": true,
  "strip_links": false
}'
```

История конвертаций

Получить список всех конвертаций текущего пользователя с пагинацией.

Требует авторизации (API-ключ или сессия).

Запрос

```
GET /api/converter/history?page=1&page_size=20
X-API-Key: mk_live_YOUR_KEY_HERE
```

Параметр	Тип	По умолчанию	Описание
page	int	1	Номер страницы (≥ 1)
page_size	int	20	Количество записей (1–100)

Успешный ответ (200 OK)

```
{
  "items": [
    {
      "id": "27c1e865-a77e-4d99-87d0-1179aef2a3e6",
      "title": "Example Article",
```

```
    "target_url": "https://example.com/article",
    "updated_at": "2026-09-27T14:30:00Z"
  }
],
"total_count": 42,
"page": 1,
"page_size": 20,
"has_more": true
}
```

Поле	Тип	Описание
items	array	Список конвертаций (отсортированы по updated_at DESC)
items[].id	string	UUID записи
items[].title	string	Заголовок страницы (или пустая строка)
items[].target_url	string	Исходный URL
items[].updated_at	string	Последняя конвертация (ISO 8601)
total_count	int	Общее количество записей (только успешные, status_code = 200)
page	int	Текущая страница
page_size	int	Размер страницы
has_more	bool	Есть ли следующие страницы

💡 **Дедупликация:** Один URL = одна запись. Повторные конвертации обновляют updated_at и перемещают запись наверх.



Опции конвертации

`include_title: true`

Добавляет заголовок страницы как H1 в начало документа:

Example Article

This is the content...

include_frontmatter: true

Добавляет YAML-шапку с метаданными (удобно для статических генераторов типа Hugo/Jekyll):

```
---
title: "Example Article"
source: "https://example.com/article"
language: "English"
words: 4521
tokens: 1234
bytes: 4521
reading_time: "23 min"
extracted_at: "2026-09-27"
---
```

Example Article

This is the content...

Поле `language` определяется автоматически через `whatlanggo` (только если включён `include_toc` или `include_frontmatter`).

include_toc: true

Добавляет оглавление (Table of Contents) из заголовков H1–H3:

Table of Contents

- `[Main Section](<#Main Section>)`
- `[Subsection](<#Subsection>)`

Main Section

Content...

 Оглавление добавляется только если заголовков H1–H3 **не менее 2**. Иначе оглавление не включается.

`strip_images: true`

Полностью удаляет все `<img>` теги. Полезно для:

-  Подготовки текста для LLM (экономия токенов)
-  Создания текстовых версий статей
-  Уменьшения размера результата

`strip_links: true`

Удаляет `<a>` теги, но **сохраняет текст ссылок**:

Было:

```
Visit our site for more info.
```

Стало:

```
Visit our site for more info.
```

Рекомендуемые комбинации

Сценарий	Опции
Для LLM / RAG	<code>strip_images: true , strip_links: true</code>
Для документации	<code>include_title: true , include_frontmatter: true</code>
Для архивирования	<code>include_title: true , include_frontmatter: true , strip_images: false</code>
Минимальный размер	<code>strip_images: true , strip_links: true , include_title: false</code>

API-ключи

Управление API-ключами — **только через сессию** (веб-интерфейс). API-ключами нельзя управлять через API.

Создание ключа

POST /api/keys

Запрос:

```
{
  "name": "My Key"
}
```

name необязательно (макс. 50 символов). Если не указан, используется "Default Key" .

Ответ (201 Created):

```
{
  "id": "27c1e865-a77e-4d99-87d0-1179aef2a3e6",
  "name": "My Key",
  "api_key": "mk_live_Xk9mP2vQ7rT4wL8nZ3sY6tB1cD5fG0hJ",
  "warning": "Save this key now. It will not be shown again."
}
```

★ Сырой ключ возвращается **только в этом ответе**. Больше его никто не покажет.

Ошибки:

- 409 Conflict + key_already_exists — ключ уже существует (у пользователя только 1 ключ)
- 500 Internal Server Error + internal_error

Получение метаданных ключа

GET /api/keys

Ответ (200 OK):

```
{
  "id": "27c1e865-a77e-4d99-87d0-1179aef2a3e6",
  "name": "My Key",
  "key_prefix": "mk_live_a1b2c3d4",
  "last_used_at": "2026-09-27 10:30:00",
  "is_active": true,
  "created_at": "2026-09-27 09:00:00"
}
```

Ошибки:

- 404 Not Found + key_not_found — у пользователя нет ключа

Удаление ключа

DELETE /api/keys/:key_id

Ответ (200 OK):

```
{
  "message": "API key deleted successfully"
}
```

Ошибки:

- 400 Bad Request + invalid_key_id — некорректный UUID
- 404 Not Found + key_not_found — ключ не найден
- 500 Internal Server Error + internal_error



Биллинг: пакеты и подписки

Получение доступных пакетов

GET /api/billing/packages — **публичный** эндпоинт (авторизация не требуется).

Ответ (200 OK):

```
{
  "free": {
    "name": "Free",
    "daily_limit": 10
  },
  "credits": [
    {
      "code": "micro",
      "name": "Micro",
      "credits": 300,
      "price": 49,
      "price_per_100_requests": 17,
    }
  ]
}
```


Поле	Описание
free	Бесплатный план: дневной лимит
credits[]	Пакеты кредитов (от дешёвого к дорогому)
credits[].code	Код пакета (micro , start , optimal , pro , ultra)
credits[].credits	Количество кредитов в пакете
credits[].price	Цена в рублях
credits[].price_per_100_requests	Стоимость 100 запросов
credits[].discount	Процент выгоды (от базового тарифа Micro)
credits[].savings_rub	Экономия в рублях
credits[].is_popular	Флаг «Хит»
subscriptions[]	Пакеты подписки (от короткого к длинному)

Покупка пакета

POST /api/billing/purchase — **требуется авторизации** (API-ключ или сессия).

 **Временный endpoint!** В продакшене будет заменён на реальную интеграцию с платёжной системой.

Запрос:

```
{
  "package_code": "optimal"
}
```

Ответ (200 OK):

```
{
  "message": "Credits added successfully",
  "package": "optimal",
  "credits_added": 3000,
  "new_balance": 3000
}
```

Ошибки:

- 400 Bad Request + invalid_package — некорректный код пакета
- 401 Unauthorized + unauthorized — нет аутентификации
- 500 Internal Server Error + purchase_failed — ошибка начисления кредитов



Проверка баланса

Получение баланса кредитов и подписки

GET /api/billing/usage — **требуется авторизации** (API-ключ или сессия).

Ответ (200 OK):

```
{
  "credits": {
    "credits": 2500,
    "daily_used": 3,
    "daily_limit": 10,
    "daily_remaining": 7,
    "has_credits": true,
    "daily_reset_at": "2026-09-27T14:00:00Z"
  },
  "subscription": {
    "is_active": true,
    "ends_at": "2026-10-01T10:00:00Z",
    "days_remaining": 4,
    "is_unlimited": true
  },
}
```

```
"is_subscriber": true
}
```

Поле	Тип	Описание
credits	object	Баланс кредитов
credits.credits	int	Платные кредиты (не сгорают)
credits.daily_used	int	Использовано запросов сегодня
credits.daily_limit	int	Дневной лимит (10)
credits.daily_remaining	int	Осталось запросов сегодня
credits.has_credits	bool	Есть ли платные кредиты
credits.daily_reset_at	time	Время следующего сброса дневного лимита
subscription	object/null	Информация об активной подписке
subscription.is_active	bool	Подписка активна
subscription.ends_at	time	Дата окончания подписки
subscription.days_remaining	int	Осталось дней
subscription.is_unlimited	bool	Безлимит (всегда true)
is_subscriber	bool	Есть ли активная подписка

Обработка ошибок

Все ошибки возвращаются в едином формате:

```
{
  "error": "machine_readable_code",
  "message": "Human-readable description"
}
```

HTTP-статусы

Статус	Код ошибки	Описание	Действие
400	invalid_payload	Битый JSON или отсутствует url	Проверьте тело запроса
400	invalid_url	Некорректный формат URL	Должен начинаться с http:// или https://
400	invalid_request	Некорректный запрос (биллинг)	Проверьте параметры
400	invalid_package	Неверный код пакета	Используйте валидный код
401	invalid_api_key	Ключ отсутствует, невалиден или удалён	Проверьте заголовок X-API-Key
401	unauthorized	Нет аутентификации	Передайте API-ключ или cookie
402	quota_exceeded	Кредиты закончились + дневной лимит	Купите кредиты в веб-интерфейсе
404	key_not_found	Ключ не найден	Создайте ключ заново
404	key_not_found	Ключ пользователя не найден (GET /api/keys)	Создайте ключ
404	user_not_found	Пользователь не найден	Проверьте данные
409	key_already_exists	У пользователя уже есть API-ключ	Удалите существующий ключ
409	user_exists	Пользователь с этим email уже зарегистрирован	Используйте другой email
409	account_already_deleted	Аккаунт уже удалён	—
410	email_previously_deleted	Email был использован для удалённого аккаунта	Используйте другой email
415	unsupported_content	URL не возвращает HTML	API работает только с веб-страницами

Статус	Код ошибки	Описание	Действие
422	conversion_failed	Не удалось распознать контент	Попробуйте другой URL
429	rate_limit_exceeded	Превышен лимит запросов	Подождите или войдите в аккаунт
429	registration_ip_limit	Превышен лимит регистраций с IP	Подождите
429	resend_cooldown	Слишком часто запрашивается OTP	Подождите
429	otp_blocked	Слишком много неудачных попыток OTP	Запросите новый код
429	registration_ip_limit	Превышен лимит регистраций с IP	Подождите
429	registration_ip_limit	Превышен лимит регистраций с IP	Подождите
500	internal_server_error	Внутренняя ошибка сервера	Повторите позже
502	fetch_failed	Не удалось загрузить страницу	Сайт недоступен или заблокировал запрос

Пример обработки ошибки

```
$ curl -X POST https://markclean.online/api/convert \
  -H "X-API-Key: mk_live_INVALID" \
  -H "Content-Type: application/json" \
  -d '{"url": "https://example.com"}'

HTTP/1.1 401 Unauthorized
{
  "error": "invalid_api_key",
  "message": "Invalid or expired API key"
}
```



Квоты, лимиты и кэширование

Система кредитов

-  **Платные кредиты** — не сгорают, не имеют срока действия
-  **Дневной лимит** (10 запросов/сутки) — сбрасывается каждые 24 часа
-  **Приоритет списания:** сначала платные кредиты, затем дневной лимит
-  **Максимальный баланс:** 500 000 кредитов

Кэширование и экономия

API использует **двухуровневое кэширование**:

Уровень	Что кэшируется	TTL	Экономия
L1	Сырой HTML	30 мин	Нет HTTP-запроса к внешнему сайту
L2	Финальный Markdown	15 мин	Мгновенный ответ

Важно: Кредиты списываются **только** при реальном HTTP-запросе к внешнему сайту. Повторные конвертации того же URL (даже с другими опциями) **бесплатны**, если HTML ещё в кэше.

Таблица сценариев

Сценарий	Fetch	Кредиты	Скорость
Первый запрос URL		-1	~2-5 сек
Повтор с теми же опциями		0	~50 мс
Повтор с другими опциями		0	~200 мс
Спустя 30+ минут		-1	~2-5 сек

IP-лимиты (для анонимов)

- **Конвертация:** 3 запроса / 24 часа с одного IP
- **Регистрация:** 3 регистрации / 1 час с одного IP (глобально), 3 регистрации / 2 минуты с одного IP

- **Логин:** 10 попыток / 30 минут с одного IP, 5 попыток / 15 минут на email
- **Recovery OTP:** 10 попыток / 1 час с одного IP
- **OTP Verify:** 20 попыток / 5 минут с одного IP
- **Смена пароля:** 5 попыток / 1 час с одного IP, 3 попытки / 1 час на пользователя

Проверка баланса

Для проверки баланса кредитов используйте:

- Веб-интерфейс (страница **Billing**)
- API: **GET** /api/billing/usage (требуется авторизация)



Примеры кода

Python (requests)

```
import requests

API_KEY = "mk_live_YOUR_KEY_HERE"
API_BASE = "https://markclean.online/api"

headers = {
    "Content-Type": "application/json",
    "X-API-Key": API_KEY,
}

# Конвертация
response = requests.post(
    f"{API_BASE}/convert",
    headers=headers,
    json={
        "url": "https://example.com/article",
        "include_title": True,
        "include_frontmatter": True,
        "strip_images": True,
        "strip_links": False,
```

```

    },
)

if response.ok:
    data = response.json()
    print(f"Title: {data.get('title')}")
    print(f"Words: {data['metadata']['words']}")
    print(f"Tokens: {data['metadata']['tokens']}")
    print(f"Credits used: {data['quota_incremented']}")
    print(data["content"][:500])
else:
    error = response.json()
    print(f"Error {response.status_code}: {error['error']} - {error['message']}")

```

Python (пакетная конвертация)

```

import requests
import time

API_KEY = "mk_live_YOUR_KEY_HERE"
urls = [
    "https://example.com/page1",
    "https://example.com/page2",
    "https://example.com/page3",
]

headers = {
    "Content-Type": "application/json",
    "X-API-Key": API_KEY,
}

results = []
for url in urls:
    resp = requests.post(
        "https://markclean.online/api/convert",
        headers=headers,
        json={"url": url, "include_title": True},
    )

```

```

    )
    if resp.ok:
        results.append(resp.json())
        print(f"✅ {url}")
    else:
        print(f"❌ {url}: {resp.json()['error']}")
    time.sleep(0.5) # Вежливый delay

# Сохраняем в один файл
with open("output.md", "w", encoding="utf-8") as f:
    for r in results:
        f.write(r["content"] + "\n\n---\n\n")

```

JavaScript / Node.js (fetch)

```

const API_KEY = 'mk_live_YOUR_KEY_HERE';
const API_BASE = 'https://markclean.online/api';

async function convertUrl(url, options = {}) {
    const response = await fetch(`${API_BASE}/convert`, {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
            'X-API-Key': API_KEY,
        },
        body: JSON.stringify({
            url,
            include_title: options.title ?? false,
            include_frontmatter: options.frontmatter ?? false,
            include_toc: options.toc ?? false,
            strip_images: options.stripImages ?? false,
            strip_links: options.stripLinks ?? false,
        }),
    });

    if (!response.ok) {
        const error = await response.json();
        throw new Error(`${error.error}: ${error.message}`);
    }
}

```



```

    IncludeTOC      bool    `json:"include_toc"`
    StripImages     bool    `json:"strip_images"`
    StripLinks      bool    `json:"strip_links"`
}

type Metadata struct {
    Words  int `json:"words"`
    Tokens int `json:"tokens"`
    Bytes  int `json:"bytes"`
}

type ConvertResponse struct {
    Title          string `json:"title"`
    Content        string `json:"content"`
    Metadata       Metadata `json:"metadata"`
    QuotaIncremented bool    `json:"quota_incremented"`
}

func main() {
    reqBody, _ := json.Marshal(ConvertRequest{
        URL:          "https://example.com/article",
        IncludeTitle: true,
        IncludeFrontmatter: true,
        StripImages:  true,
        StripLinks:   false,
    })

    req, _ := http.NewRequest("POST", apiBase+"/convert",
bytes.NewReader(reqBody))
    req.Header.Set("Content-Type", "application/json")
    req.Header.Set("X-API-Key", apiKey)

    resp, err := http.DefaultClient.Do(req)
    if err != nil {
        panic(err)
    }
    defer resp.Body.Close()

```

```

body, _ := io.ReadAll(resp.Body)

if resp.StatusCode != http.StatusOK {
    fmt.Printf("Error %d: %s\n", resp.StatusCode, body)
    return
}

var result ConvertResponse
json.Unmarshal(body, &result)
fmt.Println(result.Content)
}

```

cURL (все сценарии)

Базовая конвертация

```

curl -X POST https://markclean.online/api/convert \
  -H "X-API-Key: mk_live_YOUR_KEY" \
  -H "Content-Type: application/json" \
  -d '{"url": "https://example.com"}'

```

Для LLM (чистый текст)

```

curl -X POST https://markclean.online/api/convert \
  -H "X-API-Key: mk_live_YOUR_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "url": "https://example.com",
    "strip_images": true,
    "strip_links": true
  }'

```

Для документации (с метаданными)

```

curl -X POST https://markclean.online/api/convert \
  -H "X-API-Key: mk_live_YOUR_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "url": "https://example.com",
    "include_title": true,
    "include_frontmatter": true
  }'

```

```
}'  
  
# Для документации (с оглавлением)  
curl -X POST https://markclean.online/api/convert \  
-H "X-API-Key: mk_live_YOUR_KEY" \  
-H "Content-Type: application/json" \  
-d '{  
  "url": "https://example.com",  
  "include_title": true,  
  "include_frontmatter": true,  
  "include_toc": true  
'  
  
# История конвертаций  
curl -X GET "https://markclean.online/api/converter/history?  
page=1&page_size=10" \  
-H "X-API-Key: mk_live_YOUR_KEY"  
  
# Проверка баланса  
curl -X GET "https://markclean.online/api/billing/usage" \  
-H "X-API-Key: mk_live_YOUR_KEY"  
  
# Получить доступные пакеты  
curl -X GET "https://markclean.online/api/billing/packages"
```

? FAQ

🔒 Безопасно ли передавать API-ключ?

Да, если соблюдаете правила:

-  Используйте HTTPS (всегда)
-  Храните ключ в переменных окружения, не в коде
-  Не логируйте ключ
-  Не коммитьте ключ в Git
-  Не передавайте ключ в URL-параметрах

Что делать, если ключ скомпрометирован?

1. Откройте `/api-keys` в веб-интерфейсе
2. Нажмите "**Пересоздать**" (старый ключ будет удалён, создан новый)
3. Обновите ключ во всех ваших приложениях

 Старый ключ перестает работать мгновенно после пересоздания.

Можно ли использовать API без кредитов?

Да, у каждого пользователя есть **бесплатный дневной лимит** (10 запросов/сутки), который сбрасывается каждые 24 часа. После исчерпания и дневного лимита, и кредитов — API вернёт `402 quota_exceeded`.

Почему `quota_incremented: false` ?

Это нормально в трёх случаях:

1. **Cache hit** — этот URL уже конвертировался недавно (бесплатно)
2. **Анонимный запрос** — без API-ключа (лимит по IP)
3. **Подписка** — у пользователя активна подписка, кредиты не списываются

Работает ли API с любым сайтом?

API работает с любыми **публично доступными HTML-страницами**. Ограничения:

-  Сайты, требующие авторизации
-  Сайты с жёсткой анти-бот защитой (Cloudflare challenge и т.п.)
-  PDF, изображения и другие не-HTML форматы
-  Сайты, блокирующие по User-Agent

Есть ли лимит на размер страницы?

- Максимум: **5 MB** контента после конвертации
- Типичные статьи: 10-100 KB
- Длинные технические документы: 200-500 KB

Страницы больше 5 MB не кэшируются, но всё равно конвертируются.

Как использовать API-ключ с другими инструментами?

Postman:

- Вкладка **Authorization** → Type: **API Key**
- Key: X-API-Key , Value: ваш ключ, Add to: **Header**

Insomnia:

- Request → **Auth** → **API Key**
- Header name: X-API-Key

VS Code (REST Client):

```
@apiKey = mk_live_YOUR_KEY
```

```
POST https://markclean.online/api/convert
```

```
Content-Type: application/json
```

```
X-API-Key: {{apiKey}}
```

```
{  
  "url": "https://example.com"  
}
```

Что будет с API-ключом после удаления аккаунта?

Ключ продолжит работать, пока не закончатся кредиты. После удаления аккаунта:

-  API-ключ остаётся активным
-  Существующие кредиты можно использовать
-  Пополнить кредиты нельзя (нет доступа к UI)
-  Создать новый ключ нельзя

Это сделано для честности: вы оплатили кредиты — вы их используете.

Есть ли SDK?

Официальных SDK пока нет, но API достаточно простой для использования через стандартные HTTP-клиенты в любом языке. Примеры выше покрывают самые популярные языки.

Поддержка

-  Email: support@markclean.online
-  Email: gnutenko.s.r@gmail.com
-  Telegram: [@Slayder12](https://t.me/Slayder12)
-  Changelog: markclean-doc/changelog.md