



MarkClean API documentation

API Version: v1

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

Data Format: JSON

Authentication: API key (X-API-Key / Authorization: Bearer) for /convert and /converter/history ; HTTP session (cookie `session_id`) for account, API-key and billing endpoints



Table of Contents

1. [Overview](#)
2. [Getting an API Key](#)
3. [Authentication](#)
4. [Converting URL to Markdown](#)
5. [Conversion History](#)
6. [Conversion Options](#)
7. [API Keys](#)
8. [Billing: Credit Packs & Subscriptions](#)
9. [Balance Check](#)
10. [Error Handling](#)
11. [Quotas, Limits & Caching](#)
12. [Code Examples](#)
13. [FAQ](#)

Overview

The MarkClean API allows you to programmatically convert web pages into clean Markdown, optimized for LLMs, RAG systems, and documentation.

Key Features:

-  Convert any HTML page to Markdown
-  Automatic junk removal (ads, navigation, footers) via Readability
-  Two-tier caching — repeated requests do not consume credits
-  Flexible output options (title, frontmatter, table of contents, image/link stripping)
-  Full history of all conversions with de-duplication
-  Credit system + daily limit + subscriptions
-  API key management (create, read, delete)
-  Credit packs and subscriptions

Getting an API Key

An API key can be created **only** through the web interface (session authentication required).

Steps:

1. Log in to your account on markclean.online
2. Navigate to the **API Keys** page (or open `/api-keys`)
3. Click "**Generate Key**"
4.  **IMPORTANT:** Save the key immediately — it will be shown **only once!**

Key Format

```
mk_live_Xk9mP2vQ7rT4wL8nZ3sY6tB1cD5fG0hJ
|
└─ 32 base64url characters (cryptographically random, 24 bytes)
└─ prefix (8 characters)
```

Total length: 40 characters

 **Security:** Only the SHA-256 hash of the key is stored in the database. If you lose the key — it cannot be recovered, only a new one can be created.



Authentication

The API supports two independent authentication methods:

1. API Key (for API clients)

All requests to protected endpoints must include an API key in one of the following headers:

Method 1: X-API-Key (recommended)

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

Method 2: Authorization: Bearer

```
Authorization: Bearer mk_live_Xk9mP2vQ7rT4wL8nZ3sY6tB1cD5fG0hJ
```

Both methods are equivalent. If the key is missing or invalid, the server returns `401 Unauthorized` with error `invalid_api_key`.

2. Session (for the web interface and protected endpoints)

For the web interface and protected endpoints (API keys, billing, history, profile), an HTTP session is used:

- Cookie: `session_id` (UUID)
- Set on registration/login
- Expires after 30 days
- Sent automatically by the browser

Converting URL to Markdown

The main API endpoint. Available to **anonymous users** and **users with an API key**.

Request

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

Note: Anonymous users do not need an API key, but in that case the IP limit applies (3 requests / 24 hours).

Request Body

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

Field	Type	Required	Default	Description
url	string		—	URL of the web page to convert (http/https)
include_title	bool		false	Add the page title as an H1 (# Title)
include_frontmatter	bool		false	Add a YAML frontmatter with metadata
include_toc	bool		false	Add a table of contents (TOC)
strip_images	bool		false	Remove all tags
strip_links	bool		false	Remove <a> tags (text is preserved)

Successful Response (200 OK)

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

Field	Type	Description
title	string	Page title (may be absent)
content	string	Markdown conversion result
metadata.words	int	Number of words in the content
metadata.tokens	int	Number of tokens (for LLM, cl100k)
metadata.bytes	int	Size in bytes
quota_incremented	bool	true — a credit was consumed; false — cache hit or limit reached

cURL Example

```
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
}'
```

Conversion History

Retrieve the list of all conversions for the current user with pagination.

Authentication required (API key or session).

Request

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

Parameter	Type	Default	Description
page	int	1	Page number (≥ 1)
page_size	int	20	Number of records (1–100)

Successful Response (200 OK)

```
{
  "items": [
    {
      "id": "27cle865-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
}
```

Field	Type	Description
items	array	List of conversions (sorted by <code>updated_at</code> DESC)

Field	Type	Description
<code>items[].id</code>	string	Record UUID
<code>items[].title</code>	string	Page title (or empty string)
<code>items[].target_url</code>	string	Source URL
<code>items[].updated_at</code>	string	Last conversion (ISO 8601)
<code>total_count</code>	int	Total number of records (successful only, <code>status_code = 200</code>)
<code>page</code>	int	Current page
<code>page_size</code>	int	Page size
<code>has_more</code>	bool	Whether there are more pages



Deduplication: One URL = one record. Repeated conversions update `updated_at` and move the record to the top.



Conversion Options

`include_title: true`

Adds the page title as an H1 at the beginning of the document:

```
# Example Article
```

```
This is the content...
```

`include_frontmatter: true`

Adds a YAML frontmatter with metadata (convenient for static site generators such as 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...

The `language` field is detected automatically via `whatlanggo` (only when `include_toc` or `include_frontmatter` is enabled).

`include_toc: true`

Adds a table of contents (TOC) from H1–H3 headings:

Table of Contents

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

Main Section

Content...



The TOC is added only if there are **at least 2** H1–H3 headings. Otherwise, the TOC is not included.

`strip_images: true`

Completely removes all `<img>` tags. Useful for:

-  Preparing text for LLMs (token savings)
-  Creating text-only versions of articles
-  Reducing the result size

`strip_links: true`

Removes `<a>` tags, but **preserves the link text**:

Before:

```
Visit <a href="https://example.com">our site</a> for more info.
```

After:

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

Recommended Combinations

Scenario	Options
For LLM / RAG	<code>strip_images: true, strip_links: true</code>
For documentation	<code>include_title: true, include_frontmatter: true</code>
For archiving	<code>include_title: true, include_frontmatter: true, strip_images: false</code>
Minimal size	<code>strip_images: true, strip_links: true, include_title: false</code>

API Keys

Managing API keys is **session-only** (web interface). API keys cannot be managed via the API.

Creating a Key

POST `/api/keys`

Request:

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

name is optional (max. 50 characters). If not specified, "Default Key" is used.

Response (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."
}
```

★ The raw key is returned **only in this response**. No one will ever show it again.

Errors:

- 409 Conflict + key_already_exists — a key already exists (a user has only 1 key)
- 500 Internal Server Error + internal_error

Getting Key Metadata

GET /api/keys

Response (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"
}
```

Errors:

- 404 Not Found + key_not_found — the user has no key

Deleting a Key

DELETE /api/keys/:key_id

Response (200 OK):

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

Errors:

- 400 Bad Request + invalid_key_id — invalid UUID
- 404 Not Found + key_not_found — key not found
- 500 Internal Server Error + internal_error



Billing: Credit Packs & Subscriptions

Getting Available Packages

GET /api/billing/packages — **public** endpoint (no authentication required).

Response (200 OK):

```
{
  "free": {
    "name": "Free",
    "daily_limit": 10
  },
  "credits": [
    {
      "code": "micro",
```



```
    "savings_rub": 1198
  }
]
```

Field	Description
free	Free plan: daily limit
credits[]	Credit packs (from cheapest to most expensive)
credits[].code	Package code (micro , start , optimal , pro , ultra)
credits[].credits	Number of credits in the pack
credits[].price	Price in rubles
credits[].price_per_100_requests	Cost of 100 requests
credits[].discount	Discount percentage (compared to the base Micro rate)
credits[].savings_rub	Savings in rubles
credits[].is_popular	"Hit" flag
subscriptions[]	Subscription packs (from shortest to longest)

Purchasing a Pack

POST /api/billing/purchase — **authentication required** (API key or session).

 **Temporary endpoint!** In production it will be replaced with a real payment system integration.

Request:

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

Response (200 OK):

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

Errors:

- 400 Bad Request + `invalid_package` — invalid package code
- 401 Unauthorized + `unauthorized` — no authentication
- 500 Internal Server Error + `purchase_failed` — error while crediting



Balance Check

Getting Credits & Subscription Balance

GET `/api/billing/usage` — **authentication required** (API key or session).

Response (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
}
```

Field	Type	Description
credits	object	Credit balance
credits.credits	int	Paid credits (do not expire)
credits.daily_used	int	Requests used today
credits.daily_limit	int	Daily limit (10)
credits.daily_remaining	int	Requests remaining today
credits.has_credits	bool	Whether paid credits are available
credits.daily_reset_at	time	Next daily limit reset time
subscription	object/null	Active subscription info
subscription.is_active	bool	Subscription is active
subscription.ends_at	time	Subscription end date
subscription.days_remaining	int	Days remaining
subscription.is_unlimited	bool	Unlimited (always true)
is_subscriber	bool	Whether there is an active subscription

Error Handling

All errors are returned in a unified format:

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

HTTP Statuses

Status	Error Code	Description	Action
400	invalid_payload	Invalid JSON or missing url	Check the request body
400	invalid_url	Invalid URL format	Must start with http:// or https://
400	invalid_request	Invalid request (billing)	Check the parameters
400	invalid_package	Invalid package code	Use a valid code
401	invalid_api_key	Key missing, invalid, or deleted	Check the X-API-Key header
401	unauthorized	No authentication	Pass an API key or cookie
402	quota_exceeded	Credits exhausted + daily limit reached	Purchase credits in the web interface
404	key_not_found	Key not found	Create a new key
404	key_not_found	User's key not found (GET /api/keys)	Create a key
404	user_not_found	User not found	Check your credentials
409	key_already_exists	User already has an API key	Delete the existing key
409	user_exists	A user with this email is already registered	Use a different email
409	account_already_deleted	Account already deleted	—
410	email_previously_deleted	Email was used for a deleted account	Use a different email
415	unsupported_content	URL does not return HTML	The API works only with web pages
422	conversion_failed	Could not recognize the content	Try another URL
429	rate_limit_exceeded	Request limit exceeded	Wait or log in to your account

Status	Error Code	Description	Action
429	registration_ip_limit	Registration limit from the IP exceeded	Wait
429	resend_cooldown	OTP requested too frequently	Wait
429	otp_blocked	Too many failed OTP attempts	Request a new code
429	registration_ip_limit	Registration limit from the IP exceeded	Wait
429	registration_ip_limit	Registration limit from the IP exceeded	Wait
500	internal_server_error	Server internal error	Retry later
502	fetch_failed	Could not load the page	Site unavailable or blocked the request

Error Handling Example

```
$ 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"
}
```

Quotas, Limits & Caching

Credit System

-  **Paid credits** — do not expire, no expiration date
-  **Daily limit** (10 requests / day) — resets every 24 hours
- ⚡ **Debit priority:** paid credits first, then the daily limit
-  **Maximum balance:** 500,000 credits

Caching & Savings

The API uses **two-tier caching**:

Level	What is cached	TTL	Savings
L1	Raw HTML	30 min	No HTTP request to the external site
L2	Final Markdown	15 min	Instant response

Important: Credits are debited **only** when an actual HTTP request is made to the external site. Repeated conversions of the same URL (even with different options) are **free** if the HTML is still in the cache.

Scenario Table

Scenario	Fetch	Credits	Speed
First request for a URL	✓	-1	~2-5 sec
Repeat with the same options	✗	0	~50 ms
Repeat with different options	✗	0	~200 ms
After 30+ minutes	✓	-1	~2-5 sec

IP Limits (for anonymous users)

- **Conversion:** 3 requests / 24 hours per IP
- **Registration:** 3 registrations / 1 hour per IP (globally), 3 registrations / 2 minutes per IP
- **Login:** 10 attempts / 30 minutes per IP, 5 attempts / 15 minutes per email

- **Recovery OTP:** 10 attempts / 1 hour per IP
- **OTP Verify:** 20 attempts / 5 minutes per IP
- **Password change:** 5 attempts / 1 hour per IP, 3 attempts / 1 hour per user

Balance Check

To check your credit balance, use:

- Web interface (page **Billing**)
- API: **GET** /api/billing/usage (authentication required)

Code Examples

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,
}

# Conversion
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 (batch conversion)

```

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) # Polite delay

# Save to a single file
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}`);
    }
}

```



```

    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 (all scenarios)

Basic conversion

```

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"}'

```

For LLM (plain text)

```

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
}'

```

For documentation (with metadata)

```

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
}'

```

```
# For documentation (with TOC)
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
  }'

# Conversion history
curl -X GET "https://markclean.online/api/converter/history?
page=1&page_size=10" \
  -H "X-API-Key: mk_live_YOUR_KEY"

# Balance check
curl -X GET "https://markclean.online/api/billing/usage" \
  -H "X-API-Key: mk_live_YOUR_KEY"

# Get available packages
curl -X GET "https://markclean.online/api/billing/packages"
```

? FAQ

🔑 Is it safe to pass an API key?

Yes, as long as you follow these rules:

-  Use HTTPS (always)
-  Store the key in environment variables, not in code
-  Do not log the key
-  Do not commit the key to Git
-  Do not pass the key in URL parameters

What to do if the key is compromised?

1. Open `/api-keys` in the web interface
2. Click "**Regenerate**" (the old key is deleted and a new one is created)
3. Update the key in all of your applications

 The old key stops working immediately after regeneration.

Can I use the API without credits?

Yes, every user has a **free daily limit** (10 requests / day) that resets every 24 hours. After both the daily limit and credits are exhausted, the API will return `402 quota_exceeded`.

Why is `quota_incremented: false` ?

This is normal in three cases:

1. **Cache hit** — this URL was recently converted (free)
2. **Anonymous request** — without an API key (IP limit)
3. **Subscription** — the user has an active subscription, no credits are consumed

Does the API work with any site?

The API works with any **publicly accessible HTML page**. Limitations:

-  Sites requiring authentication
-  Sites with strict anti-bot protection (Cloudflare challenges, etc.)
-  PDF, images, and other non-HTML formats
-  Sites that block by User-Agent

Is there a page size limit?

- Maximum: **5 MB** of content after conversion
- Typical articles: 10-100 KB
- Long technical documents: 200-500 KB

Pages larger than 5 MB are not cached but are still converted.

How to use an API key with other tools?

Postman:

- **Authorization** tab → Type: **API Key**
- Key: X-API-Key , Value: your key, 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"
}
```

What happens to the API key after account deletion?

The key continues to work until the credits run out. After account deletion:

-  The API key remains active
-  Existing credits can still be used
-  You cannot top up credits (no UI access)
-  You cannot create a new key

This is for fairness: you paid for the credits — you use them.

Is there an SDK?

No official SDK yet, but the API is simple enough to use via standard HTTP clients in any language. The examples above cover the most popular languages.

Support

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