

## REST API

The *html-to-markdown* converter produces **accurate** outputs. This makes it possible to convert *entire websites* - useful for website migrations and other tasks. Still not convinced? Try the [online demo](#).

The REST API can be used to automate HTML to Markdown conversion:

**POST** <https://api.html-to-markdown.com/v1/convert>

### Headers

Content-Type:

application/json

X-API-Key:

html2md\_8jn6nD5Mfim\_HtG9fyp1banwesBhjRinBdDQKxKpvqxpHYUrJcezh5ZV\_Chsm7jcmEnSk3WDv1yhnxgbBpDbcYQnKKc4tXcjQ8W4a

👤 Hi Nathaniel Williams, this is your personal API key. Don't share it with others. [Logout](#)

### Request Body

```
{ "html": "<strong>bold text</strong>" }
```

### Response Body ● 201

```
{ "markdown": "**bold text**" }
```

Try it out in the terminal with cURL:

```
curl \
  -H 'X-API-Key: YourApiKey' \
  -H 'Content-Type: application/json' \
  -d '{ "html": "<strong>bold text</strong>" }' \
  -X POST \
  https://api.html-to-markdown.com/v1/convert
```

## Authentication

An API key is needed to authenticate with the API.

html2md\_8jn6nD5Mfim\_HtG9fyp1banwesBhjRinBdDQKxKpvqxpHYUrJcezh5ZV\_Chsm7jcmEnSk3WDv1yhnxgbBpDbcYQnKKc4tXcjQ8W4a

👤 Hi Nathaniel Williams, this is your personal API key. Don't share it with others. [Logout](#)

When making a request, you need to pass the API key using the `X-API-Key: YourApiKey` header.

Note: Your API key is confidential and should be securely stored on your server. Avoid exposing it in client-side code...

## Input

It is required to set the `Content-Type` header. This informs how the body should be decoded.

```
# Request Body for Content-Type: application/json
{ "html": "<strong>bold text</strong>" }
```

- `Content-Type: application/json` (preferred) with the input inside the "html" value. This is the preferred way because you can pass other keys to configure the converter.
- `Content-Type: text/html` send the HTML input directly in the body.

## Output

By default, the output is JSON:

```
# Response Body
{ "markdown": "**bold text**" }
```

Note: Once you convert this markdown **back to** HTML you need to be careful of malicious content. Use an HTML sanitizer before displaying the HTML in the browser.

If you wish to get a different output, you can also configure the `Accept` header.

- `Accept: application/json` (default) will return the output as json with the "markdown" key. This is the preferred way because it leaves the possibility of returning other information (e.g. frontmatter).
- `Accept: text/markdown` will return the output directly in the response.

## Options

Pass the `domain` option to convert *relative* links to *absolute* links:

```
{
  "html": "<a href='/page.html'>link text</a>",
  "domain": "https://example.com"
}
```

The response:

```
{ "markdown": "[link text](https://example.com/page.html)" }
```

## Plugins

To enable the `strikethrough` plugin, send this request:

```
{
  "html": "<s>strikethrough text</s>",
  "plugins": {
    "strikethrough": {}
  }
}
```

The response:

```
{ "markdown": "~strikethrough text~" }
```

These plugins are currently available:

- `commonmark`  
Implements Markdown according to the Commonmark Spec. This is *automatically enabled*.
- `strikethrough`  
Converts strikethrough to the "~" syntax. Implements according to the [GitHub Flavored Markdown Spec](#).
- `table`  
Converts tables according to the [GitHub Flavored Markdown Spec](#).
- **Advanced Edge-Cases** Premium  
The web contains many pages. Some of them with quirks. This plugin tries to correct some of these rare edge-cases. If you have a paid subscription this plugin is *automatically enabled*.

Missing a plugin you need? Let's build it together! Write an email to [hello@html-to-markdown.com](mailto:hello@html-to-markdown.com) to get started.

## Limits

Depending on the subscription there are different limits. See the [pricing page](#) for more info.

## Contact

If you have any questions, please don't hesitate to [write an email](#).

## Errors

```
{
  "error": {
    "code": "HEADER_XAPIKEY_MISSING",
    "title": "the header 'X-API-Key' is required",
    "detail": "did you accidentally use the 'Authorization' header?"
  }
}
```

Here is a list of common errors:

- **DECODE\_BODY\_INVALID**  
there is an error parsing the body
- **DECODE\_BODY\_TOO\_LARGE**  
request body too big
- **DECODE\_HTML\_CONTENT\_MISSING**  
no content found while parsing the body as html
- **DECODE\_JSON\_BODY\_INVALID**  
error while parsing the body as json
- **DECODE\_JSON\_CONTENT\_MISSING**  
no content found while parsing the body as json
- **DECODE\_MULTIPART\_BODY\_INVALID**  
error while parsing the body as multipart form
- **DECODE\_MULTIPART\_CONTENT\_MISSING**  
no content found while parsing the body as multipart form
- **FILETYPE\_UNSUPPORTED**  
this file type is not supported
- **HEADER\_ACCEPT\_MEDIATYPE\_UNSUPPORTED**  
the 'Accept' header defines an unsupported value
- **HEADER\_CONTENTTYPE\_CHARSET\_UNSUPPORTED**  
the 'Content-Type' header defines an unsupported charset
- **HEADER\_CONTENTTYPE\_INVALID**  
could not parse the 'Content-Type' header
- **HEADER\_CONTENTTYPE\_MEDIATYPE\_UNSUPPORTED**  
the 'Content-Type' header defines an unsupported mediatype
- **HEADER\_CONTENTTYPE\_MISSING**  
the header 'Content-Type' is required
- **HEADER\_XAPIKEY\_INVALID**  
the api key is invalid
- **HEADER\_XAPIKEY\_MISSING**  
the header 'X-API-Key' is required
- **METHOD\_NOT\_ALLOWED**  
this http method is not allowed
- **PATH\_NOT\_FOUND**  
this http path is not found
- **TOO\_MANY\_REQUESTS**

received too many requests in the timeframe

Made by [Johannes Kaufmann](#) in Berlin