

## HTML 转换 MD

### 接入点说明:

该接口功能是将网页 HTML 源码转换成 Markdown 格式的文本。本接口以每消耗 3000 个 token（对应 HTML 中正文文字数量约为 2300 个）计费一次。如果 HTML 正文文字过多，则存在多次计费的情况。

### 接口地址:

[https://route.showapi.com/3193-2?appKey={your\\_appKey}](https://route.showapi.com/3193-2?appKey={your_appKey}) [查看密钥](#) [调用帮助](#)

### 返回格式:

json

### 请求参数

### post 表单

参数名称	类型	示例值	必须	描述
html	String	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"> <html> <head> <title>libcurl - the multiprotocol file transfer library</title> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <meta content="text/html; charset=UTF-8" http-equiv="Content-Type"> <script type="text/javascript" nonce="fab99ea9780e4289b8051a24524" src="//injections.adguard.org?ts=1712884296286&type=content-script&dmn=curl.se&url=https%3A%2F%2Fcurl.se%2Flibcurl%2F&app=com.operasoftware.Opera&css=3&js=1&rel=1&rji=1&p;sbe=1"></script> <script type="text/javascript" nonce="fab99ea9780e4289b8051a24524" src="//injections.adguard.org?ts=1712884296286&name=AdGuard%20Popup%20Blocker&name=AdGuard%20Extra&type=user-script"></script><link rel="stylesheet" type="text	否	网页的 HTML 源码

参数名称	类型	示例值	必须	描述
		<pre> /css" href="/curl.css"&gt; &lt;link rel="sh ortcut icon" href="/favicon.ico"&gt; &lt;li nk rel="icon" href="/logo/curl-symbo l.svg" type="image/svg+xml"&gt; &lt;link re l="alternate" type="application/rss+x ml" title="cURL Releases" href="http s://github.com/curl/curl/releases.ato m"&gt; &lt;meta name="description" content= "libcurl is a free and easy-to-use cl ient-side URL transfer library, suppo rting DICT, FILE, FTP, FTPS, GOPHER, GOPHERS, HTTP, HTTPS, IMAP, IMAPS, LD AP, LDAPS, MQTT, POP3, POP3S, RTMP, R TMPS, RTSP, SCP, SFTP, SMB, SMBS, SMT P, SMTPS, TELNET and TFTP. libcurl su pports SSL certificates, HTTP POST, H TTP PUT, FTP uploading, kerberos, HTT P form based upload, proxies, cookie s, user+password authentication, file transfer resume, http proxy tunnelin g and more"&gt; &lt;/head&gt; &lt;body&gt; &lt;div clas s="main"&gt; &lt;div class="menu"&gt; &lt;a href= "" class="itemselect"&gt;libcurl &lt;/a&gt; &lt;d iv class="dropdown"&gt; &lt;a class="droptb n" href="/libcurl/"&gt;Docs&lt;/a&gt; &lt;div cla ss="dropdown-content"&gt; &lt;a href="/libc url/abi.html"&gt;ABI&lt;/a&gt; &lt;a href="/libcu rl/c/"&gt;API&lt;/a&gt; &lt;a href="/libcurl/bind ings.html"&gt;Bindings&lt;/a&gt; &lt;a href="/lib curl/competitors.html"&gt;Competitors&lt;/ a&gt; &lt;a href="/libcurl/c/example.html"&gt; Examples&lt;/a&gt; &lt;a href="/libcurl/featur es.html"&gt;Features&lt;/a&gt; &lt;a href="/mail/ list.cgi?list=curl-library"&gt;Mailing l ist&lt;/a&gt; &lt;a href="/libcurl/relatedlib s.html"&gt;Related libs&lt;/a&gt; &lt;a href="/li bcurl/using/"&gt;Using libcurl&lt;/a&gt; &lt;a hr ef="/libcurl/security.html"&gt;Security&lt; /a&gt; &lt;a href="/libcurl/c/symbols-in-ve rsions.html"&gt;Symbols&lt;/a&gt; &lt;a href="/li bcurl/c/libcurl-tutorial.html"&gt;Tutori al&lt;/a&gt; &lt;a href="/libcurl/theysay.html "&gt;Testimonials&lt;/a&gt; &lt;/div&gt; &lt;/div&gt; &lt;/di </pre>		

参数名称	类型	示例值	必须	描述
		<pre>v&gt; &lt;div class="contents"&gt; &lt;div class="where"&gt;&lt;a href="/"&gt;curl&lt;/a&gt; / &lt;b&gt;libcurl overview&lt;/b&gt;&lt;/div&gt; &lt;h1&gt; libcurl - the multiprotocol file transfer library &lt;/h1&gt; &lt;p&gt; libcurl is a &lt;a href="/docs/copyright.html" title="free as in both free speech and zero price!"&gt;free&lt;/a&gt; and easy-to-use client-side URL transfer library, supporting DIC T, FILE, FTP, FTPS, GOPHER, GOPHERS, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAP S, MQTT, POP3, POP3S, RTMP, RTMPS, RT SP, SCP, SFTP, SMB, SMBS, SMTP, SMTP S, TELNET and TFTP. libcurl supports SSL certificates, HTTP POST, HTTP PU T, FTP uploading, HTTP form based upl oad, proxies, HTTP/2, HTTP/3, cookie s, user+password authentication (Basi c, Digest, NTLM, Negotiate, Kerbero s), file transfer resume, http proxy tunneling and more! &lt;/p&gt; &lt;p&gt; libcurl is highly portable, it builds and wor ks identically on numerous platforms, including Solaris, NetBSD, FreeBSD, OpenBSD, Darwin, HPUX, IRIX, AIX, Tru 64, Linux, UnixWare, HURD, Windows, A miga, OS/2, BeOs, Mac OS X, Ultrix, Q NX, OpenVMS, RISC OS, Novell NetWare, DOS and more... &lt;/p&gt; &lt;p&gt; libcurl is &lt;a href="/docs/copyright.html"&gt;free&lt;/a&gt;, &lt;a href="features.html#thread"&gt;th read-safe&lt;/a&gt;, &lt;a href="features.html #ipv6"&gt;IPv6 compatible&lt;/a&gt;, &lt;a href=" features.html#features"&gt;feature rich&lt;/a&gt;, &lt;a href="features.html#support"&gt; well supported&lt;/a&gt;, &lt;a href="feature s.html#fast"&gt;fast&lt;/a&gt;, &lt;a href="featu res.html#docs"&gt;thoroughly documented&lt;/a&gt; and is already used by many know n, big and successful &lt;a href="/docs/ companies.html"&gt;companies&lt;/a&gt;. &lt;/p&gt; &lt;p&gt;&lt;b&gt;Download&lt;/b&gt; &lt;p&gt; Go to the regul ar curl &lt;a href="/download.html"&gt;down</pre>		

参数名称	类型	示例值	必须	描述
		<p>load page</p> <p>and get the latest curl package, or one of the specific libcurl packages listed.</p> <p><b>API</b></p> <p>You use libcurl with the provided <code>C API</code>. The curl team works hard to keep the <code>API and ABI stable</code>. If you prefer using libcurl from your other favorite language, chances are there's already a <code>binding</code> written for it.</p> <p><b>Howto</b></p> <p>Check out our <code>using libcurl</code> page for general hints and advice and man pages for <code>every function</code> and option. Also, check out the <code>libcurl section of everything curl</code>.</p> <p>libcurl is probably the most portable, most powerful and most often used network transfer library on this planet.</p>		
ret_type	String	markdown	否	返回格式 <code>json</code> （默认值） <code>markdown</code>
run_type	String	async	否	执行方式 <code>sync</code> : 同步返回（生成时间比较长，容易超时，建议使用异步） <code>async</code> : 异步返回（默认值）
task_name	String		否	任务名称

## 返回参数

[查看公共返回参数](#)

## 返回体

参数名称	类型	示例值	描述
remark	String	请求成功	请求描述信息
md_text	String		markdown 格式的文本（同步时返回内容）
ret_code	Number	0	0 成功，其它失败
task_id	String		异步任务 id（异步时返回，用于查询异步历史及详情）

## 请求示例

[AI 生成调用代码](#)

[查看我的 appKey](#)

[查看使用向导](#)

```
#查看 appkey: https://www.showapi.com/console#/myAppcurl -X POST
'https://route.showapi.com/3193-2?appKey={your_appKey}' \
-H 'content-type:application/x-www-form-urlencoded' \
-d 'run_type=async' \
-d 'html= <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/lo...已省略' \
-d 'ret_type=markdown' \
-d 'task_name='
```

## 返回示例

```
{
  "remark": "请求成功",
  "md_text": "",
  "task_id": "",
  "ret_code": 0}
```

## 转换详情

### 接入点说明:

HTML 转 MarkdownAPI 接口允许用户根据任务 ID (`task_id`) 查询 HTML 到 Markdown 格式转换任务的当前状态，方便用户跟踪转换进度和结果。

### 接口地址:

[https://route.showapi.com/3193-3?appKey={your\\_appKey}](https://route.showapi.com/3193-3?appKey={your_appKey}) [查看密钥](#) [调用帮助](#)

### 返回格式:

json

## 请求参数

### post 表单

参数名称	类型	示例值	必须	描述
<code>task_id</code>	String		否	提交异步任务时返回的任务 <code>id</code>

## 返回参数

[查看公共返回参数](#)

## 返回体

参数名称	类型	示例值	描述
<code>remark</code>	String	请求成功	请求描述信息
<code>md_text</code>	String		markdown 格式的文本（文档生成成功后返回）
<code>ret_code</code>	Number	0	0 成功，其它失败
<code>task_status</code>	String	waiting	文档生成状态（此状态在返回头中也包含，返回格式为 Markdown 的可从返回头中获取） 1、 <code>not_exit</code> 任务不存在

参数名称	类型	示例值	描述
			2、 <i>waiting</i> 任务执行中 3、 <i>success</i> 任务执行成功 4、 <i>fail</i> 任务执行失败
task_name	String	测试页面	任务名称

### 请求示例

[AI 生成调用代码](#)

[查看我的 appKey](#)

[查看使用向导](#)

```
#查看 appkey: https://www.showapi.com/console#/myAppcurl -X POST
'https://route.showapi.com/3193-3?appKey={your_appKey}' \
-H 'content-type:application/x-www-form-urlencoded' \
-d 'task_id='
```

### 返回示例

```
{
  "ret_code": -1,
  "remark": "Markdown 生成中, 请耐心等待",
  "task_status": "waiting",
  "task_name": "测试页面"}
```

## 转换历史

### 接入点说明:

HTML 转 MarkdownAPI 接口的转化历史接入点可查询最近的转换历史。

### 接口地址:

https://route.showapi.com/3193-4?appKey={your\_appKey} [查看密钥](#) [调用帮助](#)

返回格式:

json

### 请求参数

### post 表单

参数名称	类型	示例值	必须	描述
page	String	1	否	查询页面

### 返回参数

[查看公共返回参数](#)

### 返回体

参数名称	类型	示例值	描述
allNum	Number	5	列表总数
contentlist	Object[]		返回列表
- task_id	String	e6e723cf4	历史任务 id
- ct	String	2024-04-02 16:39:30.815	任务时间
- task_status	String		任务状态 ( <i>not_exit, waiting, success, fail</i> )
- task_name	String		任务名称
remark	String	查询成功	请求描述信息
ret_code	Number	0	0 成功, 其它失败
currentPage	Number	1	当前页码
allPages	Number	1	总页数
maxResult	Number	20	当前页最大数

### 请求示例

[AI 生成调用代码](#)

[查看我的 appKey](#)



## [查看使用向导](#)

```
#查看 appkey: https://www.showapi.com/console#/myAppcurl -X POST  
'https://route.showapi.com/3193-4?appKey={your_appKey}' \  
-H 'content-type:application/x-www-form-urlencoded' \  
-d 'page=1'
```

## 返回示例

```
{  
  "showapi_res_id": "",  
  "showapi_res_error": "",  
  "showapi_res_code": 0,  
  "showapi_res_body": {  
    "allNum": 5,  
    "contentlist": [  
      {  
        "task_id": "e6e723cf4",  
        "task_name": "测试任务",  
        "task_status": "success",  
        "ct": "2024-04-02 16:39:30.815"  
      }  
    ],  
    "remark": "查询成功",  
    "ret_code": 0,  
    "currentPage": 1,  
    "allPages": 1,  
    "maxResult": 20  
  }  
}
```