



Firecrawl capabilities, limits & change log (as of 27 Jan 2026)

Overview

Firecrawl is a cloud API for crawling, scraping, searching and extracting data from the web. The service converts pages into LLM-ready markdown or structured JSON and offers a unified credit-based billing model. This report summarises the latest documentation on the **search**, **map**, **crawl**, **extract** and **change-tracking** endpoints, discusses available **output formats** (markdown/screenshot/JSON etc.), explains **rate limits** and **pricing tiers**, outlines **API usage constraints**, describes **best practices** for compliance and cost-control, and lists **breaking changes** since mid-2024.

Search endpoint

The `/search` endpoint performs web searches and can optionally scrape the resulting pages in one call. Key features include:

- **Result formats** – The API can return result summaries in markdown, HTML, link lists or screenshots; you specify the `formats` array to control the output ¹.
- **Customisable search parameters** – You can adjust `location` (e.g., `US`, `Germany`) and `languages` to emulate local results and specify a time window using the `tbs` parameter (e.g., `qdr:d` for past 24 hours, `qdr:m` for past month) ². A `timeout` parameter sets a custom request time limit ³.
- **Search categories** – Categories such as `images`, `news`, `code` (GitHub) and `research` filter results and output appropriate JSON structure. The results include standard fields like `url`, `title` and `description` ⁴.
- **Cost and credits** – Firecrawl charges **2 credits per 10 search results**, plus 1 credit per page when scraping pages. PDF parsing costs 1 credit per page, stealth proxy mode adds 4 credits per page, and JSON mode adds 4 credits per page ⁵. To manage cost, Firecrawl advises disabling PDF parsing (`parsers: []`), using the `proxy: "basic"` instead of `stealth`, and limiting the number of results with the `limit` parameter ⁶.
- **Advanced options** – Searches can embed scraping options (e.g., `formats`, `actions`), but the FIRE-1 agent and change-tracking features are not currently supported through the search endpoint ⁷.

Map endpoint

The `/map` endpoint returns the list of URLs found on a target site. Important points:

- **Purpose** – Mapping quickly discovers links when you need to prompt users to pick pages or target a subset of pages based on a keyword ⁸.

- **Usage** – Provide the `url` and optional `limit` or `search` term. The API returns an ordered list of URLs with optional titles/descriptions ⁹. You can call map via SDK or cURL, and results are delivered immediately for small sites ¹⁰.
- **Location and language** – Like the `scrape` endpoint, you may supply a `location` object (`country` ISO code and an array of `languages`) to emulate region-specific results. Default country is **US** ¹¹.
- **Search within a site** – Passing a `search` string filters returned URLs by relevance ⁹.
- **Limitations** – The map endpoint prioritises speed and may miss some links; it is best used to gather most links quickly rather than for exhaustive crawling ¹².

Crawl endpoint

The `/crawl` endpoint recursively follows links from a starting URL and scrapes each page using the same options available in `/scrape`. Key points:

- **Crawl options** – Parameters include `limit` (maximum pages to crawl), `crawlEntireDomain` (include pages outside the initial path), `allowSubdomains` (include subdomains) and `prompt` (generate crawler options from natural language). Pages are scraped with `scrapeOptions` (e.g., `formats`, `proxy`, `maxAge`) ¹³ ¹⁴.
- **Asynchronous jobs** – A crawl request returns an `id` and job URL ¹⁵. Results are available for **24 hours** ¹⁶. Large results are paginated using a `next` URL and `skip` parameter ¹⁷.
- **SDK methods** – The SDK offers two flows: `crawl` waits until completion and returns all documents, while `start_crawl` returns immediately and requires polling with `get_crawl_status` ¹⁸. Real-time updates can be streamed via a **WebSocket** ("Crawl URL and Watch") or delivered by **webhook**; verify the webhook signature using the `X-Firecrawl-Signature` header for security ¹⁹.
- **Cost** – Crawling consumes **1 credit per page** (the same as scraping) and inherits extra costs for PDF parsing, stealth proxy and JSON extraction.
- **Best practices** – Limit the number of pages (`limit`) to avoid crawling an entire domain inadvertently; use `maxDiscoveryDepth`, `includePaths`/`excludePaths` to control scope; and add `waitFor` or actions in `scrapeOptions` to ensure dynamic content loads fully.

Extract endpoint (beta)

The `/extract` endpoint uses Firecrawl's LLMs to generate structured data from one or more pages. Key features:

- **Input** – Provide a list of `urls` (wildcards `*` are supported), a natural-language `prompt` and/or a JSON `schema` describing desired fields ²⁰. Wildcard domains cause Firecrawl to crawl the domain and extract the requested fields from every page ²¹. An `enableWebSearch` flag broadens the search to pages linked from the domain to enrich results ²².
- **Flexible schema or prompt** – You can extract without a schema by supplying only a `prompt` ²³, or omit both and let the model decide a structure. The endpoint can even be used **without specifying URLs** (Alpha feature) by sending just a `prompt` ²⁴.
- **Job workflow** – An extraction job returns an ID; use `get_extract_status` to poll until the state becomes `completed` ²⁵. Possible states include `completed`, `processing`, `failed`, and `cancelled` ²⁶, and results are kept for 24 hours ²⁷.

- **Cost** – The beta extract endpoint now uses **credits** like other endpoints; **1 credit = 15 tokens** ²⁸.
- **Limitations** – The documentation notes that extraction is still in beta: coverage of massive sites (e.g., entire Amazon) is not guaranteed, complex logical queries (“every post from 2025”) may miss results, outputs may vary between runs, and features will evolve ²⁹.

Change-tracking format

Change tracking allows you to monitor how a page’s content changes between scrapes. You must include both `markdown` and a `changeTracking` object in the `formats` array when scraping.

- **Change status and visibility** – Each document includes `previousScrapeAt`, `changeStatus` (`new`, `same`, `changed`, `removed`) and `visibility` (`visible` or `hidden`) ³⁰.
- **Modes** – Two modes are supported: `git-diff` produces a traditional line-by-line diff, while `json` compares specific fields defined in a schema ³¹ ³². You can specify both modes by setting `modes: ['git-diff', 'json']` in the format object and optionally supply a `schema`, `prompt` or `tag` to scope comparisons ³³.
- **Data model** – The change-tracking object contains a `diff` object with `text` (git style) and `json` (structured diff) and a `json` property with field comparisons ³⁴.
- **Important facts** – Comparisons always use the markdown output; the same URL must be scraped with the same tag and `includeTags` / `excludeTags` settings for consistent comparisons. Previous scrapes are matched by team and URL, and beta status means the `changeTracking` object may occasionally be missing; monitor the `warning` field ³⁵.
- **Cost** – Git-diff mode is free; JSON mode costs **5 credits per page** ³⁶.

Scrape formats and screenshot options

The `/scrape` endpoint converts a single page into multiple formats. Supported formats include:

- **Markdown** – Cleaned body content, ideal for LLMs.
- **Summary** – Short summary of the page.
- **HTML / rawHtml** – Full or raw HTML content.
- **Screenshot** – Returns a PNG; you can set `fullPage` (boolean), `quality` (0-100) and a `viewport` width/height ³⁷.
- **Links** – List of extracted hyperlinks.
- **JSON** – Structured data using a prompt or JSON schema ³⁷.
- **Images** – Extracts image URLs.
- **Branding** – Extracts brand identity (colors, fonts, typography, components) ³⁸.
- **Change-tracking** – As described above.

In the **Advanced Scraping Guide**, Firecrawl explains how to combine formats and customize scraping:

- **Object formats** – You can pass objects to `formats` for JSON, screenshot or change-tracking. For screenshot: `{ type: 'screenshot', fullPage?, quality?, viewport? }` ³⁹. For change-tracking: `{ type: 'changeTracking', modes?, prompt?, schema?, tag? }` ⁴⁰.
- **Only main content** – Set `onlyMainContent` to `false` to capture the entire page; default `true` keeps just the article body ⁴¹.

- **Caching with `maxAge`** – Firecrawl caches pages for **2 days** (172,800 ms). Setting `maxAge =0` forces a fresh scrape, while a higher value leverages the cache and reduces credits ⁴².
- **Wait for page readiness (`waitFor`)** – Add extra delay in milliseconds to allow dynamic content to load ⁴³.
- **PDF parsing (`parsers`)** – Use `parsers: ['pdf']` to parse PDFs; cost is **1 credit per PDF page**. Setting `parsers: []` returns the PDF as base64 for **1 credit** and avoids per-page costs ⁴⁴.
- **Actions** – Define a sequence of steps (wait, click, write, press, scroll, screenshot, scrape, executeJavascript or pdf) to interact with dynamic pages before scraping ⁴⁵. For example, clicking a “Load more” button and then taking a screenshot is done via actions ⁴⁵.
- **Whitelisting** – To allow Firecrawl to access your own site, whitelist the `FirecrawlAgent` user-agent and note that Firecrawl uses dynamic IPs; to permit your app to call Firecrawl’s API, allow outbound traffic to **35.245.250.27** ⁴⁶.

Rate limits

Firecrawl uses subscription-based plans that include credits and concurrency limits. Rate limits are measured in **requests per minute** and concurrency is measured in **simultaneous browser sessions**. The rate-limits documentation lists the following values ⁴⁷ ⁴⁸:

Plan	Concurrent browser sessions	/scrape & /map (req/min)	/crawl (req/min)	/search (req/min)	/agent (req/min)	/crawl/status	/agent/status
Free	2	10	1	5	10	1500	500
Hobby	5	100	15	50	100	1500	25,000
Standard	50	500	50	250	500	1500	25,000
Growth	100	5,000	250	2,500	1,000	1500	25,000
Scale/ Enterprise	150+	Custom (contact sales)	Custom	Custom	Custom	—	—

Additional notes:

- **Concurrent browsers** – Jobs exceeding the plan’s concurrency will queue until slots free up ⁴⁷.
- **FIRE-1 agent limits** – The `/scrape` and `/extract` endpoints when used via FIRE-1 are limited to **10 requests per minute** each ⁴⁹. Extract endpoints share the agent limits; batch scrape jobs share the crawl limits ⁵⁰.

Pricing tiers and credit consumption

The pricing page lists the cost of each plan and credit packs. Credits are spent on API calls, and additional packs auto-recharge at discounted rates:

Plan	Credits/month	Price (USD/mo, billed yearly)	Concurrency	Extra credit cost
Free	500 credits (one-time)	\\$0	2 concurrent	—
Hobby	3,000 credits	\\$16	5 concurrent	\\$9 per extra 1k credits <small>51</small>
Standard	100,000 credits	\\$83	50 concurrent	\\$47 per extra 35k credits <small>52</small>
Growth	500,000 credits	\\$333	100 concurrent	\\$177 per extra 175k credits <small>53</small>
Scale	1,000,000 credits	\\$599	150 concurrent	custom <small>54</small>
Enterprise	Custom credits	Custom	Custom concurrency	includes zero-data retention, SSO and advanced security <small>55</small>

Credit usage per endpoint: scraping, crawling and mapping cost **1 credit per page**; search costs **2 credits per 10 results**; the agent includes **5 free runs per day** and dynamic pricing 56. Extract also uses 1 credit per 15 tokens 28.

API usage constraints and compliance

Firecrawl's terms of service and documentation impose several restrictions:

- **Lawful use only** – Users must obey all applicable laws and may not use Firecrawl to infringe intellectual property rights or violate privacy rules. Prohibited activities include exploiting the service, reverse engineering, or reselling the service without permission.
- **Regulated activities** – Firecrawl forbids use of its API for **debt-collection, background checks, credit, insurance or employment decisions**, or any purpose regulated by the **Fair Credit Reporting Act (FCRA)**, as well as law-enforcement or intelligence gathering 57 (legacy terms). You also may not use it for discriminatory purposes or to develop weapons, facial recognition or mass surveillance tools.
- **Data privacy** – Do not scrape pages containing sensitive personal data unless you have consent and a lawful basis. Respect robots.txt and copyright when crawling websites.
- **Whitelisting** – To allow Firecrawl to scrape your own site, permit the `FirecrawlAgent` user-agent and understand that IP addresses are not fixed; to allow your app to call Firecrawl's API, whitelist outbound IP `35.245.250.27` 46.
- **Webhook security** – When using webhooks for crawl or extract jobs, verify the `X-Firecrawl-Signature` header to authenticate the sender 19.

Best practices for cost control and compliance

1. **Limit scope upfront** – For search and crawl operations, set a reasonable `limit` to avoid exploring thousands of pages inadvertently; use `maxDiscoveryDepth`, `includePaths` and `excludePaths` to confine crawling to relevant sections.

2. **Use caching** – Accept the default `maxAge` (2 days) whenever fresh data isn't required. This returns cached content instantly and avoids unnecessary credits ⁴².
3. **Control parsing costs** – Disable PDF parsing by setting `parsers: []` or limit pages with `maxPages` to reduce credits ⁴⁴. Use `proxy: 'basic'` instead of `stealth` when you don't need stealth scraping ⁶.
4. **Minimise search results** – Set the `limit` parameter to return only the number of search results you intend to process and avoid scraping unused results ⁶.
5. **Combine formats wisely** – Only request the formats you need. Screenshots and JSON extraction incur extra costs; requesting fewer formats reduces credits.
6. **Wait for dynamic pages** – Use `waitFor` or actions to wait for dynamic content and avoid repeated scrapes due to incomplete pages ⁴³. If pages require user interactions, define appropriate `actions` (click, scroll, write, screenshot, etc.) ⁴⁵.
7. **Monitor rate limits** – Keep concurrency within your plan to avoid queuing. For large batch operations, stagger requests to stay within per-minute rate limits.
8. **Security and compliance** – Verify webhook signatures, whitelist user agent/IP addresses when scraping your own site, and avoid scraping regulated data (FCRA, personal health, etc.). Always follow robots.txt and respect site terms.
9. **Use the agent for complex tasks** – The new `/agent` endpoint (Beta) and associated **FIRE-1** agent can perform complex navigation and extraction tasks; however, be mindful of separate rate limits (10 requests/min per endpoint) ⁴⁹.

Breaking changes and major updates since mid-2024

Firecrawl has evolved rapidly since mid-2024. Notable changes include:

Date/version	Breaking change or major update	Evidence
Mid-2024 (Firecrawl v2 release)	<p>A major rewrite introduced v2 with new endpoints (<code>/search</code>, improved <code>/scrape</code>), unified caching (<code>maxAge</code> defaults to 2 days), a new <code>summary</code> format, and replacement of the old <code>extract</code> format with a new <code>JSON</code> format requiring an object <code>{type: 'json', schema, prompt}</code>. Screenshot options now require an object with <code>fullPage</code>, <code>quality</code> and <code>viewport</code> settings ³⁹.</p> <p>Search categories were added, natural-language prompts for crawling were introduced, and method names were standardised (e.g., <code>scrape</code> vs <code>scrapeUrl</code>) ⁵⁸ (previous context).</p>	v1→v2 migration documentation and release notes
Apr 2025 – Launch Week III	Firecrawl launched an enhanced change-tracking feature with git-diff and JSON modes, structured comparison and Beta warnings ³³ ³⁵ . It also introduced <code>/extract v2</code> , the FIRE-1 agent , templates, LLMSTXT.new and other features (blog posts).	Launch Week III posts (April 2025).

Date/version	Breaking change or major update	Evidence
Jun 2025 - /search announcement	The <code>/search</code> endpoint (v2) allowed developers to perform web searches and optionally scrape results in one call. It supports categories (news, images, code), location/time filtering and cost-effective pricing of 2 credits per 10 results ¹ .	Search documentation.
Oct 2025 (v2.5)	Release v2.5 introduced a semantic index and custom browser stack , enabling higher coverage and speed; search became 5x cheaper ; improved concurrency and new PDF search category; screenshot quality and speed were improved; Excel scraping support was added; and unified billing introduced auto-recharge credit packs.	Changelog v2.5.
Nov 2025 (v2.6)	Unified billing merged credits and tokens; branding format improved; speed/reliability upgrades; change tracking became faster and more reliable; and instant credit purchases were enabled ⁵⁹ .	Changelog v2.6.
Dec 2025 (v2.7)	Added ZDR (Zero-Data-Retention) Search for enterprise customers, a partner integrations API, improved branding format and faster screenshots ⁶⁰ .	Changelog v2.7.
Jan 27 2026	Introduced the Firecrawl Skill & CLI for AI agents, offering a one-command install to access all Firecrawl endpoints and improved real-time data access ⁶¹ .	Changelog – Jan 27 2026.

Conclusion

Firecrawl has evolved into a comprehensive platform for AI-driven web data extraction. Its search, map, crawl, extract and change-tracking endpoints, combined with flexible output formats (markdown, screenshots, JSON), allow developers to gather structured or raw content from across the web. The service's unified credit model and defined rate limits help plan capacity and cost. Users should abide by Firecrawl's usage restrictions and implement cost-control practices—limiting scope, leveraging caching, and minimising expensive options—to stay within budgets. Continual updates, including major version changes in 2024-2025 and new features like the FIRE-1 agent and CLI, underscore the importance of tracking Firecrawl's changelog to ensure compatibility.

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [Search | Firecrawl](#)

<https://docs.firecrawl.dev/features/search>

[8](#) [9](#) [10](#) [11](#) [12](#) [Map | Firecrawl](#)

<https://docs.firecrawl.dev/features/map>

[13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [Crawl | Firecrawl](#)

<https://docs.firecrawl.dev/features/crawl>

- [20 21 22 23 24 25 26 27 28 29 Extract | Firecrawl](https://docs.firecrawl.dev/features/extract)
<https://docs.firecrawl.dev/features/extract>
- [30 31 32 33 34 35 36 Change tracking | Firecrawl](https://docs.firecrawl.dev/features/change-tracking)
<https://docs.firecrawl.dev/features/change-tracking>
- [37 38 Scrape | Firecrawl](https://docs.firecrawl.dev/features/scrape)
<https://docs.firecrawl.dev/features/scrape>
- [39 40 41 42 43 44 45 46 Advanced Scraping Guide | Firecrawl](https://docs.firecrawl.dev/advanced-scraping-guide)
<https://docs.firecrawl.dev/advanced-scraping-guide>
- [47 48 49 50 Rate Limits | Firecrawl](https://docs.firecrawl.dev/rate-limits)
<https://docs.firecrawl.dev/rate-limits>
- [51 52 53 54 55 56 Firecrawl - The Web Data API for AI](https://www.firecrawl.dev/pricing)
<https://www.firecrawl.dev/pricing>
- [57 Terms of Service | Firecrawl](https://www.firecrawl.dev/terms-of-service)
<https://www.firecrawl.dev/terms-of-service>
- [58 Migrating from v1 to v2 | Firecrawl](https://docs.firecrawl.dev/migrate-to-v2)
<https://docs.firecrawl.dev/migrate-to-v2>
- [59 60 61 Changelog | Firecrawl](https://www.firecrawl.dev/changelog)
<https://www.firecrawl.dev/changelog>