

Getting Started

Welcome to `html2img`. This guide covers the four endpoints, authentication, and the response format you can expect. The HTML and Screenshot endpoints can also return a PDF instead of a PNG via the [format parameter](#).

Authentication

All API requests require authentication using an API key. Include your key in the `X-API-Key` header on every request.

[Read the authentication guide](#) ■

Skip the raw HTTP calls with an official SDK

The API works from any language that can make an HTTP request, but if you use PHP, Laravel or JavaScript there is an official, maintained client that handles authentication, validation and error handling for you:

- [PHP SDK](#) - a typed client for any modern PHP app. `composer require html2img/html2img-php`
- [Laravel Package](#) - a facade, config file and storage helpers, built on the PHP SDK. `composer require html2img/html2img-laravel`
- [JavaScript SDK](#) - a zero-dependency TypeScript-ready client for Node.js, Bun, Deno and edge runtimes. `npm install @html2img/client`

Working in another stack? The [language guides](#) cover Ruby on Rails, Python, React and Vue with plain HTTP examples, and everything below applies to every language.

Which should I use?

Pick raw HTML for full control, the URL endpoint for screenshots of pages you already host, or named templates when you want to skip the markup step entirely.

Four endpoints

Three render endpoints that each consume one credit per image, and a free account status endpoint.

1. HTML and CSS API

```
POST https://app.html2img.com/api/html
```

Send raw HTML and CSS, get back a PNG. Run inline JavaScript up to a 30 second budget. Best for full design control.

2. Screenshot API

```
POST https://app.html2img.com/api/screenshot
```

Send a public URL, get back a PNG of the rendered page. Best for capturing pages you already host.

3. Templates API

```
POST https://app.html2img.com/api/v1/templates/[slug]
```

Send a JSON payload to a [named template](#) endpoint. Best for skipping the markup step.

4. Account Status API

```
GET https://app.html2img.com/api/me
```

Returns the account behind your API key: plan, credit balance and renewal date. It never consumes a credit and works even when your account is out of credits, so use it to verify a key or to check your balance before a batch. See the [account status docs](#) for the full response reference.

i For the Screenshot API, use the `webhook_url` parameter when render time is unpredictable. The 30 second sync timeout is enough for most requests, but a slow third-party page can blow past it. See [webhook_url docs](#).

For best results:

- Use `webhook_url` for slow Screenshot API requests
- Keep `dpi` at 1 unless you specifically need retina output
- Use sync requests for the HTML API or fast Screenshot requests

Required Parameters

HTML and CSS API

Parameter	Type	Description
<code>html</code>	string	The HTML content to render. Can include inline CSS and JavaScript.

Screenshot API

Parameter	Type	Description
<code>url</code>	string	The URL to capture. Must be a valid, publicly accessible URL.

Optional Parameters

Both APIs support the following optional parameters:

Parameter	Type	Default	Description
<code>css</code>	string	null	Additional CSS to inject into the page
<code>width</code>	integer	1440	Viewport width (1-5000 pixels)
<code>height</code>	integer	900	Viewport height (1-5000 pixels)
<code>fullpage</code>	boolean	false	Whether to capture the full page height
<code>dpi</code>	integer	1 (Screenshot), 2 (HTML)	Device pixel ratio (1-4). Fullpage captures always render at 1.
<code>webhook_url</code>	string	null	URL to receive a JSON callback with the image URL when the render finishes
<code>wait_for_selector</code>	string	null	Wait for a specific element to appear before capturing (does not work with iframes)
<code>ms_delay</code>	integer	null	Fixed delay in milliseconds before capture, 1-5000 (useful for iframes and embedded content)

The Screenshot API also supports:

Parameter	Type	Default	Description
<code>selector</code>	string	null	CSS selector to capture a specific element

⚠ Higher DPI values (2-4) significantly increase processing time and memory usage. We allow these values to support specific use cases, but they often lead to timeouts when not used with `webhook_url`. For most cases, a DPI of 1 provides enough quality.

Response Format

Successful Response

```
{
  "success": true,
```

```
{
  "id": "8a9dda43-5f42-4b93-8ff4-cd69ed32d402",
  "expires_at": null,
  "credits_remaining": 950,
  "url": "https://i.html2img.com/image-1786092598870-921691.png"
}
```

`expires_at` is `null` on paid plans, where renders stay hosted permanently. On the free tier it carries an ISO 8601 timestamp - free renders are hosted for 7 days, and upgrading makes every render still hosted permanent.

For when using `webhook_url`:

```
{
  "success": true,
  "id": "8a9dda43-5f42-4b93-8ff4-cd69ed32d402",
  "expires_at": null,
  "credits_remaining": 950,
  "status": "processing",
  "message": "Screenshot generation started",
  "url": "https://i.html2img.com/image-1786092598870-921691.png"
}
```

The `url` in a processing response is where the image will land - it is not live until the render finishes and your webhook fires.

The webhook payload, posted to your `webhook_url` when the render finishes, has its own shape. The render is identified by `log_id`, which matches the `id` from your initial response:

```
{
  "status": "success",
  "message": "Screenshot generated successfully",
  "url": "https://i.html2img.com/image-1786092598870-921691.png",
  "filename": "image-1786092598870-921691.png",
  "format": "png",
  "dpi": 2,
  "log_id": "8a9dda43-5f42-4b93-8ff4-cd69ed32d402"
}
```

If the render fails, the webhook receives `{"status": "error", "error": "...", "filename": "...", "log_id": "..."} instead - check status rather than assuming`

success. When the failure is something you can fix (an unreachable URL, a selector that never appeared), the payload also carries an `error_code` - the same codes as the [render error responses](#) below.

Account status response

`GET /api/me` returns an account snapshot rather than a render:

```
{
  "email": "you@example.com",
  "plan": "1k",
  "plan_name": "1,000 Credits",
  "active": true,
  "free_plan": false,
  "credits_remaining": 850,
  "credits_reset_at": "2026-09-01T00:00:00+00:00"
}
```

Every field is explained in the [account status reference](#).

Error Responses

Validation Error (400)

```
{
  "error": "Validation failed",
  "code": "validation_error",
  "details": {
    "html": ["The html field is required."]
  }
}
```

The Templates API returns `422` for validation failures instead; the body shape is the same.

Authentication Error (401)

```
{
  "error": "Invalid API key",
}
```

```
"code": "invalid_api_key"
}
```

A missing `X-API-Key` header returns `missing_api_key`. Verify a key without spending a credit via `GET /api/me`.

Out of Credits (402)

```
{
  "error": "Insufficient credits",
  "code": "insufficient_credits",
  "credits_remaining": 0,
  "message": "You have used your free credits. Upgrade to a paid plan to keep rendering.",
  "upgrade_url": "https://app.html2img.com/plans"
}
```

On a paid plan the response carries `credits_reset_at` (your renewal date) instead of `upgrade_url`.

No Active Plan (403)

```
{
  "error": "You must be subscribed to use this service",
  "code": "not_subscribed"
}
```

Render Error (422)

```
{
  "error": "Render failed",
  "code": "url_not_found",
  "message": "That web address could not be found. Please check the URL for typos and make sure the site exists.",
  "id": "8a9dda43-5f42-4b93-8ff4-cd69ed32d402"
}
```

Returned when the render fails for a reason you can fix. `message` is safe to show to your own users. Possible `code` values:

Code	Meaning
<code>url_not_found</code>	The URL's domain could not be resolved - usually a typo or a site that no longer exists
<code>url_unreachable</code>	The domain resolved but the page would not load (SSL failure, connection reset, too many redirects)
<code>connection_refused</code>	The site refused the connection - it may be down or blocking automated requests
<code>page_load_timeout</code>	The page took too long to load
<code>selector_timeout</code>	<code>wait_for_selector</code> never appeared within the time limit
<code>selector_not_found</code>	The <code>selector</code> element does not exist on the page
<code>screenshot_too_large</code>	The requested capture exceeds the size limits - reduce width, height or DPI

Timeout Error (504)

```
{
  "error": "Request timed out",
  "code": "timeout_error",
  "message": "Render job exceeded the allotted time. Consider passing
webhook_url for large captures.",
  "id": "8a9dda43-5f42-4b93-8ff4-cd69ed32d402"
}
```

A timeout waiting for the renderer itself returns the same status with `code: "api_timeout_error"` and no `id`.

Service Error (500)

```
{
  "error": "Service error",
  "code": "service_error",
  "message": "An internal error occurred while rendering.",
  "id": "8a9dda43-5f42-4b93-8ff4-cd69ed32d402"
}
```

A bug on our side - retry once, and if it persists [contact support](#) quoting the `id`.

Next Steps

- [Set up authentication](#)
- [Install an official SDK - PHP, Laravel or JavaScript](#)
- [See language examples](#)
- [Browse the templates gallery](#)
- [Read the parameters reference](#)



HTML to Image

Convert HTML, CSS and URLs into images and PDFs.

The HTML to image API.

Product

- [Features](#)
- [HTML to PDF](#)
- [URL to PDF](#)
- [MCP server](#)
- [Pricing](#)
- [Documentation](#)
- [Articles](#)

Compare

- [HTML to Image vs htmlcsstoimage](#)
- [HTML to Image vs Bannerbear](#)
- [HTML to Image vs Urlbox](#)
- [HTML to Image vs ApiFlash](#)
- [See all comparisons](#)

Company

- [About](#)
- [Contact](#)
- [Terms](#)
- [Privacy](#)

Image generation tools >

Certificate Generator

HTML to PDF Converter

Open Graph Image Generator

Quote Card Generator

URL to PDF Converter

Code Screenshot Generator

Invoice Image Generator

Pinterest Pin Generator

Twitter Card Generator

YouTube Thumbnail Generator

Image templates >

SOCIAL MEDIA

Facebook post

Instagram square post

Instagram story

LinkedIn post

Pinterest pin

Tweet mockup card

Twitter post

YouTube thumbnail

BUSINESS AND COMMERCE

Business card

Certificate of completion

Coupon and voucher

Event ticket

Invoice image

Real estate listing

Receipt image

CONTENT AND MARKETING

Blog hero image

Code screenshot

Email header

GitHub social preview

[Open Graph image](#)

[Podcast cover](#)

[Podcast episode card](#)

[Product card](#)

[Project showcase](#)

[Quote card](#)